

Interaction Modes for Augmented Reality Visualization

Hannah Slay, *Matthew Phillips, *Dr Rudi Vernik, Dr Bruce Thomas

Wearable Computer Lab
School of Computer and Information Science
University of South Australia
Mawson Lakes 5095, South Australia

*Defence Science and Technology Organisation
Salisbury 5108, South Australia

Slahy001@students.unisa.edu.au, Matthew.Phillips@dsto.defence.gov.au,
Rudi.Vernik@dsto.defence.gov.au, Bruce.Thomas@unisa.edu.au

Abstract

In this paper we describe a novel use of augmented reality for information visualization. We detail the use of augmented reality as a component of InVision — an open framework for the development and deployment on visualization systems. The research discussed in this paper is part of an ongoing project into pervasive computing.

1 Introduction

The current trend towards pervasive computing [1] suggests that future work environments will comprise a range of information display and interaction devices. This will include the use of more conventional approaches such as notepad computers and PDAs together with 3D immersive displays, augmented reality approaches, and other personal display devices such as IBM's bluetooth enabled LINUX watch [2]. Consistent approaches to interaction in these heterogeneous environments will be important if people are to make best use of the information available. We believe 2D and 3D information utilization is crucial to making this information accessible to people. This paper presents our investigation of extending interactions from a traditional desktop interaction paradigm to a tangible augmented reality paradigm.

The visualization system we extended is InVision. InVision (Pattison T.R., Vernik R.J., Goddburn D.P.J. and Phillips M.P., 2001) is a component based visualization environment developed by DSTO and research partners to investigate a range of issues related to the rapid assembly and deployment of adaptive visualization systems. Issues being investigated include view management and sharing, novel visualization approaches, view interaction, process-based visualization, and intelligent agent support. InVision has been used as the research platform for the work discussed in this paper,

which looks at the integration and use of Augmented Reality (AR) views.

Augmented reality refers to the process of using a head mounted display to overlay computer-generated imagery to the real world. This allows users to visualize three-dimensional objects in the real world and interact with them in a natural way. The computer-generated overlays provide additional information to the user to enhance the real world. Augmented reality approaches provide the potential for generating new types of visual forms. For example, the approach discussed in this paper makes the use of tangible fiducial marker cards, see-through head mounted displays in a form factor of glasses, wearable computers, and position-orientation tracking approaches to allow the generation of 3D images in space within real contextual surroundings. The potential benefits of these types of views over traditional virtual reality include better human interaction, contextual awareness, and less likelihood of simulator sickness. Moreover, these types of views could be generated anywhere, without the need for costly and rigid display and tracking infrastructure. This form of AR technology aligns well with the concept of pervasive computing which aims to 'unleash' the user from the computer and provide a rich integrated ambient environment.

Selection of objects is fundamental for interactions with objects. Many different methods have been suggested, but these can all be broken into two categories, ray casting and arm-extension. Ray casting allows the user to select objects at a distance by shining a virtual ray from their hand / head or any other pointing device. The user selects the object when a collision occurs between their ray and one of the virtual objects. Arm-extension techniques allow the user to select objects by providing them with a procedure to extend the virtual representation of their arms. When the arms have 'grown' to a suitable length, selection can be performed with objects within the users reach. Most research has been performed where objects are large and out of reach. Some techniques can be used from both the ray casting and arm-extension categories, but most are not applicable.

This paper discusses our research investigating the issues and problems related to interaction modes within

pervasive computing environments. In particular, the paper looks at interaction approaches when using AR

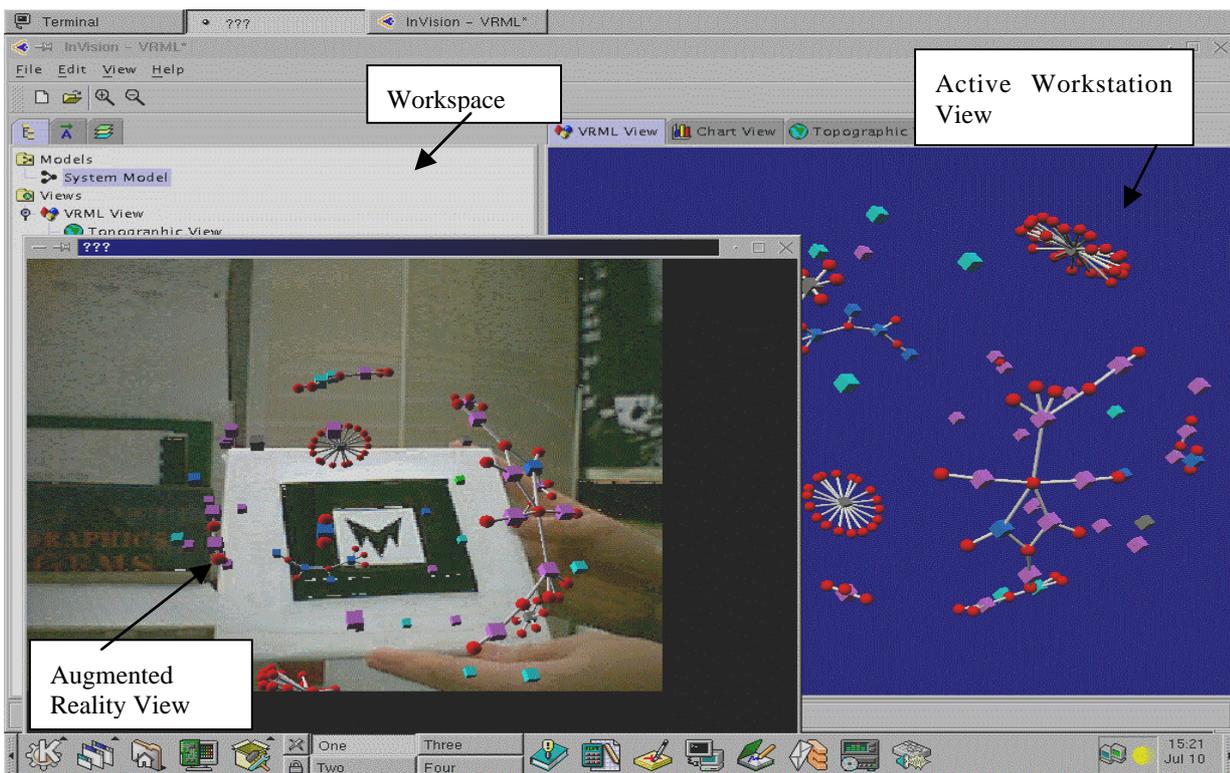


Figure 1 An example of an InVision workspace that shows the organizational structure of a large integrated research program. The workspace manages the various types of views, their interactions, and enhancements. This figure shows a screenshot of the workstation screen that shows the VRML view displayed and the same view as seen through the AR glasses where the view is generated in space relative to a fiducial marker

views as a complement to more conventional computer workstation views. We begin by discussing how an AR capability was integrated into InVision to provide the experimental platform. We then discuss issues related to AR view interaction, including related work. Our initial work has focused on selection approaches for 3D AR views. We discuss an approach that we have developed which uses ray casting to select objects in a 3D VRML graph structure. We then describe the use of fiducial markers to change the attributes of virtual objects. We then conclude with some final remarks and directions for future work.

2 Integrating AR Views into InVision

InVision uses the concept of workspaces to support the management and integration of views for particular individuals or roles (refer Figure 1). These views specify how the information in an underlying data model will be represented. InVision models can be thought of as attributed graphs that capture information about a set of things, the various relationships between these things, and sets of attributes that describe the things (eg size, colour, name). The modelling approach is generic and so can be used to support the visualisation of a wide variety of artefacts such as software systems, intranets, organisational structures, social networks, or financial systems.

A variety of visual forms could be used to represent various viewpoints of the model. These might include cluster graphs, charts (scatter, bar, kiviati), tabular grids, or textual representations (or indeed combinations of these). InVision already has components that allow for the specification and use of a wide variety of visual representations. These views can be displayed on one or more conventional display devices (eg workstation screen, large screen projection). Interaction with these more traditional visual forms is via direct manipulation interfaces using devices such as a mouse.

We have extended InVision with components that allow us to display any of the views in an InVision workspace through an AR tangible interface. The AR view capability was added to InVision by incorporating the University of Washington's Human Interface Technology Lab ARToolkit Version 2.11 (Kato, H. and Billinghurst, M., 1999) as a component into the InVision framework. Figure 1 shows an example where a VRML view has been selected from an InVision workspace for display as an active view on the workstation screen. Right clicking on the active view tag allows the user to also display this view in AR mode relative to a particular fiducial marker pattern. Figure 1 also shows the resulting view as seen through the AR glasses. The user can then inspect the model by rotating it, changing the azimuth etc. This is particularly useful for three-dimensional models. By rotating the marker, the user can examine the model from

all viewpoints. By bringing the marker closer or further from the user, their viewpoint can be zoomed in or out. This provides an interaction that can be closely mapped to the natural interactions between humans and the objects that they want to analyze.

As shown in Figure 2, a fiducial marker is simply a pattern that the AR Toolkit component is trained to recognize. Before the marker can be used, the program must be trained to recognize the pattern. The pattern details its actual size and black and white pixel content. This is used later to detect the presence of markers in the scene. When the program is running, template matching is performed to determine if any markers are present. When the program detects a marker, the actual position is calculated using the data stored in the templates and the size of the marker from the input stream. This is then used to scale the graphical image so it appears to be attached to the marker.



Figure 2: Virtual overlay of object on marker, input video and thresholding video

Figure 2 shows a fiducial marker with the three stages of computation. The lower left hand side picture shows the input to the program. A fiducial marker is clearly visible in this shot. The lower right hand side picture shows the thresholded version of the input. Red squares are drawn around all squares that the program recognizes as patterns. And the top picture shows the output shot from the program. The VRML model is positioned to lie with its center at the center of the pattern.

3 Interaction with AR Visualization

Most comparable research into interaction with AR has been performed using immersive virtual reality with objects that are out of arms reach. We are concerned with mixed reality and investigating fine selection of objects that are all within reach. Mixed reality is more suitable

for this application because of the users inherent proximity to a traditional workstation. We are only interested in creating new interactions if they can be seen to be more efficient or easier for the user to perform. If a suitable interaction technique already exists in the traditional paradigm, we are not interested in creating a new, less efficient technique.

As noted by Poupyrev et al (Poupyrev, I., Weghorst S., Billinghamurst M. and Ichikawa T., 1998) mixed reality interaction techniques can be separated into two categories, exocentric and egocentric metaphors. Exocentric interaction occurs when the user interacts with the scene from outside the Virtual Environment (VE). Egocentric interaction occurs when the use is imbedded in the VE. This paper is only concerned with egocentric interaction techniques.

Egocentric interactions can be further broken into two distinct categories. These are arm-extension and ray-casting techniques. Poupyrev's Go-Go technique (Poupyrev, I., Weghorst S., Billinghamurst M. and Ichikawa T., 1998) is an example of arm-extension. The flashlight technique (Bowman, D., 1997) and Zhai's Silk Cursor (Zhai, S., Buxton W., Milgram P., 1994) are examples of ray casting. Poupyrev's Go-Go technique (Poupyrev, I., Weghorst S., Billinghamurst M. and Ichikawa T., 1998) is an example of a virtual hand or an arm extension technique for selecting and manipulating objects. The Go-Go technique allows the user to select objects that were previously out of reach by growing the virtual representations of the user's arms. When the user has their arms within two thirds of their arm length, the interaction is considered to be between the user and the nearby objects. However, if the user extends their arms beyond this region, the virtual representation of the arm extends in the direction that the users arm is pointing.

The flashlight technique makes use of a virtual light ray that is positioned in the direction that the user's hand is facing (Bowman, D., 1997). This technique is useful for selecting groups of objects at large distances, but becomes complicated with small objects at large distances.

Zhai's Silk Cursor (Zhai, S., Buxton W., Milgram P., 1994) is a related interaction technique. Instead of having a two dimensional cursor for selecting objects, Zhai proposes a three-dimensional semi-transparent volume. The volume is semi-transparent to give clues of occlusion and depth perception in target acquisition.

Wloka (Wloka, M., 1995) suggests the use of a Virtual tricorder type device that can perform multiple tasks using the one instrument. The virtual representation of the device is controlled by a physical instrument that has a tracking device attached to it. The user can load different tasks onto the device and select the appropriate action by pressing a button that is associated with the task.

Unlike most projects, instead of concentrating on one mode of interaction we are looking at several different modes and choosing the one that is the most efficient and natural for the user. This aims to provide seamless

interaction between the conventional workstation and augmented reality.

Like the Virtual Tricorder (Wloka, M., 1995), we are using one device as the main interaction tool with our virtual objects. The function that the tool provides depends on the button that is being pressed at the time of its use. Selection of the objects is undertaken using a type of ray casting from the pointing device. The Virtual Tricorder has appeal because of its multi-modal usage. Instead of providing one tool for each function, the one tool could perform multiple tasks. Its functionality could also be extended to be a universal tool in a heterogeneous environment.

4 Selection in 3D AR Views

This paper focuses on the selection of objects in augmented reality views. This interaction was considered to be one of the most fundamental, and therefore deserving of the initial attention. Most other interactions have behavior that builds on selection of a group or individual object

There are many advantages in using fiducial markers for this project. The primary advantage of fiducial markers is their intrinsic tracking ability. The unique, asymmetric pattern used to identify the marker also allows the user to determine the position of each marker in the field of view. And secondly they provide an interaction that can be closely mapped to the natural interactions between humans and objects they wish to analyse

By rotating the marker, the user can examine the model from all viewpoints. By bringing the marker closer or further from the user, their viewpoint can be zoomed in or out.

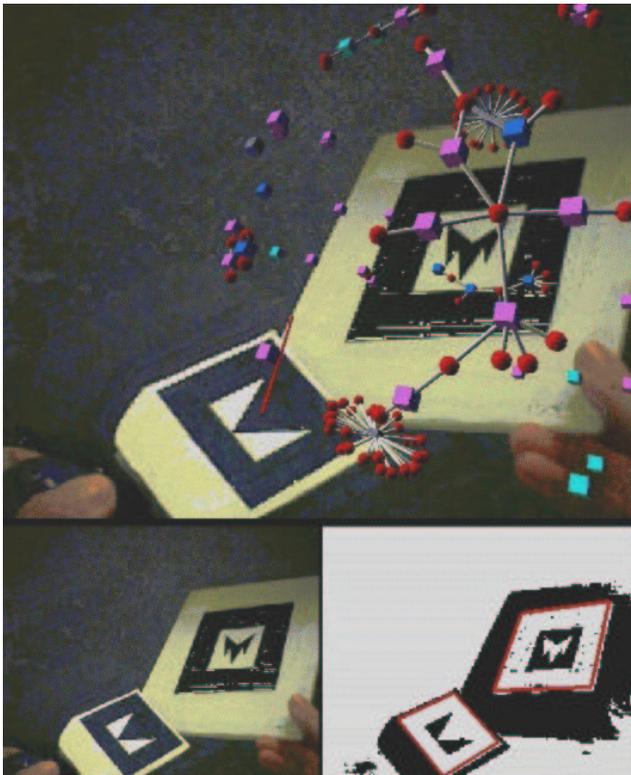


Figure 3: Selection of object using pointing device

In our approach, a fiducial marker is attached to the pointing device to track its movement (see Figure 3). When the button on the device is pressed, a virtual ray (as seen through the glasses) is generated relative to the fiducial marker attached to the device. The user then moves and controls the ray until it points to the desired object. The object can then be selected. As well as the advantages of this method, there are some related disadvantages. As with all markers, the pattern on the fiducial marker attached to the pointing device must be unique enough for the program to recognize it under all lighting conditions and when rotated through large angles.

Experimentation has been undertaken into the size and shape of the pointing device. Both arm extension and ray casting techniques were used to find the optimum pointer shape and size. A long cylindrical pointer of fixed length has been determined to work adequately for this task.

Figure 3 shows an object being selected using the pointing device. The lower left picture shows the input to the program. Two fiducial markers are clearly visible in this shot, one corresponding to the pointing device and the other to the center pattern. The lower right picture shows the thresholded version of the input. Squares are drawn around both squares that have been recognized as patterns. And the top picture shows the output shot from the program.

All models have been created using VRML. This modeling language was chosen in part because it is inherently supported by the ARToolkit, and because of its support for collision detection. It was later discovered that VRML does not support object collision detection between two objects, only between the object and avatar (camera). This led us to design and implement our own collision detection mechanism. This mechanism allows us to monitor collisions between the pointer and the virtual models, and also allows us to find which part of the model the pointer has collided with.

The ARToolkit was chosen because of its provision for real-time tracking of fiducial markers and its support of VRML. We chose to extend it to provide the capabilities for selection and manipulation of nodes.

5 Switching Attributes using Fiducial Markers

Many different techniques were considered for the task of switching the attributes of the virtual objects on and off. Three different techniques that were researched will be discussed. In each experiment, the same virtual object and the same fiducial markers were used so any difference in ease of operation could be attributed to the technique as opposed to the program's ability to recognize the fiducial markers or the characteristics of the virtual object.

5.1 Array of Markers

Firstly the fiducial markers were simply arranged on a surface. To change one of the attributes of the virtual object the associated fiducial marker was covered so it

was no longer recognized by InVision. To turn the attribute back, another marker associated with the opposite of the task that was performed first was covered so it could no longer be recognized by InVision.

5.2 Menu System

Next the fiducial markers were used as a menuing system. Five small markers were created to represent five menu options. Each layer of the menuing system had a limit of five options. To select one of the options, the card would be flipped over. The virtual representations of the cards would then change to show the next layer of the menuing system.

5.3 Toggle Switches

And finally the fiducial markers were also used as toggle switches. This technique can be seen to stem from the first technique that was examined. The fiducial markers had a pattern on each side of the card. The two patterns were associated with tasks that would provide opposite functions. For example, figure 4 shows a card that connects the nodes of a directed graph together when the first side is shown, and disconnects the nodes when it is flipped over.

The system remembers the state that each option is in. InVision continually checks to see if any cards are present in the field of view. If any changes have been made it generates the appropriate VRML model and displays this as the virtual object.

5.4 Findings

The first method was easy to use, but it was found to have some rudimentary flaws associated with it. Firstly, all markers must be in the field of view at all times. If any cannot be seen InVision assumed they were being selected as tasks to perform. This became confusing if two opposite tasks were waiting to be performed at the same time.

The second method was very restrictive on the design of the menu system. It was also very cumbersome to reach tasks in the lower layers of the menu hierarchy. Also, because of the limited number of tasks to choose from, the technique seemed very heavy-handed.

The toggle switch system was chosen because of its ease of use. Out of the three methods described in this section it was found to be the most easy and self-explanatory to use. Turning the cards over, in effect toggling the behaviour exhibited by the virtual models seemed very natural. Also, because there are half as many fiducial markers, the system seemed less cumbersome than the first method.

6 Conclusions and Future Work

The use of fiducial markers for both displaying a view and for enabling selection within the view proved beneficial for this project because of their inherent tracking ability and their ease of use. As we have shown, they are particularly useful for three-dimensional models.

Other interaction mechanisms such as creation and deletion of overlays and specifications will need to be explored. Also more experimentation will need to be performed into mixed modal interaction between traditional workstations and augmented reality. Additional research into the extension of the functionality performed by the central pointing device will also be undertaken.

7 References

NIST Pervasive Computing Overview:

http://www.nist.gov/pc2001/about_pervasive.html

IBM Bluetooth Watch.

http://www.trl.ibm.com/projects/ngm/index_e.htm

PATTISON T.R., VERNIK R.J., GODDBURN D.P.J. and PHILLIPS M.P. (2001) Rapid Assembly and Deployment of Domain Visualisation Solutions, *DSTO-TR-1100*,

BOWMAN, D., (1997) An Evaluation of Techniques for Grabbing and Manipulating Remote objects in Immersive Virtual Environments, *Interactive 3D Graphics*, 35-38

KATO, H. and BILLINGHURST, M., (1999), Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System, *In 2nd IEEE and ACM International Workshop on Augmented Reality* San Francisco USA 85-94

POUPYREV, I., WEGHORST S., BILLINGHURST M. and ICHIKAWA T., (1998) Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques, *Eurographics*, 17:41-52

POUPYREV, I., BILLINGHURST, M., WEGHORST S. and ICHIKAWA T., (1996) The Go-Go Interaction Technique — Non-linear Mapping for Direct Manipulation in VR, *UIST'96* :79-80

GOODBURN, D.P.J. and VERNIK, R.J., (1999) Deploying Systems Visualisation Environments: An Agent-Based Approach, *Proceedings of the Third International Conference on Knowledge-Based Intelligent Information Engineering Systems*,

WLOKA, M. (1995) The Virtual Tricorder: A Uniform Interface for Virtual Reality, *UIST'95* :39-40

ZHAI, S, BUXTON W. and MILGRAM P. (1994). The Silk Cursor Investigating Transparency for 3D Target Acquisition. *CHI'94* :459-464