9-2013

Binaural Spatialization for 3D immersive audio communication in a virtual world

Davide Andrea Mauro
Marshall University, maurod@marshall.edu

Rufael Mekuria

Michele Sanna

Binaural Spatialization for 3D immersive audio communication in a virtual world

Davide A. Mauro
Institut Mines–Tèlècom,
TÈLÈCOM ParisTech,
CNRS-LTCI
37-39, rue Dareau
75014, Paris, France
mauro@enst.fr

Rufael Mekuria
Centrum Wiskunde & Informatica
Sciencepark 123, 1098 XG
Amsterdam, The Netherlands
rufael.mekuria@cwi.nl

Michele Sanna
Queen Mary University
Multimedia & Vision Research Group
London, UK
michele.sanna@eecs.qmul.ac.uk

ABSTRACT

Realistic 3D audio can greatly enhance the sense of presence in a virtual environment. We introduce a framework for capturing, transmitting and rendering of 3D audio in presence of other bandwidth savvy streams in a 3D Tele-immersion based virtual environment. This framework presents an efficient implementation for 3D Binaural Spatialization based on the positions of current objects in the scene, including animated avatars and on the fly reconstructed humans. We present a general overview of the framework, how audio is integrated in the system and how it can exploit the positions of the objects and room geometry to render realistic reverberations using head related transfer functions. The network streaming modules used to achieve lip-synchronization, high-quality audio frame reception, and accurate localization for binaural rendering are also presented. We highlight how large computational and networking challenges can be addressed efficiently. This represents a first step in adequate networking support for Binaural 3D Audio, useful for telepresence. The subsystem is successfully integrated with a larger 3D immersive system, with state of art capturing and rendering modules for visual data.

Categories and Subject Descriptors

H.4.3 [**INFORMATION SYSTEMS APPLICATIONS**]: Communications Applications—*Computer conferencing, teleconferencing, and videoconferencing*; H.5.1 [**INFORMATION INTERFACES AND PRESENTATION**]: Multimedia Information Systems—*Artificial, augmented, and virtual realities*; H.5.5 [**INFORMATION INTERFACES AND PRESENTATION**]: Sound and Music Computing—*Systems*

General Terms

Human Factors, Design

Keywords

3d audio, tele-immersion, binaural spatialization, networking, synchronization

1. INTRODUCTION

In one of the definitions of Virtual Reality [12], simulation does not involve only a virtual environment but also an immersive experience. According to another author [9], instead of perception based on reality, Virtual Reality is an alternate reality based on perception. An immersive experience takes advantage of environments that realistically reproduce the worlds to be simulated.

Thus the next challenge in tele-presence is tele-immersion, where individuals that are geographically apart interact naturally with each other in a shared 3D virtual environment. Where teleconferencing allows participants to share a common space, tele-immersion allows them to share an activity. 3D environments and rendering are already well developed as games and virtual environments. On top of that, 3D tele-immersion enables participants to also be captured, reconstructed and merged into such worlds seamlessly. In other words, Visual and Aural 3D representations are not only rendered and transmitted but also captured/reconstructed on the fly in real-time, merging the real and virtual world. Advances on 3D reconstruction and rendering – and the success of consumer grade depth camera's – enable, in real-time, reconstruction of highly realistic representations of the participants as triangle 3D models, point clouds or video plus depth. Similarly, head-trackers and microphone arrays enable the capture of 3D immersive audio for tele-presence.

As many research have demonstrated the benefits of audio as larger than visual communication in an interactive dialogue, we focus our attention to the 3D audio domain in this paper. Firstly, we define the techniques used to render 3D audio in a virtual immersive environment. We detail the implemented binaural spatialization technique, that can simulate 3D audio and reverberation based on the user and sound sources in the scene. Then we describe how audio content is captured at the end points, with compact microphone arrays available in consumer grade depth camera's such as Microsoft KINECT. This allows a more sophisticated scene acquisition with respect to monophonically captured signals. Second we highlight the networking issues related to the distributed nature of the system. We demonstrate

an architecture for flexible transmission of both bandwidth savvy visual streams and loss sensitive audio streams. We apply coordination and prioritization of the streams to limit the effects of congestion introduced by bandwidth savvy visual streams. We also demonstrate the implementation of a virtual distributed clock, that is useful for synchronization of the streams and positions of the different sources and a media stream synchronization mechanism.

2. 3D AUDIO

It is now clear that such a system will greatly benefit from proper implementations of audio rendering. With a standard headphones system, normally, sound seems to have its origin inside the listener's head. This problem can solved by binaural spatialization over headphones, which gives a realistic 3D perception of a sound source located somewhere around a listener. Moreover there will exists real-time and complexity constraints that will push for implementations able to fulfill such requirements. With 3D recordings, audio, and rendering systems we mean techniques that aim to create the perception of sound sources placed anywhere in a three-dimensional space. It is important to note that a properly rendered audio environment can enhance the sense of immersion and presence in the scene, but the problem of recreating an immersive experience is not trivial.

Binaural spatialization is a technique that, unlike other systems that require a consistent number of loudspeakers and channels, aims at reproducing a real sound environments using only two channels (like a stereo recording). It is based on the assumption that our auditory system has only two receivers, i.e. the ears, that, in order to obtain a representation of the acoustic environment, exploits some physical parameters of the signal called "cues" [14, 2]. There exist mainly three different so called "localization cues": ILD (Interaural level difference), ITD (Interaural time difference), and DDF (Direction dependent filtering) — a filtering effect with respect to the position of the sound source. While the first two are regarded as interaural differences the latter is essentially a monaural attribute that still works using only one ear.

- ILD (Interaural Level Difference) [2] represents the difference in intensity between the ears and it is usually expressed in dB. It is most effective for high frequency (above 1 kHz) where the head act as an obstacle generating an acoustic shadow and diffraction on its surface.
- ITD (Interaural Time Difference) [2] represents the delay of arrival of the sound between the two ears (usually expressed in ms). In a real context both the cues cooperate in order to get a correct localization (even if these two parameters alone generate the so called cone of confusion [5]) of sound but they tend to work on different parts of the spectrum (according to the Duplex Theory originally proposed by Lord Rayleigh in 1907 [6].)

For low frequencies, whose wavelength is bigger than the radius of the head, the head itself does not act as an obstacle giving no significative intensity variations as the wave diffracts around the head. For this reason our hearing system exploits the use of ITD. While the frequency increases

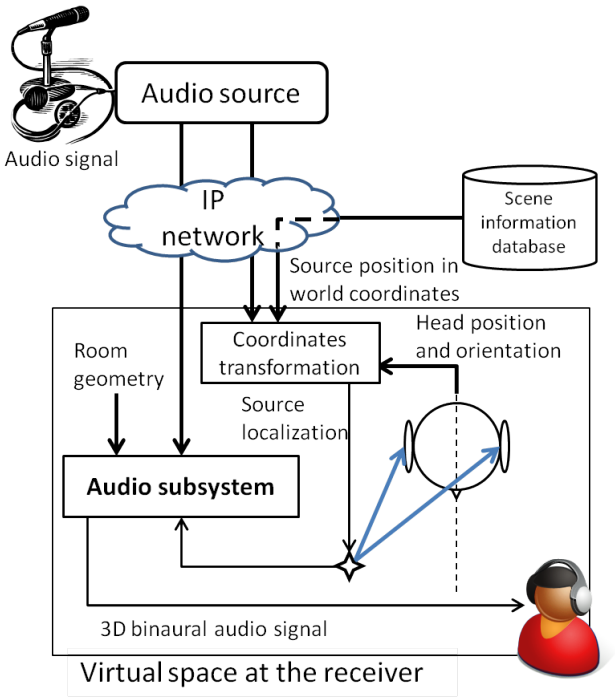

Figure 1: Overall communication architecture for the virtual scene composition and rendering.

the period of the signal becomes comparable with the ITD itself giving no opportunity to distinguish, e.g., between a sound in phase because arrived at the same timing or shifted by one period. So ITD becomes less useful for frequencies greater than 800 Hz, while some evidence suggests that is still possible to analyze changes in the spectral envelope up to 1.6 kHz [8]. The previous two interaural differences alone cannot account to explain how it is possible to distinguish between: e.g. a sound located at an azimuth of 30° and a sound with an azimuth of 150°, because they will yield to identical interaural differences. In this case a new effect arise caused by a selective filter due to the different position of sounds. This effect known as direction-dependent filtering is caused by the shape and the position of the pinna.

If it is possible to deliver a signal equal (or nearly equal) to the one which a subject would receive in a real environment, this will lead to the same perception. It is well suited by headphones where each channel can reach only the required ear but also a pair of loudspeakers can be used taking into account crosstalk and facing it with cancelation mechanisms. Binaural spatialization can be achieved through various processes, such as: equalizations and delays, or convolution with the Head-Related Impulse Response (HRIR). The latter approach is the one we have followed in our work. In order to obtain these impulses, many experiments involving the use of a dummy head[1] have been made, thus creating various databases of impulse responses. Currently, most projects using binaural spatialization aim at animating the source while keeping the position of the user fixed. However, for an immersive experience this is not sufficient: it is necessary

[1]A dummy head is a mannequin that reproduces the human head.

to know at any time the position and the orientation of the listener within the virtual space in order to provide a consistent signal [10], so that sound sources can remain fixed in virtual space independently of movements, as they are in natural hearing [1]. It is worth to note that a proper recreation of binaural cues can enhance the speech intelligibility such as in the "cocktail party effect" (where the listener is able to distinguish a single audio source in a noisy crowd) and so the benefits are not only limited to a more pleasant "aesthetic" effect.

2.1 Audio source: Capture of audio signal and virtual scene composition

For appropriate rendering of the 3D audio from a remote source, a number of different challenges arise. Audio should be captured in real-time and transmitted over network to the other participants in the virtual world. However the audio signal is not enough to perform 3D audio synthesis at the receiver. The audio source needs to be localized in a virtual world, where both sender and receiver have the same reference system. Fig. 1 shows the information exchanged in the system to allow the Audio Subsystem at receiver to perform the spatial audio synthesis. The information flow from the sender is composed of two fundamental streams:

1. The mono-channel audio signal, captured at the source.
2. The position of the audio source in world coordinates, according to a specific reference system.

With the aid of a scene information database, the receiver is able to localize the audio source in the virtual environment, and compute the reciprocal positions of the listener, whose position is well known by the receiver itself. Additionally, information about the room geometry can be used for calculating sound reverberation and enhance the realism of the listening experience. The scene information database can be localized in any point of the network, as long as a master entity (a server or the receiver itself) makes sure that both source and receiver (slaves) share the same world coordinates. Additionally, the information about the room geometry can be retrieved from the server or computed locally at the receiver, allowing a unique and personalize immersive listening experience.

As the audio source moves inside the scene, the information about its position needs to be sent and updated at the receiver in real time. On the contrary, a change in the listener's position is only relevant locally to the receiver, and although triggering a dynamic adaptation of the signal to the movement (as explained in the next section) it does not involve retransmission of parameters.

The transmission engine of this system needs to ensure that the receiver has a constantly updated and reliable information about the location of the sound source in space. Synchronization of the position information with the audio stream is vital to ensure coherent and timely spatialization of the sound.

2.2 Audio rendering

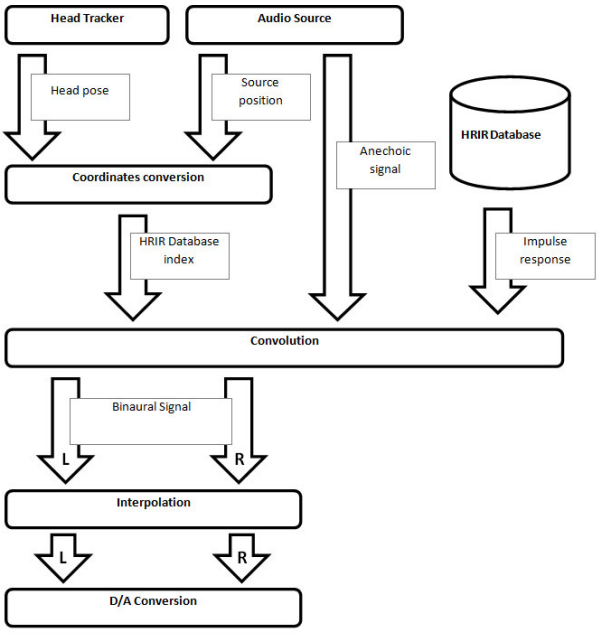

Figure 2: The workflow diagram of the audio subsystem.

Via this distributed system, the receiver is able to retrieve all the necessary information to define: 1. the position and the orientation of the listener, and 2. the position of the audio source. Given this information, it is possible to calculate the relative position of the listener with respect to the source in terms of azimuth, elevation and distance, and which impulse response to use for spatialization. Once the correct HRIR is obtained from the database, it is possible to perform the convolution between the monophonic audio signal in input and the stereo impulse response (HRIR) chosen from the database. A schematic overview of the system is depicted in Fig. 2.

Since the system is dynamic, the position of both the listener and of the source can change over time, an interpolation mechanism to switch between two different HRIRs needs to be implemented. We have designed a simple yet performing interpolation procedure based on crossfade to limit the artifacts produced by the switch between impulses: the approach is computing two audio stream for each channel when needed (current and previous impulse.) Then the new signal will gradually overcome the signal from other filter with a crossfade function. As a performance issue it should be noted that in a real time environment every redundant operation should be avoided, this means that the procedure take place only if a change has been detected in the "world". The next step is to perform the necessary operations for spatialization: FIR (Finite Impulse Response) filtering or frequency-domain convolution.

Technically we define $i(x)$ as the input signal, $h_L(x)$ and $h_R(x)$ as the impulse responses for left and right ear and

$h_{L,R}(x)$ as the output:

$$o_{L,R}(x) = \begin{cases} i(x) * h_L(x) \\ i(x) * h_R(x) \end{cases} \quad (1)$$

This technical choice depends mostly on the "size" of the filter kernels. For short filters FIR filtering performs very well, while, given the complexity, moving to longer impulse responses (such as the ones of reverberant environments) could lead to performance degradation. In the latter case implementing convolution in frequency domain via FFT is a convenient strategy. The computational complexity for the former approach is $O(n^2)$ while for the latter is roughly $O(n \log(n))$, and thanks to the convolution theorem we can rewrite it as to be the product of the spectra:

$$O_{L,R}(x) = \begin{cases} I(x) \cdot H_L(x) \\ I(x) \cdot H_R(x) \end{cases} \quad (2)$$

All the audio files used in the project are PCM uncompressed files and have a sample rate of 44.1 kHz and a quantization word of 16 bit. With this bit depth the theoretical dynamic range is ~ 96 dB. We can give then an "empirical" threshold of 4096 samples for switching between time and frequency domain approaches. The CIPIC database here used has impulses of 200 samples, allowing for efficient FIR implementation while the longer BRIRs (Binaural Room Impulse Responses) for room simulation can easily be 1 sec long (i.e. 44100 samples.)

The proposed implementation is *per se* multiplatform, is written in C++ and makes use of:

- Libsndfile for I/O;
- FFTW3;
- PortAudio for audio driver communication.

It is worth to cite that a number of different solutions and strategies emerge here and it is worth to note that GPGPU (General Purpose computing on Graphic Processing Units) can be exploited (see [4].) Hybrid approaches CPU/GPU can be implemented where the low latency of the CPUs can be used to computer the direct path of the sound while the extreme power of the GPUs, with appropriate systems to minimize the I/O delay (see [13] for details), can be used to add the reverberant part.

3. REAL-TIME STREAMING

Fig. 3 shows the immersive audio-visual media pipeline. At the sender side, different types of media are captured in real-time: audio, motion from sensors (for model-based avatars), and visual data. These media data are compressed and sent over a (lossy) IP Network to the remote destinations where they are rendered in real-time. Transmission in Real-Time of Visual data, such as video plus depth or reconstructed geometry (triangle meshes and point clouds) over a lossy IP network is challenging due to the high volume (byte size) of the data. While other work studies the streaming of 3D visual data, we are interested in reducing the artifacts such streams may have on the quality of the 3D spatial audio. We identified four technical issues of joint network streaming of 3D audio and Visual data that may reduce the quality of the received audio. The specific challenges of joint streaming of audio and 3D video are listed below.

1. Intra-stream synchronization: Due to the high volume of data sent over the network and the routing mechanism of IP Networks, packets will arrive out of order with different amounts of transmission delay (jitter), due to jitter the original time spacing between the audio is lost, degrading intra-synchronization quality and possibly the 3D Perception.

2. Distributed Position Synchronization: As the calculation of the FIR requires as input positions from other sites, sites should have time accurate information of the position of all objects in the scene in any given time. Practice has shown that terminals synchronized with the NTP protocol can still be out of sync more than 50 ms, so NTP synchronized timestamps may not be enough for accurate position synchronization.

3. Data loss due to congestion caused by high volume visual streams: (Random Early Drop) RED is employed by IP-Routers to combat congestion of if the packet volume increases. As the system sends both high volume visual and loss sensitive audio data, audio packets in bottle neck links can be lost due to congestion incurred by the high volume visual streams.

4. Media Synchronization: Visual and Audio data should be rendered synchronously to yield lip-sync. Existing synchronization requirements for audio-video synchronization indicate a maximum difference of 80 ms.

We developed a streaming component that addresses these issues to minimize artifacts, in this case for binaural spatial audio rendered that is based on positions in the virtual world with several counter measures. 1. is solved with an appropriate jitter buffer, a buffer that holds a number of packets/frames, and restores the original order and inter-frame timing. To address 2. we introduced an application layer virtual clock, this is a clock that defines a globally synchronized clock-time in the application. The globally synchronized timestamps can be used to accurately calculate the positions of the different objects at the remote terminals. To combat 3. we introduce a multi stream-coordination component. This components handles all networked media streams jointly before sending packets to the network. Coordination of streams can be deployed to protect audio streams, for example by allowing the audio stream more bandwidth or relative transmission time. To address 4. we implemented a method for synchronization of audio and multiple visual streams, based on existing synchronization requirements for audio/video systems (a difference of about 80 ms).

3.1 Synchronized Virtual Clock

To accurately determine the time instants related to the positions of objects in the virtual world, accurate wall-clock time synchronization is required between the terminals. Normally, the Network Time Protocol (NTP) is used in the internet to synchronize the clock times of terminals between different locations. However, measurements of NTP Clock Time differences between servers and clients have shown that

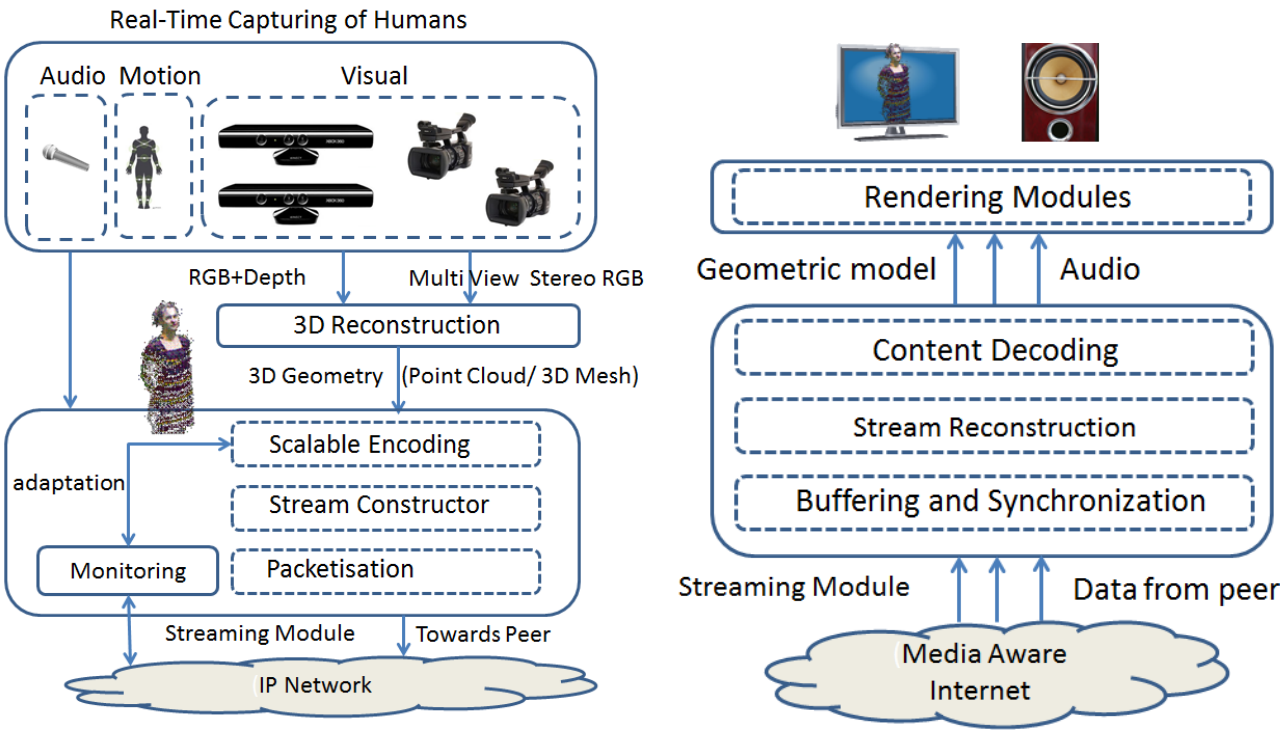

Figure 3: Source location (left) and destination location (right) of the media pipeline.

time differences up to 60 ms can occur. Moreover, not all receiver terminals are NTP synchronized with the same NTP network, and some are not NTP Synchronized at all resulting in even larger differences. For these reasons, NTP is insufficient to achieve accurate clock synchronization for distributed position measurements as needed for spatial audio rendering. Instead, we developed a virtual clock, that is synchronized between the different terminals in the session via the network and exists only within the application. The deployed protocol is very similar to the precision time protocol (PTP). The precision time protocol is an alternative protocol for synchronization of terminals in LAN/WAN Network that can achieve better precision compared to NTP. Fig. 4 illustrates the operation of the virtual clock. One of the participants is assigned as the time provider and shares its local timestamps on request to the other terminals that synchronize to this time. This is implemented by storing an estimate of the time difference between timestamps of the provider and the receiver client. The application then can obtain a globally synchronized timestamp at any time, which is the local system clock timestamp adjusted with this offset. The offset is updated at a frequency of once per second, in order to maintain the synchronization without increasing the data rate. The sequence diagram of the update protocol and the algorithm used to estimate the offset are shown in Fig. 5, the client sends its local timestamp epoch to the provider, and the provider sends back its own local and the received epoch. The offset o is computed as

$$o = Tp - (T1c - T2c)/2 \tag{3}$$

This formula is based on the assumption of symmetric delays in the network, which is generally considered a reasonable assumption. In the implementation conversion func-

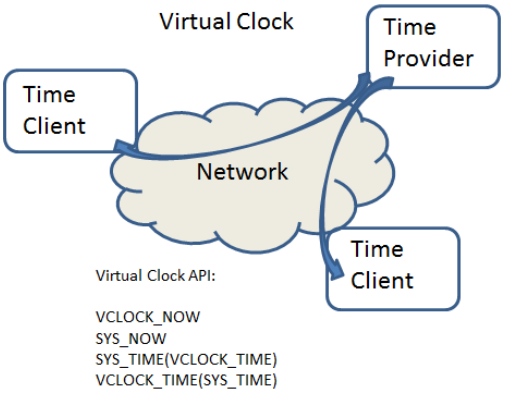

Figure 4: Global Timestamp distribution for virtual clock synchronization with local to global conversion API

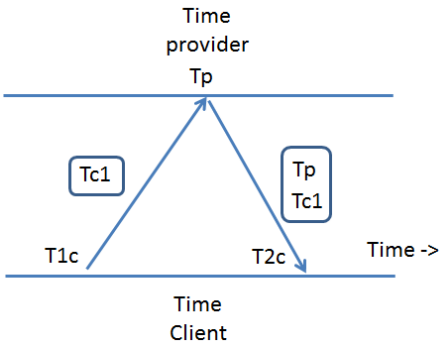

Figure 5: Sequence diagram for sync messages for virtual clock synchronization

tions are implemented to convert from local system time to the global system time, for example VCLOCK NOW gives the current virtual clock timestamp, while the macro SYS TIME(VCLOCK TIME) gives the local timestamp corresponding to a global timestamp.

3.2 Network Stream Prioritization Component

The data generated in a 3D Tele-Immersive system in Fig. 3 is high in volume and requires a low transmission delay. For such data Quality of Service (QoS) is desired, which implies certain guarantees from the network on the maximum delay, the available bandwidth and the transmission quality (package loss rate) of the channel (QoS parameters). As the internet protocol is based on the best-effort principle (which means package losses, delays and bandwidth fluctuations), this is challenging to achieve. While there are some examples of QoS protocols that are deployed in the core service provider networks, generally only best effort can be expected by an application in the internet. So often applications requiring QoS for Real-Time Media provide some alternative ways to achieve it (i.e. based on efficient error-resilient coding and/or bit-rate adaptation in case of changing network conditions). These methods achieve the desired QoS on the application level rather than the packet (network) level. In practice, reasonable results are obtained for traditional media services such as voice communication (VoIP) and video in the internet by using efficient compression and streaming methods and sometimes error correcting codes. In our system many streams that represent avatar movement data, reconstructed 3D audio and reconstructed 3D visual data are present. This makes achieving appropriate Quality of Service more difficult for two reasons. First, the QoS requirements, related to user perception and efficient data representation/compression are understood less well and therefore still ambiguous. Second, interference between streams can be an issue if high volume 3D visual data is sent to the network in an uncontrolled manner. In such a case network congestion degrades the audio quality.

To address these streaming issues, we developed a generic framework that handles all streams jointly before sending packets to the network. In case packet loss and delay increase at the receiver due to congestion, the outgoing visual data-rate is decreased to protect the audio and other loss sensitive data such as avatar movements. This works well,

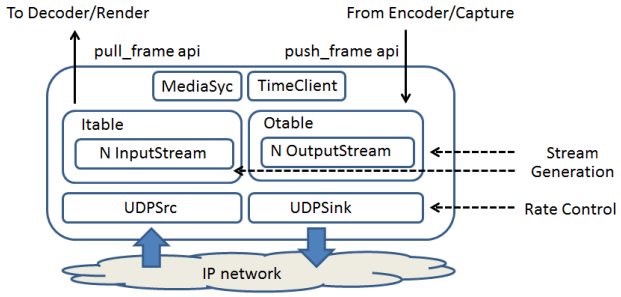

Figure 6: Streaming System for scheduling and prioritization

as a slight degradation of the visual quality or visual frame rate is generally preferable to having interruptions in an ongoing audio conversation. Due to flexibility of this component, other streaming policies can easily be added when QoS requirements for 3D tele-immersive streams are better understood. These requirements can be obtained by pilot user studies or from existing perceptual models based on the properties of the media representation. Fig. 6 illustrates the network stream prioritization component. The main subcomponents are the incoming and outgoing streams that are stored in their respective tables Itable and OTable. The incoming streams (inputstream), allow a media frame to be retrieved from a sequence of received packets via the defined pull frame function. The outputstream on the contrary creates a sequence of packets from a compressed media frame via the defined pushframe function. Packets are received through the UDPSrc, which forwards received packets to the respective input stream stored in the ITable de-multiplexing. The UDPSink reads packets from the queue generated by push frame in each Outputstream, and controls the outgoing data rate and stream prioritization by sending packets in at adjusted rates to the network. In this system, we assign extra priority and send bandwidth to the audio and position data streams. Contrary to systems based on the Real-Time Protocol, UDP and Real-Time Control Protocol, this system uses only one UDP port instead of using different UDP port numbers for each media and control stream. The advantage of multiplexing all streams over one port is that it is easier to manage the incoming and outgoing data, requires only one firewall port to be opened and avoids possible conflicts between assigned ports. The component also has an instance of the virtual Clock (Time Client) described in section 3.1, that it uses to write synchronized timestamps to the packets. Moreover, it has a mediaSync component that is described in the next section that is used to synchronize streams that are coming from the network.

3.3 Intra- and Inter- Media Stream Synchronization

Over the years, many techniques to achieve intra- and inter-media synchronization in various network conditions have been developed. They are mostly concerned with intra-stream synchronization which implies recovering the original temporal ordering of a frames in a single media stream and inter-stream synchronization which implies recovering synchronization between two different streams. A recent survey on media synchronization in packet networks was given in

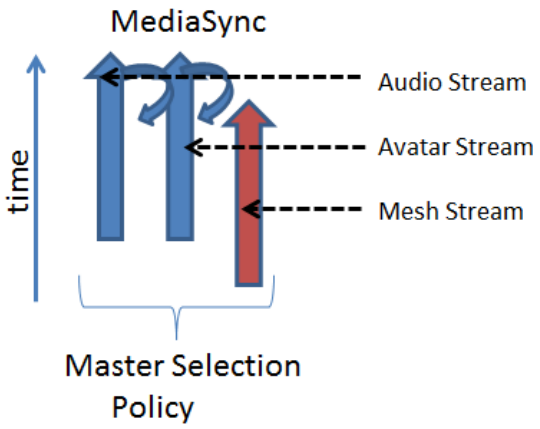

Figure 7: Streaming System for scheduling and prioritization

[3] which includes most inter-stream synchronization algorithms up to 2009.

To achieve intra stream synchronization of the 3D Audio stream, we implemented a jitter buffer that stores audio frames received from the network. Only when the number of frames is above the *start* threshold, frames will start to be passed to the rendering engine. On the other hand, when the buffer contains only $starve < start$ frames, it will stop passing frames to the render system that will render silence instead. In the mean time, the number of frames in the jitter buffer will increase to *start* frames, when renderer starts rendering again. As the renderer renders at the same fixed rate, corresponding to the frame rate for capturing the audio, intra-stream synchronization is achieved. In the current implementation we chose *starve* to be two 20 ms PCM sampled audio frames (40 ms of audio) and start to be 4 PCM samples of 20 ms (80 ms). In general such delays are considered acceptable for voice conversations. In this 3D tele-immersive system, multiple incoming and outgoing streams have to be synchronized (inter-stream synchronization). The most common method for synchronization of multiple streams is to buffer frames while they are received from the network and only render them when the timestamps of each of the media frames of the different streams are aligned within the synchronization margin. The synchronization component synchronizes multiple media streams independent of the sample-rate based on the monitoring of frame arrivals of the master stream. Before we continue discussion of this component, we define a few useful terms, the *master-selection policy* refers to the method that is used to select the *master stream*, which is the stream to which other streams are actively synchronized. The *re-synchronization policy* refers to the method of re-synchronizing the non-master, i.e., *slave streams* actively to the master stream. Possible re-synchronization policies include skipping or delaying frames or changing the playout rate or freezing frames. In the current system, we set the visual stream with reconstructed geometry (point cloud, triangle mesh), to be the master-stream (fixed master selection policy), as these frames generally experience a higher network delay than the audio frames due to its high volume. In this case, audio frames can be delayed to match the delay of the visual stream. For each received mesh frame, both capture and reception times (on the scale of the global virtual clock implemented in defined in 3.1) are stored. Based on this data, an average delay is computed, that is increased with a small extra delay for the buffer (range of 10-50ms). Then based on the estimate (currently a moving average over the last 5 frames), the slave streams are delayed (or skipped), to match the master stream. The procedure of master stream selection and re-synchronization is illustrated in figure 7.

4. CONCLUSIONS AND FUTURE WORKS

With the ever increasing availability of high powered computers, high-speed broadband and low-cost 3D capture and rendering technologies, truly immersive online interactions are becoming more of a reality. An architecture for networking and Binaural spatial Audio in Virtual environments with bandwidth savvy streams has been presented in this paper. Based on affordable and readily available technology, such technology could already been successfully deployed and provide 3D audio experience. The feeling of 3D immersion in the visual environment is thus complemented with spatialized 3D audio from all participants with consideration of the acoustics of the geometry of the virtual room. Specific synchronization mechanisms are important, as desynchronization possibly directly affects the spatial audio experience, we outlined the approach taken in this immersive framework based on a virtual clock with a PTP like protocol. Moreover, a more flexible network streaming engine has been implemented in order to coordinate the real-time and bandwidth constraint streams for the diverse media data such as reconstructed 3D, video, motion and spatial audio.

The presented audio subsystem can be also extended by adding room acoustic simulation (based on room geometry) or implementing dedicated reverb; this helps to enhance even more the sense of immersion and the "coherence" with the virtual environment. In this scenario will be also interesting to evaluate the possibility of using codings that allows to stream compressed audio and add metadata that enable the reconstruction of the 3D scene at the receiver side such as VBAP (see [11]) or DReaM [7]. Another critical aspect for the correct perception of 3D audio is related to the individualization of the HRTF's (Head Related Transfer Function) set. Given the variability in shapes of human ears each subject is "tuned" for listening through is own. In the context of the proposed framework, where reconstruction of avatars is already taken into account, if a sufficient resolution it is possible to think about individualization based on image analysis or anthropometric features computed from the acquired images.

The next steps in the development process will involve the feasibility of finding appropriate ways of splitting the rendering pipeline to exploit the computational power available in a client-server infrastructure (or in the cloud) in order to support a large number of simultaneous users. Achieving realtime rendering for multiple users is one of the key requirements and the effort will be to define and implement efficient strategies for coding and transmitting data over the network preserving synchronization. We can also mention that further investigations are currently carried out in computing 3D Audio on GPUs (Graphic Processing Units) as

well as efficient algorithms for computing room acoustic.

5. ACKNOWLEDGMENTS

This work has been partially funded by the European Commission under contract "FP7-287723 REVERIE".

6. REFERENCES

[1] D. R. Begault. *3-D sound for virtual reality and multimedia*. Academic Press Professional, Cambridge, MA, 1994.

[2] J. Blauert. *Spatial Hearing: The Psychophysics of Human Sound Localization*. MIT Press, Cambridge, MA, 1996.

[3] G. M. Boronat F., Lloret J. Multimedia group and inter-stream synchronization techniques: A comparative study. . *Elsevier Inforation Systems*, 34(2):108–131, 2009.

[4] B. Cowan and B. Kapralos. Spatial sound for video games and virtual environments utilizing real-time gpu-based convolution. In *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*, pages 166–172. ACM, 2008.

[5] E. Hornbostel and M. Wertheimer. Über die wahrnehmung der schallrichtung. *Sitzungsberichte der Preusslichen Akademie der Wissenschaften*, 20:388–396, 1920.

[6] O. Lord Rayleigh. XII. On our perception of sound direction. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 13(74):214–232, 1907.

[7] S. Marchand, R. Badeau, C. Baras, L. Daudet, D. Fourer, L. Girin, S. Gorlow, A. Liutkus, J. Pinel, G. Richard, et al. DReaM: A Novel System for Joint Source Separation and Multitrack Coding. In *Audio Engineering Society Convention 133*, 2012.

[8] B. C. J. Moore. *An introduction to the Psychology of Hearing*. Elsevier academic press, fifth edition, 2004.

[9] K. Osberg. *But what's behind door number 4? Ethics and virtual reality: A discussion*. Human Interface Technology Lab Technical Report R-97-16, 1997.

[10] S. P. Parker, G. Eberle, R. L. Martin, and K. I. McAnally. Construction of 3-D audio systems: background, research and general requirements. Technical report, Victoria: Defence Science and Technology Organisation, 2000.

[11] V. Pulkki. Virtual sound source positioning using vector base amplitude panning. *Journal of the Audio Engineering Society*, 45(6):456–466, 1997.

[12] J. Steuer. Defining Virtual Reality: Dimensions Determining Telepresence. *Journal of Communication*, 42(4):73–93, 1992.

[13] F. Wefers and M. Vorländer. Optimal Filter Partitions for Real-Time FIR Filtering using Uniformly-Partitioned FFT-based Convolution in the Frequency-Domain. In *Proceedings of the 14th international conference on digital audio effects : September 19-23 : IRCAM, Paris, France / Institut de recherche et coordination acoustique musique (Paris)*, pages 155–161. IRCAM-Centre Pompidou, 2011.

[14] W. A. Yost. *Foundamentals of hearing: An introduction*. Academic press London, 1994.