Digital Audio Compression

By Davis Yen Pan

Abstract

Compared to most digital data types, with the exception of digital video, the data rates associated with uncompressed digital audio are substantial. Digital audio compression enables more efficient storage and transmission of audio data. The many forms of audio compression techniques offer a range of encoder and decoder complexity, compressed audio quality, and differing amounts of data compression. The μ-law transformation and ADPCM coder are simple approaches with low-complexity, low-compression, and medium audio quality algorithms. The MPEG/audio standard is a high-complexity, high-compression, and high audio quality algorithm. These techniques apply to general audio signals and are not specifically tuned for speech signals.

Introduction

Digital audio compression allows the efficient storage and transmission of audio data. The various audio compression techniques offer different levels of complexity, compressed audio quality, and amount of data compression.

This paper is a survey of techniques used to compress digital audio signals. Its intent is to provide useful information for readers of all levels of experience with digital audio processing. The paper begins with a summary of the basic audio digitization process. The next two sections present detailed descriptions of two relatively simple approaches to audio compression: μ-law and adaptive differential pulse code modulation. In the following section, the paper gives an overview of a third, much more sophisticated, compression audio algorithm from the Motion Picture Experts Group. The topics covered in this section are quite complex and are intended for the reader who is familiar with digital signal processing. The paper concludes with a discussion of software-only real-time implementations.

Digital Audio Data

The digital representation of audio data offers many advantages: high noise immunity, stability, and reproducibility. Audio in digital form also allows the efficient implementation of many audio processing functions (e.g., mixing, filtering, and equalization) through the digital computer.

The conversion from the analog to the digital domain begins by sampling the audio input in regular, discrete intervals of time and quantizing the sampled values into a discrete number of evenly spaced levels. The digital audio data consists of a sequence of binary values representing the number of quantizer levels for each audio sample. The method of representing each sample with an independent code word is called pulse code modulation (PCM). Figure 1 shows the digital audio process.

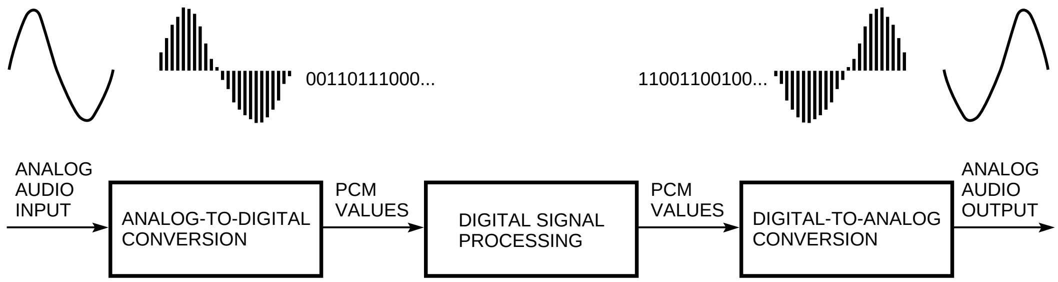


Figure 1 Digital Audio Process

According to the Nyquist theory, a time-sampled signal can faithfully represent signals up to half the sampling rate.[1] Typical sampling rates range from 8 kilohertz (kHz) to 48 kHz. The 8-kHz rate covers a frequency range up to 4 kHz and so covers most of the frequencies produced by the human voice. The 48-kHz rate covers a frequency range up to 24 kHz and more than adequately covers the entire audible frequency range, which for humans typically extends to only 20 kHz. In practice, the frequency range is somewhat less than half the sampling rate because of the practical system limitations.

The number of quantizer levels is typically a power of 2 to make full use of a fixed number of bits per audio sample to represent the quantized values. With uniform quantizer step spacing, each additional bit has the potential of increasing the signal-to-noise ratio, or equivalently the dynamic range, of the quantized amplitude by roughly 6 decibels (dB). The typical number of bits per sample used for digital audio ranges from 8 to 16. The dynamic range capability of these representations thus ranges from 48 to 96 dB, respectively. To put these ranges into perspective, if 0 dB represents the weakest audible sound pressure level, then 25 dB is the minimum noise level in a typical recording studio, 35 dB is the noise level inside a quiet home, and 120 dB is the loudest level before discomfort begins.[2] In terms of audio perception, 1 dB is the minimum audible change in sound pressure level under the best conditions, and doubling the sound pressure level amounts to one perceptual step in loudness.

Compared to most digital data types (digital video excluded), the data rates associated with uncompressed digital audio are substantial. For example, the audio data on a compact disc (2 channels of audio sampled at 44.1 kHz with 16 bits per sample) requires a data rate of about 1.4 megabits per second. There is a clear need for some form of compression to enable the more efficient storage and transmission of this data.

The many forms of audio compression techniques differ in the trade-offs between encoder and decoder complexity, the compressed audio quality, and the amount of data compression. The techniques presented in the following sections of this paper cover the full range from the μ-law, a low-complexity, low-compression, and medium audio quality algorithm, to MPEG/audio, a high-complexity, high-compression, and high audio quality algorithm. These techniques apply to general audio signals and are not specifically tuned for speech signals. This paper does not cover audio compression algorithms designed specifically for speech signals. These algorithms are generally based on a modeling of the vocal tract and do not work well for nonspeech audio signals.[3,4] The federal standards 1015 LPC (linear predictive coding) and 1016 CELP (coded excited linear prediction) fall into this category of audio compression.

μ-law Audio Compression

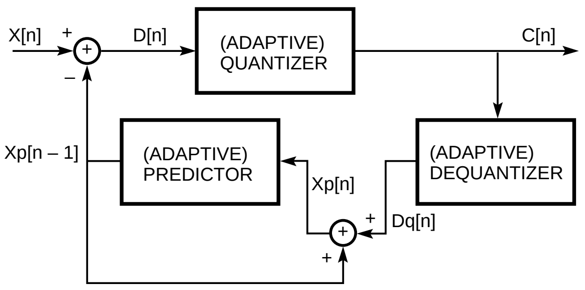
The μ-law transformation is a basic audio compression technique specified by the Comité Consultatif Internationale de Télégraphique et Téléphonique (CCITT) Recommendation G.711.[5] The transformation is essentially logarithmic in nature and allows the 8 bits per sample output codes to cover a dynamic range equivalent to 14 bits of linearly quantized values. This transformation offers a compression ratio of (number of bits per source sample) /8 to 1. Unlike linear quantization, the logarithmic step spacings represent low-amplitude audio samples with greater accuracy than higher-amplitude values. Thus the signal-to-noise ratio of the transformed output is more uniform over the range of amplitudes of the input signal. The μ-law transformation is

$$y = \left\{ \begin{array}{l} 255 - \frac{127}{ln(1+\mu)} \times ln(1+\mu|x|) \;\; for\; x \geq 0 \\ 127 - \frac{127}{ln(1+\mu)} \times ln(1+\mu|x|) \;\; for\; x < 0 \end{array} \right\}$$

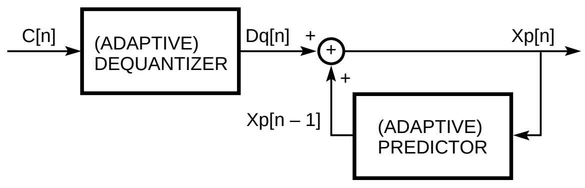
where m = 255, and x is the value of the input signal normalized to have a maximum value of 1. The CCITT Recommendation G.711 also specifies a similar A-law transformation. The μ-law transformation is in common use in North America and Japan for the Integrated Services Digital Network (ISDN) 8-kHz-sampled, voice-grade, digital telephony service, and the A-law transformation is used elsewhere for the ISDN telephony.

Adaptive Differential Pulse Code Modulation

Figure 2 shows a simplified block diagram of an adaptive differential pulse code modulation (ADPCM) coder.[6] For the sake of clarity, the figure omits details such as bit-stream formatting, the possible use of side information, and the adaptation blocks. The ADPCM coder takes advantage of the fact that neighboring audio samples are generally similar to each other. Instead of representing each audio sample independently as in PCM, an ADPCM encoder computes the difference between each audio sample and its predicted value and outputs the PCM value of the differential. Note that the ADPCM *encoder* (Figure 2a) uses most of the components of the ADPCM *decoder* (Figure 2b) to compute the predicted values.



(a) ADPCM Encoder



(b) ADPCM Decoder

Figure 2   ADPCM Compression and Decompression

The quantizer output is generally only a (signed) representation of the number of quantizer levels. The requantizer reconstructs the value of the quantized sample by multiplying the number of quantizer levels by the quantizer step size and possibly adding an offset of half a step size. Depending on the quantizer implementation, this offset may be necessary to center the requantized value between the quantization thresholds.

The ADPCM coder can adapt to the characteristics of the audio signal by changing the step size of either the quantizer or the predictor, or by changing both. The method of computing the predicted value and the way the predictor and the quantizer adapt to the audio signal vary among different ADPCM coding systems.

Some ADPCM systems require the encoder to provide side information with the differential PCM values. This side information can serve two purposes. First, in some ADPCM schemes the decoder needs the additional information to determine either the predictor or the quantizer step size, or both. Second, the data can provide redundant contextual information to the decoder to enable recovery from errors in the bit stream or to allow random access entry into the coded bit stream.

The following section describes the ADPCM algorithm proposed by the Interactive Multimedia Association (IMA). This algorithm offers a compression factor of (number of bits per source sample)/4 to 1. Other ADPCM audio compression schemes include the CCITT Recommendation G.721 (32 kilobits per second compressed data rate) and Recommendation G.723 (24 kilobits per second compressed data rate) standards and the compact disc interactive audio compression algorithm.[7,8]

The IMA ADPCM Algorithm. The IMA is a consortium of computer hardware and software vendors cooperating to develop a de facto standard for computer multimedia data. The IMA's goal for its audio compression proposal was to select a public-domain audio compression algorithm able to provide good compressed audio quality with good data compression performance. In addition, the algorithm had to be simple enough to enable software-only, real-time decompression of stereo, 44.1-kHz-sampled, audio signals on a 20-megahertz (MHz) 386-class computer. The selected ADPCM algorithm not only meets these goals, but is also simple enough to enable software-only, real-time encoding on the same computer.

The simplicity of the IMA ADPCM proposal lies in the crudity of its predictor. The predicted value of the audio sample is simply the decoded value of the immediately previous audio sample. Thus the predictor block in Figure 2 is merely a time-delay element whose output is the input delayed by one audio sample interval. Since this predictor is not adaptive, side information is not necessary for the reconstruction of the predictor.

Figure 3 shows a block diagram of the quantization process used by the IMA algorithm. The quantizer outputs four bits representing the signed magnitude of the number of quantizer levels for each input sample.

Adaptation to the audio signal takes place only in the quantizer block. The quantizer adapts the step size based on the current step size and the quantizer output of the immediately previous input. This adaptation can be done as a sequence of two table lookups. The three bits representing the number of quantizer levels serve as an index into the first table lookup whose output is an index adjustment for the second table lookup. This adjustment is added to a stored index value, and the range-limited result is used as the index to the second table lookup. The summed index value is stored for use in the next iteration of the step-size adaptation. The output of the second table lookup is the new quantizer step size. Note that given a starting value for the index into the second table lookup, the data used for adaptation is completely deducible from the quantizer outputs; side information is not required for the quantizer adaptation. Figure 4 illustrates a block diagram of the step-size adaptation process, and Tables 1 and 2 provide the table lookup contents.

Table 1
First Table Lookup for the IMA ADPCM Quantizer Adaptation

Three Bits Quantized Magnitude	Index Adjustment
000	-1
001	-1
010	-1
011	-1
100	2
101	4
110	6
111	8

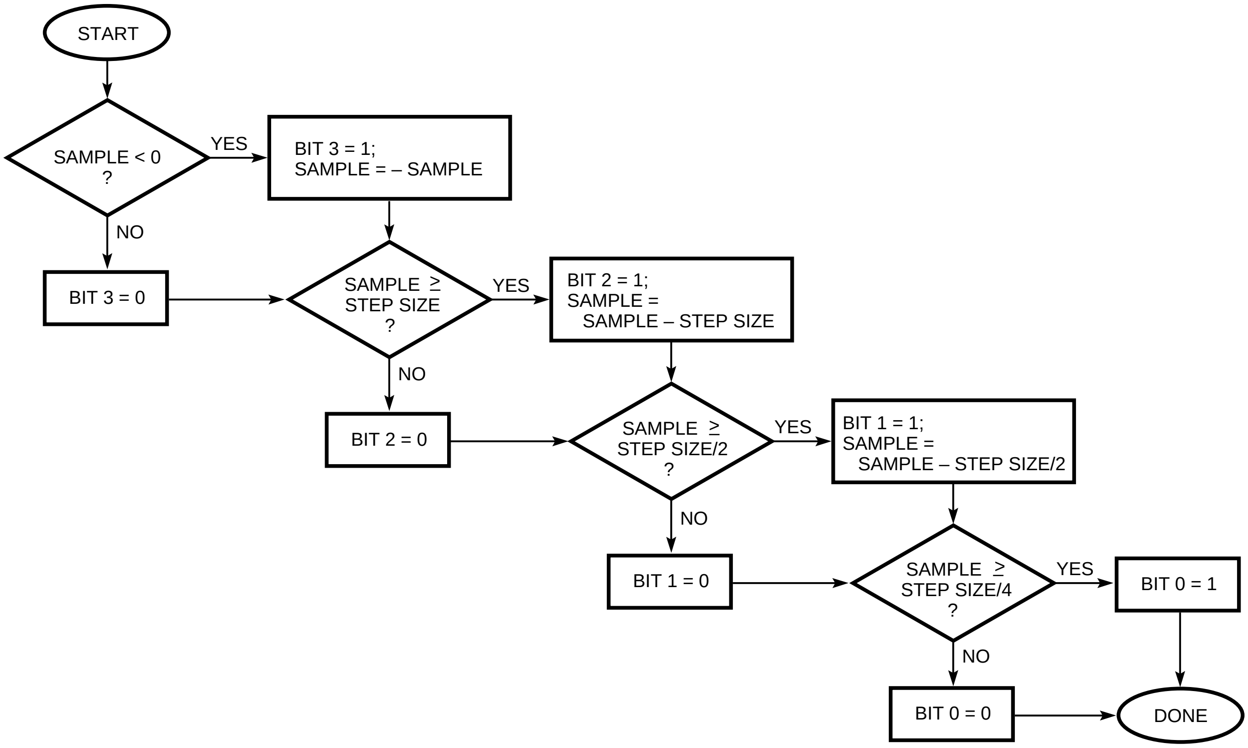


Figure 3 IMA ADPCM Quantization

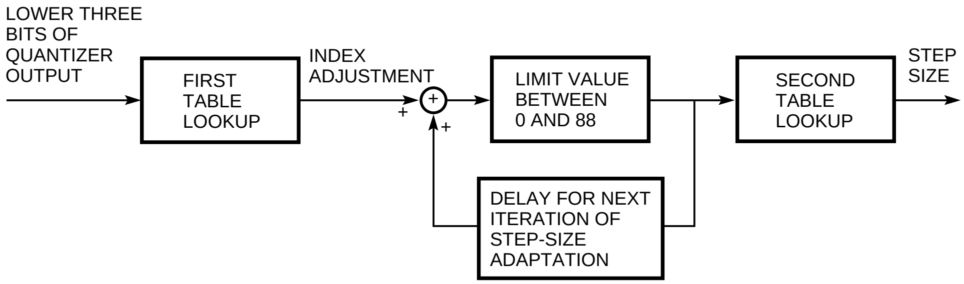


Figure 4   IMA ADPCM Step-size Adaptation

Table 2
Second Table Lookup for the IMA ADPCM Quantizer Adaptation

Index	Step Size	Index	Step Size	Index	Step Size	Index	Step Size
0	7	22	60	44	494	66	4,026
1	8	23	66	45	544	67	4,428
2	9	24	73	46	598	68	4,871
3	10	25	80	47	658	69	5,358
4	11	26	88	48	724	70	5,894
5	12	27	97	49	796	71	6,484
6	13	28	107	50	876	72	7,132
7	14	29	118	51	963	73	7,845
8	16	30	130	52	1,060	74	8,630
9	17	31	143	53	1,166	75	9,493
10	19	32	157	54	1,282	76	10,442
11	21	33	173	55	1,411	77	11,487
12	23	34	190	56	1,552	78	12,635
13	25	35	209	57	1,707	79	13,899
14	28	36	230	58	1,878	80	15,289
15	31	37	253	59	2,066	81	16,818
16	34	38	279	60	2,272	82	18,500
17	37	39	307	61	2,499	83	20,350
18	41	40	337	62	2,749	84	22,358
19	45	41	371	63	3,024	85	24,623
20	50	42	408	64	3,327	86	27,086
21	55	43	449	65	3,660	87	29,794
						88	32,767

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.