Spatialized streaming audio in multi-user virtual training environments

Matjaž DIVJAK, Damjan ZAZULA, Danilo KORŽE, Dean KOROŠEC

Faculty of Electrical Engineering and Computer Science

University of Maribor

Smetanova 17, 2000 Maribor, Slovenia

{matjaz.divjak, zazula, korze, dean.korosec}@uni-mb.si

Résumé – Dans cet article, nous décrivons l'ajout d'une communication audio 3D au *Virtual Reality Modelling Language* (VRML). Celui-ci permet la communication vocale directe entre participants des environnements virtuels d'entraînement. Afin d'obtenir les meilleurs résultats possibles, nous avons étudié divers modèles acoustiques spatiaux et diverses possibilités actuelles du langage VRML. L'adaptation d'un modèle acoustique 3D aux caractéristiques des environnements-cibles d'entraînement medical, ainsi que l'intégration -indépendante de la plateforme utilisée- d'un système de *streaming media* au langage VRML constituent notre contribution au développement de ces environnements.

Abstract *– In the paper the development of the spatialized audio communication add-on for Virtual Reality Modelling Language (VRML) is described. It enables a direct voice communication with 3D sound effects among participants of the virtual training environments. In order to achieve best results various spatial sound models and current VRML capabilities have been investigated. Our contribution is the adaptation of the spatial sound model to the characteristics of target medical training environments and the platform independent integration of a streaming media system with VRML.*

1. Introduction

For learning and training of difficult, dangerous or expensive tasks computer simulations are often used. They offer safe, controlled and adaptable training environments. Such simulations are especially useful in medicine, where "learning by accident" is not a desired form of education. For example, to practise neonatal resuscitation (post-delivery treatment of a deprived newborn), doctors work together in a room that imitates the real delivery room, but with rubber dolls representing the mother and the newborn. To experience the feeling of emergency and stress, doctors must pretend to be working with live patients, trying to solve various health complications.

The use of multi-user virtual training environment can make such practice much more realistic, unpredictable and instructive. A dynamic virtual model of a newborn can provide much greater functionality than a rubber doll. For example, the internal organs can easily be made visible during a virtual surgery. On the other hand, real environments have their own advantages. A rubber baby can easily be touched, carried or intubated, all with very realistic physical senses. Such senses are currently impossible to realistically simulate in virtual environments. Therefore, the creators of training systems are trying to improve the realism of the training by using other techniques.

Our research was focused on realistic voice communication between the trainees. Because the voice of each participant is usually recorded by a microphone, the information about the 3D position of the speaker and the surrounding environment must be added to it. Therefore, the properties of spatial sound models and streaming technologies are investigated in the next section. Then, the limitations of the Virtual Reality Modelling Language (VRML) are presented. In section 4, the adaptation of spatial sound model to the characteristics of medical training environments is described, followed by the proposed method for integration of the developed model with the VRML environments. At the end, a distributed application for medical training is described and final conclusions are presented.

2. Spatial sound and streaming

The ability of the human auditory system to localize sound sources is an important component of our perceptual systems and a significant source of information about our environment. This capability has been thoroughly studied and although some mysteries remain, the major cues for extracting directional information from sound have been known for a long time. In order to add spatial information to arbitrary sound, various systems have been introduced [1, 2]. The most promising results are achieved using the Head-Related Transfer Functions (HRTFs). The HRTF captures all of the physical cues to source localization, such as diffractions around the head, reflections from the shoulders and the pinnae, etc. It can be effectively used to create the impression of a sound being at any desired 3D location. It is even easy to distinguish between the sounds directly in front of or behind

the user's head, a feature traditional spatial systems have problems with [3].

Due to their complex nature, HRTFs are usually obtained by recording the impulse response of a suitable acoustic channel. This response is called the Head-Related Impulse Response and its Fourier transform is the Head-Related Transfer Function. Binaural signals can be synthesized from a monaural source by performing convolution between the impulse response $h(t)$ and the source signal $x(t)$:

$$y(t) = h(t) * x(t). \quad (1)$$

To eliminate the distortions of different loudspeaker settings, the resulting sound is played through the headphones.

Unfortunately, the physical structure of the pinnae that plays a significant role in elevation perception varies widely across the general population. As a result, the HRTFs are also different for each individual. If incorrect set of functions is used, inaccurate localization results may be achieved. Different approaches are used to accommodate this: standard HRTFs, individualized HRTFs, model HRTFs, etc [1]. Since they are typically measured in an anechoic setting, they don't include the effects of environmental sound reflections, which are very important for externalisation of the sound. Lastly, to achieve real-time operation, the process of spatialization is generally assisted by dedicated hardware.

Usually, the entire source signal $x(t)$ is available for processing. Such static, pre-recorded sounds are useful in many applications, but for direct, real-time communication amongst people, streaming audio formats are used. Streaming breaks audio data into a series of small packets, suitable for transmission. While the first packet is being played, the second one is being decompress and the third received. Examples of streaming are live radio and TV broadcasts with RealAudio or RealVideo technology. Today, two Internet standards are widely used for streaming applications: RTP (Real-Time Transport Protocol) and RTSP (Real-Time Streaming Protocol) [4]. To perform convolution on a streaming signal, suitable windowing must be used [3].

3. Virtual environments

The popularity of artificial environments created using the computer technology is increasing. Such virtual worlds are usually based on the standardized Virtual Reality Modelling Language and are often used in education, training, entertainment, etc. Because of limitations of the existing haptic/kinaesthetic technology, the experience of virtual environments is currently based on visual and audio representation. To increase the realism of immersion, the use of spatial audio is frequently desired. While VRML already provides the support for simple sound spatialization, it doesn't include the support for streaming sound [5]. As a consequence, no dynamic, real-time media content is available inside the VRML environments. Besides reducing the interactivity, this also makes the use of certain applications impossible, especially those requiring inter-environment communication. Therefore, the extension of the VRML specification with spatialized streaming sound was the main goal of our research.

4. Adaptation of the sound model

To achieve better performance inside the VRML worlds and still provide satisfactory localization quality, the spatial sound model had to be simplified. The following properties of target medical training environments were considered. Support for communication between multiple users is needed, therefore spatial effects must be applied to streaming audio in real-time. The platform independence of VRML language must be preserved. Since most of the time users are present inside a virtual room, their heads lie approximately on a plane. Therefore, elevation of the sound is not considered to be of crucial importance.

Using the mentioned constraints, the following spatial sound model has been selected. The location of a sound source in 3D space is described by three spherical coordinates : azimuth, elevation and range. The primary cues for azimuth are Interaural Time Difference (ITD) and Interaural Level Difference (ILD) [1]. For our purposes, only the ITD was selected. It is caused by the fact that sound must travel different distances to each ear. Because the speed of sound in air is finite, it will arrive at different times. The delay can be calculated using the model, depicted in figure 1:

$$\Delta t = \frac{D(\varphi + \sin\varphi)}{2c}, \quad (2)$$

where D is the distance between the ears, φ is the angle between the listener's view direction and the sound source location, and c stands for the speed of sound [2].

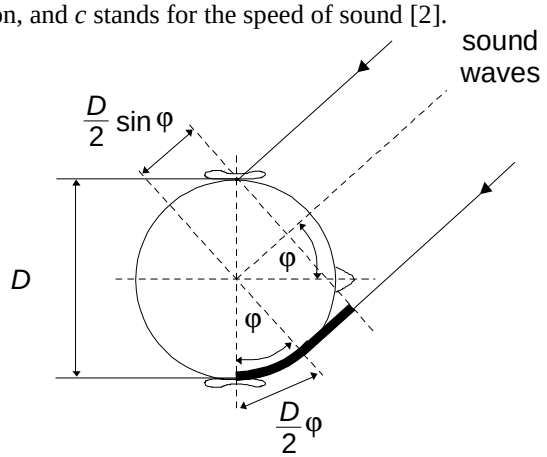

FIG. 1: Model of head for calculation of ITD

The estimation of source's range is not completely understood and different cues are used to simulate it. One of the most frequently used is loudness. It stems from the fact that distant sound sources appear quieter than sources near the listener, due to the occluding effect of the air. In order to simulate the distance to the source, the source's amplitude is modified by the equation (3), where $g(d)$ is normalized sound gain, d is the distance between the sound source and the listener, *MIN_DIST* is the distance at which the sound is heard at full volume, and *MAX_DIST* is the distance beyond

which no sound is heard. Between the *MIN_DIST* and *MAX_DIST* the gain falls off inversely with the square of distance (figure 2).

$$g(d)=\begin{cases}1; d \le MIN\_DIST \\ \dfrac{1}{(d-MIN\_DIST+1)^2}; MIN\_DIST < d < MAX\_DIST \\ 0; d \ge MAX\_DIST\end{cases} \quad (3)$$

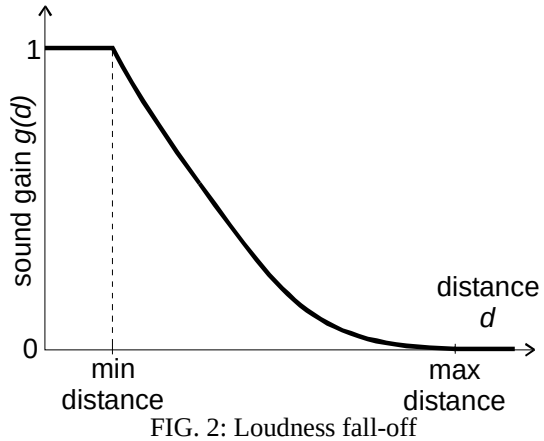

FIG. 2: Loudness fall-off

The perception of elevation is provided primarily by the pinnae. It acts as an acoustic antenna. Its resonant cavities amplify some frequencies and its geometry causes interference effects that attenuate other frequencies. Due to the complexity of mimicking such behaviour and relative unimportance for our application, elevation was decided not to be simulated.

5. Integration with VRML

To integrate the described functionality with virtual environments, we extended the VRML language through the `Script` nodes [5]. `Script` nodes provide a mechanism for creating nodes with custom behaviour and thereby expanding the basic VRML functionality. The behaviour is specified by a script that can be written in a number of programming languages, usually Java or JavaScript. The script is loaded together with the rest of the VRML world and is interpreted at run-time, completely concealed from the user's attention. The described solution doesn't violate the VRML standard and is therefore compatible with all VRML browsers that support the selected scripting language. If Java or JavaScript is used, the resulting virtual environment is also platform independent. Because JavaScript lacks the needed functionality, Java was selected as our scripting language.

Our streaming system is comprised of three parts (figure 3): a VRML world, a loader script and a Java streaming application [6]. As the VRML world completes loading and is ready to start sending events, a single loader script is executed. Its only assignment is to start the main class of the streaming application. Because the application runs in a separate thread, it executes in parallel with the rest of the VRML code.

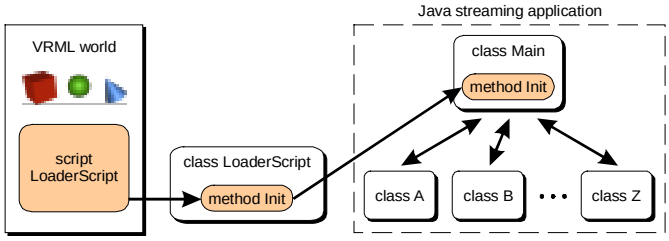

FIG. 3: Elements of the streaming system

Each visitor of the virtual environment has his own copy of the VRML code and the accompanying Java application. Each application acts as a client. It captures sound from the microphone, encodes it in a special format suitable for transmission and transmits it to other clients. At the same time it accepts audio streams from other users, calculates the corresponding sound delays from the positions of users, applies spatial sound effects to the streams and plays the audio through the headphones. Clients communicate through a special server application. The server is mainly responsible for maintaining a list of current users and relaying position and control data among the clients [6].

The basic Java packages do not include support for advanced sound processing. For manipulation of streams a Java Media Framework (JMF) extension package was used [7]. JMF provides a unified architecture and managing protocol for the acquisition, processing and delivery of real-time data. For transmission of media streams over the network it uses the RTP protocol [8]. Its plug-in architecture enables programmers to directly access media data and easily customize and extend the framework's functionality. For our purposes we implemented an effect plug-in called `SpatializeEffect`. It operates on individual RTP packets and adjusts the gain and delay of audio samples in real-time for left and right stereo channels [4].

To add the streaming system to the arbitrary VRML world, only one `Script` node with our loader script code needs to be added to the top-level VRML file. The rest of the Java client application can be packed into an archive and stored on the web server, together with the rest of the VRML environment. That way, the application is automatically downloaded each time it is needed.

6. Application for medical training

The proposed approach was used to develop a communication module for the Virtual Delivery Room system, in which multiple persons collaborate in a neonatal resuscitation training [9]. Virtual Delivery Room is a VRML environment that imitates the insides of a real delivery room. It's filled with different medical equipment, such as delivery table, heater for the newborn, life signs monitor, etc. The most important object in a room is a dynamic model of a newborn. Its health state is described by a set of vital signs and can be controlled interactively or by a user-defined scenario file.

Multiple persons (students of medicine) share this virtual environment. Each is graphically represented by an avatar and is capable of freely moving around the room and interacting with the environment. Every student has to play his designated role in the process of resuscitation. They have to constantly observe the health of the baby by monitoring its vital signs, correctly identify the problem and then carry out the prescribed procedures for helping the newborn. To do this, users must constantly communicate with each other: ask for information, describe what they are planning to do, give instructions, etc. The use of spatial sound is a great help in such training system, because it provides additional positional information.

An example of training session is visible in figure 4. While present in the virtual environment, the student has the complete control over the communication process. He can start/stop the transmission of captured audio to other users, select which users he wants to listen to and change the volume of individual audio streams. Besides that, detailed statistical information about RTP transmissions can be received, such as number of lost packets, number of bytes received, current bit rate, etc [4].

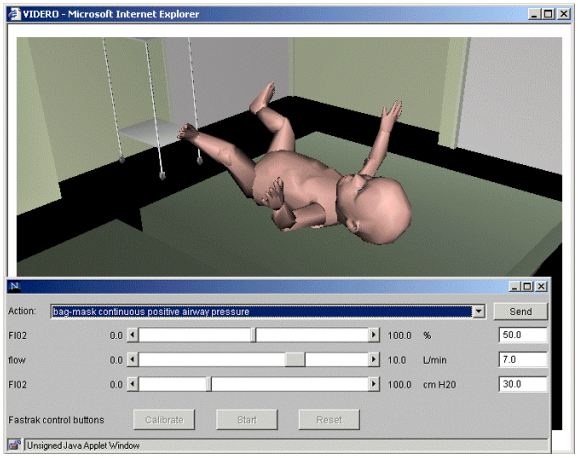

FIG. 4: Student's view of the training session

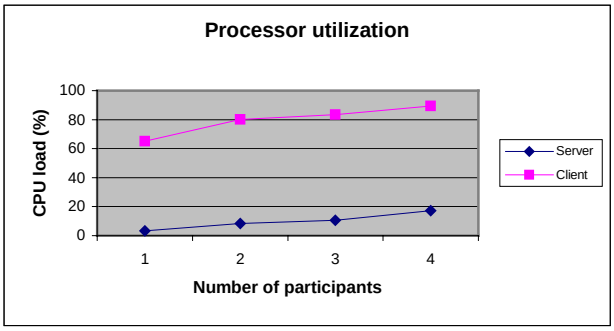

FIG. 5: Client and server CPU utilization

The virtual delivery room's streaming system was tested on a varying number of users. Several performance parameters were measured, such as processor utilization, network utilization, number of lost packets, etc. Some results are depicted in figure 5. The CPU load of the server computer is minimal, but clients are much more occupied. For example, when four users are present in the room, each client's CPU is approximately 90 % loaded. The major factors for such behaviour are presumed to be the computation of spatial sound effects and audio coding/decoding, performed by clients [6]. Audio streams are currently transmitted using the high quality MPEG format, where 44100 samples, each 16 bits in size, must be processed per second for each stereo channel to ensure undelayed delivery of voice data.

7. Conclusion

The paper presents the development of the voice communication module for multi-user virtual environments. The finished spatialized streaming audio system satisfies the requirements in case of our medical training system. Because it is implemented using Java and VRML, it is platform independent and can be easily used for other applications or upgraded with more sophisticated sound processing options.

The performance mesurements suggest that our realization is capable of providing audio conferencing utility for small groups of users only. We belive that with suitable performance optimization this limitation can be greatly reduced. An example of such improvement is the use of silence suppression algorithm to reduce the quantity of audio data that needs to be processed and transmitted through the network. Additional processor time can be saved by replacing the current MPEG audio encoding format with a lower quality encoding, such as GSM, G723 or µLAW [4].

References

[1] R. Benjamin Knapp. *Psychoacoustics of Spatial Hearing*.http://www.engr.sjsu.edu/~knapp/HCIROD3D/3D_psych/3D_psych.htm.

[2] J. Blauert. *Spatial Hearing: The Psychophysics of Human Sound Localization.* MIT Press, London, 1999.

[3] F. Filipanits, Jr. *Design and implementation of an auralization system with a spectrum-based temporal processing optimisation*. Master's Research Project, Miami, 1994.

[4] M. Divjak. *Direct audio communication inside the virtual models.* Graduation Thesis, Slovenia, 2000.

[5] R. Carrey, G. Bell. *The Annotated VRML97 Reference Manual*. Addison Wesley Press, Berkeley, 1997.

[6] M. Divjak, D. Korže. *The use of streaming audio in VRML environments*. 23rd Int. Conf. Information Technology Interfaces ITI 2001, Croatia, 2001.

[7] *Java Media Framework API Specification v2.0.* http://www.javasoft.com/products/java-media/jmf/2.1/specdownload.html, 1999.

[8] Internet Engineering Task Force. *RTP: A Transport Protocol for Real-Time Applications.* Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-avt-rtp-new-07.txt, 2000.

[9] M. Divjak, A. Holobar, I. Prelog. *VIDERO - Virtual Delivery Room*. 9th Electrotechnical and Computer Science Conference ERK 2000, Slovenia, 2000.