HIERARCHICAL LOSSLESS AUDIO CODING IN TERMS OF SAMPLING RATE AND AMPLITUDE RESOLUTION

T. Moriya, A. Jin, T. Mori, K. Ikeda and T. Kaneko

NTT Cyber Space Laboratories, NTT Corporation
Musashino, 180-8585 Japan t.moriya@ieee.org

ABSTRACT

This paper proposes a lossless audio coding scheme with hierarchical scalability in terms of sampling rate and amplitude resolution. A single bit stream contains hierarchical information that can generate waveforms ranging from 96 kHz with 24-bit amplitude resolution through lower sampling/resolution lossless waveforms to a highly compressed lossy one created using an MPEG-4 Audio coder. This bit stream structure enables dynamic rate control and hierarchical multicasting based on a simple priority control of the IP packets. These functions will be useful for high-quality archiving and broadband streaming for various types of networks and terminal equipment.

1. INTRODUCTION

Network operating companies and service providers are paying more attention to broadband infrastructures, including xDSL, fiber optics, and broadband wireless access. Bit rates for these channels have become close to those for raw audio signals, (i.e., more than 1 Mbit/s). For these channels it is becoming realistic to deliver high sampling-rate, high amplitude resolution (e.g., 96 kHz, 24 bits/sample) lossless waveforms in real time. For these demands, archiving systems must handle various types of signals with different sampling rates and amplitude resolutions.

At the same time there are still various application areas suitable for high-compression coders such as the MPEG-4 formats. It is therefore important for users and network/service providers to have interoperable universal solutions that bridge between current or legacy channels and the rapidly emerging broadband channels. In addition, even though broadband channels are widely available, there are still some occasions when packet loss occurs due to accidents or resource sharing requirements. With the current broadband waveform representations, such as PCM or other lossless coding formats, serious distortion may result if packet loss occurs in broadband streaming. It is preferable to receive a continuous signal even in the case of packet loss or unpredictable restrictions on channel capacity. A bit rate scalable coding scheme can provide hierarchical bit streams whose bit rates can be dynamically controlled during transmission. This feature is useful for efficient multi-cast transmission to avoid overloading servers due to an excessive number of demands from client sites. It is also useful for preserving quality depending on the network resources if we can control the priorities of packets.

Note that current streaming systems use one-to-one IP connections with pre-defined bit rates between each server and client. This scheme is feasible only when each bit rate is very low and the number of clients is small. For broadband streaming to a large number of clients, bit rate scalable coding schemes are essential.

In the following sections, we describe a sampling rate scalable scheme with a high-compression coding core based on MPEG-4, and show its performance.

2. CODING SCHEME

2.1. Hierarchical framework

If original waveform files are 96-kHz sampled with 24-bit resolution, we might need 48-kHz sampled lossless coded files with 16-bit amplitude resolution for some users, and highly compressed (lossy) coding by MPEG-4 for others. A block diagram of the hierarchical encoding scheme with scalable sampling-rate and amplitude resolution is shown in Fig. 1. The associated decoding process is shown in Fig. 2. We assume the following four levels of quality:

- ??96-kHz samples with 24-bit resolution,
- ??96-kHz samples with 16-bit resolution,
- ??48-kHz samples with 16-bit resolution,
- ??48 kHz samples compressed by MPEG-4 coding.

For an archiving system, we might prepare four files with these different waveform formats, where the first three are compressed by lossless coding. On the other hand, in a sampling rate scalable system with a high-compression core, all the information for generating them is hierarchically packed into a single file.

The encoding process has three steps and generates four layers of bit streams. We start with the highest quality

format at 96 kHz and 24 bits. The eight least significant bits (LSBs) are separated and transmitted as the highest layer bit stream. We do not apply any compression scheme to these bits, because we cannot expect any compression gain for these full-amplitude structure-less bits.

In the second step, the 96-kHz signal is down-sampled to 48 kHz and the time-domain error signal between the input and the up-sampled signal is compressed. This will form the second-layer bit stream. The compression ratio is very large for this case, because the error signal has a low amplitude and only high-frequency components as shown in Fig. 3.

In the third step, high-compression lossy coding is applied to the 48-kHz 16-bit signal. We use MPEG-4 TwinVQ tools at 80 kbit/s. The error signal between the 48-kHz signal and the reconstructed MPEG-4 signal is compressed. The power spectrum of the error signal is shown in Fig. 4. The lossy coder is normally designed to minimize perceptual distortion and the error signal has a non-flat spectrum. This is the third-layer bit stream. The fourth and lowest layer (core layer) is the MPEG-4 compressed bit stream.

The decoding process is the inverse process to the encoding one. If we have the bitstreams of all layers, then a 96-kHz 24-bit signal can be reconstructed. Similarly, we reconstruct a 96-kHz 16-bit signal from the lowest three layers, and a 48-kHz 16-bit signal from the lowest two layers.

2.2 Basic lossless coding

The hierarchical compression system described in the previous section uses the basic lossless coding shown in Fig. 5, which consists of frame-based simple operations or conversions such as linear prediction, format conversion to sign-magnitude representation, bit slicing, and

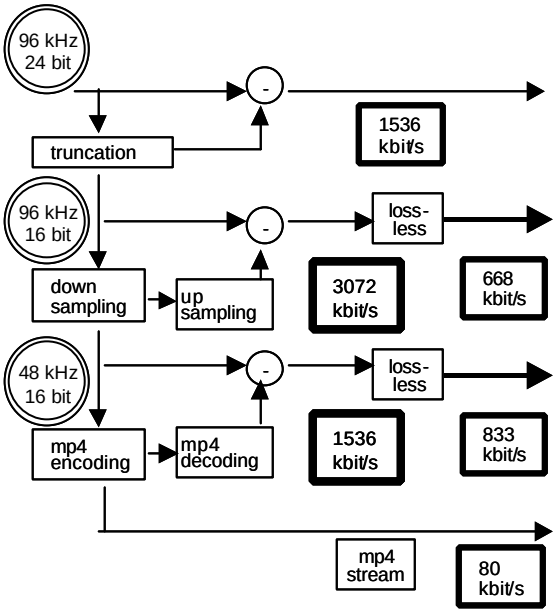

Fig. 1 Hierarchical encoder

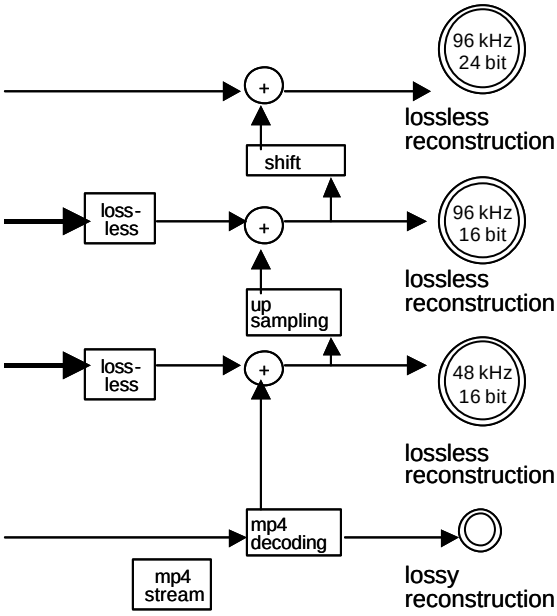

Fig. 2 Hierarchical decoder

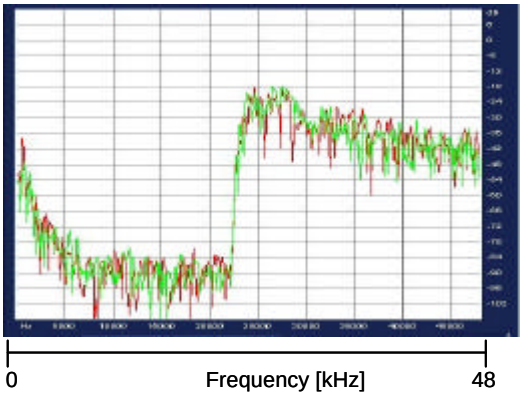

Fig. 3 Spectrum of the difference between 96 kHz original and the signal up-sampled from 48 kHz

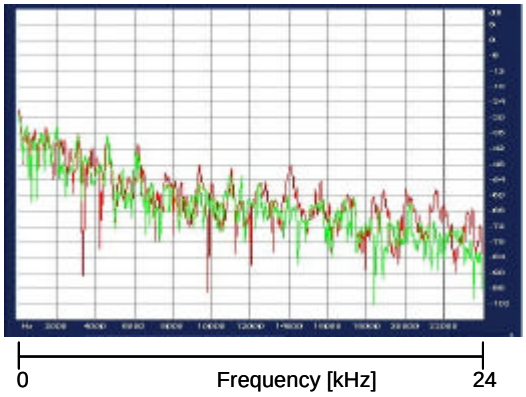

Fig. 4 Spectrum of the difference between 48 kHz original and the signal reconstructed by MPEG-4

compression. The linear prediction is based on frame all-pole type prediction, which is similar to that popular in speech coding. Therefore, a few predictive coefficients must be quantized and transmitted. The difference is that the predicted values are always rounded up to an integer to keep the accuracy. So far, we have not quantized the predictive coefficients. But the bit rate for this side information is estimated to be 5 kbit/s for a 48-kHz sampled stereo signal, which is negligible compared with other data rates.

This prediction can remove redundancy between samples and reduce the amplitude of the error signal. Time domain error signals may have some correlation between samples. Equivalently the power spectra of the error signals are not always flat. One example is that the error signal between the original signal and a band-limited signal has only rich components in the high-frequency parts as shown in Fig. 3. Another example is that an error signal between the original signal and the signal reconstructed from a perceptually optimized high-compression coder has a similar power spectrum to the original signal as shown in Fig. 4.

The second step of the basic lossless coding consists of format conversion and time domain bit slicing. If the amplitude of the time domain signal is reduced, we expect many "0" values among the most significant bits (MSBs) if we use the sign-magnitude format instead of 2's complementary format. We can make use of this property for easy compression if we use bit slicing in the time domain. In this experimental study, we used "gzip" to compress the bit-sliced signals in the final stage.

2.3 Time domain bit slicing

Time domain bit slicing is also useful for near-lossless coding and quality control, which assure the minimum number of amplitude resolution bits if we allocate separate IP packets for each bit-stream layer, as shown in Fig. 6. The left block of the figure represents the time-domain PCM signals, where a white square denotes "0" and a shaded square denotes "1". If we assign top priority to the MPEG-4 high-compression streams, we can reproduce a perceptually acceptable signal at very low bit rates although the reconstructed waveform is different from the original one in terms of SNR. The error signals also have a simple priority structure.

It is easy to achieve near-lossless quality (only some LSBs differ from the original) and graceful degradation. These features can be efficiently used for QoS control. Note that these techniques are also applicable to a real-time bi-directional communication system where a large buffer for packet jitter is not allowed due to delay constraints.

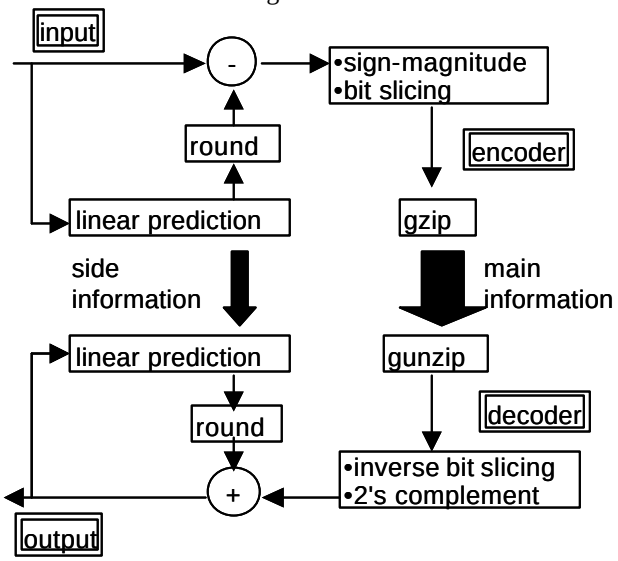

Fig. 5 Basic lossless coding module.

2.4 Performance evaluation

Compression performances are compared in Table 1. Each bit rate value is the average for two short audio files (6 s each). The second column from the left shows the differential bit rate of the proposed scalable scheme. The third column shows the accumulated bit rates necessary to reconstruct the original waveform. The fourth column shows the bit rates compressed by a stand-alone scheme for each condition of sampling rates and amplitude resolution. In this case, we used the same basic lossless coding described in the previous section. The right-most column shows the bit rates of the original linear PCM formats.

Comparing our proposed scheme and the stand-alone compression for each sampling rate condition (e.g., 3rd and

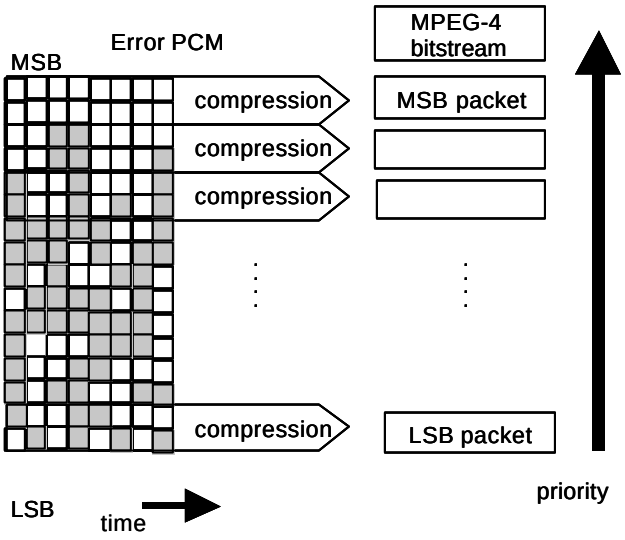

Fig. 6 Horizontal bit slicing.

4th columns: 3117 vs. 2868 kbit/s), we see that the stand-alone scheme is slightly better (3–11%) than the scalable one. However, if we need to keep all these sampling rates conditions, the standalone scheme needs 5177 kbit/s, whereas the scalable scheme needs only 3117 kbit/s, since the highest layer bitstream contains all of the bitstreams. Comparing the proposed scheme with the non-compressed scheme, we see that total disk capacity can be reduced to around 1/3. Comparing the scalable system with the stand-alone compression system, the total disk space can be reduced by around 40%.

Table 1. Comparison of compression schemes.

	scalable data rate (kbit/s)	accumulated data rate (kbit/s)	stand-alone compression (kbit/s)	original data rate (kbit/s)
96 kHz 24 bit	1536	3117	2868	4560
96 kHz 16 bit	668	1581	1420	3072
48 kHz 16 bit	833	913	881	1536
lossy coding	80	80	80	80
Total	3117	3117	5177	9248

3. APPLICATION

Let us consider the merit of the proposed coding scheme when it is used in a server-client music streaming system that can support various sampling rates and amplitude resolutions. System (1) in Fig. 7 assumes the proposed scalable coding at the server side. System (2) assumes that all necessary formats have been prepared beforehand. System (3) assumes that only the highest quality format will be stored and it has a transcoder.

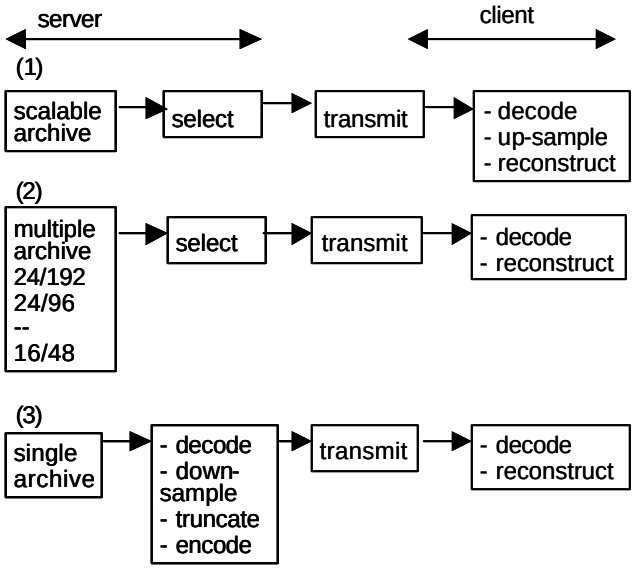

Fig.7 Comparison of server-client systems.

According to observations in the previous section, system (3) has the smallest disk space and system (1) needs slightly more disk space. System (2) consumes several times as much disk space. If we compare the computational complexity, system (2) needs the smallest load. System (1) also needs only a small load at the encoder, but needs a small amount of additional processing for upsampling at the decoder. System (3) needs a significantly large amount of computation for decoding, processing, and re-encoding to support various demands. This observation shows that the proposed scalable system (1) is most attractive among the trade-offs in terms of disk space and computational complexity. Additionally, the scalable bitstreams used in system (1) can be flexibly utilized in various network applications.

4. CONCLUSION

In our lossless audio coding scheme with hierarchical sampling rate and amplitude resolution, a single scalable bit stream contains hierarchical information that can generate a 96-kHz waveform with 24-bit amplitude resolution and lower-sampling-rate lossless waveforms including highly compressed ones. For basic lossless compression, it combines prediction and time domain bit slicing.

Without using a complex transcoder, this scheme can significantly reduce the sizes of archived audio files that support several sampling rates compared with using an independent compression coder for each format. This bit stream structure is also useful for dynamic rate control and hierarchical multicasting based on a simple priority control of IP packets.

Lossless audio coding technologies will be standardized in the MPEG-4 Audio group. The scheme proposed here is a promising candidate for the standard.

REFERENCES

[1] T. Moriya, N. Iwakami, T. Mori, and A. Jin, "A Design of Lossy and Lossless Scalable Audio Coder," *Proc. ICASSP'2000*, pp. AE-P1.11, 2000.

[2] T. Moriya, A. Jin, T. Mori, K. Ikeda, and T. Kaneko, "Lossless scalable audio coder and quality enhancement," *Proc. ICASSP'2002*, #2440, 2002.

[3] M. Hans and R.W. Schafer, "Lossless Compression of Digital Audio," *IEEE Signal processing magazine*. Vol. 18 No. 4, pp. 21–32, 2001.