IPT & EGVE Workshop (2005)
R. Blach, E. Kjems (Editors)

Binaural Acoustics For CAVE-like Environments Without Headphones

I. Assenmacher† and T. Kuhlen‡ and T. Lentz§

RWTH Aachen University Center for Computing and Communication, Virtual Reality Group
RWTH Aachen University Institute of Technical Acoustics

Abstract

The human auditory system, in contrast to the human visual system, can perceive input from all directions and has no limited field of view. As such, it provides valuable cues for navigation and orientation in virtual environments. However, audio stimuli are not that common in today's Virtual Reality applications, and this might result from the lack of middleware or user acceptance due to the need for specialized or costly hardware. Surprisingly, the lack of headphone-less near body acoustics is widely accepted, and simple intensity panning approaches that enable plausible spatial audio are used. This paper describes a networked environment for sophisticated binaural synthesis-based audio rendering in visual VR applications for a freely moving listener in a CAVE-like environment without the use of headphones. It describes the binaural acoustics rendering technique and a dynamic cross-talk cancellation system for four loudspeakers. In addition to that, synchronization issues and network coupling together with performance measurements that proof the applicability of the system in interactive Virtual Environments are discussed.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Three-Dimensional Graphics and Realism]: Virtual reality, spatial audio, binaural synthesis;

1. Introduction

Virtual Reality (VR) environments aim at immersing the user in a computer generated world. In theory, all human sensory systems have to be stimulated in a natural way to achieve this goals. In practice, most modern VR systems show a well established set of methods for the human visual system. As the visual system is considered to be the most important source of information in human perception, many efforts were put into this field of science. In contrast to the visual system, which only has a limited field of view, the auditory system is able to detect stimulation from any direction. Auditory stimulation can provide valuable additional cues for orientation and navigation in virtual environments. The ultimate goal thus would be the ability to place virtual sounds in any three dimensions and distance around the user in real-time.

The utilization of the human visual system in common VR applications is the presentation of a computer generated graphical representation for each of the two eyes, which enables stereoscopic views. The obvious fact that humans have two ears, can be used to create spatial and thus more natural sounds. The *binaural approach* is a strong and powerful method for an exact spatial imaging of virtual sound sources. Traditionally, VR systems provide auditory stimulation by means of headphones, either integrated into head mounted displays (HMD), using off-the-shelf headphones or standard stereo loudspeaker systems. More complex environments range from few loudspeakers (for *intensity panning*) up to large arrays loudspeakers and amplifiers (for *wavefield-synthesis*).

Modern VR display settings do not use HMDs that frequently, as such a device is cumbersome and heavy to wear. Instead, CAVE-like environments with light-weight glasses are used. This is more comfortable and allows free move-

† e-mail: assenmacher@rz.rwth-aachen.de
‡ e-mail: kuhlen@rz.rwth-aachen.de
§ e-mail: tle@akustik.rwth-aachen.de

ment within the virtual scene. Headphones, as an additional wearable, are considered uncomfortable. In addition to that, the sound provided by headphones is not perceived as being very natural.

Until today, there is a lack of spatial sound systems which allow the use of sophisticated near-body spatial auditory stimuli in CAVE-like environments without headphones for a freely moving listener. In order to achieve this goal, several obstacles have to be overcome. First of all, a CAVE-like environment is usually a convex setting of stiff projection surfaces. A wavefield synthesis approach is thus not applicable, as the loudspeaker arrays have to be positioned around the user and outside of the environments boundaries. The same argument holds for intensity panning approaches, but there are installations that place the loudspeakers directly in front of the projection surfaces, usually on the bottom or in the upper regions. This disturbs the visual sensation, as the loudspeakers cover parts of the scene. In addition to that, it is not a trivial task to create correct spatial auditory presentations without using headphones and only a small number of loudspeakers. Another problem can be found in the synchronization of the auditory and visual VR subsystems. Enhanced immersion dictates that the coupling between the two systems has to be tight and with small computational overhead, as each sub-system introduces its own lag and latency problems due to different requirements on the processing. If the visual and the auditory cues differ too much in time and space, this is directly perceived as a presentation error, and the immersion of the user ceases. Another topic is the avoidance of special hardware for audio synthesis such as DSP technology or other custom made sound hardware.

We developed the VirKopf system that approaches these problems with a well defined architecture that allows the creation of true spatial near-body audio in a virtual environment without headphones for one moving listener using two to four loudspeakers, software-based binaural synthesis, and cross-talk cancellation on a standard PC platform.

The remainder of this paper is organized as follows. First, we will give a brief overview of the related work in this area and different approaches to spatial acoustics. Then we will introduce the *binaural approach* that we use to realize the auditory VR. After that, we will briefly describe the system layout in terms of hardware set-up and software architecture. As VirKopf is realized via network interconnected hosts for visual and auditory rendering, it is important to see how much time is spent on the network communication between the systems and how this will effect lag, latency and synchronization issues. This is described in detail after the system presentation. The paper will close with a discussion about the methods used and open topics for further research.

2. Related Work

The VirKopf system is implemented as a distributed architecture. It is obvious that video and audio processing take a lot of computing resources for each subsystem, and it is unrealistic to let this processing happen on a single machine. For that reason, the VirKopf system realizes the computation of video and audio data on dedicated machines that are interconnected over a network connection. This idea is obvious and was already successfully implemented by [BV93] or [MD94]. There are even commercially available solutions, which are realized by dedicated hardware that can be used via a network interface, e.g., the Lake HURON machine. Recent approaches of audio serving technology for various applications can be found in association to the DIVA project [Sav99], [SHLV99]. Other approaches such as [Sto95] or [NSG02] are not implemented as a networked client server architecture but rely on a special hardware setup. This is true for the Lake HURON machine as well, which is a DSP-based spatial sound hardware that allows the usage of different audio rendering techniques by a plug-in mechanism. However, the binaural acoustics module of the HURON machine is limited to the usage of headphones. On the contrary, our approach could possibly be implemented as a plug-in extension to the Lake HURON machine, but does not need to, as it runs on standard PC hardware. Concerning software, [HGL*96] concentrate on data structures, description facilities and synchronization issues for sound and graphics in VEs, but do not describe a special sound rendering technique.

The VirKopf system differs from these approaches in some respects. A major difference is the focus of the VirKopf system to enable a true spatial sound experience for a *moving* listener *without* the need for headphones in *immersive* VR environments. Second, it is not implemented on top of constrained hardware requirements such as the presence of specific DSP technology for audio processing. The VirKopf system realizes a software-only approach and can be used on off-the-shelf custom PC hardware. In addition to that, the system does not depend on specially positioned loudspeakers or a large number of loudspeakers. Four loudspeakers are sufficient to create a surrounding acoustic VE for a single user using the binaural approach, which is described in section 3.

3. Audio Rendering

There are several different approaches producing sound with a spatial placing. One is the multi channel audio often used in home cinema systems to surround the listener with sound. Such systems are based on intensity panning and work quite well for applications which do not require a very exact placing of virtual sound sources. For a more accurate virtual placing of sources mainly two different solutions are available. The wavefield synthesis approach [The03] is to surround the listener with a huge number of loudspeakers, so a sound field similar to the real acoustic environment is reproduced in nearly the whole listening area between the speak-

ers. For our work, the *binaural approach* is chosen and will be presented in the following section.

3.1. Binaural Synthesis

Binaural acoustics deal with the idea not to simulate the sound field in the whole area but only at two points, the ears of the listener. For a correct spatial sound image it is necessary to produce the sound pressure at the ears of the listener as it would appear in a real situation. A binaural signal is a representation of this sound pressure at the ears. The concept *binaural* indicates that two different signals are required, one for each ear [Bla97]. The two signal channels are presented simultaneously but must be perceived separately, one channel at one ear (see chapter 3.2). If it is possible to reproduce a binaural signal at the points it is defined for, the spatial impression of the sound scene should be like reality [Møl92].

A binaural signal can be produced by filtering an anechoic mono sound file with appropriate head related transfer function (HRTF) for the desired angle between the head of the listener and the virtual source. The applied transfer function is related to the head and not to the room. This implies that the virtual source moves with the listener. For the realization of a room related virtual source, the HRTF must be changed when the listener turns his head. The main advantage of a dynamic synthesis is the almost complete elimination of front back confusion as it often appears using the static binaural synthesis with non-individualised HRTFs as, e.g., reported in [WAKW93]. Figure 2 shows the arrival time at each ear in relation to the listeners orientation. The time offset between the signal reaching the ears is called interaural time delay (ITD). In this example the ITD is almost equal regardless of whether the sound is reproduced at position 1 or 2. Although the frequency dependent interaural level difference (ILD) is still different for the two source positions, this is often not a sufficient cue. For that reason the signal could be perceived by the user as if coming from a non-existent mirror source due to the congruent differences in the ITD. Using a dynamic synthesis, the ITD increases when the listeners median plane (see figure 1) turns away from source (1) and decreases when the median plane turns toward source (2). This fact makes a source well defined in its position through the ancillary information, the relative movement of the listener.

Another advantage of the binaural synthesis is the ability of near to head source imaging. Other than panning systems where the virtual sources are always in or behind the line spanned by the speakers, binaural synthesis can realize a source at any distance to the head by using an appropriate HRTF measured in the correct distance. Especially at closer distances the interaural level difference is higher through the shading of the head. Certainly, an extended database is needed for near-to-head distances, which is present in the VirKopf system.

3.2. Cross-talk Cancellation

The easiest way to achieve the required channel separation is to use headphones, but this is not suitable in a CAVE where much effort is made to keep the user free of bothering head mounted displays, tracker cable etc. So, in this case the use of loudspeakers is recommended. To bring the binaural signal to the ears, more precisely the left signal to the left ear and the right signal to the right ear without interferences,

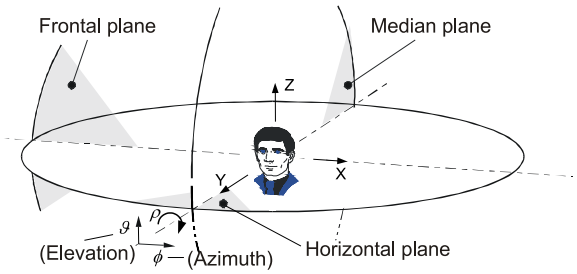

Figure 1: *Planes and terms that are used for the determination of the user's head.*

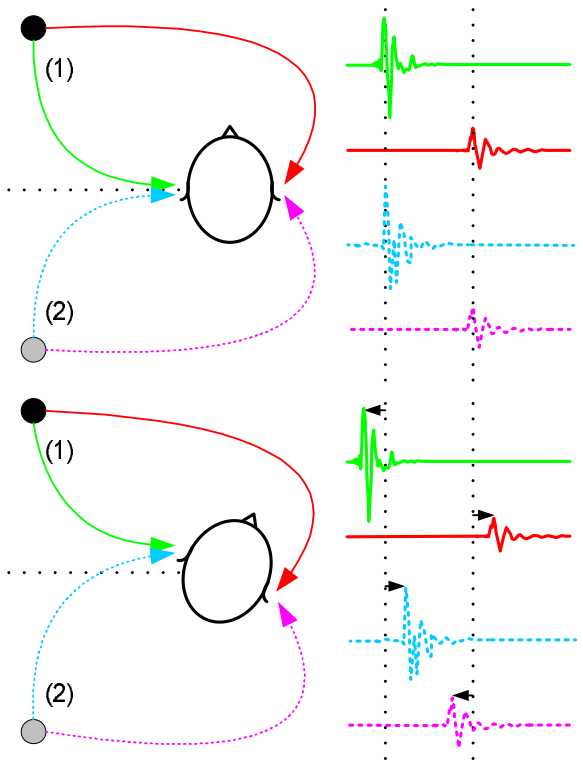

Figure 2: *Variance of interaural time delay depending on the relative head orientation of the user.*

a cross-talk cancellation (CTC) system is needed [Bau63]. Figure 3 shows the principle of a static CTC.

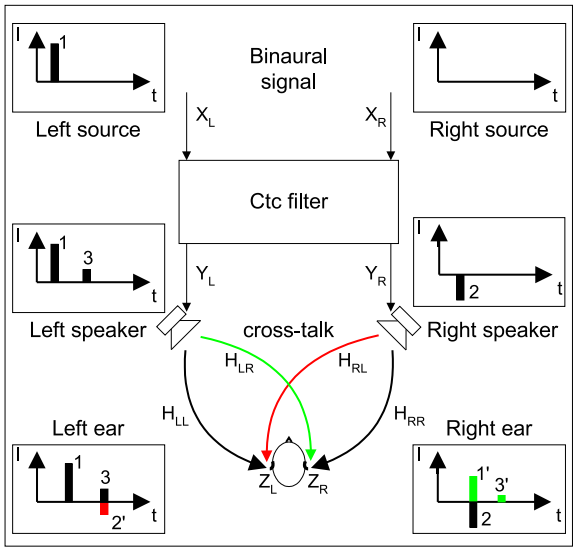

Figure 3: *The Principle of static cross-talk cancellation.*

It is called static because the cancellation works only for one position of the listener ("sweet spot"). The filters are calculated based on the transfer functions from the left speaker to left ear (H_{LL}) and to the right ear (H_{LR}) as well as from the right speaker to the right (H_{RR}) and to the left ear (H_{RL}) measured in that specific position. The postulation that the signals at the ears are the same as the binaural input signal leads to the equation set that characterizes the cross-talk problem (see figure 3).

$$Z_L = Y_L \cdot H_{LL} + Y_R \cdot H_{RL} \stackrel{!}{=} X_L \quad (1)$$

$$Z_R = Y_R \cdot H_{RR} + Y_L \cdot H_{LR} \stackrel{!}{=} X_R \quad (2)$$

For a detailed description of the mathematics see [Møl89] and [BC96]. However, these four transfer functions from each speaker to each ear are not independent concerning the relative position and orientation of the head to the speakers. This is the reason why CTC works initially only for one position. Making the CTC work in an environment where the user should be able to walk around and turn his head needs a dynamic CTC system which is able to adapt during the listener's movements [Gar97], [LS02]. Since the user of a virtual environment is already tracked to generate the correct stereoscopic video images, it is possible to calculate the CTC filter online. The dynamic solution overrides the sweet spot limitation of a normal static cross-talk cancellation. A requirement is a database containing "all" possible HRTFs. In the system presented here, a resolution of one degree for both azimuth (Φ) and elevation (ρ) was chosen. The distance between the loudspeaker and the head affects the time delay and the level of the signal. Using a database with HRTFs measured in a decided distance, these two parameters must be adjusted by modifying the filter group delay and the level according to the spherical wave attenuation for the actual distance. To provide a full head rotation of the user a two loudspeaker set-up is not sufficient as the dynamic cancellation works only in between the angle spanned by the loudspeakers, so a dual CTC algorithm with a four speaker set-up is used. For a detailed description of the dual CTC approach see [LR04], [LB04].

3.3. The Complete Audio System

The complete system's layout with all components is shown in figure 4. The input section, connected to the head tracking

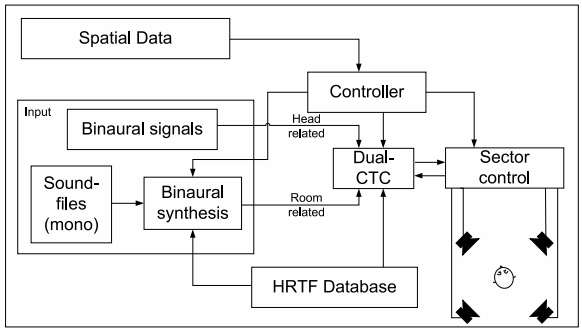

Figure 4: *The complete system of dynamic binaural synthesis and cross-talk cancellation.*

device and to the HRTF database, accomplishes the multi track convolution and mixing of the mono sound files or sound device input channels. The other main part of the system is the dual CTC unit including the sector control. It is also connected to the database and the head tracking. A filter update will be performed when the weighted sum of the listener's movement in all degrees of freedom is above 1 (see equation 3). The threshold can be parameterized in six degrees of freedom, positional values $(\Delta x, \Delta y, \Delta z)$ and rotational values $(\Delta\Phi, \Delta\vartheta, \Delta\rho)$. The lateral movement and head rotation in the horizontal plane are most critical so Δx and $\Delta\phi$ are chosen as $\Delta x = 1$ cm and $\Delta\Phi = 1.0^\circ$ to dominate the filter update. The threshold always refers to the value where the last exceeding occurred. The resulting hysteresis prevents a permanent switching between two filters as it may occur when a fixed spacing determines the boundaries between two filters and the tracking data jitter a little bit.

$$s = \left[\frac{|x_{new} - x_{old}|}{\Delta x} + \frac{|y_{new} - y_{old}|}{\Delta y} + \frac{|z_{new} - z_{old}|}{\Delta z} + \frac{|\phi_{new} - \phi_{old}|}{\Delta\phi} + \frac{|\vartheta_{new} - \vartheta_{old}|}{\Delta\vartheta} + \frac{|\rho_{new} - \rho_{old}|}{\Delta\rho} \right] \geq 1 \quad (3)$$

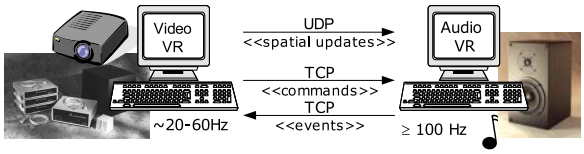

Figure 5: *A sketch of the system layout and the communication channels that are used in the VirKopf system.*

One of the fundamental things required of the sound output device is that the channels work absolutely synchronously. Otherwise the calculated cross-talk paths do not fit with the given condition. On this account the special audio protocol ASIO designed by Steinberg for professional audio recording was chosen to address the output device [Ste04]. One of the advantages of ASIO is a synchronized double buffer structure.

4. System description

The connection between the two systems consists of two bidirectional (TCP) and one unidirectional (UDP) communication channel. Figure 5 depicts the system layout. The first TCP channel establishes the connection to the audio server and allows to control the VirKopf sound system. The second TCP channel is automatically created by the VirKopf system and is used for server sided events, errors and exception messages to the VR client application. The UDP channel exists for the fast rate transmission of spatial updates of the listener and various sound sources in the virtual environment, encoded as a table of positions and orientations. The TCP channels are expected to be used at a low frequency, while the spatial update channel continuously delivers positional information at a high, but constant data rate. A more complete overview of the system setup of the VirKopf system is described in [AKLV04]. The following passage will discuss two of the main synchronization issues the system has to deal with.

The first issue is the latency that the system has in starting, pausing, stopping and altering of attributes of virtual sound sources. This issue is important for the matching of a suddenly appearing sound to a specific object or event. [VdPK00] have shown that it is an advantage if a sound that indicates a specific situation (e.g., a sound that is emitted from a hammer that hits a steel plate) is optimally presented 35 ms after the situation has been visually perceived by the user. The visual rendering engine fills a back buffer which is displayed after a pointer swapping in the video hardware. Audio control commands are collected during the application phase. These commands are sent to the audio server right before the next frame swap, as we can not assume to have access to this low level feature of buffer swap control of the underlying graphics API. Network transfer takes some time, as well as the processing of the audio logic. The sound waves should reach the user's ears close to but after the display of the visual scene. Exact measurements about this requirement for the VirKopf system are presented in section 5.1.

The other synchronization issue to deal with is the latency for updates that are necessary when the is user moving. The latency discussion on that issue has to consider the head-tracking technology that is used in the system. This point deals with the lag in updates if the user moves his head, possibly at a very fast rate. [BSM*04] have stated that an update lag of 70 ms at maximum remains unnoticed for the user. This includes the runtime from the loudspeakers to the user's ears. It is then sufficient to show that the VirKopf system provides an output with a valid filter set for virtual sound objects within that time at the users's ears, beginning with the detection of tracker sensors in the used tracking hardware as an end-to-end latency. Section 5.1 will show that VirKopf meets this requirement.

4.1. The Visual VR Subsystem

The visual VR subsystem is modeled in a quite simple and straightforward way. A graphics API is fed with a scenegraph structure that is rendered during a *rendering phase*. After this phase is over, the *application phase* is used to update the application's internal data structures, which usually results in the modification of the scenegraph for the next frame painting. During the rendering phase, it is assumed that there is no way to modify the scene that is currently rendered. This can only happen during the application phase. This rather simple model matches most of the models that are provided by typical VR toolkits.

During the application phase, the VR programmer has the possibility to send control commands to the auditory VR subsystem, which includes basic sound control (e.g., starting and stopping) as well as changing of sound parameters. In addition to that, if the VR application changes the position or orientation of sonified objects, the data is written to a table that is transported over the UDP spatial update channel before the next frame is rendered.

4.2. The Auditory VR Subsystem

The auditory VR subsystem is more complex in its basic architecture than the visual subsystem. First of all, audio data is processed in a stream like fashion. A sound stream is played from the beginning to the end and then looped again, if desired. During this sequence, the audio data is partitioned into buffers that are processed block wise. The next buffer is calculated from raw audio material by applying finite impulse response (FIR) filters related to the information from the tracking system that determines the position and orientation of the listener in the current scene. A dedicated thread fetches the next buffer and passes it to the sound hardware. It is obvious that this thread has to run fast and smoothly, as

otherwise sound distortion (e.g. "clicks") might occur which disturb the listening experience. Usually, this thread is executed by low-level hardware or interrupt routines that are executed by the local soundboard. This assures the required constraints on the timing of the routines. Another thread fetches the new positional information or is fed with control instructions that change the calculation of the filtering processes.

5. Cost model

Besides a nice and flexible application architecture, it is important to see that certain costs for computational overhead still fit the real-time requirements that are given by the interactive VR setting. In order to measure the overall system performance, we give a cost model that is derived from the architecture as depicted above.

Tracking is used to identify the user's head position and orientation at discrete sample slots. Different tracking methodologies exist, each with its own advantages and drawbacks. The tracking device usually needs some time to identify a sensor within the tracked space, and a proper encoding and transmission to the VR application. The effect of the tracking hardware to the cost model will be the time from a current situation up to the processing of the sample within the VR application's application phase.

An audio command is then dispatched, serialized, sent over the network, deserialized and finally processed by VirKopf. This accounts to TCP as well as UDP transmissions. For serialization and deserialization, VirKopf relies on the serializer patterns that are implemented in the used VR framework. As a consequence, VirKopf sticks to that framework's rule that any date on the network has to be delivered in Big Endian byte-order. Currently, spatial updates are transmitted as tables, where each table spatial data of up to 25 sound sources and one listener. If there are more than 25 altered sound sources, more than one table will be sent until every changed sound source is transmitted.

The network measurements were performed as follows. For the TCP channels, packets of the maximum size (360 bytes for the current version of the VirKopf) were transported from the client to the server and then sent back. The time it took to complete this operation was measured in terms of CPU ticks on the client side. On the assumption that the packets traveled each way (from client to server, from server to client) at basically the same speed, the measured time for a round trip was halved to get the time for a single way transport. For the UDP channel, packets of maximum size (1024 bytes for the current version of the VirKopf) were transported from the client to the server machine. On the server side, the message was received and immediately sent back to a dedicated address on the client side. The time for this round trip was measured on the client side and halved.

As depicted above, the audio processing subsystem is inherently multi-threaded. Figure 6 depicts the following processing scheme. One thread receives, deserializes and dispatches commands from the network connection. Incoming spatial data that is sent over the UDP update channel is dispatched by two independent threads. One thread does the synthesis filter calculation that introduces the spatial information to the raw audio data for the upcoming buffer. The information is used in the convolution process that is driven by the low-level architecture of the used sound system. Another thread calculates the needed filter settings for the CTC. The results from the convolution with the new filter settings for the CTC get output by the sound hardware. As mentioned

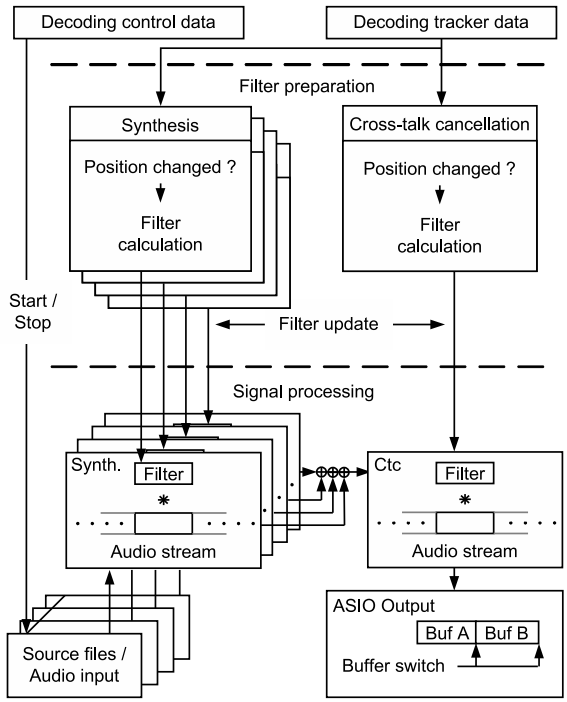

Figure 6: *Audio processing in the audio layer with multiple threads, one for decoding control data, two for decoding tracker data and a low level thread for audio hardware buffer switching.*

above, these two threads are independent, which means that whenever a synthesis filter is calculated, it is considered to be the valid filter applied to the next buffer of the audio stream which gets output by the sound hardware in conjunction with the current CTC filter settings. Whenever the CTC filter is calculated, is becomes the current CTC filter for the next buffer output.

Due to the buffer switch of the sound hardware, it is possible that a filter set is just ready after a buffer switch has taken place and old filters are applied to the next buffer, as filter changes only have effect at the beginning of a buffer frame. Costs for this processing stage are calculated for that

worst case scenario, as the sum of the times needed for the filter calculation and a missed buffer swap. The output is directly put as a signal to the loudspeakers and need a certain runtime to reach the user's ears.

The above section tried to follow the isolated steps that are needed for an update in position and orientation of the user's head, starting at the tracking hardware to the visual VR application and the auditory VR sound processing up to the user's ears again. It is clear that the time for this processing must not exceed a certain threshold, as this will disrupt the user's immersion into the virtual scene, because she notices that visual and auditory presentation do not match. Especially in the binaural approach chosen here, a low latency is much more important than, e.g., with the use of a simple panning approach.

It is obvious that the architecture introduces network latency issues as well as tracking gaps and processing delays in the audio algorithms that might influence the overall performance. In order to show that the current set-up does fulfill the real time and interactivity requirement, detailed measurements were taken with special regards to these aspects.

5.1. System Performance

The following section will give details about the measurements that were taken using the following setup. The tracking PC comprises of a Pentium-4 with 2.4 GHz CPU speed and 256 MB of RAM in conjunction with a four camera set-up. As a client visual VR machine, a Linux dual Pentium-4 machine with 3 GHz CPU speed and 2 GB of RAM was used, in conjunction with an NVidia GeForce FX 5950 Ultra as graphics output device. The host for the auditory VR subsystem was a Pentium-4 machine with 3 GHz CPU speed and 1 GB of RAM. As audio hardware an RME Hammerfall system is used. This hardware allows the sound output streaming with a scalable buffer size and therefore a minimum latency of 1.5 ms. The network interconnection between the machines was a standard Gigabit Ethernet this is used for normal network communication in our laboratory. For the current system we employ an optical tracking system, the A.R.T. tracking system. This optical tracking system recognizes targets which are defined by a rigid geometrical arrangement of four to twenty spherical retro-reflecting markers, called *Rigid Body*. The 2D data that is captured by the set of cameras is computed to a 6DOF information by a standalone PC and transported - using a network connection and a UDP protocol - to a number of VR application host machines where it can be fed to the VR application for head movement or interaction updates [A.R04]. This tracking method introduces some latencies as follows. The 2D images that were captured by the cameras have to be processed in order to calculate a 6DOF data. First of all, the 2D data has to be transported - using a network connection - to the PC system where the 6DOF data will be calculated from all incoming images. The costs for the processing depend on the number of cameras used as well as the number of Rigid Bodies that have to be identified. After that, the 6DOF data is encoded as ASCII string and sent to the VR applications. Our standard test application uses two bodies that are tracked, one for the head and one for a FlyStick interaction device. According to the specification, the A.R.T. tracking system can identify these two bodies with a latency of 18.2 ms from the flash of the camera to the deliverance on the ethernet connection. For the network transport, we assume a time as was measured for the UDP communication of the spatial updates in the VirKopf system of 0.70 ms.

Stage	Mean (ms)	σ (ms)	Remarks
Command/Event channels			
Serialization	0.15	0.08	
Deserialization	0.16	0.08	
Transmission	1.20		worst case
TCP overhead	1.51		
Spatial update channel			
Serialization	0.10		25 sources
Deserialization	0.12		25 sources
Transmission	0.70		worst case
UPD overhead	0.92		

Table 1: *Average time in ms and standard deviation for the overhead of network communication. The standard deviation is calculated from the times of the different TCP channel commands.*

Serialization and deserialization introduce the overhead of byte-swapping, as in the current set-up both participants are running in Little Endian byte-order, and as stated above, byte swapping has to be performed. Following from table 1 the mean time for the network latency on the slow command channels is 0.31 ms and 0.22 ms on the UPD spatial update channel. For the calculation of the end-to-end costs, only the time for the spatial update on the UDP channel is considered. As described in the cost model section, two measurements were taken to calculate the costs for the raw network transport. The TCP command and event channel transport was measured as follows. The mean time for transmitting a TCP command was 0.15 ms $\pm$ 0.02 ms. The worst case transmission time on the TCP channel was close to 1.2 ms. As a total cost for the TCP transmission phase, 1.2 ms is assumed consequently.

UDP communication was measured for 20000 samples and a single table can be transmitted in 0.26 ms $\pm$ 0.01 ms. It seems surprising that UDP communication is more expensive than TCP, but this may be a result from different packet sizes that were used for the measuring. The maximum value for a UDP transmission that was found is close to 0.7 ms.

The latency of the audio system is the time elapsed between the incoming of a new position and orientation for either a source or a listener, and the point in time the out-

Cost Item		Time (ms)
Tracking (60Hz)		18.20
UDP transport	+	0.70
VR app. transform	+	0.10
Serializing, deserializing	+	0.22
UDP spatial update	+	0.70
Filter processing	+	11.80
Sum	=	31.72

Table 2: *Total costs for a end-to-end trip of a spatial update from a tracking sample to the output of the loudspeakers.*

put signal is generated with the updated filter functions. The output block length of the convolution is 256 taps as well as the chosen buffer size of the sound output device, resulting in a time between two buffer switches of 5.8 ms at 44.1 kHz sampling rate for the rendering of a single block. The calculation of a new CTC filter set (1024 taps) takes 3.5 ms on a 3 GHz PC and 0.1 ms to process a new binaural filter (512 taps) for each sound source. In a worst case scenario the filter calculation just finishes after the sound output device fetched the next block, so it takes the time playing this block until the updated filter becomes active at the output. That would cause a latency of one block. In such a case the overall latency accumulates to 11.8 ms. This value results as the CTC filter takes 3.5 ms which is larger than the 2.5 ms for 25 sound sources that can be delivered by a single spatial update call from the visual VR application.

As a summary of the above depicted measures, the total time of the end-to-end trip of the information from the movement of the listener's head to the sonification at the loudspeakers is shown in table 2. The longest time interval is caused by the tracking system. It is clear that the total cost for the end-to-end trip would benefit from faster tracking.

Note that all these values are taken from the worst-case assumptions on the system behavior. The average case looks much better, in addition to the fact that only the fast response to a head change of the listener is critical to the perception of the spatial audio.

5.2. Results

The VirKopf system tries to realize a real-time processing of spatial data meeting on the one hand the interactivity requirement of a VR application to have a frame rate of at least 15 frames per second and in addition to that no lag larger than 70 ms in between the presentation of the visual virtual scene and the auditory virtual scene. The end-to-end time for a spatial data from the tracking device to the audio signal as emitted from the loudspeakers is in a worst case scenario $\approx$ 32 ms. This does not include the runtime from the loudspeakers to the listener's ears. In our CAVE-like installation, the maximum distance to the loudspeakers is 3 m, resulting in a sound runtime of $\approx$ 10 ms. The resulting sum of 42 ms from the tracking device to the user's ear satisfies the 70 ms requirement. Changes in sound parameters from the application can be processed by the audio system in less than 1.51 ms. The frame rate requirement dictates that the application (assuming an empty scene) may not take more than 66.6 ms computation and rendering time. This requirement is easily met with the figures given above (1.51 ms for TCP and 0.92 ms for UDP), assuming a blocking transfer scheme and worst case transmission times.

More interesting is the timing for the spatial updates, which, for best results, appear with at least 100 Hz. The costs for encoding and decoding of a typical sound table that contains the current listener's position and orientation are 0.22 ms, the costs for network transport are given as 0.7 ms, resulting in a total of 0.92 ms from the application's tracker update to the input on the audio host in a worst case scenario. The number of sound sources per table is limited. Currently, a threshold value for the audio serving machine is the calculation of the convolution for 25 sound sources on the machine that was used due to computational resources.

As a consequence, it is clear that enriching a typical VR application with spatial sound does not have much impact on the real-time ability of the application. It is obvious that the networked setup introduces additional lag times and latency, but is feasible with modern Gigabit Ethernet interconnections.

The VirKopf described in this paper offers a comprehensive and well performing approach to implement near body spatial audio in VR applications on a software-only basis and allows the deeper examination of the interaction between the visual and the auditory systems of human perception. It is independent of a specific VR toolkit, relies only on standard hardware and has low requirements on the OS or the presence of other software components.

6. Discussion

It is clear that the average case is a lot better than the worst case that has been discussed in the results section of this paper, but even the worst case scenario still fulfills the requirements well, and the current implementation leaves room for improvements. For example, the byte-swapping can be turned off for the network communication, as both hosts use the same byte-order. The network communication is modeled to be blocking along the send and receive calls, which slows down the application processing and can be turned to an asynchronous scheme for further latency reduction. Overhead for the kernel network processing was measured twice. This is not a drawback, as a subtraction of this overhead would benefit the total results, which are already working in the worst case analysis. Tracker latency introduces the biggest gap for the end-to-end trip costs, and a tracking with an update rate of 100 Hz is mandatory for a well working calculation of the CTC. If the tracking system is not able

to deliver data samples at this rate, predictive tracking will be implemented and examined in future research. The goal is to deliver samples at a constant rate of at least 100 Hz by inserting interpolated samples when no current data is available from the tracking unit. Another drawback of the current implementation is the lack of high-level data structures and control instances besides the pure management of sounds and attributes of sounds. The idea is to provide special data structures as in [HGL*96] or to provide the application programmer with a special pattern or artificial language to control the sound rendering and associated update policies.

On the audio rendering side, the influence of reflections and the (semi-) automatic calibration of the sound system is of strong interest. Currently, the position of the loudspeakers has to be measured manually. This is error-prone and should be replaced by a smarter calibration method. In addition to that, the influence of the virtual room, e.g. reverberation, on the sound and the directivity of sound sources will be a research topic in the near future.

7. Conclusion

The main goal of our work is to realize true immersive auditory and visual VEs for a moving listener without headphones in CAVE-like environments. In order to reach this goal, a threefold computer set-up is used. One PC collects tracking data from an optical tracking device (A.R.T. tracking system) and sends it to a machine that hosts the visual VR application where the incoming data is processed according to specific demands of the application (e.g. transformation, applying to objects etc.). Spatial updates as well as command structures are sent to another host which is a dedicated audio rendering machine with a standard sound board connected to a four loudspeaker set-up.

This paper showed that although this complex infrastructure is used and through the overhead of network interconnects, this architecture is still capable of delivering true spatial near body auditory VR that can be used to provide a more natural VR experience for improved interaction and immersion in CAVE-like virtual environments.

References

[AKLV04] ASSENMACHER I., KUHLEN T., LENTZ T., VORLÄNDER M.: Integrating real-time binaural acoustics into vr applications. In *Virtual Environments 2004, Eurographics ACM SIGGRAPH Symposium Proceedings* (June 2004), pp. 129–136.

[A.R04] A.R.T. GMBH: *A.R.T. tracking systems, http://www.ar-tracking.de*, 2004.

[Bau63] BAUER B.: Stereophonic earphones and binaural loudspeakers. *Journal of the AES 9* (1963).

[BC96] BAUCK J., COOPER D.: Generalization transaural stereo and applications. *Journal of the AES 44* (1996), 683–705.

[Bla97] BLAUERT J.: *Spatial Hearing, Revised edition.* Cambridge, Massachusetts: The MIT Press, 1997.

[BSM*04] BRUNGART D. S., SIMPSON D. D., MCKINLEY R. L., KORDIK A. J., DALLMAN R. C., OVENSHIRE D. A.: The interaction between head-tracker latency, source duration, and response time in the localization of virtual sound sources. In *Proceedings of ICAD 04 - Tenth Meeting of the International Conference on Auditory Display, Sidney, Australia* (July 2004).

[BV93] BURGESS D. A., VERLINDEN J. C.: An architecture for spatial audio servers. In *VR Systems Fall93 Conference, New York* (1993).

[Gar97] GARDNER W. G.: *3-D audio using loudspeakers*. PhD thesis, Massachusetts Institute of Technology, 1997.

[HGL*96] HAHN J. K., GEIGEL J., LEE J. W., GRITZ L., TAKALA T., MISHRA S.: An integrated approach to motion and sound. *Journal of Visualization and Computer Animation 6*, 2 (1996), 109–123.

[LB04] LENTZ T., BEHLER G.: Dynamic cross-talk cancellation for binaural synthesis in virtual reality environments. In *Proceedings of the 117th Audio Engineering Society Convention San Francisco, USA* (2004).

[LR04] LENTZ T., RENNER C.: A four-channel dynamic cross-talk cancellation system. In *Fortschritte der Akustik: CFA/DAGA, Strasbourg Fance* (2004).

[LS02] LENTZ T., SCHMITZ O.: Realisation of an adaptive cross-talk cancellation system for a moving listener. In *21st Audio Engineering Society Conference, St. Petersburg* (2002).

[MD94] MULDER J. D., DOOIJES E. H.: Spatial audio in graphical applications. In *Visualization in Scientific Computing*, Göbel M., Müller H.„ Urban B., (Eds.). Springer-Verlag Wien, May 1994, pp. 215–229.

[Møl89] MØLLER H.: Reproduction of artificial head recordings through loudspeakers. *Journal of the Audio Engineering Society. 37* (1989).

[Møl92] MØLLER H.: Fundamentals of binaural technology. *Applied Acoustics 36* (1992), 171–218.

[NSG02] NAEF M., STAADT O., GROSS M.: Spatialized audio rendering for immersive virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology, Hong Kong, China* (2002), pp. 65 – 72.

[Sav99] SAVIOJA L.: *Modeling Techniques for Virtual Acoustics*. PhD thesis, Helsinki University of Technology, December 1999.

[SHLV99] SAVIOJA L., HUOPANIEMI J., LOKKI T.,

VÄÄNÄNEN R.: Creating interactive virtual acoustic environments. *Audio Engeneering Society 49*, 9 (September 1999).

[Ste04] STEINBERG: *ASIO 2.0 Audio Streaming Input Output Development Kit*, 2004.

[Sto95] STORMS R. L.: Npsnet-3d sound server: An effective use of the auditory channel, 1995.

[The03] THEILE: Potential wavefield synthesis applications in the multichannel stereophonic world. In *24th AES International Conference on Multichannel Audio* (2003).

[VdPK00] VAN DE PAR S., KOHLRAUSCH A.: Sensitivity to auditory-visual asynchrony and to jitter in auditory-visual timing. In *Human Vision and Electronic Imaging V, Proceedings of the SPIE* (2000), Rogowitz B. E., Pappas T. N., (Eds.), vol. 3959, pp. 234–242.

[WAKW93] WENZEL E., ARRUDA M., KISTLER D., WIGHTMAN F.: Localisation using nonindividualized head-related transfer functions. *Journal of the Acoustical Society of America 94 (1)* (1993).