MPEG-4 Structured Audio Systems

Mihir Anandpara
The University of Texas at Austin
anandpar@ece.utexas.edu

Abstract

The MPEG-4 standard has been proposed to provide high quality audio and video content over the Internet. This content is represented in the form of audiovisual objects. However, different parts of the audiovisual scene are encoded separately depending on the nature of the data to be encoded. The standard calls for a structured coding technique that ensures synthesis of high quality audio and clear composition of the separate parts. To enhance the clarity and quality of the signal presented to the user, custom effects are added to the audio signal. One of these effects is a reverberation effect, which produces a decaying response to a signal from the sound source. It is one of the effects which model the acoustic environment. We model an artificial reverberation system and suggest ways to incorporate it into the MPEG-4 standard.

I. Introduction

Streaming audio and video content on the Internet has become increasingly popular. Several standards have been proposed for dealing with streaming audio and video. MPEG-4 is the first standard that addresses presentation content as a set of audio visual objects. The main functionalities in MPEG-4 are content-based coding, universal accessibility, and coding efficiency [2].

Traditional audio coding techniques can be divided into two categories. Lossless encoders remove entropic redundancy from the sound signal. This redundancy exists due to the fact that successive samples of the data are correlated, and some redundancy may be eliminated using this principle. On the other hand, lossy encoders (MP-3, Real Audio) remove perceptual redundancy from a sound signal. These encoding schemes remove those details from the sound signal that cannot be perceived by the human ear using psycho-acoustic principles.

The MPEG-4 standard has been developed for state-of-the-art representation, transmission and decoding of multimedia objects at a low bit-rate. The traditional coding techniques discussed above are not enough to represent audio signals containing a large amount of musical content or sound effects, and still maintain bandwidth efficiency. However, sound signals, especially music signals, represent *structural redundancy*. In a soundtrack, many notes or sound events sound the same or very similar. Also, many soundtracks contain repeating patterns, such as drumbeats. If all parts of the soundtrack can be represented symbolically, a lot of redundant information can be eliminated [4]. This characteristic of soundtracks motivates the use of a symbolic representation of signals through sound-synthesis models, yielding a high compression ratio [6].

In the MPEG-4 standard, different parts of an audio-visual scene can be encoded as seperate components. This separation of components allows each component to be encoded with an appropriate encoding scheme. For example, simple audio content can be encoded using a *General Audio* encoder based on perceptual (or natural) audio coding techniques [12], [11]. Any voice component can be encoded with a speech encoder [5], [1]. Any other audio components with substantial musical content can be encoded as structured audio. At the receiving terminal, these different components of the MPEG-4 transmission stream are decoded separately. The MPEG-4 standard provides for a processing layer, known as AudioBIFS (Audio Binary Information for Scene Description), which takes the uncompressed outputs of these decoders and composes them to form a coherent audio scene. The sound presented to the user after being processed by the AudioBIFS layer should contain any effects for presentation of high quality audio for the user, such as reverberation.

This paper is organized as follows. Section 2 gives a description of the structured audio component in MPEG-4. Section 3 gives an overview of the effects processing and audio composition capabilities of the MPEG-4 standard. In Section 4, we provide a model for a reverberation effect in LabVIEW, and some simulation results. Section 5 gives an overview of the software implementation

of the system.

II. Structured Audio in MPEG-4

The MPEG-4 standard [3] allows for structured audio representations to be encoded and synthesis algorithms specified as a computer program. A new language known as the Structured Audio Orchestra Language (SAOL) has been developed for representation of structured audio and effects in MPEG-4 audio scenes. Any MPEG-4 structured audio scene can be divided into two parts - the *orchestra* and the *score*. SAOL defines an orchestra as a set of *instruments*, where each instrument describes some digital signal processing algorithm that synthesizes or manipulates sound. The structured audio decoder/synthesizer consists of a *scheduler* that is initialized by compiling the SAOL orchestra. The scheduler controls a digital signal processing system that synthesizes the sound based on the algorithm described in SAOL at the audio sampling rate (or a-rate). The scheduler also reads information from the score file at the control rate (or k-rate) and manipulates the sound output accordingly. The output of this decoding process is an uncompressed *primitive media object.*

III. AudioBIFS: Sound Composition and Effects Processing

As described earlier, different parts of an MPEG-4 audio-visual scene are encoded and transmitted separately. The respective decoders decode these parts and output uncompressed primitive media objects. The primitive media objects output by the different decoders are not played directly. Instead, these objects are combined into one coherent audio signal and presented to the user. The processing layer which accomplishes this task is known as Audio Binary Information for Scene Description (AudioBIFS), which is a part of the BIFS (Binary Information for Scene Description) standard defined for composing the entire MPEG-4 scene from different audio and video objects and presenting it to the user. The AudioBIFS system also supports abstract effects post-processing of audio signals and virtual-reality composition. The goal is to provide functionality to present sound based on the listener's acoustic environment and allow custom digital audio effects to enhance the quality of the composed signal.

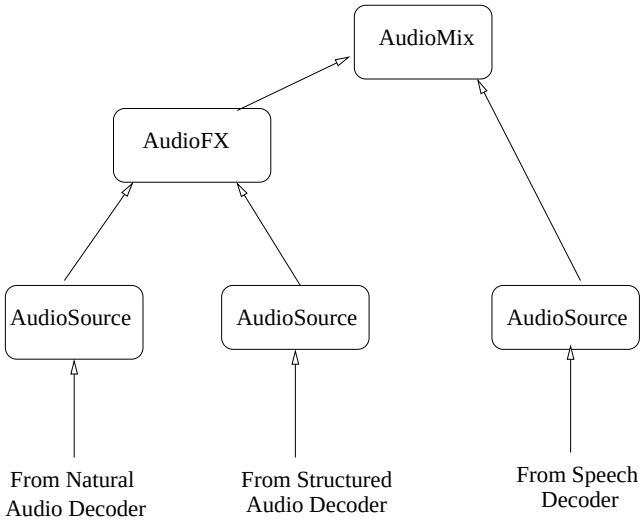

Fig. 1. AudioBIFS scene graph

The AudioBIFS layer uses a *scene graph* structure to organize and compose audio material. A node in the graph represents some operation on the audio signal, while the edges of the graph represent the signal flow. For example, in Figure 1, raw uncompressed data is received from the different audio and speech decoders by the **AudioSource** nodes, which attach the decoders to the AudioBIFS system. Custom digital audio effects are added to the two audio signals in the **AudioFX** node. The different audio streams are finally combined together through the **AudioMix** node and presented to the upper layer for composition with the visual scenes, or sent to the audio output of the system. A detailed description of all the AudioBIFS nodes is presented in [7].

To simulate the listener's acoustic environment, a reverberation effect can be specified to the **AudioFX** node through the SAOL opcode **reverb**. The **AudioFX** node also has functionality to allow the content designer to algorithmically specify any abstract effects in SAOL.

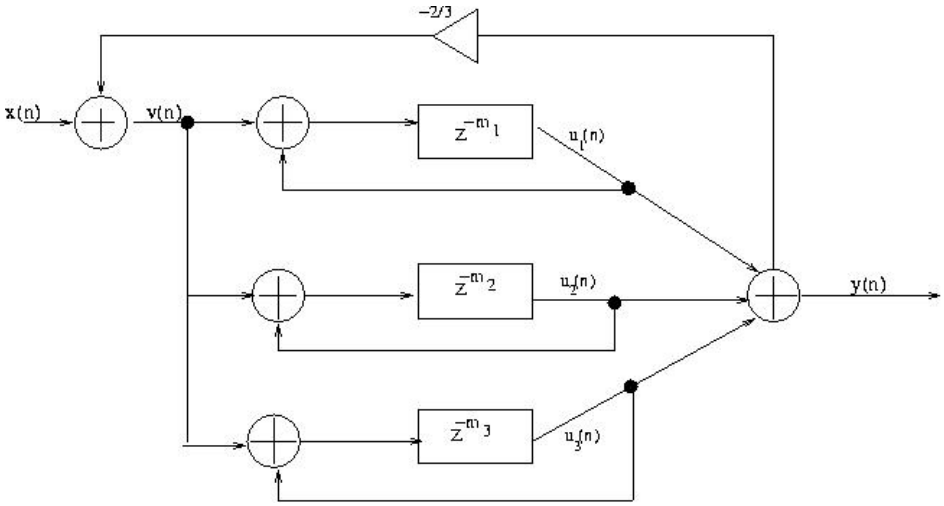

Fig. 2. Jot's Artificial Reverberation System

IV. Reverberation Modeling in LabVIEW

Reverberation results from reflection of sound from other objects. Due to these reflections, the signal received by the listener consists of the reflected components in addition to the original sound. To make any synthesized audio signal sound natural, reverberation must be applied to it, such that it models impulse of the acoustic environment of the listener. This response depends upon several factors, such as, the dimensions of the room, the nature of the walls, and presence of other objects in the room. Several systems for modeling artificial reverberation effects have been studied [8]. We have modeled a reverberation system based on a delay and feedback network. The block diagram of this system is shown in Figure 2. It is based on the work by Jot [10]. It consists of a parallel bank of infinite impulse response (IIR) comb filters, whose output is fed back into the input through a gain block. As shown in Figure 2, the intermediate signals $u_1(n), u_2(n), u_3(n)$ are the IIR comb filter outputs. The difference equation for $u_1(n)$ is given as:

$$u_1(n) = v(n) + u_1(n - m_1) \tag{1}$$

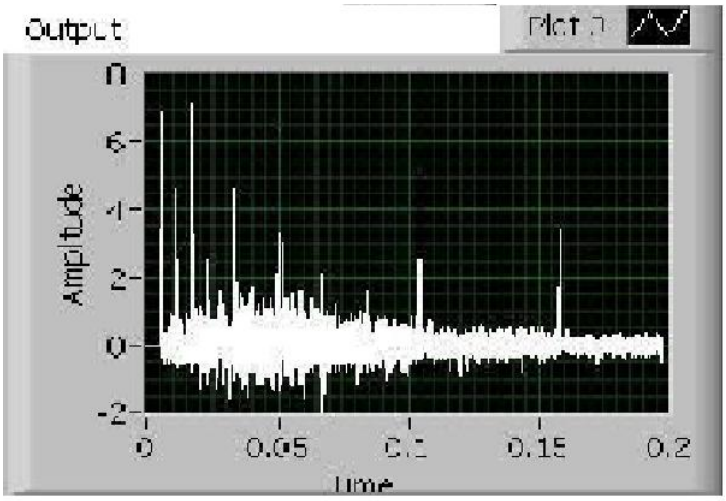

Fig. 3. Impulse Response of the reverberation system

The impulse response of this system will give a measure of the decay of the reverberated signal with respect to the original signal, and provide details on the delay and reverberation quality. We model this system in LabVIEW. An impulse input is modeled as a very narrow triangle wave. This impulse is added to a stationary white Gaussian noise process and fed into the reverberation filter. The feedback and delay in the filter is modeled using a circular buffer in LabVIEW, where the buffer values are rotated once during the execution of all the blocks. This is done by adding a delay of one token on the feedback arcs. The impulse response of this filter is shown in Figure 3. According to this response, we can see that there are some distinct high amplitude echoes in the time just after the impulse. The response then settles down and decays towards zero. One cause of concern are the periodic spikes in the signal. This may be occurring due to limit cycles in the IIR filter response. These periodic spikes could be removed by adding a low pass filter at the output of the z^{-m_i} block to bring the filter poles away from the edge of the unit circle, and reduce the pole Q (Quality Factor).

The reverberation time, also known as RT60, is defined as the time taken for the signal amplitude to decay to -60dB below the original sound signal amplitude. In general, the reverberation time of the artificial reverberation system depends upon the values of m_1, m_2 and m_3 and the gain of the filter.

The **reverb** opcode for SAOL provides flexibility to specify a frequency dependent reverberation time response, that is, the user can specify RT60 values for different frequency components in the input signal. In order to modify the response of the reverberation filter, we need to add absorbent filters, $h_i(z)$ after each z^{-m_i} block [10]. The effect of this operation is to bring the poles of the comb filters closer to the origin of the unit circle, thereby, dampening the response of the filter and causing it to decay faster. The amount of pole displacement is determined by the desired reverberation time response. A first order low pass filter design for an absorbent filter is given as [9], [10]:

$$h_i(z) = g_i \frac{1 - a_i}{1 - a_i z^{-1}} \tag{2}$$

g_i gives the desired reverberation time at dc, and a_i gives the desired reverberation time at high frequencies. They are given by:

$$g_i = 10^{-3m_i T/Tr(0)} \tag{3}$$

$$a_i = \frac{ln(10)}{4} log_{10}(g_i)(1 - \frac{1}{\alpha^2}) \tag{4}$$

where,

$$\alpha = \frac{Tr(0)}{Tr(\pi/T)}$$

$$T = sampleperiod$$

The impulse response of the damped reverberation filter is shown in Figure 4. We can see here that the response decays faster towards zero, and does not show any periodic spikes that were seen in the lossless filter.

The MPEG-4 AudioBIFS system is that it does not have embedded functionality to add the effects of air absorption and Doppler effects due to relative motion between the source and the listener. Hence, in the second version of the standard, three nodes were added to the existing AudioBIFS node set. One of those nodes, the **AcousticScene** node has fields which specify any artificial reverberation

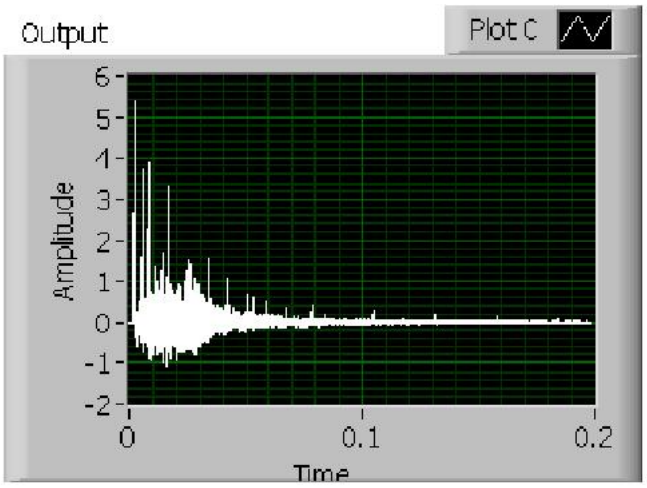

Fig. 4. Impulse Response of the damped reverberation system

effects based on the topology of the listener's environment. The delays $m_i's$ in the comb filters and the absorbent filter characteristics can be appropriately modified to suit the parameters defined in the **AcousticScene** node. This results in a better quality sound, since the reverberation and spatialization effects are based exactly on the listener's acoustic environment.

V. Software Implementation

As a part of this project, we have developed a system in C++ to study the working of some of the nodes in the AudioBIFS scene graph, such as the **AudioSwitch**, **AudioMix** and **AudioDelay** nodes. The **AudioFX** node also has a SAOL execution engine, which executes all the instruments specified in the orchestra at the a-rate, reads control parameters from the score at the k-rate, and modifies the behavior of the system accordingly. The implementation of the **reverb** algorithm modeled above can also be included in this system, such that the function is called whenever the reverb opcode is called in the SAOL orchestra.

VI. Conclusion and Future Work

In this project, we studied the nature and characteristics of the structured audio coding component of the MPEG-4 standard. Structured audio represents sound synthesis and processing algorithms as

a computer program written in a special modeling language known as SAOL. We also studied and implemented the MPEG-4 standard for audio composition, known as AudioBIFS.

One of the major considerations involved in presentation of synthetic sound is to simulate the listener's environment, and add effects to the sound signal such that it sounds natural. AudioBIFS has the capability to process the sound with any effects specified in SAOL. One of the effects generally added to a signal is reverberation, which is the phenomenon of multiple echoes reaching the listener after reflection with other objects in the surrounding environment. We modeled a digital reverberator system in LabVIEW and studied some enhancements to the system to meet the acoustic requirements of the system.

Models similar to the reverberation model can be constructed and incorporated into the software implementation of the AudioBIFS system. After that, hardware implementation details can be specified on high-performance digital signal processors or multimedia processors, that can implement filtering operations efficiently.

REFERENCES

[1] A.Gersho, "Advances in speech and audio compression," *Proceedings of the IEEE*, vol. 82, pp. 900–918, 1994.

[2] A.Puri and A.Eleftheriadis, "MPEG-4: An object-based multimedia coding standard supporting mobile applications." [Online]. Available: citeseer.nj.nec.com/puri03mpeg.html

[3] I. Y. M. B.Grill, B.Edler and E.Scheirer, "ISO/IEC JTC1/SC29/WG11 (MPEG) document N2203," in *ISO/IEC* $11496 - 3$ *(MPEG-*4 *Audio) Final Committee Draft*, 1998.

[4] W. B.L.Vercoe and E.D.Scheirer, "Structured audio: Creation, transmission and rendering of parametric sound representations," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 922–940, 1998.

[5] B.S.Atal and M.R.Schroeder, "Predictive coding of speech signals and subjective error criteria," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-27, pp. 247–254, 1979.

[6] E.D.Scheirer, "Structured audio, kolmogorov complexity, and generalized audio coding," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 8, pp. 914–931, 2001.

[7] J. E.D.Scheirer, R.Vaananen, "AudioBIFS: Describing audio scenes with the MPEG-4 multimedia standard," *IEEE Transactions on Multimedia*, vol. 1, no. 3, pp. 237–250, 1999.

[8] J.A.Moorer, "Signal processing aspects of computer music : A survey," *Proceedings of the IEEE*, vol. 65, pp. 1108–1137, 1977.

[9] J.M.Jot, "Digital delay networks for designing artificial reverberators," *Proceedings of the* 90^{th} *AES Convention*, 1991.

[10] ——, "An analysis/synthesis approach to real-time artificial reverberation," *Proceedings of the IEEE Int. Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 221–224, 1992.

[11] N. N.Jayant and R.Safranek, "Signal compression based on models of human perception," *Proceedings of the IEEE*, vol. 81, pp. 1385–1422, 1993.

[12] S.R.Quackenbush, "Coding of natural audio in MPEG-4," *Proceedings of IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3797–3800, 1997.