EG UK Theory and Practice of Computer Graphics (2005)
M. McDerby, L. Lever (Editors)

Ubiquitous Virtual Reality: Accessing Shared Virtual Environments through Videoconferencing Technology

Thies Pfeiffer[1] and Matthias Weber[2] and Bernhard Jung[2]

[1]SFB 360, University of Bielefeld, Germany
[2]International School of New Media (ISNM), University of Lübeck, Germany

Abstract

This paper presents an alternative to existing methods for remotely accessing Virtual Reality (VR) systems. Common solutions are based on specialised software and/or hardware capable of rendering 3D content, which not only restricts accessibility to specific platforms but also increases the barrier for non expert users. Our approach addresses new audiences by making existing Virtual Environments (VEs) ubiquitously accessible. Its appeal is that a large variety of clients, like desktop PCs and handhelds, are ready to connect to VEs out of the box. We achieve this combining established videoconferencing protocol standards with a server based interaction handling. Currently interaction is based on natural speech, typed textual input and visual feedback, but extensions to support natural gestures are possible and planned. This paper presents the conceptual framework enabling videoconferencing with collaborative VEs as well as an example application for a virtual prototyping system.

Categories and Subject Descriptors (according to ACM CCS): J.6 [Computer-Aided Engineering]: Computer-aided design I.3.6 [Computer Graphics]: Interaction techniques I.3.7 [Computer Graphics]: Virtual reality

1. Introduction

Videoconferences were first developed to enhance telephony systems with video functionality. These systems are now in use since decades, one of the first was Picturephone developed by AT&T [AT&T03]. But these early systems were big, expensive, and inflexible and therefore remained in a small niche for a long time.

Today, with the emergence of inexpensive broadband Internet connections and the availability of webcams of decent quality, videoconferences are finally receiving their deserved attention. Most of the recent instant messaging services provide videoconferencing facilities on an increasing range of platforms, even extending to mobile devices, such as notebooks, handhelds or smartphones. Remote natural face-to-face communication between humans is no longer a curiosity. Standardised conferencing systems have made videoconferencing available to a wide range of users, on various platforms, with little costs.

On the other side there are many solutions for remotely accessing technical systems, starting with simple remote shells or remote desktop environments. Some allow several users to collaborate, e.g. using shared whiteboards, shared applications, or complex collaborative VEs. Some provide natural communication facilities. Some are low-cost, some are ubiquitous. But as far as we know, only speech applications running on telephony servers provide a low-cost ubiquitous access to technical systems using natural communication - and they are currently restricted to speech only.

We believe that videoconferencing has the potential to extend from human face-to-face communication to a natural communication with remote technical systems using both speech and visual information, e.g. gestures or facial expressions. In this paper we present a concept for the integration of videoconferencing facilities into a VE. One advantage of videoconferencing is, that it works in both directions. Remote users can call the VE and interact with it in a videoconferencing session. Conversely, the call can also be initiated from within the VE, for example calling a remote user to provide some information or ask for advice. In terms of the virtuality continuum introduced by Milgram and

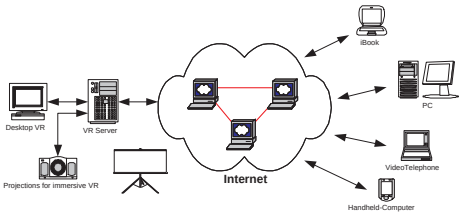

Figure 1: *Using the videoconferencing extension, the VE on the server can be accessed ubiquitously from a diverse range of clients relying on standard software only.*

Kishino [MK94] this quality shifts the system from the sole VR field to a mixed reality scenario as real and virtual worlds get connected and mixed. This leads to an extension of conventional VR systems which are normally not that tightly coupled to videoconferencing software. As a consequence, such an extension has an ubiquitous character as people can see and interact with VEs everywhere, even through mobile devices (see Fig. 1).

Examples of applications include but are not limited to:

- Information Desks, e.g. for exhibitions, museums, cinemas, etc.,
- Technical Support, e.g. guiding through a design process,
- Educational Systems,
- Unified Messaging services, appointment scheduling,
- Interactive Entertainment, e.g. games, quiz shows, interactive television.

In some of these scenarios the system would reactively wait for incoming calls, but in others, such as technical support, it would be advantageous if the session could be initiated from both sides: A customer who has problems assembling a cupboard could call the support system of a furniture store and get an audio-visual hands-on walk-through. Or, an interior architect actively working on a project in her CAD system could seamlessly call her clients and get their feedback on crucial design decisions. Moreover, if an intuitive interface was supplied, it could also be possible that the clients directly interact with the system, e.g. changing the shade of some colours or move some furniture by means of spoken language instructions or gestures.

While in these examples a human user initiates the session, cases where a technical system itself takes the initiative are also not far fetched. A Unified Messaging service could provide a virtual avatar calling interested users and remind them interactively of their scheduled appointments.

After having presented related work (section 2), section 3 goes into detail on our concept of integrating videoconferencing facilities in a VE. Section 4 shows an implemented example application where a user engaged in a construction task calls a remote expert asking for advice. The expert takes over and explains assembly procedures to the caller. In section 5, we describe the implementation of the system. We discuss our approach in section 6 and finally conclude in section 7.

2. Related Work

As pointed out in the Introduction, videoconferencing is now getting more and more popular as the Internet is evolving, network bandwidths are increasing, and codecs are getting better in terms of quality and compression ratio. Several videoconferencing protocols exist for conferencing on the Internet. These include not only proprietary protocols like the one used by Skype [Sky05] but also standard protocols, such as H.323 adopted by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) or the Session Initiation Protocol (SIP) adopted by the Internet Engineering Task Force (IETF), for both see [Wil00]. The availability of standard compliant software (like NetMeeting, GnomeMeeting, etc.) on many platforms makes videoconferencing appealing for projects such as VirtualSchool [ICN*00]. Free implementations of these protocols are available and are already used. An example using augmented reality (AR) is [BFSK03] which is based on Studierstube [SFH*02].

With the growing adoption of videoconferencing via the Internet, several systems have been built that connect people by means of shared VR environments. Such systems combine videoconferencing and VR and often provide the possibility to cooperatively work on a shared task. One example for such a system is AliceStreet [Ali04] which shows live videos of all participants arranged around a conference table. The participants can then, for example, discuss about PowerPoint presentations which are shown on a projection wall. AliceStreet uses the H.323 protocol. Another example is Coliseum [BBT*03] which acquires a 3D-representation of the user and arranges these 3D-representations of conference participants around a conference table. Coliseum employs its own UDP-based media streaming protocol. Other systems comparable to Coliseum have also been proposed.

Some AR-based systems like the ones developed at Studierstube [SFH*02] combine videoconferencing with virtual objects. The Studierstube system needs a special client-software on every conference participant's device; this enables processing of camera images in high quality, i.e. without loosing quality due to image transmission via the videoconference protocol. After receiving information from the optical trackers, the image itself is transmitted via the H.323 protocol and the tracking data is sent through TCP/IP or multicast. Another related AR-based system was developed by Billinghurst and Kato [BK00]. They show the virtual images of the conference collaborators attached to real-world cards which are the optical markers for the AR system.

E-learning systems like VirtualSchool [ICN*00] connect students and teachers via videoconferencing in order to en-

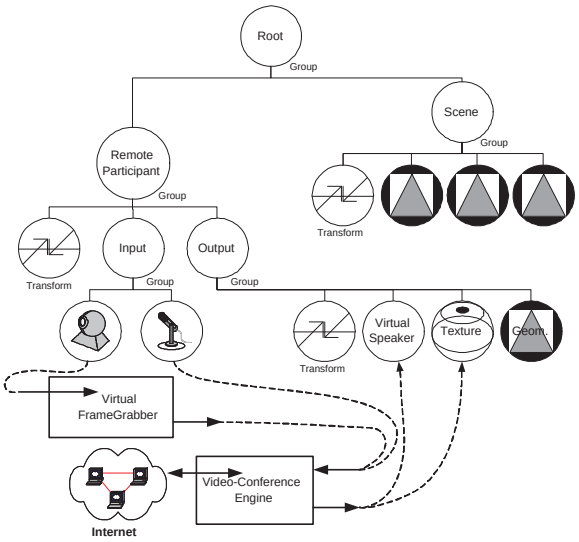

Figure 2: *A scenegraph (OpenInventor notation) showing a possible embedding of the videoconferencing system in VR applications. The fields of the videoconference-engine are connected to nodes in the scenegraph, providing or accepting audio and video feeds.*

hance cooperation/collaboration in their classes. Another interesting project is Virtual Harlem [PLJ*01] which shows New York's Harlem in the 1920's, though it is restricted to audio conferencing only. Students can connect to this VE and discover it collaboratively, optionally guided by a teacher.

As far as we are aware, there is no other system or concept providing an ubiquitous interactive audio-visual two-way access to VR systems. We achieve this building on standard videoconferencing protocols and videoconferencing clients readily available on multiple platforms. In addition, none of the existing systems makes further use of the videoconferencing functionality to support the remote users' interaction with the shared VE via natural communication modalities such as speech and gesture. A detailed elaboration of the conceptual background of our approach is given in the next section.

3. Concepts

In this section we will describe our proposal of a videoconferencing interface to a VE. A major requirement was that it should integrate smoothly into common VR applications. Therefore we adopted the common scenegraph based architecture of current VEs (see Fig. 2). The presentation will be guided by the following questions:

1. Where is the videoconferencing gateway to the VE located?
2. How is the VE presented to the remote participant?
3. How is the remote participant represented in the VE?
4. What interaction models are supported?

3.1. The Gateway between Reality and Virtual Reality

The videoconferencing-engine is at the heart of the videoconferencing interface to VR. It has no visual or auditory representation itself, but provides a set of fields, either offering or accepting video textures, audio signals, and data streams. Fields are members of nodes (e.g. in the scenegraph) that contain arbitrary variable data. They can be connected to other fields of the same type. Connected fields get updated automatically whenever the source field changes its value. The field concept is offered by most of the scenegraphs, with its most prominent representative being OpenInventor [SGI05].

To provide scenegraph nodes with textures, audio and data, the fields of the videoconferencing-engine can be connected to fields of appropriate scenegraph nodes. These internal connections going from the videoconferencing-engine to the VE define the internal interface. At the same time the engine operates as server accepting videoconferencing sessions and allowing internal processes to initiate calls from within the VE. It is also responsible for managing the lifecycle of established connections and initiates clean-up processes when the connection has been terminated. Mediating between this external and the internal interface the engine defines a gateway between Reality and VR.

3.2. Bringing the VE to the Remote Participant

The remote participant should be able to access the VE at least by visual and auditory means. For the transmission of audio and video from the VE to the client, a virtual microphone and a virtual camera are used (see the nodes below the *Input* group node in Fig. 2). These are virtual devices that are represented as separate nodes in the scenegraph, which can be positioned ad libitum. In most cases, both will be placed below a common group node, together with a visual representation of the location of the remote participant in the VE.

The concept of a virtual camera is common in VEs. In general, it is a node defining the position and orientation of a view to be rendered. The difference here is that the view is not rendered to a screen (although it can be, so that both participants share the same view), but to a virtual framegrabber device which serves as video source for the videoconferencing-engine. The virtual microphone differs only in that it is not graphics that is rendered, but audio.

3.3. Representing the Remote Participant

With the technology presented so far the remote participant can only passively experience but not interact with the VE. While this might be sufficient for content-delivery services,

for most applications it will not. To allow for interaction between the VE or local participants and the remote participant, the latter has to be represented in the VE.

A geometrical shape, associated with the remote participant and located in the scenegraph, just beside the virtual camera and the virtual microphone, is a simple first approach to this problem. This way, the VE and the other participants are at least made aware of the presence of the remote participant. If the remote participant does provide a video stream, the feeling of presence will be improved significantly. To accomplish this, the videoconferencing-engine exports the incoming video via special video textures. These can be mapped to geometry in the VE and the remote participant can thus be visualised. The position of the visual representation is thereby independent of the position of the virtual camera. It can be presented several times at different locations or, if not appropriate, be completely dropped.

Together with a visual representation it is also possible to have an auditory one. The signals received from the remote participant can either be sent directly to the sound card, as in conventional videoconferencing systems, or it can be routed to a sound node in the scenegraph. This makes the voice of the remote participant appear as coming from a specific position, which is especially important for immersive VR settings and strengthens the perceived presence of the participant.

3.4. Interaction with the VE

With their video and audio transmission capabilities, videoconferencing systems naturally support human-to-human communication in VEs. Besides the issues of presentation and representation discussed so far, they can also serve as a medium for interaction with the VE.

Some VEs, such as [JHW98], offer functionalities that allow the user to modify the scene content via natural language instructions, either spoken or typed. In such cases, our framework also enables the remote user to verbally interact with the VE. To support this, the incoming audio signals may optionally be routed through a speech-recognition system. If the audio signal is too noisy and speech-recognition not appropriate, the participant can fall back to a textual chat protocol which is also supported by most videoconferencing standards. In either way the incoming instruction can then be processed further by the VE.

In addition, the participant can also interact visually with the system. This could involve face recognition to identify specific participants or the recognition of facial expressions to extract emotional states. Interpreting gestures seems quite demanding. A promising approach is applied in recent webcam based video games such as EyeToy [Son05]: here video signals of the participants are fed back into the content channel to allow the participants to synchronise their gestures with the content. Another approach could be directing the video camera of the remote participant to the screen instead of the participant. That way she can point to entities on the screen which are then extracted from the video. All these solutions would require no special support within the clients. Instead they rely on heavy image recognition on the server side.

Figure 3: *The designer works on a model of a toy aeroplane. He has some difficulties and asks the expert via videoconferencing about the positioning of the tail unit. (Desktop based VR application / Linux)*

Having described these concepts, we will provide a scenario to illustrate their applicability in a VE for construction tasks.

4. Application to Construction Tasks

To demonstrate the interconnectivity of videoconferencing and VEs we implemented a prototype system on top of an existing VE application for interactive construction tasks [JHW98], which has recently been extended to support face-to-face videoconferencing [WJ04]. In this application the user can use a toy construction kit to assemble certain models, e.g. an aeroplane, via speech and mouse interaction.

In Figure 3 we see our first author, Thies, sitting at his desk at the University of Bielefeld. He has nearly finished building the aeroplane. The task remaining is to connect the tail unit with the fuselage area of the plane.

Thies 'Connect the tail and the fuselage.'
... nothing happens
Thies 'Connect the red block to the fuselage.'
... still nothing happens. After trying several times, he decides to ask the second author, Matthias, for advice, an expert on constructing toy aeroplanes. Unfortunately Matthias is at work at the ISNM in Lübeck, which is located approximately 300km away. So Thies calls him using videoconferencing from inside the VE.

Figure 4: *Using the videoconferencing interface the expert can share the view of the designer. He can communicate with the remote participant and guide him using voice communication or instruct the system directly using its speech interface. (Microsoft's NetMeeting / Windows XP)*

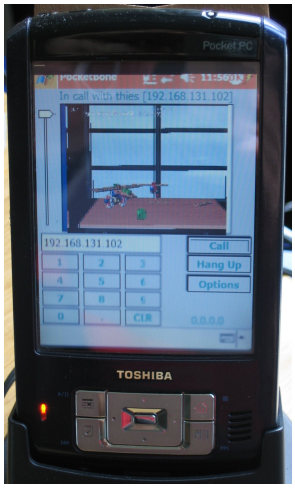

Figure 5: *The same session seen from a handheld device. Unfortunately the OpenSource videoconferencing software used (PocketBone / Pocket PC) is in its early beta stages. While audio is already working quite well, only the lowest video resolutions are supported. But the scene (although mirrored) can still be recognised and instructions via speech can be given.*

Matthias 'Hi, Thies! How are you?'
... after some negotiation ...
Matthias 'Ah, I get it. Well, I'll show you.'
Matthias 'Put a bolt bottom-up through the near hole of the bar.'
... the system reacts and puts the blue bolt in the right position.
Matthias 'Connect the tail with the bolt.'
... the system reacts and finally the aeroplane model is completed.
Thies 'Oh, thank you very much, bye Matthias!'
Matthias 'Bye!'

So Thies has finally completed his work on the aeroplane. He had decided to use the videoconferencing capabilities to ask Matthias for advice. Matthias could see the current situation in the VE on his laptop (see Fig. 4), but he could have also used his PDA (see Fig. 5). He communicated with Thies using speech which is concurrently routed to the same speech-recognition system that Thies used to instruct the VE before. The system is restricted to simple instructions related to the domain of construction tasks and skips their small-talk and negotiation. It actually skips some of the instructions, too, due to poor speech recognition rate, so in reality they had to be repeated several times - this is a situation where the textual input comes in handy. And while we are at confessing, the complete dialogue is actually handled in German and only transcribed to English for the purpose of this paper.

After having had a glance at the possibilities of using videoconferencing as a medium for natural interaction with VEs, we will go into detail on the realisation in the next section.

5. Realisation

Our implementation of a videoconferencing interface to a VE extends a face-to-face videoconferencing module [WJ04] built on top of the OpenSource implementation of the H.323 videoconferencing protocol, OpenH323 [Pos04]. It is realised as a module for the VR framework Avango [Tra01], which is based on SGI's OpenGL Performer. This framework has already been adapted to the cooperative construction task scenario of the Collaborative Research Centre 360, *Situated Artificial Communicators*, at the University of Bielefeld, in particular the CODY Virtual Constructor [JHW98], an intelligent VE for knowledge-based virtual prototyping with a natural language interface.

5.1. Intelligent VE with CODY

Figure 6 shows an abstract view of the architecture of our realisation. The multi-agentsystem to the right implements the knowledge-based system for virtual prototyping. Agents specialised in parsing, semantic analysis and reference resolution interpret German natural language instructions. Other agents serve as knowledge-bases containing conceptual and geometric information about parts, assemblies, and their respective connective regions. The most important agents are those responsible for dynamic conceptualisation of the current state of affairs, interpretation of natural language instructions, and simulation of assembly steps.

The visualisation and interaction handling is accom-

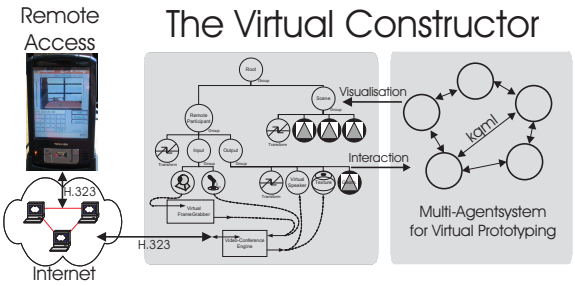

Figure 6: *The multi-agentsystem to the right represents knowledge about natural language processing and virtual prototyping. The VE in the middle (see Figure 2) is responsible for visualisation and interaction handling and providing the videoconferencing interface. To the left a PDA is shown with an active connection to the virtual prototyping application running in the VE.*

plished by the VE running on top of the Avango framework. It synchronises its representation of the current scenery with the knowledge-bases via a TCP/IP link to the multi-agentsystem. This is a two-way process; if the parts are modified in the VE, their new positions are updated in the knowledge-bases as well. The VE is also responsible for speech and gesture processing for manipulative actions. Complex natural language instructions are forwarded to the corresponding agent in the multi-agentsystem.

The videoconferencing interface is also integrated in the VE. It allows remote clients to connect to the VE over the Internet. Their video signal is rendered onto a visual representation of a monitor in the VE. Their audio-signal is rendered to the 3D sound-server VSS [Gou03] and routed through the speech recognition system, to allow for a speech based interaction with the VE via videoconferencing. In addition, a data channel is used to transmit typed text, which is rendered to screen and fed into the knowledge based construction system.

The communication from the system to the remote user facilitates position dependent audio, speech synthesis, and the video display of the virtual camera.

For a more detailed description of the CODY Virtual Constructor we refer to [JHW98].

5.2. The Videoconferencing Interface

The objectives for the implementation of the videoconferencing interface were as follows:

- Support for audio, video and data transfer
- Support for a range of clients, esp. embedded and mobile devices
- Low latencies, slender bandwidth consumption

In [WJ04], Weber and Jung compare different approaches to videoconferencing, e.g. MPEG2, in VE based on Open-Source software. They conclude that the videoconferencing standard H.323 and its implementation in the OpenSource library OpenH323 meets the aforementioned requests best. The provided features such as peer-to-peer connectivity, platform portability and the open architecture simplify its usage additionally. Together with its free availability this convinced us to base our videoconferencing interface on the H.323 protocol.

Standard	Frame size	Bandwidth	Impl.
H.261	176x144 (QCIF) / opt. 352x288 (CIF)	$\geq$ 64 kbps	yes
H.263	176x144 (QCIF) / opt. 128x96 - 1408x1152 (SQCIF - 16CIF)	$\leq$ 64 kbps	yes
H.264	resolution-independent	$\geq$ 20 kbps	no

Table 1: *Available video codecs, state of implementation.*

Standard	Audio bandwidth	Bandwidth	Impl.
G.711	3.1kHz	56/64 kbps	yes
G.722	7kHz	48/56/64 kbps	yes
G.723/ G.726	4kHz	5.3/6.3 kbps	yes
G.723.1	3-4kHz	5.3/6.3 kbps	no
G.728	3.1kHz	16 kbps	no
G.729	4kHz	8/13 kbps	no
Speex (not part of H.323)	4-16kHz	6-24 kbps	yes

Table 2: *Available audio codecs, state of implementation.*

The H.323 protocol comes with a bundle of supported audio and video codecs. They are the crucial elements of a videoconferencing protocol, as they define the quality and robustness of the connection. An overview of the supported codecs is given in tables 1 and 2 for video and audio respectively.

The frame sizes of the video codecs are generally given as multiples of the Common Intermediate Format (CIF), which is defined as 352x288. For convenience, the table shows both notations. As can be seen, the supported frame sizes range from 128x96 to 1408x1152, although not all of them are mandatory. The effective resolution used is negotiated by the participants and may depend on the capabilities of the clients and the properties of the connection. Small devices, such as PDAs, will prefer low resolutions, as their screen-size would not support more than 640x480 or even less. During a videoconferencing session the codecs and resolutions used do not need to match between the participants. A high quality codec can be used for sending the VE to the remote participant, while only a low quality transmission is needed for visualising the participant in the VE. Particularly H.264, though

not supported by OpenH323 at the moment, supports very high quality video, but its bandwidth consumption can go up to 4Mbps or even more. The bandwidths specified in both tables have to be seen as a guideline, the real consumption depends heavily on the parameters used.

6. Discussion

6.1. Ubiquity

Complying with established videoconferencing standards our interface is compatible with a wide range of software like Microsoft's NetMeeting, GnomeMeeting or PocketBone, running on low resource mobile devices including PDAs and smart phones as well as laptops and powerful workstations. Together they provide ubiquitous access to virtual and mixed realities.

6.2. Face-To-Face Communication

An important mechanism in natural communication is turn-taking: getting the right to contribute to the dialogue at a specific moment. If such a process is initiated by one of the conference partners but is not transmitted to the other side immediately, this could lead to confusion within the involved participants [FLSS03]. Thus, the important factor here is the latency of the interface: the time from the event to its perceptual presentation at the remote end. The chosen videoconferencing standard provides means for low latency transmissions, depending merely on the properties of the connection which defines the basic latency for all distributed approaches. An approximate measuring during the session in the application example yielded a latency below one second on a distance of ca. 300km.

6.3. Interaction with the VE

In local VR installations, user interaction with the VE can build on a variety of specialised VR input devices such as data gloves, optical trackers, etc. In the ubiquitous VR environments envisioned by our approach, which brings the VE to regular desktop PCs, PDAs, and smartphones, such input devices are usually not available. For remote users, all interactions are therefore based on the communication channels provided by the videoconferencing protocols, such as audio, video, chat. In our current implementation, spoken or typed natural language instructions can be issued by the remote user.

6.4. WYSIWIS

With the videoconference interface to VEs we support the WYSIWIS (what you see is what I see) concept introduced by Stefik et al. [SBF*86] for multi-user interfaces. In its strict interpretation WYSIWIS demands that all conference partners see the same image. This can easily be realised within our system when the render area for the display shares its viewport with the virtual camera. Stefik et al. also propose several dimensions in WYSIWIS that can be relaxed:

- *space*: every user should see every visible object,
- *time*: no delays in updating or viewing,
- *population*: all people have to share all objects, no subgroups possible,
- *congruence*: images have to be identical, different views of the same scene are not possible.

The space and congruence constraints are reliably met when both endpoints of a connection use fullscreen desktop applications, with both having a shared viewport. Non-fullscreen applications would violate the space constraint. When the videoconferencing is used inside a VR installation, the congruence constraint has to be relaxed. The videoconference can not compete with the immersiveness of, e.g. a CAVE (which has several walls and is typically stereo) and therefore, in general, has to use a different, smaller viewport (which can be compared to the representation of one wall that uses mono). For common videoconferencing systems, the time constraint also has to be relaxed as these systems always have latencies and a small delay due to encoding and decoding, network latencies, etc. As there are only two people communicating, at least for the moment, there is no need to build subgroups of people and the population constraint can be kept strict. Even with more people in a conference it might not be necessary to relax this constraint.

Thus, with a VR installation our system allows a relaxed WYSIWIS. On the desktop running in fullscreen and with a low latency it conforms to strict WYSIWIS. However, if these constraints do not hold, the desktop VR setting conforms only to relaxed WYSIWIS. The major issue here is an increase of latency, which can be due to low computing power or the connection properties, e.g. on handhelds or smartphones.

6.5. Additional Benefits

At this point we want to emphasise, that the implementation of the videoconferencing interface introduced was built using freely available software only. The one exception is the interfaced VE, as it is running on SGI's OpenGL Performer which has a commercial license.

7. Conclusion

We presented a new method for remote access of virtual environments based on established videoconferencing standards. A wide range of clients, from mobile devices to laptops or workstations, are supported, most of them out-of-the-box. This makes virtual environments ubiquitously accessible.

Our demonstration of a VR application for virtual prototyping gives a glance at the possibilities of this new integrative technology. Local and remote users have visual and

auditory access to the shared VE, at interactive rates. The participants can communicate with each other in a face-to-face-manner. Additionally, all users can interact with the VE using a natural language interface by means of typed instructions or speech commands.

Future work will include improvements on the range of interaction possibilities. Support for additional modalities will be investigated, such as basic visual interaction, e.g. using simple gestures to select certain objects in the virtual world. Facial expressions could be used to identify the attentional or emotional state of the participant. We also think of a server generated GUI system where the controls are encoded on top of the video signal and pushed to the client. Also, the integration of a facial recognition system is planned to identify emotional states of the participant.

Furthermore, we are planning to test the system in an information desk setting with an embodied communicating agent as communication partner.

8. Acknowledgement

This research is supported by the Deutsche Forschungsgemeinschaft (DFG) in the Collaborative Research Centre SFB 360 as well as the projects "Virtual Workers" and "Virtuelle Werkstatt".

References

[Ali04] ALICESTREET, LTD.: Alicestreet Conference Center. Web address: http://www.alicestreet.com, 2004.

[AT&T03] AT&T LABS RESEARCH: Train of Thought - 1970 The Picturephone. Web address: http://www.research.att.com/history/70picture.html, 2003.

[BBT*03] BAKER H. H., BHATTI N., TANGUAY D., SOBEL I., GELB D., GOSS M. E., MACCORMICK J., YUASA K., CULBERTSON W. B., MALZBENDER T.: Computation and performance issues in coliseum: an immersive videoconferencing system. In *Proceedings of the eleventh ACM international conference on Multimedia* (2003), ACM Press, pp. 470–479.

[BFSK03] BARAKONYI I., FAHMY T., SCHMALSTIEG D., KOSINA K.: Collaborative Work with Volumetric Data Using Augmented Reality Videoconferencing. In *Proceedings of ISMAR'03* (2003).

[BK00] BILLINGHURST M., KATO H.: Out and About: Real World Teleconferencing. *British Telecom Technical Journal (BTTJ), Millennium Edition* (January 2000).

[FLSS03] FRIEBEL M., LOENHOFF J., SCHMITZ H. W., SCHULTE O. A.: "'Siehst Du mich?"' – "'Hörst Du mich?"' – Videokonferenzen als Gegenstand kommunikationswissenschaftlicher Forschung. *kommunikation@gesellschaft 4* (2003).

[Gou03] GOUDESEUNE C.: The Virtual Sound Server,. Web address: http://www.isl.uiuc.edu/Software/software.htm, 2003.

[ICN*00] ISENHOUR P. L., CARROLL J. M., NEALE D. C., ROSSON M. B., DUNLAP D. R.: The Virtual School: An integrated collaborative environment for the classroom. In *Educational Technology & Society* (2000), vol. 3.

[JHW98] JUNG B., HOFFENKE M., WACHSMUTH I.: Virtual Assembly with Construction Kits. In *Proceedings of the 1998 ASME Design for Engineering Technical Conferences (DECT-DFM '98)* (1998).

[MK94] MILGRAM P., KISHINO F.: A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems E77-D*, 12 (1994).

[PLJ*01] PARK K., LEIGH J., JOHNSON A., CARTER B., BRODY J., SOSNOSKI J.: Distance Learning Classroom Using Virtual Harlem. In *Proceedings of VSMM 2001* (2001).

[Pos04] POST INCREMENT: Vox Gratia. Web address: http://www.voxgratia.org, 2004.

[SBF*86] STEFIK M., BOBROW D., FOSTER G., LANNING S., TATAR D.: Wysiwis Revised: Early experiences with multi-user interfaces. In *Proceedings of the Conference on Computer-Supported Cooperative Work* (1986), pp. 276–290.

[SFH*02] SCHMALSTIEG D., FUHRMANN A., HESINA G., SZALAVARI Z., ENCARNAÇÃO L. M., GERVAUTZ M., PURGATHOFER W.: The Studierstube Augmented Reality Project. In *PRESENCE - Teleoperators and Virtual Environments* (2002), vol. 11, MIT Press.

[SGI05] SGI: Open Inventor. Web address: http://oss.sgi.com/projects/inventor/, 2005.

[Sky05] SKYPE: Skype - Free Internet telephony that just works. Web address: http://www.skype.com, 2005.

[Son05] SONY COMPUTER ENTERTAINMENT INC.: EyeToy. Web address:http://www.eyetoy.com, 2005.

[Tra01] TRAMBEREND H.: Avango: A Distributed Virtual Reality Framework. In *Proceedings of Afrigraph '01* (2001), ACM.

[Wil00] WILCOX J. R.: *Videoconferencing: the whole picture*, third ed. Telecom Books, 2000.

[WJ04] WEBER M., JUNG B.: Implementierung kollaborativer Mixed-Reality-Systeme: Ein Vergleich von Open-Source Videokonferenz-Software. In *Virtuelle und Erweiterte Realität - 1. Workshop der GI-Fachgruppe VR/AR* (Aachen, 2004), Brunnett G., Göbel M.,, Müller S., (Eds.), Shaker-Verlag, pp. 173–180.