A 6KBPS TO 85KBPS SCALABLE AUDIO CODER

Tony S. Verma *

Lab. of Acoustics and Audio Signal Processing
Helsinki University of Technology, Finland

Teresa H.Y. Meng

Department of Electrical Engineering
Stanford University, USA

ABSTRACT

Scalable audio coding is important in network environments, such as the Internet, where bandwidth is not guaranteed, packet loss is common, and client connection data rates are heterogeneous. Signal models provide a general frame work for attacking a wide range of challenges in the unicast delivery of real-time audio over packet switched networks. The specific signal model in this work generates a parametric representation for general wide-band audio signals. The model consists of three complementary components: sines, transients, and noise. Because the human hearing system ultimately judges the validity of a model for audio signals, psychoacoustic principles are explicitly considered in the three part model. Once analyzed, the parameters are quantized, compressed and packed into a single 85Kbps bit-stream. From this bit-stream, bit-streams at several bit-rates between 6Kbps and 85Kbps may be readily extracted. The audio coder offers a wide range of scalability while the audio quality of the coding scheme gracefully degrades from perceptually lossless to low-quality.

1. INTRODUCTION

In recent years model based approaches to audio compression using analysis/synthesis techniques have seen increasing interest. In large part due to the need for efficient transmission of general audio content over lossy packet networks like the Internet, modeling approaches have gained popularity. The Internet is a network environment where bandwidth is not guaranteed, packet loss is common, and the data rates at which clients may connect varies dramatically. This indicates a scalable compression scheme that exhibits packet loss robustness is desirable. This ensure clients will receive the best quality audio possible given their connection rate and current network conditions. Unlike transform based approaches, which impose a rigid structure to the audio signal, model approaches by their nature describe the underlying audio signal in a flexible manner. It is this flexibility which makes model based approaches amenable to both scalability and packet loss robustness. An excellent review of recent developments in model based audio compression techniques can be found in [1].

The scalable compression scheme described in this paper is similar in certain respects to existing work. There are a number of audio compression schemes that use sinusoidal modeling, e.g., [2, 3, 4]. The coder presented here, however, uses different modeling, quantization, compression and bit-packing techniques. The coder also offers a much larger degree of scalability than previous work. Using a three part complementary signal model consisting of sines, transients and noise, the coder produces an 85Kbps compressed bit-stream. Embedded within this bit-stream are lower bit-rate streams. These streams are easily extracted allowing scalable transmission of audio. The various bit-streams gracefully degrade in fidelity from perceptually lossless quality (at 85Kbps) to very-low audio quality (at 6Kbps).

The first two sections of the paper give an overview of the audio encoder and decoder. These sections also briefly review the three part signal model. Section 4 describes the quantization, compression and bit-packing for the model parameters. The section describes how scalability and packet loss robustness are achieved. The final section of the paper concludes by assessing the quality of the scalable audio coder.

2. ENCODER OVERVIEW

Figure 1 shows a block diagram of the scalable audio encoder. The encoder models the input audio signal (mono, bandlimited to 16KHz) in terms of sines, transients and noise. The parts of the model complement each other. Because each model assumes an underlying structure to the signal, modeling errors are inevitable. The three parts of the model work together, however, to alleviate model mismatch problems. The three part model operates in series. Each model captures signal components that are coherent to its underlying assumptions. At the same time, each model compensates for the previous models short-comings.

First transient signal components are modeled using the transient model introduced in [5]. This transient model is the frequency domain dual to the sinusoidal model. It provides a flexible low order model for transient signals. The theme behind the transient model is that sinusoidal modeling works well on quasi-stationary signals. Unfortunately transients violate this assumption. However, by duality, a signal that is transient in the time-domain is quasi-stationary in the frequency domain. It is therefore possible to model transient signals using a sinusoidal model provided the signal is first transformed into a well chosen frequency domain. The Discrete Cosine Transform (DCT) provides an appropriate mapping. During analysis, the transient model takes the DCT of non-overlapping frames of the input signal. Each of these frames is about 1s in duration. Within each 1s DCT frame, sinusoidal modeling extracts transient signal parameters. The DCT domain sinusoidal model uses windows of length 1024 points overlapped and hopped by $1/2$ the window length. Once extracted, the encoder passes the transient parameters to the quantization block for compression; the compression of these parameters is described in Section 4.

Additionally, the encoder synthesizes the time domain transients from the extracted parameters in order to create a time-domain residual. Employing a 'fast' transient reconstruction tech-

*A majority of this work was completed while the author was a graduate student researcher at Stanford University. In addition, this work been supported by the Academy of Finland.

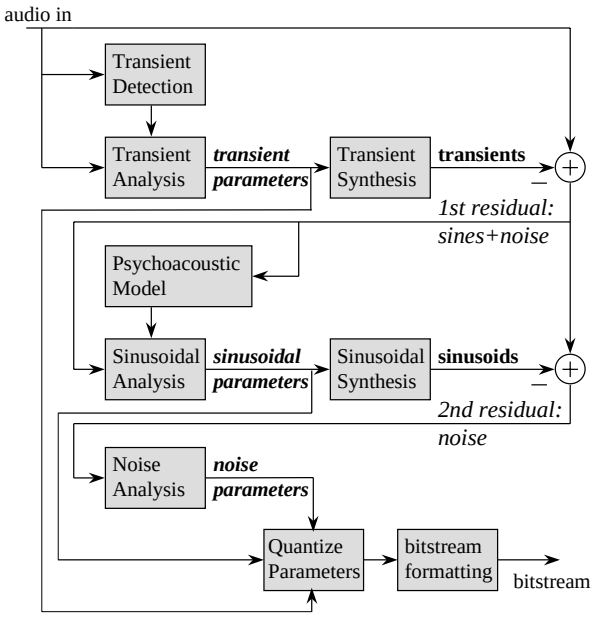

Figure 1: Scalable audio encoder block diagram.

Bit Rate	Parameters
80Kbps	Sines quality layers 1-5, transients, noise
45Kbps	Sines quality layers 1-4, transients, noise
32Kbps	Sines quality layers 1-3, transients, noise
20Kbps	Sines* quality layers 1-3, transients, noise
16Kbps	Sines* quality layers 1-2, transients, noise
12Kbps	Sines* quality layers 1-2, noise
10Kbps	Sines* quality layers 1-2
6Kbps	Sines* quality layer 1

Table 1: Large step scalable bit-rates for the scalable coder. Transient parameters require 4Kbps, while noise parameters require 2Kbps. Sines* indicates that phase information for the sinusoids is not transmitted at these bit-rates.

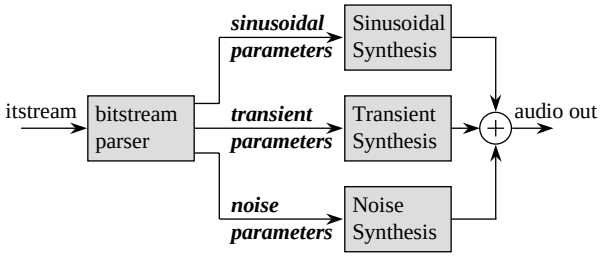

Figure 2: Scalable audio decoder block diagram.

nique described in [6] avoids the two steps of synthesizing DCT-domain sinusoids and applying an inverse DCT. This residual, consisting mainly of sinusoids and noise, is passed to the sinusoidal model.

The encoder next models sinusoidal components of the signal. The sinusoidal model used in this block is a perceptually weighted sinusoidal model described in [7]. While many different techniques for sinusoidal modeling exist, the goal of each technique is similar: for the l-th frame of the input, find a set of K sinusoids to represent the quasi-stationary portions of the signal. The scalable audio coder uses a sinusoidal model that explicitly takes into account the human hearing system by using a psychoacoustically weighted matching pursuit algorithm. This analysis-by-synthesis approach iteratively extracts sinusoidal components according to the perceptually important signal-to-mask ratio. The analysis yields the parameter set $\left\{a_k^l, f_k^l, \phi_k^l\right\}$, where a_k^l, f_k^l, and ϕ_k^l are respectively the magnitude, frequency and phase of the k-th peak in the l-th analysis frame. The frame-size for the sinusoidal model is set at a constant 64ms. The hop-size and frame-overlap are $1/2$ the analysis frame-size. Because transients in the signal have been modeled and removed, pre-echo problems are greatly diminished justifying the relatively long frame-size. The bandwidth of the sinusoidal model is restricted to 8KHz. This is because, for a wide range of audio signals, signal components above 8KHz tend to be noise-like [8, 4], and are better left to the noise model.

The encoder passes the extracted sinusoidal parameters to the quantization block for compression. At the same time, the encoder synthesizes the sinusoidal components to create another residual. This second residual consists mainly of noise-like components.

The final modeling stage is for noise-like signal components. The encoder uses the equivalent rectangular bandwidth (ERB) noise model introduced in [9]. The model is based on the observation that for noise-like signals, energy in the ERBs describes the underlying signal with perceptual accuracy. The noise model models the entire audio signal bandwidth of the noise-like residual. From 20Hz to 16KHz 15 ERBs are used. The noise model is also a frame-based algorithm. The frame size is 32ms and the hop-size and frame-overlap is $1/2$ frame. Once again, the encoder passes the noise model parameters to be quantized and compressed.

The quantizer and bit-packing blocks compress the model parameters then finally place compressed parameters into a single embedded bit-stream consisting of many quality layers. These blocks also control scalability. Table 1 gives the large-step scalable bit-rates that the coder achieves. These are also the quality layers embedded in the bit-stream. Fine-grain scalability, bit-rates between the ones given in the Table, are achieved using a scalable entropy code. This scalable entropy is described in [6] and is applied to the quantized sinusoidal data. The computational complexity of the encoder is high. This is justified, however, because each piece of audio is encoded once off-line; therefore, it does not affect streaming performance. Details of the quantization and scalability are given in Section 4.

3. DECODER OVERVIEW

Figure 2 shows the decoder block diagram. The decoder, because it uses relatively few operations, meets the requirements of low complexity. The first step in decoding is bit-unpacking. The bitstream is composed in a manner where each quality layer is fully decodable on its own without information from other layers. This allows scalability and packet loss robustness. The decoder parses each packet converting the information back into model parameters. Each synthesis engine, i.e., sinusoidal, transient or noise, receives their respective model parameters. The synthesis blocks then reconstruct a version of the original audio signal.

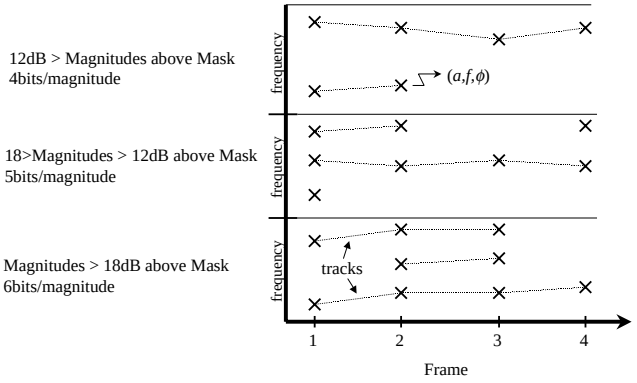

Figure 3: Sinusoidal tracks and quality groupings based on bit assignment.

4. QUANTIZATION, COMPRESSION AND BIT-PACKING

Once extracted, the sine, transient, and noise model parameters are quantized independently of one another. The resulting quantized parameter sets are placed in different parts of the layered bit-stream. For the highest bit-rates, the server transmits model parameters corresponding to each of the three models. At the lowest bit-rate, the server only transmits sinusoidal parameters.

4.1. Compression of sinusoidal parameters

The sinusoidal parameters, $\left\{a_k^l, f_k^l, \phi_k^l\right\}$, are the perceptually most important part of the three part model; consequently, the sinusoidal parameters require a far greater number of bits than the transient or noise parameters. In addition, the manner in which the sine parameters are quantized and placed in the bit-stream allows these parameters to have the greatest degree scalability. The steps for sinusoidal model quantization and bit-packing include parsing parameters into perceptually relevant groups, magnitude bit assignment and quantization, frequency quantization, phase parameter quantization, frame-to-frame parameter tracking, scalable entropy code application, and finally bit-stream placement.

Each frame of the parameter set contains K sinusoidal peaks, each consisting of a magnitude, frequency and phase. The sinusoidal analysis algorithm ordered these peaks according to perceptual significance. This ordering plays a crucial in the large-step scalability of the sinusoidal parameters. First, peaks are placed together in groups of perceptual importance using the signal-to-mask ratio information from the psychoacoustic model. Figure 3 demonstrates how peaks are grouped. Peaks with signal-to-mask ratios within certain ranges are grouped together because they require the same number of bits, on a logarithmic scale, to describe in a perceptually lossless fashion. For example, all magnitudes that have a signal-to-mask ratio between 12dB and 18dB require 5 bits on a logarithmic scale. Peaks with larger signal-to-mask ratios require a greater number of bits to prevent quantization noise from creeping above the masking threshold; moreover, these peaks are perceptually more important than peaks with smaller signal-to-mask ratios. Once grouped and magnitudes quantized, the peaks in each group are re-ordered from lowest frequency to highest frequency. These operations are done frame-by-frame.

Frequency parameters are quantized to the just noticeable difference frequency (JNDF) scale. A just noticeable difference is the amount something can vary, in this case the frequency of a sinusoid, before a typical human can detect the change. Quantizing the frequency parameters to 10bits on the JNDF scale, most listeners will not be able to distinguish the original from the quantized frequencies.

Phase parameters are uniformly quantized on the unit circle using 6 bits. Because the ear is relatively insensitive to phase, phase is only transmitted for the highest quality bit-rates as indicated in Table 1.

The data-rate of the magnitude and frequency information is still quite high. Exploiting interframe dependencies greatly reduces the data rate. Within each quality group, the idea of tracking, commonly used in many sinusoidal modeling schemes, is exploited. Figure 3 depicts the highly correlated frequency tracks. What is not shown is the corresponding magnitude tracks which also show a high degree of correlation. These sinusoidal parameter tracks are allowed to have from two to five sinusoidal peaks. The tracks are restricted to have fewer than five peaks for robustness to packet loss. To each track, differential pulse code modulation (DPCM) is separately applied to the frequency and magnitude parameters. Absolute frequency and magnitude values are only kept for the first peak of each track. Therefore each track is described by the following information: an absolute frequency value, between one and four differential frequencies (which depends on the length of the track), an absolute magnitude value, between one and four differential magnitudes, and between two and five absolute phase values. To further reduce the bit rate, while allowing fine-grain scalability, the scalable entropy code described in [6] is applied to the differential frequencies and differential magnitudes. Unfortunately the phase parameters are statistically uniformly distributed around the unit circle; accordingly, DPCM and entropy coding will not reduce their bit-rate.

4.2. Bit-packing and scalability of sinusoidal parameters

Scalability and packet loss robustness depend heavily on how bits are placed into packets. Within the bit-stream, the perceptual groups of peaks (constituting a quality layer) are packed independently of one another. Each perceptual group in this layering scheme contributes to the overall quality of the reconstructed sinusoidal signal. As such, the server essentially has a quality control knob, which it adjusts according to network and/or client conditions. When client or channel conditions dictate the current bit-rate is too high, the entire set of peaks in the least important perceptual group is dropped until an allowable bit-rate is met. The allowable bit-rates are shown in Table 1.

Through informal listening tests, it was found that loss of a few partials over a short period of time is far less devastating than loss of entire time-slices of sinusoidal data. This is true even when interpolation is used to make up for lost time-slices. This idea is used in the bit-packing scheme for the sinusoidal data. Sets of tracks are placed within the same packet. The sinusoidal model is a frame based algorithm. Within any given frame, a track may start; after the start peak, the track may include from one to four differential peaks. All tracks within the same perceptual grouping and the same start frame are placed in the same packet. While this breaks the frame-based nature of the sinusoidal model, it allows a large degree of robustness to lost packets. If the network drops a packet, all tracks with that start frame are lost (limited to tracks within that perceptual grouping). However, certain tracks that had start frames within the previous four frames and future

tracks that start within the next four frames will reduce effects of the lost packet. In addition, any tracks from other perceptual groupings will help reduce the effects of the lost packet. Finally, the noise and transient models will help reduce the effects of the lost packet. In addition, because packet loss on the Internet tends to be bursty, the server may use packet interleaving schemes help alleviate the lost packets in a row problem.

4.3. Compression of transient parameters

The transient parameter set is the frequency domain dual to the sinusoidal set. As such, transient parameters are quantized in a similar fashion to the sinusoidal parameters. The steps for transient parameter compression include the following: magnitude quantization, time (frequency in the DCT-domain) quantization, parameter tracking, entropy code application, and finally bit-stream placement.

The transient quantizer compresses blocks of transient parameters that correspond to 1s of time-domain transients and places the information together in single packet for storage and transmission. Within each 1s block transient magnitudes are quantized to 6 bits on a logarithmic scale, transient times (DCT domain sinusoidal frequencies) are quantized to 16 bits, and phase parameters of the transient model are discarded because they are not perceptually important. As with the sinusoidal parameter compression, the encoder tracks the DCT-domain sinusoidal frequencies to take advantage of correlations between adjacent parameters. The correlation for the transient parameters is extremely high; a track often exists for the entire DCT block. The transient track length is not limited as in the sinusoidal track case. Therefore many transients are described by a single magnitude absolute, a single DCT-domain frequency absolute and many differentials. Again, to further reduce the bit-rate, an entropy code is applied to the differentials.

4.4. Bit-packing and scalability of transient parameters

The server either transmits an entire compressed 1s block of transient data or does not. This is the only aspect of scalability for transients. In addition, a transient packet loss results in a second of missing transients. However, this loss is acceptable at low bit-rates or for occasional dropped packets. In a typical piece of music, there are only a handful of transients in a 1s block. Additionally the sinusoidal model is perceptually more important; furthermore, the compressed sinusoids have a large degree scalability and packet loss robustness. These combined properties allow the music received at the client to be of reasonable quality in the face of transient losses.

4.5. Compression of noise parameters

The noise parameters are quite different compared to the sines or transient parameters. For each frame, the noise parameters consist of 15 ERB energy values. The steps for noise compression include energy quantization, intra-channel DPCM, entropy coding and finally bit-stream packing.

The encoder groups together 10 frames of noise parameters, which corresponds to about 160ms, for compression and placement into a single packet. The first step is quantization of the ERB energy values. In each frame, the 15 ERB energy values are quantized to 6 bits on a logarithmic scale. Once again the encoder uses DPCM and an entropy code to take advantage interframe dependencies. Within the 10 frame packet, each ERB channel is described by one absolute value and 9 differential values. The differential values are encoded with a scalar scalable entropy code. Although not currently exploited, this allows a wide degree of fine-grain scalability for the noise parameters.

4.6. Bit-packing and scalability of noise parameters

The server either transmits an entire compressed 160ms block of noise data or does not. This is currently the only aspect of scalability for the noise parameters. If packets are dropped during transmission, the client uses linear interpolation to restore missing data. Again packet interleaving schemes are used to minimize the number of contiguous dropped packets.

5. QUALITY ASSESSMENT AND CONCLUSIONS

Informal listening tests were performed to assess the quality of the scalable audio coder. It was found that the coder reasonably scaled and gracefully degraded in quality from perceptually lossless audio at 85Kbps to very-low quality audio at 6Kbps. The perceptually lossless bit-rate of 85Kbps is about a 33 percent higher than the 64Kbps MPEG-AAC perceptually lossless bit-rate. It is believed with more work, the 85Kbps perceptually lossless rate could be significantly reduced. At the lower bit-rates, from 6Kbps to 16Kbps, the coder was found to be competitive with current low bit-rate audio coders (e.g., Real Audio, or MPEG-4). From 16Kbps to 45Kbps, the coder was found to be of lower quality than current audio coders (e.g., MPEG-AAC). It should be noted, however, that scalable coder was compared against fixed bit-rate coders which are painfully tweaked for their given bit-rate. A server would need to store multiple compressed versions of the same file in order for these fixed-rate coders to meet the scalability of the audio coder presented.

6. REFERENCES

[1] H. Purnhagen, "Advances in parametric audio coding", in *Proc. WASPAA*, October 1999.

[2] B. Edler, et al., "ASAS-analysis/synthesis codec for very low bit rates", in *AES 100th Convention*, May 1996.

[3] K. Hamdy, et al., "Low bit rate high quality audio coding with combined harmonic and wavelet representations", in *Proc. ICASSP*, May 1996.

[4] S. Levine, *Audio Representations for Data Compression and Compressed Domain Processing*, PhD thesis, Stanford University, 1998.

[5] T. Verma, et al., "Transient modeling synthesis: a flexible transient analysis/synthesis tool for transient signals", in *Proc. ICMC*, September 1997.

[6] T. Verma, *A Perceptually Based Audio Signal Model with Application to Scalable Audio Compression*, PhD thesis, Stanford University, 1999.

[7] T. Verma and T. Meng, "Sinusoidal modeling using frame-based perceptually weighted matching pursuits", in *Proc. ICASSP*, March 1999.

[8] D. Schulz, "Improving audio codecs by noise substitution", *JAES*, vol. 44, no. 7/8, pp. 107–116, July/August 1996.

[9] M. Goodwin, "Residual modeling in music analysis/synthesis", in *Proc. ICASSP*, May 1996.