Remote Collaboration Using Augmented Reality Videoconferencing

Istvan Barakonyi Tamer Fahmy Dieter Schmalstieg

Vienna University of Technology
Interactive Media Systems Group
Email: {bara | fahmy | schmalstieg}@ims.tuwien.ac.at

Abstract

This paper describes an Augmented Reality Videoconferencing System, which is a novel remote collaboration tool combining a desktop-based AR system and a videoconference module. The novelty of our system is the combination of these tools with AR applications superimposed on live video background displaying the conference parties' real environment, merging the advantages of the natural face-to-face communication of videoconferencing and AR's interaction capabilities with distributed virtual objects using tangible physical artifacts. The simplicity of the system makes it affordable for everyday use. We explain our system design based on concurrent video streaming, optical tracking and 3D application sharing, and provide experimental proof that it yields superior quality compared to pure video streaming with successive optical tracking from the compressed streams. We demonstrate the system's collaborative features with a volume rendering application that allows users to display and examine volumetric data simultaneously and to highlight or explore slices of the volume by manipulating an optical marker as a cutting plane interaction device.

Keywords: Augmented Reality, Videoconferencing, Computer-supported Collaborative Work, Volume Rendering

1 Introduction and Related Work

1.1 Motivation

Computer-supported collaborative work (CSCW) is one of the evident application domains that Milgram et al.'s definition of Augmented Reality (AR) suggests [7]. Users of AR can see the real world, which provides a reference frame for their actions. They can see themselves and their collaborators, enabling smooth communication with non-verbal cues during the collaborative work. Moreover, a virtual space with synthetic objects is aligned with and superimposed onto the real world and shared among the users, thus changes made to manipulated objects during the collaborative session are distributed and immediately visible to all participants.

Unfortunately, this form of shared AR requires that the collaborators are sharing the same physical space, making it incompatible with remote collaboration over great distances. For remote collaboration, the state of the art is communication tools like audio/video conferencing and application sharing that help bridge the distance by displaying the remote parties' real environments. Application sharing provides a synchronized view into a 2D or 3D workspace, while a videoconferencing tool enables the use of natural conversation, facial expression and body gestures.

Unfortunately, standard solutions like Microsoft NetMeeting [8] enforce using separate tools for audio/video and application content, and are generally not suitable for 3D application sharing. In contrast, AR technology provides the opportunity of bridging the gap between seeing the remote collaborator and the effects of the collaborator's actions on the shared application content. In the following, we give a brief review of previous work on AR remote collaboration tools.

1.2 Video and audio conferencing

The incorporation of collaboration tools into virtual and augmented environments has already been examined by some researchers. An early work using a wearable computer to create a "wearable conferencing space" is described in the work of Billinghurst et al. [1]. Here the remote collaborators are represented by static virtual images superimposed over the real world and spatialized audio coming from their virtual locations. Users wear a video-see-through head-mounted display (HMD), and Internet telephony is used for audio transmission. Later Billinghurst and Kato [2] extended this work by presenting remote collaborators by live video images, which are attached to tangible objects that can be freely positioned in the user's AR space perceived through a video-see-through HMD with optical marker-based head tracking. With the help

of their virtual representations the conference parties become part of the local user's real environment. The users appear flat as the live video is texture-mapped onto a flat polygon. The latter 2D video texture techniques were extended by Prince et al. [10] into live 3D actors. Their system called 3D-Live uses 15 video cameras around the remote collaborator and captures the real person in real-time using a shape-from-silhouette algorithm. The system is able to generate a virtual image of the real person from an arbitrary viewpoint of the AR user, who can move around the physical object and thus the virtual actor. By perceiving the whole body of the remote user in 3D non-verbal communication like pointing gestures or body motion becomes possible.

1.3 Application sharing

Application sharing does not allow for non-verbal communication like a video stream does, however, it can be also a powerful tool for collaborative work. Perceiving and interacting with a desktop at a distant location provides easy access to remote applications for assistance and joint work. Early work using application sharing in 3D environments is described by Dykstra [4], where texture mapping techniques are used to run X applications on surfaces of objects in 3D virtual spaces. An AR extension of this work by Regenbrecht et al. [11] places interactive 2D desktop screens in an augmented 3D environment physically situated on a real office desk. Users wear video see-through HMDs and interact with standard 2D application windows that are attached to physical clipboards and can be placed at any position around the user. Ishii et al. [5] designed seamless, real-time shared workspaces for collaborative work. Their groupware system called TeamWorkStation combines two or more translucent live video images of computer screens or physical desktops using a video synthesis technique. They used two cameras for each collaborator: one for capturing the face, and the other for capturing the desktop image and the hand gestures. On these shared physical whiteboards users can draw images together and explain concepts to each other with hand gestures using real tools like a pen, brush etc. Users wear a head set for voice chat, which allows for a more personal and smoother communication between the collaborators.

1.4 Contribution

Despite the obvious potential of AR for bridging the cognitive gap between seeing a remote collaborator and the collaborator's actions in a shared application, only a limited amount of attention has been paid to this area. In particular, there is no videoconferencing tool using standard hardware that inserts 3D graphics overlays from a 3D application directly into the image of the remote participant.

In this paper, we present a system that achieves that goal. We describe an AR videoconferencing tool, which runs AR applications superimposed on live video background displaying the conference parties' real environment (Figure 1). Rather than pointing out a feature in a shared application with an abstract graphical widgets ("telepointer"), our AR system allows a user to simply point out a feature to the collaborator with one's own hand. The system merges the advantages of natural face-to-face communication of videoconferencing with AR interaction in a physical space via tangible objects.

Figure 1: A screenshot of our AR videoconf. system

2 System design

2.1 AR videoconferencing system

Our aim was a system similar to a conventional videoconferencing tool, so we opted against the use of an HMD. The use of HMDs socially precludes bidirectional videoconferencing, as participants will be seen wearing HMDs that cover a significant part of their face. Instead, a desktop AR setup based on a conventional desktop/screen workstation was chosen. Video acquisition and marker-based tracking (with the ARToolKit software [3]) is done simultaneously from a single camera placed at the top of the screen in a typical videoconferencing configuration (see section 6.1 for details). Application content is placed on tangible optical markers held by the user or placed on the physical desktop in front of the screen. Application manipulation is performed with a combination of markers and mouse/keyboard input. Since the physical space of local and remote participant is not shared, we provide two views per workstation: One showing the local user in a kind of mirror display, and one showing the remote user. Both views contain the same application objects and the state of objects is

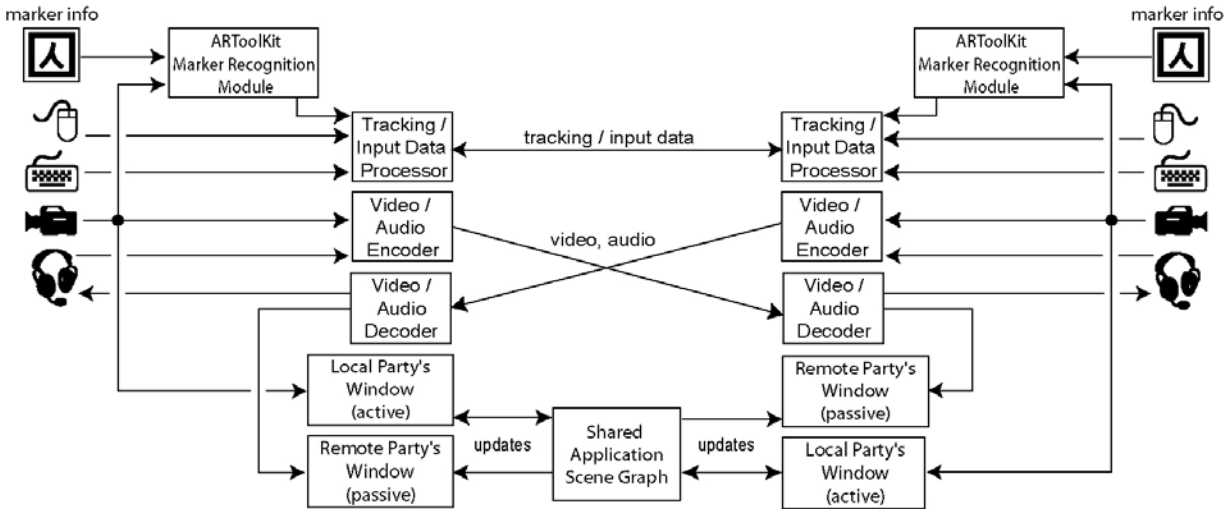

Figure 2: AR videoconference system architecture

synchronized, i.e. modifications will be reflected simultaneously in both views.

In general, application sharing is based on an exchange of user input and system response messages, while videoconferencing uses streaming transmission of video information. Typically, the video must be drastically compressed to meet the available bandwidth requirements, and as a consequence the resulting image quality is moderate. From these observations, we arrived at the following variants for information distribution:

1. The application runs locally including tracking and rendering, then the complete result (graphical output of the local window, i. e., the local participant with overlaid application objects) is streamed to the remote participant. This approach has the disadvantage that the delivered image has the worst possible quality because the graphical representation of application objects is compressed along with the raw video information.
2. The raw video stream is directly transmitted to the remote participant. The remote workstation performs tracking and rendering from this video stream in the same way it acts upon the local video stream.
3. Like variant 2, but tracking is computed locally, and the derived marker poses are transmitted along with the video stream. Rendering of application objects for the remote view is done at the destination machine.

Variant 3 avoids duplicating tracking efforts for local and remote participant as well as attempting to track from a compressed video stream (see section 3 for a detailed evaluation). While the application logic for sharing the experience is slightly more complex than variants 1 or 2, it was therefore selected as the best solution. Figure 2 shows an overview of the system. It is based on the *Studierstube* [12] platform and inherits its collaborative features based on a distributed shared scene graph from this foundation. In brief, video handling is added on top of an existing distributed application sharing framework.

Tracking is performed locally, and the result is both passed on to the local and the remote AR system (via the network). The local AR system also uses the input/tracking information to compute changes to the application's shared scene graph, which are then communicated to the remote application instance. In addition, the locally captured video stream is compressed and transmitted for display in the remote participants "remote view".

The shared collaborative application loaded into the system is represented by a distributed scene graph, which is initially loaded into all participating application clients and afterwards changes are distributed to update the scene. The remote parties on both sides act like slaves: they don't generate video streams or scene graph updates but only display video of the remote party and render the scene graph of the shared application, and interaction with the virtual objects is disabled as well.

There are the following main reasons why our system is superior to a video-only solution:

- significantly better image quality
- stereo rendering capability for the overlaid 3D objects, as this is not possible with compressed video streams
- no duplicate calculation of tracking data
- more precise tracking information as it is extracted from the higher-quality local video
- interaction capabilities with the virtual objects

Added values of our system compared to an interactive application sharing approach are the following:

- much higher speed and lower bandwidth
- no disturbance coming from competition for a single cursor or similar control resource
- no additional software needed for properly handling real-time video

We limit our discussion to an AR videoconferencing system that uses a single camera for both image acquisition and optical tracking.

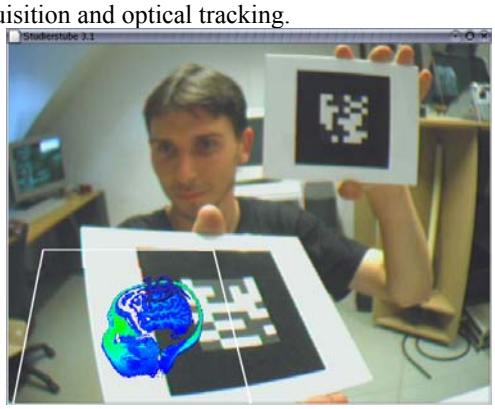

Figure 3: Tangible markers as application interface

2.2 Tangible interface for interaction

The *Studierstube* platform supports multi-user interaction on various configurations, including our desktop-based setup. The interaction and object manipulation is done with tangible optical markers (see example in Figure 3) that have been assigned various, application-specific functions, thus connecting the physical world and the virtual space. This interface allows for a more natural and intuitive way of interaction compared to three-dimensional virtual menus and pointers that turned out to be particularly clumsy in desktop-based Mixed Reality applications.

In our system design tracking data for the interaction props and application-specific tracked objects comes from optical markers and keyboard/mouse commands. However, the underlying architecture dealing with tracking data supports a wide range of other tracking and interaction input devices, such as magnetic, ultrasound or infrared tracking systems as well as 2D/3D input devices including data gloves and graphics tablets, which allows for experimenting with various ways of interaction in the applications.

3 Marker Recognition in Compressed Video

To verify our considerations regarding the quality of tracking from compressed video, we conducted an experiment comparing ARToolKit tracking quality of uncompressed to compressed video. While it is obvious that tracking quality will suffer, the amount of degradation was not clear.

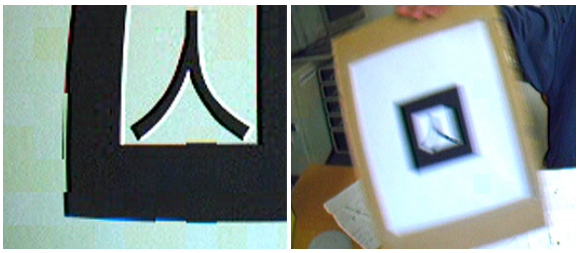

Figure 4: Marker recognition problems in compressed video: quantization effects destroying edges (left), motion blur (right)

There are two main reasons for degradation of the marker recognition:

1. The video codec uses image blocks to encode the frames. Pixel color values are quantized per block, therefore there may be jumps in the color of adjacent blocks creating undesirable new sharp edges. Moreover, if the image transmission latency is too big, these blocks may get shifted relative to each other, therefore the edges of the original marker border will be lost, causing the marker recognition system to loose the marker. The left image of Figure 4 illustrates this effect.
2. If the user moves around the marker too quickly, the image gets motion-blurred (see the right image of Figure 4 for an example). This effect again causes the marker recognition system to lose the marker, as it cannot recognize the pattern inside the marker border anymore. It is especially a significant problem if the marker is far away from the camera as the marker pattern appears as a rather small, fuzzy image in the video stream, the quality of which is further degraded by video compression.

Figure 5: Markers used in the experiment: (left) "Kanji", (middle) "Hiro", (right) "Point"

We compared the tracking performance from compressed vs. uncompressed images with the three markers shown in Figure 5: "Kanji" is a clear, simple

marker pattern, which is easy to recognize even from a larger distance. “Hiro” is more complex; if the image is too small, the pattern image gets easily unclear, rendering it very difficult to be recognized. “Point” contains a small square, which can get quickly lost in a bad-quality video stream. We registered 80mm x 80mm markers with a default threshold value of 100. In the ARToolKit software the “confidence” value describes the marker recognition quality, i.e. how “confident” the program is about the recognized pattern and its position and orientation. Its values range from 0.0 (lost marker) to 1.0 (absolutely certain). We measured marker recognition quality based on three different aspects: average marker loss (the average number of video frames where the marker was lost), maximum confidence value for the best achievable quality ($0.0 \leq \leq 1.0$), and average confidence value for the average quality during the measurements ($0.0 \leq \leq 1.0$). A marker is “lost” if it cannot be recognized at all.

For both the uncompressed and compressed video 12 measurements were made respectively: we held the three markers consecutively in front of the camera at four different distances: 250, 500, 750 and 1000mm. The measurement results are shown in Figure 6. While keeping the distance constant we moved around the markers for 30 seconds, trying to cover a wide range of different poses.

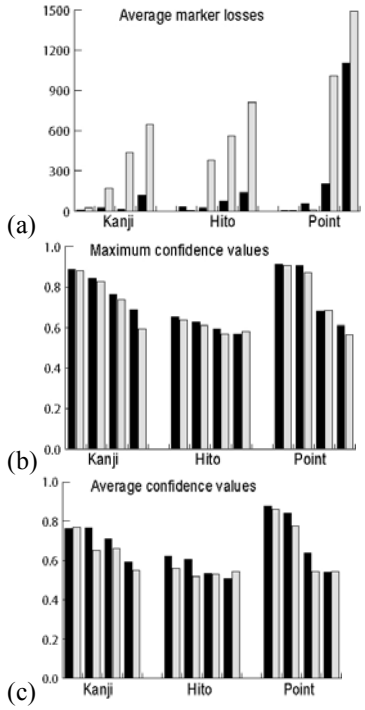

Figure 6: Measurement results

The black columns in the figures represent markers in an uncompressed video stream; the gray columns stand for markers in a compressed video stream. Diagram a) shows the average number of frames where the marker was lost, b) shows the maximum confidence value reached, and c) illustrates the average confidence value measured for each marker respectively.

The results suggest that at shorter distances uncompressed and compressed videos perform similar, while at larger distances, the compressed video performs a lot worse compared to the uncompressed one, especially because the markers get lost frequently and it is also difficult to find them again. The measurement results suggest that processing the optical markers in the uncompressed frames and sending the data over in a separate channel allows for more reliable tracking.

4 Implementation

4.1 Hardware setup

One of our major goals was to keep a simple, low-cost setup that is affordable for everyday use. We wanted to avoid solutions using expensive tracking systems, high-bandwidth networks or costly proprietary conferencing software. Our desktop-based augmented reality setup for each client consists of a 1.5GHz PC with 512Mb RAM and NVIDIA Quadro4 graphics card, a flat-panel LCD monitor, a lightweight PointGrey FireWire camera flexibly mounted on the top of the monitor and pointing at the user and numerous optical markers. We use the aforementioned ARToolKit software for getting tracking information from the optical markers. The markers can be easily made at home. For optionally viewing the 3D scene in stereo CrystalEyes shutter glasses are used, but in this case we need to replace the flat LCD display with a CRT monitor.

4.2 Videoconferencing module

The videoconferencing module is based on the OpenH323 software [9], which is an open source protocol stack incorporating a set of communication protocols developed by the International Telecommunications Union (ITU) and used by programs such as Microsoft NetMeeting and equipment such as Cisco Routers to transmit and receive audio and video information over the Internet. The subset of communication protocols we chose relies on the call control and media control protocols and the H.261 video compression standard for low-bandwidth video transmission. The video stream of the FireWire camera

images gives a resolution of 320 x 240 pixels / frame with a frame rate of 30 fps, which is encoded using the Common Intermediate Format (CIF) standard providing 352 x 288 pixels / frame and a frame rate of ca. 10-15 fps. This frame size does not allow for displaying fine details in the images, however, the bandwidth demand of the system is low. The actual required bandwidth depends on the speed of the motion in the camera image but even in the worst case it does not exceed 150 kbps.

5 Applications

We chose a volumetric rendering application for demonstrating the capabilities of our tool since the users can jointly and interactively examine a shared set of volumetric data, which nicely illustrates the collaborative features of our system and serves as an appealing visualization software in the medical and geological domain. We are using Systems in Motion's SimVoleon library [8], which visualizes volumes by using 2D texture slices. SimVoleon loads VOL-format files and can handle arbitrary volume dimensions, i.e. non-power-of-two dimensions and different dimension along the axes. It supports mixing of volume data and polygonal geometry, picking operations on voxels, and changing of the transfer functions in real time.

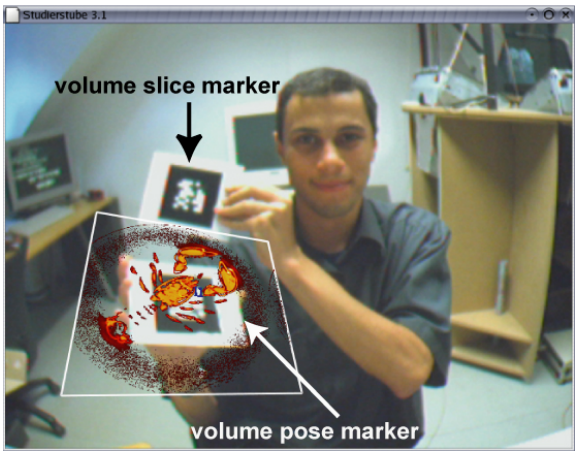

Figure 7: Volume rendering application running on top of our AR videoconferencing system

In our application the collaborators independently choose a volume that they can examine as a whole or in slices. Different settings of the volume like e.g. axis, interpolation, color maps etc. can be adjusted. Optical markers are used to move the volume as a whole and to allow the user to view arbitrary slices of the volume. In the screenshot of Figure 7 the user positions the whole volume in his personal workspace using the marker in his right hand while moving through the volume slices with the marker in his left hand. If the markers are removed from the working volume, the associated settings stay unchanged allowing for ordinary non-verbal communication in the video stream like gesturing, pointing out features and so on.

The system makes it possible to remotely and interactively explore and discuss data sets. Potential users of such an application are e.g. surgeons examining parts of the human skeleton acquired by Computer Tomography or geologists studying processed seismic data sets acquired by onshore/offshore surveys using a series of geophones.

We have already presented our system at two public demos: at the ISMAR 2003 conference in Tokyo and at a demo session at the SPE Forum Series 2003 in France. Anecdotal experience shows that in general users find the interface of our system rather intuitive and self-explaining, the communication with virtual objects and tangible markers with the videoconference background appears natural, and last but not least the whole application has a remarkable fun factor.

6 User Study

We asked a number of people in our group to test our system. They were asked to try out the AR application presented in the previous section with four different setups. After the try-out they were requested to fill out a simple questionnaire with questions about suitability for collaborative work, smoothness of communication during the collaborative session, and general impressions about performance and speed. In the following section we will describe our findings and experience based on answers on the questionnaires, oral discussions and anecdotes.

6.1 Personal workspace

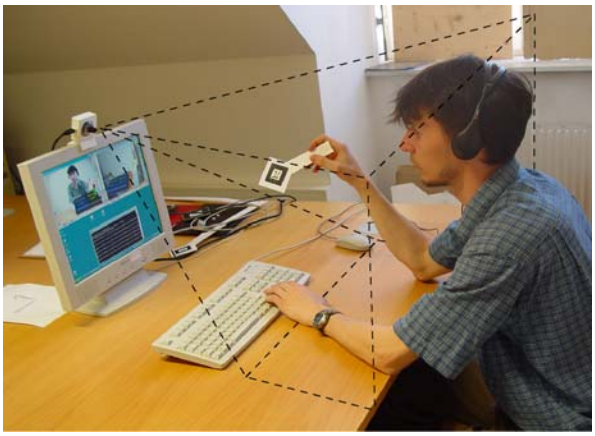

Figure 8: Ergonomic workspace arrangement

A number of suggestions helped us to incrementally improve the workspace layout. We learned that in a conference application a convenient and ergonomic working environment is of high importance. Users need to have enough space for gesturing to the communication partners and for placing and moving around the markers and interaction props for conveniently using the AR applications. We placed the video camera on the top of a tiltable flat panel monitor, which we moved a bit away from the user so that she does not simply become a "talking head" in the video but her hand gestures can be clearly seen. In addition, she gains additional desktop space for marker manipulation. Figure 8 illustrates the arrangement of our experimental setup. We found the following user preferences concerning workspace arrangement:

- The conference party should be visible in the image from the waist to the top of the head for effective gesturing.
- Sufficient desktop space in front of the user has to be available so that the markers don't need to be held in hand all the time but can be put onto the table and moved around.
- The optical markers have to be large enough ($\geq$ 8cm) to be reliably recognizable even from a larger distance and quicker motion. In addition, light sources have to be positioned appropriately to avoid reflections on the markers while still keeping the workspace sufficiently illuminated, as both factors may significantly decrease recognition confidence.

6.2 Software setup

The four setups described in the following were tested in our user study. All of the setups used the Microsoft NetMeeting software [8] for voice communication. The hardware setup stayed unchanged.

Setup 1. *AR videoconferencing tool with video background acting like a mirror.*
Setup 2. *AR videoconferencing tool with a video background displaying the video exactly as it arrives from the camera.*
The first two setups represent the same implementation of the AR videoconferencing system with two slight differences: in the first setup the video stream of the camera is horizontally mirrored, essentially turning the window into a mirror, while in the second setup no mirroring is used. The difference between setup 1 and 2 concerns only the local view, the remote party's view was never mirrored.

Setup 3. *Shared, local 3D application window with no video background.*
This setup was created to determine how conventional 3D application sharing competes with our specifically designed AR videoconferencing. While there no video background was provided, users could share viewpoints, and of course all modifications to the application objects were shared.

Setup 4. *Shared, local 3D application window with no video background, plus NetMeeting used for video-, sound- and text chat-based communication.*
Finally, we added conventional videoconferencing tools in the shape of NetMeeting to a non-video version of the 3D application. NetMeeting offered users various collaboration tools like text-based chat, voice and video conversation as well as a shared whiteboard. This setup examines whether users prefer a video stream rendered in a separate window or in the background of the AR application. It also focuses on the use of collaborative communication tools.

While we did not attempt any quantitative evaluation, we asked the users to evaluate the different setups from the following aspects:

- How suitable are the AR applications with the various setups for productive collaborative work?
- How smooth was the communication between the users during the collaborative session (i.e. frequency of misunderstandings and interaction conflicts)?
- How was the performance of the tested system? How much latency was there in the video and in rendering of the shared scene graph?

Our study resulted in the following outcomes:

1. Application sharing is practically unusable for concurrently modifying a shared AR application object because of the significant image update latency of the video stream. However, users engage in a natural dialog, taking turns at making modifications to a single object of interest.
2. A really important finding is that the AR applications need to have some relations to the video background. In our application objects are attached to markers or physical objects for quick manipulation and to provide a shared physical geometric frame of reference to the users. We also attempted to simply overlay extra, auxiliary "floating" application objects over the video stream, which was found more disturbing than helpful, in particular if the conference party's face and gestures are occluded. In these cases users

preferred the setup with the AR application having only a plain background and the video stream of the conference party in two separate windows. On the other hand, some of the users felt that they can keep their partner better in sight if the video is in the background of the virtual objects, as they would rarely follow the video stream in a separate window otherwise. This finding resulted in the workspace layout presented in section 6.1, which allows for essentially avoiding user / object occlusions without breaking the "shared real + virtual space" illusion.

3. The application elements should be arranged in a way that they do not cover too much from the video background, especially the collaborator's face. This can be achieved by binding the whole AR application grouping all appearing objects to one of the tangible optical markers so that they can be positioned initially at an appropriate 3D location to avoid annoying occlusions and so that it can freely moved around if they block the view.
4. Without exception users preferred the setup where the video background served as a mirror, i.e. the video captured by the camera was horizontally mirrored. They considered it natural and intuitive, while the non-mirrored video stream was described as unnatural and sometimes misleading.

7 Conclusions and Future Plans

We presented a low-cost, desktop-based AR videoconferencing tool for remote collaboration. We justified our system design concepts with measurements and an initial user study, and demonstrated a collaborative application built onto this framework. Our future plans include the following:

- Making the system capable of handling several conference parties as currently only two users are supported. This would raise some problems in communication (e.g. who speaks and when in the audio channel) and in implicit object locking, therefore a collaboration protocol needs to be designed and utilized.
- Experimenting with other codecs for higher image quality and better compression to further reduce required bandwidth or to have a larger frame size.
- Distributing viewpoint changes would merge the advantages of application sharing and the AR videoconferencing setup, as the remote user would see exactly what the local user sees, not only get the same video and scene graph. This would enable more sophisticated communication and assistance, for instance interaction elements or important objects could be simply zoomed onto for explanation instead of moving them closer to a fixed user viewpoint.

8 Acknowledgements

This system was sponsored by the Austrian Science Fund *FWF* (contract no. Y193) and the European Union (contract no. IST-2001-34204). contract no. IST-2001-34204). Systems in Motion kindly provided us with their Volume Rendering library SimVoleon. Our thanks go to Gerhard Reitmayr for his help and to our colleagues for their valuable comments and constructive criticism on the system.

9 References

[1] M. Billinghurst, J. Bowskill, M. Jessop, and J. Morphett, "A Wearable Spatial Conferencing Space", *Proceedings of ISWC'98,* IEEE Press, pp. 76-83

[2] M. Billinghurst and H. Kato, "Real World Teleconferencing", *Proceedings of the Conference on Human Factors in Computing Systems (CHI 99)*, Pittsburgh, USA, May 15th-20th, 1999.

[3] M. Billinghurst, H. Kato, S. Weghorst, and T.A. Furness, "A Mixed Reality 3D Conferencing Application", T*echnical Report R-99-1 Seattle: Human Interface Technology Laboratory",* University of Washington, 1999.

[4] P. Dykstra, "X11 in Virtual Environments", *Proceedings of IEEE 1993 Symposium on Research Frontiers in Virtual Reality,* San Jose, CA, USA, Oct. 25-26, 1993.

[5] H. Ishii, M. Kobayashi, and K. Arita, "Iterative Design of Seamless Collaboration Media", *Communications of the ACM,* Vol. 37, No.8, August 1994, pp. 83-97

[6] http://www.sim.no/products/SimVoleon/

[7] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino, "Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum", *Proceedings of Telemanipulator and Telepresence Technologies,* SPIE 2351, 1994, 282-292.

[8] http://www.microsoft.com/windows/netmeeting

[9] http://www.openh323.org

[10] S.J.D. Prince, A.D. Cheok, F. Farbiz, T. Williamson, N. Johnson, M. Billinghurst, and H. Kato, "Real-Time 3D Interaction for Augmented and Virtual Reality", SIGGRAPH'02 Sketches and Applications, San Antonio, USA, July 21-26,2002.

[11] H. Regenbrecht, G. Baratoff, and M. Wagner, "A Tangible AR Desktop Environment", *Computers & Graphics,* Vol. 25, No.5, Elsevier Science, Oct. 2001, pp.755-763.

[12] D. Schmalstieg, A. Fuhrmann, G. Hesina, Zs. Szalavári, M. Encarnação, M. Gervautz, and W. Purgathofer, "The Studierstube Augmented Reality Project", *PRESENCE - Teleoperators and Virtual Environments*, MIT Press., 2002.