UNIVERSITY OF SOUTHAMPTON

Integrating Hypermedia Techniques with Augmented Reality Environments

by

Patrick Alan Sousa Sinclair

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering and Applied Science
Department of Electronics and Computer Science

June 2004

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Patrick Alan Sousa Sinclair

Augmented Reality systems, which overlay virtual information over the real world, can benefit greatly from the techniques established by the Open Hypermedia research field. Storing information and links separately from a document can be advantageous for augmented reality applications and can enable the adaption of content to suit users' preferences. This thesis explores how Open Hypermedia systems might be used as the information systems behind AR environments. This provides benefits to augmented reality developers, not only because of the existing Open Hypermedia methods but also because of the applicability of Open Hypermedia interaction techniques to the augmented reality domain.

Tangible augmented reality techniques, in which graphics are overlaid on physical objects that can be manipulated as input devices, can be used to interact with the resulting information spaces by exposing the adaptation processes in the Open Hypermedia systems. This thesis describes the development of various physical interaction metaphors that allow users to physically manipulate the underlying hypermedia structures to their liking, resulting in a natural and intuitive way to navigate complex information spaces.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisor, Kirk Martinez, for his support and advice throughout the work described in this thesis. I would like to express my gratitude to my second supervisor, Wendy Hall. My thanks are also extended to the many members of the IAM research group that have helped me in one way or another over the years, namely: Dave Millard, Mark Weal, Gary Wills, Chris Bailey and Danius Michaelides. I would also like to thank Dave De Roure for allowing me to be associated with the Equator project, which gave me a better perspective on research and has provided many opportunities during my time at IAM.

I also want to thank my parents, as well as the rest of my family, for their support and dedication over the last four years. Obrigado.

Finally, many thanks to my girlfriend, Clara Cardoso, for putting up with me and believing in me.

Chapter 1

Introduction

Augmented Reality (AR) systems combine real world scenes and virtual scenes, augmenting the real world with additional information. This can be achieved by using tracked see-through head mounted displays (HMD) and earphones. Rather than looking at a hand-held screen, visual information is overlaid on objects in the real world.

Recent advances in computing hardware have made computing devices smaller and more mobile; wearable computers that can be worn on a person's body are now possible. These devices provide a completely new form of human-computer interaction as they can always be active and ready to interact with their users. As more processing power is packed into smaller units, more possibilities for mobile AR applications emerge; we are beginning to see extremely complex computer vision tracking systems running on mobile devices. The impact of this technology on AR will result in many new applications for our everyday lives.

Hypermedia is a technology concerned with linking between nodes in documents (Conklin, 1987); linking can be used to create non-linear documents that cannot be easily represented on a page (Nelson, 1967). As users navigate between documents it can be said that they are browsing through an information space. This makes the "browser" an important component in traditional hypertext systems, as it is where the user views and interacts with the hypermedia content; features such as support for navigation and multimedia playback are essential.

There are many possibilities for integrating hypertext and AR. It is possible to link between digital information spaces to objects and locations in the real world. Museums are the ideal target for this technology, for instance, when someone is reading a document describing an object in the real world they might be guided towards it. Some projects have even considered the layout of a museum's physical space between users and their intended destinations, for example if there were stairs or lifts. Similar or related objects could be also suggested. One could also imagine useful applications out in city streets or in the countryside. As people walk past or look at an object that interests them, there may be a link to information about that object.

One can envisage a scenario where an object and its information are seamlessly blended into a single interaction space, so that a description can be linked to an actual feature on the object, and selecting a feature, perhaps by touching it, would trigger information to be displayed.

This information can be presented in a variety of ways. With AR, many types of media can be overlaid on objects, such as images, videos and so on. Dynamic, multimedia labels could be used to highlight details on objects. It would be possible to overlay actors to recreate certain scenes, for example visitors to an art gallery would be able to see and listen to some of the artists, seemingly brought back to life. How to combine these different types of media to create a suitable presentation raises interesting issues; the implementation is also complex.

1.1 Approach

The merging of the real world with a virtual information space is one of the fundamental problems of AR research. Not only must AR systems be able to track real objects and display virtual imagery over them, they must be able to associate all manner of information with that object and decide how to present this information to the user.

This thesis examines how *Open Hypermedia* techniques can be used in AR systems to provide detailed information about specific features on objects, such as museum artefacts. By highlighting and describing interesting object features to users, they can obtain a better understanding of the objects.

In early hypertext systems, link anchors were embedded inside the documents. This is still prevalent in several popular systems, such as the World Wide Web. However, several problems may arise from doing this: one example is dangling links, when a link destination is non-existent because it has been moved or deleted. By storing links separately from documents in link databases, or linkbases, Open Hypermedia systems avoid many of these problems. Linkbases can be processed or indexed, and links can be applied in various ways. For example, a *generic link* is attached to any occurrence of a word in any document in the system.

Such an approach can be extended to AR systems, where information must be kept separate from the real world objects being described. This thesis demonstrates this technique, by applying label information to objects displayed in an AR environment through the use of an Open Hypermedia link server. It involves marking up three-dimensional models of objects, so that each feature can be identified and localized in space. Auld Linky (Michaelides et al., 2001), a contextual link server implementing the

Fundamental Open Hypertext Model (FOHM) (Millard et al., 2000), is used to supply labels to objects presented in the ARToolKit (Kato et al., 1999; Billinghurst et al., 1999; Billinghurst and Kato, 1999; Kato and Billinghurst, 1998), a vision based tracking system designed for the rapid development of AR applications. In a process similar to that of generic linking, relevant links and descriptions are obtained by querying the linkbase with the list of features present on an object. Being a contextual link server, the descriptions returned are dependent on the current context of the system; any irrelevant information is filtered out. By knowing the positions of the features in respect to the object, various effects can be used to present the information around the object.

Methods are required for exposing the information displayed in the AR environment using these technique. For example, users may wish to view more detailed descriptions about features, so they must be able to interact with the AR system. It can be challenging to create effective user interfaces in these circumstances: as users are wearing an HMD and immersed in the AR environment, and it is impractical to use traditional user interface devices such as a mouse.

Tangible user interfaces use physical objects, such as wooden blocks, as input and output devices instead of traditional devices such as mice and keyboards. *Tangible AR* overlays information on the physical input devices, providing enhanced display possibilities. For instance, to show the state of one of the wooden blocks, it would have to be physically modified or embedded with some kind of display. With AR, images can be projected onto the block, so there are many possibilities of displaying the object's state, including animations, 3D objects and so on. Actions performed with the objects can act as triggers for the user interface.

A tangible AR approach can provide natural and intuitive manipulation of the labelled objects. This led to the design and implementation of a suitable selection mechanism, which provided more possibilities for label display. Unselected labels could now be minimised, thus avoiding cluttering the display. Many large labels would obscure the object, ruining users' experience with the system. This selection mechanism has permitted the exploration of hyperlinking within AR environments, by allowing users to follow links between features on different objects.

Adapting the content to users' preferences is extremely important. Different people have different interests, so information should be adapted to the individual. Simple scenarios usually identify different user stereotypes, such as experts and novices or children and adults. Some systems will go beyond this and attempt to gauge the users' preferences automatically or semi-automatically, providing a finer control of adaptation. Other systems might track what users have seen to avoid repeating information and to point the user to material they haven't yet looked at.

Open Hypermedia link structures can be large, complex networks. Although adaptive techniques can simplify and focus the information being shown, the new contextual dimension can also add to the complexity of the hypermedia structures being served and it can be difficult for users to conceptualise the context in which they are accessing information.

This thesis describes tangible interfaces that allow users not only to control the visible hyperstructure, but also the process of adaptation that generates each view on the information. Several interfaces have been designed that expose the underlying hypermedia structures in novel, powerful ways, overcoming the limitations of traditional approaches. In this thesis, the design process and implementation details of such interfaces are discussed.

A formative evaluation has been conducted on various aspects of the tangible interfaces to obtain users' reactions on these systems. Various usability problems with these interfaces were identified and some possible solutions are suggested.

1.2 Contributions

This thesis presents the following novel contributions for the field of AR:

- The use of an Open Hypermedia link server for presenting information on objects in AR environments. This approach uses labels automatically placed around objects, with each label describing an individual feature on an object. The dynamic nature of this approach resolves several authoring problems present in existing AR systems, is extremely flexible, and addresses the adaptation of information to suit users' requirements. There is scope for applying powerful presentation techniques with this approach.
- The proposal of various tangible AR interaction techniques for manipulating the information applied by the Open Hypermedia link server, such as a label selection mechanism, and the ability to handle hyperlinks between objects within the AR environment.
- New tangible AR interfaces for adapting the presented information in novel and interesting ways. These interfaces offer direct manipulation of the information space, giving users control over the adaptation process.

1.3 Structure

The format for the rest of this thesis is as follows:

Chapter 2, *Augmented Reality* presents a brief overview of the AR field, including its origins, a general description of how an AR system works and some of the issues involved in creating realistic AR environments. Some examples of real world uses for AR systems are described, including medicine, industrial applications such as maintenance and in augmenting broadcast video, for instance coverage of sporting events, in real time. An important area of interest for AR research has been collaborative applications, and some of the issues regarding these systems are described. A marker based AR system for prototyping the designs introduced in this thesis, and several aspects of such systems are described. Various existing AR systems and projects are presented, including the ARToolKit, Studierstube and Archeoguide. A brief discussion about the future of AR systems concludes this chapter.

Chapter 3, *Augmented Reality and Hypermedia*, presents a scenario involving museum environments where spatially overlaid AR is used to present information about museum objects. Several AR projects are described that have focused on issues in information interaction, especially around museums. Ways in which Open Hypermedia systems can benefit information display in AR are described and a brief overview of the hypermedia field is given. A technique to place dynamically created, adaptive labels over 3D models of museum artefacts is introduced. The approach uses the Open Hypermedia concept of keeping data and links separate using linkbases; the linkbase is used to attach relevant descriptions to the respective area of the 3D model. The linkbase is served by the Auld Linky link server, a context based link server implementing FOHM.

Chapter 4, *Interaction Techniques*, describes how certain properties of the ARToolKit, the vision based AR system chosen for prototyping interfaces, could be used in tangible interaction techniques. This led to the creation of various simple prototypes. The experience gained constructing these led to the design of an interaction metaphor that allows users to select and highlight labels on objects presented in an AR environment. Hyperlinks between different features on objects are also investigated, and these are displayed by drawing an annotated line between links' source and destination anchors. This work was based on the labelling system described in Chapter 3. The main research interest behind this thesis is to investigate tangible interaction techniques for manipulating the information that is presented about objects in AR environments. This led to the design of a variety of tangible interfaces, of which two seem to warrant further evaluation. "Salt and pepper" allows users to construct recipes of information by sprinkling different types of context onto objects. "Waves" uses the position of context dispensers in relation to objects to affect the information displayed about an object. During the design and implementation of these interfaces it was discovered that the label selection technique required some refinements, such as moving labels so they do not obscure the object.

Chapter 5, *Evaluation* discusses issues that were considered when planning the evaluation of the interfaces presented in Chapter 4. The plan for the formative evaluation conducted is described, and involved six subjects experiencing the different tangible AR interfaces implemented through an HMD. The evaluation is split into three stages. To allow users to become accustomed to the AR environment the first stage projected objects only, with no associated information (i.e. labels). The second stage compared the two types of labelling, fixed and mobile, as well as the linking mechanism between objects. In the third stage the two approaches for manipulating information on objects, “salt and pepper” and “waves”, are compared. Generally, feedback was positive and users appeared to appreciate both the use of labels as well as the tangible interaction techniques to manipulate the presented information. The results of the questionnaire filled out by the users, together with their comments and observations of the users working with the interfaces, raised several improvements that could be made to the systems.

Chapter 6, *Conclusions and Future Work*, reviews the work on applying information using an Open Hypermedia link server, the various interfaces for manipulating the information in AR and discusses the results of the evaluation. Issues in using the various AR technologies to create the interfaces are discussed, as are the experiences of the interface and metaphor design. Ideas for future work are described.

Chapter 2

Augmented Reality

2.1 An Overview of Augmented Reality

Ivan Sutherland can be considered as one of the pioneers of AR for his work on developing the head mounted display (HMD) (Sutherland, 1968). At the beginning of the 1990s much research was done in the field of virtual reality and there was a growing interest in blending these systems with the real world. A special issue of the Communications of the ACM in July 1993 presented existing attempts to merge electronic systems with the physical world instead of replacing it (Wellner et al., 1993). This issue helped launch AR research. In 1997 Azuma wrote an overview of the field (Azuma, 1997) which was used as a starting point for many people new to AR; this has been updated and extended recently (Azuma et al., 2001). In the last few years, interest in augmented reality has grown considerably, with several conferences starting including the International Workshop and Symposium on Augmented Reality, the International Symposium on Mixed Reality and the Designing Augmented Reality Workshop.

Azuma defined AR as systems that combine real and virtual, are interactive in real-time and register in 3D. Note that this definition does not restrict AR to a particular display technology, such as HMDs, nor does it not limit AR to the sense of sight - potentially, AR can be applied to all senses.

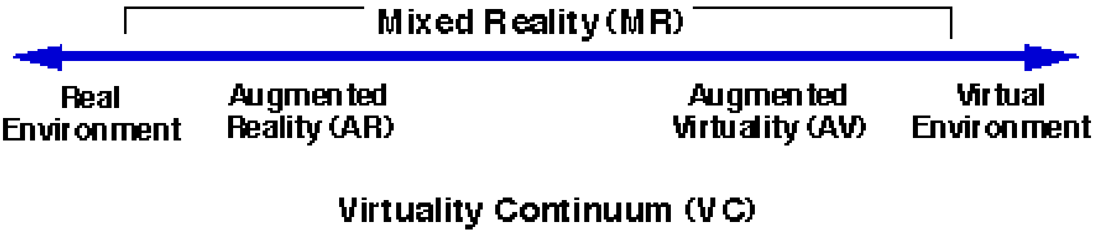


FIGURE 2.1: Virtuality continuum

In *virtual reality* (VR) environments users are totally immersed in and are able to interact with a completely synthetic world. AR environments on the other hand overlay virtual imagery onto the real world. Milgram et al. introduced a *virtuality continuum* (Milgram and Kishino, 1994), shown in Figure 2.1, where real environments are shown on one end and virtual environments are shown on the other. Between these two extremes lies what has been defined as *mixed reality*, with two categories: AR and *augmented virtuality* (AV), where images or video feeds of the real world are embedded in virtual environments. Other work has also focused on techniques to project and interact with virtual environments in the real worlds (Koleva, 1999). A subset of AR is *mediated reality* (Mann, 1994) where the perceived real world view is altered: for example, modern buildings could be removed or changed to obtain a historic view of a city.

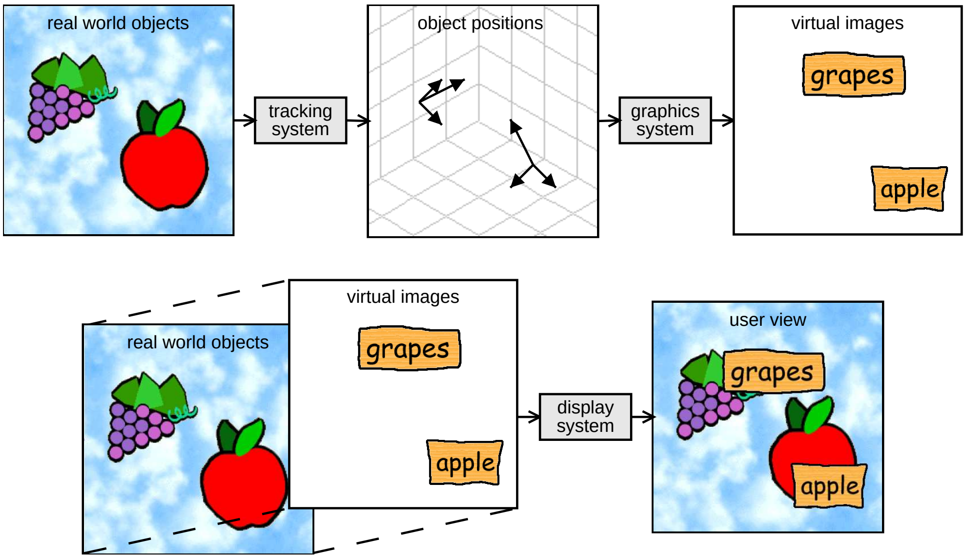


FIGURE 2.2: Typical AR system

A typical display based AR system (i.e. virtual objects are presented in a visible form rather than using sound or touch) has three major components, as illustrated in Figure 2.2. The *tracking system* determines the position and orientation of objects in the real world. The *graphics system* uses information provided by the tracking system to draw virtual images in the correct place, for example over the real objects. The *display system* combines the real world with the virtual images and sends the result to the user, for instance to an HMD, but a normal display such as a monitor could also be used.

A popular approach to user augmentation is to use see-through HMDs; there are two types, video and optical. Optical see-through HMDs use a transparent screen through which the real world can be seen. Video see-through HMDs combine a closed-view HMD with one or two head-mounted cameras; video from these cameras is overlaid with virtual material and is shown on the display. The user views the real world through the video on the HMD. There are various problems with both approaches, including high costs, weight and size. Optical see-through HMDs don't have enough brightness, resolution, field of view and contrast to give the illusion that the real and virtual have been blended together. Video see-through displays suffer from parallax errors as cameras are often mounted too far away from the user's eyes, resulting in a significantly different point of view. Another problem is that most displays can only focus the eyes on a particular distance, which is problematic as AR involves looking at objects in different locations. There are some interesting systems being developed that will tackle these problems; these have been described in detail by Azuma (Azuma, 1997; Azuma et al., 2001).

There are alternative types of display in addition to the HMDs described above. Small handheld, flat panel LCD displays can act as a window or magnifying glass showing real objects with an AR overlay (Rekimoto, 1997). Another approach is to project the virtual image directly onto the real object using a projector, which can be mounted in the environment or worn by the user. The advantage of an HMD over a handheld display is that it offers hands free operation and a more natural AR experience as the augmentation happens over the user's view; a handheld display is less cumbersome and can also be used as an input device by means of a touch screen technology.

The future of HMD design looks very promising; indeed, looking at past developments (Mann, 1997) shows how far they have come and gives an idea of what to expect. HMDs should eventually become small enough to fit inconspicuously into a pair of sunglasses. Virtual retina displays (Pryor et al., 1998), such as the ones developed by Microvision, project images directly onto the retina using low powered lasers; current prototypes are small enough to be mounted onto regular eyeglasses.

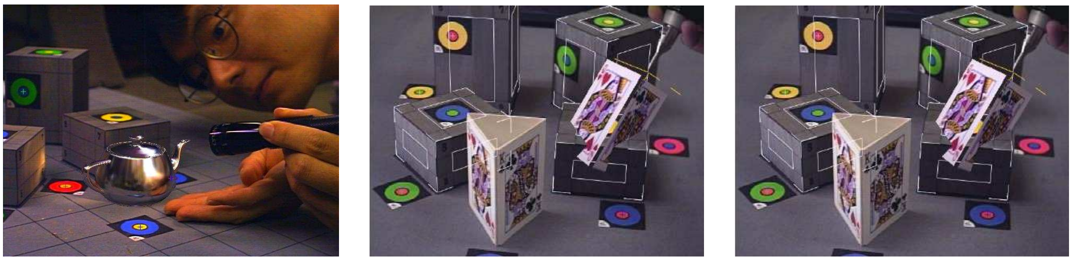
Sound is an often overlooked aspect of AR. Synthetic, directional sound could be provided by headphones and microphones could detect incoming sound from the environment. Haptic feedback, concerned with the sense of touch or force on the body, is also important; for example, gloves that provide tactile feedback might augment real forces in the environment. Smell is another sense that could possibly be augmented.

Certain objects in the real world can be tracked so that the system knows where to overlay virtual information. The two most popular tracking methods in AR are magnetic and optical tracking. Magnetic tracking involves a device transmitting a magnetic field that is detected by various sensors in the environment; these sensors pass this information, usually by a wire, to a filter that works out the sensors' position and orientation. Optical tracking uses cameras to track positions of objects in the real world using computer vision techniques; currently most optical systems rely on tracking special markers, such as fiducials, placed in special areas in the environment. Magnetic tracking is reliable but inaccurate due to magnetic fields, while optical tracking is precise but unreliable because of occlusion, shading and fast movements. Using different techniques together can improve tracking (State et al., 1996). For example, a magnetic tracker can provide a rough location for an object to narrow down the image area processed by the visual tracking system.

Outdoor or mobile AR is an interesting problem, as the environment isn't controlled or previously prepared and it is impractical to use markers. Various innovative approaches to tracking are being developed in this area, as most current systems are very limited; for instance, the Global Positioning System not only is too inaccurate for AR applications, it also needs optimal conditions such as a clear view of the sky.

Outdoor AR is also affected by hardware issues as the equipment must be portable, lightweight, comfortable, and low powered for longer battery life but still powerful enough to run complex operations; some of these issues are looked at in 2.4.

The registration problem in AR is ensuring that virtual objects are properly aligned with the real world to give an illusion that the two worlds coexist. In some applications, for example medical AR systems, accurate registration is crucial (Azuma, 1997). Registration errors have direct results, as the virtual objects will seem out of place in the user's view. Most existing AR systems require accurate calibration between the tracking device and the display device to minimise registration errors. Simplifying the calibration process has been a goal in a lot of AR research.



(a) Real light shining off virtual object

(b) Virtual cards colliding with real box

(c) Virtual object casting shadows

Figure 2.3: Improving realism in AR (State et al., 1996)

An interesting area of AR is making the overlaid virtual imagery look as realistic as possible, improving the user's experience in the AR environment as the additional visual cues are critical to seamless real-virtual world integration. Techniques being worked on in this field include calculating the ambient light (Drettakis et al., 1997) and applying this to the virtual model. One impressive demonstration shows light from a real torch being reflected off a virtual teapot. Work has been done on virtual object casting shadows that take account of the lighting in the real scene. Occlusion and collision of virtual objects into real objects also needs to handled (Breen et al., 1996). The photos shown in Figure 2.3 demonstrate these approaches (State et al., 1996).

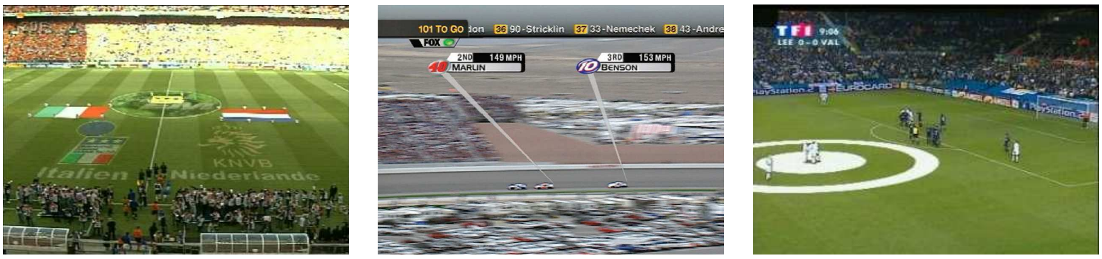
Some researchers have been investigating how human factors and perceptual problems affect users of AR systems, especially in determining the effects of long-term use of AR. The results of such studies will aid the design of more user friendly AR systems in the future.

2.2 Applications

Example applications for AR include medicine where it has been used as a visualisation and training aid for various types of surgery (Rolland et al., 1997; Bajura et al., 1992). Medical data could be rendered and combined in real time with a view of the patient. Doctors would be able to access various sources of information without having to look away from their patients. An interesting use of AR has been in the treatment of Parkinson's disease, where patients with difficulty in walking find that wearing an HMD projecting regularly spaced objects in front of them allows them to walk again (Weghorst, 1997). Steve Mann (Mann, 1994, 1997) has been exploring mediated reality applications for people with visual disabilities, where computer vision techniques might compensate for ailments such as blind spots.

AR has been used in several projects dealing with the provision of information to maintenance technicians; the idea is that instructions may be easier to understand if they were available as 3D drawings superimposed on the actual equipment. An early AR system was a laser printer maintenance application (Feiner et al., 1993b); several systems have been developed to be used in factories, such as Curtis et al (Curtis et al., 1998), one system is used to overlay 3D models of pipelines over the factory's machinery (Navab et al., 1999).

There are a few projects aiming to bring AR from the lab into industrial usage. ARVIKA (ARVIKA) is a German project with partners from both educational and industrial backgrounds, especially from aeronautical and automotive companies, that is looking at how AR can help in the development, production and service of complex technical products and systems, such as designing and safety testing automobiles.



(a) Projecting virtual logos (Epsis)

(b) Labelling racing cars (Azuma et al., 2001)

(c) Enhancing a free kick (Epsis)

Figure 2.4: AR for broadcasting sports events

A commercial use of AR is in augmenting broadcast video in real-time, for instance in advertising, sports and even real time studio applications (Epsis). Sports examples include adding virtual imagery such as club logos onto a sports ground (Figure 2.4a), highlighting hard to see objects such as a hockey puck in ice hockey, labelling race cars as they drive around the track (Figure 2.4b) and adding informative visuals such as lines indicating certain rules such as offside (Figure 2.4c) (Azuma et al., 2001).

2.3 Collaboration

An important area of interest for AR research has been collaborative applications, where multiple people can simultaneously view, discuss and interact with virtual objects. AR addresses two major issues with collaboration, as it provides seamless integration with existing tools and practices and it enhances practice by supporting remote and collocated activities that would otherwise be impossible (Billinghurst and Kato, 1999). Projector based systems leave the user free of bulky equipment, able to see each other and allow all participants to see the same augmentations; however, virtual information can only be displayed on the projected surfaces. With see-through displays information can be added anywhere: the Transvision system (Rekimoto, 1996b) used a handheld display, while several approaches have incorporated HMDs - such as Studierstube (Fuhrmann et al., 1998; Schmalstieg et al., 2000b), Emmie (Butz et al., 1999) and Shared Space (Billinghurst et al., 1998b). Collaboration between different systems has also been investigated, such as mobile AR-equipped soldiers collaborating with units in a VR military simulation.

One of the challenges in collaborative AR is to ensure users share an understanding of the augmented space, in the same way they naturally understand the physical space around them. If the augmented graphics are overlaid differently, providing each user with a slightly different view of the world, it may make it hard for a user to work out what another is referring to or pointing at. However, with HMDs each user has their own personalised view of the world, so information can be adapted to users' needs and private information can be displayed on the HMD. The Emmie system (Butz et al., 1999) discussed a notion of privacy management and presented an approach using real world metaphors such as lamps and mirrors. A careful balance is required between shared and private information in collaborative AR.

A collaborative system developed by Boeing (Curtis et al., 1998) allows onsite maintenance workers using wearable computers to communicate, through audio and video links, with experts at remote sites.

Collaboration is also a key requirement for entertainment applications, and several AR games have been developed. These include AR air hockey (Ohshima et al., 1998), a multiplayer combat game (Ohshima et al., 1999), AR chess (Szalavri et al., 1998) and AR enhanced billiards (Jebara et al., 1997).

2.4 Wearable and Ubiquitous Computing

One of the most important applications of AR is mobile systems: as the real world is being augmented, the more one can move around in the real world, the more interesting objects can be discovered and augmented.

Wearable computing and mobile AR research are very closely tied. A wearable computer can be anything from small wrist-mounted to bulky backpack computers; they should be mobile, augment reality and should provide context sensitivity (Billinghurst and Starner, 1999). Many mobile AR issues, such as tracking, miniaturisation of hardware and advances in display technologies are also being tackled by the wearable computing community. Traditionally, wearable computers were built from scratch by a small number of enthusiastic hackers, using hacked laptops, prebuilt components and custom built parts. They are now being sold as consumer electronics devices from companies such as IBM, Via and Xybernaut. Advances are constantly being made in the hardware as computing keeps getting smaller, faster and less power consuming; interesting input and output devices are being developed as well as novel user interfaces.

The human-wearable computer relationship is particularly interesting as the aim is to make wearables context sensitive, i.e. continually gathering and filtering information from the environment including conversations, locations visited, gestures and ambient sounds. The Rememberance Agent (Rhodes, 1997) stored this type of information to be recalled later when it determined, by monitoring the environment, that the user might need or find it useful. Some researchers have looked at emotional feelings by adding body sensors, such as body temperature or blood volume pressure, and using this information to adjust the interface to the user's mood.

The traditional desktop metaphor using windows, icons, menus and pointers, or WIMP, is not suitable for wearable computing (Rhodes, 1998). WIMP interfaces assume that interacting with the computer is the user's primary task, requiring his full concentration. Wearable computer users cannot afford this; they may be trying to cross the street or riding a bicycle. There may be distractions such as wind, rain or background noise.

WIMP interfaces assume that users are sitting at a desk while a wearable computer should be able to sense the environment around them. To accomplish this a successful wearable user interface must combine different types of input and output, depending on the user's context and needs. Input and output interfaces fall along a spectrum of user attention required. Passive sensors (such as GPS, cameras and microphones) require no user action for input. Direct manipulation interfaces such as WIMP applications demand the user's hands, eyes and full attention. In between these points are methods that require a low degree of attention, such as touch typing or performing pre-learned gestures.

Software agents that automatically act on the user's behalf, perhaps responding with simple messages, don't distract the user. Full text or multimedia requires the user's full attention to take in information. Ambient interfaces fall in the periphery of attention; for example, hearing a sound whenever an action occurs.

Collaboration is useful with wearables as many real life occupations are based on spontaneous meetings. From previous work in teleconferencing, collaborative virtual environments and computer-supported collaborative work researchers have determined that wearable conferencing spaces should have three key attributes (Billinghurst and Starner, 1999): high quality audio communication, visual representations of the collaborators and an "underlying spatial model for mediating interactions" (i.e. support many simultaneous users and allow them to read each others' body language).

Another important aspect of AR assisted mobile computing is looking at possible interactions with devices embedded in the real world. The term ubiquitous computing was coined by the late Mark Weiser at the beginning of the 90s (Weiser, 1991). In this vision, computers as we know them will disappear, giving way to many small computing devices embedded in everyday objects, all networked together. These devices will help users to focus on the tasks they are performing rather than worrying about interacting with the computer itself.

Initial work in this area involved smart rooms, where multiple sensors keep track of people inside them. The room's environment, such as the temperature and ambient light, can be intelligently configured to suit the preferences of the room's occupants. These can be identified by using active badges, which broadcast their owner's identity to any nearby sensors.

A major challenge is to create context-aware (Schilit et al., 1994) applications that adapt according to their location, environmental conditions (e.g. available computing resources, lighting, and background noise) and the social situation.

Some early research tended to split ubiquitous and wearable computing apart, focusing on the advantages and disadvantages of each approach. The obvious distinction is that while ubiquitous computing involves dozens of devices embedded in the environment aiding users, wearable computing focuses on a single portable assistant worn on a body.

One of the possible problems with ubiquitous computing is reliability (Rekimoto and Nagao, 1995): having so many devices embedded in the environment it is likely that some will eventually fail, either due to hardware or software trouble, or simply because of dead batteries. Detecting failures may be difficult as there will be so many computers. Another problem is cost; although the price of computers is always coming down it will still be expensive to embed devices in every single document in an office, for example.

There are also serious security issues to deal with in an ubiquitous computing world (Rhodes et al., 1999; Minar et al., 1999), as everything a user does is monitored and recorded by sensors in the environment. This data must be stored: a central database attracts attention; storing in several places means there are more potential security loopholes. Someone may not trust an environment to keep his data or profile safe, for example when a businessmen enters a competitor's company building.

Maintaining data in ubiquitous environments is another problem. Each time a person joins a work group or community, each device or central profile database must be updated. With wearable computing, sensors are kept on the person rather than in the room, so there's no need to transfer profiles as they travel with the user. As the wearable is always interacting with their owner, profiles can automatically evolve over time. Security is improved - if data isn't transmitted it can't be intercepted.

However, wearables have trouble maintaining localized information, so if one location changes, all wearables need to know about it. Wearables need to be ready to interact with any type of device they may encounter - perhaps this can be tackled using technologies such as Sun's Jini (Waldo, 1998). There may be resource management conflicts, for example two wearable users may try to control a resource, such as a stereo, at the same time. There may be ways to determine a wearable's location. In short, it is impossible to provide total privacy, but wearable computers can distribute personal data on a need-to-know basis.

Rekimoto (Rekimoto and Nagao, 1995) presents an alternative to ubiquitous computing where instead of embedding devices into the environment, paper tags are stuck onto objects. A wearable device that tracks and identifies the tags can then be used to obtain information about a certain object. As paper tags won't break down and are cheap to produce, the system is more reliable. However, as the focus shifts to the wearable computer, much work on the infrastructure is needed to support it; for example, wireless networks might be needed to query a central database.

The future will see environments where ubiquitous computing and wearable devices are used in conjunction; several researchers have been considering these possibilities (Rekimoto and Nagao, 1995). Ubiquitous computing is seen in part as the provider of the infrastructure supporting wearable devices.

2.5 Augmented Reality Interaction

Tangible Augmented Reality (Kato et al., 2000) applies tangible user interface techniques (MIT; Ishii and Ullmer, 1997) to augmented reality environments. Tangible interfaces are based on the observation that people have mastered the ability of sensing and manipulating their physical environments, so instead of using traditional input and output devices, such as a mouse, keyboard and monitor, tangible user interfaces are based on interacting with physical objects, such as simple wooden blocks. By overlaying virtual images over the physical objects, augmented reality provides enhanced display possibilities for these interfaces (Poupyrev et al., 2000a,b), eliminating the need to integrate displays into the physical objects, use bulky projectors or external monitors.

There is a wide potential for these types of interfaces as different actions performed to the real objects, such as arranging, shaking or moving can be used to trigger events; this is impractical or even impossible with traditional devices. Tangible interfaces allow natural two-handed interaction, collaboration between various people around the interface and provide a more stable environment. Unlike in traditional interfaces, physical objects won't disappear or move by themselves when the system changes state.

Tangible AR provides the opportunity to merge the "physical space in which we live and work with the virtual space in which we store and interact with digital information" (Poupyrev et al., 2002). This synergy results in an *augmented space* where digital information and objects can simply be manipulated as if they were real. Without the need for special-purpose input and output devices, interaction becomes intuitive and seamless as we can use the same tools to work with both digital and real objects. However, *spatial discontinuities* occur as the interface is limited to certain surfaces and cannot be extended beyond these; also, there is limited support for interacting with 3D virtual objects.

Another approach is *3D AR interfaces* that provide a seamless spatial augmented space around the user. Information presentation can be fixed to the user's viewpoint (head-stabilised), fixed on the user's position so it varies as he looks around (body-stabilised) and fixed to locations in the real world (world-stabilised) (Billinghurst et al., 1998a). These interfaces rely on using special-purpose input devices that are not normally present in the real world to interact with the augmented space. As the user is forced to switch between interacting with the virtual and real environment, the natural workflow breaks down, causing *interaction discontinuity* (Poupyrev et al., 2002).

Tangible interfaces are not ideal for all situations; there may be certain tasks that are impractical to perform by wielding physical items, such as complex searches. Rather than implement everything using tangible AR techniques, a balance must be struck between the two approaches and features distributed between them. In this way it is possible for tangible AR interfaces and traditional interfaces to complement each other; for example, blocks could be used to select and move representations of documents and a keyboard could be used to enter text into these documents.

2.6 Marker-based Augmented Reality

Several AR systems have been developed that are based around easy to track markers, usually printed on paper. Computer vision techniques can be used to accurately determine a card's position and orientation, enabling AR systems to overlay virtual objects over the cards. As they are easily distinguishable, relatively low processing is required to track them, making them ideal for mobile AR systems. The ARToolKit discussed in Section 2.7.1 and Rekimoto's work discussed in Section 2.7.5 are examples of these systems.

A very interesting point raised by Rekimoto (Rekimoto and Ayatsuka, 2000) is that many portable devices have an inbuilt camera or camera attachments. This trend will continue in the future and AR systems based on computer vision techniques, such as marker tracking, could benefit from this.

There are many advantages with paper based markers. They are extremely cheap and easy to produce, extremely versatile as they can be placed anywhere - especially if they are produced on sticky paper. By encoding an ID onto the marker, it is possible to link from a physical object or location to some form of digital information.

If markers are placed on cards or easily manipulated objects they can be used as input devices for Tangible AR interfaces. Another use is to place tags onto objects to give the appearance that the user is interacting with the object rather than a marker. One example is users physically dragging and dropping documents onto printers or data projectors from their wearable AR device (Rekimoto and Ayatsuka, 2000; Butz et al., 1999).

There are some more unusual uses for markers. Rekimoto (Rekimoto and Ayatsuka, 2000) discusses using tags on television broadcasts and on web pages; the AR system would recognise the tag and load up the information associated with the tag.

It is also possible to use tags to track users' locations in indoor environments where other forms of tracking aren't accurate enough. Marker tags are placed in certain locations throughout the building, each location having a unique tag. The system uses tag tracking to provide an exact fix of the users' position and other methods, such as gyroscopes and compasses, when no tags are visible. This approach has the potential to provide fairly accurate tracking with little cost in terms of modifying the environment. Systems that have used this method include Cybercode (Rekimoto and Ayatsuka, 2000).

Physical icons, or *phicons* (Ishii and Ullmer, 1997), have a strong coupling between the physical and virtual properties so the shape and appearance hint at the corresponding virtual object or functionality. In the Tiles interface, markers are generic data containers and are able to hold any digital data or even none at all; markers' physical properties are decoupled from the information attached on them. Operations performed on the tiles are the same for different types of tiles, resulting in a consistent interface.

Affordance in tangible interfaces is stronger than with traditional desktop applications as physical objects can provide more insights into their functionality and behaviour than their virtual on-screen counterparts.

The markers' physical design is important (Poupyrev et al., 2002) as it could influence the nature of the interaction: properties such as size or shape (flat or three-dimensional), or using markers that snap together like a jigsaw puzzle. Markers could be applied to three-dimensional shapes, such as cubes, pyramids or spheres; doing so may influence or even improve the manner in which the user handles the marker. If a hollow object is used small objects could be placed inside, creating a rattling effect when the object is shaken; this kind of physical feedback might reinforce the affordance of markers.

Instead of using paper, technologies such as LCDs or flat screen displays could be used to display markers that change their appearance depending on the situation. For example a photocopier could indicate an error code and alter its marker tag if it had run out of paper; the AR system would recognise the code and display an appropriate message (Starner et al., 1997). However, these markers would be more expensive and more troublesome to maintain. An alternative approach would be for objects in the environment, such as the photocopier, to communicate their status to the AR system to display the relevant information. This would require a more complex ubiquitous computer infrastructure.

A tangible AR system can be very ad-hoc and reconfigurable as users are free to place markers wherever and however they want. Configurations are created spontaneously depending on the users' activities and evolve alongside them. How to design such components and issues of system awareness are important research questions that need to be addressed (Poupyrev et al., 2002).

Examples of tangible AR systems include Tiles (Poupyrev et al., 2000b, 2002), a system for designing aircraft instrument panel layouts (see Section 2.7.1 for more detail. DataTiles (Rekimoto et al., 2001) uses transparent tiles on a flat panel display to show information. The information shown on tiles can be altered by how they are laid out on the flat panel display.

2.7 AR Systems

2.7.1 ARToolKit

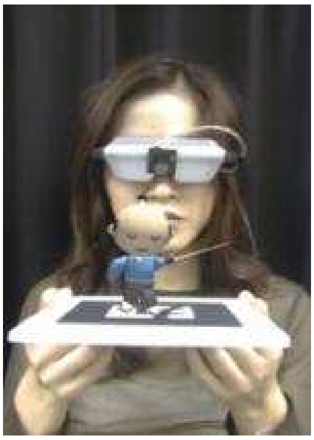


Figure 2.5: ARToolKit

The ARToolKit (Kato et al., 1999; Billinghurst et al., 1999; Billinghurst and Kato, 1999; Kato and Billinghurst, 1998) library developed at the University of Washington is designed for the rapid development of AR applications (Figure 2.5). It provides computer vision techniques to calculate a camera's position and orientation relative to marked cards so that virtual 3D objects can be overlaid precisely on the markers.

The ARToolKit was created as part of the Shared Space project, which aimed to enhance face-to-face and remote collaboration. It allows users to see each other and the real world at the same time, supporting natural communication between users and intuitive manipulation of virtual objects. For remote collaboration, a virtual video conferencing window is overlaid on the local real environment, supporting spatial cues and removing the need to be physically present at a desktop machine to conference. A collaborative web browser was developed that enables users to load and place virtual web pages around them in the real world.

Several applications have been implemented that use the ARToolKit. Augmented Groove (Poupyrev et al., 2000a) is an AR disk jockey system; users can play music together, with or without traditional music instruments, by manipulating markers on a table. The MagicBook (Billinghurst et al., 2000) is a traditional story book where AR marker cards are printed on the pages. As a reader looks at the book using the ARToolKit the pictures pop off the page and come to life as 3D animated virtual scenes.

Tiles (Poupyrev et al., 2000b) is an authoring interface for easy spatial layout of digital objects. One example using Tiles is a system for prototyping aircraft instrument panels, where markers are arranged on a whiteboard and each marker represents a different dial or instrument.

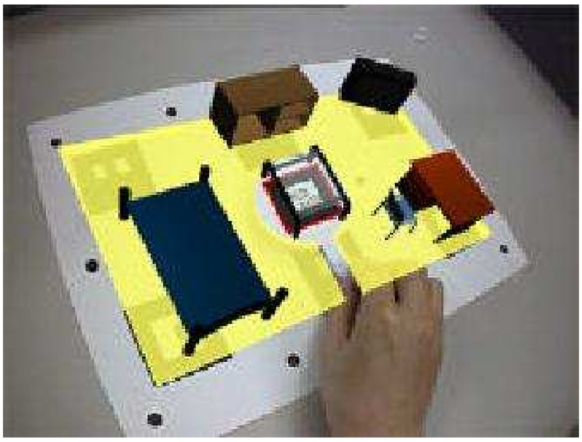
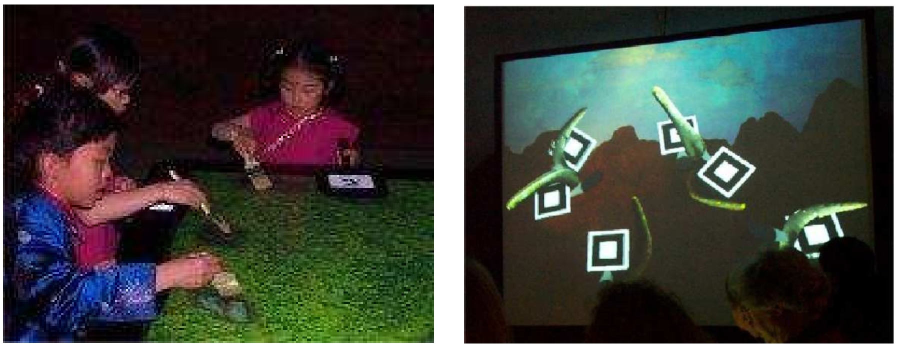


Figure 2.6: Virtual Object Manipulation in AR (VOMAR)

An interesting application of ARToolKit is VOMAR (Kato et al., 2000), Virtual Object Manipulation in AR. This system extends the ARToolKit and uses several types of markers to register a virtual image. This provides better tracking reliability, as a marker will continue to be tracked even if it is partly obscured. The improved tracking has enabled the implementation of a paddle interface which has been demonstrated in an

AR interior design application, shown in Figure 2.6. The user can pick up objects such as chairs or tables from a catalogue and place them inside the room, all using the paddle.



(a) Brushing dirt

(b) Virtual artifacts

FIGURE 2.7: ARToolKit in the Seattle Art Museum Project

In 2001 the ARToolKit was used as part of the Seattle Art Museum Project to create an interactive exhibit, illustrated in Figure 2.7. The goal is to give people the experience of discovering archaeological artefacts themselves and to be able to pick up and hold virtual objects. An interface was created where visitors dig up virtual dirt to reveal buried artefacts. Artefacts can be projected onto ARToolKit markers held by visitors, providing unrestricted views of objects and allowing visitors to compare different artefacts with their friends. However, no artefact information, such as a label or even a name, is provided.

An important aspect of the ARToolKit is that it is freely available, open source and has fairly low system requirements: a computer, a video camera and some marker cards. This has made the ARToolKit very important to many newcomers to AR, and has been used by many research groups worldwide.

2.7.2 University of Columbia

The Computer Graphics and User Interfaces lab at the University of Columbia has produced a lot of important work on AR. Early prototypes included an AR photocopier instruction manual (Feiner et al., 1993b) and an architecture application (Feiner et al., 1995) that shows the hidden structure system of a building.

A lot of work at Columbia has focused on building infrastructures for AR systems. One of their projects is COTERIE (MacIntyre and Feiner, 1996), a test bed for fast prototyping of distributed virtual environment systems. It is designed to support the creation of virtual environments with multiple simultaneous users interacting with many heterogeneous displays and input devices.

MARS (Feiner et al., 1997) is a wearable AR campus tour guide that can overlay building

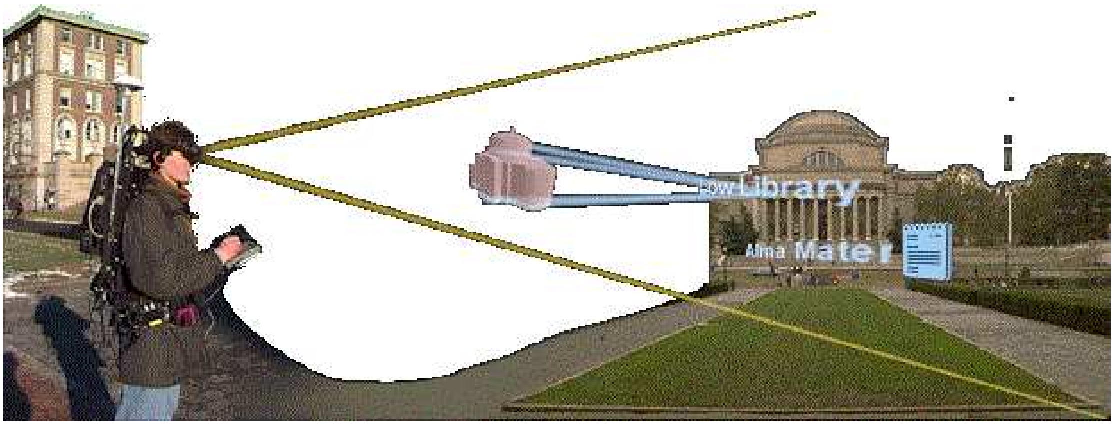


Figure 2.8: Mobile Augmented Reality System (MARS)

names on the actual buildings. The system only labels buildings, not specific building features, and uses a hand-held device to present more detailed information. This work has been extended so that images, video, audio and even 3D models can be overlaid on the campus to provide a situated documentary (Hollerer et al., 1999). This can be seen in Figure 2.8.

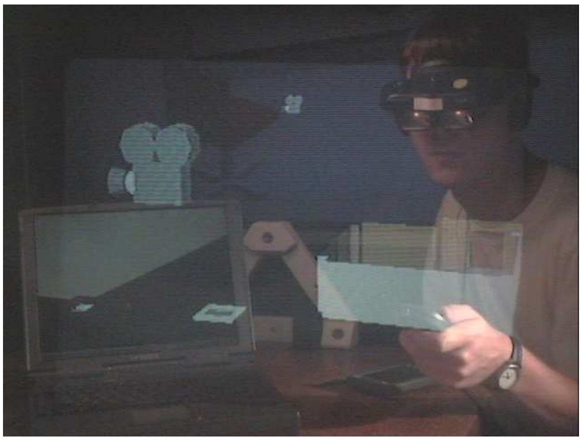


Figure 2.9: Emmie

EMMIE (Butz et al., 1999) is a hybrid user interface that aims to provide services usually available in conventional desktop interfaces in a multi-user AR environment. These services include management of information, such as images or text, over different displays and between different users. Information privacy is addressed in the system; for example, private items do not appear on public displays. Virtual objects, such as documents or available display devices, are represented as icons and can be attached to objects, people or fixed locations. These icons are displayed in AR and are visible using HMDs. An interesting feature of EMMIE is drag and drop behaviour: documents can be dragged and dropped onto a printer or monitor and an appropriate action is taken. Figure 2.9 shows a projector icon that can be used to project videos by dragging and dropping a video file onto it.

2.7.3 Studierstube

Studierstube (Fuhrmann et al., 1998; Schmalstieg et al., 2000b) is an AR system that focuses on scientific visualisation in collaborative virtual environments, especially for face-to-face co-operation between experts from different fields.

Studierstube allows multiple collaborating users to simultaneously study 3D scientific visualisations in a study room. Each participant uses an individually head-tracked see-through HMD providing a stereoscopic real-time display. It uses augmented props, tracked real world objects that are overlaid with computer graphics, as 3D controllers. Users choose their individual viewpoints and are also offered customised views of data; for example, two users in the same room may see different aspects of the same object at the same time.

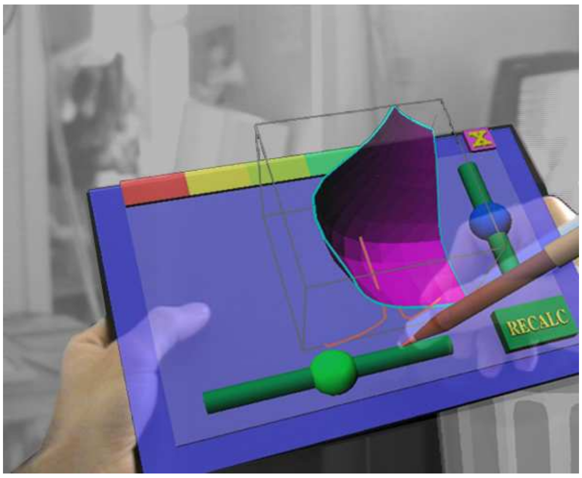


Figure 2.10: Personal Interaction Panel (PIP)

One of the most interesting features of Studierstube is the Personal Interaction Panel (PIP) (Szalavri and Gervautz, 1997), which is a two-handed physical interface composed of pen and pad, both fitted with magnetic trackers (Figure 2.10). As the pen and pad are real physical objects they provide haptic feedback and guide the user when interacting with the PIP. Conventional 2D interface elements, such as buttons or sliders, as well as novel 3D interaction widgets are overlaid on the pad and can be manipulated by the pen.

Studierstube has been used for different applications besides scientific visualisations. One application focused on collaborative gaming in AR (Szalavri et al., 1998), where users played board games in Studierstube. Construct3D (Kaufmann et al., 2000) is a 3D AR construction tool that aims to teach mathematics and geometry. Recent work has involved expanding the system to be more open and distributed (Schmalstieg et al., 2000a). There has also been research into an open object oriented approach to mix and match different types of trackers in novel ways (Reitmayr and Schmalstieg, 2001).

2.7.4 Annotating the Real World

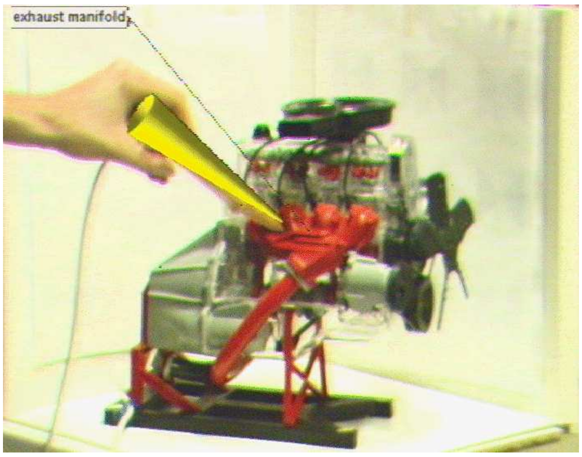


Figure 2.11: Annotating an engine in AR

Rose et al (Rose et al., 1995) presented an AR system where an automobile engine is annotated in AR with labels that identify the engine's components. As the user points to a specific part of the engine, the AR system draws lines and text labels describing the selected component. The engine is tracked so that the labels move as the viewer's orientation moves. The lines attaching the annotation tags to the engine follow the appropriate components, allowing the user to easily identify the different parts as the view changes. The text for the label can be defined in a database, offering a fair degree of flexibility. Annotation labels are two-dimensional (2D) boxes and are drawn on the same horizontal plane. The system must track which parts of the engine are visible to avoid annotating invisible features.

2.7.5 Jun Rekimoto

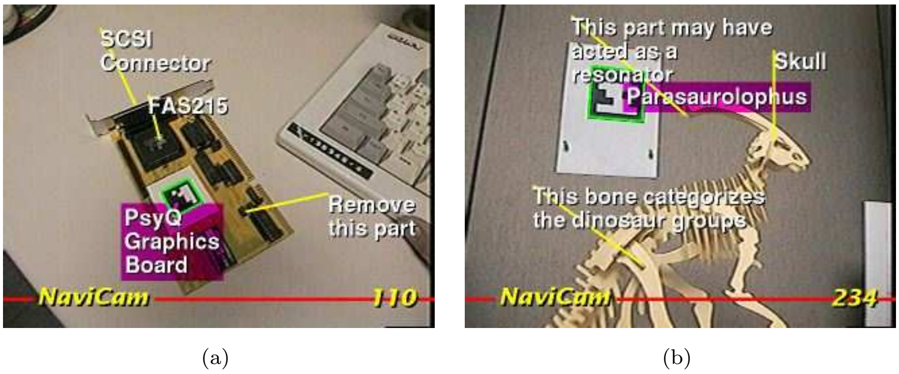


Figure 2.12: Cybercode

Jun Rekimoto has been involved in developing various interesting AR systems and applications.

Navicam (Rekimoto and Nagao, 1995) was one of his early mobile AR systems, which can be both handheld or viewed through an HMD. It tracks paper tags using a camera and computer vision techniques. The paper tag acts as a bar code, providing an ID for that tag. The position and orientation of the tag relative to the camera can be estimated so that virtual images can be overlaid on the tags through the display device. The aim of Navicam is to recognise the current real world situation and present information about it.

One of the prototype Navicam applications involved attaching a tag next to each museum object. As the system can recognise the tag ID, it can work out which object the visitor is looking at. It can also use the tag's position as a base for overlaying information over the object. This can be seen in Figure 2.12(b); note the text labels pointing at parts in the model.

Cybercode (Rekimoto and Ayatsuka, 2000) is very similar to Navicam but with improved computer vision techniques so that more bits can be encoded in the ID pattern, resulting in a larger number of possible ID tags.

2.7.6 Archeoguide

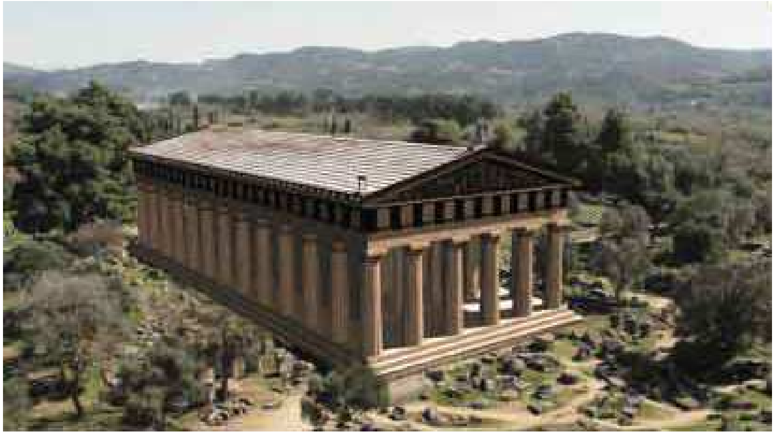


Figure 2.13: Augmenting an archaeological site

ARCHEOGUIDE (Augmented Reality-based Cultural Heritage On-site GUIDE) (Vlahakis et al., 2002) is an EU project looking at using virtual and augmented reality to present tours around archaeological and cultural historic sites. The goal is to have mobile AR units, including laptops, tablets and palmtops, providing information adapted to visitors' profiles and optional tours; various sorts of audiovisual information can be presented as the user is guided around the site.

Differential GPS is used to track users' location so that information about nearby items; when they enter certain areas of a site a wireless LAN is used to download information from a central server. The system is designed around a client-server architecture that allows access to many people at a time. The central server uses a database with geospatial extensions, so an object's location is stored along with all information about it. Various types of multimedia content are available including text, image, sounds, video, 3D models and animated human avatars. The server serves as a platform for custom authoring and browsing tools to aid in the creation of content, and third party applications are also used especially in the creation of the detailed 3D models of the historic monuments.

The project looked at existing documentation standards for archaeological sites and monuments, especially CIDOC (International Council of Museums, 1995), when they were designing their system. The CIDOC standard is used to create an inventory of cultural sites, and their system allows users to augment CIDOC databases with various types of media objects. All content is described with metadata, used to specify the intended target audience, and scripts can be set up for personalised tours through a sequence of objects.

An impressive feature of Archeoguide is the techniques that have been developed to render near-photorealistic three-dimensional reconstructions of historical buildings over the actual site, viewable through a mobile AR unit with an HMD (Figure 2.13). Differential GPS in conjunction with a compass is used to get a rough position estimate which is then refined with image tracking; the technique compares calibrated reference images from the database with the current video stream, captured through a camera on the mobile AR unit, for tracking users' position. If there are pre-rendered images of the reconstructed monument from the same point of view they can be added to the user's view. Thus visual augmentation of reconstructed buildings can only occur in specific previously calibrated positions, and is not available throughout the whole site.

As users look at a building that has been added to a scene, the system queries the central server for a personalized audio narration stream that is played while they look at the building; they can stop or change the commentary by looking in a different direction. They can also request navigation information or more detail about a certain building. The system has also been developed for tablet and handheld computers that only use GPS and no camera tracking; instead of overlaying the buildings as the user looks at the site they show static images taken from the users' current position.

2.8 Future of AR

There are still many issues to be investigated in AR research. See-through displays, especially HMDs such as the Sony Glasstron, are not designed to work outdoors; for example, the image is not bright enough to be used in strong sunlight. As existing limitations of displays are resolved and realistic experiences will be possible.

The miniaturisation of components also continues. Users of mobile AR systems have to wear an HMD, computer, sensors, batteries and so on, resulting in a heavy backpack. Certain components, such as USB connectors, are not rugged enough for outdoor use and can cause problems.

As mobile computers such as laptops become faster more interesting and useful processing can be done for visual and hybrid tracking. Tracking in unprepared environments, such as outdoors, is a tricky problem, and current solutions are still limited in factors such as range and precision. They are also expensive, bulky and impractical, often requiring a complex calibration process.

Another interesting factor is the social acceptance of such devices: will the hardware become compact and practical enough so that they become as commonplace as mobile phones or PDAs? Issues ranging from fashion to privacy (as the tracking data of people's locations could be misused) will affect the way people adopt and use these devices.

There is still a lot of work to be done on AR user interfaces: methods to display and interact with data in AR must be understood. A lot of research has been focused on low-level issues, such as depth perception, how latency affects manipulation and so on. Higher-level concepts and issues need to be considered: what information should be shown, how should one represent it and what interaction metaphors are appropriate?

2.9 Chapter Summary

This chapter has presented a brief overview of AR, describing its origins, a general description of how an AR system works and some of the issues involved in creating realistic AR environments. Some examples of real world uses for AR systems were described, including medicine, industrial applications such as maintenance and in augmenting broadcast video, for instance coverage of sporting events, in real time. An important area of interest for AR research has been collaborative applications, and some of the issues regarding these systems are described. The field of wearable and ubiquitous computing has been strongly related to AR research, especially with mobile AR systems. Techniques for interaction with AR environments is the area that has interested me the most during the work presented in this thesis, especially the merging of tangible interfaces with AR displays. A marker based AR system for prototyping the designs introduced in this thesis, and several aspects of such systems are described. Various existing AR systems and projects are described, including the ARToolKit, Studierstube and Archeoguide. A brief discussion about the future of AR systems concludes this chapter.

Chapter 3

Augmented Reality and Hypermedia

The principal feature of AR systems is that information is overlaid on the real world; this makes information display design a key area in AR research. Various approaches to displaying information have been identified in AR research, including head stabilised, body stabilised and world stabilised (Billinghurst et al., 1998a). Of these, world stabilised, where information is fixed to locations in the real world, would appear to be the most beneficial as users don't have to switch between interacting with the virtual and real environment, minimising interaction discontinuity (Poupyrev et al., 2002).

Tang et al. conducted a detailed investigation of the various ways in which AR can increase effectiveness of manufacturing tasks (Tang et al., 2002b,a). By projecting information over the workplace, AR reduces head and eye movement so in theory user performance should increase (Haines et al., 1980). AR reduces the cost of attention shifting, as the information is seamlessly integrated with the real environment and overlaid information is taken in as part of the human cognitive process (Neumann and Majoros, 1998). Users don't have to switch between the instructions and the task, increasing performance. Overlaid graphics can be used to augment human attention, for instance using an arrow to highlight an object. AR supports spatial cognition and mental transformations: people tend to memorize information more effectively when ”docked” to a frame of reference in the real world. Neuroscience research suggests a strong relationship between spatial location and working memory (Kirsh, 1995). AR systems can benefit from these aspects, as information is spatially placed around objects in the real world (Biocca et al., 2001).

Tang et al. conducted an experiment to determine the impact of these factors (Tang et al., 2002b,a). It compared users performing an assembly task with four types of instruction materials: printed media, instructions on an LCD monitor display, instructions projected on a see-through HMD and spatially registered AR. The experiment looked at time taken, accuracy and mental workload. The results showed little difference between instructions projected on the see through HMD and the traditional displays. The spatially registered AR, although having little effect on the time taken to complete the task, resulted in better accuracy and reduced the mental workload. They concluded that these results indicate that AR is a more effective instructional media, although certain current technical limitations may hinder its practical use. They also noted that attention tunnelling, where users focus their attention on cued areas thus ignoring others, can be a problem in certain AR systems where users need to be aware of unexpected events or objects in their environments.

During my research, I have considered spatially registered AR as an approach for displaying information about an object. My main interests have been to point out interesting aspects of individual objects, especially detailed descriptions of object features, and to create tangible interfaces to this content. To accomplish this it is important to place information over objects, so applying informational materials over objects in the AR environment was the first challenge I aimed to overcome. These methods must take various issues into account; for instance, the process of adding information should be open and extendible, and the authoring process should be accessible and as simple as possible. In Chapter 4 I describe my work on tangible interfaces to the augmented information.

Museum environments stand to benefit from such AR techniques, and in this Chapter a scenario based around museums and museum visitors is described. I then mention several AR projects that have investigated issues in information interaction, especially around museums. I propose that the use of hypermedia will benefit information display in AR, and describe an overview of the hypertext field. I introduce Open Hypermedia technologies, FOHM and Auld Linky, which I have used to create a demonstration system, based on Open Hypermedia concepts, to present adaptive hypermedia information about objects in AR.

3.1 Scenario

Problems with current museum information displays are well studied, and have been the source of several projects in different research areas. For instance, exhibit labels are usually limited to only a few words and must be pitched at the average museum visitor while in reality different people will want different types and volumes of information. Other approaches, such as paper maps, written guides and audio tapes, do not guarantee flexibility: audio systems force a predefined or limited path; descriptions are not easily related to each other and are based on the author's perspective; all of the material is pre-written so it may be out of date, and the material is static and is troublesome to update. Visitors may also be interested in related objects, documents and sites and they may wish to read material before and after their visit to the museum.

Virtual museums, such as online web sites or CD-ROM multimedia applications, are more flexible since the display of information can be determined by the visitor's preferences, interests and interaction history. Virtual representations of real museums are popular, using point and click three-dimensional models (Barbieri and Paolini, 2000), but this is not the same as viewing the real exhibit. Even with advanced display systems the perception of size and colour may be different and the experience of seeing a virtual real object is not as emotive as seeing the real thing.

An alternative to virtual museums is to use multimedia information kiosks or portable devices in the real museum to display information about objects the user is looking at. Several systems, such as GUIDE (Cheverst et al., 2000) or Hyperaudio (Not et al., 1997a,b; Sarini and Strapparava, 1998) have been developed where a portable, usually hand-held, display device presents information as the visitor walks around the museum.

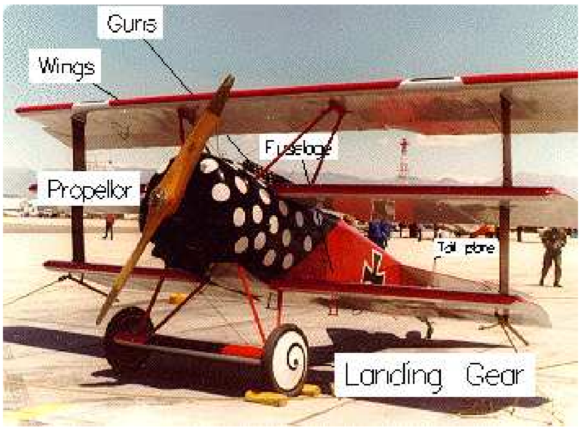


Figure 3.1: Real world object augmented with information

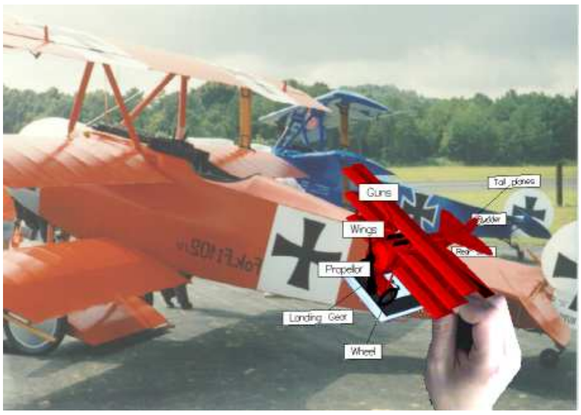


Figure 3.2: Virtual object and information projected onto the user's view.

The use of Augmented Reality (AR) in museums promises great advances in natural interaction with objects and their data. Museum visitors could be equipped with mobile AR systems as they walk around. When visitors approach an artefact they are interested in, the system could track the object's position so that information could be overlaid directly on the object's features, as illustrated in Figure 3.1.

In practice, this is not an easy process and few systems have approached tracking individual objects. However, easy to track markers can be attached to the objects and the system can use these to determine an object's position (as described in Chapter 2).

These systems also allow virtual objects to be projected onto markers held by the user, as shown in Figure 3.2. The user can manipulate the marker to view the object, for example rotating, moving and zooming around it. This is important as often artefacts are stored in display cabinets, which can restrict a museum visitors' view. Using AR, details that they might otherwise have missed could be pointed out to them. Visitors could zoom into regions of the artefacts, allowing them to view details that are usually too small to notice. Large objects such as buildings could be scaled down so users get an overall view of them. The system is by nature unobtrusive, as visitors can dismiss the displayed information by simply putting away their markers when they wish to rest or take a better look at the real artefacts. Visitors could carry virtual objects around with them on the marker as they wander through the museum. This could be tied in with the museum shop system, so visitors could "carry" models and information about their favourite objects home with them. The system could also present a selection of books and other materials of their favourite subjects to purchase as they leave.

There are other advantages as any model can be loaded and projected onto the user's view. Similar objects, for example of the same style or era, could be projected next to the artefact for comparison. These objects might be stored remotely, from another wing of the museum to the other side of the world. Objects being restored or kept in storage, due to lack of exhibition space, could also be shown. Different views could be presented, for example an x-ray view or a reconstruction of how the object originally looked. Visitors would be able to view parts of the artefact that are usually concealed or unclear.

As museum visitors wear their own private HMDs, information being presented about a museum artefact can be adapted personally to each individual. The selection and presentation of this information can be adapted according to the visitor's goals, preferences, knowledge, and interests. This is useful as visitors will be interested in different types of artefacts and they may also be interested in different aspects of an object.

As the system is portable, visitors are free to explore the museum. The visitor could ask for other objects in the museum that interest them. Suggestions might include physical spaces such as rooms and areas as well as specific objects. The mobile AR system would run on a wearable computer connected to the central museum database over existing wireless networking technologies. Ideally, this connection would link the visitor to the information repository no matter where they were in the museum.

3.2 Information in AR

In this section, an overview of certain AR systems and applications that have tackled information presentation is given. In this study, there has been an emphasis on investigating the different types of information that can be presented, and how this information is applied to the AR system. This study has been conducted over the period of my research, and although recent developments are described they have not had as much influence as the earlier work.

There are many kinds of information sources that may be rendered within an AR environment. These can vary from pregenerated "static" 3D models (although these may have animated features) to rendering dynamic information over real and virtual objects in the AR environment.

The rendering and manipulation of pregenerated 3D object models has been used very effectively in projects such as the ARToolKit Seattle Museum project (Billinghurst, 2001), where users participate in a virtual archaeological dig and can manipulate their finds in an AR environment.

The Virtual Showcase project is concerned with developing an AR display cabinet for museum and educational applications (Bimber et al., 2002), where high quality textured models of objects are projected. Recently they have been exploring more sophisticated effects by arranging AR scenes into sequences that form simple narratives (Bimber et al., 2003). They also support interaction by embedding links and hotspots into the Virtual Showcase Modelling Language files that specify each application (Ledermann, 2002).

The ARVIKA project demonstrated a system at the International Symposium on Mixed and Augmented Reality in 2002 that used an AR interface to support mechanical maintenance by rendering 3D instructional diagrams over real world objects (in this case a combustion engine) (Dick, 2002). The system uses a set of separate 3D models displayed one after another as the user progresses through the task. While this benefits from simplicity, it can cause problems at the authoring stage where many different but similar 3D models have to be defined.

PowerSpace is a system that attempts to tackle this authoring problem by allowing content to be arranged around a phantom model (which will be replaced by the real object in the AR environment) within Microsoft PowerPoint and organised into different slides (Haringer and Regenbrecht, 2002). These can then be imported into the PowerSpace viewer where the 2D elements are given 3D positions and multiple sequences can be authored through the slides.

Rather than embedding information directly into the 3D content in a prerendering process, many systems combine it at runtime. This kind of just-in-time inclusion offers considerable advantages for the flexible coupling of models and information. It also makes the information easier to maintain as the alternative is to continually author or modify 3D models in a complex package.

Grafe et al present an interactive exploration of a museum exhibit, where a camera mounted on a swan neck can be moved and aimed at parts of the exhibit. By detecting markers placed on the artefact, text labels describing the features are shown. These labels increase in detail as the camera is brought closer towards a feature (Grafe et al., 2002). The information displayed in the labels is loosely tied to the objects, and can be changed by using a different configuration file.

Rose et al (Rose et al., 1995) presented an AR system where an automobile engine is annotated in AR with labels that identify the engine's components. The text for the labels is defined separately from the 3D model, so that different annotation sets can be presented by simply changing the annotation file that is used.

In his CyberCode project, Rekimoto combines multiple real world objects with a sizeable database of annotations (Rekimoto and Ayatsuka, 2000). Unique identifier marker cards are placed into the world, and each one is linked to a database entry. He also makes the point that dynamic information application is necessary if the presentation is to be adjusted for the user in any way, for example to personalise annotations (Rekimoto and Nagao, 1995).

Many systems use AR for displaying and manipulating 3D content but present other information separately in a more conventional manner, such as selecting an artefact to view from a database application.

The ARCO project is using an AR interface for displaying high quality 3D models of museum artefacts. The system uses an underlying database holding rich information about each object. The AR environment is only one interface (meant for local museum visualisation), and only displays the model of the artefact. The database is exposed through a web interface, where views of the complex metadata are presented (Mourkoussis et al., 2002).

ARCHEOGUIDE is an AR system for archaeological sites, where high quality building reconstructions can be overlaid on the actual terrain (Ioannidis et al., 2002). It uses a client-server multimedia database with geospatial extensions, so information about an artefact is stored in relation to its location. Like ARCO, the AR environment is used for rendering high quality reconstructions of the objects (in this case buildings). ARCHEOGUIDE maintains profiles of the users and tours of the information in order to provide personalised views which are then presented separately on a mobile device (tablet PC or PDA) (Vlahakis et al., 2002). ARCHEOGUIDE was described in more detail in Section 2.7.6 (Chapter 2).

Many systems allow the user to manipulate the information space inside the AR environment but use conventional interface metaphors to either display or interact with the information. For example, early work by Feiner explored the projection of a windowing system in an AR environment (Feiner et al., 1993a). In addition to the tracking system, it used conventional input devices (i.e. mouse and keyboard). The underlying information was organised in a hypermedia system which had the ability to make links between arbitrary windows in the display and to attach windows to real objects and locations.

The MARS system is another example of a composite interface (Feiner et al., 1997). It is a wearable AR campus tour guide that can overlay name labels over real buildings, and uses a hand-held device to present more detailed information of the user's choosing. The AR environment contains menus and pointing devices, so in this case interactions in the AR environment cause changes in the conventional display. This work has been extended so that images, video, audio and even 3D models can be overlaid on the campus to provide a situated documentary (Hollerer et al., 1999).

KARMA is an AR photocopier maintenance application built on IBIS, a knowledge-based system for generating maintenance and repair instructions (Feiner et al., 1993b). The user interaction with the sophisticated rule-based back-end is completely implicit as the system monitors the user's position in relation to the photocopier and the current state of the maintenance task.

3.3 Approaches to Hypertext in Augmented Reality

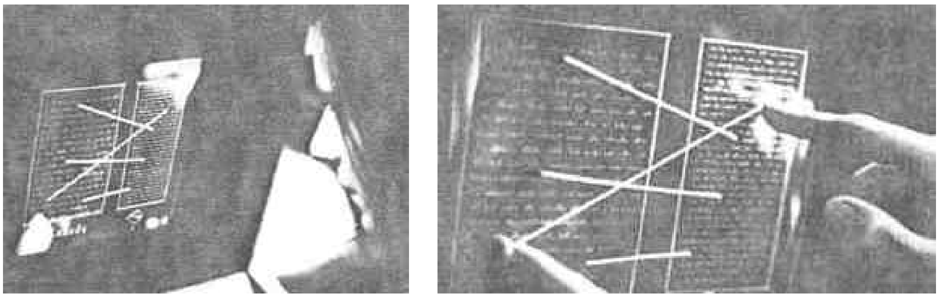


FIGURE 3.3: Nelson's CosmicBook (Nelson, 1972)

In the 1970s Ted Nelson described CosmicBook (Nelson, 1972), a system with visible hypertext connections; this means that links between documents were shown as lines between windows. Figure 3.3 shows a prototype that illustrated this concept.

Hypertext is a non-linear information medium, where users control the flow of material by following links between nodes in documents. There is an obvious similarity between hypertext and AR: hypertext involves links between nodes and AR involves associating information to objects in real world. A hypermedia system where linking can cross the boundary between the real and virtual worlds could be seen as a starting point for AR systems. Such a system might present all types of media within the AR environment and display the hyperlinks between the various forms of information, perhaps as envisioned in Nelson's CosmicBook.

A large problem for AR is the process of applying information stored in the system to the real objects in the environment. The question of authoring is also important: existing AR systems tend to author static 3D models together with related information in 3D modellers, which can be time consuming and expensive (Dick, 2002). A dynamic process, where information is automatically placed around the real object, appears to be a better approach in terms of authoring, as the complex positioning of information and interactive events (e.g. animations) are handled automatically. This has been explored in the field of *Open Hypermedia systems* (described in Section 3.4.1): links stored separately from the documents can be applied dynamically by the system. AR environments can stand to benefit from using these kinds of techniques, and these issues are described in detail in Section 3.5.

The field of adaptive hypermedia, where the hypertext presented to users is adapted to their interests, is well suited for HMD AR setups. Information can be adapted and presented on individual user's displays; this avoids the problem of displaying personally adapted information on shared displays.

The use of hypertext techniques in AR environments has been explored by several projects, some of which are described below.

3.3.1 Hypertext-based Augmented Reality Systems

Starner described physically based hypertext, where hyperlinks are associated with physical objects: for example linking to documents for instructions, repair information, history or view information left by previous users (Starner et al., 1997). Physical hypertext can lead to more efficient use of workplace resources, guide tourists through historical landmarks or overlay role-playing games onto the physical world. A prototype system to give a tour of the MIT laboratory space was developed that tracked visual tags; these tags were used to add labels to the real world. To avoid overwhelming the user with information as they entered a room, the system used arrows to point out important features and only added a label when the user demonstrates an interest in the feature.

One novel use of this system was that computation could be assigned to passive objects. One example was that a tagged plant could “ask” passers by to water it when it had not been watered for a while; the system kept a schedule of when the plant had been watered on the network so that when the plant tag was viewed the appropriate message would be displayed on the plant. The system appears to behave like an ubiquitous computing environment; however, a sparse infrastructure is needed as no computing device is actually embedded in the actual plant.

The Dypers system avoids using cards or markers (Jebara et al., 1998). The user indicates a visual object to associate with some media content, such as an audio or video clip, by taking a snap shot of it. A real-time computer vision system is used to detect the object when it is later encountered, triggering Dypers to play back the appropriate media clip (a physical link). The system was implemented with a museum environment in mind. As visitors walk around a museum they associate an exhibit object with a description, for example the object's label or the tour guide's comment. After the visit they can recall the description for each object seen. Usability Tests showed that users of the Dypers system could remember more of the trip than those using traditional memory aids, such as a notepad.

The HyperAudio project (Not et al., 1997a,b; Sarini and Strapparava, 1998) studied the navigation of museum areas using a portable hand-held device. By integrating the virtual and physical spaces an augmented space is created. The system provided information based on the visitor's preferences and context, could help locate items, suggest new locations to visit, avoid getting lost and avoid misunderstanding concepts. Visitors interacted with the system as they walked around the museum by triggering location sensors, which displayed information for that location. The system had a notion of the space around its user and was able to point users to relevant information and objects of interest to the users. When recommending locations, the system could take into account their location inside the museum, including how far the visitor would have to walk and whether they had to climb stairs or use elevators. Users could also control the hand-held device to view information, ask for related information or objects, comparisons, for example paintings by the same author or of the same period, suggested paths and the presentation of instructions. Output could be either content presentation or suggestion of next steps to visit. HyperAudio uses pre-recorded audio files assembled automatically or at the user's request. The visitor's profile determined how these files are assembled and which suggestions to make, providing adaptivity. The interaction history was also stored to avoid repetitions, to remind the visitor of previous presentations and to introduce comparisons.

The GUIDE (Cheverst et al., 2000) project is an outdoor tourist guide, which uses a hand-held computer to display information. The presented information depends on users' current environment, such as their location, time of day, the weather and so on. Information is adapted to users' interests. The GUIDE system is able to present up to date dynamic information on locations visited by the user, so real world locations were linked to a set of documents related to that place.

3.4 From Hypertext to Open Hypermedia

During my research I have considered how the use of hypertext, and in particular open hypertext, can benefit AR in terms of associating information within AR environments. This section covers a brief overview of the hypertext field, describing the developments that led to Open Hypermedia and adaptive hypermedia.

In 1945, Vannever Bush described an automated library, called the 'Memex', in his landmark paper 'As We May Think' (Bush, 1945). This theoretical device, designed to augment human memory, would allow individuals to store all their books, records, and communications. A mechanical system would facilitate rapid and flexible consultation of the material, by mimicing the associative connections of the human brain rather than a sequential index of records. Trails through the information space would be annotated and shared among others; Bush envisioned the machines forming huge repositories of human knowledge available to all, from lawyers and patent attorneys to physicians and historians.

In would take nearly two decades before real systems with Memex-like features were developed. During the 1960s, Englebart created the oN-Line System (NLS), which was based on Bush's ideas but used electronics rather than mechanical components (Engelbart, 1962). It was a sophisticated system that featured the world's earliest graphical user interface, involving a television display and the first mouse. The NLS was able to cross-reference research papers for sharing among geographically distributed researchers; it provided groupware capabilities, screen sharing among remote users, and reference links for moving between sentences within a research paper and from one research paper to another.

Around the same time Ted Nelson was working on the Xanadu project, which aimed to build a electronic literary system for worldwide use with a consistent organized general system for data management. It was Nelson who coined the term hypertext. Nelson described Hypertext as "a combination of natural language text with the computer's capacity for interactive branching, or dynamic display ... of a nonlinear text ... which cannot be printed conveniently on a conventional page" (Nelson, 1967).

Early steps in hypertext research were described in a survey by Conklin (Conklin, 1987), who identified the distinguishing feature of hypertext systems to be "machine supported links", and that other common features prevalent at the time, such as text processing and window-based user interfaces, were merely extensions of this concept. Conklin also identified hypermedia as an extension of the hypertext concept to other types of media, such as images, sound and video.

Two problems with hypertext were identified: disorientation and cognitive overload. As the organisation of information becomes more complex, it becomes harder to navigate around, especially with the non-linear structure of hypertext. Users need to know where they are and where want to go. This problem gets worse as the number of nodes within a hypertext increases. Hypertext offers more degrees of freedom, which makes it easier to get lost. The second problem is the cognitive overload from extra difficulties in keeping track of a reading context.

The survey also discussed the key advantages of hypertext systems. With hypertext, it is easy to trace references between documents by following links between documents. Hypertext systems should allow users to add new links and annotations without altering the original documents. Systems should allow structure, both hierarchical and non-hierarchical, to be imposed on unstructured collections of documents. Local views on large amounts of data should be supported, allowing easy reconstruction of large or complex documents. Views on documents should be customisable by offering the ability to link to certain parts of documents, thus improving material reuse. This can lead to documents being split into small modular units, which is useful for both authoring and presenting documents. References, i.e. links, should move with the document text so that linking remains consistent. Hypertext systems should support collaboration of authors on creating and annotating documents.

Several systems were developed in the 1980s that explored new forms of hypertext, and addressed the problems identified by Conklin. These systems were mostly monolithic systems with their own proprietary data formats and constructed as single large applications. In 1988 Halasz identified seven issues for the next generation of hypertext systems (Halasz, 1988). He classified existing systems as first generation, which were large, often mainframe based, and used with large teams of collaborators. Second generation systems were similar but had explored different interfaces, especially integration with different types of media. The problems inherent in these systems, which he classified as seven challenges, inspired the design of the next generation of hypertext systems.

3.4.1 Open Hypermedia

In the early 1990s, research into Open Hypermedia systems started taking off. This movement aimed to introduce hypertext functionality into all applications across the desktop, leading to a separation of links and content. In Open Hypermedia architectures, information about links between documents is stored and managed separately from the documents themselves, which remain in their native formats. Links become objects in their own right, just as important as the documents they belong to. A link consists of a set of associated anchors, such as locations in documents, and some related information. Links are stored in link databases (linkbases), which can be placed on servers called link servers.

There are various implications for this approach to both authors and readers, especially when working with a complex, distributed information system. A single link can be applied over different documents and links can be easily applied to a wide range of media types including images or video. Processing of link information is possible, such as searching and dynamic indexing. New links can be applied to documents by swapping linkbases, with no manual authoring. Links can be added to read-only media, such as CD-ROM or web pages. Open Hypermedia systems facilitate maintenance: rather than edit all links embedded in documents, linkbases can be edited directly. Also, linkbases can be processed and checked, avoiding broken links to documents that have been moved or deleted.

Another important aspect is that links can now be applied to documents in several different ways. The most common example is the use of generic links, where a source anchor is defined as any occurrence of a particular text string in any document. Whenever the system finds the text string, it automatically adds a link, which can be followed to the link's destination as defined in the linkbase. Advantages of generic links are that for the cost of a single link, many linkages are available; also, as new documents are introduced they immediately have access to all the generic links that have been previously defined (Davis et al., 1992a).

There are some problems with Open Hypermedia; applying too many links to a document will present users with many choices, which may overwhelm them. Certain words have different meanings, causing problem when applying generic links. In these cases the system needs to determine the context around a word so that the most relevant links can be applied.

3.4.1.1 Microcosm

Microscosm is an Open Hypermedia system developed at the University of Southampton that was designed with the problems and limitations of existing hypertext systems in mind (Fountain et al., 1990). These included the fact that authoring hypertext and converting normal text into hypertext required a significant effort, limiting the amount of data available to a user. So far, most systems were closed and ran as stand-alone packages that didn't communicate with other applications; this resulted in poor extensibility, for example in supporting new types of media. As proprietary data formats were used, systems could not communicate with each other or share information unless it was converted, which often aggravated the authoring process. As most systems embedded links in the documents, it was hard to provide or add links to read-only media, such as CD-ROMs, unless the links were already in place.

Microcosm was also built around a set of guidelines, which included having no distinction between authors and users, so that anyone can add, edit or remove links and annotations. A loosely coupled, modular architecture was adapted, so new functionality could be easily added by creating and integrating components; this was accomplished with a multi-process system with no interdependencies between the subsystems. By storing links separately from the documents, two levels of information are created: data (text and multimedia) and metadata (the links, i.e. relationships between data items). This enabled the creation of tools to analyse and manipulate the linkbases.

Microcosm could apply links to documents in a number of ways (Davis et al., 1992a,b). Specific links use specific points in documents, for instance defined using offset from the start of a text file or regions in images, as the source and destination anchors. Dynamic links are automatically created by the system, for instance by analysing and searching information held within the database and linkbases; if a user highlights a word the system could search for relevant destinations in several documents. Local links are dynamic links from objects in a specific document connecting to a particular object in a destination document. Generic links, as discussed previously in Section 3.4.1, use any occurrence of specific words as anchors.

Documents would be loaded into special document viewers that would request links from the system. Links being returned to the user would pass through a filter chain. New filters could be written and added to the filter chain, altering the behaviour of the system. Microcosm could be integrated with several types of viewers and applications, including unrelated third party systems. Fully aware viewers were written natively for Microcosm and were able to access all features. Partially aware viewers could be customized or altered to communicate with Microcosm. Unaware viewers were unable to access Microcosm at all but could be used as targets for link destinations; however, once an unaware application was launched no hypermedia functionality would be available.

The work done on Microcosm provided an insight into the advantages and disadvantages of Open Hypermedia in general.

Advantages include the ability to handle large numbers of documents and provide links between them. Links can be inserted without altering a document's source, allowing linking to be performed transparently in native applications. Being able to process linkbases provided powerful functionality for hypertext applications. Authoring is simplified through the use of linking tools such as local and generic links. Open Hypermedia can be used to link between different types of media such as text, images, audio and video.

Disadvantages include problems integrating the system with partially or non Microcosm-aware applications; for instance, not being able to make a program scroll down to the right place in a document when following a link. If the actual documents are edited or deleted, links stored in linkbases may no longer be accurate; and in distributed sets of linkbases it is difficult to maintain integrity. A solution to this problem is to timestamp links and nodes so that when a change is detected the system will attempt to repair the link or alert the user (Davis, 1995).

3.4.2 Hypertext Interoperability

Interoperability between different hypertext systems has been an important goal throughout the history of hypertext research. Halasz identified the problem of as one of his seven issues for hypertext systems (Halasz, 1988), and it has been regarded as a requirement for industrial strength hypermedia applications (Malcolm et al., 1991).

During the 1990s many approaches for tackling the problems of interoperability between different hypermedia domains and systems were investigated. This included research into a common hypertext model capable of supporting different systems, such as the Dexter Model (Halasz and Schwartz, 1994). The Multicard system provided a protocol for exposing a set of hypermedia tools through a standard interface called the M2000 protocol (Rizk and Sauter, 1992).

The Open Hypermedia Protocol (OHP) (Davis et al., 1996) was based on the observation that each existing Open Hypermedia system at the time required its own proprietary clients. This made writing new Open Hypermedia systems complicated as a new set of clients for that system must be written from scratch, and creating a new type of client would require the design of the underlying hypermedia model supporting it. OHP proposed to solve this issue by presenting a common interface for communication between clients and servers, thus enabling client reuse between different hypermedia systems. This allowed researchers to focus on other areas, such as designing more advanced link services.

Different hypertext domains had emerged by this time. Besides traditional *navigational hypertext*, where link-based navigation is used to travel between nodes in documents. Research into illustrating hypertext structures to indicate users' position within large networks of linked documents led to *spatial hypertext*, where the visual layout of nodes is used to express relationships in the information. For instance, distance can be used to show similarity of different objects, and graph displays can be used to show the interconnectivity between the visible nodes. *Taxonomic hypertext* was another new domain, where categorisation of similar types of nodes is used for rich navigation methods, such as intelligent set-based querying.

To handle these different types of hypertext, OHP was split into different domains, with OHP-Nav being developed as a text-based protocol for navigational hypertext. However, it has been argued that the community should have concentrated on the model of hyperstructure rather than the protocol (Millard and Davis, 2000), particularly as several areas of the model, including Context, Behaviour and Computation, were never formally agreed. This led to the development of independent extensions to OHP-Nav by institutions within the OHS Working Group and efforts to provide a general or extendable model for all hypertext domains.

3.5 An Open Hypermedia Approach to AR

The Intelligence, Agents, Multimedia group (IAM) has a long history in Open Hypermedia systems research, and was the birth place of landmark systems such as Microcosm. Conducting AR research within the IAM group has naturally provided a unique view of AR from an Open Hypermedia perspective. Open Hypermedia techniques have a lot to offer in terms of storing and presenting information in AR.

In AR systems, the augmented information must also be kept separately from the real world objects being described. Many AR systems project virtual objects into the real world using 3D models. It is complicated to store information related to object features that can be dynamically added to these 3D models. This problem intensifies where the AR system augments information over real objects. If Open Hypermedia can associate information to virtual models projected into AR environments, the same techniques are applicable in overlaying information over real world objects.

As described above, Open Hypermedia provides various useful functions for authoring and viewing associative information. Open Hypermedia has proved successful at providing links between different types of media, so it should be suitable for tackling the problem of linking between the real and virtual worlds. Processing the associations (i.e. linkbases) can result in easier maintenance and different ways of applying information, both of which may prove useful for AR. Open Hypermedia concepts for linking, such as generic links, provide many advantages in authoring content; for example, with generic links an object can be added to a system that has no information explicitly associated to it.

Open Hypermedia facilitates adaptive hypermedia in several ways: a simple form of adaptation can be achieved by using different linkbases with a document, depending on the audience. For instance, an expert would view a different set of links to a novice. There are more intricate ways to perform this: contextual Open Hypermedia systems can associate a context level to a link, so that that link is only available under a certain context. In fact, some research suggests that most forms of adaptive hypermedia can be implemented with such contextual Open Hypermedia systems (Bailey et al., 2002).

There seemed many advantages to integrate an Open Hypermedia link server into an AR environment. I started investigating developing my own link server and hypertext model for AR. At that time, the Fundamental Open Hypermedia Model (FOHM) based Auld Linky link server was being developed and early versions were released.

3.5.1 The Fundamental Open Hypermedia Model

The Fundamental Open Hypermedia Model (FOHM) grew out of work carried out on OHP, and addresses the problem of interoperability between different hypertext domains.

FOHM defines a common structure model and a set of related semantics capable of consistently describing the three types of hypertext previously described in Section 3.4.2.

The advantage of a common data model such as FOHM over the OHP architecture is that multi-domain structure can be created and new domains explored. One early demonstration of FOHM allowed domain-specific browsers (e.g. a navigational hypertext browser) to interpret and display information from other domains.

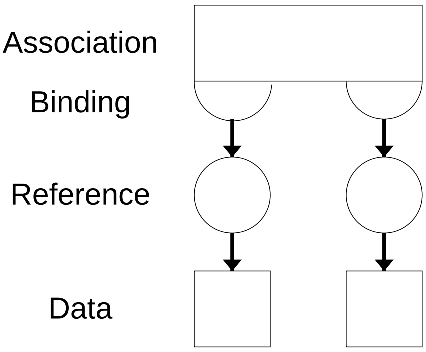
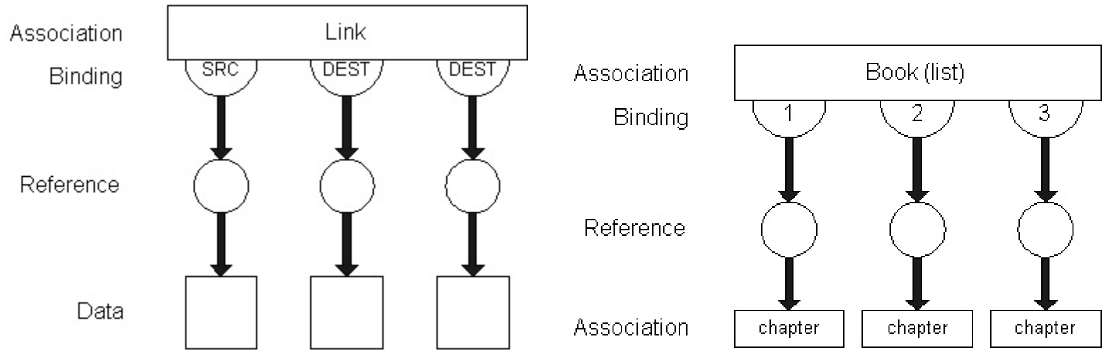


Figure 3.4: The basic FOHM model

FOHM uses four objects to describe hypertext structures. *Associations* represent relationships between *Data* objects, which are wrappers for pieces of data lying outside the scope of the FOHM model. These normally represent a document but any file, stream or even individual items could be held within a Data object. Instead of placing Data objects directly in Associations, *Reference* objects that point at either whole or specific parts of the Data objects are used, for example a certain paragraph within a document, or sections of a film or audio clip. They are attached to the Associations via *Bindings*. Each Association has a structure type and a feature space; each Binding must state its position in that feature space, effectively stating how it is bound to the Association structure. The basic FOHM structure is shown above in Figure 3.4.

Essentially, Associations bind together different objects, with Bindings attaching References to the Association. The references point to objects in the system, and although in the diagram above the objects are items of data other FOHM objects can be used, for example an association.

The most common type of link in navigational hypertext is a directed link that can be represented by specifying each Binding's "direction" feature as either "source", "destination" or "bi-directional". Figure 3.5(a) shows an association that describes the relationship between a source binding and two destination bindings, where the data items attached through references form the source and destinations of the link. Associations can be used to model many types of structures such as lists or maps by using appropriate reference features, such as position, colour or shape. Figure 3.5(b) shows a FOHM representation of a book's structure, which is composed of a list of chapters. Note that the references for each chapter point to associations rather than data items as was the



(a) A navigational link in FOHM

(b) A book structure represented in FOHM

Figure 3.5: FOHM examples

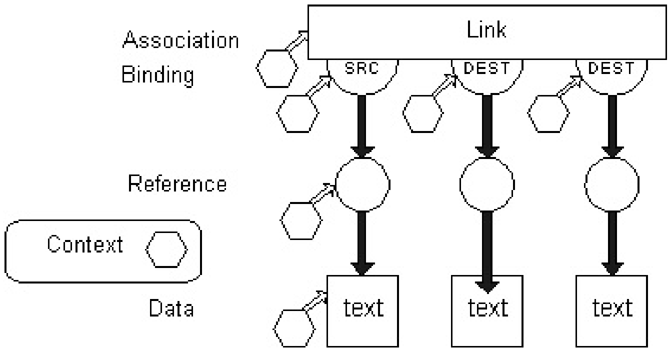


Figure 3.6: FOHM context

case in the first example.

FOHM differs from other Open Hypermedia models by allowing Context objects to be attached to any part of the FOHM structure, as shown in Figure 3.6. Context objects contain metadata that is used to narrow down a query by returning only the links relevant to a certain context; for instance, context could be used to specify that certain destinations of a link are only appropriate for adults. The details about context objects are defined by the implementation of the FOHM model. In addition to context, behaviour objects can also be attached to FOHM structures to trigger actions; these are interpreted by clients so the content of the behaviour object does not need to be understood by the FOHM model implementation itself.

Auld Linky is a FOHM-based link server developed at the University of Southampton for experimenting the effect of context on hyperstructure. It was designed to be stand alone, simple to use and versatile enough to be used in different projects within the IAM group. It is implemented in Perl and consists of a number of components that can be compiled into a single executable; these components include a FOHM API to store, look up and match structures together, and a query interface that exposes the link server as a webserver-like process using HTTP and XML.

Powerful pattern matching techniques are used for querying the link server for relevant links; FOHM structures are constructed in the query and matched against each structure stored in the linkbase. Context can be used to limit the number of matches and is implemented as a set of attribute value pairs. Context can also be extended to use constraints when the default matching (a string comparison) is not sufficient; for example, greater or lesser than comparisons can be invoked. This can also be used to implement level of detail structures, where links stored in the linkbase are associated with different values depending on the destination of each link; the links applied to a document could then be restricted to those relevant to a users' interest level.

There are two ways to use context with Auld Linky. Context objects can be attached to any part of the FOHM structure sent in the query, so it can be matched against the structures in the linkbase as before. The second approach is to attach a context object on the query itself, which acts as a filter on the query results.

There were many reasons for using FOHM and Auld Linky in my work. Instead of designing and developing my own linkbase format and link server, which would have been beyond the scope of my research, it has allowed me to concentrate on aspects of AR of more interest to me, such as interaction. FOHM is strongly supported within IAM and has been used in collaborative projects outside the group, so assistance has been available when necessary. A lot of interesting research and various projects have been based around FOHM, and I felt that this work could be applied to and benefit information display techniques for AR environments.

FOHM and the Auld Linky link server facilitate linking between different types of hypermedia domains. As described above, I have considered Open Hypermedia to be a useful way to associate virtual information to real world objects. The link server can be used to explore linking between the real and virtual worlds, which has also been the focus of research conducted within the IAM group as part of the Equator project.

FOHM context is an extremely flexible tool. It can be used to model levels of detail in content, which is useful in adapting content to users' preferences. Many forms of document structure, such as book chapters or slides in a slideshow, can be defined in FOHM and marked up with context information. This can be used to facilitate versioning of documents, that is different views on a stored structure according to the viewer's perspective (Griffiths et al., 2002). There are some interesting implications of context to AR systems; these have been explored in my work and are described in Section 3.5.2.

An important benefit of FOHM is being able to reuse information and FOHM structures to produce different types of presentations, including over a wide range of displays. For instance the material used in a web site could also be applied to an AR environment or hand-held PC displays.

In terms of implementation, Auld Linky is a web server-like process that uses HTTP and XML to expose linkbases to other applications. This makes integrating Auld Linky with AR environments straightforward, as HTTP and XML are well supported on many platforms. A disadvantage is that the overhead in communicating over HTTP can result in lag for real-time applications, although this can be addressed with caching.

Millard et al. have been investigating interesting uses for FOHM, such as narrative and adaptive hypermedia (Weal et al., 2001). In the future, it might be interesting to apply some of this work to AR environments. Stories and documentaries could be presented around objects using AR, and more advanced adaptive hypermedia techniques would result in more effective adaptation of information to users' interests (Bailey et al., 2002).

Authoring FOHM structures can be problematic as no dedicated tools are available; linkbases are written in XML, which can result in complex files. This can be beyond the technical knowledge of many users. Writing in pure XML can be time consuming and unforgiving, although new techniques are being introduced that semi-automate the process. Authoring material for AR systems, in particular the use of 3D models, will no doubt complicate this process. It may be useful to provide authoring interfaces where users are able to view the 3D models together with the material being created.

3.5.2 Linky and Augmented Reality

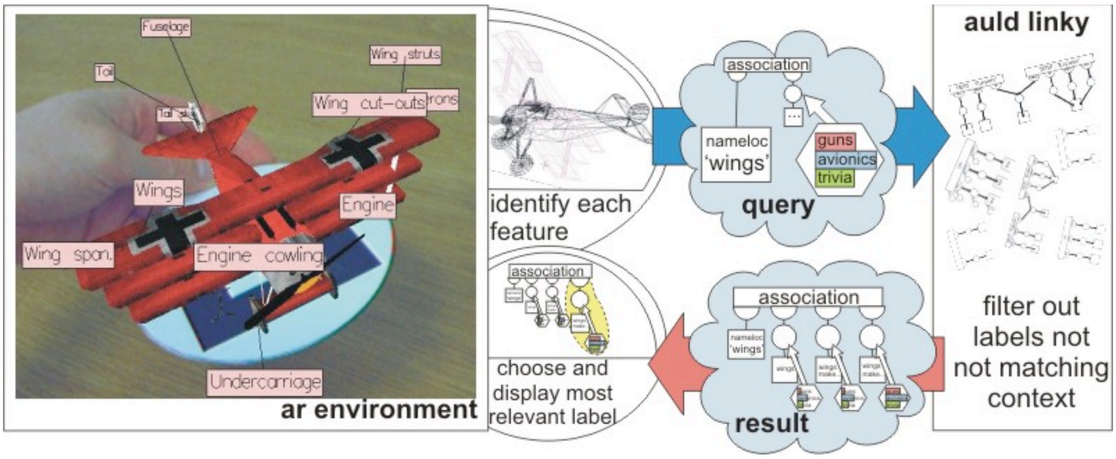


Figure 3.7: System architecture overview

Following the decision to use the Auld Linky link server, methods to apply information to objects in AR environments were investigated. The concept behind the chosen technique, illustrated in Figure 3.7, uses a link server to provide information to the AR system.

The AR environment being used is based on the ARToolKit library developed at the University of Washington, which is designed for the rapid development of augmented reality applications. It provides computer vision techniques to calculate a camera's position and orientation relative to marker cards so that virtual 3D objects can be overlaid precisely on the markers. The ARToolKit can distinguish between the different marker card patterns so that it knows which virtual object should be placed on which physical marker.

In a typical ARToolKit application, users wear a head-mounted display (HMD) with a camera mounted onto the HMD. Video from the camera is displayed on the HMD providing the illusion of the display being transparent. Graphics are overlaid by the system on any visible marker cards. The system is also responsible for tracking the marker cards using the camera information. Users will often sit at a desk, providing a comfortable workspace to manipulate the ARToolKit marker cards in front of them. Alternatively, users might wear a lightweight version of the system as they moved around a museum space. An ARToolKit application can also use the marker cards' physical properties, such as their orientation and position relative to other cards, to trigger events. For example, bringing two cards together could make virtual objects interact.

The ARToolKit was chosen because at the time it was the only freely available open source AR system. A significant advantage is the low requirements, especially compared to other systems that require costly tracking devices: all that is required is a PC with video input, resulting in relatively low costs for deployment under real conditions such as museums. A wide variety of cameras are supported through the DirectShow interface. The only special requirement is an HMD, although the ARToolKit can also be used on monitors or projector displays.

ARToolKit uses the OpenGL API, which makes it flexible for creating custom model viewers and visual effects for interactions. The ARToolKit is available with a built-in VRML library; this library has been used for displaying all models in my research. VRML, although relatively primitive compared to recent developments in 3D content display, is extremely flexible. The display of high quality textured models is possible, and there is good support in terms of modellers and conversion utilities. A wide range of 3D models are available online, although high quality models are usually not available for free. VRML also affords simple interaction through scripting and animations.

There are several ways to deploy an ARToolKit application in a museum environment. As described in Section 3.1, virtual object models can be projected onto markers that users can control and manipulate in their hands. It is important that these virtual objects are as realistic as possible to give users the illusion that the objects are real; high quality texture-mapped 3D models are used to accomplish this. Markers can be placed around the real museum artefacts, whose position can be carefully calibrated so that information can be overlaid over the real object. In practice, this approach can be troublesome: many markers must be used so that at least one remains visible at any time to ensure tracking, which is impractical within museums. Another problem is that often museums will place many items together, so it can be challenging to display information over a crowded display case for instance. Future versions of the ARToolKit have been announced that track flat textures, for instance paintings, instead of marker cards. This will be extremely useful in museums, especially art galleries, as no markers need to be placed inside the museum environment.

For augmenting information over real world objects, AR systems can use an internal 3D model of the object to arrange how information can be placed around it. By tracking the object in the real world, the AR system uses the virtual model to draw the information registered over the object. This technique is often used in other areas of AR, such as rendering shadows and lighting over virtual objects. The AR system needs a virtual model of the scene, with the position of the light source and any surfaces surrounding the object, so that shadows and reflections are drawn accurately on the object.

Various issues must be considered for acquiring models to be used in AR environments. Manual techniques involve artists creating objects in a 3D modeller, which can be expensive as this can be a time consuming process. There are several automatic techniques, such as laser scanning and video based scanning, and although the equipment can be expensive it is indispensable for capturing large collections of objects. Accuracy, including geometry, textures and colour, can be extremely important in environments such as museums.

My technique to apply information to AR environments relies on the use of 3D models that are either projected completely into the scene or used for overlaying information over real world objects. Text labels describing each feature are placed around the object with a leader line pointing to that feature, resulting in a spatialised view of the information around the object. Overlaying the object with augmented information clearly presents the relationship between the data and the object, so it is important for the information to be displayed over or alongside the objects. This approach also allows users to learn about the specific details of an object, and point out interesting features that they might not notice.

3.5.2.1 Object Models in AR

Labels are served by the link server as links between the features of objects displayed in the AR environment and textual content. The process of attaching a label to an object feature on the 3D model is as follows.

The location of object features to be described by labels must be defined so that the AR system can decide a suitable position to place the label. The aim is to place labels as close as possible to their respective feature, so that users have a clear idea of the association between the labels and features. To accomplish this, the 3D object models are split into their various components. For instance, the triplane shown in Figure 3.8,

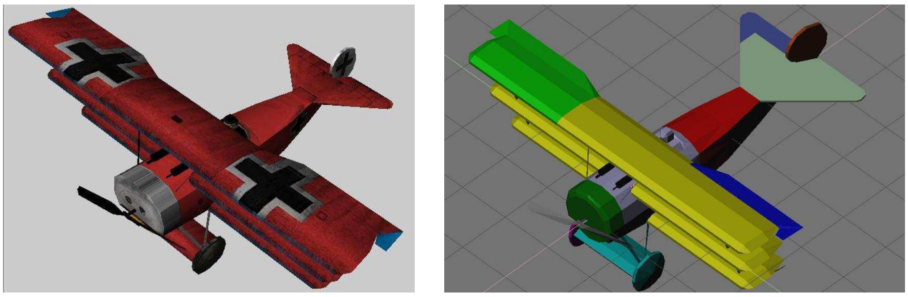


Figure 3.8: 3D model: complete (left) and split into different features (right)

has wings, a fuselage, propeller and so on. Each of these features is separated from the object mesh.

To avoid damaging the quality of the object displayed in the AR environment, a copy of the 3D model is loaded into a 3D modeller package, where it is edited into its separate components. The package that has been used during in my work is the recently open sourced Blender, which was chosen because it is available for free; it is powerful and extensible through scripting for features such as importing and exporting different types of files. The interface can be customized in various ways, which might be useful in the future to develop an authoring environment aimed at this process of authoring content for AR interfaces. The actions performed in Blender are available on most 3D modellers, such as 3D Studio, so the techniques described here are transposable.

In the modeller, each feature to be described is given an identifier, which used in conjunction with the object's identifier can be used to uniquely identify a feature. Currently, feature names are used as the identifier; such as an aircraft's 'wings' or 'propeller'.

This approach has been appropriate for the few models used in the demonstration applications that have been implemented so far. However, for deployment in real environments, such as museums, a more robust form of identifier naming should be considered. For instance, different sources or institutions may use different standards for naming; it is important that naming is standardized, especially to apply generic links to any kind of object. Naming is an established problem in Open Hypermedia, and has been investigated by several projects and researchers. (Tzagarakis et al., 2000). Future work should investigate how these approaches to naming could be best applied to AR.

At this stage, with the identifiers in place, the model is exported as an X3D file. X3D is a next generation, extensible 3D graphics specification based on XML; it is being developed by the Web3D Consortium (Web3D) and the World Wide Web Consortium (W3C). It is the successor of VRML97 and it is backward compatible with current VRML viewers. As X3D is an XML format it is easy to parse and manipulate.

In the AR environment, both the display and the tagged X3D model are loaded, although only the display model is visible. The AR system is able to use the feature positions in the X3D model to place the labels suitably around the displayed object.

3.5.2.2 FOHM Structures

All label and link information about the objects is stored on the link server. The AR environment queries the link server to obtain a set of links and labels for an object.

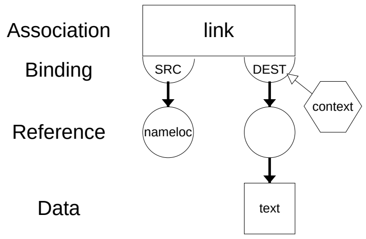


Figure 3.9: Annotation example

Each label or link stored in the link server is represented by an association, with a source binding containing the feature's identifier as a FOHM *Nameloc* and at least one destination binding. An example is shown in Figure 3.9. FOHM Namelocs identify selections within any object or file by name, and in (my approach) have been used in two ways.

By only specifying the feature name, labels can be generic: for example, a description about aeroplane wings would have *'wings'* as the feature identifier could be applied to all aeroplanes with wings. Generic labels are extremely useful for objects with similar features as generic descriptions can be applied to all similar objects without having to explicitly author descriptions for each object. New objects can then be labelled without any material having been specifically written about them providing the features of the object are suitably tagged.

Labels can also be attached to a specific object by authoring both the object and feature identifier in the Nameloc. This ties feature to objects, so features specific to an object can be described. For example, the historic information about an object is usually only relevant to that specific object.

The descriptions are stored as FOHM Data objects. Currently, the text is authored into the linkbase itself as a text string within the content of the Data object. This material can also be stored within external documents and remain accessible through FOHM reference objects, and this may be explored in the future. Descriptions are currently limited to plain text only, although FOHM supports any type of data format. Other types of data, such as images, animations, or even 3D models have been considered, but support for these are necessary in the AR environment. New user interface techniques are required to accommodate these different types of media, which would be interesting to investigate in the future.

Besides links to textual content being displayed as labels, links between different object features have also been investigated. *Link labels* have are stored in the same way as descriptive labels, but have a destination as well as a source binding. Nameloc objects are used in the same way for both source and destination bindings, allowing for generic and specific linking. Currently only links to other objects accessible within the AR environment are supported, and this might be extended in the future to include links to other types of destinations, such as real world locations and objects and other information sources such as books or the web.

3.5.2.3 FOHM Context

The AR environment queries the link server with the object feature and receives any matching labels and links. As described in Section 3.5.1, FOHM enables the link server to narrow down what is returned by the use of context objects. In terms of AR applications, there are several ways in which context can be used.

Context objects can be used to tailor descriptions to the user's preferences. For example, each description binding can have a context stating the type of user: children or adults. When the query is made, only descriptions suitable for either one (or both) will be returned. This approach was briefly explored in a short paper and poster presented at the adaptive hypermedia workshop in 2001 (Sinclair and Martinez, 2001) (see Appendix A).

This notion of adaptation through context can be used with a finer grain than simple child versus adult adaptation. In most of the demonstration applications implemented, context objects mark different levels of detail on descriptive text used in the labels. Chapter 4 describes various approaches for allowing users to manipulate the level of detail they wish to see about an object.

Context can be applied to any parts of the FOHM structures, including the source and destination bindings for a label. For instance, there may be a label that when a certain level of detail is reached, should be enabled as a link. A context node can be attached to the destination anchor, so that when that context is matched the destination anchor becomes visible, transforming the label into a link.

The types of information displayed with an object can also depend on what users are interested in at the time. There could be several subjects that relate to an object. For instance, an aeroplane might have mechanical, armament and trivia information. This is also modelled with context, so for instance a description of an engine would be tied to the mechanical context. Levels of detail can also be applied to these descriptions so that users obtain the information relevant to their interests; for instance, a user may wish to know a lot about the mechanical details of an aircraft but are only slightly interested in the armament information.

3.5.2.4 Label Placement

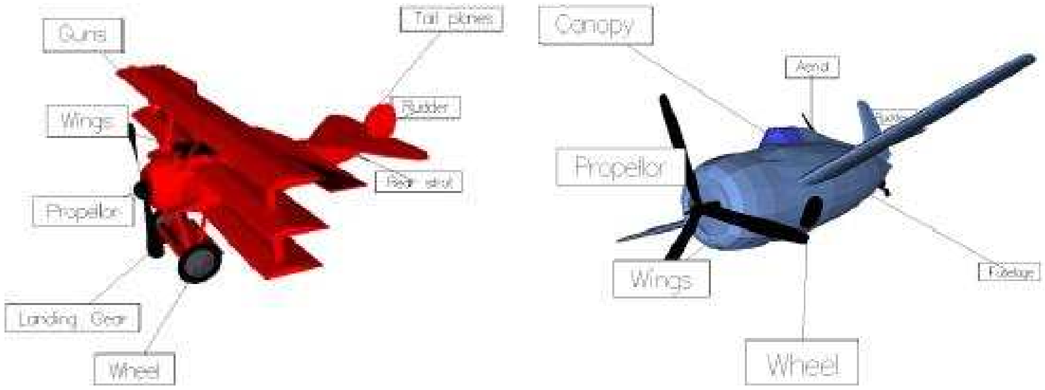


Figure 3.10: Labelling examples

As the AR environment receives label information from Linky, it must decide how to place this information around the object. The results of a query is shown in Figure 3.10.

For the initial implementation, a simple label placement algorithm used the position of features in relation to the object to calculate where labels should be placed. To avoid occlusion and collision with the object, all labels were placed outside the object's bounding box. When placing labels, a check was made to ensure that labels did not occlude one other. If a collision with another label occurred, the label was shifted until free space is found for it.

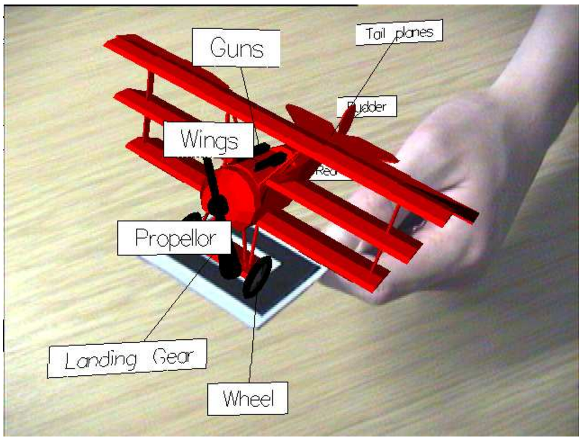


Figure 3.11: Labelled triplane

Figure 3.11 shows an aeroplane model being held in the AR Environment. Labels are drawn as billboarded labels with leader lines drawn between the label and their respective feature on the object. Note that as labels are also 3D objects, labels far from the screen appear small. Also, if a label is behind the artefact, it will be occluded by it. For example, the 'Rudder' label on the aircraft in Figure 3.11 in the diagram is smaller than the 'Wheel' label. The 'Rudder' label is drawn behind the aircraft's rudder.

As the focus of this work is the integration of the information with the AR environment, this simple labelling approach has been sufficient for the demonstration applications. However, complex view management techniques, such as the work described in (Bell et al., 2001), could be applied to more conveniently place and display the labels around objects. In addition it might be interesting to combine the contextual link server approach with existing information filtering techniques to reduce display overload in AR scenes (Hollerer et al., 2001).

3.5.3 Implementation Details

To obtain information from the link server, the AR environment needs to perform queries and parse the results so they can be applied. As previously described, Auld Linky is a stand alone process and queries and results being passed as FOHM XML strings over HTTP. This requires the AR environment to be able to communicate over HTTP and have an XML parser. During the course of my research, different approaches have been used to perform this.

Initially the FOHM XML parsing was performed by a Java servlet that would generate the label information as static X3D files, which were then converted to VRML using XSL stylesheet transformations and overlaid over the realistic object model in the ARToolKit or in a VRML browser. This XSL conversion was necessary as the ARToolKit didn't support X3D; X3D was generated rather than VRML as it was easier using the Java XML parser. The labels would be static and generated when the object was loaded, although a new set of labels would have to be generated at runtime this would have been slow. With this approach no user interaction could be performed as the labels were presented as static VRML objects, and although VRML scripting could be used there was a desire for more flexibile user interface.

This led to implementing the label display under OpenGL directly in the ARToolKit, which required the use of a C/C++ XML parser to perform the FOHM parsing. The use of OpenGL allowed for dynamic user interaction with the labels, which is described in Chapter 4. However, parsing FOHM with the C/C++ XML parser was not ideal as there is no C/C++ FOHM API; only the limited subset of FOHM that was necessary for my linkbase of labels was implemented. I felt that more complete FOHM support would be required for later work, especially any collaborative work that might occur in the future. This would require a thourough FOHM C/C++ API, but I was reluctant to spend much time creating an API that might require continuing maintenance.

While there was no C/C++ support, a strongly established Java FOHM API was widely used within IAM. I decided that if I could integrate this with the ARToolKit, I would solve the problem of creating my own FOHM API and also enable possible future collaborative work with other researchers comfortable with the existing Java API. In addition, performing XML parsing with Java is more straightforward than with C/C++, so the implementation would be simpler.

To integrate the C/C++ based ARToolKit with the Java API, I am using the Java Native interface (JNI). Through JNI, the ARToolKit is able to control a Java class for querying and parsing the results from Auld Linky; these results are converted into C/C++ data structures with JNI so they can be used in the OpenGL ARToolKit environment.

Performing the querying and parsing under Java rather than C/C++ does incurr a slight performance drop, but so far this has been negligible with the linkbases used in the prototypes. However, the time taken for an Auld Linky query to complete can be an issue, mostly due to the overhead of communicating over HTTP. This can cause the ARToolKit process to halt temporarily as queries are made and results in a slight pause in the AR display. To overcome this, a threaded querying approach was attempted where the ARToolKit and the querying processes ocurr in parallel. However, this caused queries to take longer to complete; in many cases users would wait over a second for a label to be applied after they had requested it. This was found to be unnacceptable for users, especially under certain interfaces that require a constant updating of the object information. As a temporary solution for the demonstration applications, all label information is cached when the ARToolKit is started, and all queries are made on the cached information.

3.5.4 Discussion

The technique presented here for applying information to objects in an AR environment is, in my opinion, flexible and powerful. Although it has been used for relatively simple labelling of 3D objects I believe it could be extended for a wide selection of information types and advanced user interface techniques.

Most existing approaches to AR information presentation, as described in Section 3.2, have focused on projecting high quality visual representations of objects into real world scenes. Certain systems have used simple labels to present information about objects, but these often rely on other display mechanisms for showing more complex information; examples of such systems include MARS, ARCO or ARCHEOGUIDE. AR environments that have explored complex presentation of information have been troubled by authoring of the augmented material; for example, the ARVIKA demonstrator needed many different 3D scenes to be created for a single presentation. Recently, the AR community is beginning to focus more on this area, with several projects addressing the issue of authoring and presentation of material in AR. The PowerSpace system is one of these. However, the material is placed statically around objects and there are limitations in linking between the information. The contextual Open Hypermedia approach offers many advantages, as discussed above. These have allowed me to explore aspects not looked at by earlier systems such as KARMA or the engine labelling system by Rose et al (discussed in Section 3.2).

In terms of the authoring effort, my technique requires 3D models to be broken down into their subfeatures, which can be time consuming and expensive. This process appears similar to the ARVIKA demonstration or the PowerSpace tool, as it requires the use of a 3D modelling application. However, once a model has been prepared it is more straightforward to add new information: only the content (e.g. text) needs to be authored as it is placed dynamically, unlike other systems where many 3D complex scenes need to be manually created. This would be useful in an industrial setting, where there may be many different maintenance tasks for a single object, each requiring a large number of steps. The Open Hypermedia approach also allows generic information to be applied automatically to objects.

Contextual hypermedia techniques can be applied in different ways in AR environments by considering the type of context to be used, such as user context, object context, scene context and task context. User context reflects the interests and knowledge of the user of the system, for instance where information is adapted depending on the user's experience (child and adult, novice and expert etc.). Object context is specific to individual objects in the environment, and can be used to determine how information about each object is displayed. Scene context could be derived from the spatial relationships of the objects within it, for example bringing two objects close to each other changes what information is displayed there. Task context could be used to change what information is shown as users progress through tasks. During my work I have mainly focused on manipulating object context, although I have touched upon handling scene context. In the future I would like to examine how to handle other types of context within AR environments.

During my work, I have only used billboarded labels to present information. In the future I would like to experiment with other means of presenting information. Instead of using simple text for labels, images, videos and other models could be used, enhancing the way information is presented to users. For example, an aircraft engine could be labelled by a model of the engine itself. When the label is selected, an animation could start showing the engine working. The label could also act as a link's source anchor, which could be selected and followed to the engine model that could in turn be labelled and manipulated. The use of 3D models and images as labels is an interesting way to preview a link destinations, as it immediately gives users an idea of what they might find when they follow a link.

Dynamic placement of animated icons around an object could be very useful for maintenance applications, for example arrows could illustrate how to dismantle an object or indicate where screws need to be placed. These types of animations have been used in systems such as the ARVIKA demonstration system but are complicated to author as they need to be manually set up. The use of FOHM behaviours might be a way to manage such dynamic animations around the object.

3.6 Chapter Summary

This chapter has presented a scenario involving museum environments where spatially overlaid AR is used to present information about museum objects. Several AR projects are described that have focused on issues in information interaction, especially around museums. I proposed that the use of Open Hypermedia will benefit information display in AR. A brief overview of the hypermedia field was given.

A technique to place dynamically created, adaptive labels over 3D models of museum artefacts is described. The approach uses the Open Hypermedia concept of keeping data and links separate using linkbases; the linkbase is used to attach relevant descriptions to the respective area of the 3D model. The linkbase is served by the Auld Linky link server, a context based link server implementing FOHM.

Chapter 4

Interaction Techniques

The display of information over the real world is a key feature of AR systems and as a result user interaction with the augmented information space is crucial. In recent AR systems, there has been an emphasis on tangible interaction. During my PhD I have been interested in exploring interfaces where users can explicitly interact with the information directly using tangible AR interfaces. This involves physically manipulating virtual objects to select and highlight information users are interested in. My main interest has been to develop different tangible interfaces to adapt the information displayed about objects.

This chapter describes how the ARToolKit, the AR environment chosen for prototyping, could be used in tangible interfaces. This involved experimenting with the ARToolKit to create several simple tangible AR interfaces that uses the physical properties of ARToolKit markers to trigger interaction events. The experience gained in these was invaluable for developing methods for selecting and highlighting the information exposed by the labelling technique described in Chapter 3. Various issues were explored when creating this technique, including:

- Possible methods for label selection with the ARToolKit.
- Reducing visual clutter when many labels are visible.
- Linking in AR environments, in particular possibilities with the ARToolKit.
- Reflecting mixtures of information in label presentation.

A second aspect of my work on tangible interfaces has been to allow users to adapt the information that is presented to them in the AR environment. This has led to the development of the *"Salt and pepper"* interface, which allows users to shake label information that they are interested in onto objects in the AR environment. Problems with illustrating how much information remains available as users are adding labels led to design and prototyping of other interfaces. This process of experimentation resulted in the design of the *"waves"* interface, where users can alter the information that is applied by changing the distance between objects and context dispenser cards. Refinements to the interfaces were made to improve various usability aspects, such as ensuring that labels do not obscure objects.

4.1 The ARToolKit

The ARToolKit library is introduced in Chapter 2 and arguments for its use are discussed in Chapter 3. It uses computer vision techniques to calculate a camera's position and orientation relative to marked cards so that virtual 3D objects can be overlaid precisely on the marker. A simple example is shown in Figure 4.1.

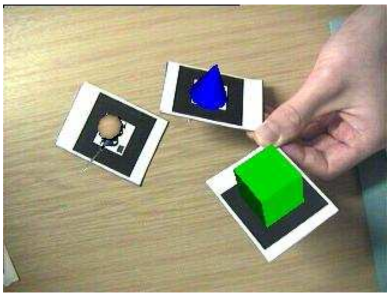
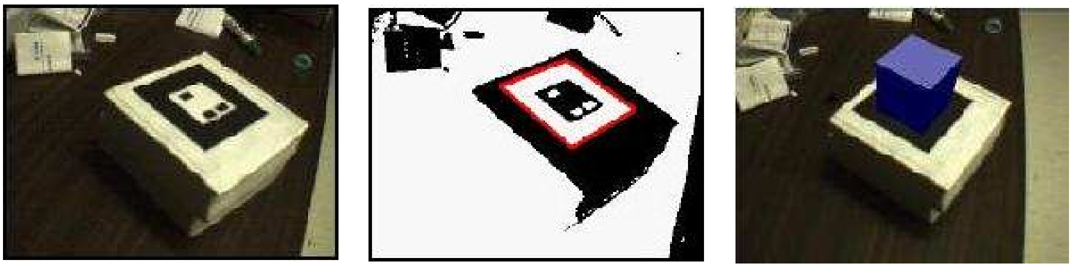


FIGURE 4.1: ARToolKit example

Reasons for using the ARToolKit have already been explored in Chapter 3. These included the open source nature of the library together with its low requirements. This section describes in detail how the ARToolKit works and how various attributes and factors have affected the design and implementation of the interfaces.



(a) Input video

(b) Thresholded video

(c) Virtual overlay

FIGURE 4.2: ARToolKit process

An overview of how the system works, taken from the ARToolKit manual (Kato et al., 1999) and illustrated in Figure 4.3, is as follows:

> "First the live video image (Figure 4.2(a)) is turned into a binary (black or white) image based on a lighting threshold value (Figure 4.2(b)). This

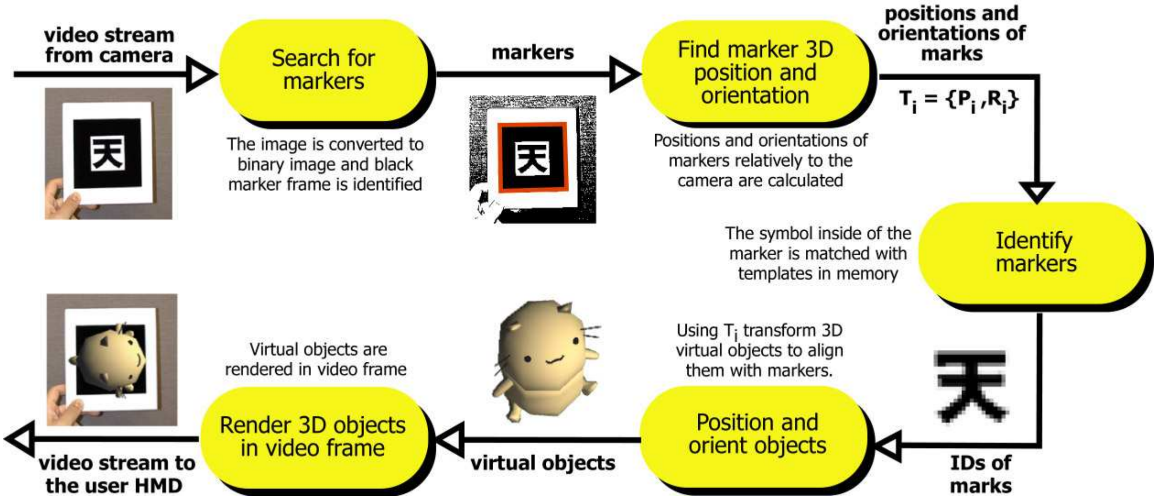

video stream from camera
Search for markers
The image is converted to binary image and black marker frame is identified
markers
Find marker 3D position and orientation
Positions and orientations of markers relatively to the camera are calculated
positions and orientations of marks
$T_i = \{P_i, R_i\}$
Identify markers
The symbol inside of the marker is matched with templates in memory
IDs of marks
Position and orient objects
Using T_i transform 3D virtual objects to align them with markers.
virtual objects
Render 3D objects in video frame
Virtual objects are rendered in video frame
video stream to the user HMD

Figure 4.3: ARToolKit system diagram

> image is then searched for square regions. ARToolKit finds all the squares in the binary image, many of which are not the tracking markers. For each square, the pattern inside the square is captured and matched again some pre-trained pattern templates. If there is a match, then ARToolKit has found one of the AR tracking markers. ARToolKit then uses the known square size and pattern orientation to calculate the position of the real video camera relative to the physical marker. A 3x4 matrix is filled in with the video camera real world co-ordinates relative to the card. This matrix is then used to set the position of the virtual camera co-ordinates. Since the virtual and real camera co-ordinates are the same, the computer graphics that are drawn precisely overlay the real marker (Figure 4.2(c)). The OpenGL API is used for setting the virtual camera co-ordinates and drawing the virtual images."

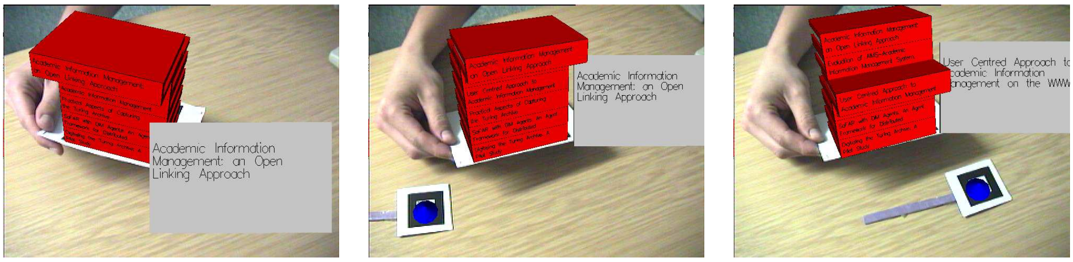
As described in Chapter 3, the overlaid graphics can be created with the OpenGL API or by using a VRML library. Initial work focused mainly on VRML models, as interaction could be implemented with the VRML scripting language. As VRML is a text format, various efforts were made on dynamically generating models and information as VRML models. However, this approach was later abandoned as the OpenGL API provides more flexibility and better performance.

4.2 Early Interaction Experiments

Initial work involved developing simple experiments and prototypes to explore the capabilities of the ARToolKit. The experience gained in building these was invaluable in later work on more complex interaction metaphors.

These experiments examined how the physical properties of the marker cards could be used to trigger interaction events in AR systems. Different types of behaviours and gestures were considered, including marker cards' position in relation to each other, distances between different markers, marker occlusion and orientation. When possible, the prototypes that were developed with the ARToolKit focused on AR information display techniques for museums. This section describes the different test systems that were implemented.

4.2.1 Position


Academic Information Management: an Open Linking Approach
Academic Information Management: an Open Linking Approach
User Centred Approach to
cademic Information
nagement on the WWW

(a) Stack of books loaded in ARToolKit

(b) Select previous book

(c) Select next book

FIGURE 4.4: Simple position interaction

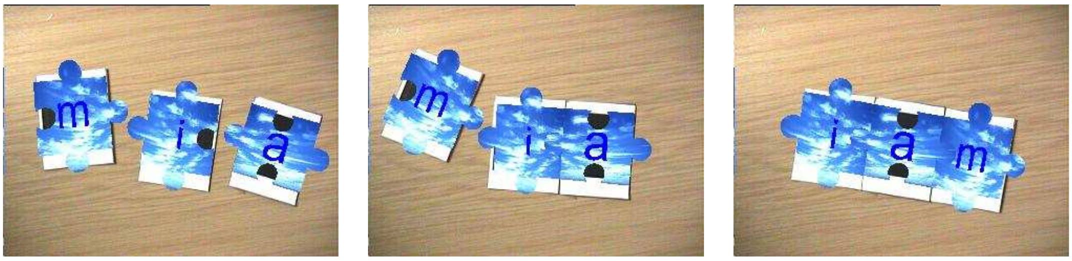
The first prototype used the results of the SOFAR (Moreau et al., 2000) Dynamic CV Agent developed during the third "Agent Fest" held in the IAM group in November 2000. The Dynamic CV agent collects all sorts of information about people from other agents in the SOFAR agent framework. The agent was flexible enough for its output to be used in various ways, such as rendering to the web or into an AR application.

The type of query used in the prototype returned the publication list for someone in the group. These results were used to generate a VRML model of a pile of books, with each book representing a publication. As the publication title may not be readable on the book's spine, it was decided that a floating text box would display the selected publication's full details. The model would be loaded into the ARToolKit where the user would view it, as shown in Figure 4.4(a).

This raised a problem on how to select each publication. One solution was to track a marker card to cycle through the stack of books. By placing the marker card on one side of the stack card, the next book would be selected and its details highlighted on the floating text box. Putting the marker card on the other side would select the previous book. This is shown in Figure 4.4(b) and Figure 4.4(c).

4.2.2 Distances Between Markers

A second prototype was developed that looked at the distances between markers relative to their orientation; for example markers may need to be the right way up to trigger an event in some applications.



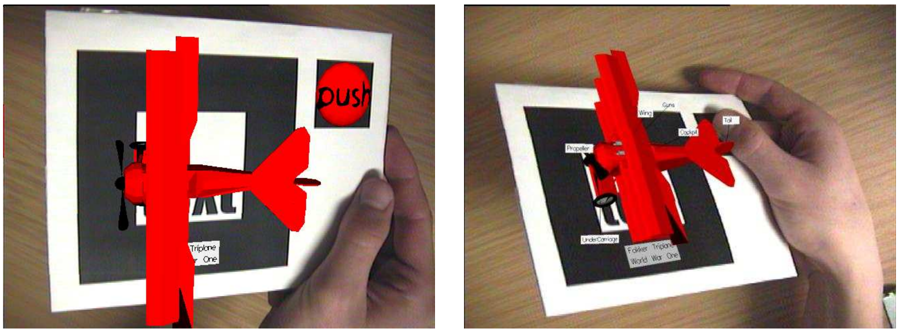
(a) AR jigsaw (b) Connecting two pieces (c) All pieces connected

FIGURE 4.5: Position interaction depending on the marker's orientation

The prototype developed is essentially a jigsaw puzzle in AR, as can be seen in Figure 4.5. There are three marker cards; each card holds one of the three virtual jigsaw pieces. Placing two cards next to each other in the correct order and the right way up will join the two pieces. When all three are together in the correct order the three pieces join up.

This prototype showed that distances between markers can be used to control the interactions between them.

4.2.3 Occlusion



(a) Model (b) When the button is pressed, labels are placed on the model.

FIGURE 4.6: Buttons triggered by marker occlusion

When a marker is occluded or not visible, the ARToolKit will not be able to draw it. Monitoring when certain markers vanish and appear could be used to trigger events and mimic the action of buttons. For example, when the user occludes the button in

Figure 4.6 labels appear around the object. This simple technique could potentially perform all manner of user interactions.

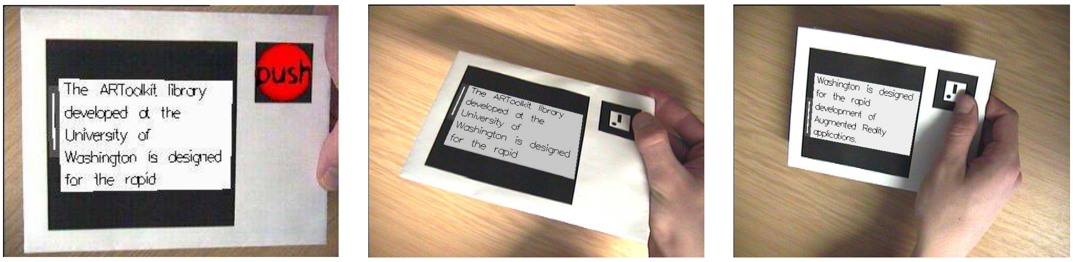
However, the ARToolKit is very sensitive to occlusion: the whole marker must remain in view at all times to be identified by the tracking system. Users accidentally obscuring a corner or part of the marker will cause tracking errors, such as flickering, the wrong object being displayed or even complete loss of tracking. This can be frustrating as the marker may seem visible to the user but due to one obscured corner no object is displayed. Various other conditions may also cause occlusions, such as shadows, swift movement or poor lighting conditions. Any of these actions could trigger the button, producing interface events that the user was not expecting. This sensitivity to error is obviously not acceptable for user interfaces.

Current research in tracking techniques may result in more occlusion tolerant marker-based AR systems. There is even the possibility that these techniques will be able to track occluding objects, for instance hands or fingers, as they move over markers. If this is possible users will be able to use their fingers to interact with user interface elements, such as buttons and sliders, that are overlaid on the marker (Malik et al., 2002; Gordan et al., 2002).

Another possibility is to integrate different types of tracking technologies, such as vision and magnetic trackers. This has been explored in certain projects, such as the Personal Interaction Panel described in Chapter 2. Unfortunately this would increase the cost and complexity of the AR system.

4.2.4 Orientation

The orientation of marker cards was also used to trigger user interface events. Figure 4.7 shows a text box projected onto a card. When the user presses the button and tilts the card down, the text scrolls down; when the card is tilted up the text scrolls up. Similar techniques have also been explored for scrolling on handheld displays (Rekimoto, 1996a).


The ARToolkit library developed at the University of Washington is designed for the rapid
push
The ARToolkit library developed at the University of Washington is designed for the rapid
Washington is designed for the rapid development of Augmented Reality applications.

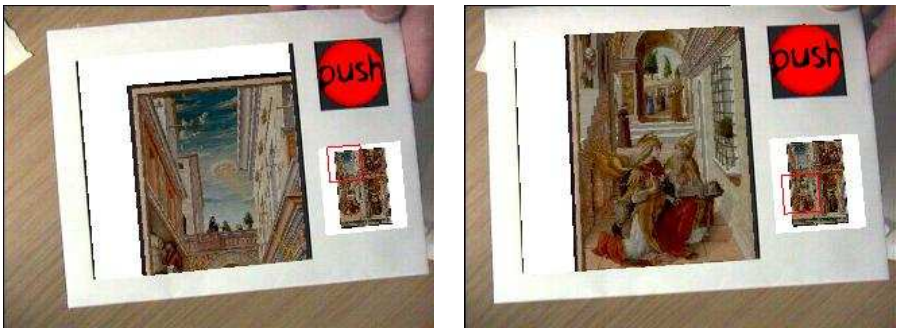
(a) Text box

(b) Scrolling down

(c) Scrolling up

FIGURE 4.7: Scrolling text box

This idea was also used for panning around an image, as shown in Figure 4.8. The image can be panned both vertically and horizontally.



(a) Image panning

(b) Feature of an image

Figure 4.8: Panning around an image

This approach removes the need to click on scroll bars or buttons to pan around text or images. This is an effective use of markers' physical properties as a user interface control, and is more suited to the tangible AR environment than, for example, traditional scrollbars activated by clicking on buttons. There are possible advantages in mobile applications; for instance, a wearable computer user may have trouble accurately manipulating a traditional scrollbar on a HMD or hand held PC if they are walking.

Although this technique was not used directly in later work, it could still be useful in certain applications involving text or images projected onto markers. The largest problem is determining when to enable scrolling: with the prototype, users obscured the button to start scrolling. As explained above, this approach is too prone to error to be effective in a user interface. Another problem is that to perform scrolling with orientation, a frame of reference is needed. As the ARToolKit only calculates the position of markers in relation to the camera, it is impossible to determine what way the user wishes to scroll from the marker's orientation alone. Solutions to this problem may involve background markers, for instance on a table or on the floor, to act as a frame of reference. If when a user wants to start scrolling could be determined, differences in orientation from that time could be used for scrolling. A very simple solution would be to scroll only in relation to the camera, so markers parallel to the camera plane would be still and scrolling is affected by the deviation in orientation from that point.

4.3 Tangible Interaction of Information

Following from the work on labels and the link server, there was a desire to provide interaction with the labelled object. The initial implementation generated a static set of labels around the object that could be viewed by simply manipulating the object to obtain a better view of the desired label. I believe that a more powerful approach to displaying and browsing this information is necessary, and this led to the development of techniques to manipulate labels. This section describes in detail the design and evolution of label interaction methods.

As described in Chapter 3, labels are placed around the objects, close to their respective feature, with a leader line drawn between the feature and the label. This was effective for early experiments with simple labels, for instance containing only each feature's name, as the object remained visible even with many labels, allowing users to clearly view the relationship between the object features and labels.

An important feature of all the prototypes involving labels is that users are able to control when labels are visible. Figure 4.6 shows an interface where users could press a button to hide the labels. Later developments offered more intricate controls but still followed this idea that users can remove the labels at any time, allowing them to have a clear view of the object. This reflected the desire to provide unobtrusive interaction with the information, in the same way that users can simply hide the ARToolKit markers when they wish to only view the real world.

For many applications there is a need to place detailed descriptions in the labels, increasing the label size. This resulted in objects becoming obscured by the labels, causing too much visible clutter. There was also an interest in providing hypertext linking between different objects and features, which requires being able to select and activate links. These issues prompted the development of a mechanism to select individual labels and links on an object.

Various methods of selection were investigated. An obvious choice is to use a pointer interface, where users can point to the feature they are interested in. Pointing devices in the ARToolKit are usually implemented with a marker attached to a stick, resembling a paddle, so it can be held comfortably without the user obscuring the marker (Kato et al., 2000). Unfortunately paddle interfaces suffer from the ARToolKit marker occlusion problems described above. Selecting features from a marker with the paddle will often result in the marker being obscured by the paddle; if the object isn't visible the user can't make their selection. Most ARToolKit paddle systems use multimarker setups, where an array of markers is placed on one card so that as long as one of the markers is visible the object can be tracked. This approach isn't feasible for manipulating objects on marker cards, as large arrays of markers are needed to be effective; multimarkers are mostly used for table top applications. ARToolKit markers must be fairly large to be robustly tracked, especially under poor lighting conditions, so multimarker cards are too large to be comfortably held. Even if smaller multimarker cards were possible, the use of the paddle might still obscure all of the markers, either directly or because of the paddle's shadow.

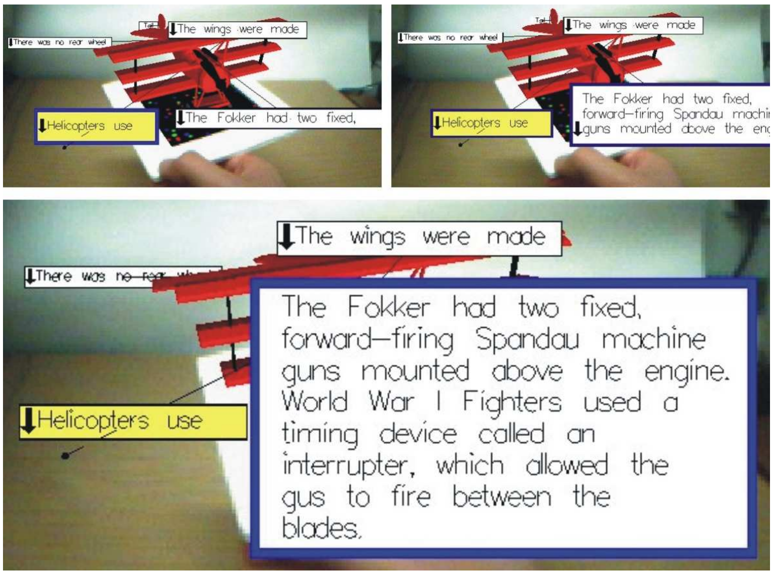
Future versions of the ARToolKit may improve this situation. A new version of the ARToolKit was announced in September 2002 that tracks textures instead of individual markers (Kato, 2002). This would make the tracking more occlusion tolerant, as only a portion of a texture needs to be visible at a time. As processing speeds increase and the use of devices such as firewire cameras becomes more widespread, higher resolution input video is possible, which would allow smaller markers to be used. There are also the new types of marker tracking, described earlier in Section 4.2.3, which may improve tracking reliability.

Even with these advances, I believe that using paddle markers for selection is inappropriate for selecting object features. The features on the object may be small and very close together so the paddle would have to be extremely sensitive. Features may also be hidden inside other features (for example, an aircraft engine is often not directly visible as it is placed underneath the fuselage). A simple paddle system had been implemented as part of the early experiments described in Section 4.2. The experience from this system had not been positive, as it felt rather awkward and cumbersome for selecting individual object features.

An alternative approach is to use mixed tracking, such as the Personal Interaction Panel (Szalavri and Gervautz, 1997) from the Studierstube system (Fuhrmann et al., 1998; Schmalstieg et al., 2000b), but this would increase the cost and complexity of the AR environment. Embedding hardware into marker cards was considered (for instance, a wireless mouse could be integrated onto a marker so that the user interface could determine clicking for selection). Touch sensitive plates, such as the QMatrix system provided by Quantum Research Group (QRP, 2001), are also an option. Areas of the plate could be used to select from a list of available features. Again, there is a desire to avoid embedded or mixed tracking, and the touch sensitive plate approach feels overly complex for a tangible AR interface.

The desire to keep the interface tangible led to the chosen solution, which was also inspired from the early experiments dealing with orientation. With the labelling approach from Chapter 3 all features are described with labels positioned around the object. The problem of selection is simplified if the labels are selected rather than the features themselves, avoiding issues such as selecting small adjacent features and hidden features. Selecting labels is sufficient for my purposes, as there is no point in selecting features for which there is no available information.

Selection is performed using the orientation of the object. Instead of being arbitrarily placed around the object, the labels are set to uniform slots around it. Each label is



There was no rear wheel

The wings were made

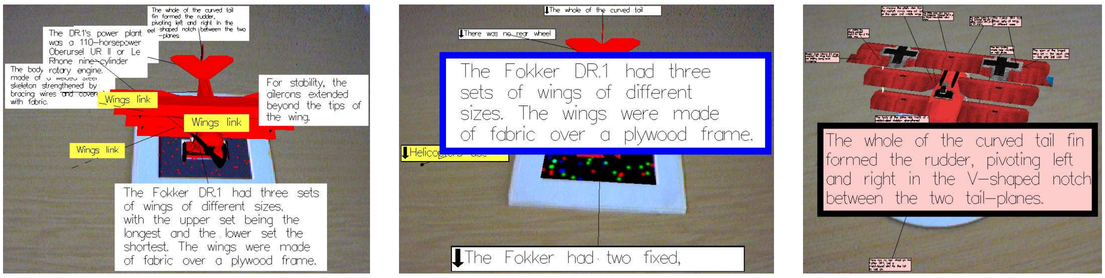
Helicopters use

The Fokker had two fixed,

The Fokker had two fixed, forward–firing Spandau machine guns mounted above the engine. World War I Fighters used a timing device called an interrupter, which allowed the gus to fire between the blades.

FIGURE 4.9: Selecting a label

allocated to the nearest free slot from its position, so that labels are still placed close to their relevant features. The rotation of the object is used to perform the selection, so that the label closest to the centre of the screen is the one selected. The initial implementation used a drum-like approach, where the available slots were spaced all the way around the object. This has been replaced with a semi spherical implementation, where the slots are spaced around the object using an icosahedron.



The whole of the curved tail fin formed the rudder, pivoting left and right in the V–shaped notch between the two –planes.

The DR.1's power plant was a 110–horsepower Oberursel UR II or Le Rhone nine–cylinder rotary engine.

The body made of a welded steel skeleton strengthened by bracing wires and covered with fabric.

Wings link

For stability, the ailerons extended beyond the tips of the wing.

The Fokker DR.1 had three sets of wings of different sizes, with the upper set being the longest and the lower set the shortest. The wings were made of fabric over a plywood frame.

(a) No minimising

The whole of the curved tail

There was no rear wheel

The Fokker DR.1 had three sets of wings of different sizes. The wings were made of fabric over a plywood frame.

Helicopters use

The Fokker had two fixed,

(b) Folding Shutter

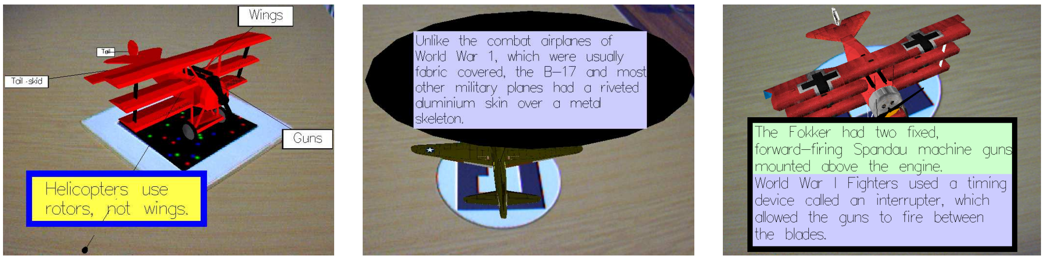
The whole of the curved tail fin formed the rudder, pivoting left and right in the V–shaped notch between the two tail–planes.

(c) Thumbview

FIGURE 4.10: Evolution of minimising unselected labels

Being able to select labels solved the issue of clutter as large unselected labels could be reduced or partially hidden and then expanded when selected to make them readable. This idea went through several iterations, as illustrated in Figure 4.10. Initially, labels would fold away like a shutter, leaving only the first line visible with an icon indicating that more lines of text were visible. The selected label would be drawn with a blue outline. This approach did not give a clear indication of how much more text was visible, as only the first line remained visible. Multi-line labels would be drawn with a small font size when selected so as not to take up too much screen space, but the first line on all unselected labels would be drawn with a larger font; unselected labels seemed to attract more attention which could confuse users.

These issues were resolved by changing the way the labels are drawn. Instead of only viewing the first line of text, a thumbview of unselected labels is used so that the amount of text in a label can be clearly spotted. The currently selected label is drawn with a larger font, avoiding the problem described above. To make the interface less cluttered, all unselected labels are drawn the same size. The selection process is animated: as a label is selected it slowly grows larger and the previously selected label slowly shrinks. This animation process was carefully implemented to avoid false triggering and flickering when the ARToolKit has trouble tracking markers.



Helicopters use rotors, not wings.

Unlike the combat airplanes of World War 1, which were usually fabric covered, the B–17 and most other military planes had a riveted aluminium skin over a metal skeleton.

The Fokker had two fixed, forward–firing Spandau machine guns mounted above the engine.
World War I Fighters used a timing device called an interrupter, which allowed the guns to fire between the blades.

(a) Original link label (selected)

(b) Round link label

(c) Mixed information label

FIGURE 4.11: Link labels and mixed information labels

With an effective selection mechanism, linking could now be explored. Links can be authored as having a description as well as a destination anchor; the description label clearly indicates the reason for the association between different object features. This suited the selection mechanism well, as a label is needed to be able to select a feature. Link labels were designed to look different to normal descriptive labels; early experiments simply used two different background colours: yellow for links and white for descriptions. This is illustrated in Figure 4.11 (a). Link labels also have hashed lines protruding out of them, appearing to continue the line connecting the object's feature to the label.

Later work involved different types of labels being applied to objects, such as avionics, armament or trivia information for aircraft. To make a clear distinction between the different types of information, it was felt that the background colour of a label should reflect the type of information. Certain labels are a mixture of different types of information, so these labels are drawn in a mixture of colours. For instance, the label shown above in Figure 4.11 (c), describes the guns of an aircraft and so this section of the label is drawn in green. Some trivia information is also present in this label; as the trivia section is a different type of information this section is drawn in blue. To avoid confusion links can no longer be differentiated by colour, so a round outline was used for links to distinguish them from labels, as shown in Figure 4.11 (b). The currently selected label's border colour was also changed, from blue to black.

In hypertext systems users must be able to activate the links they are interested in.

In an AR system, there are many ways in which link activation can be achieved. The action can be implicit, for instance if the source and destination anchors are visible the link could be drawn between the two. There could also be an explicit action, such as the user requesting to follow a link to a new object. In mobile AR systems links can also be activated as the user moves around the environment, such as walking from one place to another or looking at objects from different angles.

For the work described in this thesis, the ARToolKit has solely been used for projecting virtual objects onto marker cards, and as a result real objects and real locations have not been explored. This has limited the possibilities for linking, as only links between the virtual objects loaded into the system can be investigated.

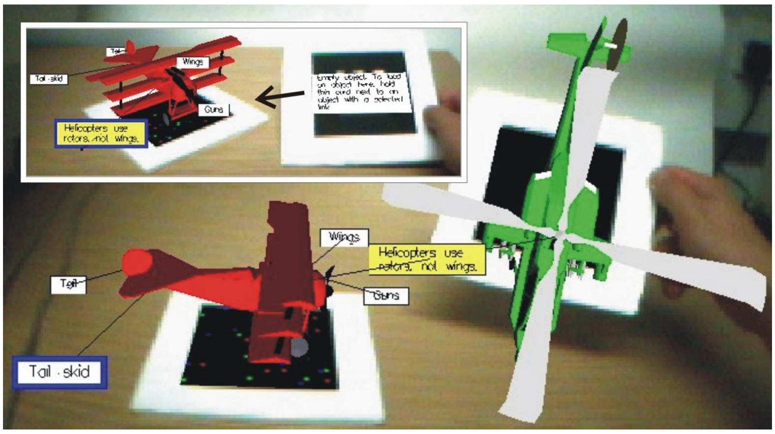

Tail
Wings
Tail - skid
Guns
Helicopters use rotors, not wings.
Empty object. To load an object here, hold this card next to an object with a selected link
Wings
Helicopters use rotors, not wings.
Tail
Guns
Tail - skid

FIGURE 4.12: Following a link

Within these limitations, a mechanism for following links to new objects was implemented. Users start with a labelled object on a marker, and various empty marker cards, i.e. that have not been associated with a virtual object. To activate a link to a new object, users select the desired link label and bring one of the empty marker cards towards the current object, triggering the destination object to be loaded on the empty marker. In this way the user can follow links to new objects, which are displayed on the empty marker cards. Users are able to follow links to as many objects that are available on the system.

AR environments offer great possibilities for displaying the relationships, i.e. links, between objects, and this has been explored within my work. Whenever two objects with active links between them are visible, the links are drawn with an elastic line from the source to the destination anchors, with the description being displayed in the middle of the line. This is similar to Nelson's Cosmic Book system from the 1960s, illustrated in Figure 3.3, where links are visible entities between parts of the document. In my work I have extended this idea to draw links between features of different objects displayed in an AR environment. Links between anchors on the same object are also drawn as a curved line with descriptive label in the middle.

Active link labels drawn between the two objects can no longer be selected by rotating the object as before. This is a problem with large labels, as they may obscure too much of the scene and even get in the way of the source and destination anchors. To avoid this, a different selection technique for selecting floating link labels was tried: the size of active link labels is controlled by moving the anchor objects. When the two anchors are close together the labels are drawn small and when the anchors are far apart the labels are drawn larger as there is more space. This action may feel somewhat strange, as the process of highlighting or viewing a link label involves moving the anchors apart.

4.4 Metaphors for Interacting with Information

Open Hypermedia link structures can be large, complex networks. When adaptation techniques are used, such as FOHM context mechanism described in Section 3.5.1, the complexity of the resulting information space is increased. It is desirable to allow users to not only control the visible hyperstructure, but also the process of adaptation that generates each view. Tangible AR interfaces can be used to expose this adaptation process in novel, powerful ways, overcoming the limitations of traditional approaches. This section describes the work carried out on the design and implementation of tangible interfaces for interacting with the underlying hypermedia information structures.

When considering possible designs for such systems, various aspects were taken into account. These included:

Accessibility The interfaces should be very easy to use and accessible to all types of users, including young children. For example, in museums there are many visitors that feel uncomfortable using computers.

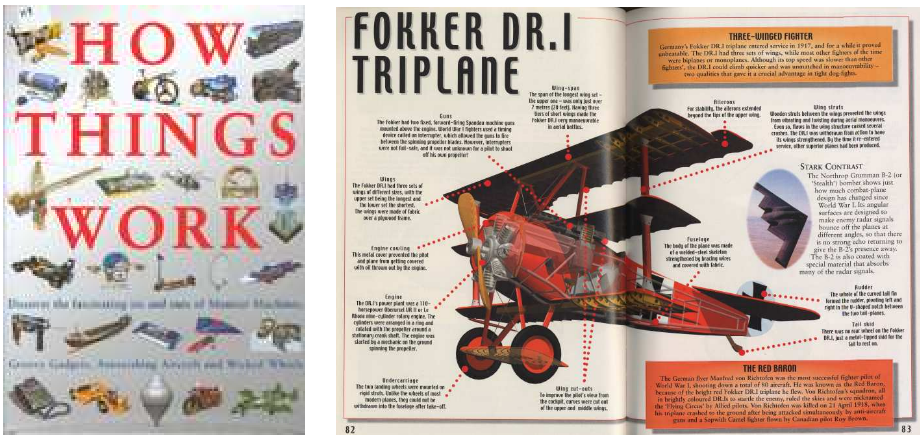
Natural The interfaces should feel natural and intuitive to use, and this is reflected in the systems conceived which make strong use of real world metaphors.

Context mixtures Allowing users to mix different types of information as they wish was a goal in the design of these interfaces, as described in Section 4.5.

Context state It is crucial to give users feedback on the current state of the adaptation process.

Technology The ARToolKit uses an optical tracking system, and its limitations affected the design and implementation of the interfaces.

Work on these interfaces required a subject area from which test material could be used for prototyping. Airplanes had been chosen for the initial work on labelling object features, and there were many reasons for continuing with them. Aircraft generally have many features that can be described and there are many sources of background information. Models, both real and virtual, are easily available and there are some local museums where evaluations and tests might be carried out. There are also many types of complex information that can be viewed in the context of an aircraft: for example military history, mechanical details and so on.



HOW THINGS WORK

FOKKER DR.I TRIPLANE

Wing-span
The span of the longest wing set – the upper one – was only just over 7 metres (20 feet). Having three tiers of short wings made the Fokker DR.I very manoeuvrable in aerial battles.

Guns
The Fokker had two fixed, forward-firing Spandau machine guns mounted above the engine. World War I fighters used a timing device called an interrupter, which allowed the guns to fire between the spinning propeller blades. However, interrupters were not fail-safe, and it was not unknown for a pilot to shoot off his own propeller!

Wings
The Fokker DR.I had three sets of wings of different sizes, with the upper set being the longest and the lower set the shortest. The wings were made of fabric over a plywood frame.

Engine cowling
This metal cover prevented the pilot and plane from getting covered with oil thrown out by the engine.

Engine
The DR.I's power plant was a 110-horsepower Oberursel UR II or Le Rhone nine-cylinder rotary engine. The cylinders were arranged in a ring and rotated with the propeller around a stationary crank shaft. The engine was started by a mechanic on the ground spinning the propeller.

Undercarriage
The two landing wheels were mounted on rigid struts. Unlike the wheels of most modern planes, they could not be withdrawn into the fuselage after take-off.

Wing cut-outs
To improve the pilot's view from the cockpit, curves were cut out of the upper and middle wings.

82

THREE-WINGED FIGHTER
Germany's Fokker DR.I triplane entered service in 1917, and for a while it proved unbeatable. The DR.I had three sets of wings, while most other fighters of the time were biplanes or monoplanes. Although its top speed was slower than other fighters', the DR.I could climb quicker and was unmatched in manoeuvrability – two qualities that gave it a crucial advantage in tight dog-fights.

Ailerons
For stability, the ailerons extended beyond the tips of the upper wing.

Wing struts
Wooden struts between the wings prevented the wings from vibrating and twisting during aerial manoeuvres. Even so, flaws in the wing structure caused several crashes. The DR.I was withdrawn from action to have its wings strengthened. By the time it re-entered service, other superior planes had been produced.

STARK CONTRAST
The Northrop Grumman B-2 (or 'Stealth') bomber shows just how much combat-plane design has changed since World War I. Its angular surfaces are designed to make enemy radar signals bounce off the planes at different angles, so that there is no strong echo returning to give the B-2's presence away. The B-2 is also coated with special material that absorbs many of the radar signals.

Fuselage
The body of the plane was made of a welded-steel skeleton strengthened by bracing wires and covered with fabric.

Rudder
The whole of the curved tail fin formed the rudder, pivoting left and right in the V-shaped notch between the two tail-planes.

Tail skid
There was no rear wheel on the Fokker DR.I, just a metal-tipped skid for the tail to rest on.

THE RED BARON
The German flyer Manfred von Richtofen was the most successful fighter pilot of World War I, shooting down a total of 80 aircraft. He was known as the Red Baron, because of the bright red Fokker DR.I triplane he flew. Von Richtofen's squadron, all in brightly coloured DR.Is to startle the enemy, ruled the skies and were nicknamed the 'Flying Circus' by Allied pilots. Von Richtofen was killed on 21 April 1918, when his triplane crashed to the ground after being attacked simultaneously by anti-aircraft guns and a Sopwith Camel fighter flown by Canadian pilot Roy Brown.

83

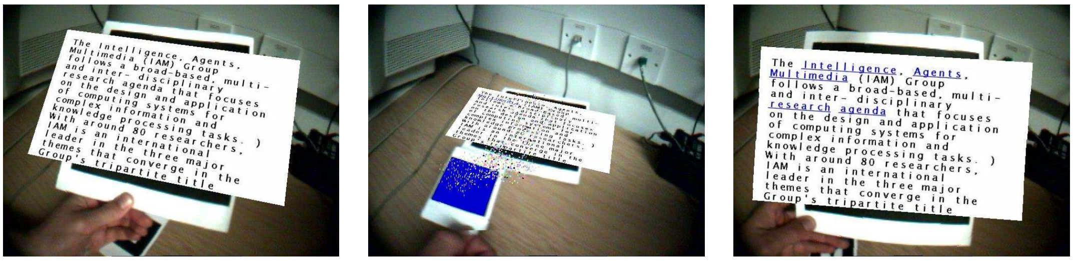
Figure 4.13: "How Things Work" (Parker et al., 2000)

For performing demonstrations and the evaluation, several aircraft models and an appropriate source of information for the feature labels was needed. It was important to keep the quality and nature of information consistent among all aircraft models used; to accomplish this a single source rich in aircraft feature information was sought. After considering various Internet sites and other sources, "How things work" (Parker et al., 2000) was chosen (see Figure 4.13). This book contains a selection of diagrams on various topics, such as transport, machinery and aviation. This book contains detailed information about a good selection of famous airplanes, and the material strongly focuses on aircrafts' individual features.

As the aircraft described in the book are well-known it was easy to acquire high quality 3D models from various free 3D content websites (3DCafe, 2003). The quality of meshes varied greatly, and this caused problems in obtaining a consistent set of aircraft to use in the prototype. Complex models can slow down frame rate in the ARToolKit drawing process, especially if many objects are visible at once. To overcome these problems, models from flight simulator games were used. These models are of good quality yet remain efficient for drawing. Virtually any aircraft can be found on various official and independent websites dedicated to these games, ensuring a consistent look and feel of all models. To convert these files to VRML and X3D a conversion process was necessary using various open source and shareware applications.

The models were treated using the technique described in Chapter 3. There was a desire to experiment with a complex source of granular information so the information from the book was manually compiled into a linkbase, where the textual descriptions were broken up into various levels of detail.

4.5 Salt and Pepper



The Intelligence, Agents, Multimedia (IAM) Group follows a broad-based, multi- and inter- disciplinary research agenda that focuses on the design and application of computing systems for complex information and knowledge processing tasks.) With around 80 researchers, IAM is an international leader in the three major themes that converge in the Group's tripartite title

Figure 4.14: Early "salt and pepper" demonstration

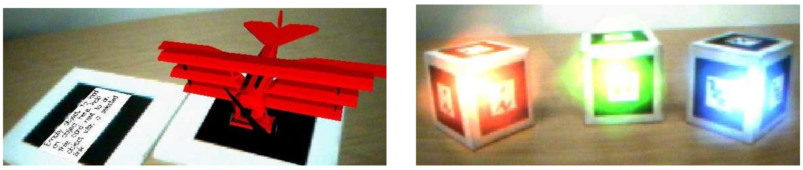
The origin of the "salt and pepper" metaphor was to allow users to shake a linkbase marker over an object or page of text so that links fly onto it. The principal motivation for the idea was in avoiding link overload, where too many links are applied to a document. The user can shake on the desired number of links and then remove them later by simply shaking them off. The first prototype, shown in Figure 4.14, involved sprinkling links from a linkbase marker onto a paragraph of text on an ARToolKit marker card.

I felt that this technique would be well suited for manipulating information applied to 3D objects using the technique described in Chapter 3. At this stage, the label placement implementation was still basic, with labels being generated as static X3D scenes by a servlet. Work on the "salt and pepper" interface advanced the development of methods for querying the link server and user interface techniques, such as selecting labels.

The "salt and pepper" interface functions as follows. When the user first picks up an object on a marker card, there are no labels attached to it. Instead of relying on a user profile to affect what labels are applied, users themselves are given control over what information they view about the object.

Open Hypermedia was a big inspiration for this interface. Linkbases can act as collections of links that share a common purpose (for example a linkbase of technical links), so by combining different linkbases users can tailor their view of a document. The "salt and pepper" interface allows people to physically manipulate the linkbases, containing label information, alongside the objects being labelled so that they can tailor the labels that they are shown.

In the ARToolKit, two types of markers are provided: object markers and 'spice pile' markers. The object markers are used to display objects while each spice pile represents a different linkbase of labels. When the user picks up a spice pile and shakes it, small



(a) Object markers

(b) Three spice markers

FIGURE 4.15: Types of marker cards in the system

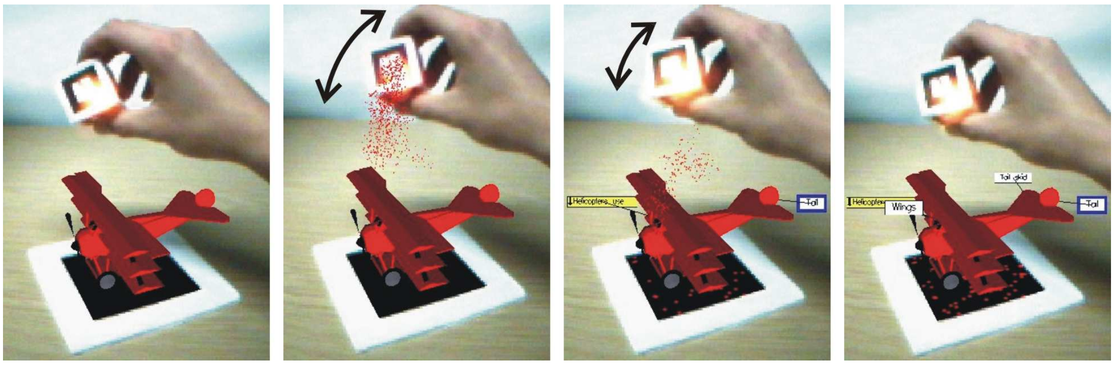


FIGURE 4.16: Sprinkling labels onto an object

particles drop from it and fly onto the visible objects. These particles represent the information labels that pop up on the object when the particles land. This process is illustrated in Figure 4.16.

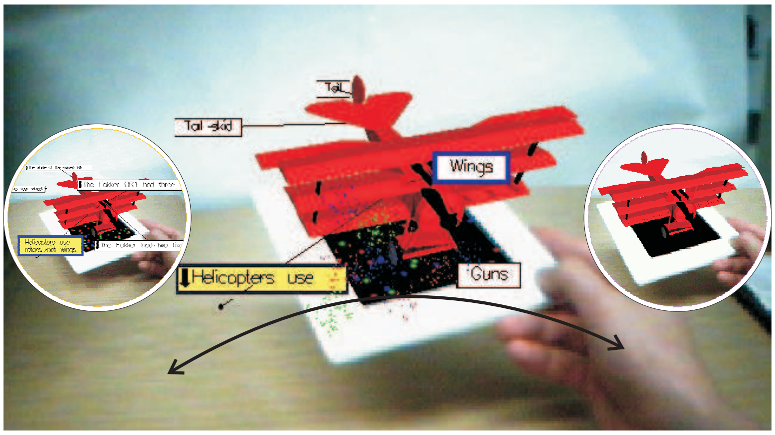
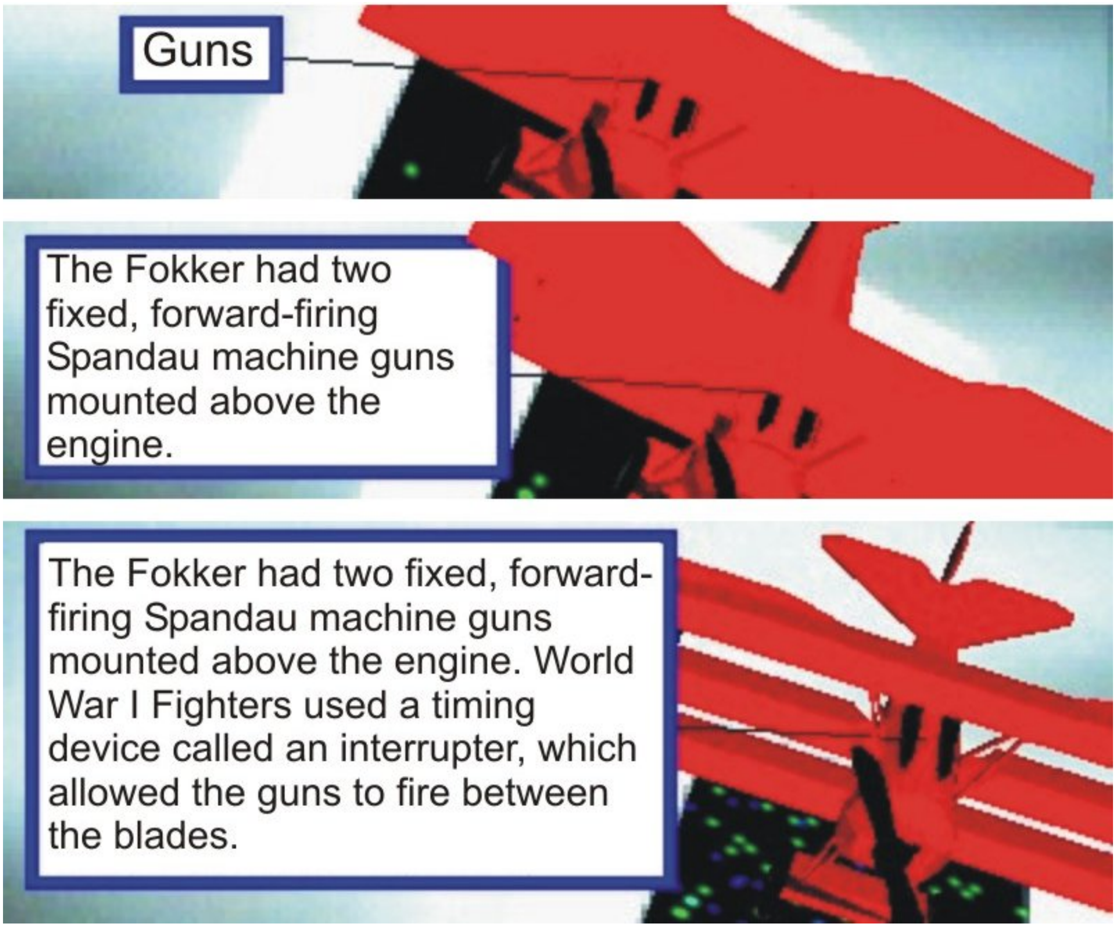


FIGURE 4.17: Shaking labels off an object

When users feel there are too many labels on the object they can pick up the object and shake it so that the labels fly off and disappear (Figure 4.17). The order that the labels fly off could be in the reverse order in which they were put on. Users can keep shaking an object until there are no labels left, leaving the user free to sprinkle a completely different set of labels on.

Different spice piles represent different areas of information, so that the type of label that is added to an object depends on which pile has been sprinkled. In the initial concept each spice pile would represent a linkbase whose content was closely tied to a subject area. With this simple linkbase approach, labels could only be added or removed, and a label's content would remain the same until it was removed.

4.5.1 Sprinkling Context



The Fokker had two fixed, forward-firing Spandau machine guns mounted above the engine.

The Fokker had two fixed, forward-firing Spandau machine guns mounted above the engine. World War I Fighters used a timing device called an interrupter, which allowed the guns to fire between the blades.

FIGURE 4.18: Evolution of a label as context is sprinkled on

What might be desirable to a user is information that evolves rather than just appearing. Labels could change and might even disappear as the user sprinkles. This required a shift in metaphor, away from sprinkling *labels*, towards sprinkling *context*. The content of an aircraft's labels reflects its current context, which can be manipulated using the context shakers, one for each context modelled in the information space: avionics, armament and trivia. As a user shakes one of the context shakers, context 'spice' particles fly from the shaker onto the aircraft. As the particles land, that context for the object is increased; as this occurs, new labels reflecting the increasing context level are applied and pop up on the object.

For example, Figure 4.18 shows a sequence where a user is sprinkling context from an 'armament context pile'. Initially, the aeroplane has a simple label on it that indicates the 'Guns' on the model. As the user sprinkles more context on, the objects context changes to reflect the fact that it should be shown more in the context of armaments. Consequently, the label evolves into a label stating the type of gun used on that specific aeroplane and then more detail is added. Labels can also have contextual restrictions that determine not only when they should be applied but also when they should be removed. For example, it may be appropriate to label a group of features, such as an airplane's landing gear, with one label initially. As the user requests more avionics context, this label might be replaced with several labels describing the various sub features such as the wheels and undercarriage.

Sprinkling allows users to manipulate the sophisticated relationships between the different types of contexts. For example, a particular piece of information might require a certain level of avionics context and a different level of trivia context before it would appear. Giving users the ability to mix and match the information they view about an object is very powerful, as they can discover for themselves the recipe of information that most appeals to them.

This shift to sprinkling context was the reason for the work discussed earlier on using different background colours for labels, depending on the type of information described in a label. This allowed users to easily relate between the context shakers and the labels added to objects.

The visualisation of hypertext linking described above, where the links are drawn as lines between source and destination anchors, is given an interesting effect in the "salt and pepper" interface. When several objects are visible, context can be added to the all of the objects at once. As links between the objects also evolve, users can appreciate how these relationships evolve as they manipulate the context space.

4.5.2 Evolution of Context Shakers

Throughout the development cycle of the "salt and pepper" interface, various changes were made to the context shakers. In informal testing sessions it was discovered that users often had trouble activating the sprinkling mechanism, which led to several efforts in improving different aspects of the interface.

The worst problems encountered with normal flat marker cards were that users often accidentally covered a portion of the ARToolKit pattern, and holding the card so that it remained visible whilst being shaken was uncomfortable. Cube shaped markers were used instead, as users can easily hold a cube leaving one of the sides visible. The same marker pattern is used on each of a cube's faces so that the cube can be held in any way and still be tracked.

To construct the cubes different options were investigated, such as building them out of polystyrene or wood or using children's building blocks or 3D jigsaw puzzles. The size of the cubes was an important consideration: they must be small to be comfortably held, but they must be large enough so that they can be accurately tracked.

The chosen solution was to use toy cubes with puzzles inside them (the puzzles involve passing ball bearings through holes). These cubes have a very appealing attribute in that the ball bearings inside rattle when shaken, resulting in very effective aural feedback. The only drawback is that they are small, so under some circumstances they can be difficult to track with the ARToolKit.

With the marker patterns on all faces of the cube, up to three sides of the cube may be visible at a time. Very often this causes false triggering as the ARToolKit switches between the visible markers, giving the impression that the marker is being moved from side to side, confusing the shaking recognition system. Different markers could be used on the each side of a cube, but this would result in the ARToolKit having to distinguish too many markers. Besides slowing down the tracking process, this can cause more false positive identifications (i.e. the ARToolKit overlays the wrong object on a marker). A possible solution is to use two identical marker patterns on the opposite sides of each cube, so only one of the markers is visible at a time. Unfortunately, users need to more aware of which side the cube has to held so that a pattern is always facing the camera.

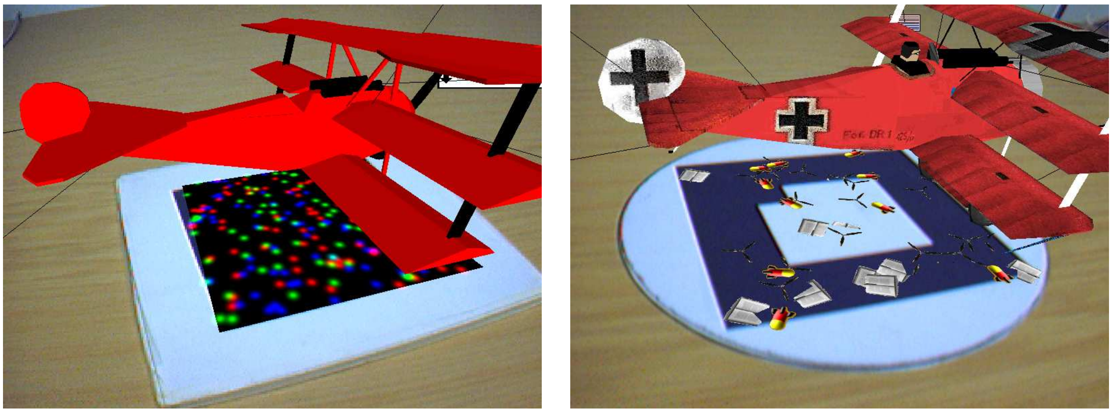
4.5.3 Removing Particular Contexts

One problem with the "salt and pepper" interface is that it provides little control when removing context from a mixture of information. For instance, two types of contexts could be sprinkled onto an object, such as avionics and armament. User may then wish to view the only the armament information about that object, so they must shake both types of information off the object and then sprinkle armament context back until the previous information level is reached.

The addition of a "hoover' effect to the context shaker objects could provide greater control in refining mixtures of information. By bringing a context marker close to an object, it could attract the particles on the object belonging to that context and "suck" them off the object base. However, it was considered that adding this functionality would stray too far from the "salt and pepper" metaphor, so this feature was not implemented.

4.5.4 Visually Marking Context

While context sprinkling gives users an excellent appreciation of how they are moving through the context space, it does not show them their current state (i.e. the amount of context applied to an object). This proved to be a complex problem, and various approaches to providing visible feedback of an object's context levels were investigated.



(a) Coloured particles

(b) Image particles

FIGURE 4.19: Particles on the marker base

The first method allocated a different colour to each of the spice piles and associated particles. It was thought that mixing different contexts could alter the colour of an object's base to reflect the quantities of the different contexts. Interpreting the quantity of context applied to an object by looking at the colour mixture was considered too confusing. Instead, when particles landed on an object they were drawn on the object's base so that users knew how much of that context had been sprinkled on. This is shown in Figure 4.19 (a).

Initially, the relationship between the shakers and particles and the actual information was abstract. Shakers were represented by a coloured pile of dust, and each unit of context was drawn as a swarm of coloured squares. When cube shakers were introduced the pile model was changed to an animated, shimmering cloud. It was decided that there should be a more explicit relationship between the information subject area and the shaker objects. For instance, a propeller could be used to model avionics information, bombs could represent armament information and books could represent trivia. This idea has been extended to the particles themselves, so propellers and books are drawn flying to the objects and on the object bases, providing a more straightforward identification of the context mixture. This is illustrated in Figure 4.19 (b).

These approaches to indicating the amount of context on an object overlooked a crucial aspect. Users need an idea of an object's context levels, that is how much context has been sprinkled on and also how much more can be added before the maximum level of detail is reached. This second point is extremely important, otherwise users might keep sprinkling when there is no more information to add. Certain aircraft might not be affected by one context at all, for instance, gliders usually have no related armament information so the user should realise it is pointless to sprinkle any armament context on.

An approach to this problem might be to add some kind of level indicator on each context shaker indicating the level of context of that shaker. For example, if the shaker is half full, the user can see that he has sprinkled half of the available context. The problem is that context shakers are used for multiple objects, each with their own context levels. A context shaker would need to change to reflect the levels of whichever object it is being shaken over, and would need to indicate which object it is acting on if several are visible at once.

The best solution for this problem is to keep the context state indicators on the objects themselves, as was explored within the early work with the base particles described above. The challenge is to find ways to effectively transmit the full picture of an object's context level to the user. There was also a strong desire to keep within the shaker and particle metaphor.

One last attempt was made to resolve this problem without resorting to a more generic indicator such as a dial. It involved placing the number of particles for the maximum level of available information for that object on the base from the start. However, these particles are faded out or semi-transparent. As context particles are sprinkled onto the object, the particles fill in. The user can then tell how much information has been sprinkled on by looking at the number of filled in particles, and can tell how much more can be sprinkled by looking at the amount of faded out particles. When all particles have been filled in, any particles landing on the object bounce off.

Unfortunately when this was implemented, it was discovered that the base would become too cluttered, making it too confusing and inefficient to interpret. Other ideas were considered, but none of them were satisfactory.

4.6 Prototyping Interfaces

Following the experience with the "salt and pepper" interface, I began investigating alternative interfaces for manipulating information displayed about an object that could address the issue of indicating the current context state.

The design of these interfaces was constrained by the limitations of the ARToolKit. Although various problems in the ARToolKit could be overcome by using mixed tracking or embedding hardware into the markers, this was avoided to reduce the complexity of the system.

Different setups for the interfaces were considered. There are various ways to construct ARToolKit applications besides using see-through HMDs. The augmented video stream can be projected onto screens using a fixed camera aimed at a particular area such as a table top. A mirror-like approach is often used in installations where the camera is placed in front of the user, where the display is horizontally mirrored for more natural interaction. These types of installations may suit environments such as museums due to the high cost of HMDs. While less suitable for personalised displays, it does allow for straightforward collaboration between users, as many users are able to view the augmented scenes without the need for an HMD for each individual.

In the following few sections I describe several other interface designs and provide some brief analysis. During this stage of experimentation there was a desire to examine and create natural interfaces that are fun to use, and several real world metaphors were tried. There was also a wish to experiment with different approaches to ARToolKit interfaces to manipulate information, such as multimarker card setups. Another example is an attempt to restrict the number of markers required by using markers' positions in relation to the display to trigger interface events. This might be useful in mobile AR systems, where it can be impractical for users to manipulate many markers in front of them.

4.6.1 Dipping

Dipping was an idea that was considered soon after the "salt and pepper" interface. In order to add labels to an object, users could pick up the object and dip it into a vat of information, resembling pots of paint. When it comes out, the object would be covered with the labels relevant to the vat it was dipped in. Repeated dips in the same vat would result in increasingly complex labels.

This 'sheep dip' metaphor could be used with multiple vats, one for each type of information, and could be used to explore various effects. For instance, dipping in a second vat might replace the labels on the model with those relevant to the second vat. This is akin to dipping a real model in different coloured paint pots, the new colour replacing the old. The second vat could also add new labels to the labels already on the model in a cumulative fashion; the idea of colour combining could be used to indicate the level of the mixture. For example, dipping in 'blue' and 'yellow' vats would result in 'green' labels. To remove links, a vat of solvent could be made available in which to dip objects.

There were a few problems with this metaphor. As with "salt and pepper", the resulting colour mixtures from dipping in different pots could indicate how much context had been applied but would not effectively reveal how much more context there is to add, especially for objects with no information about a certain topic. Another problem is that by dipping objects into vats of paint, users might expect that the object should be painted as well.

4.6.2 Uncovering

A number of interfaces were considered that took a reverse approach to "salt and pepper". Rather than adding information to object, information can start hidden and slowly be revealed. For instance, imagine lifting a sheet to uncover information about objects.

One idea involved the use of fans. An object would start with labels covered in clouds that could be blown away by holding a marker containing an animated electric fan. As the clouds disperse, the text in the label would become visible. The level of detail in the text could be set by changing the fan speed, depend on the time a fan is held up or by the distance between the labels and the fan.

This idea, although simple, is a powerful way to show objects' current context levels. However, there was an issue of clutter: the labels and clouds would remain visible and obscure the object. The level of detail needs to be reduced; some mechanism to return the clouds to labels is necessary. Mixing different types of information with the fans was the largest obstacle encountered: this might require different coloured clouds and fans for each subject area, where each fan only acts on its own type of cloud.

A similar idea was considered that involved unwrapping bandages from labels, but this shared the same problems of information mixing as the fan metaphor. Another challenge for these interfaces is coming up with efficient physical gestures for performing the uncovering of information.

4.6.3 Bees

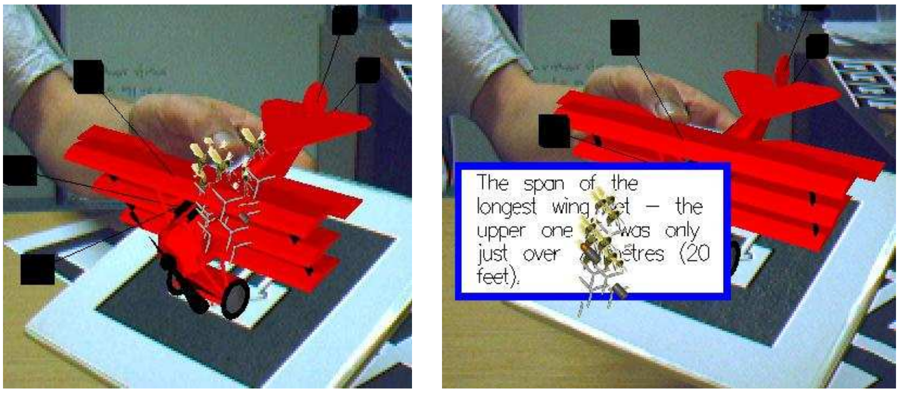


Figure 4.20: Bees swarming around an object to perform label selection

The concept behind the bees interface was to dynamically build labels as the user examines an object, with some type of workers appearing to construct the information. The level of detail presented in a label would depend on the number or size of the workers, and different types of workers could be used for presenting the information from different subject areas.

This led to the idea of having swarms of bees fly around the object building the current label being examined, as if they were building honeycomb structures in a hive. Early experiments with swarming bees, shown in Figure 4.20, were very promising as it was fun and satisfying to see the bees flying around the object and they gave a very clear indication of the currently selected label.

To manipulate the labels' level of detail, bees need to be added or removed from the object. This could be achieved by placing a bee hive for each type of available information on a large multimarker card on a table top. Bees could then be added to an object by placing the object marker on one side of a hive, and removed by placing the object on the other side of the hive. Alternatively, the bee hives could be implemented using the position of an object in relation to the screen; the left side of the screen could add bees and the right hand side could remove bees.

There were several plans for indicating the current context state for this interface. Bunches of flowers could be placed at an object's base, one for each type of context. The number of flowers per bunch depends on the available context for that object, so an airplane rich in avionics context would have many avionics flowers. As bees are added to an object, some sit on the flowers whilst the rest swarm around the object; the number of occupied flowers would indicate the amount of applied context, and the number of empty flowers would indicate how much more information can be added.

A problem with this is that different types of bees are needed for each subject area, and users must be able to distinguish between these. To clearly present the numbers of different types of bees, different swarms for each type was considered. This led to another idea to present context levels: the number of bees per swarm would depend on the maximum level of detail for that subject. Bees could pick up pieces of context, so now the number of bees in a swarm holding a piece of context would reflect the level of detail.

Although the mixing of different types of information was well handled in the bees interface, it was felt that it was over complicated and it was abandoned. The ideas for showing context state, although simple in origin, would become too confusing in a context rich environment. Drawing labels as honeycombed structures would have been tricky to implement, and may have raised new problems. The whole premise of bees flying around objects felt intuitively strange. More abstract variations of this idea were considered, but had the same fundamental problems.

4.6.4 Menus

With the ARToolKit, the position of markers in relation to a monitor screen or users' view through the HMD can be used to trigger user input events. For instance, holding a marker on the side of the screen could bring up a menu, where items could be selected by timing how long the marker is held at a certain position. There are many possibilities for displaying and manipulating context state with such systems, so these were explored in various prototypes and experiments.

Interfaces constructed around this approach will not be as tangible as some of the other systems described in this thesis, and are more similar to traditional mouse based menu systems. This was part of the experimentation into different types of interactions with the ARToolKit markers. There are some possible benefits for using this kind of interface, and may provide functionality that tangible interfaces are not able to cater for. For example, choosing from a list of options that are hard to model in a natural, tangible way. Another example where these interfaces can be used is in mobile AR applications where users don't have a surface or tabletop in front of them to manipulate the markers on.

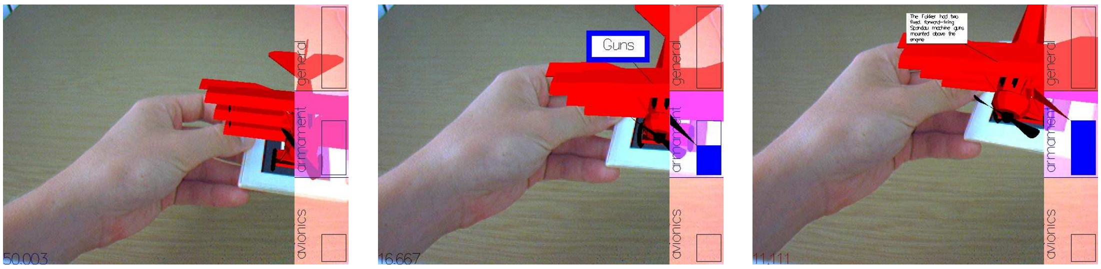

general
armament
avionics
50.003
Guns
16.667
The Fokker had two fixed, forward-firing Spandau machine guns mounted above the engine
11.111

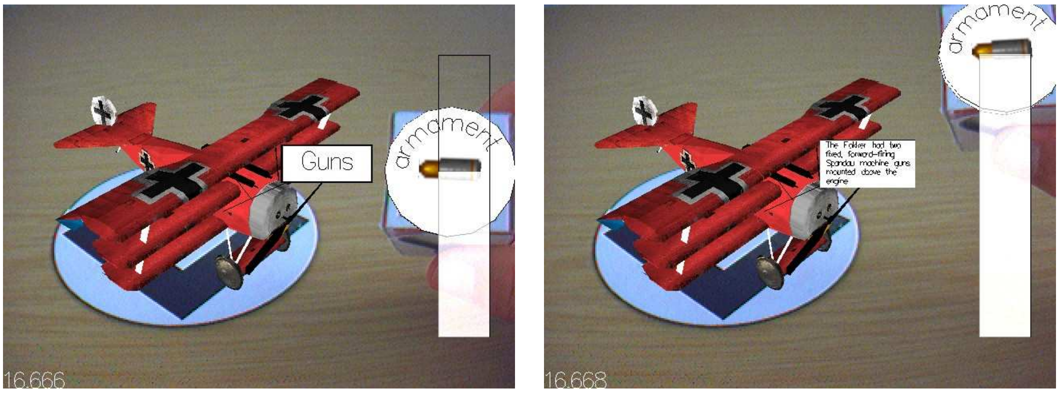
FIGURE 4.21: Adding context using a menu

Different menu based interfaces using this technique to manipulate the context on an object were prototyped. In the prototype shown in Figure 4.21 the user holds up an object marker near the edge of the screen to bring out the menu. The menu is divided into three sections, one for each type of information. By holding the marker next to one of the slots, it will activate the slider bar for that type of information; moving the object up and down changes the amount of context for that object.

Such slider bar systems can clearly present the amount of available state, as well as allowing users to manipulate it. With a slider, the maximum possible amount of information is shown (in terms of the length of the slider) and the slider level clearly shows objects' current context state.

A simpler system was also considered. Both sides of the screen would be used as menus, one side for adding and the other for removing context; each of the menus is split into the different types of information as above. Users manipulate context by holding the marker next to the desired menu item. This would require the use of a dial to indicate how much context had been applied to the object.

Another interface was implemented where, as with the "salt and pepper" system, different context markers are used for the each subject area. When one of these markers is held over an area of the screen, the slider bar is active and affects that type of context



The Fokker had two fixed, forward-firing Spandau machine guns mounted above the engine

FIGURE 4.22: Second approach at a menu interface

level for any visible objects. This is shown in Figure 4.22. This approach overcomes problems encountered with the earlier slider bar, which could be tricky to use as the ARToolKit often loses the marker off the side of the screen or users' view. It is also more straightforward to use, as the selection of the context is made by picking up the desired context marker instead of holding the marker by the menu item for a certain amount of time.

Currently, no indicators for the level of context have been implemented (besides when the sliders are active). Dial indicators could be placed on the objects to indicate the level of context when the slider bars are inactive. Different dials would be needed for each of the contexts. These indicators could be highlighted when the slider bars are active and minimised when the user simply wants to look at the object. These dials will be important to indicate cases in which there is little or no available information for an object.

A large problem with this interface is possible false triggering by users accidentally placing markers at the side of the screen, even despite the efforts taken to avoid this in the second prototype. Another issue is that for non-HMDs setups the camera has to be carefully placed so that users can comfortably hold the markers anywhere on the screen, especially on the corners.

Although these menu based approaches have been left at prototype stage, many of these ideas could quickly be brought forward. A menu based system can have many other uses besides manipulating context levels, and should be considered for providing new forms of functionality to ARToolKit interfaces.

4.6.5 Airfield

The aim of this interface was to explore the spatial layout of objects to manipulate context. This was achieved by calculating the position of the object markers on a flat surface, such as a table. The distance between the object and certain areas of the table would determine the level of context for that object.

The chosen subject area involves information about aircraft models, which led to an airfield metaphor being designed where visual representations of the different context themes acted as the context controls. For example, the avionics, armament and trivia contexts were represented as a hanger, ammunitions dump and control tower respectively. The airfield was overlaid on a large multimarker card, and the possibility of mixing real objects and augmented imagery was considered. For instance, physical models of the control tower or hanger could be placed on the virtual airfield.

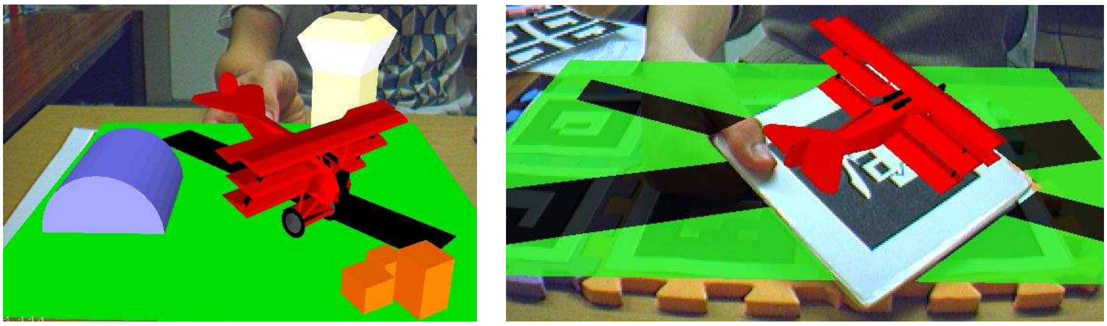


Figure 4.23: Initial airfield metaphor

Work on a prototype for this interface made sole use of augmented images, as shown above in Figure 4.23. With the ARToolKit multimarkers, the overlaid imagery obscures the real scene completely: even users' hands and markers are covered, making it tricky to move the object around the airfield (Figure 4.23(left)). This was overcome by using transparency effects on the airfield and drawing a 2D plane on the object marker so that the underlying video stream is visible, making the marker itself visible (Figure 4.23(right)). Different types of multimarkers were used and it was discovered that twelve small markers provide more robust tracking than the default six marker sheet.

The concept for the interface was as follows. As an object marker is brought closer to a context object, that context level is raised; conversely moving it away reduces that context level. The context alters only when the object is on ground, so that users can pick objects up and look at the information closely without altering the context.

The airfield was laid out with each context object spaced an equal distance apart. This is impractical for mixing contexts: not all combinations of the different contexts are possible by moving an object around on a 2D plane between fixed context objects. For example, if a lot of the avionics context is desired, the airplane being looked at should be placed next to the hanger - but what if both the avionics and armament context are needed? The airplane can't be placed next to both context objects at once.

An alternative approach was then investigated where instead of moving the objects

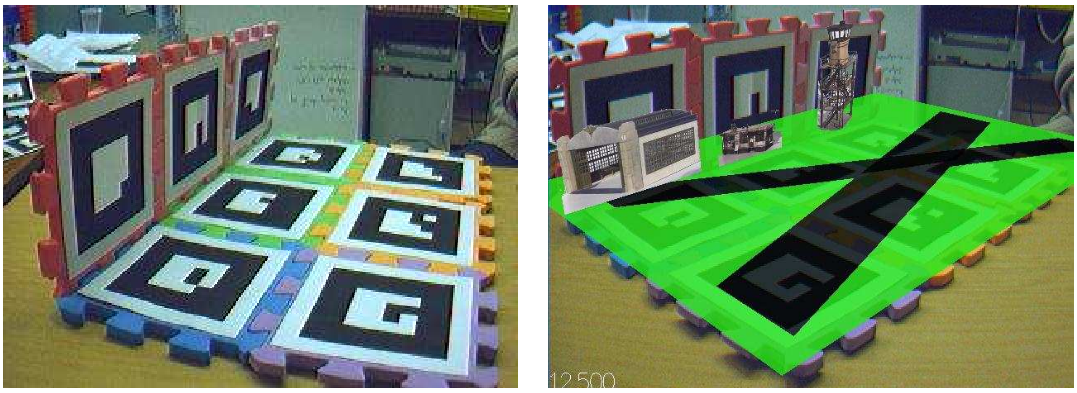


Figure 4.24: Improved airfield metaphor

around the airfield, contact with one of the context dispensers changes context. For instance, touching the hanger would increase the avionics context. In this case it is desirable to provide strong physical feedback, allowing users to feel the objects touching. This led to a new approach to the airfield layout, shown in Figure 4.24. The multimarker is extended vertically on one of the sides, and the context dispensers are placed along this ”wall”. Users can now hit objects against a physical wall. This also improves the robustness of tracking as at least one of the vertical markers usually remains visible when the horizontal markers are obscured.

This setup can create problems with non-HMD setups where the camera is facing the user. The vertical wall needs to be visible, yet cannot be placed directly in front of the camera as this would get in the user's way. During the implementation and debugging phase, the airfield worked well with the wall placed on the side of the scene, but this would need to be investigated further for installation in a real or evaluation environment.

To remove context, several techniques were considered. There could be two sides to each object marker, thus the addition or removal of context would depend on which side of the object had hit the context dispenser. Another option was turning the aircraft upside down when hitting against the context dispenser to decrease that type of context. Also, another gesture could be used where the aircraft is tapped against the airfield base to remove all forms of context, like shaking in “salt and pepper”.

As users hit an object against a context dispenser to raise and lower context, a new set of labels is applied; this often has the effect of several labels appearing at once. For more natural interaction a new way of applying labels was tried. As an aircraft is hit against a context dispenser, the labels are applied one at a time; the context is only increased when there are no more labels to be added for that context level. When the level of detail increases, new content is added to the old label. The process is animated: users see information “blobs” fly from the dispensers onto objects. To implement this, a new approach to querying was required that involved caching labels until they could be applied.

This approach to querying highlighted a problem with the mixture of different types of information. In the material authored for the prototype, there are many labels that contain different types of information; for instance, there may be a description detailing the armaments of an aircraft that is complemented with a piece of trivia. To fully expose this label, users need to add the armament context and then add some trivia context. The problem encountered is what happens if the user starts adding trivia before adding avionics context? If the linkbase contains no sole trivia labels for that object, there would be no apparent effect of hitting the trivia dispenser as no labels are applied. However, when the user then adds some armament context, the mixed label would appear complete with the trivia information, as the trivia context level would have already been set by the previous actions. This might have the effect of confusing the user.

This issue of mixing made the raised the importance of clearly indicating an object's current context levels. Several approaches were considered to accomplish this yet none were deemed appropriate. At this time, it seemed unlikely that an interface overcoming these problems could be designed. If this proved to be the case the mixing mechanism would have to be reconsidered.

4.7 Waves

The experience gained from the unsuccessful experiments described proved to be very important as it inspired the design of a new interface, "waves"; the work on distances on the airfield was in particular influential. As it became clear that rich mixtures of information couldn't be obtained with the context dispensers fixed in place on the airfield surface a new approach was considered. This involved being able to freely move the context dispensers around a surface, such as a table; mixing information becomes a matter of placing the desired context dispensers next to the object markers. However, at the time there was a desire to construct an interface around the ARToolKit multimarker setup, so this idea was overlooked. With the problems encountered on the airfield "one-by-one" interface, the idea to move the context dispensers was reconsidered.

In the "waves" interface, the distance between objects and context dispensers modifies the context on the objects. Moving an object alongside a context dispenser sets that context to full on the object so all information is applied; when the context dispenser is moved away from the object then the context level decreases. The use of distance required a new design for the ARToolKit marker cards: to keep distances the same all the way around the marker circular disks are used. Unlike with the earlier interfaces, all object and context markers are of the same shape and size. This is shown in Figure 4.25.

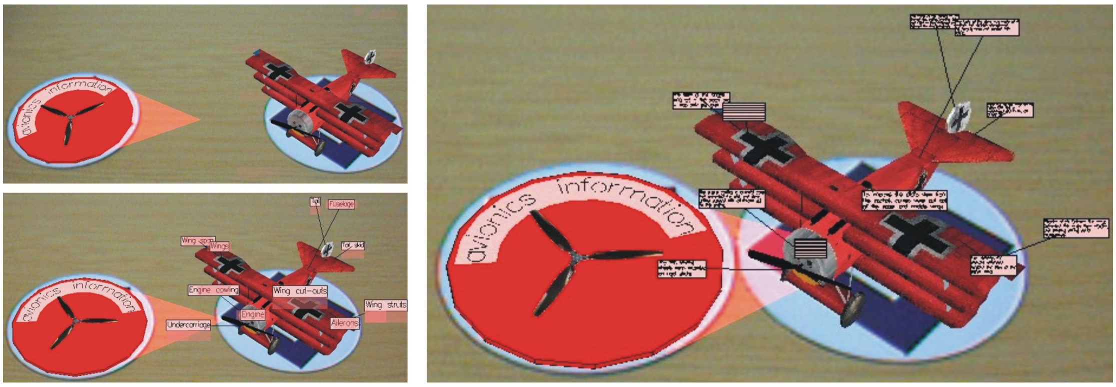


FIGURE 4.25: Waves: distance affects the information applied

The problem of indicating how much of each context is available for an object remains. To accomplish this, “waves” are drawn from each visible context dispenser to each visible object. These waves also show the range of the context dispensers, so that users can see the area in which context dispensers are active on an object.

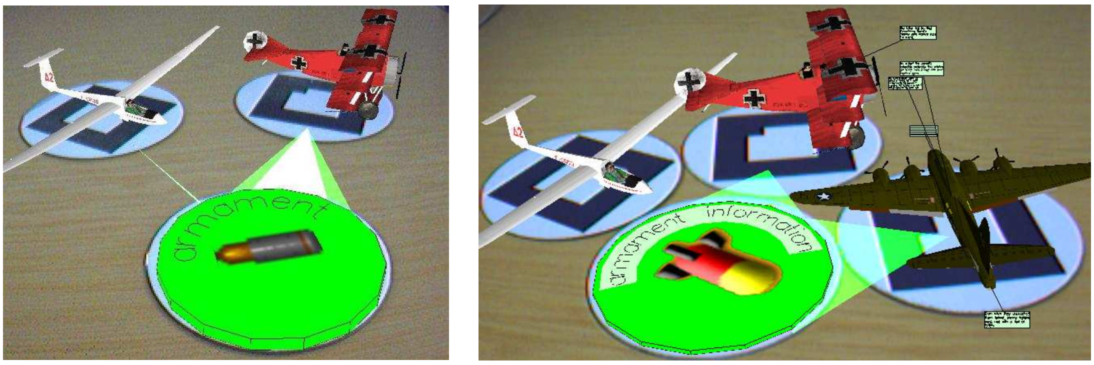


FIGURE 4.26: Wave width (left) versus wave length (right)

Each object might have different amounts of available information on the various subjects, and this can be represented in the waves. Initially, all waves were drawn with the same length but different widths depending on the amount of available context. The amount of information applied to an object would depend on the thickness of the wave where it intersected the object marker. For example, a military aircraft would have a thick wave for armament, while a civil aircraft would have a very thin line to indicate that no information was available. This is illustrated in Figure 4.26 (left).

It was decided that the use of thickness for indicating quantities of information was not ideal. It was hard for users to distinguish the amounts represented in “waves”; also, the thin lines used for no available context were hard to see and were somewhat confusing to the user. A different approach that took more advantage of the distance between objects was preferred, where the length, rather than the width, of waves indicate the amount of available information for an object. Long waves are used for objects with a lot of information, and small waves are drawn for objects with less information. Where there is no information available for an object, no wave is drawn. This process can be seen in Figure 4.26 (right).

The "waves" interface introduced a new aspect in that users need be able to activate and deactivate the context marker cards when they wish to focus on an object's labels. As the information displayed about an object changes depending on the distance from the context marker cards, the labels' contents would keep changing as users move the object being examined if context dispensers are present in the scene. The initial plan was to limit context manipulation to when the object markers are on the same plane as the context markers, for example on a table top. ARToolKit tracking errors made this approach too unreliable; for example, an object's context would remain affected against the user's wish when they were examining an object.

This led to an alternative method for triggering context activation which uses a physical property of the ARToolKit markers: visibility. Users can hide context markers when they don't wish to use them by either turning them upside down or placing them outside the camera's view. This increased the ways in which context markers are used to affect context: rather than just moved around a table-top they can be manipulated while being held, which may be slightly more practical under certain circumstances.

An advantage of "waves" is that it is no longer essential to visually indicate objects' current context levels. The distance between the objects and context markers, together with the displayed wave, provides all of the cues required when manipulating context: how much information has been applied and how much more can be added. However, it might be useful to provide an indicator when the user is viewing the object and no waves are visible to remind him of the information that has been applied. This was not implemented in the current prototype as it was considered that the distance between the objects was a sufficient indication of the amount of context.

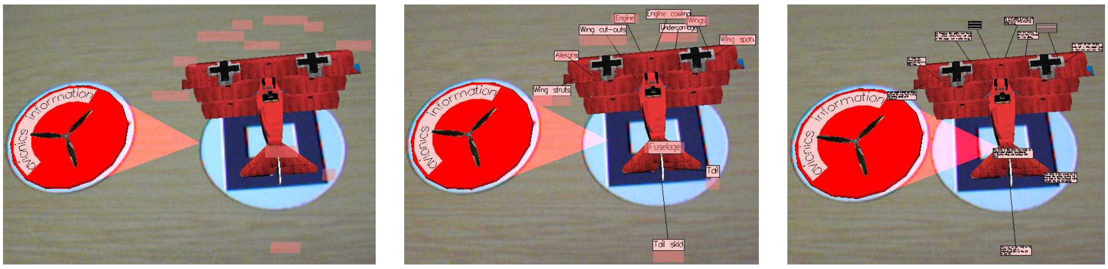


Figure 4.27: Transparent labels

With the "waves" interface, there has been an attempt to avoid confusing the user as they mix different types of information, as described in Section 4.6.5. The appearance of mixed information labels is dependent on more than one context conditions, and it can be hard to transmit to the user what these conditions are. To overcome this, semi-transparent labels are placed around an object when a context dispenser is visible to indicate the available labels for that context. As the context dispenser is brought towards the object, the transparent labels become brighter, and when the wave reaches the object they become opaque and the label text is visible. This is illustrated in Figure 4.27. If there is more information available for that context, in the form of both new labels or new material for existing labels, it is again represented as transparent labels over the object. This can be seen on certain labels, such as the one on the tail, in the middle image of Figure 4.27.

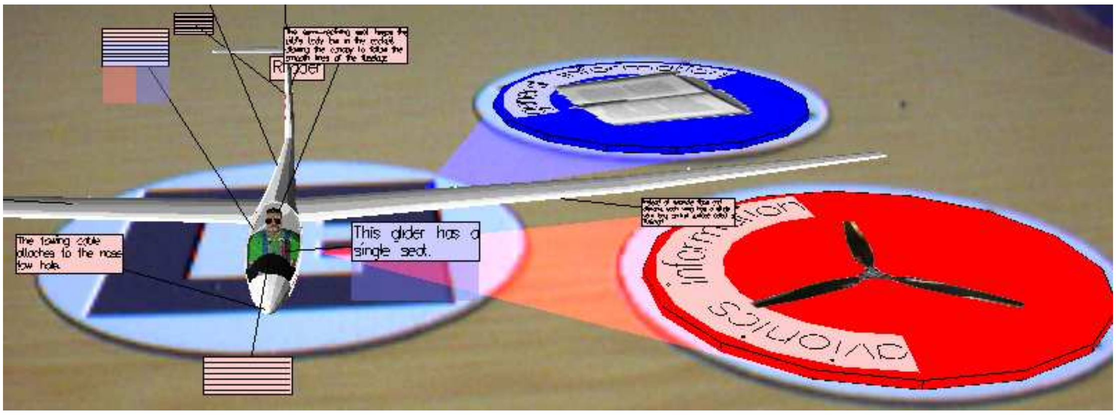

Rudder
The towing cable attaches to the nose tow hole.
This glider has a single seat.
information
avionics

FIGURE 4.28: Mixing information with "waves"

Another example is shown in Figure 4.28. Here you can see two waves active on an object, avionics and trivia. There are two transparent labels, one for the glider's landing wheel and another pointing to the cockpit of the glider. Note that the cockpit label has two transparent labels, one red and one blue, indicating that the user can add bring both the avionics and trivia waves closer to the object to view more information about this topic.

The use of transparent labels provides an indication of what users can expect when bringing different context dispensers towards an object, which helps when mixing different types of information.

4.8 Discussion

The period of experimentation into tangible interfaces resulted in two systems that are ready for user evaluation: "waves" and "salt and pepper". Certain designs described in Section 4.6 were left at an advanced prototype stage, such as some of the menu based systems. These could be advanced and perhaps completed with little effort. Other experiments remained as concepts although some work on rough prototypes was carried out, such as the bees interface and the work on the airfield systems.

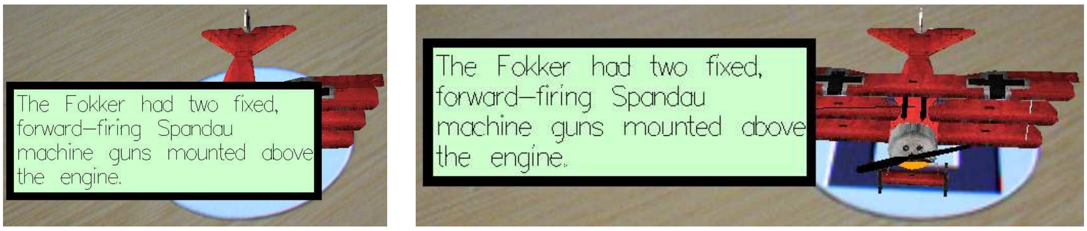
In Section 4.4 a list of issues considered during the design and implementation was outlined, which included accessibility, nature of use, context mixture, context state and appropriateness of the ARToolKit technology. I consider that all of the interfaces I developed were easy to use and accessible. Table 4.1 explores how each interface relates to the remaining issues that were considered.

Interface	*Natural*	*Mixtures*	*State*	*Technology*
Salt and Pepper	Very natural metaphor	Strong at handling mixtures	Hard to convey state, especially potential state	Gesture tracking hard to get right
Dipping	Natural metaphor, but will users expect objects to become painted?	Strong mixtures, as with salt and pepper	Similar to salt and pepper, hard to convey potential state	No obvious issues
Uncover	Natural metaphor	Weak at mixtures	Strong way to present potential state	Activation hard, i.e. turning on fans. Cluttered display
Bees	Natural to use, but feels strange	Weak, hard to show different types of information	Weak, would be confusing	No obvious issues with ARToolKit
Menus	Not as natural as other interfaces	Reasonable at mixtures	Strong, use of slider bar	Can be hard to control markers at edge of screen
Airfield	Natural, especially considering subject matter	Problem with advanced mixtures	Would depend on the approach, positioning strong but hitting against objects tricky	Multiple markers confusing, optimal layout of markers is necessary for hitting against objects
Waves	Natural, once users have hang of it	Strong	Strong	Very well suited, as long table top is used

Table 4.1: Discussion

4.8.1 Refining Labelling

During this work, problems with the labelling techniques became apparent. The advantage in using label positions' in relation to the center of the screen for selection is that the system feels natural and intuitive to use. As the labels remain in front of the object the users' view of the objects is obscured, which can be frustrating when trying to view the feature being described by a label. This situation is most apparent when large labels



The Fokker had two fixed, forward–firing Spandau machine guns mounted above the engine.

The Fokker had two fixed, forward–firing Spandau machine guns mounted above the engine.

Figure 4.29: Fixed labelling (left) versus moving labelling (right)

are used to describe object features. An example is shown in Figure 4.29 (left).

To avoid this, different ways to move labels from in front of the object were investigated. It was determined that only selected labels need to be moved, as when labels are unselected they are small and so do not hamper the user's view .Several approaches to movement were tried. The initial procedure was to move labels beyond the outline of the object, and this involved investigating different methods for calculating an object's outline so that labels do not obscure the object and at the same time leave the labels close to their respective feature.

It was important that the distance moved by a label was as short as possible for various reasons. Moving a label could be considered distracting, and users may become confused by labels being very far from the feature they describe. Importantly, the selection process relies on users manipulating the objects so that the label they are interested in is the closest to them. Moving the currently selected label might cause confusion as it will no longer appear to be the closest label on the object. For this reason especially it was important to minimise the distance that selected labels moved.

It was decided that moving labels beyond the outline of an object was not ideal, as the distance was usually fairly large. Instead, the outline of the currently selected label's feature was used. Initially the outline of the whole feature was used, but certain features would cause strange results, such as wings which are straight and narrow. To avoid this, it was decided that labels should only move a fixed distance from the center of the feature; this distance was fixed in screen coordinates so it would remain the same no matter how far the object is away from the camera. This is illustrated in Figure 4.29 (right).

4.8.2 Activating Selection

When many labelled objects are visible the display can become cluttered and confusing, especially when the system is highlighting the selected label on each object. While it would be interesting to investigate different ways in which to display information when there are many visible objects, it was considered beyond the scope of this thesis. There has been much existing background research into the area of contextual displays, and there even may be opportunities to carry out novel work in applying such techniques to AR environments. However, there was not enough time to investigate this thoroughly and a quick solution was needed so that the evaluation on the interfaces could be carried out.

The “waves” interface approach to activating the context dispensers, where users hide the context markers in order to focus on and select an object’s labels, inspired the chosen solution. The idea of hiding the context markers was applied to all other objects: an object’s labels are only selectable when the object is the only visible object in the scene. Although this approach did not remove the problem of labels cluttering the display when many objects are visible, preventing labels from being selected improved the situation sufficiently so that the system could be evaluated.

However, this change forces users to constantly hide and reveal markers. One of the most effective ways to perform this action is to flip unwanted markers over so that the marker side of the card is face down. This results in users not being able to identify which objects are associated with each markers when these are face down. To overcome this, an image of each object is placed on the reverse side of the marker card, indicating the identity of the object associated with each marker.

4.9 Chapter Summary

This chapter has described how certain properties of ARToolKit markers could be used in tangible interaction techniques, which led to the creation of various simple prototypes. The experience gained constructing these led to the design of an interaction mechanism that allows users to select and highlight labels on objects presented in an AR environment. Hyperlinks between different features on objects were also investigated, and these are displayed by drawing an annotated line between links’ source and destination anchors. This work was based on the labelling system described in Chapter 3.

The main research interest behind this thesis is to investigate tangible interaction techniques for manipulating the information that is presented about objects in AR environments. This led to the design of a variety of tangible interfaces, of which two seem to warrant further evaluation. “Salt and pepper” allows users to construct recipes of information by sprinkling different types of context onto objects. “Waves” uses the position of context dispensers in relation to objects to affect the information displayed about an object. During the design and implementation of these interfaces it was discovered that the label selection technique required some refinements, such as moving labels so they do not obscure the object.

The next chapter describes and presents the results of a preliminary formative evaluation of the labelling selection mechanisms and these two approaches for manipulating information presented about objects.

Chapter 5

Evaluation

User evaluation is a critical part of any system involving user interaction. It is the process that aims to identify usability problems in user interface design, and hence, end use (Mack and Nielsen, 1994). Any problems identified can then be used to recommend improvements for the interface design. Gabbard et al wrote that "High usability is not something that happens by accident or by good luck; through specific usability methods, it is engineered into a product from the beginning and throughout the development lifecycle" (Gabbard et al., 1999). Common usability problems include missing functionality, poor user performance on a critical or common task, catastrophic user errors, low user satisfaction and low user adoption of a new system.

In this chapter the design and execution of a formative evaluation of the interfaces presented in Chapter 4 is described. This evaluation has been used to obtain users' reactions to the tangible interaction techniques to view information about objects in AR. Several usability problems with the prototype systems were identified; these are discussed and possible solutions are suggested.

5.1 Evaluation of Tangible AR Interfaces

Informal evaluations were a constant part of the development process. Whenever a new feature was implemented, it would be informally tested during development. Very often other people would get a chance to use the interfaces, either informally within the lab or during demonstration sessions. These events were the source of much useful feedback, from both the users' comments and through general observation of users with the systems.

An example of this kind of informal evaluation occurred with the early work on creating buttons by covering ARToolKit markers. By simply allowing several people to play with the "button", it was obvious that, due to nature of the tracking system, the system would not be able to differentiate when the user was pressing the button and when the marker was being concealed by poor lighting, fast movement or even accidental occlusion. In another instance, a demonstration session prompted the use of marker cubes instead of flat cards for the shakers in the "salt and pepper" interface. Users were having trouble handling the markers in a way that they could be shaken and tracked by the system at the same time; noticing this led to the use of cubes as shakers instead, which have been better accepted by users.

However, a formal evaluation is crucial when developing user interface metaphors, so that some idea of what impact the metaphors might have if they are extensively used. Various evaluation techniques were considered. Heuristic evaluation is performed by usability experts, who compare existing user interfaces to usability principles or guidelines (Faulkner, 2000; J. Preece et al., 2002). Many different sets of guidelines exist for traditional user interfaces. As AR is an emerging field, there are few usable heuristics or guidelines that focus on AR applications have been established or standardized. The aim of heuristic evaluation is to identify any possible usability problems, by considering that the system being evaluated follows or does not conflict with the guidelines.

Gabbard and Hix have been involved in evaluating several virtual reality systems, such as a battlefield (Hix et al., 1999) and medical (Gabbard et al., 1999) visualisation virtual environments. They have extended their evaluation work on virtual environments (Gabbard and Hix, 1997) to AR systems, and have developed a set of guidelines that may be specifically applied to AR systems (Gabbard, 2000). Unfortunately there were too few usability experts available to conduct a formal heuristic evaluation. The guidelines set by Gabbard and Hix were too general and would be difficult to apply to the interfaces that I had developed. There were also doubts about the status of these guidelines, as they appear untested.

Time constraints affected the possibilities for the type of evaluation that could be carried out. This led to a straightforward formative evaluation approach to obtain general user feedback on the interfaces that had been developed. A formative usability evaluation can be used to refine and improve user interaction by placing representative users in task-based scenarios and assessing the interface's ability to support user exploration, learning and task performance. The aim is to determine how these problems might affect usability, and consider approaches to overcome these problems.

An evaluation has been designed to cover all aspects of the interfaces that had been developed, from simply viewing objects in AR, labelling and the information manipulation techniques. As this evaluation dealt with such a wide range of issues, specific aspects could not be examined in great detail. In the future, perhaps smaller evaluations could be carried out that focus on particular issues. It is hoped that the results of the formative evaluation will point out the more obvious usability problems, which might affect the results of future tests into specific details.

To perform the evaluation, observation, talk aloud and questionnaire based techniques have been used. With the talk aloud approach, users are given the interface and some task scenarios to perform. The users are asked to express their opinions and feelings as they are performing the different tasks, and are observed by the evaluator. This can be used to understand how the user approaches the interface and what considerations they keep in mind as they use it. If users express a different sequence of steps than that dictated by product to accomplish their task goal it may be that the interface design is convoluted. Thinking aloud gives a better understanding of users' mental model of systems. Users can also provide new terminology to express an idea or function that could be incorporated into the design (Faulkner, 2000).

5.2 Evaluation Plan

A pilot study was conducted following the evaluation plan presented here. It involved six people, four males and two females. This group of users was picked randomly, but the selection process avoided computer scientists from the research group whenever possible. Their ages ranged from 21 to 31 years, the average age was 24.6 years and the standard deviation was 3.09 years. Users were mostly from an engineering or scientific background. It was found that most had little or no AR and VR experience, although two subjects had significant experience with 3D computer gaming.

The evaluation consisted of analysing various aspects of participants' performance of a set of tasks for each of the different interfaces. Their actions and opinions were noted as they performed each task, and a questionnaire was used to capture their opinions in a quantitative way. The evaluation was split into three main stages, with each stage analysing a different aspect or interface. For clarity, these stages are presented in Table 5.1 and are described in more detail below.

As the system requires the use of an HMD, users were given instructions by the evaluator throughout the evaluation, and encouraged to interact with the system. The evaluator provided assistance when necessary. Most of the evaluation stages had a number of substeps (numbered in the Table 5.1). All users tried each step, and all steps were performed in order except for steps 5 and 6, whose order was randomised to avoid learning affecting users' judgment of the information manipulation techniques. When users completed one of these steps, they would remove the HMD and fill out the portion of the questionnaire relevant to the aspect of the system that they had just used. A copy of the evaluation script can be found in Appendix C along with a copy of the questionnaire in Appendix D.

The evaluation was performed on a table, so the user could comfortably manipulate the marker cards in front of them. Users were wearing the Cy-Visor Mobile personal display, model DH-4400VP. A small analogue video camera is mounted on top of the

Stage	*Evaluation Steps*	*Description*
AR Environment	1 - AR Environment	To allow users to become accustomed to the AR environment, only aircraft are projected onto marker cards without any information (i.e. labels).
Labelling and Linking	2 - Fixed Labelling	Objects are presented with a set of descriptive labels that reflect the fixed label selection technique, where labels are static and so obscure the object.
	3 - Mobile Labelling	Objects are presented with the same set of labels but now the mobile label selection technique, where labels move so they don't obscure the object, is in place. Users were asked to pick their preferred selection approach, and the chosen technique was used during the remainder of the evaluation.
	4 - Linking	The objects are now presented with a different set of labels, which include various link labels. Users' feedback on the use of links between objects was recorded.
Manipulating Information	5/6 - Salt and Pepper	Users are given time to get used to the "salt and pepper" interface and then are asked to set the information to a number of levels. The order in which users experience these interfaces is random
	5/6 - Waves	The same is performed to the "waves" interface. The questionnaire sections for both interfaces are identical, allowing a comparison between the interfaces to be performed.

TABLE 5.1: Evaluation outline

display, and the video stream was captured by a Hauppage WinTV video capture card on a 2.4GHz Pentium 4 PC running the ARToolKit applications. The HMD with the mounted camera used for the evaluation is shown in Figure 5.1 (a). The camera captured video at a resolution of 320x240, although the objects drawn by the ARToolKit used the HMD's maximum resolution of 800x600. The ARToolKit display was mirrored on a second computer so the evaluator could view the users' progress and observe their interactions with the system. This setup is shown in Figure 5.2 (b).

For each part of the interface tested, users were asked on the questionnaire how they felt about that aspect of the system. This involved two questions, which were also asked in each stage of the evaluation so that a comparison could be performed on users' feelings throughout the evaluation. The questions were:

- What was your general impression of the system? *(Boring - Exciting)*
- What did you think of this system? *(Very difficult to use - Very easy to use)*

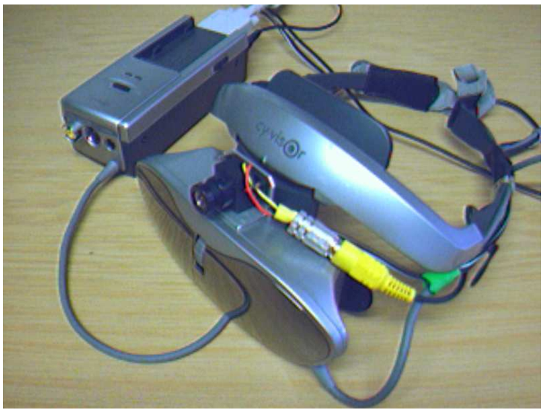


Figure 5.1: Cy-Visor Mobile Personal Display with mounted camera

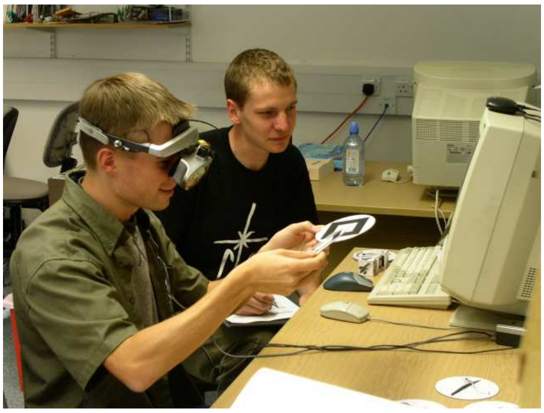


Figure 5.2: Evaluation setup

In each of the evaluation stages, different issues and features were being tested; these are described in detail in the following pages.

5.2.1 AR Environment

This stage allowed users to become accustomed to the ARToolKit. Previous experience performing demonstrations showed that novices to AR can take time to understand what is happening in front of them, let alone become comfortable manipulating the marker cards. A brief explanation of how the ARToolKit tracking system works was given and users were encouraged to test the system to the limits, for instance holding the marker cards at a very sharp angle to the camera. They were also taught to hold the markers so that they remained unobscured by fingers and shadows.

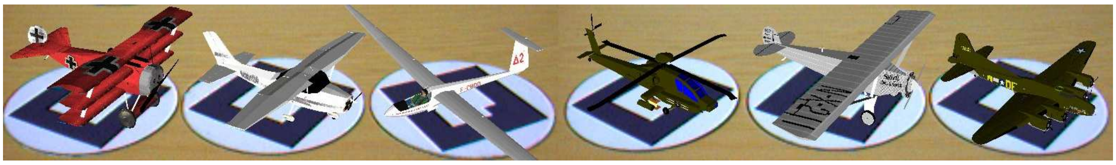


Figure 5.3: Aircraft used for the evaluation

Users were encouraged to examine the various aircraft objects used in the evaluation: a Fokker DR1, a Cessna 172, a glider, an Apache AH64 helicopter, the St. Louis and a B17 Flying Fortress. These objects are shown in Figure 5.3. The user had access to these same objects for each of the evaluation stages.

The evaluator observed their general reactions to the system, and noted any peculiarities and troubles in their interaction with the marker cards.

This initial stage of the evaluation was used to obtain feedback on different aspects of the system, including the quality of the experience in terms of immersion and visual quality, as well as how they felt manipulating the marker cards. Besides the two standard questions described above, i.e. users' feeling towards the system (boring to exciting) and ease of use, users were asked:

- Did you have the impression that the virtual objects (i.e. aeroplanes) were part of the real world or did they seem separate from the real world? *(Separate from real world - Belonged to real world)*
- Did you have the impression that you could have touched and grasped the virtual objects? *(Not at all - Absolutely)*
- Rate the quality (visual) of the aeroplane models. *(Low quality - High Quality)*
- How did you find manipulating the objects? *(Awkward - Natural)*
- How did you find holding the objects? *(Uncomfortable - Comfortable)*

5.2.2 Labelling and Linking

As described in Chapter 4, there had been attempts to improve the labelling technique so that labels did not obscure the object. This section of the evaluation examined the results of these efforts; by allowing users to experience both label selection techniques and choose their preferred method to use in the remainder of the evaluation.

The same objects from the first stage were provided, but this time a set of labels was automatically applied to them. This set of labels was not restricted to a particular topic, such as avionics or armament, and as such no colour coding of the labels, as described in Chapter 4; the labels were all coloured white. Users could select and examine the labels but were not able to modify the label content. They were given time to become used to the fixed labelling technique by themselves, and then asked by the evaluator to select particular labels, read them and identify the feature highlighted by each label. This might involve the evaluator asking the following:

- "Pick the label next to the tail plane"
- "Read out this label"
- "Can you identify the feature this label points at?"

They were then asked to fill in the section of the questionnaire about fixed labelling, which asked their opinions on the labels. This included the following questions:

- Did the labelling deteriorate in any way your experience with the object? *(Yes, very much so - Not at all)*

- Rate the readability of the labels. *(Unreadable - Readable)*
- How did you find label selection? *(Awkward - Natural)*
- How did you find selecting a particular label? *(Very difficult - Very easy)*
- Did labels obscuring the object deteriorate your experience of the system? *(Yes, very much so - Not at all)*

The same objects with the same set of labels were then shown to the user; however, the second labelling technique was now activated. After allowing users to select some labels they were asked if they noticed any differences about the labels, and the evaluator noted their response. They were then asked as before to select particular labels, read them and identify the feature pointed at by the label. A section of the questionnaire was then filled in, which was identical to the earlier part on labelling but had the addition of the following questions:

- Did the label movement deteriorate label readability? *(Yes, very much so - Not at all)*
- Did the label movement deteriorate label selection? *(Yes, very much so - Not at all)*
- Did the label animation deteriorate in any way your experience with the object and labels? *(Yes, very much so - Not at all)*

Users were also asked to select their preferred label selection technique, which was used for the remainder of the evaluation.

By exposing the two labelling techniques in this way, users gained significant experience with the label selection techniques before they looked at the information manipulation interfaces. The experiment was conducted in this way so that issues related to labelling would not affect the evaluation of the information manipulation interfaces.

To conclude this stage of evaluation, a third study was dedicated to obtaining feedback about the use of links between objects, described in Chapter 4. The objects were presented with a set of link labels as well as normal descriptive labels. Users were asked if they could identify the difference between links and labels, and the idea of links between objects was introduced. Users were asked to manipulate the links across objects so that the label could be read. Their opinions were then noted on the questionnaire on the following:

- How did you find distinguishing labels and links? *(Hard - Easy)*

- What did you think about links between objects? *(Useless - Useful)*
- Did you enjoy using links between objects? *(Unenjoyable - Enjoyable)*
- How did you find readability of links (between 2 objects)? *(Unreadable - Readable)*

5.2.3 Manipulating Information

Two interfaces for manipulating information were evaluated: “salt and pepper” and “waves”. As these interfaces are similar in terms of functionality, the evaluation consisted of performing the same tasks and answering identical questions on both systems in the questionnaire. The results of users’ opinions on both systems could be compared to each other. The order in which users experienced each interface was randomised to avoid learning affecting users’ opinions on the interfaces.

Users were given time to become accustomed to each interface, and taught how to manipulate a context dispenser adequately. All users were given the avionics context dispenser to start with. From previous experience conducting demonstrations, many users often have trouble shaking the context dispensers in the “salt and pepper” interface. Users have tended to shake them as if they were real shakers, resulting in poor tracking as either the marker is covered by the users’ hands or they shake too vigorously and the ARToolKit is unable to keep up with the shaker speed. It was important to train the users to shake the markers effectively so that this did not affect their feelings towards the interface. There had been limited previous demonstration experiences with the “waves” interface so it was harder to predict problems users might have with this. Users were talked through the use of the context dispensers until they were comfortable with the system.

There were many issues to cover when evaluating these two interfaces. These included how users felt about adding and removing information, sensitivity, setting information to a certain level, how they felt about mixing information, general enjoyment and feelings about the interface and so on. When users were comfortable handling the context dispensers, they were asked to set the information level to the maximum, minimum and intermediate levels. The evaluator then asked about their opinions on viewing the amounts of information they had applied and how much more there was to apply.

Another aspect of the interfaces was mixing the different types of information. Once users had experienced manipulating one type of information (they initially only had access to avionics information), two other context dispensers were provided: armament and general information. Users were then encouraged to mix the different types of information, and asked to examine issues such as what happens when they try to add armament context to a civil aircraft without any guns. On the questionnaire they were asked what they felt about label colouring, identifying the different types of context and the ease of use in mixing information.

The questionnaire was used to obtain users' opinions on the two interfaces with the following questions:

- How did you find adding information? *(Awkward - Natural)*
- How did you find removing information? *(Awkward - Natural)*
- How did you find setting the amount of information to a desired level? *(Awkward - Natural)*
- How did you find the sensitivity of adding or removing labels (e.g. did labels appear faster than you wished?) *(Sensitivity was a problem - Sensitivity was fine)*
- Could you clearly view how much information had been applied to an object? *(Unclear - Clear)*
- How did this (i.e. being able to see how much information had been applied) affect your experience with the system? *(Strongly affected - Not affected)*
- How did you find mixing different types of information? *(Hard - Easy)*
- How did you find identifying the different types of information? *(Hard - Easy)*
- What did you think of the colour of the labels? *(Useless - Useful)*
- How did you find the enjoyment of using the system? *(Unreadable - Readable)*

5.2.4 Post Evaluation

The users were asked some questions about their overall experience with the system, and were asked to add any comments if they wished. The questions asked here were:

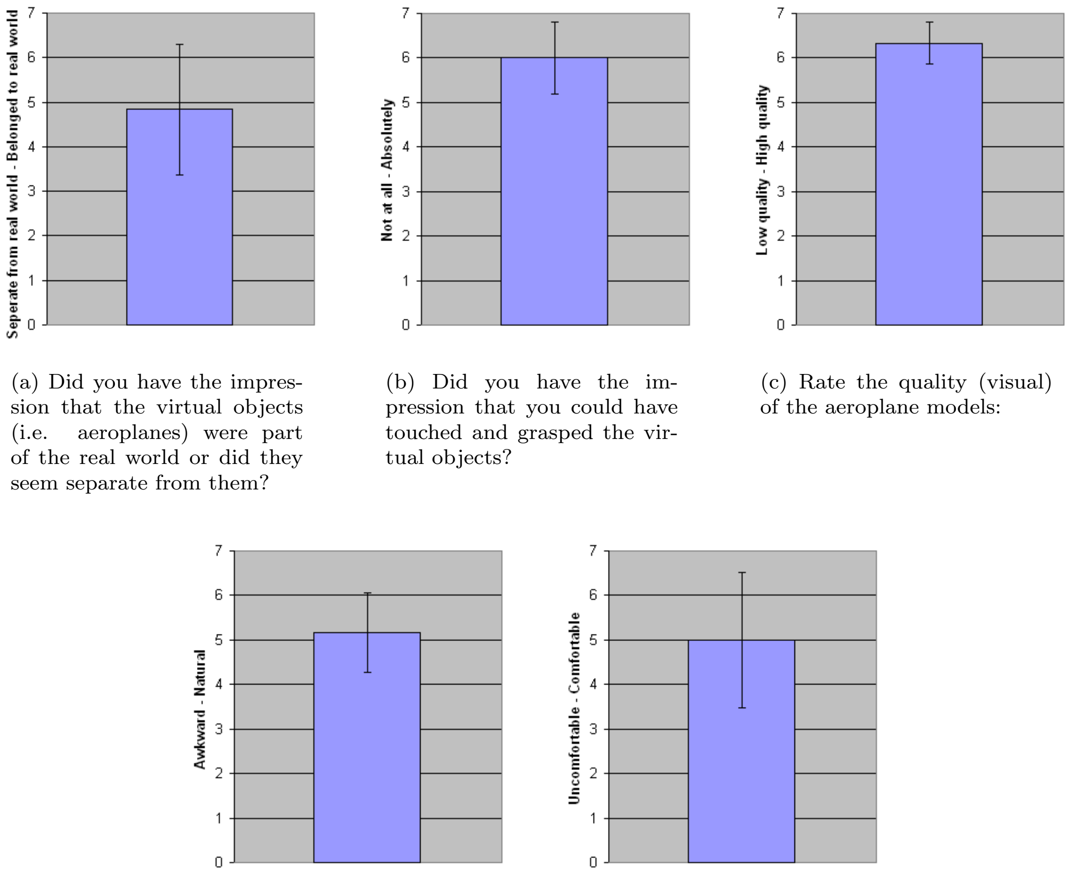
- How would you rate the comfort of the 3D glasses? *(Bad - Very Good)*
- Would you try out the same or a similar technology again? *(Not at all - Yes, very much)*

5.3 Results

The results from the user questionnaire are presented over the following pages, and discussed in more detail in Section 5.4. A copy of the evaluation script can be found in Appendix C along with a copy of the questionnaire can be found in Appendix D. All of the user comments are shown in Appendix E.

5.3.1 AR Environment

Figure 5.4 shows the results of the user feedback recorded on the questionnaire for the first part of the evaluation, where users were presented objects with no information overlaid on them. Each graph shows the average response of the user for that question, with answers ranging from 1 to 7 on the scale shown on the Y-axis of each graph. The error bars in each graph show the standard deviation of the data.



(a) Did you have the impression that the virtual objects (i.e. aeroplanes) were part of the real world or did they seem separate from them?

(b) Did you have the impression that you could have touched and grasped the virtual objects?

(c) Rate the quality (visual) of the aeroplane models:

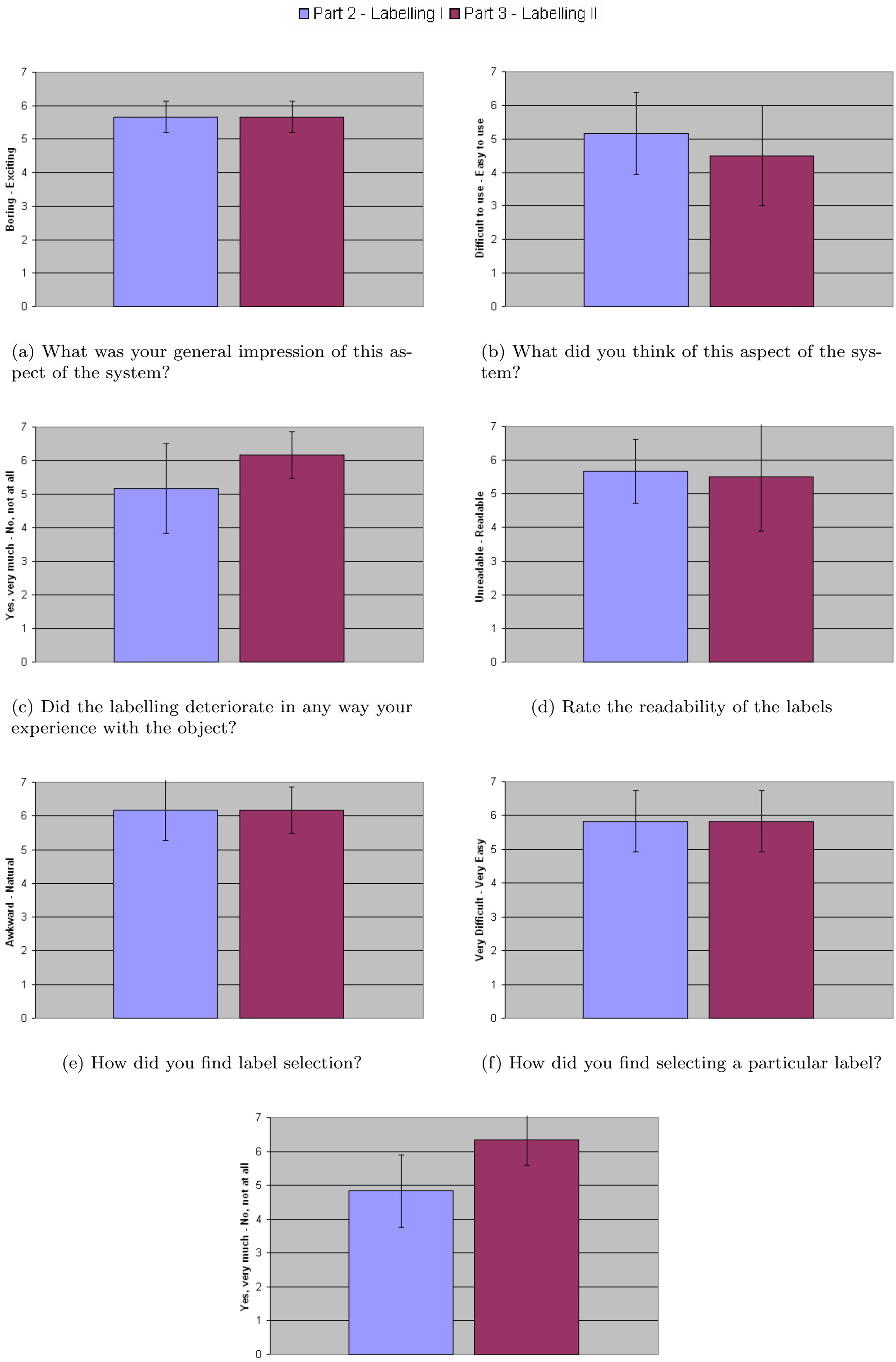
(d) How did you find manipulating the objects?

(e) How did you find holding the objects?

FIGURE 5.4: AR environment results

5.3.2 Labelling

Figure 5.5 shows the results of the user feedback recorded on the questionnaire for the second and third steps of the evaluation, where users were presented objects with a fixed set of labels overlaid on them. Each graph shows the average response of the user for that question for both fixed labels and mobile labels. As before, answers range from 1 to 7 on the scale shown on the Y-axis of each graph and the error bars in each graph show the standard deviation of the data.



(a) What was your general impression of this aspect of the system?

(b) What did you think of this aspect of the system?

(c) Did the labelling deteriorate in any way your experience with the object?

(d) Rate the readability of the labels

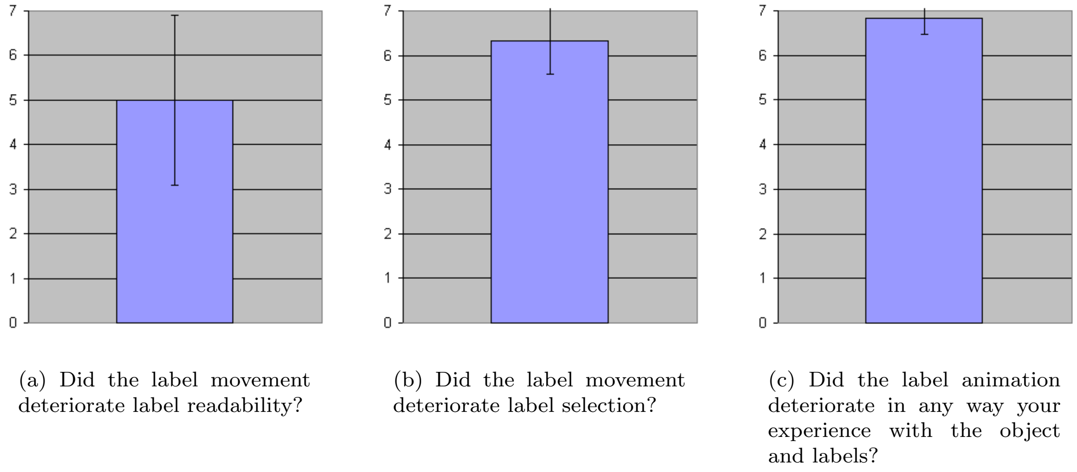
(e) How did you find label selection?

(f) How did you find selecting a particular label?

(g) Did labels obscuring the object deteriorate your experience of the system?

Figure 5.5: Labelling results

An additional three questions were asked about the mobile labelling technique, the results of which are shown on Figure 5.6. Answers range from 1 to 7 on the scale shown on the Y-axis on each graph. Error bars indicate the standard deviation of the data.



(a) Did the label movement deteriorate label readability?

(b) Did the label movement deteriorate label selection?

(c) Did the label animation deteriorate in any way your experience with the object and labels?

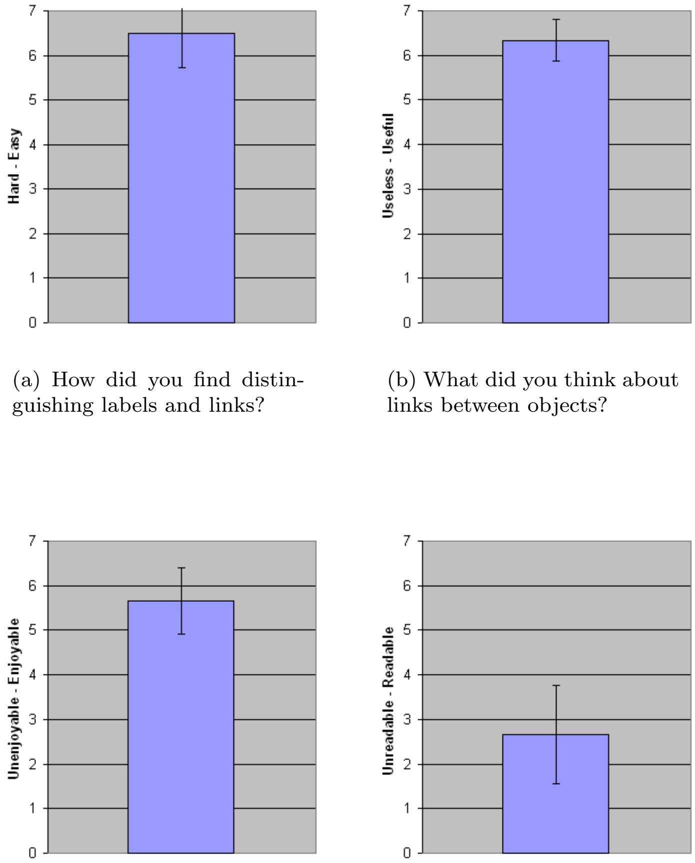
Figure 5.6: Animated labelling results

All but one user immediately identified the difference between the types of labelling, resulting in an average of 83%. However, this user appreciated the difference as soon as it was revealed.

Overall, users showed no preference to either forms of labelling, with a 50% - 50% split between the two approaches.

5.3.3 Links

Figure 5.7 shows the results of the user feedback recorded on the questionnaire for the fourth evaluation step. Users were presented objects with a set of link labels as well as normal labels overlaid over the objects. Each graph shows the average response of the user for that question, with answers ranging from 1 to 7 on the scale shown on the Y-axis of each graph. The error bars in each graph show the standard deviation of the data.



(a) How did you find distinguishing labels and links?

(b) What did you think about links between objects?

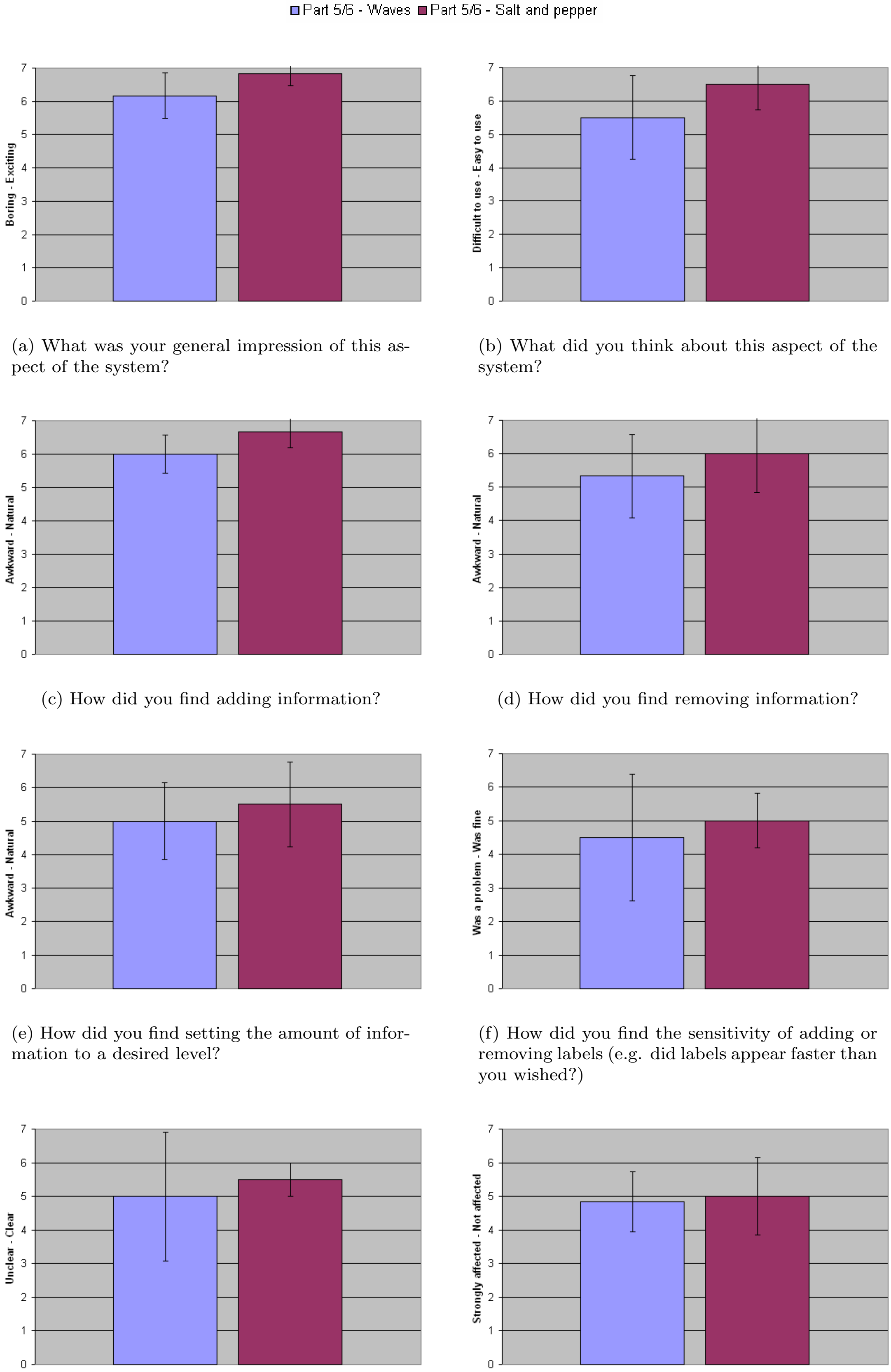
(c) Did you enjoy using links between objects?

(d) How did you find readability of links (between 2 objects)?

FIGURE 5.7: Linking Results

5.3.4 Manipulating Information

Figure 5.8 shows the results of the user feedback recorded on the questionnaire for the fifth and sixth steps of the evaluation. In each of these steps users were introduced to the information manipulation techniques. The order in which the interfaces were experienced was randomised for each user. Each graph shows the average response of the user for that question for both the "waves" and the "salt and pepper" interface. As before, answers range from 1 to 7 on the scale shown on the Y-axis of each graph and the error bars in each graph show the standard deviation of the data.



(a) What was your general impression of this aspect of the system?

(b) What did you think about this aspect of the system?

(c) How did you find adding information?

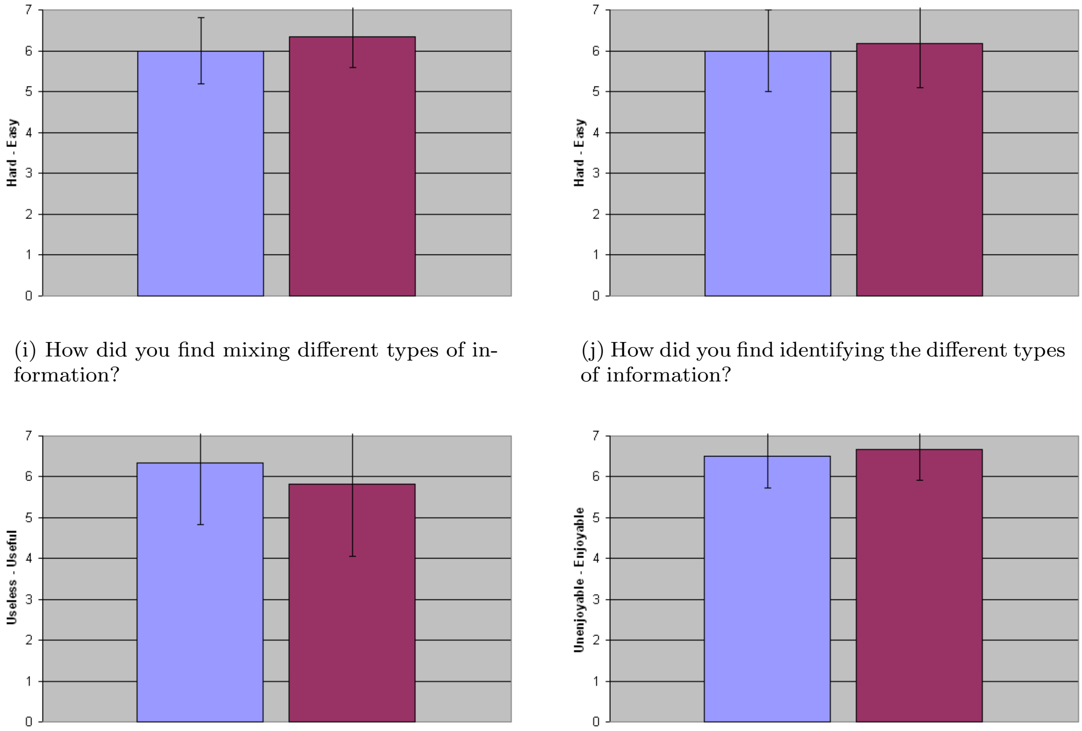
(d) How did you find removing information?

(e) How did you find setting the amount of information to a desired level?

(f) How did you find the sensitivity of adding or removing labels (e.g. did labels appear faster than you wished?)

(g) Could you clearly view how much information had been applied to an object?

(h) How did this (i.e. being able to see how much information had been applied) affect your experience with the system?



(i) How did you find mixing different types of information?

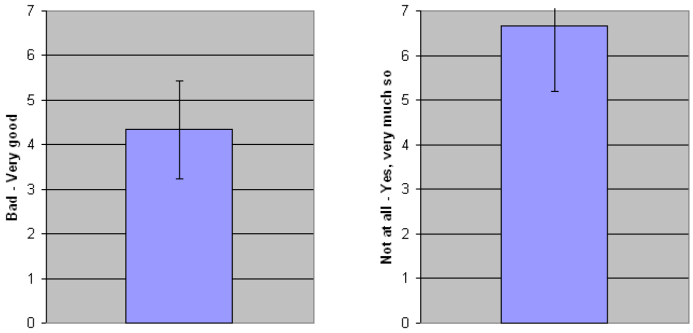
(j) How did you find identifying the different types of information?

(k) What did you think of the colour of the labels?

(l) How did you find the enjoyment of using the system?

FIGURE 5.8: Manipulating information results

5.3.5 General



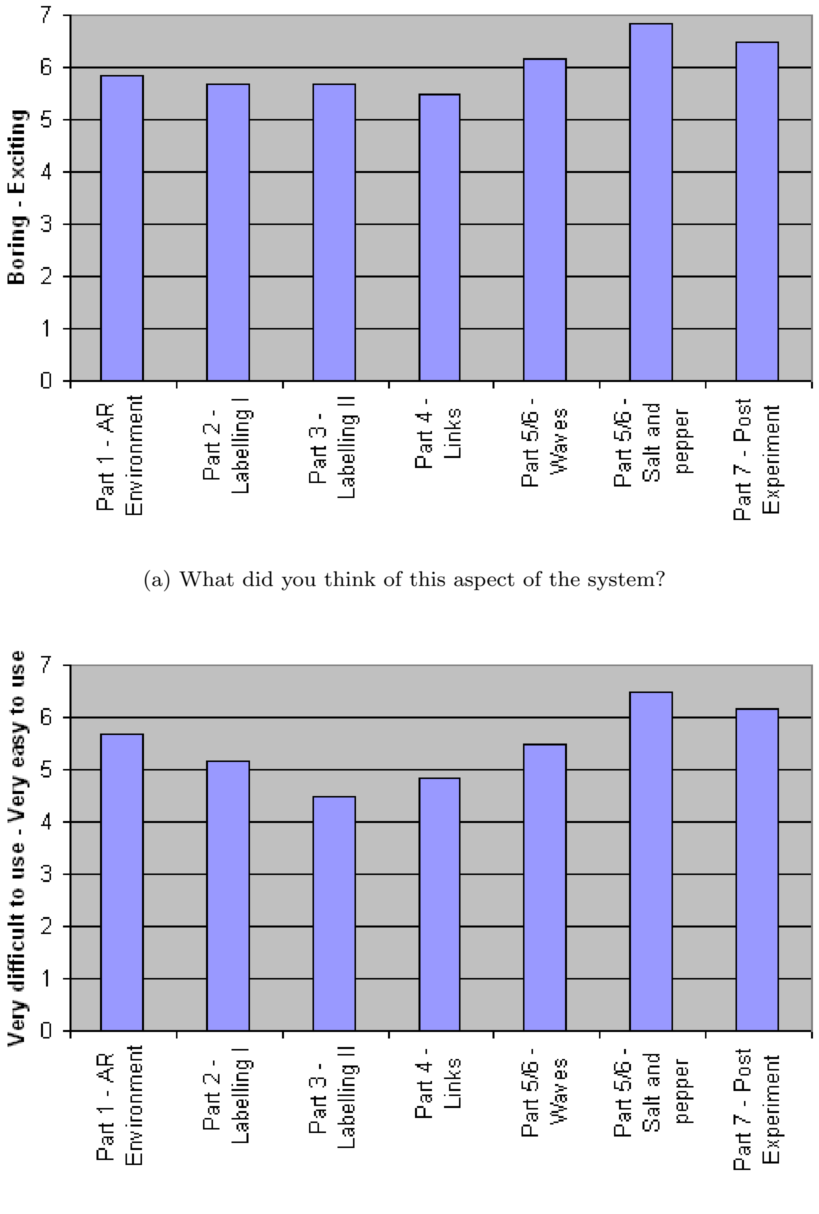
(a) How would you rate the comfort of the 3D glasses?

(b) Would you try out the same or a similar technology again?

FIGURE 5.9: Post experiment results

Figure 5.9 shows the results of the user feedback recorded on the questionnaire after they had completed the evaluation, where users simply rated their opinions on their experience as a whole. Each graph shows the average response of the user for that question, with answers ranging from 1 to 7 on the scale shown on the Y-axis of each graph. The error bars in each graph show the standard deviation of the data.

5.3.6 Overall reactions



(a) What did you think of this aspect of the system?

(b) What was your general impression of this aspect of the system?

FIGURE 5.10: Results across all aspects of the system

For each step in the evaluation, users were asked the same two questions on what they felt about that aspect of the system, in terms of their reaction to it and the ease of use. The results for each step in the evaluation process are shown in Figure 5.10, with answers ranging from 1 to 7 on the scale shown on the Y-axis of each graph.

5.4 Discussion

The results across all aspects of the system shown in Figure 5.10 provided some interesting conclusions. Users appeared to appreciate the features of the information manipulation techniques, as they gave slightly higher scores for these approaches. It also seems that users' opinions of the system increased through the experiment, as their reaction to the overall system was higher than the first stage where only objects without information were presented to them. Users seemed to prefer the "salt and pepper" interface over the "waves" interface. Also, the results showed some trouble with the labelling and linking techniques; this is discussed in more detail below.

5.4.1 AR Environment

Most users seemed to enjoy using the ARToolKit, and some users even started playing with the aircraft. Generally there was a very positive reaction from the users towards the AR environment.

Some users had trouble with the video see through characteristics of the HMD. The users' view the real world through the camera, and as the camera is mounted slightly above their eyes there is a slight displacement from the real world. Many users had trouble focusing the HMD, which caused trouble later when they tried reading labels. Other comments included the HMD display was of poor quality; this is mostly due to the low resolution of the camera video stream. Some users commented that the HMD's field of view was too narrow, as often the tracking would lose markers off the side of the screen. One user complained that they could not estimate distances easily through the HMD, as he was holding two objects at different distances from the camera but thought they were at the same distance.

Learning not to obscure the ARToolKit markers depended on the individual; most people became more comfortable with this after a little time. Several people would hold the marker very close to the side or bottom of the video stream, so the marker tracking was lost. This affected the mobile labelling technique when people tried to move the object to read the labels. As users got used to the system the problem became less apparent. In fact, some people wanted to try the system to the limits, testing the robustness of tracking by experimenting with obscuring the marker corners and holding markers at sharp angles to the camera.

Some users seemed to have trouble holding the markers, and this comment was raised often. Some people suggested using a holder at the bottom, but this would cause trouble with the “waves” interface where objects should be moved around the tabletop. People complained that the markers, mounted on CDs, were too large. People reacted well to the “salt and pepper” cubes, as they are fairly small they are easier to hold and hide when necessary. Most users had trouble picking up markers off the tabletop. A possible solution will be to create smaller disks out of foam card or thick cardboard, but further investigation is needed to discover an optimal marker size so that the ARToolKit tracking remains accurate. Space also needs to be left around the markers so that users are able to hold them comfortably without obscuring the marker patterns.

Users appreciated the image identifying the rear of each marker cards, as described in Chapter 4. This simple feature was extremely useful as users had to rely on the images when choosing which object to view. However, some users expressed that the images would be more effective in colour, as the quality of the HMD display made it difficult to identify the aircraft by shape alone.

To enable label selection the system requires that only the object users are interested in is visible. This was accepted well by users and caused little trouble. At each evaluation stage all markers started face down, most users would automatically place the markers faced down when they had finished viewing them. This resulted in few occasions where users had trouble with selecting labels because there were more than one visible objects. It may be useful to implement a visual indication to highlight that an object is ready for label selection to avoid confusion when multiple objects are visible.

The questionnaire was used to determine the quality of the immersive experience by asking users what they felt about the virtual objects and the quality of the aircraft models. As can be seen from the results, shown in Figure 5.4, people were happy with the quality of immersion. One user commented that the *“objects seem incredibly realistic, in particular their movement following the card”*. This is somewhat surprising as this was not the focus of the evaluation prototypes, and is probably due to users' limited experience with AR systems. The refresh rate of the ARToolKit is slow when compared to magnetic tracking systems and there are systems that have focused on extremely realistic rendering of the virtual objects, yet the users found that the simple ARToolKit approach was perfectly acceptable and even exceptional. This result might hint that novice users are very tolerant to the quality of the models and AR environment, although it is likely that a comparison to more realistic systems will not result favourably towards the ARToolKit. Also, in museum installations it may be that users will be more demanding. For example, an art expert may not accept viewing a piece of artwork through a video see-through HMD such as the one used in the evaluation.

One user commented that the *“illusion on ‘real’ became broken after a little while”*, as they became aware of the limitations of the ARToolKit. For instance, when the ARToolKit draws virtual objects over real world objects even when these are closer to the camera than the virtual object. For example, in Figure 5.11 (left) the glider wing

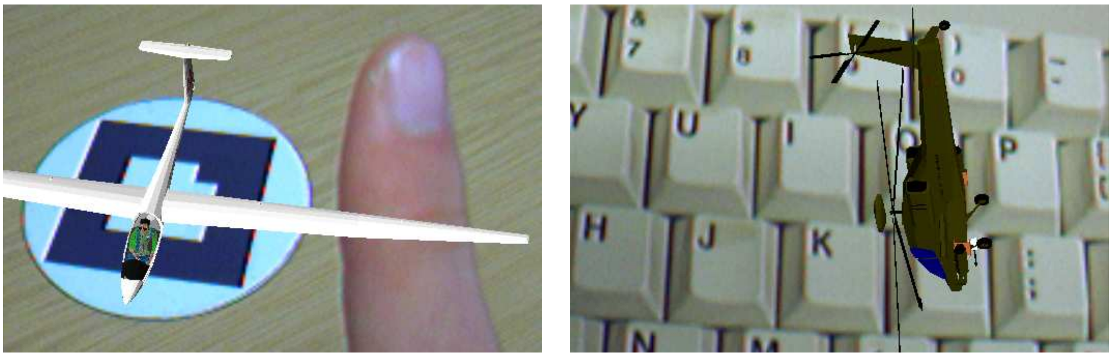


Figure 5.11: Typical problems with the ARToolKit

being drawn over a user's finger although the finger is closer to the camera than the marker. Other factors include flickering, where the ARToolKit is unable to cleanly track a marker which causes the virtual object to flicker in and out of the scene. Virtual objects are sometimes placed on the wrong marker or even drawn over the background scene, as shown in Figure 5.11 (right). All of these factors affected users' experience during the evaluation.

5.4.2 Labelling

This stage of the evaluation consisted of testing the fixed labelling technique to the mobile labelling. Several users were strongly affected by the way that labels obscured the object, and were more critical in their responses. One user commented that *"labels did get in the way of the object and the thing they were pointing to, so I can't see both the description and object at the same time"*.

The questionnaire results indicate there was little difference in the users' reactions between both labelling techniques and the AR environment with objects only (Figure 5.10).

Generally people were able to read labels with little trouble, as can be seen in the results from the question on readability, shown in Figure 5.5 (d). There were some complaints regarding the HMD, such as the HMD screen being dirty and some users had trouble focusing the display. People were able to select labels with few problems, with most trouble being caused by ARToolKit tracking errors or users accidentally obscuring markers.

Most users picked up label selection quickly and found it natural to use, as can be seen in Figure 5.5 (e) and (f). Most users found that the interface became easier to use with time. This was due to both experience with the interface and getting used to handling the markers in front of the HMD. This suggests that novice users of tangible AR interfaces will be extremely affected by experience with the systems, not only the interface itself but also the limitations of the technology. For example, with time users were able to recognise by themselves when they were obscuring the markers and the effect this had on the interface.

The study on mobile labelling, where labels move out of the way to allow users to view the object, provided interesting results. There was a problem with this technique that caused trouble to users. As labels move from over the top of the objects, they sometimes moved partially off the screen, forcing users to adjust the object's position so that the label is visible on the screen. This problem was particularly prominent on the larger labels. This would often require the object to be moved some distance away from the user before the label was completely visible, which resulted in the label becoming hard to read. Some users resorted to moving their heads rather than the objects in their hands, which often resulted in the markers going off the field of view. By moving the object around the screen, sometimes the label the user was interested in would become unselected. However, some users said they actually found it natural to move the object in order to make labels readable, even after experiencing many of these conditions. The label movement system also had some problems where under certain conditions the label would move jerkily, also reducing label readability.

These problems are reflected in the results for animated labelling. Users generally rated this feature as the hardest to use, as can be seen in Figure 5.10 (b). However, the problems described did not affect the users' opinion on label readability; this may be because once a label was in a suitable position they had no problems reading it. Label movement also had little effect on either of the label selection questions, as shown in Figure 5.5 (e) and (f).

The most significant differences between the two labelling methods were found in the questions about the labels deteriorating their experience with the system and with the objects, shown in Figure 5.5 (c) and (g). These results reflect the advantage in moving labels from over the objects, allowing users to freely view the object features and related descriptions. The difference between the two techniques could in reality be larger, as users became more aware of the faults of the fixed labelling technique after they experienced mobile labelling.

Users were able to identify the relevant label features with both label selection techniques, although they found it was easier to accomplish with the mobile labels. With the fixed labelling system, users were forced to unselect the current label to be able to have a better view of the object and thus be able to identify a label's respective feature. With the mobile labelling technique, users were able to immediately see features pointed at by the labels.

Although users were generally able to identify features pointed at by labels, some had trouble working out the exact area of certain features. For instance, labels describing aircraft fuselages (the body of the aircraft) users thought the line pointed at the tail, whilst it was really pointing at the fuselage area of the aircraft between the cockpit and the tail. It may be desirable to highlight the feature area in some way when a label is selected.

Three extra questions were dedicated to the mobile labelling technique, and the results are shown in Figure 5.6. These questions examined factors such as whether the movement affected label readability and selection and what users thought about the animation used in the label movement. It can be seen that users felt that the movement affected readability, but did not affect selection as much. There was a wide range of opinions regarding the readability; this was due to some users feeling comfortable with moving the objects to make labels readable. The problems weren't very prominent with all users, also contributing to the wide range of responses. The reaction to the animation used for labels' movement was very positive.

Users were asked to choose their preferred selection mechanism, and the result was split 50%. Generally, users that had not experienced many of the problems with the mobile labels described above chose this technique. The other users appreciated the benefits of the mobile labelling, but felt more comfortable with the more straightforward fixed labelling due to the problems with mobile labelling. This shows that if the problems with the mobile labelling system can be overcome, it could significantly improve users' experience with the system.

5.4.3 Linking

Figure 5.7 (a) shows that users were able to easily distinguish between link labels and normal descriptive labels. Users appreciated the use of labels between objects and seemed to enjoy them, as shown in Figure 5.7 (b) and (c).

There was a problem when selecting link labels between two different objects. As described in Chapter 4, this was achieved by moving the source and destination objects apart which would increase the size of the link label between the two objects. However, the system would only expand the label to a certain size, which would invariably be too small for the text to be legible. This is reflected in the poor response in the readability question in Figure 5.7 (d) and needs to be resolved. Some users overcame this problem by using the normal label selection to read the contents of a link label, alternating between the two modes by hiding the target destination object. It was discovered that users enjoyed the process of highlighting a link label by moving the two objects apart.

Other issues were raised during the evaluation of links, from both user feedback and general observation of users with the interface. These are mostly straightforward and can be easily resolved in the future.

Users complained that the leader lines between object features and the link labels were hard to see against dark backgrounds, such as the black ARToolKit marker and some of the objects. This comment was also raised about the leader lines in normal descriptive labels. A solution to these issues may involve using thicker lines and investigating effective colour schemes, perhaps animated, to use with leader lines. A related issue is that some users could not identify link direction (i.e. which feature was the anchor and which was the destination), so this needs to be highlighted in some way.

When the destination object of a link label is not visible, the current implementation does not provide any indication of the target object. Users were forced to look through the other objects in turn until they found the respective destination object, which was frustrating to several users. One approach to solve this problem would be to indicate the target by displaying a thumbnail of the target object in the link label or as part of the leader line.

A minor point was raised about links between different objects. When a destination object appears or disappears, the link label immediately jumps across to spanning between the objects or back to its normal position. No animation between these steps was used. Users noticed the lack of this animation, which contrasted with the animation present in all other label movement and selection processes. It was also felt that this animation might be used to give an idea of link direction, as users would be able to see the link move across to reach the destination anchor.

5.4.4 Manipulating Information

It was surprising to note that all users picked up the interfaces very easily. In past demonstration sessions many novice users had experienced great difficulties with the “salt and pepper” shakers, but this problem was not encountered during the evaluation. Reasons for this may be the improved implementation of the shaker tracking that was used in the evaluation for the first time. This implementation was more sensitive and easier to use, but was also more prone to accidental triggering. As the interface relied on users hiding the shakers when these were not in use relied, accidental triggering was kept to a minimum and did not affect the evaluation. Another factor might be that the evaluator was giving instructions and a quick demonstration on how to perform the shaking. It would be interesting to test how users would fare without any assistance.

In most of the responses, it can be seen that users preferred “salt and pepper” to the “waves” interface. Salt and pepper is a very fun interface to use, and this can be seen in the results in Figure 5.8 (a) and (l). One user commented that the “salt and pepper” interface felt *“very natural and easy to use”*. Many users had more trouble being comfortable with the “waves” interface, possibly due to the awkward size of the marker cards used. The size made a big impact on activating and deactivating the “waves” context dispensers. Users that tried the “salt and pepper” interface after having used "waves" commented that they much preferred the size of the "salt and pepper" shakers for covering and hiding.

Users rated "waves" harder to use, as shown in Figure 5.8 (b), and took longer to understand what was going on. Users preferred using the "waves" interface in the air rather than on the table top, and even after suggesting the table they insisted on holding the markers in the air. This may have affected their opinions of the system, as the "waves" interface had been specifically designed for use on a flat surface. It was interesting to note the users behaviour towards this issue: they used the markers in they way they preferred rather than following the evaluator's suggestion.

Regarding the process of manipulating information shown on the objects, users decided that "salt and pepper" was slightly more natural to use. Three different aspects were examined: adding information, removing information and setting information to a certain level. The results are shown in Figure 5.8 (c), (d) and (e).

With the "waves" interface, the labels themselves are used to show the amount of information that has been added to an object. In the "salt and pepper" interface, particles drawn on the marker base are used in addition to the labels. This is reflected in the responses shown in Figure 5.8 (g) and (h). While the "waves" interface clearly shows the amount of available information through the distance between the object and context dispensers, users decided that the base particles in the "salt and pepper" interface were more effective.

Several users didn't realise that the information in the labels was changing as they moved the "waves" around the objects. Certain labels hold little information, perhaps being two or three lines long. These labels attract more attention than large labels as all labels are drawn at the same size when minimised, with each line of text shown as a thin line in large labels. It was discovered that users did not notice the contents of the larger labels changing when the wave dispensers were moved around, so did not appreciate the evolution of information. This also affected the responses to the question on sensitivity, shown in Figure 5.8 (f). As some users were unable to clearly see the information changing with the "waves" interface they thought the problem lay in the sensitivity.

The problems with the "salt and pepper" interface discussed in Chapter 4 were apparent, even though they did not seem to affect users' responses on the questionnaire. When users shook context onto an object that has no more available information, the context particles bounce off. Users were not aware of this occurring, so the bouncing animation needs to be highlighted to be more visible. Users did not seem to be too affected by the limitations of the "salt and pepper" interface in displaying the amount of remaining available information. For example, all users would keep shaking context onto objects when all information had already been added. Most users had trouble identifying when they had reached an intermediate level of information applied to objects.

Many of these issues might be resolved by adding a visual indication of the amount of information on an object, perhaps by using a dial or slider bar. This had not been considered in the past as efforts in this area had focused on techniques that followed the “salt and pepper” metaphor, such as particles landing on the marker base. Unless a better approach is discovered the use of dials should be investigated.

The questionnaire feedback on mixing different types of information is shown in Figure 5.8 (i). It can be seen that users manifested a slight preference for the “salt and pepper” interface in this area. All users understood what happened when they tried to add contexts such as armament to airplanes with no guns; this was the case with both interfaces.

Some users disliked the “salt and pepper” interface forcing them to remove all types of context at the same time, and would prefer to have finer control over this. One user actually suggested something similar to the vacuum idea mentioned in Chapter 4, where individual contexts could be removed by holding the relevant context shaker next to objects. This had not been implemented at the time as there were concerns that it might spoil the shaking metaphor. However, it appears to be a useful feature that might be investigated in the future.

Activating the “waves” context was considered a problem by some users, mostly because of the size of the “waves” context marker. This issue could be improved by the suggestions described above, where markers can be made of foam card of thick cardboard. On the whole users were happy to flip the “waves” context markers, although they were very receptive of alternative approaches that were suggested, such as embedding a button into the markers to activate the context. Some users took an unexpected approach to handling context activation when they used the table top to manipulate the markers. They would move their heads towards the table top until only the object they were interested was visible, allowing them to focus solely on that object and its labels.

Users responded positively towards the colour coding of context dispensers and labels, as can be seen in Figure 5.8 (j) and (k). It is interesting to note that in (k) users preferred the use of colour on the “waves” interface than with “salt and pepper”. This could be due to the fact that the context dispensers are larger in the wave markers than the cubes used in “salt and pepper”, so the label colouring is easier to identify with the “waves”.

5.4.5 General

Users were asked about the comfort of the HMD, and the results are shown in Figure 5.10 (a). Users were not very positive towards the HMD, and many had complained during the evaluation of several problems they had encountered.

In contrast, when asked about if they would consider this kind of technology again the response was extremely positive. This suggests that despite the problems encountered with the technology such as tracking and display users seem excited by the potential of AR systems. It could be argued that many of the more serious problems discovered during the evaluation were down to underlying AR technology issues rather than the interfaces themselves.

5.5 Conclusions

Generally, feedback was positive and users appeared to appreciate both the use of labels as well as the tangible interaction techniques to manipulate the presented information. This was principally reflected in the users' comments as well as through the discussion with users as they used the interfaces. The questionnaire results also hint at conclusion, as users' opinions of the system tended to increase as they used the more complex features of the system.

The evaluation process was also extremely useful in raising the various usability problems and suggestions for improvements. Resolving these issues will no doubt increase the usability of the systems and allow for more complex future evaluations. The problems included:

- The design of the marker cards, which had been mounted on compact discs, was flawed. The markers were too large, making them uncomfortable to hold and handle. As compact discs are very slim, they can be hard to pick up. Markers should be made out of foam card or thick cardboard in the future, and should be smaller so they are more comfortable to manipulate. The images on the reverse of the marker cards were useful, but would be more effective in colour.

- A visual marker to indicate that users are able to select labels on an object will avoid users getting confused if they don't notice that there is more than one object visible at once.

- Label leader lines, pointing from object features to the labels, are drawn as thin black lines. It was discovered that this is not clear enough, especially when the black marker cards are in the background. Also, often the leader lines were confused as features on the objects themselves. These need to be changed so they are more visible, perhaps by using thicker lines, colour and animation. It may also be desirable to highlight features that the labels are describing on the object itself.

- Changes must be made to the mobile labelling technique which caused various problems. In particular, care must be taken that by moving a label out of the way the label does not move off the screen. Resolving this issue will greatly improve the usability and hence users' experience of the system.

- The display of link labels between two objects must be tweaked so that the labels are more easily readable. The direction of links needs to be highlighted. This could be achieved by providing animations as links between two objects are activated. Also, link destination objects need to be displayed so that users are able to pick the destination object to view the link.

- When designing “waves”, it had been decided that the distance between the context dispensers would provide enough visual feedback to tell how much information had been added. However, users were generally not able to appreciate the change in the information displayed by looking at the labels. An indicator or dial should be provided to the “waves” interface to overcome this.

- A similar approach should also be applied to the “salt and pepper” interface. During the design process, this had been discarded because of the desire to keep within the shaker and particle metaphor. However, this is an example of where sticking within the constraints of a metaphor can cause serious usability problems: it will be extremely useful to view a clear indication of the amount of remaining information through an ‘unnatural’ indicator. The same can be said about the much needed inclusion of the ‘hoover’ mechanism, where individual contexts can be removed by holding the context shaker close to the object. This mechanism will offer greater control for the removal of information in the “salt and pepper” interface.

Unfortunately, the evaluation was unable to cover certain aspects of the interfaces in great detail, such as users’ mental model as they mixed different types of information with the tangible context dispensers. The comprehensive nature of the evaluation meant that each evaluation took close to an hour per user to complete. It would have been too demanding on the users to take more in depth analysis as this would have increased the length of the evaluation.

With the evaluation it was noted that it can be hard to determine the usefulness of the interfaces as users were casually reading labels out of curiosity rather than need. Their feedback and opinions may be completely different if they had to rely on the information to perform a task. For example, installations of this technology in real environments might produce different results. It may also be possible to conduct an evaluation where users are tested about the knowledge they acquired through the use of the interfaces.

Future evaluations could be carried out that focus on more specific aspects of the interfaces. The results of the evaluation performed will be extremely useful, as the issues raised can be resolved to improve users’ experiences with the interfaces. The usability problems described above can be eliminated so that they do not affect the results of future evaluations.

Other types of evaluation could also be carried out. For example, it may be interesting to compare these tangible adaptable information interfaces with more traditional approaches, such as paper based museum guidebooks or web based systems. Mobile computing guides, based on devices such as PDAs, are being used in various cultural and museum-based projects. A comparison could be made between such systems and AR based displays.

5.6 Chapter Summary

This chapter has discussed issues that were considered when planning the evaluation of the interfaces presented in Chapter 4. The plan for the formative evaluation conducted was described, and involved six subjects experiencing the different tangible AR interfaces implemented through an HMD. The evaluation was split into three stages. To allow users to become accustomed to the AR environment the first stage projected objects only, with no associated information (i.e. labels). The second stage compared the two types of labelling, fixed and mobile, as well as the linking mechanism between objects. In the third stage the two approaches for manipulating information on objects, "salt and pepper" and "waves", were compared. Generally, feedback was positive and users appeared to appreciate both the use of labels as well as the tangible interaction techniques to manipulate the presented information. The results of the questionnaire filled out by the users, together with their comments and observations of the users working with the interfaces, raised several improvements that could be made to the systems.

Chapter 6

Conclusions and Future Work

In this Chapter I reflect on the work presented in this thesis. This includes discussing the technique to apply information to AR environments using an Open Hypermedia link server and the various interfaces for manipulating the information in AR. Issues encountered in the AR technology used to create the interfaces are discussed, as are the experiences of the interface and metaphor design. Ideas for future work are also described.

6.1 Open Hypermedia in Augmented Reality

This thesis has discussed ways in which Open Hypermedia systems can benefit information display in AR. Chapter 3 introduced an Open Hypermedia technique for presenting information over objects, such as museum artefacts, in AR environments. This system is able to place dynamically created, adaptive labels over 3D models of objects. It uses the Open Hypermedia approach of keeping data and links separate using linkbases; the linkbase is used to attach relevant descriptions to the respective area of the 3D model. The linkbase is served by the Auld Linky link server, a context based link server implementing FOHM.

Various AR systems have investigated hypertext, and some of these were described in Section 3.3.1. However, there has been very little work on applying Open Hypermedia techniques to AR environments, in particular using this approach in the augmentation process for overlaying information over objects.

The technique presented in this thesis can offer several advantages over existing AR labelling systems, including those described in Section 3.2. Many of these systems have focused on projecting high quality visual representations of objects into real world scenes. Some systems have used simple labels to present information about objects, but rely on other display mechanisms for showing more complex information. The technique has advantages over similar systems that also use labelling for presenting information overlaid on objects within AR. For instance, the dynamic approach is more flexible than manually authoring many static scenes in a 3D modeller.

Several benefits arise from the Open Hypermedia link server approach. New models can be added to the system and be annotated with existing descriptions through the Open Hypermedia generic linking mechanism. Different sets of descriptions can be applied by adding additional linkbases. As artefact information is kept separate from the artefact models, it is easier to author and maintain. Descriptions can be modified by updating the linkbases as often as necessary. For example a collections management database in a museum can be used to automatically update some parts of the dynamic information. Links and hence descriptions are more easily maintained with linkbases, they can also be changed in a link editor without the need of a 3D modeller.

Whilst the success of the technique cannot be assessed, as this would require using the system under real conditions, experience with the Open Hypermedia approach to labelling objects in AR has been very positive. It has enabled the design and implementation of the tangible interfaces for manipulating the adaptation process, where it was used extensively. Users generally appreciated labelling in these interfaces, and could see the benefits of viewing labelled objects in AR environments. The labelling process is completely transparent to the user, as with the implementation of caching (see Section 3.5.3) there is no noticeable delay when information is requested. It might be interesting to investigate more advanced labelling issues, described in the future works section (Section 6.5), and perhaps compare the functionality and performance of the technique to other AR systems.

Many of the authoring issues inherent in other AR systems are overcome through the use of the Open Hypermedia approach, as described in Section 3.5.4. However, a new set of problems is inherited. Authoring FOHM linkbases can be very time consuming, and requires considerable technical knowledge (linkbases are edited in XML). Little support is available for creating complex information structures, which can hinder the development of material for AR applications. Recent developments, involving XML style sheets for automating much of the FOHM authoring process, will improve the situation, and a possible integrated environment is discussed in the future work section (Section 6.5).

6.2 Interacting with the Information

Tangible AR systems offer a very natural interaction style, which suit educational applications, such as information displays in museums. During the evaluation presented in Chapter 5, users recognised the potential of being able to hold and manipulate virtual objects in their hands, and considered this approach to be far superior to traditional 3D object displays such as VRML browsers.

The labelling technique introduced in Chapter 3 added descriptions of individual object features to the AR environment. Presenting adequate amounts of information in the labels required a selection mechanism so that labels can be minimised and highlighted when required, reducing visual clutter on the display. This selection mechanism used the orientation of the labelled object, so that the label closest to the users' viewpoint is selected. This approach has an issue in that large labels often obscure an object when selected. To overcome this, the process was modified so that the selected label moves from in front of the object. In the evaluation both approaches were compared, and although there was a problem with the moving labels implementation (labels often moved partially off the screen and became unreadable), users appreciated how the label movement improved the visibility of an object.

With an effective selection mechanism, it was possible to explore links between features on different objects in the AR environment. This is accomplished by the use of link labels, which are drawn as an annotated line between a links' source and destination anchors when these are visible. During the evaluation, although a graphical hitch made link labels effectively unreadable, users were able to realise the usefulness of this approach, and enjoyed using the linking mechanism. Other issues were also raised that should be investigated in the future work.

6.3 Interface and Metaphor Design

The contextual nature of the Auld Linky link server was extremely influential on the course of the research presented in this thesis. Initially, context was used to provide straightforward adaptation of the material presented depending on the type of users. This was described in Section 3.5.2.3. The power of contextual Open Hypermedia led to the design of tangible interfaces where users are able to physically manipulate the adaptation process and thus changing the nature of the presented information.

This involved designing several interfaces, of which two seemed to warrant further evaluation. "Salt and pepper" allows users to construct recipes of information by sprinkling different types of context onto objects. "Waves" uses the position of context dispensers in relation to objects to affect the information displayed about an object. The feedback from the evaluation was positive and users appeared to appreciate these tangible interaction techniques to manipulate the presented information. This was principally reflected in the users' comments and through discussions held with users. The questionnaire results also showed the strength of the interfaces, as users' opinions tended to become more favourable as they used the more complex features of the system.

The design of these interfaces made strong use of real world metaphors, especially in "salt and pepper". Users reacted well to these metaphors, as was shown during the evaluation in Chapter 5, and as a result I consider that the use of real world metaphors to be extremely powerful in tangible AR interfaces. However, I discovered that in order to avoid certain usability problems it is it acceptable to stray from these metaphors. In the "salt and pepper" interface the "hoover" functionality, which would give users more control over the context applied to an object, was not implemented as it might "break" the sprinkling metaphor. However, during the evaluation users requested finer control for removing information from a mixture of contexts, with one subject even describing a mechanism similar to the "hoover" idea. Another instance where designing within the metaphor was harmful was the decision of not showing a dial or slider to indicate context levels on the "waves" interface.

Another interesting aspect is that users preferred "salt and pepper", which uses a strong real world metaphor, to the "waves" interface, where a more abstract metaphor involving distances is used. Although there were several usability problems that may have affected users' opinions (as discussed in Section 5.4), it could be that users did indeed prefer the strong real world metaphor. It might be interesting to carry out further investigations in this area.

The nature of the complex information space used as the source for the label descriptions caused various problems when designing the interfaces. The main issue was that the interfaces had to handle mixtures of different types of information, so several promising interfaces were discarded. For example, if there was only one type of information the fan and clouds metaphor could be an interesting way of exposing levels of detail.

6.4 Reflections on the Technology

Much was learnt about building applications with the ARToolKit, and both its advantages and disadvantages were appreciated while implementing the various interface prototypes.

The ARToolKit tracking mechanism is very prone to errors, such as swift movement, users accidentally obscuring markers, ambient lighting and so on. These issues must be taken into consideration when designing tangible interfaces with ARToolKit. There are restrictions in the way that the marker's properties are used in the interface. For example, while occlusion could not be used for triggering button events (discussed in Section 4.2.3) it was very effective for activating the context dispensers in the "waves" interface.

The main advantage in using the ARToolKit is its low requirements, especially when compared to other technologies such as magnetic tracking devices, which can be very expensive. An ARToolKit interface only requires the computer running the software, a camera and a display. The use of an HMD is optional as the system can be displayed on a monitor or projector, although this might require different aspects to be considered when designing the interfaces. As shown in this thesis, the ARToolKit is perfect for prototyping and conducting simple user evaluations. However, other tracking systems (magnetic or ultrasound) are more robust, accurate and have faster refresh rates. In the future, as this type of technology becomes more accessible and cheaper, it will be more feasible to create powerful tangible AR interfaces for installations with limited resources, such as schools and small museums.

During the evaluation, many users criticized the HMD, commenting on problems such as the video see-through nature of the display, not being able focus, poor quality of the video stream and that the field of view was too narrow. From this feedback, and from my experience with the HMD, I believe that affordable, high quality optical see-through displays are very important in the design of future AR systems. Until then, it may be wiser to investigate monitor or projector based displays, which will affect the design of the tangible interfaces.

6.5 Refining the Existing Interfaces

During the evaluation, several usability problems were discovered and several suggestions were made that would improve the interface. These included refining the label movement system so that labels did not move outside the screen and various improvements to the linking mechanism, such as indicating link destinations. Besides these, there are several improvements that might also be desirable.

It would be interesting to apply the labelling technique to real objects. This could be achieved by placing the real object next to ARToolKit markers and calibrate the tracking system with the object's position. Labels could then be applied to the object, although these must be occluded by the real object when they are behind it.

A problem with the labelling technique is that the 3D models for objects must be broken down into their subfeatures manually using a 3D modeller, such as Blender. This can be time consuming and complicated so one of the future tasks is to provide a more intuitive authoring system. An idea for this is to customise and extend the Blender (NaN) modeller to provide a group of tools for splitting and labelling the artefact 3D models. Blender is a good platform to develop for as it is free and it is extremely customisable using Python scripting. Tools for authoring the information associated with the object features could also be added, for example a suitable FOHM editor incorporating the recent work on stylesheets that automate part of the FOHM editing process. This would result in a powerful, integrated authoring environment for creating and editing material for AR applications.

It would be interesting to incorporate FOHM behaviours to objects, and use this to activate events on the presented 3D models. When the user has selected a feature, it could be animated or highlighted, audio could be played and so on. This approach could also be used to implement animations for highlighting certain features on an object, which is important in various industrial maintenance applications. For instance, arrows could show how an object is taken apart or point out which screws need to be removed. Another improvement would be to display other types of media in labels, such as images, videos or even 3D objects.

The issue of adapting how the label information is displayed when there are many objects visible at once was discussed in Section 4.8.2. The current implementation avoids this problem by only allowing users to select labels on an object when no other objects are visible. However, there are many possible strategies which could be investigated in the future.

6.6 Future work

The ideas and extensions suggested above are very closely related to the current implementations. Whilst designing these systems, I have been considering different interfaces that offer substantially different experiences and raise new research challenges.

6.6.1 Embedding Functionality into Markers

The lack of a reliable mechanism for triggering specific user interface events, such as buttons, in ARToolKit interfaces can be a problem. In the "waves" interface, although users were happy to hide or flip marker cards over to activate context dispensers, some users commented they would prefer to press a button to perform this action. This problem could be overcome by embedding hardware into the marker cards, and various approaches were considered.

A very crude solution might be to incorporate the innards of a wireless mouse into an ARToolKit marker. The mouse buttons would then be used within ARToolKit interfaces, and there may innovative uses for other mouse features, such as the scroll wheel. It would also be interesting to implement interfaces with the mouse itself (i.e. as a 2D pointing device), merged with the ARToolKit marker.

Some interfaces may require this kind of functionality on various markers, which would be impractical as this would require a different mouse for each marker. Smart-its are small-scale, wireless embedded devices that can be attached to many types of objects (Gellersen et al., 2002). A system can communicate with various smart-its at once and they are extremely versatile as different types of sensors can be placed on a smart-it. Examples include buttons, temperature sensors and touch sensitive plates. Smart-its could be embedded into various ARToolKit markers, for instance on each context dispenser marker in both the "waves" and "salt and pepper" interface. Beyond simple mechanisms such as buttons, smart-its could incorporate hardware shaker sensors, which used in conjunction with the ARToolKit tracking system could provide more robust shakers in "salt and pepper".

6.6.2 Narratives in AR

Narrative is a powerful technique for information presentation. Recently in Southampton an effort has been made to construct narratives using contextual link servers and FOHM (Weal et al., 2001). These techniques could be applied to AR environments, so that documentary style information can be projected onto objects.

This raises new challenges in designing tangible interfaces to navigate within the narrative spaces. There are several obstacles to overcome, including mechanisms for moving forwards and backwards through a story, and some kind of selection mechanism if a branching narrative is used. Various possibilities for presenting the documentaries could be considered, such as speech, labels or subtitles.

As a tangible AR documentary is being followed, it may discuss a certain feature on an object. The user might be unsure what feature is currently being described, so the system could point it out. This could provide an innovative interaction mode where users are encouraged to explore the object by following the features being described by a narrative. For instance, an arrow could point out the current feature. When this is not in view, the arrow could point towards it so that users manipulate the object so that they can get a clear look at the feature. This could also be applied to augmenting real objects, with users walking around the object and following the flow of the narrative. In this way, users can be led through a dynamic tour of both that object and its respective information space.

An extension of this idea involves anthropomorphic agents presenting narrative information. Instead of hearing an off screen narrator or presenting subtitles, a visible character could talk to the user either by speech synthesis or using some kind of speech bubbles. This character could also address the highlighting issue, for example it could would walk around objects pointing out features being discussed. The use of characters might raise other interesting possibilities for presenting information. For instance, if two characters are used they can talk to each other. One idea was to have an angel and a devil, inspired by traditional cartoon representations of conscience, flying around an object. The angel might describe useful aspects of features while the devil might would talk about the controversial aspects of the object. Another idea would be to have the angel always telling the truth while the devil might tell entertaining lies. The use of these characters in this way might be a good way of getting the user's interest and giving them an involving experience.

6.6.3 Advanced Interfaces to Information Spaces

During the design of the interfaces for manipulating the presentation of the information, the nature of the structures for modelling the information space restricted the design. As the structures were stored in a linkbase, the complex information space was effectively static. This resulted in problems when designing interfaces to handle mixtures of information, as described in Sections 4.6.5 and 4.7. It was hard to gauge the impact of this during the evaluation, as users were casually reading the labels and were not forced to understand or learn the information presented.

I believe there is scope for interesting work to be undertaken with these types of interfaces. The "waves" interface began to explore this area by allowing the information to be changed depending on the position of the context dispensers. In the future, this could be expanded so that systems might dynamically recognise the context of the objects present in a scene.

For example, imagine a scene containing several modern aircraft. The AR environment would query the information space for information on these objects, and their relationships between each other. When an older aircraft is added to the scene, the context of the original presented information would change to a more historical perspective as within the scene the overall context reflects both modern and historic aircraft. The labels on the aircraft might change to indicate the differences between modern and historic capabilities of aircraft. In this metaphor, an overall scene is created and the context of the labels in the scene is derived from the combining of the objects within it.

To accomplish this, more advanced systems for creating and adapting the information are required, perhaps including dynamic generation of material. There is a lot of ongoing research in this field, some of which has been performed at Southampton and has involved Auld Linky and FOHM (Bailey et al., 2002; Kim et al., 2002). Interacting with these kinds of information spaces will become increasingly complex, as can be seen from the simple example described here. I believe that tangible AR interfaces are able to expose such information spaces in natural, intuitive and easy to use ways.

Research into this area might show that not only are Open Hypermedia techniques useful for applying information to AR environments, but also that tangible AR interfaces can offer powerful advanced interaction opportunities for manipulating complex information spaces, resulting in a relationship beneficial to both research fields.

6.7 Summary

In this thesis I have shown how Open Hypermedia systems might be used as the information systems behind AR environments. This provides benefits to AR developers, not only because of the existing Open Hypermedia tools but also because of the applicability of Open Hypermedia interaction techniques to the AR domain. AR environments are able to expose the adaptation process possible with Open Hypermedia systems with tangible interaction techniques, resulting in a natural and intuitive way to manipulate complex information spaces.

Appendix A

Adaptive Hypermedia in Augmented Reality

Poster presented at the Third Workshop on Adaptive Hypertext and Hypermedia at the Twelfth ACM Conference on Hypertext and Hypermedia

Appendix B

Salt and Pepper: Tangible Hypermedia using the ARToolKit

Poster presented at ART02, The First IEEE International Augmented Reality Toolkit Workshop

Appendix C

Evaluation Script

A copy of the evaluation script used during the evaluation of the system

Appendix D

Evaluation Questionnaire

A copy of the evaluation questionnaire filled out by users during the evaluation of the system

Appendix E

Qualitative Evaluation Results

Comments made by users during the evaluation

Part 1 - AR Environment

- To hold the objects you need to put your fingers around them which sometimes occludes the target system.
- Possibly because it is the first time that I am using the HMD and I am not used to hold and manipulate the objects I had some problems with objects disappearing.
- The objects seem incredibly realistic, in particular their movement following the cards.
- Instead of planes, why not hamburgers and food products, with labels indicating quality, calories and protein content? Possible funding from McDonalds.
- Illusion of realness became broken after a little while.

Part 2 - Labelling I

- Labels could be a bit transparent (like in some games) so you can see a bit of what's behind them.
- Again, in the beginning it is a bit strange to deal with the object and labelling, but after some experience the process becomes much easier.
- Large labels or small text could become difficult to read, need readjustment of focus on the HMD.
- Labels did get in the way of the object and the thing they were pointing to - I can't see both description and objects at the same time.

Part 3 - Labelling II

- Just a slight software glitch as the labels go off screen.
- As I had the experience of Part 2 of using the system in this part I was much more aware of the sensibility needed to deal with the system. As I go through the experience I also go through a process of learning how to use the system.
- The labels tend to go to a corner of the screen, which means there is the need to adjust the position a little more to be able to read them. This is still quite natural, though.
- Before it didn't bother me that the labels covered the aeroplane, but after seeing them moving aside I like this method more and the previous one looks much worse.

- The other labels were more straightforward.
- Had got used to labelling selection by now. Feel glasses needed wider field of view - labels sometimes moved off screen.

Part 4 - Links

- Would not know which objects to pick up without being told, i.e. the link needs an address, e.g. plane B etc.
- It's difficult to read since, the maximum you can increase the labels is too far away to read.
- The links appeared in a very small font because the objects occupied a large area of the screen. It would be useful to zoom in on the text for reading only.
- It was easy to read the link label when only one aeroplane was showing, but once the link was enabled it became quite difficult. On the links in the objects itself it would help if links were of a brighter colour.
- Took a little while to get used to - quite fun. Text too small to read properly.

Part 5 - Manipulating Information I - Salt and Pepper

- Removing information is tricky, you need to shake and mind your timing. Would be easier to have another shape to remove information instead of shaking the main shape.

  The colours for different types were useless because I could not remember which colour was which, maybe a little icon would help.
- It was easier to add information than to take it away.
- The system feels very natural and easy to use.
- Couldn't remove specific information (i.e. just guns)

  Bit unclear when information had been added, particles gave some clue but not 100% clear.

Part 6 - Manipulating Information II - Waves

- Mixing types was tough because you have to cover the type shape to avoid the information disappearing while you're grabbing the other type shape.

- I found the shakers in salt and pepper easier to understand the amount of information added.
- It was not so natural to switch on the labels and holding them at the desired level of information since every time one moves the information card the labels change. On the other hand it is a good arrangement to have in a fixed position, e.g. on a table and move around the objects.
- Too sensitive - steep learning curve. Cards needed to be smaller. Information types were being used felt jumpy when holding, needed to put them on the table top.

Part 7 - Post Experiment

- Fingers occluded the shapes a lot, especially when turning around from nose to tail etc. Suggest adding a handle to the bottom of the shape.
- As it was the first time I had used this type of technology I took some time to get used to it. But after an hour I was more comfortable using the system.
  I think that the technology needs to evolve a bit to be user friendly, but it is going in the right way. Users need some time to get used to the system.
- The system was very easy to use and it felt incredibly natural and with the right level of information selectable by the user.
  The 3D glasses were comfortable to use but were awkward to focus, making the labels difficult to read at times. Also, the screen area felt small, one could only have 1 or 2 objects in the screen with a comfortable level of detail (.e. near enough to see the details).
  Overall the system is very impressive. I'd like one of these, please!
- Overall very exciting system. It takes only a few minutes to get used to manipulating the objects! And it's nice to be able to read the information you want, when you want in such an easy way.
- Technology needs improving, more comfort and better quality markers. More colourful markers.

Bibliography

3DCafe. http://www.3dcafe.com, 2003.

ARVIKA. Augmented reality for development, production and servicing, www.arvika.de.

R. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments 6*, 4:355 – 385, August 1997.

R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6): 34–47, Nov/Dec 2001.

C. Bailey, W. Hall, D. Millard, and M. Weal. Towards open adaptive hypermedia. In In P. De Bra, P. Brusilovsky, and R. Conejo, editors, *Proceedings of Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 36–46, 2002.

Mike Bajura, Henry Fuchs, and Ryutarou Ohbuchi. Merging virtual reality with the real world: Seeing ultrasound imagery within the patient. Technical Report TR92-005, The University of North Carolina at Chapel Hill, 1, 1992.

T. Barbieri and P. Paolini. Cooperative visits to www museum sites a year later: Evaluating the effect. In *Proeedings of Museums and Web 2000*, April 2000.

Blaine Bell, Steven Feiner, and Tobias Hollerer. View management for virtual and augmented reality. In *UIST*, pages 101–110, 2001.

M. Billinghurst, J. Bowskill, N. Dyer, and J. Morphett. An evaluation of wearable information spaces. In *Proceedings of Virtual Reality Annual International Symposium (VRAIS '98)*, 1998a.

M. Billinghurst, S. Campbell, D. Hendrickson, W. Chinthammit, I. Poupyrev, K. Takahashi, and H. Kato. Magic book project, http://www.hitl.washington.edu/magicbook/, 2000.

M. Billinghurst and H. Kato. Collaborative mixed reality. In *Proceedings of International Symposium on Mixed Reality (ISMR '99). Mixed Reality - Merging Real and Virtual Worlds*, pages 261–284, 1999.

M. Billinghurst, H. Kato, S. Weghorst, and T.A. Furness. A mixed reality 3d conferencing application. Technical Report R-99-1, Human Interface Technology Laboratory - University of Washington, 1999.

M. Billinghurst and T. Starner. Wearable devices: New ways to manage information. *IEEE Computer*, 32(1):57–64, 1999.

M. Billinghurst, S. Weghorst, and T.A III Furness. Shared space: An augmented reality approach for computer supported collaborative work. *Virtual Reality*, 3:25–36, 1998b.

Mark Billinghurst. Seattle art museum project, http://www.hitl.washington.edu/people/grof/sharedspace/ download/projects/sam/, 2001.

Oliver Bimber, L. Miguel Encarnao, and Dieter Schmalstieg. The virtual showcase as a new platform for augmented reality digital storytelling. In *Eurographics Workshop on Virtual Environemnts IPT/EGVE'03*, 2003.

Oliver Bimber, Stephen Gatesy, Laurence Witmer, Ramesh Raskar, and L. Miguel Encarnao. Merging fossil specimes with computer generated information. *IEEE Computer*, 35(9), September 2002.

F. Biocca, A. Tang, D. Lamas, J. Gregg, P. Gai, and R. Brady. How do users organize virtual tools around their body in immersive virtual and augmented reality environments?: An exploratory study of egocentric spatial mappings of virtual tools in the mobile infosphere. Technical report, East Lansing, MI: Media Interface and Network Design Lab, 2001.

D. Breen, R. Whitaker, E. Rose, and M. Tuceryan. Interactive occlusion and automatic object placement for augmented reality. In *Proceedings of Eurographics '96*, 1996.

V. Bush. As we may think. *The Atlantic Monthly*, 176(1):101 – 108, 1945.

A. Butz, T. Hollerer, S. Feiner, B. MacIntyre, and C. Beshers. Enveloping users and computers in a collaborative 3d augmented reality. In *In Proceedings of IWAR '99 (Int. Workshop on Augmented Reality)*, 1999.

K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou. Developing a context-aware electronic tourist guide: Some issues and experiences. In *Proceedings of CHI 2000*, pages 17–24, April 2000.

J. Conklin. Hypertext: An introduction and survey. *IEEE Computer*, 20(9), 1987.

D. Curtis, D. Mizell, P. Gruenbaum, and A. Janin. Several devils in the details: Making an ar application work in the airplane factory. In *in Proceedings of the First International Workshop on Augmented Reality (IWAR 98)*, 1998.

H. Davis. *Data Integrity Problems in an Open Hypermedia Link Service. Ph.D.thesis.* PhD thesis, University of Southampton, 1995.

H. Davis, W. Hall, I. Heath, G. Hill, and R. Wilkins. Microcosm: An open hypermedia environment for information integration. Technical Report CSTR 92-15, University of Southampton, 1992a.

H. Davis, W. Hall, I. Heath, G. Hill, and R. Wilkins. Towards an integrated information environment with open hypermedia systems. In *ECHT 92. Proceedings of the ACM conference on Hypertext*, pages 181–190, 1992b.

H. Davis, A. Rizk, and A. Lewis. Ohp: A draft proposal for a standard open hypermedia protocol. In *Proceedings of the 2nd Workshop on Open Hypermedia Systems, ACM Hypertext 96*, pages 27–53, 1996.

Peter Dick. Bmw augmented reality based repair instructions. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, September 2002.

G. Drettakis, L. Robert, and S. Bougnoux. Interactive common illumination for computer augmented reality. In Julie Dorsey and Phillipp Slusallek, editors, *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pages 45–56, New York, NY, 1997. Springer Wien.

D. Engelbart. Augmenting human intellect: A conceptual framework. Technical Report AFOSR-3223, Contract AF 49(638)-1024, Stanford Research Institute, October 1962.

Epsis. Real-time tracked virtual imaging for broadcast video, http://www.epsis.com.

Xristine Faulkner. *Usability Engineering.* Macmillan Press Ltd, 2000.

S. Feiner, B. MacIntyre, M. Haupt, and E. Solomon. Windows on the world: 2d windows for 3d augmented reality. In *Proceedings of UIST '93 (ACM Symp. on User Interface Software and Technology)*, pages 145–155, 1993a.

S. Feiner, B. MacIntyre, and T. Hollerer. A touring machine: Prototyping 3d mobile augmented reality systems for exploring the urban environment. In *Proceedings of The First International Symposium on Wearable Computers (ISWC '97)*, pages 74 –81, 1997.

S. Feiner, B. MacIntyre, and D. Seligmann. Knowledge-based augmented reality. *Communications of the ACM*, 36(7):52–62, July 1993b.

S. Feiner, A. Webster, B. MacIntyre, and E. Keller. Architectural anatomy, 1995.

A. Fountain, W. Hall, I. Heath, and H. Davis. MICROCOSM: An open model for hypermedia with dynamic linking. In *European Conference on Hypertext*, pages 298–311, 1990.

A. Fuhrmann, H. Lffelmann, D. Schmalstieg, and M. Gervautz. Collaborative visualization in augmented reality. *IEEE Computer Graphics and Applications*, 18(4):54–59, 1998.

J. Gabbard. Researching usability design and evaluation guidelines for augmented reality (ar) systems, http://www.sv.vt.edu/classes/esm4714/student_proj/class00/gabbard/, 2000.

J. Gabbard and D. Hix. Taxonomy of usability characteristics in virtual environments. Technical report, Virginia Polytechnic Institute and State University, Final Report to the Office of Naval Research, November 1997.

J. Gabbard, K. Swartz, K. Richey, and D. Hix. Usability evaluation techniques: A novel method for assessing the usability of an immersive medical visualization ve, 1999.

HW. Gellersen, A. Schmidt, and M. Beigl. Multi-sensor context-awareness in mobile devices and smart artifacts. *Mobile Networks and Applications (MONET)*, (Oct), 2002.

G. Gordan, M. Billinghurst, M. Bell, J. Woodfill, B. Kowalik, A. Erendi, and J. Tilander. The use of dense stereo range data in augmented reality. In *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002)*, September 2002.

Michael Grafe, Raphael Wortmann, and Holger Westphal. Ar-based interactive exploration of a museum exhibit. In *Proceedings of The First IEEE Augmented Reality Toolkit Workshop*, September 2002.

J. Griffiths, D. Millard, H. Davis, D. Michaelides, and M. Weal. Reconciling versioning and context in hypermedia structure servers. In *To be published in Proceedings of the First Symposium on Metainformatics*, 2002.

R. Haines, E. Fischer, and T. Price. Head-up transition behavior of pilots with and without head-up display in simulated low-visibility approaches. Technical report, Technical Report: NASA Ames Research Center, Moffett Field, 1980.

F. Halasz. Reflections on notecards: Seven issues for the next generation of hypermedia systems. *Communications of the ACM*, 31(7):836–852, 1988.

F. Halasz and M. Schwartz. The dexter hypertext reference model. *Communications of the ACM*, 37(2):30–39, 1994.

Matthias Haringer and Holger T. Regenbrecht. A pragmatic approach to augmented reality authoring. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, September 2002.

Deborah Hix, J. Edward Swan II, Joseph L. Gabbard, Mike McGee, Jim Durbin, and Tony King. User-centered design and evaluation of a real-time battlefield visualization virtual environment. In *VR*, pages 96–103, 1999.

T. Hollerer, S. Feiner, and J. Pavlik. Situated documentaries: Embedding multimedia presentations in the real world. In *Proceedings of ISWC '99 (Third International Symposium on Wearable Computers), San Francisco, CA*, 1999.

Tobias Hollerer, Steven Feiner, Drexel Hallaway, Blaine Bell, Marco Lanzagorta, Dennis Brown, Simon Julier, Yohan Baillot, and et al. User interface management techniques for collaborative mobile augmented reality. *Computers and Graphics*, 25(5), October 2001.

International Documentation Committee International Council of Museums. Cidoc: Draft international core data standard for archeological sites and monuments, 1995.

N. Ioannidis, V. Vlahakis, J. Karigiannis, A. Demiris, D. Stricker, and T. Gleue. Hybrid image processing techniques for the reconstruction of lost worlds and the revival of ancient life. In *DPS in Arts Session, 14th International Conference on Digital Signal Processing*, July 2002.

Hiroshi Ishii and Brygg Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of the Conference on Human Factors in Computing Systems CHI 1997*, pages 234–241, 1997.

J J. Preece, Y. Rogers, and H. Sharp. Interaction design: Beyond human-computer interaction, 2002.

T. Jebara, C. Eyster, J. Weaver, T. Starner, and A. Pentland. Stochasticks: Augmenting the billiards experience with probabilistic vision and wearable computers. In *In Proceedings of the International Symposium on Wearable Computers*, October 1997.

T. Jebara, B. Schiele, N. Oliver, and A. Pentland. Dypers: Dynamic personal enhanced reality system. Technical Report 463, MIT Media Laboratory Perceptual Computing, 1998.

H. Kato. Real-time tracking system based on matching templates generated from image texture. In *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002)*, September 2002.

H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, 1998.

H. Kato, M. Billinghurst, R. Blanding, and R. May. *ARToolKit Manual*, 1999.

H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top ar environment. In *Proceedings of the International Symposium on Augmented Reality (ISAR 2000)*, October 2000.

H. Kaufmann, D. Schmalstieg, and M. Wagner. Construct3d: A virtual reality application for mathematics and geometry education. *Virtual Reality in Education special issue of Education and Information Technologies*, 2000.

S. Kim, H. Alani, W. Hall, P. Lewis, D. Millard, N. Shadbolt, and M. Weal. Artequakt: Generating tailored biographies with automatically annotated fragments from the web. In *Proceedings of the SAAKM Workshop at ECAI*, June 2002.

D. Kirsh. The intelligent use of space. *Artificial Intelligence*, 73(1):31–68, 1995.

B. Koleva. Mixed realities - http://www.crg.cs.nott.ac.uk/research/technologies/mixed, 1999.

Florian Ledermann. Vsml - an integration tool to bridge the various components of the vs software framework, http://www.studierstube.org/virtualshowcase/, 2002.

B. MacIntyre and S. Feiner. Language-level support for exploratory programming of distributed virtual environments. In *In Proceedings of ACM Symp. on User Interface Software and Technology (UIST '96)*, 1996.

R. Mack and J. Nielsen. Executive summary. *Usability Inspection Methods*, pages 1–23, 1994.

K. Malcolm, S. Poltrock, and D. Schuler. Industrial strength hypermedia: Requirements for a large engineering enterprise. In *Proceedings of the 91 ACM Conference on Hypertext*, pages 13–24, 1991.

S. Malik, C. McDonald, and G. Roth. Hand tracking for interactive pattern-based augmented reality. In *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2002)*, September 2002.

S. Mann. 'mediated reality'. TR 260, M.I.T. Media Lab Perceptual Computing Section, Cambridge, Massachusetts, http://wearcam.org/mr.htm, 1994.

S. Mann. Wearable computing: A first step toward personal imaging. *IEEE Computer*, 30(2), 1997.

Danius T. Michaelides, David E. Millard, Mark J. Weal, and David De Roure. Auld leaky: A contextual open hypermedia link server. In *OHS-7/SC-3/AH-3*, pages 59–70, 2001.

P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems (Special Issue on Networked Reality)*, E77-D(12): 1321–1329, December 1994.

D. Millard and H. Davis. Navigating spaces: The semantics of cross-domain interoperability. In *Proceedings of the 2nd Workshop on Structural Computing, ACM Hypertext 2000 Conference*, 2000.

D. Millard, L. Moreau, H. Davis, and S. Reich. FOHM: a fundamental open hypertext model for investigating interoperability between hypertext domains. In *Proceedings of the Eleventh ACM Conference on Hypertext and Hypermedia HT'00*, pages 93–102, 2000.

N. Minar, M. Gray, O. Roup, R. Krikorian, and P. Maes. Hive: Distributed agents for networking things. In *Proceedings of ASA/MA'99, the First International Symposium on Agent Systems and Applications and Third International Symposium on Mobile Agents*, 1999.

Media Lab MIT. Tangible media group - http://tangible.www.media.mit.edu/groups/tangible.

L. Moreau, N. Gibbins, D. De Roure, S. El-Beltagy, W. Hall, G. Hughes, D. Joyce, S. Kim, D. Michaelides, D. Millard, S. Reich, R. Tansley, and M. Weal. An agent framework for distributed information management. In *In The Fifth International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents 2000*, 2000.

N. Mourkoussis, F. Liarokapis, J. Darcy, M. Pettersson, P. Petridis, P. Lister, and M. White. Virtual and augmented reality applied to educational and cultural heritage domains. In *Business Applications of Virtual Reality (BAVR)*, April 2002.

NaN. Not a number (nan) corporation, http://www.blender.nl.

N. Navab, B. Bascle, M. Appel, and E. Cubillo. Scene augmentation via the fusion of industrial drawings and uncalibrated images with a view to marker-less calibration, 1999.

T. Nelson. Getting it out of our system. In G. Schechter, editor, *In Information Retrieval: A Critical Review*, pages 191–210. Thompson Books, Washington, D.C., 1967.

T. Nelson. As we will think. In *Proceedings of Online 72 Conference, Brunel University, Uxbridge, England*, 1972.

U. Neumann and A. Majoros. A. cognitive, performance, and systems issues for augmented reality applications in manufacturing and maintenance. In *IEEE VRAIS '98*, 1998.

E. Not, D. Petrelli, O. Stock, C. Strapparava, and M. Zancanaro. Augmented space: Bringing the physical dimension into play. In *8th ACM International Hypertext Conference*, 1997a.

E. Not, D. Petrelli, O. Stock, C. Strapparava, and M. Zancanaro. Person-oriented guided visits in a physical museum. In *4th International Conference on Hypermedia and Interactivity in Museums [ICHIM'97]*, pages 162–172, 1997b.

T. Ohshima, K. Sato, H. Yamamoto, , and H. Tamura. Rv-border guards: A multi-player mixed reality entertainment. *Transactions of VRSJ*, 4(4):699–705, 1999.

T. Ohshima, K. Satoh, H. Yamamoto, and H. Tamura. Ar2 hockey system: A collaborative mixed reality system. *Transactions of VRSJ*, 3(2):55–60, 1998.

S. Parker, P. Lafferty, and S. Setford. *How Things Work*. Parragon Books, 2000.

I. Poupyrev, R. Berry, J. Kurumisawa, K. Nakao, L. Baldwin, I. Shigeo, M. Billinghurst, and H. Kato. Augmented groove. In *Proceedings of the ACM SIGGRAPH 2000 Conference Abstracts and Applications*, 2000a.

I. Poupyrev, M. Billinghurst, H. Kato, D. Tan, and H. Regenbrecht. A tangible augmented reality interface for prototyping aircraft instrument panels. In *In Proceedings of International Symposium of Augmented Reality 2000*, 2000b.

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

H. Pryor, T. Furness, and E. Viirre. The virtual retinal display: A new display technology using scanned laser light. In *Proceedings of Human Factors and Ergonomics Society, 42nd Annual Meeting*, pages 1570–1574, 1998.

QRP. Quantum research group, http://www.qprox.com, 2001.

G. Reitmayr and D. Schmalstieg. Opentracker - an open software architecture for reconfigurable tracking based on xml. In *Proceedings of the IEEE VR 2001 conference*, March 2001.

J. Rekimoto. Tilting operations for small screen interfaces. In *Proceedings of UIST'96*, 1996a.

J. Rekimoto. Transvision: A hand-held augmented reality system for collaborative design. In *Proceedings of Virtual Systems and Multimedia (VSMM) '96*, 1996b.

J. Rekimoto. Navicam: A magnifying glass approach to augmented reality systems. *Presence: Teleoperators and Virtual Environments*, 6(4):399–412, 1997.

J. Rekimoto and Y. Ayatsuka. Cybercode: Designing augmented reality environments with visual tags. In *In Proceedings of Designing Augmented Reality Environments (DARE 2000)*, 2000.

Jun Rekimoto and Katashi Nagao. The world through the computer: Computer augmented interaction with real world environments. In *ACM Symposium on User Interface Software and Technology*, pages 29–36, 1995.

Jun Rekimoto, Brygg Ullmer, and Haruo Oba. Datatiles: a modular platform for mixed physical and graphical interactions. In *Proceedings of the Conference on Human Factors in Computing Systems CHI 2001*, pages 269–276, 2001.

B. Rhodes. The wearable remembrance agent: A system for augmented memory. *Personal Technologies Journal Special Issue on Wearable Computing, Personal Technologies*, 1997(1), 1997.

B. Rhodes. Wimp interface considered fatal, 1998.

B. Rhodes, N. Minar, and J. Weaver. Wearable computing meets ubiquitous computing: Reaping the best of both worlds. In *Proceedings of The Third International Symposium on Wearable Computers (ISWC '99)*, 1999.

A. Rizk and L. Sauter. Multicard: An open hypermedia system. In *ECHT 92. Proceedings of the ACM conference on Hypertext*, pages 4–10, 1992.

J. Rolland, D. Wright, and A. Kancherla. Towards a novel augmented reality tool to visualize dynamic 3d anatomy, 1997.

E. Rose, D. Breen, K Ahlers, D. Greer, C. Crampton, M. Tuceryan, and R. Whitaker. Annotating real-world objects using augmented vision. In *Computer Graphics International '95*, 1995.

M. Sarini and C. Strapparava. Building a user model for a museum exploration and information - providing adaptive system. In *The Ninth ACM Conference on Hypertext and Hypermedia*, 1998.

B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, 1994.

D. Schmalstieg, A. Fuhrmann, and G. Hesina. Bridging multiple user interface dimensions with augmented reality. Technical Report TR-186-2-00-15, Vienna University of Technology, 2000a.

D. Schmalstieg, A. Fuhrmann, G. Hesina, Zs. Szalavari, L. M. Encarnao, M. Gervautz, and W. Purgathofer. The studierstube augmented reality project. Technical Report TR-186-2-00-22, Vienna University of Technology, December 2000b.

Patrick Sinclair and Kirk Martinez. Adaptive hypermedia in augmented reality. In *in Proceedings of the 3rd Workshop on Adaptive Hypertext and Hypermedia Systems, ACM Hypertext 2001 Conference*, 2001.

T. Starner, S. Mann, B. Rhodes, J. Levine, J. Healey, D. Kirsch, R. Picard, and A. Pentland. Augmented reality through wearable computing. *Presence*, 6(4):386–398, August 1997.

A. State, G. Hirota, D. Chen, W. Garrett, and M. Livingston. Superior augmented-reality registration by integrating landmark tracking and magnetic tracking. In *Computer Graphics Proceedings, Annual Conference Series 1996, ACM SIGGRAPH*, 1996.

I. Sutherland. A head-mounted three dimensional display. In *Proceedings of Fall Joint Computer Conference*, 1968.

Z. Szalavri, E. Eckstein, and M. Gervautz. Collaborative gaming in augmented reality. In *Proceedings of VRST'98*, pages 195–204, November 1998.

Z. Szalavri and M. Gervautz. The personal interaction panel - a two-handed interface for augmented reality. In *Proceedings of EUROGRAPHICS'97*, September 1997.

A. Tang, C. Owen, F. Biocca, and W. Mou. Comparative effectiveness of augmented reality in object assembly. In *Proceedings of ACM CHI 2002*, April 2002a.

A. Tang, C. Owen, F. Biocca, and W. Mou. Experimental evaluation of augmented reality in object assembly task. In *Proceedings of ISMAR 2002, IEEE and ACM International Symposium on Mixed and Augmented Reality*, September - October 2002b.

M. Tzagarakis, N. Karousos, D. Christodoulakis, and S. Reich. Naming as a fundamental concept of open hypermedia systems. In *Proceedings of the eleventh ACM on Hypertext and hypermedia table of contents*, pages 103 – 112, 2000.

V. Vlahakis, N. Ioannidis, J. Karigiannis, M. Tsotros, M. Gounaris, D. Stricker, T. Gleue, P. Daehne, and L. Almeida. Challenges and solutions of a personalized augmented reality guide for archaeological sites. *Computer Graphics in Art, History and Archaeology: Special Issue of the IEEE Computer Graphics and Applications Magazine*, 22(5), Sept.-Oct 2002.

W3C. World wide web consortium, http://www.w3c.org.

J. Waldo. Jini architectural overview - technical white paper, 1998.

M. Weal, D. Millard, D. Michaelides, and D. De Roure. Building narrative structures using context based linking. In *Proceedings of Hypertext 2001*, 2001.

Web3D. Web3d consortium, http://www.web3d.org.

S. Weghorst. Augmented reality and parkinson's disease. *Communications of the ACM*, 40(8):47–48, 1997.

M. Weiser. The computer for the 21st century. *Scientific American*, 265(3), 1991.

P. Wellner, W. Mackay, and R. Gold (Editors). Computer augmented environments: Back to the real world. *Special Issue of Communications of the ACM*, 37(7):24–26, July 1993.