Regenbrecht, H., Wagner, M., & Baratoff, G. (2002). MagicMeeting - a Collaborative Tangible Augmented Reality System. Virtual Reality - Systems, Development and Applications, Vol. 6, No. 3, Springer, 151-166.

MagicMeeting - a Collaborative Tangible Augmented Reality System

Holger T. Regenbrecht[a] **Michael T. Wagner[b]** **Gregory Baratoff[a]**

[a]DaimlerChrysler AG, Research and Technology

Virtual Reality Competence Center, P.O. Box 2360, 89013 Ulm / Germany
+49 731 505 4307
{Holger.Regenbrecht|Gregory.Baratoff}@DaimlerChrysler.Com

[b]shared-reality.com, igroup.org

Heilmeyersteige 156/4, 89075 Ulm / Germany

mtw@shared-reality.com

ABSTRACT

We describe an Augmented Reality system which allows multiple participants to interact with 2D and 3D data using tangible user interfaces. The system features face-to-face communication, collaborative viewing and manipulation of 3D models, and seamless access to 2D desktop applications within the shared 3D space. All virtual content, including 3D models and 2D desktop windows, is attached to tracked physical objects in order to leverage the efficiencies of natural two-handed manipulation. The presence of 2D desktop space within 3D facilitates data exchange between the two realms, enables control of 3D information by 2D applications, and generally increases productivity by providing access to familiar tools.

We present a general concept for a collaborative tangible AR system, including a comprehensive set of interaction techniques, a distributed hardware setup, and a component-based software architecture which can be flexibly configured using XML. We show the validity of our concept with an implementation of an application scenario from the automotive industry.

Keywords

Augmented Reality, Collaboration, CSCW, Tangible User Interfaces, 3D user interfaces.

1. INTRODUCTION

The MagicMeeting system presented here is a collaborative Augmented Reality system designed to support a scenario where a group of experts meet to discuss the design of a product. In our concrete case these are experts from the automotive industry who meet to discuss the design of various car parts and aggregates. The system provides the necessary hardware and software infrastructure for the participants to examine digital mockups (virtual 3D models) placed right there on the meeting table around which they are sitting. A tangible user interface allows them to interact with the digital mockup almost as if it was a physical object. At the same time the model can be copied to and from - and can be controlled by - standard 2D applications running on desktop computers.

Augmented Reality attempts to enrich a user's real environment by adding spatially aligned virtual objects (3D models, 2D textures, textual annotations, etc) to it (see [1,2,3] for definitions). The goal is to create the impression that the virtual objects are part of the real environment. In our setting, AR is used to "place" the *virtual* model on top of the *real* table by superimposing a perspectively correct view of the 3D model onto a users' view of the real scene. The users experience the augmented environment through head-mounted displays (HMDs). In order for the impression of a single shared space to be believable, an AR system must meet three major technical challenges. It must

(1) generate a high quality rendering of the objects,

(2) precisely register (in position and orientation) the virtual objects with the real environment, and

(3) do so in interactive real-time.

In a collaborative AR system multiple users share the same augmented environment. They can all simultaneously see and interact with the common virtual objects present in this environment. And, since the real world is part of what they see, they can see each other (which facilitates communication), and they have access to all the supporting material they brought to the meeting such as drawings and laptop computers. Users can collaborate either face-to-face in the same physical location, or remotely via teleconferencing. In any case, each user has his own view on the private and shared objects of the augmented space.

Our MagicMeeting system combines collaborative AR technology with:

(1) new interaction techniques utilizing ordinary desktop items (Tangible User Interfaces, see [18]),

(2) interactive 2D desktop screens integrated into the 3D environment [4,5,13], and

(3) linking mechanisms between 2D desktop applications and the augmented 3D space.

In section 2, we provide the motivation for our work by describing the purpose and structure of a design meeting in the automotive industry and by analyzing differences between physical and digital mockups. In section 3 work related to ours is discussed. In Section 4 we derive a set of requirements from the application scenario and present the MagicMeeting concept that aims to fulfill them. The different interaction techniques available within the system are presented in section 5. Section 6 discusses the distributed hardware setup and the component-based software architecture which implement the MagicMeeting concept. In section 7 we discuss user impressions of the system and in section 8 we summarize our work and discuss improvements and extensions to our current system that we are planning to work on in the future.

2. MOTIVATION

In the process of creating a complex modern-day industrial product many aspects need to be considered and often contrasting requirements need to be resolved. Because of the overall complexity of the design task and because of the variety of expertise necessary, it is nearly impossible for a single person to come up with a satisfactory design. Rather, a successful design is usually the result of an iterative process in which experts from different areas collaborate to best satisfy the various constraints and demands. Thus, during the design phase experts repeatedly meet to discuss individual parts or part aggregates of which the product is composed. At several points in this process physical prototypes or mockups of the product are built. Such mockups either represent only an approximation of the end product, or they focus on some aspects while neglecting others. But, even though they do not possess the full functionality of the end product, they provide a concrete object of discussion which allows further design decisions to be made. In the automotive industry such a meeting is called a "zone review". (A zone corresponds to a physical or functional aggregate of an automobile.) One objective of such a meeting is to decide whether the discussed parts pass certain "quality gates", i.e. if they satisfy the imposed requirements.

With the advent of computer aided design (CAD) software and desktop computers powerful enough to run it on each designer's workplace, there has been a move to replace physical mockups with digital mockups (DMUs). This approach saves the non-negligible costs of producing the mockups, and is more flexible, since changes can be easily incorporated into the digital model, sometimes even on-the-fly, during the same meeting. As a result, the time needed for each design iteration is reduced, and the design space is possibly better explored. These two factors are of strategic importance, since they allow the company to improve the product's quality and to reduce its time-to-market.

Whenever a new technology rushes in and promises to deliver better results cheaper and faster, one easily forgets about what was left behind. We found it instructive to analyze the traditional work practices associated with physical mockups in order to gain a better understanding of the ingredients necessary for a successful design meeting. This allowed us to uncover aspects not addressed by the new digital technology.

In an automotive zone review several experts usually meet face-to-face to discuss various car parts. (Sometimes a colleague might join them remotely via a teleconferencing link.) Mockups are usually built as one-to-one models. So, for larger parts or aggregates the experts will stand around the physical mockup, walking around it to inspect it from all sides. In most cases, however, the parts will be of moderate size, allowing the experts to sit around a table and to discuss the parts placed there. A table-top setting with sufficient space is necessary anyway, since the experts will bring along supporting material (drawings, data sheets, handbooks, ...) and tools (pens, pencils, erasers, calculators, laptop computers, ...) for use during the meeting. While discussing a part, a participant might grab it, bring it closer to get a better look at it, to compare it to one of his drawings, to check whether it fits with a second part, etc. Finally, after a decision is taken, each part is marked as to whether it passes the quality gate, whether it needs to be discussed at the next meeting, or whether it needs to be redesigned.

If a mockup is available in digital instead of in physical form, it can be visualized on a large-size projection screen on the wall. Typically, one of the participants will present a slide show (e.g. using MS PowerPoint) on a projection screen, with the others sitting at the table and listening. A stereo rendering solution would allow the participants to view the model in 3D. Then, changes to its shape, material properties and surface color could be quickly applied by one of the participants to jointly evaluate a few design variations. If this is to be done efficiently, a Virtual Reality (VR) software tool should be used. While CAD software emphasizes accuracy and expressive modeling, VR technology, with its focus on interactive manipulation, real-time rendering, and immersive visualization, can offer a compelling sense of the shape, appearance, and physical behavior of the model. This would be a first important step towards achieving believability. However, the model would still exist in a virtual world separate from the real world in which the meeting takes place. This would make it difficult for the participants to relate the model to any of their supporting documentation.

Furthermore, note that the nature of the interaction with the digital mockup is quite different from the interaction with the physical mockup. The participants are now restricted to visually examining the mockup, with the tangible aspect (grasping the object, feeling it, bringing it close, etc.) completely gone. Moreover, except for the presenter, they can't even chose the viewpoint from which to see the model. It is as if their hands and their feet were tied ! Since the participants lack the ability to get involved, it is more difficult for them to examine and evaluate a digital mockup than a physical one.

In the work presented here we attempt to restore some of the naturalness of the direct and active hands-on style of interaction afforded by a physical mockup. We restore some of the tangibility by providing various interface props (physical placeholders that can be manipulated in the same way as the real objects) with which the digital models can be manipulated. We restore the ability to select one's own viewpoint by providing each user with a see-through HMD. And we restore the impression of a single shared space by integrating the virtual 3D model display, 2D desktop applications, and the real space above the meeting table in a single augmented reality environment.

3. RELATED WORK

Our work combines many aspects of computing and user interfaces and borrows ideas from the fields of tangible user interfaces, collaborative augmented reality, distributed VR systems, and component-based software architectures.

Ishii's influential idea of "Tangible Bits" [18], which couple digital information with physical objects and architectural surfaces, opened up the new research area of tangible user interfaces. In the metaDESK system [19], standard 2D GUI elements like windows, icons, and menus, are given a physical instantiation as wooden frames, "phicons" (physical icons), and trays, respectively. The concept is demonstrated with "Tangible Geospace", a prototype application for interaction with a geographical space. MediaBlocks [20] are small, electronically tagged wooden blocks that serve as containers for online media, and provide seamless gateways between tangible and graphical interfaces. "Illuminating light" [23] is a rapid prototyping tool for optical engineers. It lets users place simple objects (phicons) representing real-world optics (lasers, mirrors, beamsplitters, and recording film) on a table and superimposes the virtual light beams resulting from a simulation of the optics arrangement (light emanates from the laser, bounces off a mirror, splits in two when it hits the beamsplitter, and is finally absorbed by the recording film). The phicons, which are labeled with different patterns of colored dots, are tracked by an overhead camera.

The ARtoolkit public-domain marker-based tracking library [9] allows real-time 3D pose tracking for an arbitrary number of markers. The "MagicBook" [11] implements a catalogue metaphor by associating virtual models with markers printed on the pages of a book. In a prototype interior design application [10], ARtoolkit is used to track a paddle with which furniture models can be scooped up from a MagicBook and placed (by letting them slide off the paddle) in a virtual room.

For transferring data between different computers, Rekimoto introduced Pick&Drop [15], a pen-based direct manipulation technique that allows objects to be picked up on a display and dropped on another one. The pen-manager on the network provides the illusion that the pen physically picks up and drops the (electronic) object. The system supports this operation between any collection of palm-sized, desk-top, and wall-sized pen-sensitive displays. A follow-up system [16] introduces Whiteboard techniques, which enable multiple users to work on a shared whiteboard by creating annotations on their palm-top computers.

"Studierstube" [6,7,8] is a collaborative AR system which supports multi-user and multi-context interaction in a shared virtual space. Each user perceives a shared space augmented with one or several virtual 3D datasets (surface or volume models). Users wear HMDs and interact with the data using a pen and a personal interaction panel (PIP), a hand-held physical board onto which virtual controls are superimposed. For example, EMMIE allows users to place data and applications in the shared space by dragging them off of displays into the "virtual ether". The data can then be processed by dropping it onto an application icon (typically represented by an icon). This is in contrast to the tangible user interface employed in our MagicMeeting system which heavily relies on props and requires all data and applications to be attached to physical objects. "Shared Space" [10] is a multi-user AR card matching game which explores augmented face-to-face communication in a table-top setting. "SeamlessDesign" [33] is a virtual/augmented system for rapid collaborative prototyping of 3D objects.

There have been previous attempts at combining 2D applications with 3D environments [4,5]. In [4], 2D X-windows are transparently displayed on an HMD using an overlay technique, their positions corresponding to tracked 3D locations. Due to the overlaying technique, a window is always oriented perpendicularly to the user's line of sight and is of fixed size. In contrast, [5] uses texture mapping to map the portion of the frame buffer corresponding to a window to an arbitrarily oriented plane in 3D space. Our implementation of 2D in 3D windows is based on our work on the "MagicDesk" [13], a single-user tangible AR system. This implementation also uses texture mapping, and improves the update rate by employing compression techniques. In contrast to [4,5], which target the X-Windows system, we have based ours on the Microsoft Windows platform.

4. CONCEPT

Although it would be desirable to have a universal meeting environment that can support many different kinds of scenarios (e.g. management presentations, architectural design review, etc.) we focus here on the design review scenario described in the motivation section. An analysis of the meeting scenario as it happens today leads one to take into account at least the following requirements when setting up a convincing augmented equivalent:

(1) a physical place for face-to-face communication should be provided

(2) two to eight people take part in an average meeting

(3) a presentation wall for slide show presentations should be available

(4) participant should be able to bring their own materials and tools (e.g. paper documents, notebook computers, calendars, mobile phones, personal digital assistants, etc.)

(5) networking and intra- and inter-net access should be provided

(6) space for "napkin sketches" and coffee mugs is needed

With the availability of AR technology and therefore the possibility to integrate digital information, the following additional desires come up:

(7) seamless access to all electronic data

(8) visualization of three-dimensional content

(9) simple interaction with 2D and 3D content

(10) access to tele-conferencing capabilities for remote collaboration

and nevertheless

(11) there should be no additional (disturbing) equipment.

We have implemented a solution which tries to fulfill these requirements as much as possible. MagicMeeting is a prototype environment available at our laboratory for ongoing usability studies. The final goal is the successful transfer to our automotive design department for everyday use. We next present the system concept, discuss its capabilities and limitations and evaluate how it measures up to the requirements formulated above.

Meeting Environment

Participants in a meeting situation are used to sitting around a table with the inherent possibility of face-to-face communication. MagicMeeting provides a table for up to four people (see figure 1). The users wear HMDs with video see-through capability. The HMDs can be clipped upwards, so the users do not have to look at the miniature screens all the time and are able to communicate directly.

Figure 1: MagicMeeting environment

Requirement #2 states that up to eight people should be included in such a meeting. Unfortunately, space, time, and financial constraints limit us to four users. This is, however, not a limitation of the system as such. As described in section 6, our distributed hard- and software system should easily scale to a few tens of users, which is more than enough to satisfy our zone review scenario requirements.

As in an ordinary meeting a large (back) projection screen for 2D presentations is provided. Usually the video image for this display comes from the central presentation server. However, an input selector allows it to take its input from any computer present in the room, including additional notebook computers brought to the meeting by the participants. To simultaneously discuss different 2D presentations a second (extra) TFT display is installed on the table.

The table is large enough to serve as an environment for paper documents, notebook computers, and all other equipment and utensils needed in a meeting. Computers brought to the meeting by the participants can be networked in any manner using the outlets (Ethernet, VGA, etc.) provided.

Figure 2: Four users looking at one common model (seen from extra camera)

The main advantages in using MagicMeeting instead of "traditional" meeting equipment can be shown in three domains:

(1) The possibility of presenting (and interacting with) virtual 2D desktops within a 3D environment: Besides the 2D screens that are physically present (projection, extra monitor, notebook computers) an unlimited number of virtual 2D desktops can be placed in the augmented space. A characteristic of our system is that all such virtual windows are attached to props, allowing natural tangible interaction. (See “2D workspaces within the 3D world” in the next section.)

(2) The interactive visualization of a shared 3D object integrated into the meeting environment (see figure 2): Although it is possible to place 3D at any location within the environment, we have purposely restricted

the location of 3D models to the space above the “cake platter". This focuses the discussion, and allows a very intuitive tangible form of collaborative interaction.

(3) The link between the 2D and 3D realms: To allow an almost seamless transition from 2D to 3D a comprehensive set of interaction techniques between 2D and 3D is implemented. This enables a continuous workflow.

While the setup described so far could serve as a universal AR-supported meeting environment we have to consider the specific requirements given by our design review scenario. The basic interaction techniques needed for this are described in the next section.

5. INTERACTION TECHNIQUES

The main goal of the MagicMeeting system is an almost seamless integration of 2D and 3D data in one shared environment. For this, the user interface should provide intuitive and efficient access to the displayed information. We achieve this by relying on tangible interaction techniques based on props.

Besides more "traditional" AR interaction techniques like mouse raycast, MagicBook, and models-on-marker (e.g. [13]) some new techniques are introduced here.

"Cake platter"

This turnable, plate-shaped device functions as the central location for placing shared 3D objects (figure 3). The objects or models can be placed on the platter using different interaction techniques, e.g. by triggering the transfer from a 2D application or by using transfer devices brought close to the cake platter.

Figure 3: 3D model area on “cake platter”

Each user participating in the meeting can physically turn the platter with his hands. This way he can choose any particular view onto the object on the platter which is of interest to himself, or he can turn the object to point out a feature he wants to discuss with a colleague.

Furthermore, since the cake platter is not attached to the table, it can be lifted, brought closer, or tilted for more detailed inspection. The main advantage of this kind of tangible interface is the very natural interaction. Users don’t have to be given explanations on how to turn the virtual object. Hundreds of people in our laboratories have used the cake platter without asking for instructions.

Personal Digital Assistant

Many employees in our enterprise own a PDA. It therefore makes sense to incorporate this device into the MagicMeeting system.

We use PDAs (in our case a PalmPilot IIIc) as catalogues of virtual models (an example of the MagicBook metaphor [11]), the main form of interaction within our system being model selection and transfer to and from the cake platter (see Figure 4).

Figure 4: Using a PDA for object transfer to/from cake platter

For model selection several different markers (up to 12 at a time) are displayed as thumbnail images on the PDA screen (see figure 5). The user selects a model by simply touching the corresponding marker with a pen or with his finger. The selected marker grows to screen size and the appropriate model is displayed in large.

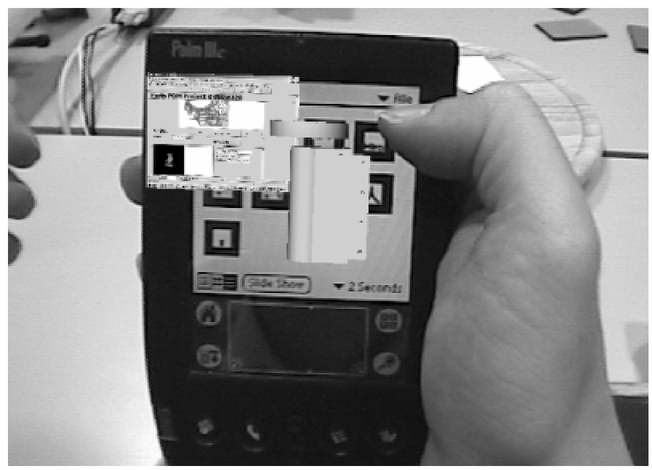


Figure 5: PDA (PalmPilot IIIc) for model selection. Two of the markers have content visually attached (using AR-overlay) to them (a desktop window and a 3D model, respectively).

Once a model is displayed in large on the PDA it can be placed on the cake platter for further. To do this the user brings the PDA close to the cake platter. After a short delay the models are exchanged: the model on the PDA moves onto the cake platter, and the model on the cake platter moves onto the PDA (see figure 4). We opted for this exchange method instead of one-way model "transport" after some usability trials. Compared to providing a special interaction element to specify the direction of transport the exchange method seemed to be the easier one.

A further interesting form of interaction made possible by PDAs is the direct (peer-to-peer) exchange of models between participants. This functionality is afforded by the PDAs’ built-in infrared transmission ability. If one user wants to obtain a model from another one he simply asks the other user to send him the marker of the desired model (see figure 6). This procedure takes only a few seconds and is very intuitive for PDA users.

Before, during, or after a MagicMeeting users can prepare their own set of models on the PDA simply by uploading the appropriate markers. This can be done via IR as described above, by using the stationary station (cradle) of the PDA connected to a PC, or by connecting the PDA directly to a computer (usually a notebook computer). For this reason we provide cradles in our laboratory as well as network outlets to connect notebook computers to the intra- or inter-net.

Unfortunately the display of the models on the PDA is not as stable as with markers printed on paper. The reflections on the PDA screen surface have to be reduced to achieve sufficient results. This can be done by using special cover slides or by preparing the PDA screen with anti-gloss spray.

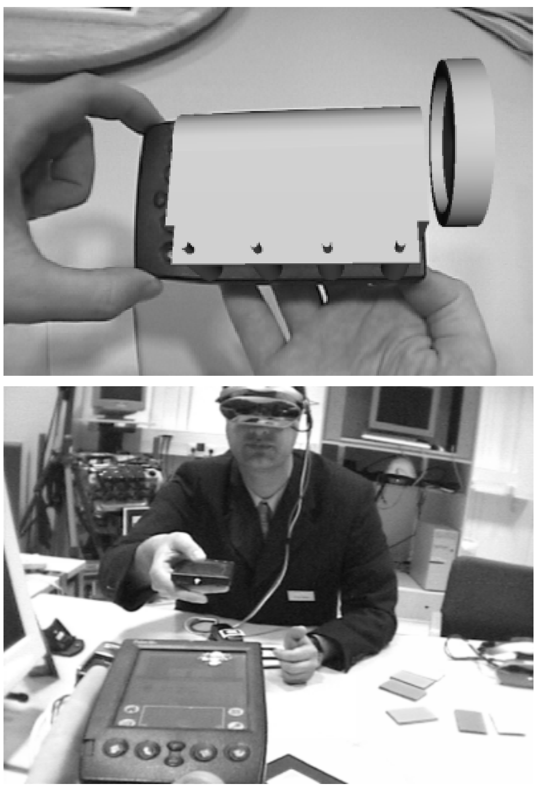


Figure 6: Exchanging objects via IR link of the PDA's

Clipping plane

A common technique for seeing what is "inside" a virtual object is to cut it with a clipping plane. There are several traditional interfaces for controlling clipping planes. In 2D applications Arcball and related techniques using a 2D mouse are employed. In 3D environments this is done with 6DOF input devices (like SpaceMouse or Polhemus Stylus) or with special input devices like the CubicMouse [28]. In the MagicMeeting setup a hand-held real (transparent or opaque) plane is used to clip through the virtual model on the cake platter (see figures 7 and 8). The transparent version of the clipping plane allows a very natural handling because of the direct mapping of function and device. Unfortunately the tracking is not always stable, because reflections on the transparent plane lead to the non-detection of markers on the platter seen through the plane. For this reason we provide an opaque version which is shown in figure 8, where a marker attached to any appropriate object can serve as a clipping plane device. In the current system the plane with the marker is attached to an ordinary office stapler.

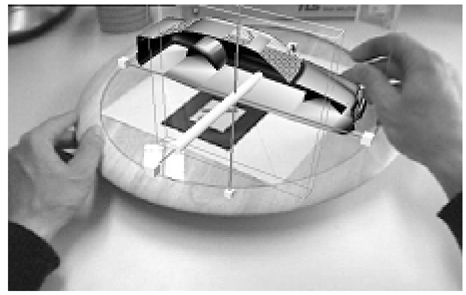


Figure 7: Transparent clipping plane

Figure 8: Opaque clipping plane

Lighting

To evaluate the surface properties of a 3D model, our system allows the simulation of a light source controlled by a light prop. Instead of using a virtual light only, the direction and distance of the light are controlled by moving a real light “device” such as an office lamp or a flashlight (figure 9).

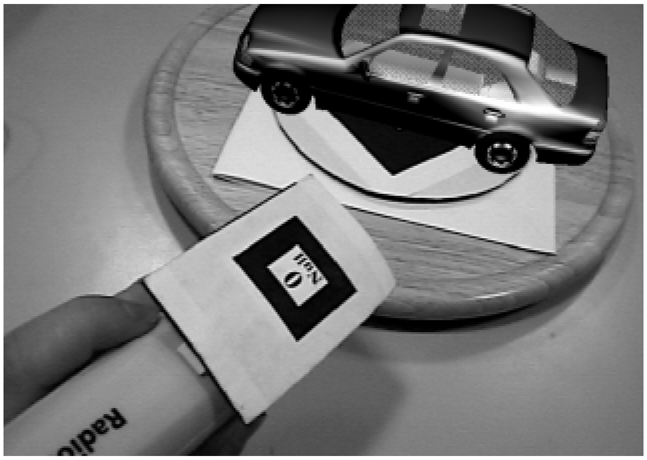


Figure 9: Real flashlight controlling a virtual light source

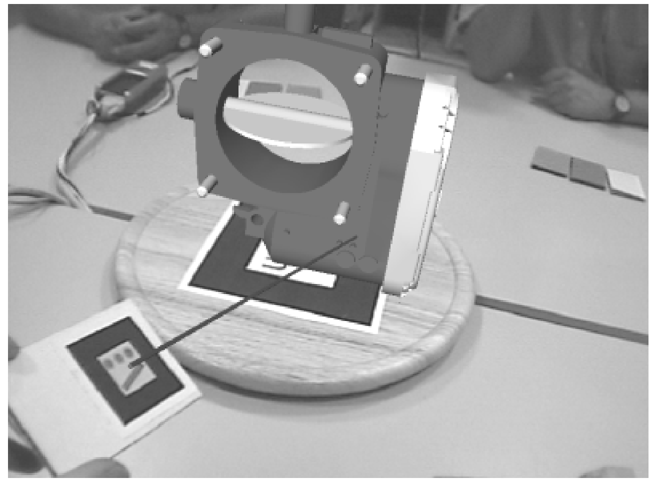


Figure 10: Simple color annotators for design review

Annotations

Each user has at his disposal tools with markers on them to color parts of the model for discussion purposes. This is a common procedure in design review scenarios. After annotating a part of the model on the cake platter with one of the three standard colors (red, yellow, green), an update message is sent to a database containing data about the person annotating, the part's design status (e.g., "part needs to be redesigned") corresponding to the selected color, and the part itself as 3D information.

The annotation interface itself is a very simple one. Each user has three different cards with markers attached to them. The cards represent the possible annotation colors. When a user points towards the 3D model with one of the annotation cards, the ray - which otherwise has a fixed length - connects the card and the 3D model. The part of the model which is hit by the ray turns the color of the card. Pointing once again on the same part with the same color reverses the action, causing the part to switch back to its original color. All users can simultaneously annotate the object on the cake platter, the order of the annotations being arbitrated by a synchronization mechanism of the centralized database.

We have chosen three colors for annotations because of the requirements of our users in the design zone review. Of course, The MagicMeeting system itself is not limited to three colors, nor does it impose any restrictions on the textures and geometries used.

2D workspaces within the 3D world

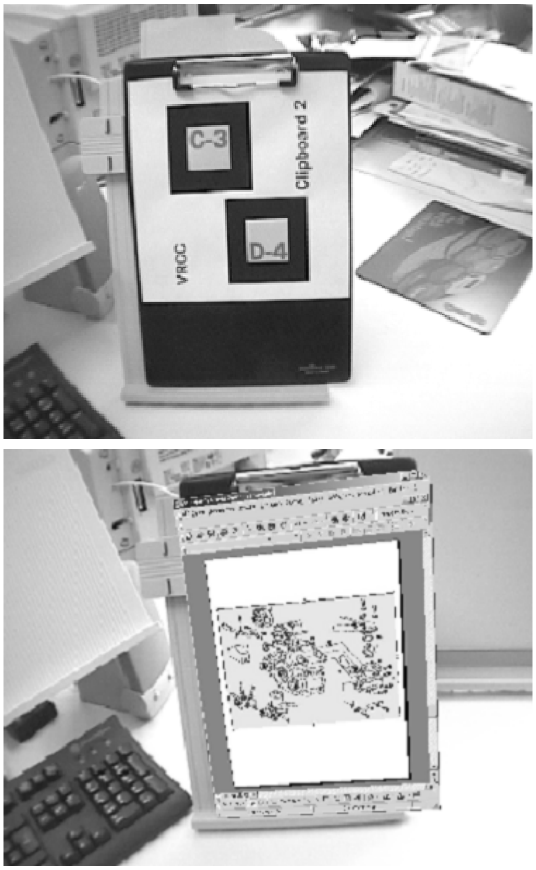
Apart from some specialists (e.g. CAD engineers) the standard environment for a computer user today is the two-dimensional desktop screen. Almost all applications work within and are designed for this interface. This is a fact that we cannot and will not ignore. Therefore it is essential for a successful new 3D application to integrate as much as possible elements of the traditional workflow of the user into the environment to be provided.

We have chosen an approach which shows interactive 2D applications within our 3D environment as they are: as two-dimensional. They can be placed in space like any other 3D object. There are three types of 2D display in MagicMeeting:

(1) physical computer screens (CRT monitors, TFT displays, notebook computer screens, large projection screens) as used in a standard office or modern meeting environment. The users can look at the screens either through their HMDs or directly.

(2) entire 2D desktop screens (e.g. MS Windows or X-Windows) attached to an object in MagicMeeting space. The content of a real desktop screen is transmitted via network to MagicMeeting, so users can work with their standard environment and applications with almost no limitations.

(3) single windows (belonging to a single application) attached to objects in 3D space. This is especially useful when elements of the mixed environment need to be controlled by a standard 2D interface. For example, the color of an object can be controlled by using a standard 2D color editor dialog instead of inventing/implementing a new 3D dialog.



Figures 11: 2D Windows applications attached to marker-tracked clipboards

In our environment, 2D content is attached to physical clipboards or picture frames (see figure 11). This allows a very natural handling of 2D application space within the 3D MagicMeeting environment, because they can be moved and placed in a tangible way.

The interaction with 2D applications is done by either using a standard 2D mouse, in which case the mouse cursor on the augmented window behaves like one in a standard desktop environment, or by using the 6DOF mouse mode, where the 6DOF ray is used to control the 2D application (see section below).

In some cases it is necessary to input text strings. This can be done (1) by simply using a real keyboard, (2) by using a virtual keyboard on the virtual screen operated by the 2DOF/6DOF mouse, or (3) by speech input. The latter method seems most natural. However, it is best used for issuing commands, since is not robust enough for general purpose text input.

Computer mouse as 2DOF/6DOF interaction device

The standard interaction device in today's desktop applications is the computer mouse.

Figures 12: Desktop mouse used as 2DOF or 6DOF device

The MagicMeeting system also makes use of this device, but in two ways: (1) The interaction with the two-dimensional screens within the system (large projection screen, extra monitor, notebook computer on the table, virtual 2D screens attached to markers) is done in the way users are familiar with the mouse. (2) If a user lifts his mouse and turns it upside down, a virtual ray appears which can be used for 6DOF interaction in 3D space (see figure 12). This mode is mainly used for object manipulation on the cake platter and for spatial interaction with the virtual windows in 3D space.

With this kind of interface a single well-known device supports the traditional form of interaction as well as new forms of interaction in 3D, with the turn metaphor providing the transition from 2D to 3D.

2D - 3D link

To integrate 2D and 3D information into one shared environment we have implemented several mechanisms:

(1) interactive computer desktops (here, MS Windows) can be placed within the 3D environment,

(2) 3D data contained in 2D applications (e.g. as email attachments) can be transferred onto the cake platter,

(3) 2D applications, such as Netscape (via Java) or Microsoft Office (via Visual Basic) can control the models displayed in the environment (see figure 13),

(4) data out of the 3D space (such as the image of a clipped plane) can be imported into a 2D application (see figure 14).

Within our application scenario the link functionality is used in many ways. Imagine one of the MagicMeeting participants standing next to a large projection screen and giving a talk using a 2D presentation tool, e.g. a Web Browser. (We believe that even in the near future two-dimensional presentations will continue to be given.) The audience follows the talk by direct or "through-the-HMD" viewing. At key points during the presentation, the speaker can select a prepared object model from the current slide, and a network operation loads the appropriate 3D geometry onto the cake platter. Figure 13 shows a picture of an engine part in an HTML presentation and the corresponding 3D model which was loaded onto the platter.

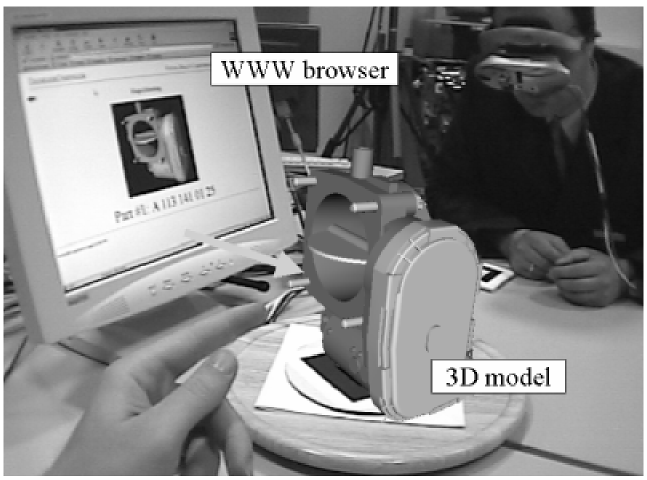


Figure 13: Mechanisms in 2D applications for data exchange between 2D and 3D space (Java, Visual Basic)

To integrate MagicMeeting with the working processes of the users a more comprehensive link between 2D applications and 3D space is needed. One first approach is the connection of a database to our system. We provide an interface to a Microsoft Access database via VisualBasic. To illustrate this function, we placed data from a Product Data Management (PDM) system into the MS Access database. When the user selects the "examine" function in the database application the appropriate 3D model is loaded onto the platter. Conversely, it is possible to send back clipping information or annotation information from the MagicMeeting system to the database for archival. This information consist of numeric, alphanumeric, as well as pictural data (such as a snapshot of the clipped object).

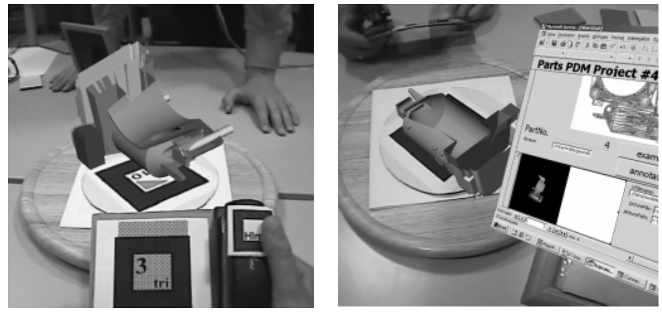


Figure 14: Transfer and transformation of clipped image to data base

6. IMPLEMENTATION

The hardware configuration as well as the software implementation are specifically designed for instantiating the MagicMeeting concept described above. Because no off-the-shelf (OTS) or standard solution is available at the time, we have developed an integrated hard- and software solution.

6.1. Hardware setup

A schematic overview of the main components of the system is given in figure 15.

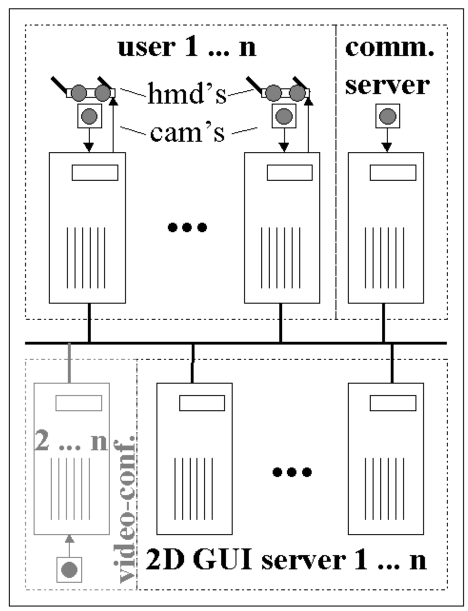


Figure 15: Schematic hardware setup

Each user wears an HMD-camera combination connected to a dedicated PC, one for each user. The user PCs and the server PC are connected via a 100 MBit network. Also within the same network are the PCs or workstations responsible for the 2D GUI display within the 3D environment, as well as the (notebook) computers brought to the meeting by the participants.

The four user PCs and the communication server PC have the same hardware configuration: Dual-Pentium-III processors running at 933 MHz, Microsoft Windows 2000 operating system, bt878 video capture card (Hauppauge WinTV Go!), nVidia GeForce2 graphics board, and 15"

TFT display. For the user PCs the VGA output is connected to their HMD as well as to their TFT display using a VGA splitter. All PCs are connected by a network switch using the same subnet.

Choosing the right display unit is a very difficult task, because there is no optimal solution available. After evaluating several HMD-camera combinations (e.g. Sony Glasstron PLM-S700E with Toshiba IK-CU50 or Olympus EyeTrek with Visual Pacific PC-605) with respect to weight, comfort, price, and availability of the HMD, and camera resolution, size and weight, we decided to use the combination of the Cy-Visor glasses and a Visual Pacific PC-206 camera.

The Cy-Visor glasses have a resolution of 800x600 at 60Hz. In comparison to the Sony Glasstron models they are generally available over the counter and are relatively inexpensive (around USD 1,000). The VP PC206 is a very cheap pinhole color camera (approximately USD 200, PAL interlaced), which has sufficient quality for our video see-through approach in combination with the Cy-Visor glasses.

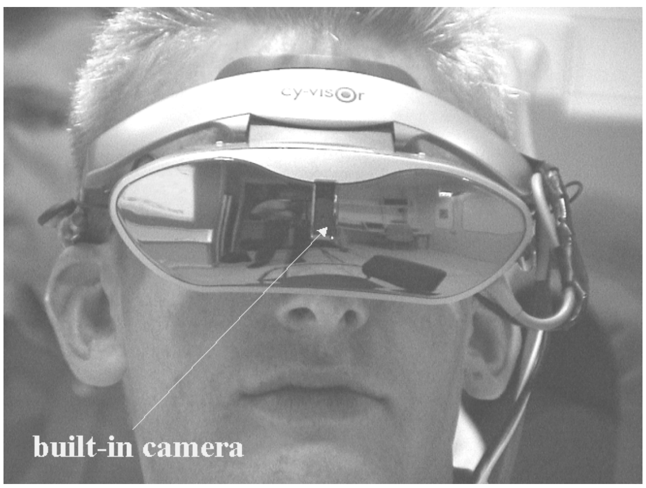


Figure 16: Modified Cy-visor head-mounted display

Because MagicMeeting was designed and developed to be used by hundreds of users a very robust solution for the HMD-camera combination was needed. We therefore placed the camera inside the HMD behind the front face of the Cy-Visor glasses (see figure 16). To do this we removed the mechanism for manually controlling the interpupillary distance (IPD) from the HMD and built in the camera in its place. This final solution works very well except that the Cy-Visor glasses are obviously not designed for everyday use. In particular, the cables tended to break frequently.

In principle our concept and implementation does not depend on any particular HMD or camera. But it is very advisable to evaluate the devices according to the intended scenario and user needs.

6.2. Software setup

The following section describes a component-based architecture for distributed augmented reality environments developed for the MagicMeeting system.

Requirements

Since MagicMeeting is an experimental platform we wanted it to be lightweight, flexible, and dynamically configurable to be able to easily implement different application and interaction scenarios while using a large variety of interaction devices and metaphors.

To be able to set up distributed multi-user scenarios the system architecture should provide the ability to distribute software components onto different computers running different operating systems.

Since we are dealing with an interactive real-time graphics environment, performance is a big issue. For reasons of efficiency we opted for the C++ language.

Simple Component-Oriented Architecture

Our requirements are best met by an approach which allows one to build software components encapsulating a specific functionality or behavior and to assemble them into larger ones of higher complexity, a process similar to building models with the popular LEGO™ system.

More specifically,

"*A component denotes a self-contained entity (black-box) that exports functionality to its environment and may also import functionality from its environment using well-defined and open interfaces. […] Components may support their integration into the surrounding environment by providing mechanics such as introspection or configuration functionality*" [22]

There are different frameworks for a component-oriented software architecture in the area of client-server computing. We have examined client side models such as JavaBeans [24] and Component Object Model (COM) [25], as well as server side models such as Enterprise Java Beans (EJB) [26] and its superset Corba Component Model (CCM) [27].

Although the (Enterprise) JavaBeans concept with its InfoBus addition seemed most appropriate, we could not use it because we rely on the C++ programming language. We rejected COM, which is limited to Microsoft Windows-based environments, because it would have violated the heterogeneity requirement. CCM, being rather complex, seemed oversized for our lightweight approach. However, it provided valuable inspiration for our design.

Our approach defines a component as a named software entity which encapsulates a certain functionality, role, or interaction metaphor. It interfaces with other components through event ports which consume (event sink) or emit (event source) events of a specified type. It exposes internal values through attributes and exposes internal actions with a command interface (figure 17).

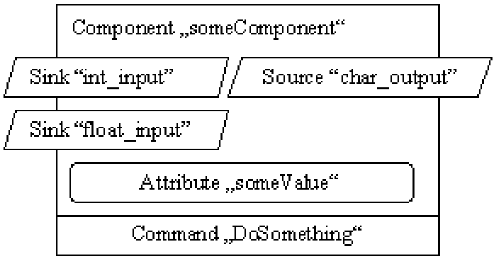


Figure 17: A component description with named event sinks, event sources, and a command interface

Each event sink, source, attribute, and command has a unique name. Event sources are connected to one or more event sinks or to an event channel which broadcasts the emitted event. Attributes are both event sinks and sources and maintain a state.

We have also defined a named container component which we call “controller” (see figure 18). It provides a context for registered components as well as a communication infrastructure for event routing and distribution. Since the controller is itself a component it is easy to build complex hierarchical component structures.

Controllers and components register with a global naming service to allow event and command routing across process boundaries.

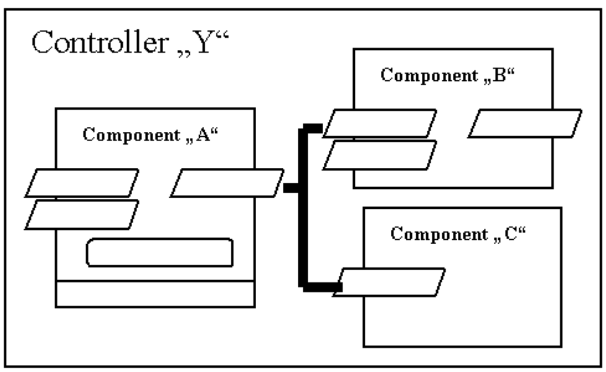


Figure 18: A controller component description with three registered components. There are two event routes defined from A to C and from A to B.

Controllers can dynamically add and remove components as well as event routings, which allows dynamic and flexible configuration of applications.

XML-based component specification and configuration

We use XML to describe the components including its event sinks and sources, attributes, command variables and events, in a way comparable to an IDL (Interface Definition Language) definition. To validate the XML description we have defined an XML schema. We also use an XML schema to describe the structure of events, attributes and command parameters (figure 19).

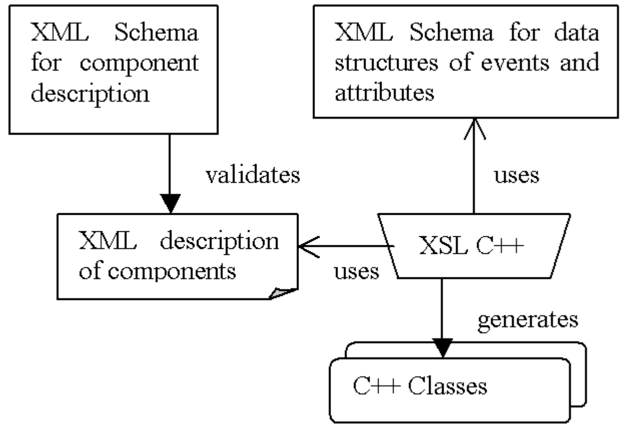


Figure 19: XML framework for code generation

An XSL style sheet generates C++ classes according to our component structure. The classes are augmented with component-specific functionality using callback mechanisms or derivation. Since the data structures of events and attributes are described in XML schema we can automatically generate data access and distribution code such as serialization. A different XSL style sheet could generate a CORBA compliant IDL or Java code, which leaves the door open for integration with other component models.

We also use XML to describe configuration and deployment of the components for a specific scenario. These are also validated against the associated XML schema.

We have chosen to use XML as description platform since it is an open and standardized meta language ideally suited for hierarchical structures and because there are many software tools available for editing and parsing XML documents.

MagicMeeting components

Our MagicMeeting environment uses different components and controllers playing different roles:

- device abstraction components (e.g. marker tracker or mouse),
- interaction components implementing interaction metaphors (e.g. RayPicker),
- adaptor components connecting event ports of different type (e.g. SensorPoseAdaptor),
- decorator components adding additional functionality to other components (e.g. SmoothingComponent for Tracker output),
- visualization-related components called “areas”.
- The visual pendant to controllers are “area manager” components which are containers for area components. They provide a spatial context for registered areas and use layout algorithms to arrange registered areas automatically.

We next illustrate the architectural approach with a scenario in which two designers have to collaboratively evaluate a model of an engine part. One designer works in an ordinary desktop working environment, while the other works in an augmented reality environment (see figure 20).

As explained in the Interaction section, the designers can annotate specific engine parts by coloring them. The color of a part is changed by selecting it with a ray casting device, which would be a mouse in the desktop environment and a tracked marker emitting a virtual pick ray in the 3D environment.

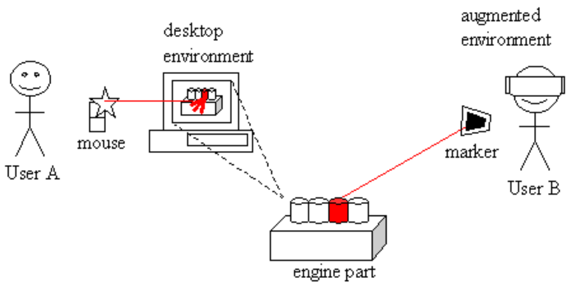


Figure 20: Two users interact on a virtual engine part using different interaction devices

The components taking part in this scenario can be grouped into the categories mentioned above.

With respect to the component categories introduced above, the desktop mouse and the marker tracking system (in our case a modified version of the popular ARToolkit system [9]) correspond to event emitting device abstraction components. These are connected (by way of adaptor components) to the raypicker interaction component which triggers the component responsible for toggling the color of model parts residing in the model area component (see figure 21).

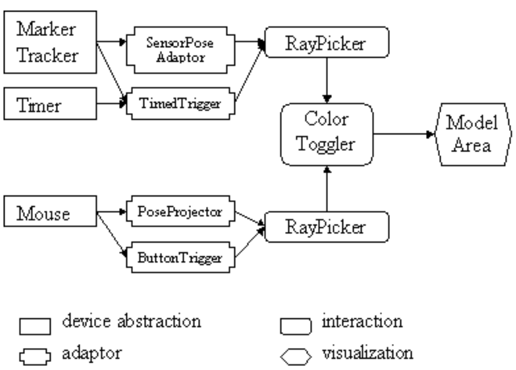


Figure 21 : Interconnected components used in the scenario

The adaptor components are used to post-process data and to convert the events to fit the event type required by connected event sinks. The MarkerTracker sensor event sources emit raw events of type SensorPose containing a position, orientation and a visibility flag. The ray picker component has an event sink "Pose" which controls the pick ray's direction. The SensorPoseAdaptor component filters the raw SensorPose events and passes the filtererd data on to the Pose event sink.

To support distributed deployment of components we use a global hierarchical namespace with which all components register. A component can have multiple copies in different processes residing on different machines. All these copies can be kept synchronous by our framework since they have the same name. For example, this feature is used to synchronize the event notification mechanisms of controller components.

The namespace for our example scenario is the following :

```
Devices     /Mouse
            /Tracker
Users       /A     /PoseProjector
                   /ButtonTrigger
                   /RayPicker
            /B     /SensorPoseAdaptor
                   /TimedTrigger
                   /RayPicker
Areas       /Examiner   /ModelArea
                        /ColorToggler
```

Based on this naming scheme routes (connections between sinks and sources) can be defined externally using XML.

7. USING THE SYSTEM

Until today a couple of hundred users have tried our collaborative system. Most of the time this happened during or after a presentation given at our laboratory and lasted only for 10 minutes or less. We have not performed any formal usability studies yet but we will give some user impressions nevertheless.

The users' first impression was always very positive. The participants especially liked the fact that the virtual model in the middle of the table (on the CakePlatter) is actually viewed from an individual perspective. Visitors watching the scene on the extra monitors did not always realize this, and would often ask one of the four users if they really had their own view onto the same model. The "immersed" users realized this instantly.

Also very impressive for the users was the easiness of using the CakePlatter. Although the manipulation is constrained to one axis only nobody had complaints about this. Many users felt that a real model or part was sitting on the platter. Because the CakePlatter itself has a certain weight and inertia the virtual model seems to possess these properties, too.

A little bit more complicated is the usage of the interaction devices. The main reasons are the occlusion problems caused by the markers. For instance, when the marker of an interaction device, e.g. the annotation card, hides the marker(s) on the CakePlatter, the model on the platter suddenly disappears and the intended interaction can not be completed. So we have to explain this fact to the users, which is often not obvious, especially for users unfamiliar with virtual reality or vision-based systems. After this explanation most users are able to handle the interaction devices successfully. They play around with the tools and see the feedback immediately.

A second problem occurs when using a PDA. The reflections on the screen surface of the PDA are very disturbing, often prevent the displayed markers from being recognized and the models to be overlaid. The user has to tilt the PDA to find a reflection-less position. So, right now this device can be successfully operated only by our staff or by "talented" users.

Sometimes users complained about not seeing the ray of another user when he or she was annotating. The reason for this is that the size of the markers on the annotation cards is simply to small to be recognized from distances over half a meter. Instead, while discussing the model the participants used their hands to explain what they just annotated, or would announce what they were planning to annotate and in what color. This type of communication is very natural and was intended.

Besides these problems the interaction devices themselves seem to be very intuitive. The users can simply try them out and see the effects within the augmented world. The only explanations wee needed to give were to just say what the devices are good for. E.g. "This is a clipping plane. Keep in mind the marker problem!", and the user can start exploring the interface. After a couple of seconds he or she is able to operate the device accordingly. The same can be said about the light, the model exchange, and so on.

Interestingly, almost nobody realized that the system is a monoscopic see-through one. We guess that the ability to freely move the head and to turn the model on the platter compensates the lack of stereoscopic viewing to a high degree. All users had a three-dimensional impression of the model and of the real meeting environment.

The overall verbal judgment of the system was always very positive. The users enjoy exploring the system and most of the time they come up with some ideas on how to apply this technology to their special industrial or academic working environment.

8. CONCLUSION AND OUTLOOK

We have presented a multi-user augmented reality system which allows up to four users to have a design zone review meeting. New technologies and interaction techniques were introduced and validated with our implementation of the MagicMeeting system. Although our motivation and the requirements came from a specific scenario the results can be transferred to many other applications.

The MagicMeeting system as described in the previous sections is up and running. We have extensively experimented with it, and over hundred people have already used it. The system runs stably and is easy to use. Based on our experiences and on the feedback received from the users we have identified several aspects of the system that could be improved.

The biggest challenge is the improvement of the hardware components. However, this is mostly outside of the scope of our research (except for the design of new interaction devices, see below). Concerning the improvement of HMD technology (better resolution, larger field of view, more comfort) we must rely on the continued development by other manufacturers. The same holds for projection display technology and computer hardware in general.

The challenges within our scope of competence and interest are the following:

- We are going to develop and implement new interaction techniques with new or modified interaction devices. Even at this stage of evaluation we doubt whether for instance the trigger-less annotation cards are the best way of annotating 3D objects. Developing appropriate devices for this task is one of the aims of a national project we are involved in (see www.vrib.de).
- One of the main challenges in realizing a convincing and robust solution is the tracking quality. Besides the improvements being made by hardware manufacturers we are working on a hybrid sensor fusion approach which allows us to combine different types of tracking devices. Currently, work is underway to combine the marker-based tracking with an inertial sensor to stabilize head tracking. Our next step will be to add several fixed cameras overlooking the MagicMeeting table, in order to stabilize tracking over the entire work space. We are currently investigating calibration algorithms for properly fusing such outside-in tracking with HMD-based inside-out tracking.
- We are planning to investigate alternative display technologies that would allow us to replace (or complement) the HMD-based approach with projection-based systems [31,32]. An approach that appears particularly promising is “extended VR” [29].
- In order to successfully transfer the MagicMeeting technology to our automotive design department, we need to integrate the component architecture into our Virtual Reality software system DBView [30], perform further usability tests, and fully integrate our system into the working processes of our end users. In particular, this will require a proper interface with their product data management system.
- We are currently working towards providing teleconferencing capabilities in our system. Our first approach will be to include one remote participant via internet using a hardware setup consisting of cameras, loudspeakers, microphones, and markers. The data transmission will be based on the same VPN technology already in use for the display of 2D windows within 3D. An underlying standard teleconferencing protocol will ensure portability.

Additionally, we are working on improving some of the currently implemented interaction methods:

- applying constraints to clipping planes (model-related, world-related, user-related)
- manipulating the properties of the virtual light (intensity, angle, color)
- providing different forms of light (bulbs, neon lamps,...)
- integration of speech input for issuing commands

ACKNOWLEDGMENTS

We would like to thank Claudia Ott, Ralph Schoenfelder, Mark Billinghurst, Hirokazu Kato, Ivan Poupyrev, Bernd Kounovsky, Christian Bader, Steffen Cersowsky, and Beat Bruederlin for their contributions to MagicMeeting and for their comments on earlier versions of the paper.

Parts of this project were funded by the German Federal Ministry of Education and Research within the "VRIB" project (01 IR A05 A).

REFERENCES

1. Milgram, P., Takemura, H., Utsumi, A., Kishino, F., Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum. *Proceedings of Telemanipulator and Telepresence Technologies*. 1994. SPIE Vol. 2351, 282-292.
2. Azuma, R., A Survey of Augmented Reality. *Presence:Teleoperators and Virtual Environments*, 1997. 6(4): pp. 355-385.
3. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., and MacIntyre, B. Recent Advances in Augmented Reality. *IEEE Computer Graphics and Applications*, vol. 21, No. 6, November/December 2001, pp. 34-47.
4. Feiner, S., MacIntyre, B., Haupt, M., and Solomon, E. Windows on the world: 2D windows for 3D augmented reality. *Proc. UIST '93* (ACM Symp. on User Interface Software and Technology), Atlanta, GA, November 3-5, 1993, 145-155.
5. Dykstra, P., X11 in Virtual Environments. In *Proc. IEEE 1993 Symposium on Research Frontiers in Virtual Reality*. San Jose, CA, October 25–26, 1993.
6. Schmalstieg D., A. Fuhrmann, Zs. Szalavari, M. Gervautz. Studierstube - Collaborative Augmented Reality, *Proc.* Collaborative Virtual Environments ‘96, Nottingham, UK, Sep. 1996.
7. Fuhrmann, A., Löffelmann, H., Schmalstieg, D., and Gervautz, M. Collaborative Visualization in Augmented Reality. *IEEE Computer Graphics and Applications*, vol. 18, No. 4, July/August 1998, pp. 54-59.
8. Schmalstieg, D., Fuhrmann, A., Hesina, G., Bridging multiple user interface dimensions with augmented reality systems. *Proc. Int. Symp. On Augmented Reality (ISAR’2000)*, Munich, Germany, October 2000, pp. 20-29.
9. Kato, H., Billinghurst, M., Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System, *Proc. 2nd Int. Workshop on Augmented Reality*. 1999. pp. 85-94.
10. Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., Tachibana, K. Virtual Object Manipulation on a Table-Top AR Environment. In *Proc. IEEE Int. Symp. On Augmented Reality (ISAR’2000)*, Munich, Germany, October 2000, pp. 111-119.
11. Billinghurst, M., Kato, H., and Poupyrev, I. The Magic Book : A Transitional AR Interface. *Computers & Graphics*, Special Issue on "Mixed Realities - Beyond Conventions".
12. Regenbrecht, H., Baratoff, G., Poupyrev, I., and Billinghurst, M. A Cable-less Interaction Device for AR and VR Environments. *Proc. Int. Symp. On Mixed Reality (ISMR 2001)*, Yokohama, Japan, March 14-15, 2001.
13. Regenbrecht, H., Baratoff, G., & Wagner, M. (2001). A tangible AR desktop environment. Computers & Graphics, Vol. 25, Number 5, Elsevier Science, october2001, pp. 755-763.
14. Regenbrecht, H. and Wagner, M. Prop-based interaction in mixed reality applications. In Smith, Salvendy, Harris, & Koubek (eds.): Proceedings of HCI International 2001, Lawrence Erlbaum Associates, Publishers, Mahwah/NJ, Vol. 1, pp. 782-786. 2001.
15. Rekimoto J. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments, *Proc. UIST ‘97*, pp. 31-39, 1997.
16. Rekimoto J. A Multiple Device Approach for Supporting Whiteboard-based Interactions, *Proc. CHI ‘98*, pp. 344-351, 1998.
17. Butz A., T. Höllerer, S. Feiner, B. MacIntyre, C. Beshers. Enveloping Computers and Users in a Collaborative 3D Augmented Reality, *Proc. IWAR ‘99*, pp. 1999.
18. Ishii H., B. Ulmer. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms, *Proc. CHI ‘97*, pp. 234-241, 1997.
19. Ullmer, B., Ishii, H., The metaDesk: Models and Prototypes for Tangible User Interfaces. *Proceedings of UIST'97*. 1997. ACM. pp. 223-232.
20. Ullmer B., H. Ishii, D. Glas. mediaBlocks: Physical Containers, Transports, and Controls for Online Media, *Proc. SIGGRAPH ‘98*, pp. 379-386, July 1998.
21. W. Pree, M. Stal, C. Szyperski What charaterizes a software component. Software-Concepts and Tools 19(1) guest editor: W. Pree, Springer Verlag, Heidelberg, 1998
22. M. Broy, A. Deimel, J. Henn, K. Koskimies, F. Plasil, G. Pomberger. What characterizes a (software)

component? Software - Concepts and Tools, Volume 19, Number 1, 1998, pp. 49-56

23. Underkoffler J., Ishii, H. Illuminating Light: An Optical Design Tool with a Luminous-Tangible Interface. In *Proceedings of CHI '98*, April 1998.

24. Hamilton, Graham. Java Beans, http://java.sun.com/products/javabeans/docs/spec.html, 1997.

25. Rogerson, D.E. Inside COM. Microsoft's Component Model. Microsoft Press, Redmond, 1997.

26. DeMichiel, L.G. Enterprise JavaBeans. http://java.sun.com/products/ejb/docs.html, 2001.

27. OMG:CORBA components, Volumes I-III. http://www.omg.org/cgi-bin/doc?orbos/99-07-01/02/03, 1999.

28. Fröhlich, B. and Plate, J. The CubicMouse - a new device for three-dimensional input. *Proceedings of CHI 2000*, Turner, 2000.

29. O. Bimber L.M. Encarnação, and P. Branco. The Extended Virtual Table: An Optical Extension for Table-Like Projection Systems. to appear in PRESENCE: Teleoperators and Virtual Environments, vol. 10, no. 6, December 2001.

30. Sauer, J. Virtual Reality in der Produktentwicklung [Virtual Reality in Product Development]. VDI-Berichte [Reports] Nr. 1614, 2001.

31. Yoshifumi Kitamura, Takashige Konishi, Sumihiko Yamamoto, Fumio Kishino. "Interactive Stereoscopic Display for Three or More Users", *Proc. SIGGRAPH 2001*, pp.231-239, August 2001.

32. Bimber, O., Fröhlich, B., Schmalstieg, D., Encarnacao, L.M. "The Virtual Showcase", IEEE Computer Graphics and Applications, Vol. 21, No. 6, pp. 48-55, (2001).

33. Kiyoshi Kiyokawa, Haruo Takemura, Naokazu Yokoya, "SeamlessDesign for 3D Object Creation", IEEE MultiMedia, ICMCS '99 Special Issue, Vol.7, No.1, pp.22-33, 2000.