Augmented Reality Interface Toolkit

Fotis Liarokapis, Martin White, Paul Lister
University of Sussex, Department of Informatics
{F.Liarokapis, M.White, P.F.Lister}@sussex.ac.uk

Abstract

This paper proposes a high-level augmented reality interface toolkit that allows the combination of audio-visual information with a real world environment in an easy and interactive way. The system is based on MFC libraries, OpenGL, OpenAL, Microsoft Vision SDK and the vision tracking libraries from the well known ARToolKit. Simple and cost effective hardware complements the software solution. This AR interface toolkit can be used as an exemplar for the development of other applications. Users can interact with the presented information in several different ways. Realistic augmentation is also supported such as soft and hard shadows, without sacrificing the overall efficiency of the system. To illustrate the feasibility of our AR interface toolkit a cultural heritage application for museum environments is briefly presented.

Keywords--- **Augmented Reality, Real-time Interfaces, Human-Computer Interaction.**

1. Introduction

One of the reasons that computers are becoming more important tools in our everyday life is because the user interfaces are becoming easier to use and operate. In addition, new human computer interaction techniques have been developed offering a greater degree of freedom [1] compared with the traditional windows style interfaces. However, to maximise performance, most interfaces are designed for particular applications [2] and the designers often need to redesign the system in order to apply them into other applications.

On the other hand, augmented reality (AR) is an increasingly important and promising area of computer graphics, vision and user interface design. Technically, AR is not a single technology but a combination of several technologies with the aim of enhancing a user's perception [3] and interaction with the real world. The ideal AR system must be able to mix computer-generated information with the real world in real time in such a way where the user cannot understand the difference.

Even if a lot of work has been done AR still has a long way to go before it can be applied to commercial applications. One of the major constraints of AR is its lack of ability to allow participants to effectively control the computer generated information in the AR environment. To reduce the complexity of human-computer interactions in augmented environments, specific input interfaces can be employed such as a SpaceMouse [4] or InertiaCube [5] and so on. Through these types of interfaces users can easily control the augmentation (e.g. transformation of a virtual object in the real world). Nevertheless, the design and implementation of a generic AR application is not a trivial task.

During the last decade, a number of different types of AR systems have been developed. Here some of the most important are discussed. One of the earliest supported a full X11 server [6] with three different types of windows including surround-fixed, display-fixed and world-fixed windows. Another experimental system [7] extended interactions from a traditional desktop paradigm to investigate a range of issues related to the rapid assembly and deployment of adaptive visualisation systems.

Furthermore, the use of tangible interfaces allows users to naturally interact with the computer-generated information. A tangible AR interface is the MagicBook [8] which uses a real book to transfer users from reality to virtuality. Virtual objects are superimposed on the pages of the book and users can interact with the augmented scene. Another example of an AR tangible interface [9] is a tabletop system designed for virtual interior design. Multiple users can interact with virtual furniture, and manipulate the computer-generated objects. Another well known system, the Studierstube Personal Interaction Panel (PIP) [10] provides the user with a blank physical board, on which virtual controls are drawn.

To improve the interface design human computer interaction techniques must be also considered. An efficient AR toolkit should provide a combination of various aspects including voice; gesture; animation; and persistent data [11] respectively. Collaboration is another issue that plays an important role. Collaborative hybrid systems have been designed in the past [12] to combine 3D widgets and tracked displays or input devices. Another example is a wearable stereo AR platform [13] that uses a pen and a pad interface to support 3D manipulations of virtual information in the near field.

In this paper, a user-focused interface environment based on AR see-through display technology called Mixed Reality Interface Toolkit (MRIT) is proposed. The AR interface aims in assisting the user to generate realistic augmented reality environments quickly. The architecture of the system is designed to be as general as

Proceedings of the Eighth International Conference on Information Visualisation (IV'04)
1093-9547/04 $ 20.00 IEEE

possible so that it can be easily adapted to many application domains. Different graphics algorithms for presenting a realistic augmentation of computer-generated information have been applied. To evaluate the potential of the system, an application scenario specifically designed for museum environments is presented. Other scenarios that could be used in practice include applications such as: interior design, medical, education and in general any type of indoor learning and training system suitable for the table-top.

2. System Architecture

The software and hardware technologies used for the specification of the system, are chosen on the basis of producing a cheap and robust AR interface toolkit.

In terms of software components two types of tools are used: content production tools such as 3ds max, Photoshop for image preparation, and so on; and software libraries used to develop the actual AR Interface Toolkit. The content production tools are necessary for the preparation of the computer-generated information into a suitable format for visualisation in AR. Other content production functionality include: basic modelling with 3ds max; audio capturing and processing using Cool Edit Pro; and camera calibration based on Matlab.

The AR Interface Toolkit uses API libraries for video operations based on Microsoft Vision SDK and fiducial based tracking [14] drawn from ARToolKit. In addition, computer graphics and sound functions are coded in OpenGL and OpenAL [15] APIs respectively. The above technologies are integrated into a high level C++ framework. Finally, an MFC windows application was designed in order to connect the various components and to produce the final AR Interface Toolkit software environment.

The AR Interface Toolkit is configured for two display methods: a monitor based system, and a video see-through approach. The monitor-based approach is cost-effective and provides less immersion than the video see-through configuration. On the other hand, the video see-through configuration does not provide a high-resolution visualisation since the HMD used supports 800×600 pixel resolution. However, it can be more robust as an AR tool because it fully immerses the user.

The current configuration of the system can support up to eight display types, not just monitors. In the simplest case, standard CRT monitors can be used to reduce the cost. However, in a more immersive environment the monitors may be replaced with HMDs or even have a combination of both. The objective of the hardware architecture is to present the final visualisation amongst a range of displays including see-through HMDs, plasma screen or additional flat panel displays as shown in Figure 1.

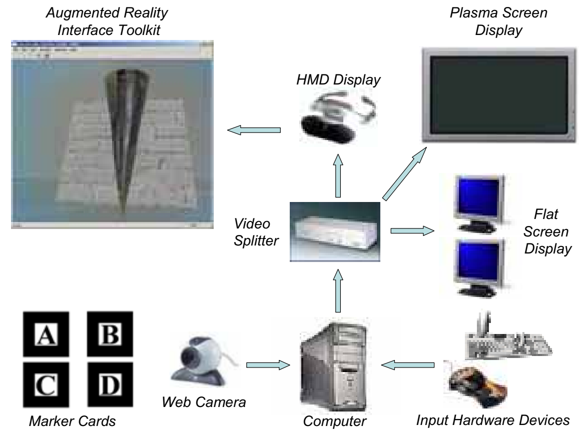

Figure 1 Operation of the system

In this work, the most complex configuration includes two HMDs, five monitors and a plasma screen display. Different types of input hardware devices such as standard I/O components (i.e. keyboard, mouse) and VR hardware (i.e. SpaceMouse) together with physical marker cards provide multiple means of interaction. The output of the computer is sent to the VGA splitter (Figure 1) that permits many different display technologies to be used. This allows a simple form of collaboration between users by displaying virtual information in AR to multiple users. In the immersive configuration, each participant can connect their HMD to the video splitter and interact in an immersive environment with the augmented information. Multiple users can then see the same AR information and collaborate with the AR application.

3. Augmented Reality Interface

During the last decade many tracking systems have been built based upon computer vision techniques. The main objective of the video see through AR interface is to realistically render scenes as efficient as possible based on the convergence of computer graphics and computer vision techniques. The requirements for an effective AR interface system are based on the ability to perform real time video capturing, high-speed image processing [16] and realistic rendering and manipulation. In brief, our system is designed on the basis of the following criteria:

- Using an object oriented (OO) programming style so that the system can be easily reusable and extensible;
- Independent of the rendering context which allows the mixing of various types of digital information;

To further explore the library structure of the AR platform, four basic layers of hierarchical abstraction are designed as illustrated in Figure 2 below.

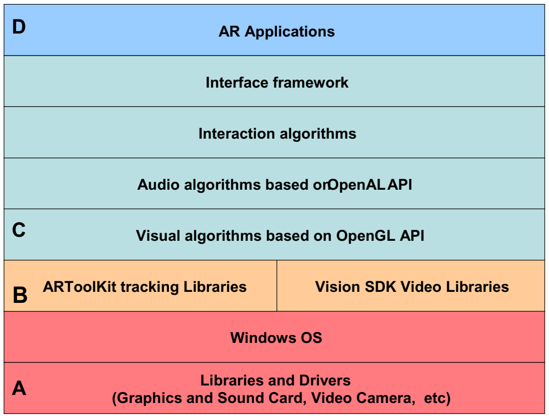

Figure 2 Layers of AR Platform's libraries

The lower level (Figure 2) specified as section A, consists of libraries like graphics, sound, video and drivers (i.e. for the monitor, the graphics card, the sound, the video camera etc). These are needed in order for the system to operate properly and they can be easily installed manually. This layer also includes the OS (MS windows) on which the system is designed on.

The second layer (section B) consists of the ARToolKit and Vision SDK's software libraries used for tracking and video operations respectively. Both libraries have been re-designed and incorporated into the implementation framework using an OO style.

The layers in section C contain the four main functional sub-layers that briefly describe the technical capabilities of the system. They include a number of visual algorithms used for the rendering of 3D models, 2D images and textual information in the AR environment. The second sub-layer is responsible for the audio augmentation based on OpenAL's API, which has a very similar structure to OpenGL. The third sub-layer offers a smooth and fast manipulation option to the users based on quaternion rotations. The final part describes an interface framework which allows communication between all the described layers and the user.

Finally, the top layer (section D) offers a number of challenging AR applications based on the functionality and tools implemented on the toolkit. A user adopting the AR Interface Toolkit can easily build new applications by either incorporating the existing functionality or by extending it. An example application for museum environments is illustrated in section 6.

4. Audio-Visual Augmentation

A high-level audio-visual AR platform has been developed to assist user's interaction with digital information in order to maximize their knowledge and understanding. The digital information consists of multimedia content information [17] including: 3D representations of real objects; photographs, pictures, diagrams and icons; textual annotations; and 3D sound. For our testing purposes, a tabletop environment has been setup consisting of two flat white boards joined together to form a perpendicular angle.

4.1 Realistic object augmentation

Realistic visualization is an area of continuous development where a high degree of realism could be produced in real time if tremendous computational power is available. However, when designing an interactive real-time system it is important to compromise between realism and efficiency. The selection of the most appropriate 3D format for visualisation is a crucial task in order to achieve a high level of realism in the system. The most common 3D file format is 3ds and it is supported by most commercial 3D software packages. The main advantage of 3ds file format is that it contains sufficient information for a realistic representation of the 3D scene and it is also easy to parse.

In addition, using OpenGL's virtual lighting, fast realistic shadows can be generated in real-time. Nevertheless, the matching of the real light with the virtual would require much more processing power and so is impractical to implement. The generation of realistic shadows plays a crucial factor for the generation of a realistic scene in an AR system. In reality, all objects have their shadows so in an ideal system virtual shadows should be cast into real objects and real shadows into virtual objects. In theory, shadows can be distinguished [18] in two broad categories: planar and curved surface shadows.

To generate planar augmented shadows, an algorithm that creates a shadow projection matrix is defined. This matrix is based on the plane equation coefficients and the position of the virtual light. Initially, a plane needs to be defined by specifying three points in space. From these points two vectors can be defined, and the cross product of the two vectors can be easily calculated. An illustration of this is shown in Figure 3 below:

Figure 3 Example of augmented shadow

This shadow algorithm is very fast and produces accurate and realistic results when applied to complex objects. However, it does not account for the generation of soft shadows but can be roughly estimated by calculating multiple shadow projection matrices instead of one. The modifying algorithm calculates apart from the matrix representing the virtual hard shadow other matrices representing the virtual soft shadows. The flaw of this technique is that depending on the complexity of the augmented information the overall performance of the system can be sacrificed to certain extent.

Other 3D effects that can be applied effectively and do not sacrifice efficiency are transparency and fog. Transparency is an effect that can be applied very successfully [9] in some types of AR application scenarios. The 3D models used in the visualisation contain transparent surfaces so that in the rendering part, transparent objects can be created through the alpha blending mechanism. Using the OpenGL functionality the alpha channel (α) of the colour mode (RGBA) can be controlled to limit the amount of light that penetrates through surfaces (opacity).

Fog is another graphics effect that can be used to create the illusion of partially transparent space between the camera and the 3D object. This effect can be easily implemented by blending in the environment a distance dependent colour that is defined as fog factor f, using an atmosphere intensity attenuation function. In this way, the rendered information can be simulated through a hazy or smoky atmosphere. OpenGL supports three types of fog densities: linear, exponential and Gaussian and all of them can be applied using the toolbar menu.

4.2 Image Augmentation

Images are widely used as a means to increase realism based on a technique known as texture mapping. In the past, pictures have been used with success to help people comprehend information more effectively than text or auditory instructions [19] and for communicating. The augmentation of images is a highly cost effective means to present simple 2D information in real world. The use of their operation may be performed in a number of different ways depending on the application scenario. The most obvious cases can be categorised as descriptive symbolic, iconic and functional.

Descriptive images are the most popular as they refer to situations where a scene can be described or the image itself can tell a self-explanatory story. Symbolic images identify a basic underlying principle or symbol and usually allow both simple and complex symbolism and whose interpretation can change over time. Iconic image representations try to identify a case of a multinational meaningful icon that is not related to a specific language (i.e. English). Furthermore, with functional images a single (or multiple) operation can be expressed or supported.

In our system, descriptive images are used to explain a 3D real object. Symbolic augmentations are utilised to express a rough representation of a scenario or an object. Iconic images are used to shown directions or other useful annotations. Finally, functional images act as virtual buttons and a specific operation is assigned on them.

4.3 Annotations

Annotations are an important issue for the effectiveness of the AR interface because it allows assigning textual annotations with respect to 3D representations of real objects. The textual annotations can be augmented in two different ways depending on the presentation style, which can be either a label or a detailed description of the digital information. Label text has been used in the past [20] [21] to point out specific parts of a complex system using the minimum textual information. As the pose changes, the label transforms respectively with the 3D model. In this case, the most important aspect is to ensure that the augmented labels do not obscure each other and that the information is clearly visible to the user.

The descriptive text provides complete textual information about an object or an operation but the text always stays aligned on the screen so that it can be easily readable. This is achieved by parsing text files that included detailed information about the 3D object. The main obstacle occurred when the text file was very big and the textual information could not fit into the screen.

To overcome this problem, depending on the size of the window the maximum amount of textual information can be wrapped. When the information does not fit into the window, then it is broken into parts and the user can control the visualisation of the text using a page-up and page-down technique. A more sophisticated solution would include the design of a 'clever' text parser that could 'measure' the text size so that the maximum information can fit into the visualisation window.

4.4 Audio Augmentation

Most AR applications have not incorporated a 3D sound component although it is argued that it can lead to strong AR illusions [22] and therefore contribute to the sense of immersivity. As a result, when sound component is absent from a system, then participants can easily become isolated from the environment. The most important issue when designing 3D sound is to 'see' [23] the sound source. This will give to the participants the psychological impression that the sound source exists.

Our augmented sound methodology has some similarities with the Augmented Sound Reality (ASR) approach [24] but the AR interface toolkit is based on OpenAL API which is easier to implement and provides sufficient functionality. The implemented 3D audio system is capable of loading and mixing music sound files. Wave sound sources are digitally processed and transformed into the real environment using again the marker cards. The user can move the sources in 3D space using either the keyboard, the menu toolbar or by simply manipulating the marker card.

A distinctive feature of the 3D audio system is the multiple augmentations of wave sound sources in real-time. This is an option that can be extremely useful for simulating AR scenarios that require surround audio. The sound files may be overlaid into the same marker, or into a different marker, depending on the needs of the application.

5. Human-Computer Interaction Techniques

Human-computer interactions are one of the most important issues when designing any visualisation system and especially when developing a generic AR interface. Thereby, interactions in the system have to be performed in a natural way so that inexperienced users can become familiar quickly with the augmented reality environment. The AR Interface Toolkit allows the combination of three existing techniques including the use of the *toolbar menu*, the use of *input hardware devices* and the *physical manipulation* of the marker cards.

The toolbar menu allows the user to have full control of the visualisation. To register computer generated information into a marker card, the user has to select from the toolbar menu which type of digital information (3d model, image, text, video and audio) wants to visualise and then click on it (Figure 4). The corresponding information will be correctly aligned to the marker card.

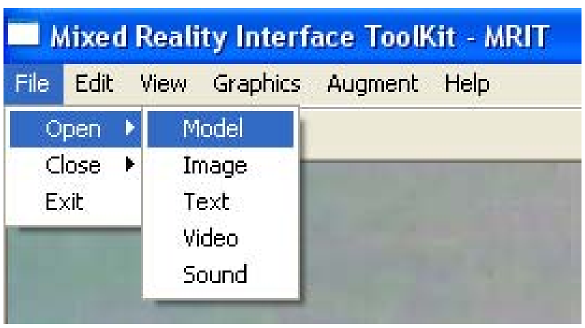

Figure 4 Interface menu toolbar

Other functionality available to the user includes major graphics operations that are described in the following section. Digital information can not be moved below an invisible ground defined by the position of marker cards. This gives full power to the users to control the registration of virtual information.

The second method is based on popular interaction input devices like the keyboard and the mouse and also on the sophisticated SpaceMouse device. In both the keyboard and the SpaceMouse, predefined keyboard controls (hot keys) offer the same functionality as with the toolbar menu. This has the advantage of changing the parameters of the virtual objects faster, i.e. change lighting condition; texturing information; switch from solid mode to wireframe mode. Using the SpaceMouse users can zoom, pan and rotate the rendered information in six degrees-of-freedom (DOF) in a natural way using only one hand.

Lastly, users can physically examine the augmented information similarly to [9] by manipulating the marker cards in the real environment. In this way, the virtual objects can be observed in a natural way. This it becomes feasible for the inexperienced user to experiment with various scenarios to perceive a different perception of the visualisation.

6. Museum Application

A cultural heritage application has been developed to experimentally explore the potentials of AR in a museum environment. The system illustrates how museums can use the AR interface to visualise cultural artefacts. From an individual's point of view, with no previous technical experience, this application can be considered to provide extra knowledge with regards to the history, meaning and significance of cultural heritage. Therefore, the application has a wider scope in educating the person that uses it. Our experiences gained in developing this system has been applied to an EU funded project [25] called ARCO.

One of the greatest advantages of the application is its ability to exhibit a large number of artefacts in a limited space (i.e. table-top environment). To provide the reader with an example of the AR visualisation in an immersive environment, Figure 5 presents a 3D representation of an old spice jar overlaid in a predefined marker.

Figure 5 Spice Jar

The virtual spice jar can be manipulated in six DOF so that the user can examine the physical aspects of the artefact in great detail. In addition, the menu toolbar can be accessed by the user according to his/her preferences. For instance, graphics operations like shadows, transparency, shading, etc., can be changed by just clicking on specific menu options of the toolbar—such menu options could be transposed to another input device, e.g. keyboard or SpaceMouse. Another useful option is that the lighting conditions of the virtual artefacts can be simply monitored through the use of virtual lights. To enhance further the visualisation process, more media information can be added into the tabletop environment by following the same procedure, as illustrated in Figure 6.

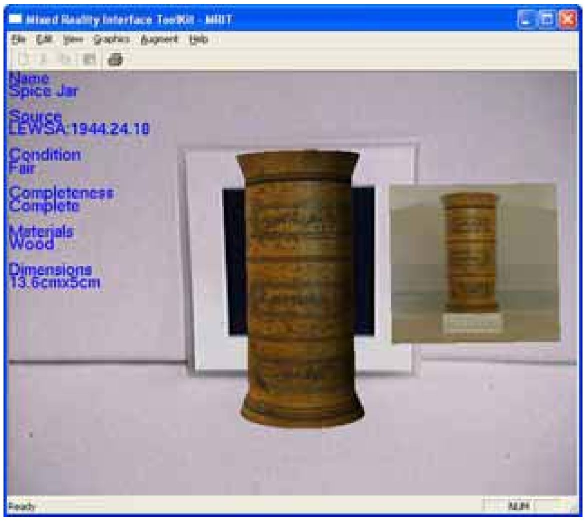

Figure 6 Spice Jar with media information

The purpose of Figure 6 is to show the various options that the system offers. In particular, the 3D object can be augmented, while at the same time relevant to the object textual annotation and real image are added. In contradiction to Figure 5 where lighting information is retrieved from the 3ds file, in this example it is manually applied from the menu toolbar. One of the important goals of the archaeological application is to present museum artefacts in an attractive manner so that would make museum visitors, especially children, more interested in cultural heritage

7. Conclusions and Future Work

The work presented in this paper illustrates the potentials of how AR interface technology could be used as a pilot for designing new types of AR table-top applications. Our system differs from existing methods because it provides users with complete control of an audio-visual augmentation and at the same time allow interactions within the real environment in three different ways: first, by the natural manipulation of the specifically designed marker cards in 3D space; secondly, by using the menu bar that contains a list of functions; and lastly, through the use of input hardware devices.

The performance of the system is measured to be in the range between 15 and 25 frames-per-second (fps) depending on the complexity of the scene. The end user latency of the AR interface system is considered to be analogous to VR systems, since the digital information is delayed appropriately.

In terms of robustness, the only flaw of the AR system occurs when the camera fails to detect the marker card. This causes the computer-generated information to disappear from their sight of view. This is due to the fact that ARToolKit's tracking algorithms are not very effective and produce registration errors. A way to improve this is to calibrate the camera model more accurately than ARToolKit does using a calibration tool [26] written in Matlab. This tool provides a way of computing distortion parameters by computing the camera parameters.

It is worthwhile mentioning, that the implications of completely immersive AR systems must include other forms of augmentation such as virtual touch. In the future we will integrate haptic devices (i.e. virtual gloves) to provide tactile feedback. Further research is required to improve the usability of the system according to human factors. The design of a user-centred interface should take into consideration physiological aspects like measuring the long-term effects of working with AR environments.

8. Acknowledgements

This research was partially funded by the EU IST Framework V programme, Key Action III-Multimedia Content and Tools, Augmented Representation of Cultural Objects (ARCO) project IST-2000-28336.

9. References

[1] Rekimoto, J., Nagao, K. The World through the Computer: Computer Augmented Interaction with Real World Environments, Proc of UIST '95, (ed. B.A. Myers), ACM Press, Pennsylvania, pp 29-36, 1995.

[2] Curtis, D., Mizell, D., et al., Several Devils in the Details: Making an AR Application Work in the Airplane Factory, Proc International Workshop Augmented Reality '98 (IWAR'98). San Francisco, 1 Nov. 1998, pp. 47-60, 1999.

[3] Azuma, R. T., A survey of augmented reality, Presence: Teleoperators and virtual environments, vol. 6, 1997, pp. 355- 385.

[4] SpaceMouse PlusXT, Available at [http://www.logicad3d.com/products/PlusXT.htm], Accessed at 28/02/2004

[5] InterSense InertiaCube^2, Available at [http://www.isense.com/products/prec/ic2/InertiaCube2.pdf], Accessed at 28/02/2004

[6] Feiner, S., MacIntyre, B., et al. Windows on the World: 2D Windows for 3D Augmented Reality, Proc of ACM Symposium on User Interface Software and Technology, Atlanta, November 3-5, Association for Computing Machinery: 145-155, 1993.

[7] Slay H., Phillips M., Vernik R., Thomas B., Interaction Modes for Augmented Reality Visualization. Appeared on the Australian Symposium on Information Visualization, Sydney, December, 2001.

[8] Billinghurst, M., Kato, H., Poupyrev, I., The MagicBook: A Traditional AR Interface. Computer and Graphics, 25, 745-753, 2001.

[9] Kato, H., Billinghurst, M., et al., Virtual Object Manipulation on a Table-Top AR Environment, Proc. International Symposium on Augmented Reality 2000, Munich, 5-6 Oct., pp. 111-119, 2000.

[10] Szalavári, Zs., Gervautz, M., The personal interaction panel - a two-handed interface for augmented reality,. Proc. 18th EUROGRAPHICS. Budapest, 4-8 Sept. 1997. pp. 335-346.

[11] Grasset R., Gascuel J-D., MARE: Multiuser Augmented Reality Environment on table setup. ACM SIGGRAPH Conference Abstracts and Applications, 2002.

[12] Butz, A., Höllerer, T., Feiner, S., MacIntyre, B., Beshers, C., Enveloping Users and Computers in a Collaborative 3D Augmented Reality, In Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality '99, October 20-21, San Francisco 1999.

[13] Reitmayr, G., Schmalstieg, D., A Wearable 3D Augmented Reality Workspace. Fifth International Symposium on Wearable Computers (ISWC'01), October 8 -9, 2001.

[14] Kato, H., Billinghurst, M., Poupyrev, I., ARToolkit User Manual, Version 2.33, Human Interface Lab, University of Washington, November 2000.

[15] OpenAL Specification and Reference, Version 1.0 Draft Edition, Published June 2000, Available at: [http://www.openal.org/home/]

[16] Uchiyama, S., Takemoto, K., et. al. MR Platform: A Basic Body on Which Mixed Reality Applications Are Built, In Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR'02), IEEE, 2002.

[17] Liarokapis, F., Petridis, P., et al., Multimedia Augmented Reality Interface for E-Learning (MARIE)', World Transactions on Engineering and Technology Education, UICEE, Vol.1, No.2, 173-176, 2002.

[18] Möller, T., Haines E., Book Title: Real-Time Rendering, A K Peters, Ltd, pp 374, 1999

[19] Najjar, L.J., Principles of educational multimedia user interface design, In R. W. Swezey & D.H. Andrews (Eds.), Readings in training and simulation: A 30-year perspective (pp. 146-158). Santa Monica, CA: Human Factors and Ergonomics Society

[20] Klinker, G., Ahlers, et al. Confluence of Computer Vision and Interactive Graphics for Augmented Reality, PRESENCE: Teleoperations and Virtual Environments, Special Issue on Augmented Reality, Vol. 6, No. 4, pp. 433-451, August 1997.

[21] Sinclair, P and Martinez, K., Adaptive Hypermedia in Augmented Reality, Proceedings of Hypertext, 2001.

[22] Goudeseune, C., Kaczmarski, H., Composing Outdoor Augmented-Reality Sound Environments, Proc. ICMC, Havana, 2001.

[23] Yewdall, D., Practical art of motion picture sound, Boston, MA: Focal Press, 1999.

[24] Dobler, D., Haller, M., Stampfl, P., ASR – Augmented Sound Reality, ACM SIGGRAPH 2002 Conference Abstracts and Applications, San Antonio Texas, pp. 148, 2002.

[25] White, M., Liarokapis, F., et al, A lightweight XML driven architecture for the presentation of virtual cultural exhibitions (ARCOLite), IADIS International Conference of Applied Computing 2004, Lisbon, Portugal, 23-26 March 2004

[26] Camera Calibration Toolbox for Matlab, Available at [http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/ref.html], Accessed at 01/02/2004.

Proceedings of the Eighth International Conference on Information Visualisation (IV'04)
1093-9547/04 $ 20.00 IEEE