WireAR - Legacy Applications in Augmented Reality

Gerhard Reitmayr[*], Mark Billinghurst[†] and Dieter Schmalstieg[*]
[*]Vienna University of Technology [†]The HIT Lab NZ, University of Canterbury
{reitmayr|schmalstieg}@ims.tuwien.ac.at, mark.billinghurst@hitlabnz.org

Abstract

Current Augmented Reality (AR) applications require that the application software be written to support a specific AR interface set up. WireAR was developed to enable output from any OpenGL application to be viewed in an AR fashion. This enables the output from any legacy graphical or scientific visualization applications to be viewed in a collaborative AR setting. This demonstration shows how the output of standard desktop visualization programs can be embedded into an augmented reality experience.

Keywords: Augmented Reality, Computer Supported Collaborative Work, OpenGL.

1 Introduction

Augmented Reality is a powerful user interface metaphor that extends our current spectrum of interaction with computers. However the special needs of this technology such as integrating sensor input from non-standard devices or rendering and output formats for head worn displays require the development of dedicated software components and applications to deal with these topics.

Opposed to the requirement of developing dedicated software, there already exists a wide range of standard visualization software packages for desktop environments such as AVS or VTK. Adapting such applications to new interface modalities such as AR can range from difficult to impossible, because they do not provide necessary programming interfaces or are only available in executable form.

This work demonstrates a new software technology based on OpenGL to integrate such legacy applications into an AR user interface. The main idea is to use the information about 3D visualizations transported via the OpenGL interface to render the visualizations within the AR user interface (see Figure 1).

2 Implementation

In the WireAR project any OpenGL based 3D visualization application can work as the geometry server for an

Figure 1. In this image a fish tank model is presented on a tangible marker. The simulation of the fish swimming in the aquarium is computed and rendered by a legacy OpenGL application.

AR application. This is accomplished by intercepting the function calls to the OpenGL rendering pipeline and transmitting them to a dedicated AR interface application. It transforms the stream of function calls and executes them to present a unified AR interface. This work is based on the WireGL [2] work of Stanford University that allows any OpenGL application to display its output on a tiled projector wall screen display.

An improved version of the WireGL software called Chromium [3] was integrated into the AR software framework Studierstube [5]. In an appropriate configuration the Chromium framework intercepts the OpenGL calls made by the legacy application. Then it forwards the resulting command stream to a dedicated 2D rendering window. It also generates a copy of the stream that is send to the Studierstube framework (see Figure 2).

Within the Studierstube framework we extended the underlying scene graph library OpenInventor to include a dedicated geometry node that renders the incoming stream of OpenGL commands during a render traversal of OpenInventor. It suppresses certain OpenGL commands such as
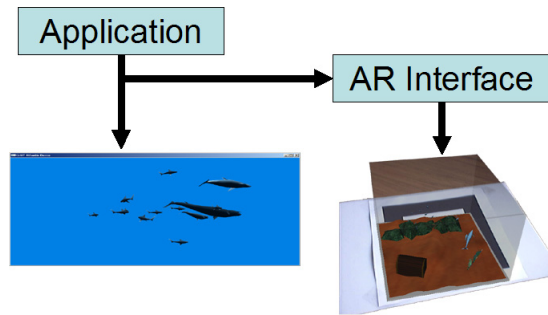
Figure 2. This figure describes the data flow of OpenGL commands through the application. Chromium renders the incoming command stream into the application window and copies it to the AR interface application.

setting the viewport, manipulating the view matrix, controlling buffer tests and others to avoid interference with the standard operations of the scene graph library. In addition to that it also manipulates the model matrix stack to render the geometry at the correct location with respect the current transformations in the scene graph.

Not all information can be captured that way. The traditional GUI interface of the original application is not presented in the AR interface, unless it is drawn using OpenGL. Moreover the use of OpenGL itself is limited to pure 3D models. Advanced methods such as multi-pass rendering, screen aligned graphics and direct usages of the various buffers are obstructed by the fact that the rendering state needs to integrate with the OpenInventor rendering.

3 Demonstration

The demonstration setup consists of a computer with a standard desktop interface consisting of a screen, keyboard and mouse. A head mounted display with a camera attached to it is connected to a second video output of the computer. This allows it to serve two different images for the desktop interface and the augmented reality interface. The camera provides the video image for video-see-through augmented reality which is presented in the HMD and for tracking of fiducials with the ARToolKit [4] library.

The AR interface application presents the content of the visualization application in a manner similar to the SharedSpace [1] project. Users wear a head mounted display with a video camera attached. In the display they can see video of the real world with computer graphics overlaid on it. Moreover they can manipulate their view of the content by a tangible user interface.

Users interact with the demonstration in one of two roles. One user interacts with the legacy application directly in the operator role. A second user interacts with the AR user interface in the spectator role.

Operator role

The computer runs and displays a standard desktop 3D application. Thus, a user of the desktop system is able to manipulate and interact with the application in a standard way by relying on the features of the WIMP user interface. She acts as the operator in this scenario.

Spectator role

Simultaneously, the output of the application is captured by the WireAR software and presented in the AR view. This view is displayed on a tangible marker that can be manipulated by the user of the AR display. Such an interface provides a natural means of interacting with the three dimensional content of the application.

4 Future work

In future work we want to investigate how to incorporate feedback from the AR user interface into the legacy application. In addition to that we want to improve the integration of the legacy application's output into the 3D environment by allowing more graphical interaction between the scene graph based rendering framework and the OpenGL stream from Chromium framework.

Acknowledgments

References

[1] M. Billinghurst, S. Weghorst, and T. Furness. Shared space: An augmented reality interface for computer supported collaborative work. In *Proc. of Collaborative Virtual Environments Workshop '96*, Nottingham, Great Britain, September 19–20 1996.

[2] G. Humpreys, M. Eldridge, I. Buck, G. Stoll, M. Everett, and P. Hanrahan. WireGL: A scalable graphics system for clusters. In *Proc. SIGGRAPH 2001*. ACM, 2001.

[3] G. Humpreys et al. Chromium. http://www.cs.virginia.edu/ humper/chromium_documentation/, visited April 15th, 2003.

[4] H. Kato and M. Billinghurst. Artoolkit. http://www.hitl.washington.edu/artoolkit/, May 2003.

[5] D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. Szalavari, L. M. Encarnao, M. Gervautz, and W. Purgathofer. The Studierstube augmented reality project. *PRESENCE - Teleoperators and Virtual Environments*, 11(1), 2002.