Case Study of a Software Structure for a 3D-Audio-Device

Thomas Lehmann
Heinz Nixdorf Institute, Paderborn University
Paderborn, Germany

Abstract

The simulation of an auditory environment is audio signal processing depending on an acoustical model method and a given scene model. For signal processing we use a set of signal processing objects which can be glued to a signal processing network. The object structure depends on the requested acoustic model method in consideration of real-time aspects of an interactive virtual reality (VR). The final processing is executed on a common computer system.

In this paper I would like to briefly show the inner structure of this signal processing tool for 3D-audio and the communication interface to the visual animation tool. I outline the structures and the generation of the signal processing network depending on two acoustical model methods for the simulation of an auditory environment. Furthermore concepts for some improvements of these basic structures are shown considering the limited processing resources in real-time systems.

Key words: 3D sound, object oriented signal processing, interfaces, real-time, interactive virtual reality

1 Introduction

Simulation of a virtual reality means visual and acoustical simulation. The visual presentation is often described by a specific language (e.g. VRML) or built with a front end editor. An acoustical description is not explicitly given in these domains. The visual presentation is processed by libraries (e.g. OpenInvetor) or specialized hardware.

In our group we work on interactive illustrations of complex technical systems like manufacturing systems, robotic systems or software applications. The illustrations are used in the fields of presentations or rapid prototyping. One aim is to achieve a desired result in the shortest possible time-scale. Therefore we use 3D-primitives like cubes, spheres etc. in a 3D visual animation tool. The primitives are controlled by dynamical simulation tools or software agents which form a Animated Agent Layer in this tool. Because the interaction with the user requests real-time behavior. On common platforms the 3D-visual processing leads to less resolution or frame rate. To improve the perception of the given technical scene we added textual output for explanations (like the bubble help system) and 3D sound[5].

The designer of the presentation, for example the user of multimedia authoring tool, is normally not a specialist for acoustics or sound processing. Taking this into account our audio tool has the following aims:

- an easy interface to the controlling application,
- support less acoustical knowledge by the user,
- operational on a common, general purpose hardware platform and
- dynamical reconfiguration of the sound processing system to achieve real-time restrictions.

To operate on common systems all signal processing must be undertaken by software. The design of the software structure should support dynamical reconfiguration and internal scalability. Our approach is the use of signal processing objects in a dynamical changeable graph embedded in a supporting framework.

A physically accurate simulation is not required here. So in combination with the real-time aspect modeling methods with less computing power requirements are preferred. To assist the user of the tool, the acoustic of a 3D-scene should be derived from the visual scene description. Only the acoustical modeling method can be chosen.

In this paper I outline a case study of a 3D sound system and briefly show the inner structure.

2 The 3D-Audio Device Framework

The framework of our 3D-audio device is subdivided into three modules: the master control unit (main module), the scenestorage and the sound sources with the embedded signal processing system. The signal processing system is described later.

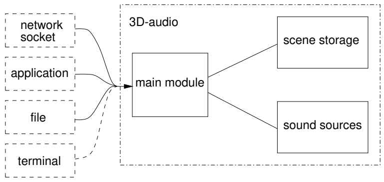

Figure 1: Modules of the device.

The main module contains the application interface and controls the interaction of the other modules. It can be linked with the controlling application by different communication channels, so that the visual animation tool can work on a different computer system than the sound device and the communication is done via network sockets.

All information is exchanged by means of one interface protocol in the form of commands. The commands are of a textual nature, so any character based interface (script file, terminal, serial interface, socket) can be used to transmit the instructions. Even a direct linked application must use this chanel protocol. The interface can be accessed by more than one client. Thus instead of using the camera view of the VR, another computer serves head tracker information using the socket port of the interface.

The commands can be subdivided in four groups: configuration, description of the sound sources, listener description and scene description. Configuration commands select the output sample rate, internal buffer size, etc. The description of the sound sources contains attributes like the position, sound files and the requested modeling method. The listener description is confined to position and direction. The scene description commands describe the actual scene and the changes within the scene at runtime. The scene is modeled with primitives similar to the representation of our visual environment. All primitives are described by shape, position, direction and some object specific parameters. The information is stored in the scene storage that can handle queries about the scene for sound processing.

All commands are parsed and converted to method calls. Notice that there is only one command for auditory purpose: the specification of the requested model method for each sound source.

The third module represents all sound sources in the scene. Each sound source is described by a position[1] and some sound files as sound information. The processing of the sound information is carried out as follows.

3 Signal processing

The signal processing cue is subdivided into two parts: the positioning of virtual sound sources (VSS) in space and the simulation of the auditory environment.

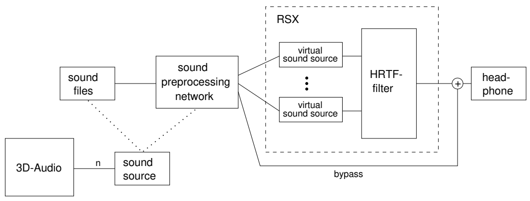

Figure 2: The sound processing system.

The main idea of our signal processing system is the use of signal processing objects like in the SPKit [6]. The objects are glued to a signal processing network. The network is represented by an object graph with signal processing nodes. The topology is based on the acoustical model method and parameters are derived from the scene data, via the query interface of the scene storage. In some cases the topology is derived directly from scene data. In figure 2 the signal processing framework is shown. Each sound source is linked to sound files and controls assigned parts of the sound processing network. The results of the network are positioned in space by HRTF[2]-filter or are directly bypassed to the headphone.

The *sound positioning* of one virtual sound source is done by virtual sound source objects. From their position in space in relation to the (virtual) listeners position the distance and the direction in listener based coordinates is derived. The distance causes a processing of a distance cue, mainly an attenuation by the inverse square law. The direction information is used to select the HRTF-filter. This filter gives the emitted sound a direction.

[1] In this version any sound source is seen as a unidirectional emitting point source

[2] Head-Related Transfer Function

The sum of all processed sound sources is emitted by a headphone. Even bypassed sound for a directionless diffuse sound field is added to this sum signal. Headphones are used in our system to avoid the 'sweet spot' problem and to limit the needed resources to common available systems.

The listeners position can be traced by some kind of head tracker. It is needed in CyberStage environments or together with the use of head mounted displays. In our system we use a normal display to visualize our virtual reality, so the head is softly fixed to the monitor and the camera position is used as the listener's position.

As described above the signal processing structure for the *auditory environment* is based on the acoustical model method. Our aim is to create real-time signal processing so only methods with less detailed physical granularity can be used. In our system we use the image model method and reverberators.

In the *image model method* (IMM) a reflection of sound ray is modeled by establishing a dedicated virtual sound source behind the reflecting surface. The surface information is served by the scene storage. Each VSS can even be seen as the origin of a ray, which is reflected on another surface and forms the reflections of the next order. Thus with more than one surface the number of virtual sound sources is exponentially growing with each order. The recursion can be done until a terminating depth (order) or a termination condition is valid. For example each source (first-order and higher) must be on the same side of a surface as the origin source (zero-order) to have an influence on the auditory environment. By using this model you get a signal processing tree with one VSS at each node. From their position the delay of the sound signal, the distance attenuation and the direction can be derived and used for acoustical placement (s.a.). Filtering effects of the reflecting surface can be modeled by inserting corresponding filters to the signal processing path. It can be shown that, if linear time–invariant filters are used, the number of filters is less growing than the number of VSS. Then the tree of filter elements has a more fractal structure (see [7]).

The derivation of the VSS is independent to the listeners position. Hence, during signal processing a hearability test for each virtual source has to be performed. It is like watching a light bulb in a mirror. The bulb can not be seen in the mirror from all positions. The ray between the viewer and the virtual bulb has to intersect the mirror. Hence it follows that sound reflections from a surface can not be heard at any position[3], and the ray between the VSS and listener must intersect the reflecting surface. Because of this limitation the shape of the reflecting surface is normally not the same as the shape of the reflecting object and has to be stored individually for each VSS. A description of the structures and algorithms of the hearability test is shown in [7]

Spatialization of sound means the simulation of a room impulse response. A room impulse response is a time domain representation of an acoustical environment[3]. The room impulse response can be measured for a given position in a room and simulated with high order finite impulse response filters (FIR).

To reduce the order, structures of comb- and allpass-filters are used to design *reverberators* (see Schröder-reverberator [8] or the Gardner-reverberator [4]). They achieve more statistical properties of the room impulse response. Thus the result of a reverberator sounds spatial but the sound does not include information about the surrounding space as in reality. On the other hand reverberators which are based on this approach need less processing power than the IMM. The filter attributes are often empirically determinated only, so our reverberator objects can choose from sets of parameters depending on the source surrounding range. The output signal of the reverberators is bypassed to the rendering path because reverberation has no direction. The direct sound is still rendered by a VSS.

Both modeling methods shown above can be combined and enhanced in different ways. The main aim is to increase the quality of sound and the perception of the auditory environment in respect to the limited processing power, so that some simple effects can enhance the perception of the given scene.

Obstacles in a natural scene scatter and reflect sound. From the listeners point of view they can be modeled as a transmission attenuation of the sound signal if the obstacles are in the line between source and listener. In the virtual auditory environment the intersection of the direct sound and an obstacle can be found with modified ray tracing algorithms. Depending on the obstacle and surface information a filter is inserted into the signal path.

A heuristic approach to model acoustical coupled rooms is the use of hierarchical structured reverberators. Depending on the position of the sound source, the listeners position and the enclosing space one or more reverberators are connected in serial.

[3] Scattering and refraction can not be modeled with the IMM

4 Implementation

To use signal processing object graphs in a real-time environment we have used some special cues. First all needed objects are created at start time, so that during processing no further allocation of memory and creation of objects is needed. The signal processing graphs are even so established at start up. Reconfiguration of the signal paths is done by switches in the graph and not by topology reconfiguration.

To reduce the number of method calls in the system the sound data is processed in blocks. Even the interactive real-time aspect allows latencies of up to 10ms. Therefore with CD-quality (44.1kHz) 4100 data elements can be processed on block in one time slot.

In our test-application we are using the Intel Realistic Sound Experience (RSX)[2] to do a HRTF-filtering of the virtual sound sources. Other cues like distance attenuation are not used because the RSX implementation bases on the VRML 2.0[1] specification (with attunation $\sim -20_{\mathrm{dB}} \cdot |\vec{r}|$) of 3D-sound rendering and does not reflect the physical characteristic of a natural environment (attunation $\sim -2 \cdot \lg(|\vec{r}|)_{\mathrm{dB}}$).

5 Conclusion

The advantage of our approach is the easy construction of audio signal processing algorithms by structures of processing objects. The 3D-audio tool uses this feature to generate the processing structure depending on a given scene, so the user finishes his work by defining the 3D-world and choosing the desired model method.

The main drawback is the need of large main memory resources for the processing object. Furthermore the visual description of the scene is stored in the visualization tool and in the audio tool. Thus future aims are the reduction of memory requirements.

Still, the computing power of common platforms results in a less quality of sound, but with the integration of DSP-features in general purpose processors (like the 'AltiVec' by Motorola [9]) this approach can provide a solution for future applications.

References

[1] VRML 2.0 specification.
URL: http://vrml.sgi.com/moving-world/spec.Dis/

[2] Intel 3D realistic sound experience, 1997.
URL: http://www.intel.com/ial/rsx/

[3] Durand R. Begault. *3D Sound*. AP Professional, 1 edition, 1994.

[4] William Grant Gardner. *The virtual acoustic room.* Master of science, Massachusetts Institute of Technology, Cambridge, Massachusetts, September 1992.

[5] C. Geiger, G. Lehrenfeld, T. Lehmann, V. Paelke, and C. Reimann. *Design of interactive illustrations using 3d animation and spatial sound.* IASTED International Conference on Computer Graphics and Imaging, Halifax, Canada, June 1998.

[6] Kai Lassfolk *Sound Processing Kit.* The Proceedings of The 1995 International Computer Music Conference 1995 http://www.music.helsinki.fi/research/spkit/documentation/SPKit.html

[7] Thomas Lehmann. *Interaktiver 3D-Sound für Echtzeitanimationen.* Diploma Thesis, Jan. 1998.

[8] M.R.Schröder and B.F. Logan. *"colorless" artificial reverberation. Journal of the Audio Engineering Society*, 9, 1961, pages 192–197.

[9] Andreas Stiller. *Prozessorgeflüster*. c't, 11, 1998