Setup for demonstrating interactive binaural synthesis for telepresence applications

Madsen, Esben; Olesen, Søren Krarup; Markovic, Milos; Hoffmann, Pablo F.; Hammershøi, Dorte

Published in:
Acustica United with Acta Acustica

Publication date:
2011

Document Version
Early version, also known as pre-print

Forum Acusticum 2011 - Aalborg - Denmark

Setup for Demonstrating Interactive Binaural Synthesis for Telepresence Applications

Esben Madsen, Søren Krarup Olesen, Miloš Marković, Pablo Hoffmann, Dorte Hammershøi
Section of Acoustics, Department of Electronic Systems, Aalborg University, Aalborg, Denmark

Summary
In the telepresence research project BEAMING, a prototype system has been set up to demonstrate basic audio and video interaction between two distant locations: the *Destination*, where 2 *Locals* are present and the *Visitor Site* where 1 *Visitor* is present. This paper describes the auditory parts of this system as well as interfaces to relevant parts of the complete system, including tracking and network streaming.
In the demonstration, the *Visitor* is wearing headphones and a microphone. At the *Destination*, the two *Locals* are both wearing a microphone, while the *Visitor* is represented using a fixed position *Totem* with a single loudspeaker. The Position and movements of participants, particularly the head, are tracked and from this sound is rendered to include binaural cues so the visitor is able to move around in a limited space while perceiving *Destination* sound as "stationary".
This setup includes 3 main tasks: Tracking coordinates are combined to calculate directions. This is handled by sharing "global" coordinates across the sites and adding local changes with a low latency, ending with a "direction of sound" for each source. Audio is recorded and transmitted over network. Here bandwidth, latency and transmission reliability must be adjusted to obtain the best compromise. Bandwidth use and reliablity can be improved at the cost of latency. Finally the binaural synthesis for each source is processed at the listener's site (here *Visitor*) to have a minimum latency on responding to movement.
The combined system was evaluated by the user experience at the demonstration, with the overall conclusion that interactive binaural synthesis is an important aspect of a fully immersive telepresence application and that we should continue in this direction examining different approaches.

PACS no. 43.60.Dh, 43.72.Qr

1. Introduction

This paper describes the setup used for demonstrating interactive binaural audio synthesis at the first annual review of the BEAMING[1] project, which is a four year collaboration research project funded by the EU FP7 programme (project no. 248620) with the goal of implementing a telepresence system going beyond the current state of art.

Binaural synthesis may be described as the process of rendering sound 3-dimensional by applying a model of how humans perceive directionality of sound. The process includes using digital filters representing Head Related Transfer Functions (HRTF's), which are a set of transfer functions from specific directions to inside the ears of a head in free field conditions. Optimally the transfer functions should be measured in the ears of the person for whom sound is rendered, but with a good dummy head a decent result can be achieved. The background and process is well described by eg. [2] and [3].

The overall goal of BEAMING is to improve current remote communication means to the level of a *Visitor* achieving the sensation of "really being there", without actually being physically present. Likewise for people physically at the *Destination*, the goal is to feel that it is exactly this particular *Visitor* who "is there" and to have a natural interaction between *Visitors* and *Locals*.

The purpose of this first review was to a large degree for partners to make prototype demonstrations of how various types of technology may be used in the project. It is on this premise that the setup of this paper is evaluated.

From the viewpoint of auditory modality, the goal was to demonstrate how binaural audio synthesis can be used to make the experience of communication more immersive than for example a regular video

[1] BEAMING: Being in Augmented Multi-Modal Naturally Networked Gatherings [1]

conference by utilizing tracking of head positions and movements to make it interactive.

The demonstration was built around a scenario with 1 *Visitor* located in Barcelona (where the reviewers were present) who visits the *Destination* in London, where 2 *Locals* are located.

At the *Visitor Site*, is a setup including an OptiTrack (motion capture) system for full body tracking and a head mounted display, which is also tracked.

At the *Destination* are different video solutions, including a Microsoft Kinect™, which is used for tracking positions of the locals, including head positions relative to the *Avatar* (*Visitor* representation).

The *Destination* is set up by the one of the partners, the VECG group of the Department of Computer Science at University College London and the Visitor site is an installation at the EVENT Lab at the Faculty of Psychology of the University of Barcelona. The locations of people and equipment are nicely illustrated by the upper part of Figure 1.

2. Design Considerations

Before implementing the system, some considerations were made regarding the design. All the considerations were based on the specific scenario of the review, while also taking a more generic use into account. The considerations involve issues regarding network and processing of audio as well as the overall topology of the proposed system and the necessary equipment.

2.1. Binaural Processing and Latency

When utilizing head tracking for making binaural synthesis interactive, the delay from a head movement to the corresponding change in synthesis must be sufficiently low in order to support the illusion of an external sound source. A study found that these latencies are distinguishable above approximately 30 ms [4], suggesting that this may be a suitable upper limit of accepted delay in an implemented system.

When using internet connections for transfer of data, the delay from sending a request to receiving a reply may easily exceed these 30 ms. Even though lower latency can be achieved, it is not predictable and no guaranties can be given on upper time limits due to the way internet routing works.

In order to avoid too high latency when the *Visitor* rotates the head, it is necessary to apply the binaural synthesis as late in the processing and transmission chain as possible. When synthesizing 3D audio for the *Visitor*, the binaural processing therefore needs to be carried out at the *Visitor Site*.When synthesizing 3D audio for the *Visitor*, the binaural processing therefore needs to be carried out at the *Visitor Site*.

This decision implies that when multiple *Locals* are at the destination, their audio streams should be transferred in a way so the binaural synthesis may be applied after transfer. A straight forward way of doing this is to transfer one audio stream for each local. This solution is also optimal for the quality of applying binaural synthesis, since this will be most realistic with a sound that is as direct as possible.

When using binaural synthesis, one should remember to consider some aspects relating to the model since both the room and the source characteristics have an influence on how well the model applies. For this demonstration it was decided to limit these considerations to note that human voice works well as a source for binaural synthesis and that we ignore the influence of the room, assuming that direct sound is more relevant to source localization than reverberant sound.

2.2. Bandwidth

In the context of this particular demonstration, the bandwidth usage is not a large obstacle from the maximum of 3 channels of simultaneous audio streaming (2 from *Destination* to *Visitor Site* and 1 the other way). For future versions of the system meant for multiple *Visitors* and many locals, the system should however be able handle this with as little increase in bandwidth usage as possible and bandwidth should in general be utilized economically as it is to be shared with video and other data. The number of transferred streams should therefore be kept as low as possible when adding more people so, from this perspective, the solution with using one stream per source is not optimal. In general the number of streams needed with this setup for any number of *Visitors* and *Locals* would be

$$n_{\text{in}} = n_V \tag{1}$$

$$n_{\text{out}} = n_V \cdot (n_L + n_V - 1) \tag{2}$$

Where
n_V is the number of *Visitors*
n_L is the number of *Locals*

When any type of participant joins, the result is that extra streams are added to all *Visitors*, so the number of outgoing streams will increase rapidly for higher numbers of participants and this solution therefore does not scale optimally.

A better solution would be if it was possible somehow to limit the number of streams per *Visitor* to a fixed value, since this would limit the growth to be a linear function of the number of *Visitors*. One idea to solve this is to use a microphone array or grid which covers the entire *Destination* and perform a processing which selects and conditions the audio for the binaural synthesis at the *Visitor Site*. A different solution could be to try to capture the sound field around the *Avatar* and recreate this virtually at the *Visitor Site*,

thus limiting the sent streams to be the number of microphones mounted on the *Avatar*. These methods, will be examined more for future implementations, however for this demonstration, it was decided to use the method with one stream for each participant.

Apart from the number of streams, it is also important to consider the bandwidth used for each individual stream and thus consider using some type of compression. Most audio codecs are either good at obtaining a high quality despite compression (MP3,AAC etc.) or provide a low delay in encoding (Speex, AMR-WB and other algorithms based on Code-Exited Linear Prediction), leaving a gap for those wishing high quality *and* low latency, for instance for IP-telephony. More recent advances within network audio have addressed this need and the Constrained-Energy Lapped Transform (CELT) algorithm has been proposed to provide a low (less than 10 ms) latency along with a good audio quality (using a 44.1 kHz sampling rate) [5]. Compared to a number of different algorithms CELT has proved to have a comparable quality with far less delay, although the codec is not yet implemented in a stable version for production use.

2.3. Equipment

With regards to the equipment needed for the demonstration, different scenarios were considered.

Starting with the *Visitor Site*, a comparison can be made to existing virtual reality installations. It is often seen that loudspeakers are used to produce the sound [6, 7], for instance using ambisonics.

In some cases it is reasonable to avoid headphones and tracking, for instance in applications where the user should be free of all constraints. In this case however a head mounted display and tracking is already being used for video, so adding a microphone and a pair of headphones will not be a dramatic increase in worn equipment.

At the *Destination*, there is a wish to keep *Locals* as free as possible from any worn equipment, preferably not requiring them to wear anything at all. This of course pose some challenges in implementation with regards to recording audio of locals in a manner suitable for binaural synthesis. The solutions mentioned above with microphone arrays and grids are possible ways to deal with this, however for this demonstration it was decided to use a head mounted microphone for each *Local*.

When presenting audio of the *Visitor*, solutions were either to have a fully symmetric setup and present 3D audio over headphones to *Locals*, or simply to use a loudspeaker as the "mouth" of the *Avatar*. While the headphone solution would be easy to develop, since it is exactly the same processing as for the *Visitor*, it imposes a requirement of full head tracking of *Locals* and also adds another piece of equipment they should wear. By giving the *Avatar* a "mouth", these are no longer issues, however there is a risk of introducing echo of the *Visitor* and some tests and considerations about echo canceling are needed.

For this demonstration it was decided to use a speaker and to attempt using it without echo canceling, since implementation time was limited. A simple test with the chosen microphones revealed no audible echo or feedback when used approximately 1 m in front of speakers with a higher sound level than should be used in the setup.

Other equipment includes a PC with the software and a connected usb sound card as well as tracking systems provided by the partners at the demonstration sites (London and Barcelona), which will provide tracking data over network.

3. The Setup

The final setup which were to be used in the demonstration is a result of the above considerations as well as some other design decisions, such as communication protocols, some of which were already used in other parts of the BEAMING project. An overview of the final setup can be seen on Figure 1.

Apart from the full version, two limited implementations were made as fallback, in case something went wrong in the time scheduled for demonstration. Many external factors could fail or interfere with the demonstration, such as the network connection between *Visitor Site* and *Destination* or the different tracking systems.

3.1. Equipment and Installations

The following equipment is used in the setup:

- Headworn RØDE HS-1 microphones
  - 2 at the *Destination* and 1 at *Visitor Site*
- Edirol UA25EX usb sound cards
  - 1 at each location
- A PC running the software (described later)
  - 1 at each location
- Tracking systems provided by partners
  - At both locations
- Beyerdynamic DT 990 Pro headphones
  - For the *Visitor*

The *Destination* room in London, which is normally used as an office, is prepared over a few days before the review, where everything (including video) is set up.

Tracking information at the *Destination* is obtained by a Microsoft Kinect used by the video group. The head positions of *Locals* are provided over a LAN[2] connection using UDP[3] as x, y and z coordinates in meters, with the Kinect camera as the point of origin.

[2] LAN: Local Area Network

[3] UDP: User Datagram Protocol, a fast, low level network protocol with no feedback of whether data is received

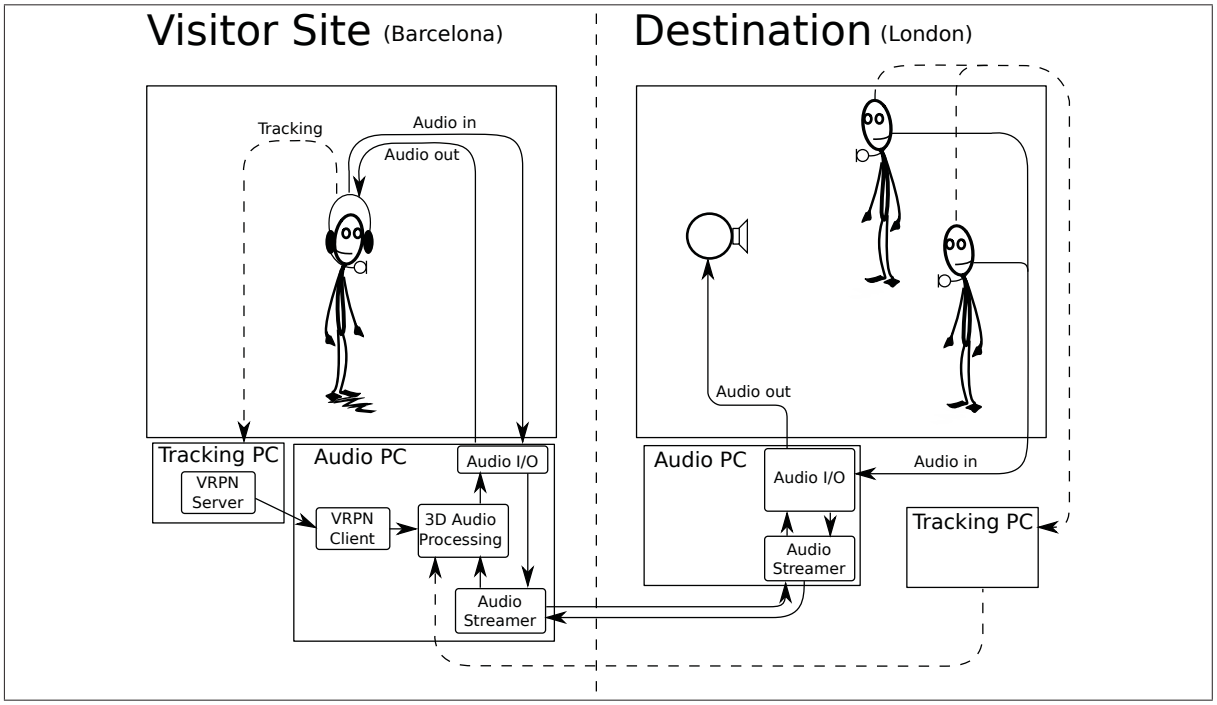

Figure 1. An overview of the audio setup for the demonstration taking place in London and Barcelona, including data paths and main software modules.

At the *Visitor Site* in Barcelona an installation exist in a room with an OptiTrack motion capture system and a head mounted display, which is tracked with an InterSense-900 system.

Positional and rotational information of the *Visitors* head is provided over LAN as x, y, z coordinates and a rotation in quaternions, using the Virtual-Reality Peripheral Network (VRPN) classes which are also used in other parts of the system. VRPN has the advantage of providing a shared interface for many different types of devices, such as trackers, used in virtual reality applications, so changing a tracker to a different model is easy.

At both locations, the computers which are running the software are connected to the internet. They are directly accessible from outside their respective locations on selected ports.

3.2. Software

The software is written to be as cross compatible as possible, meaning it should work on Windows, Linux and Mac OS X. The main test and development for this demonstration was done on a Linux platform, but most parts were also tested on a Windows 7 installation. To make programs and libraries cross compatible, the following decisions were taken:

- Audio I/O is implemented with the Portaudio Portable Cross-Platform Audio I/O library
- Graphical User Interface, where used, it is implemented with the Qt framework (general project decision)
- Network communication is implemented using the RakNet network engine which is "designed for speed, ease of use, application independence, platform independence, and feature set" (project suggestion)

The overall structure of the software is that it is modular and to a large degree symmetric between *Destination* and *Visitor Site*, with each site having both a client and a server. The client part is responsible for sending audio via the network and the server for receiving it.

One way of describing the software is in terms of the data paths, illustrated on Figure 1.

To present the *Visitor's* audio at the *Destination*, the audio is first recorded from the *Visitor Site* with the audio I/O module. This audio is then handled by the Audio Streamer module, which is responsible for an optional compression and the transmission of audio. The audio is transferred from the *Visitor Site* to the *Destination* where a different instance of the Audio Streamer module receives and, if necessary, decompresses the audio. Ultimately it is then handled by the Audio I/O module and presented to the *Locals* through a loudspeaker.

Capturing the *Destination* in terms of audio is equivalent to the data path described above until audio is received by the Audio Streamer module. An ad-

dition is that head positions of the *Locals* are tracked and transferred to the *Visitor Site*. At the *Visitor Site* the head position and rotation is likewise tracked and combined with the *Locals'* head coordinates. The directions of audio relative to the *Visitors* head is calculated from the two coordinate sets and applied to the audio streams in the 3D Audio Processing module before being presented to the *Visitor* in a set of headphones.

The three main modules in the software are the Audio I/O module, the Audio Streamer module and the 3D Audio Processing module. The I/O and Streamer modules are implemented as Qt classes, and the 3D Audio Processing is implemented as a separate C++ library to make it useful in different applications.

Starting with the 3D Audio Processing, it is one of the central modules and is responsible for filtering audio streams with appropriate Head Related Transfer Functions according to a selected direction. The HRTF database is contained in the library and it is left to the user of the library to consider source characteristics and reverberation to ensure that the output corresponds to the intended auditory model. This library is intended to be "pluggable", in the sense that it should be possible to practically insert it anywhere in a piece of software, with very little work and to use it either as a shared or static library. It is also designed to be thread safe in the sense that processing of data and control (change of direction) can take place from different threads/parts of a program. To use the library, a direction is given with one function and the data to be filtered is input through another function. The two functions may be used independently from different threads, implemented so the filter to use is queued when selecting a direction and the queue is used and emptied by the filtering function. The filtering function uses mono audio samples as input and supplies the filtered output samples as a free field binaural signal in two channels.

Audio I/O is implemented around the Portaudio library, using C++ bindings, in order to achieve a cross platform audio solution. When creating an instance of the class, Portaudio is initialized. After this, the class itself must be initialized by defining whether it is to be used for input or output, how many channels to use and supplying a buffer to use for input or output, which may later be changed if one desires. The remaining controls are calls to start and stop playback/recording as well a method to test if the stream is active (playing or recording is taking place). In the demonstration application, the 3D audio processing was implemented directly in this class with a compiler declaration determining whether to apply it, this however should not be the final solution.

The Audio Streamer module is responsible for transmitting and receiving the audio data as effectively as possible. Therefore it is also designed to include compression and decompression of the data on the fly, although this feature is not yet fully implemented. When implemented, the current plan is to use the CELT codec rather than transmitting raw data as is the case now. Network communication is a crucial part of this module and is implemented with the RakNet engine, which is based on UDP and implements a number of features on top of this, such as monitoring of the connection. One of the RakNet features used here includes methods for defining a "type" or ID for each packet, which is used to inform the receiver whether the stream is compressed. Another useful feature is the ability to balance latency and reliability by sending packets with different priorities and requirements for reliability and ordering. In this version of the software, the network part is set up to be completely symmetric, in the sense that the sending part is always the client which initiates the connection and the receiver always has the role of the server, thus having two independent network connections and both a client and a server in both ends.

3.3. Fallback Versions

In order to have a working demonstration if external factors failed, two limited versions of the software was written: the *Visitor*-only version and the trackerless version.

The *Visitor*-only version was made to allow demonstration in case there were issues with the network connection between *Visitor Site* and *Destination* so only LAN was available. This edition was made by using a second tracked object as a virtual source at the *Visitor Site*. The audio was obtained using the input from the *Visitor's* microphone directly and having someone talk into it from some distance. In this way the sound of a speaking person could be moved around the *Visitor* while the *Visitor* was still free to move around and perceive the sound correctly, thus demonstrating the interactivity aspect applied to a live sound.

The trackerless version is the most limited edition by excluding all tracking and simply supplying an option to set the wanted direction in a simple GUI, again with the directly connected microphone as in the previous case.

4. Demonstration and Conclusions

The demonstration was held on February 11th 2011 at the University of Barcelona (UB), with *Destination* equipment set up at University College London (UCL) a few days before. Unfortunately miscommunication with the IT department at UB (as well as a couple of other minor issues) meant that the required connection from UCL to UB (UB acting as a server) could not be achieved, thus the full demonstration could not take place. An important conclusion from these network issues is that we should not rely on the

Visitor Site to have an open network which is externally accessible. A solution to this issue is in the future to work with a normal client-server architecture, under the assumption that the *Visitor* is always a client connecting to a server at the *Destination*

Instead of using the full solution, work was put into setting up and testing the *Visitor*-only version with the installed InterSense system, using the head mounted display as the head position and using a so-called "wand" (tracked controller for InterSense) as the virtual source before this was demonstrated for the reviewers.

The response from the demonstration was overall very positive regarding both the "realism" and usefulness in the project. One comment was that this audio technology should be more closely incorporated in the work of the other partners, thus making the experiences more immersive. Other comments included the wish for a system, which is not dependent on the *Locals* wearing any equipment, so another important conclusion from this demonstration is that work should be put into examining methods of obtaining good recordings for generic binaural reconstruction based on different types of microphone arrays.

The overall conclusion from the review is that interactive binaural synthesis is an important aspect of a fully immersive telepresence application, that we should continue in this direction and attempt to reach solutions with different approaches.

Acknowledgement

The BEAMING Project is sponsored by the EU as a four year collaborative FP7[4] project (project no. 248620), started on January 1st 2010.

References

[1] BEAMING Project. Beaming website. Internet, April 2011. http://beaming-eu.org.

[2] J. Blauert: Spatial Hearing. The Psychoacoustics of Human Sound Location. The MIT Press, 1983.

[3] D. Hammershøi and H. Møller: Binaural Technique: Basic Methods for Recording, Synthesis, and Reproduction - In: Communication Acoustics, pages 223–254. 2005.

[4] D. Brungart, A. J. Kordik, and B. D. Simpson: Effects of headtracker latency in virtual audio displays. J. Audio Eng. Soc, 54(1/2):32–44, 2006.

[5] J.-M. Valin, T. Terriberry, C. Montgomery, and G. Maxwell: A high-quality speech and audio codec with less than 10-ms delay. Audio, Speech, and Language Processing, IEEE Transactions on, 18(1):58–67, jan 2010.

[6] J. Hiipakka, T. Ilmonen, T. Lokki, M. Gröhn, and L. Savioja: Implementation issues of 3d audio in a virtual room. In 13th Symposium of IS&T/SPIE, Electronic Imaging, volume 4297B, San Jose, California, USA, jan 2001.

[7] M. Naef, O. Staadt, and M. Gross. Spatialized audio rendering for immersive virtual environments, 2002.

[4] Seventh Framework Programme for Research and Technological Development http://cordis.europa.eu/fp7/home_en.html