ARWin - A Desktop Augmented Reality Window Manager

Stephen DiVerdi, Daniel Nurmi, Tobias Höllerer
Department of Computer Science
University of California, Santa Barbara, CA 93106
{sdiverdi,nurmi,holl}@cs.ucsb.edu

Figure 1. A typical ARWin desktop, as seen through a video see-through head-worn display (Sony LDI-A55 with Point Grey Firefly camera). Applications are (clockwise from right) weather report, tagged phone, business card, flowers, web browser, clock.

Abstract

We present ARWin, a single user 3D augmented reality desktop. We explain our design considerations and system architecture and discuss a variety of applications and interaction techniques designed to take advantage of this new platform.

1. Introduction

For the past twenty years, desktop computer use has been dominated by the 2D Windows-Icons-Menus-Pointing (WIMP) paradigm. Through major advances in computing power, 3D graphics and tracking support, it is now possible to create 3D augmented environments with unprecedented ease. Standard desktop computers, coupled with a head-worn display and camera, are now equipped for the task of presenting the user with an augmented version of his physical desktop workspace. We have developed such a 3D environment, which allows both legacy 2D applications and novel 3D applications to populate the physical space above the user's desk, some of which enhance the functionality of the physical objects on the desk (see Figure 1). To further the integration of these applications into the augmented environment, we have also implemented some intuitive interaction techniques appropriate to 3D.

Our system is possible thanks to research in user interfaces and augmented environments. We rely on work advancing vision-based tracking and tangible interfaces [3], migration of 2D windows into 3D environments [6], 3D data visualization [5], and 3D AR workspaces [7]. We are us-

ing concepts from Windows on the World [2], to integrate legacy 2D applications in the augmented environment, as well as the Tiles system [3], to provide a tangible interface to our augmented components. The result is a novel generic application architecture for general purpose computing. While AR lends itself very well to multi-user collaborative work [1, 7], our scenario purposefully focuses on support of the single-user case, which is how most computer users spend the majority of their time.

Our prototype environment, ARWIN, allows the user to work in a familiar fashion with traditional 2D GUI applications, while introducing novel applications that are developed specifically with the 3D augmented workspaces in mind. These applications can mimic or extend traditional desktop objects such as a clock or calendar, or can spatially visualize information, such as web or file hierarchies. Thanks to the extra dimension in a volumetric workspace, these applications can also interact in more intuitive ways, based on physical relationships such as proximity. The result of our work is the core ARWin architecture and the applications we developed to showcase its capabilities.

2. Architecture

The core of the ARWin architecture consists of two components – the event manager and the display manager (see Figure 2). Together, they replace the functionality of a regular X Windows environment running a window manager such as TWM or Sawfish.

The event manager's responsibility is to process input and generate ARWin specific events. For example, the camera input prompts the event manager to perform marker recognition and tracking. Markers affixed to the desk and walls in the user's workspace are detected by the ARToolkit [3]. This provides a global coordinate space in the volume above the user's desk. Other input, such as keyboard and mouse events, can either be passed along to the display manager unmodified or may trigger more complicated events, such as window dragging or menu interaction.

The display manager's responsibility is to use the application knowledge base to determine where events should be delivered. The *draw* event, for instance, must be delivered to all visible applications while keyboard and mouse events need only be delivered to the application that is currently in focus.

An ARWin application is implemented by extending a generic App3d superclass that provides callback functions for event handling, as well as generic window manager services. A special case of application, called AppWin, allows users to operate legacy X applications (xterm, emacs, etc.) within ARWin. AppWin runs the X application on a virtual VNC enabled X server [4]. Our custom VNC client connects and draws the image data to an OpenGL texture map,
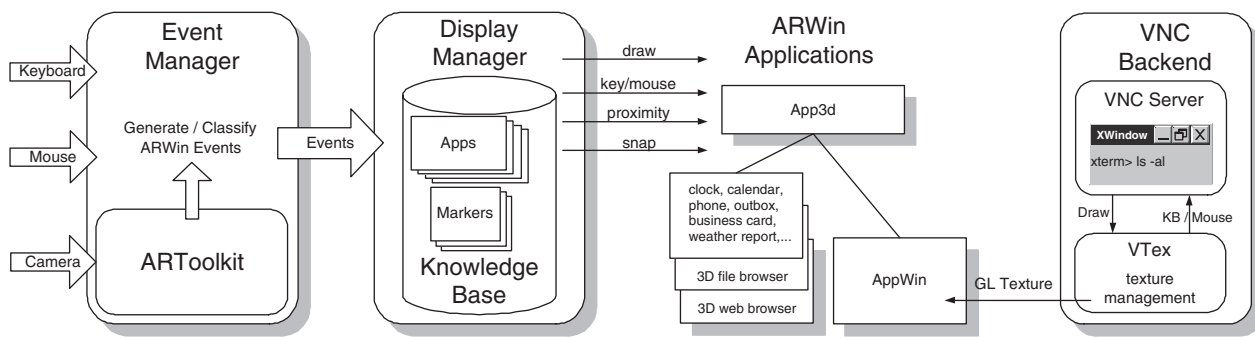
Figure 2. ARWin system architecture. The event manager turns hardware input into events. The display manager distributes events among applications which subclass App3d. AppWin uses a texture map of a VNC server to display X-applications.

which is applied to a polygon in the workspace. To maximize application visibility, the focused application is always oriented to face the user head-on.

3. Applications and Interactions

Developing intuitive 3D applications and their interactions constituted a major part of our work (see Figure 1). ARWin applications can be categorized based on their geometry and level of physicality. Some are completely virtual, such as the enhanced web browser which uses a cone tree [5] to visualize the current web hierarchy. Other applications are simply tagged physical objects, such as the phone or business card, which provide ARWin with knowledge about these objects so they may interact in the virtual environment. There are virtual applications that mimic physical objects to provide enhanced functionality, such as the clock and flowers. Finally, there are hybrid objects such as the weather report, with virtual components (sunny/cloudy indicator) that enhance otherwise functional physical components (wall thermometer).

For application interaction, ARWin, similar to [7], generates proximity events. Objects which contain useful metadata are called data containers and are represented by an orange arrow. Objects which can operate on metadata are data handlers and are represented by a blue arrow. When a handler and a container are placed near each other the metadata is transferred so the handler can process it (see Figure 3).
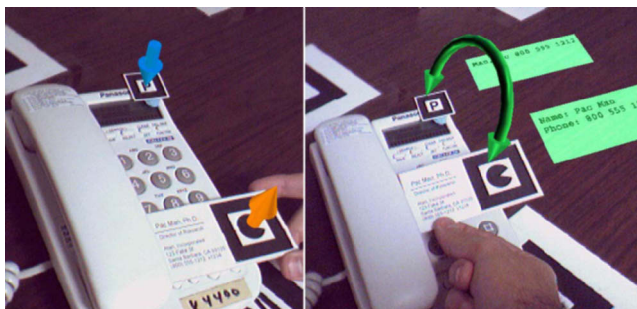

Figure 3. Application interaction. *left*: **phone (handler) and business card (container).** *right*: **phone receives card data, displaying and storing relevant information.**

For some applications, such as a web browser or xterm, it is appropriate to provide two modes of operation - one in 3D and one in 2D. For these applications, there is a *snap* event, which snaps the application to the full display resolution, mimicking a traditional 2D interface.

4 Conclusions

We have presented ARWin, a research augmented reality desktop environment. We have demonstrated ARWin's ability to execute traditional X applications as well as novel applications which showcase our interface enhancements.

In the future, ARWin will be an invaluable testing environment for technology integration. We will explore user interaction techniques such as gesture recognition, along with more advanced application interaction and data visualization technologies. Tracking is currently limited by camera resolution and the careful calibration of markers, and could be improved by investigating hybrid tracking approaches. We would also like to address the low resolution of available HMDs by allowing applications to snap to a high-resolution desktop LCD monitor instead of the HMD. 3D holographic imaging systems could replace the HMD in the long run.

References

[1] A. Butz, T. Höllerer, S. Feiner, B. MacIntyre, and C. Beshers. Enveloping users and computers in a collaborative 3D augmented reality. In *IWAR '99*, pages 35–44, San Francisco, CA, Oct. 20–21 1999.

[2] S. Feiner, B. MacIntyre, M. Haupt, and E. Solomon. Windows on the world: 2D windows for 3D augmented reality. In *ACM UIST '93*, pages 145–155, Atlanta, GA, Nov. 3–5 1993.

[3] I. Poupyrev, D. Tan, M. Billinghurst, H. Kato, H. Regenbrecht, and N. Tetsutani. Developing a generic augmented-reality interface. *Computer*, 35(3):44–50, Mar. 2002.

[4] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper. Virtual network computing. In *IEEE Internet Computing*, pages 33–38, Jan./Feb. 1998.

[5] G. Robertson, J. Mackinlay, and S. Card. Cone trees: Animated 3D visualizations of hierarchical information. In *ACM CHI'91*, pages 189–194, 1991.

[6] G. Robertson, M. van Dantzich, D. Robbins, M. Czerwinski, K. Hinckley, K. Risden, D. Thiel, and V. Gorokhovsky. The task gallery: a 3D window manager. In *ACM CHI '00*, pages 494–501, Apr. 1–6 2000.

[7] D. Schmalstieg, A. Fuhrmann, and G. Hesina. Bridging multiple user interface dimensions with augmented reality. In *IEEE/ACM ISAR 2000*, pages 20–29, Oct. 5–6 2000.

IEEE
COMPUTER
SOCIETY