A Demonstration of Mobile Augmented Reality

Mark Assad, David J. Carmichael, Daniel Cutting, Adam Hudson
University of Sydney
{*massad, dcarmich, dcutting, ahudson*}*@it.usyd.edu.au*

Abstract

This demonstration paper presents the AR phone, a mobile phone acting as an augmented reality (AR) interface into an intelligent environment. We describe the design and implementation of a prototype system whereby a phone can capture images, transfer them to an intelligent environment and receive some form of visually augmented result. In conclusion we explore some demonstrations of the system.

1. Introduction

Intelligent environments (IEs) aim to complement users' daily experiences by providing them with computational resources and services that blend seamlessly into their surroundings. One of the most important aspects of such an environment is how users interact with it. This should be as intuitive and unobtrusive as possible, which makes traditional computing interfaces such as keyboards, mice and monitors undesirable. Techniques that allow a user to interact with an IE in a more natural manner are preferable. Augmented Reality (AR) offers much promise in this area.

Current AR interfaces are typically head mounted displays (HMDs) or web cams attached to PCs. Both of these have significant drawbacks. HMDs are expensive and thus *inaccessible* to ordinary users. Web cams are *non-intuitive* since the input device is usually quite separate to the display device.

The ubiquitous technology of the mobile phone is a plausible platform for deploying AR-based applications. In particular, the emerging generation of phones with colour screens, cameras and wireless networking have all of the capabilities required to implement an AR interface, although they may not have the processing power to perform the augmentation internally.

This paper describes the implementation of the AR phone, an accessible and intuitive system built around a mobile phone which can be used to present an AR interface into an intelligent environment. We discuss the implementation of a prototype system and look at where the project is headed. Further technical details on the background and design of the AR phone can be found in (1).

2. Architecture

Our initial AR phone architecture is a simple, distributed prototype system comprising three components:

- an interface application running on a phone
- wireless access points distributed throughout the physical environment
- a centralised augmented reality service module.

The phone acts as a 'viewport' of an AR service, with the bulk of the processing taking place in an IE. This keeps the applications on the phone simple and allows them to be deployed across a range of phones and other devices with differing input and output capabilities. The phone takes a video stream and relays it wirelessly to one or more access point modules deployed around an IE. The access point modules are responsible for disseminating the received input to a centralised augmented reality service module. This processes the raw image input and returns the augmented results to the access point, which in turn transmits it back to the phone for display to the user. The process is illustrated in Figure 1.

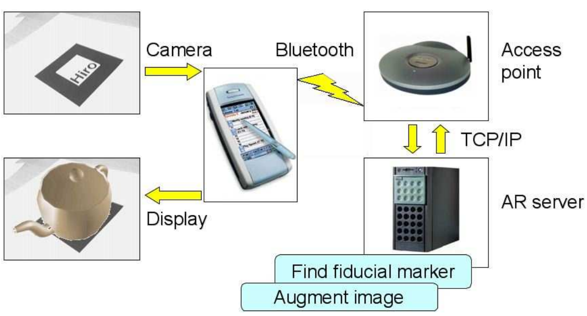

Figure 1: The AR phone pipeline.

3. Implementation

3.1. Phone Application

The application running on the phone is designed in a similar fashion to the AR-PDA (2) and Handheld AR (3) projects, although these both use PDAs as distinct from phones. The application captures an image, sends it to an access point via Bluetooth, receives the augmented result and then displays it on the phone's colour display. The application has initially been developed on a Sony Ericsson P800 mobile phone running version 7.0 of the Symbian OS. The P800 is a good choice since it includes a built-in camera, a large colour screen and Bluetooth communication in the standard model. As of August 2003 there is no support for capturing video from the internal camera: it supports only the capture of snapshot images. We hope to be able to use a video stream in the future.

The application is able to scan its local area for any available Bluetooth access points which are then displayed to the user for selection. Once an access point is selected a L2CAP connection is established between it and the phone, which the application uses to ultimately communicate with the AR server.

3.2. Access Points

The application running on the phone connects to the AR service though the access points. These access points have been developed to run under Linux with the BlueZ Bluetooth stack.

3.3. AR Service

The AR service is a generic module that is able to receive image input from the access points and perform processing and manipulation of the data. The specifics of the processing are dependent on the service provided. The decoupling of the service processing allows for multiple AR applications to be developed for the phone using common access points and user interfaces. After processing of the image, the result is returned to the access point requesting the service, which then passes it onto the phone.

4. Demonstration

We implemented a basic AR service for our prototype which simply augments a raw image with a three dimensional object. The image augmentation service examines the input image for a particular fiducial marker which, if found, is overlaid in three dimensional space with a virtual teapot. This is achieved with the AR toolkit (4) and the OpenGL GLUT library. The largest bottleneck in this system is the transmission of the uncompressed bitmap image over Bluetooth: Depending on the chosen input resolution, the round trip time varies from two to 15 seconds.

5. Future Directions

In the near future we hope to create user interface modules for other mobile devices including the Nokia 7650 mobile phone and the Toshiba e740 PDA with an attached camera. In the case of the latter, the use of a higher bandwidth 802.11b wireless connection instead of Bluetooth should lead to a great performance improvement in our prototype implementation.

It should also be possible to improve performance by applying compression algorithms to the data transmitted over Bluetooth. Since the data forms an image, we are able to use lossy compression techniques like JPEG which should significantly reduce the amount of traffic.

It may also be possible to move some parts of the AR toolkit into the user interface module such as the thresholding of images or detection of markers. This would mean far less data would need to be transmitted over Bluetooth and should increase performance. It should be noted that this approach is a hybrid computational model which differs from the conceptually clean approach of an IE performing all computation "within the walls".

References

[1] Assad, M., Carmichael, D. J., Cutting, D., Hudson, A. AR phone: Accessible Augmented Reality in the Intelligent Environment. *To appear at OZCHI 2003, November 26-28, 2003, Brisbane, Australia.*

[2] Geiger, C., et al. Mobile AR4ALL. *http://citeseer.nj.nec.com/kleinjohann01 mobile.html*

[3] Interactive Media Systems Group, Institute for Software Technology and Interactive Systems, Vienna University of Technology *http://www.ims.tuwien.ac.at/research/ handheld_ar*

[4] Kato, H. and Billinghurst, M. (October 1999). Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System. *Proceedings International Workshop on Augmented Reality* (IWAR99).