The Audio Data Type
Coding & Compression Basics

Kevin Jeffay
Department of Computer Science
University of North Carolina at Chapel Hill
jeffay@cs.unc.edu

August 26, 1999

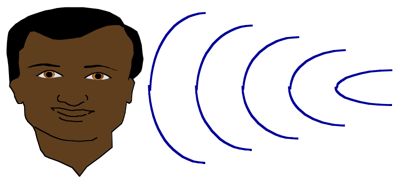
http://www.cs.unc.edu/~jeffay/courses/comp249f99

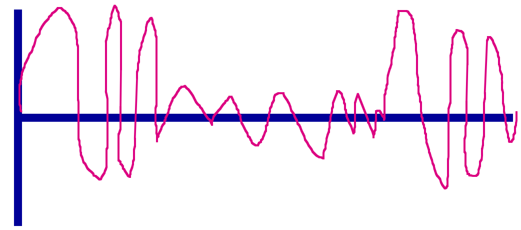
The Audio Data Type
Outline

- Basic audio concepts
- Digital audio
  - » Sampling & quantization
- Coding/Compression
  - » Generic compression algorithms
    - PCM
    - DPCM
    - ADPCM
  - » Audio-specific algorithms
    - μ-Law
    - MPEG-Audio
  - » Speech-specific algorithms
    - LPC
    - CELP

The Audio Data Type

Basic audio concepts



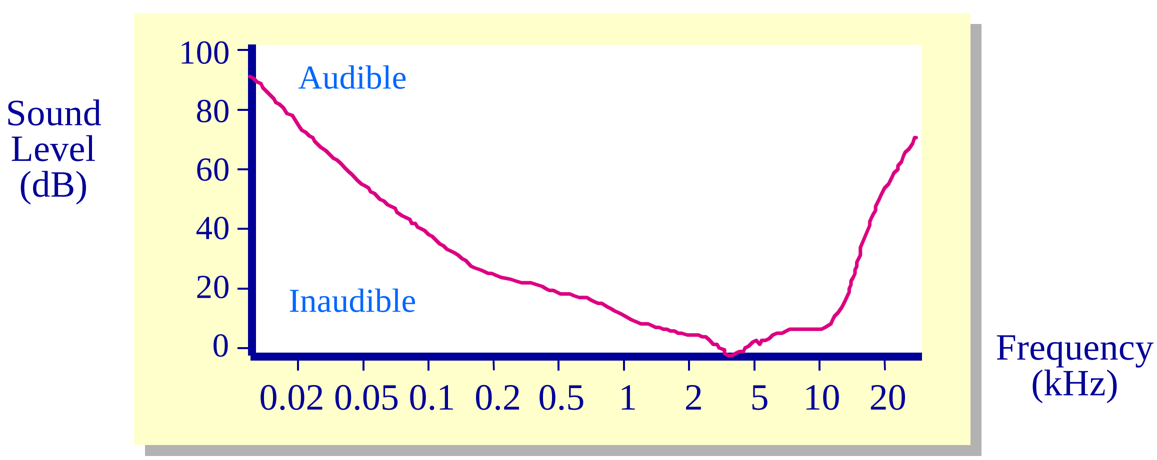


- Audio — A disturbance of the air pressure near the ear
  - » Humans can perceive changes occurring in the range of 20 - 20,000 changes (cycles) per second (Hz)
  - » Natural human voice in the range of 80 - 3,400 Hz
  - » Humans can perceive changes in magnitude of air pressure (amplitude) from 0 - 120 dB

$$\text{dB} = 20 \log_{10} \frac{x}{y}$$

The Audio Data Type

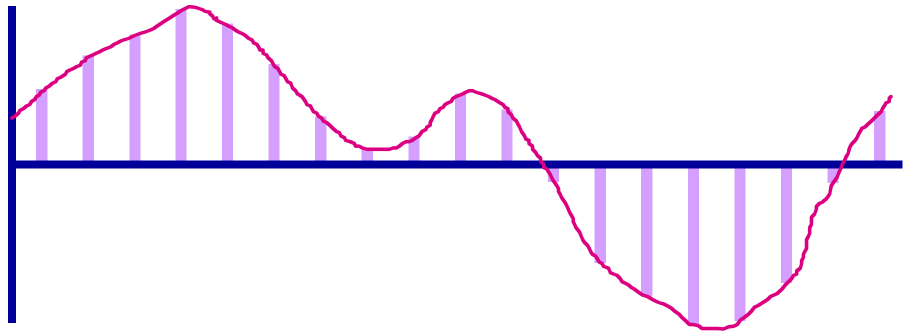
Basic audio concepts



- Human perception of sound is a function of frequency and signal strength
  - » (Many audio coding schemes exploit this relationship.)

Digital Audio

Sampling



- Pulse Amplitude Modulation (PAM)
  - » Each sample's amplitude represented by 1 value
- Sampling theory (Nyquist)
  - » If input signal has maximum frequency (bandwidth) *f*, sampling frequency must be at least 2*f*
  - » With a low-pass filter to interpolate between samples, the input signal can be fully reconstructed

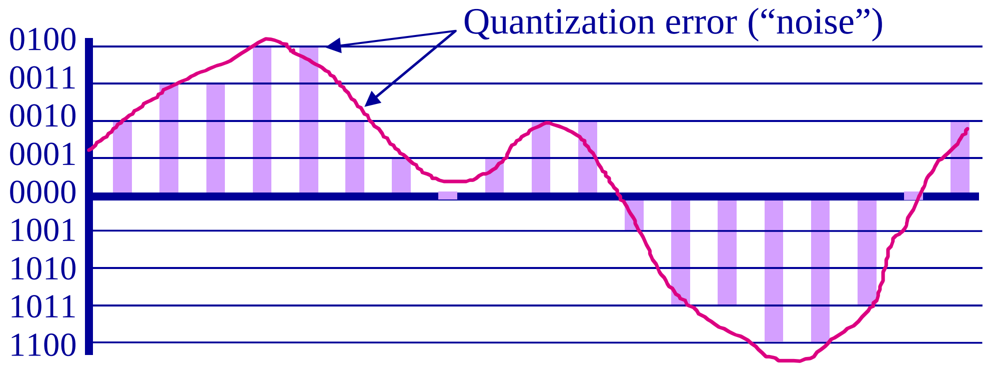
Digital Audio

Canonical sampling rates

Format	Sampling Rate (kHz)	Bandwidth (kHz)	Frequency Range (Hz)
Telephony	8.0	3.0	200 - 3,200
Teleconferencing	16.0	7.0	50 - 7,000
Compact Disk	44.1	20.0	20 - 20,000
Digital Audio Tape	48.0	20.0	20 - 20,000

Digital Audio

Quantization & encoding



- Pulse Code Modulation (PCM)
  - » Each sample's amplitude represented by an integer code-word
  - » Each bit of resolution adds 6 dB of dynamic range
  - » Number of bits required depends on the amount of noise that is tolerated

$$n = \frac{\mathrm{SNR} - 1.76 - 20\log_{10}\frac{A}{A_{max}}}{6.02}$$

Digital Audio

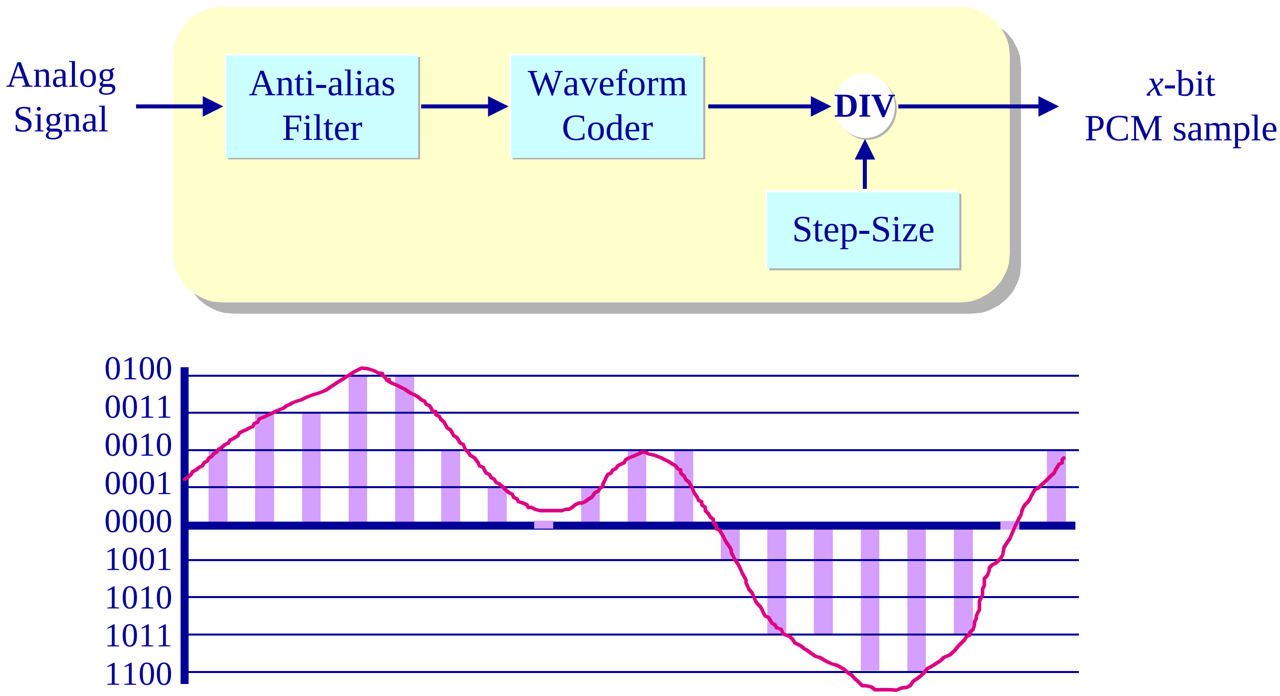
Canonical bit-rates

Format	Bandwidth (kHz)	Sampling Rate (kHz)	Sample Size (bits)	Dynamic Range (dB)	Bit Rates (kbps)
Telephony	3.0	8.0	8	48	64
Teleconferencing	7.0	16.0	16	96	256
Compact Disk	20.0	44.1	16	96	1,410
Digital Audio Tape	20.0	48.0	16	96	1,536

- What's "a lot" to transmit over the Internet?

Digital Audio

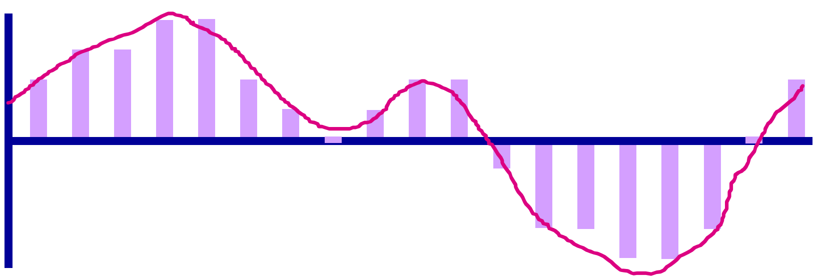
A linear PCM analog-to-digital coder



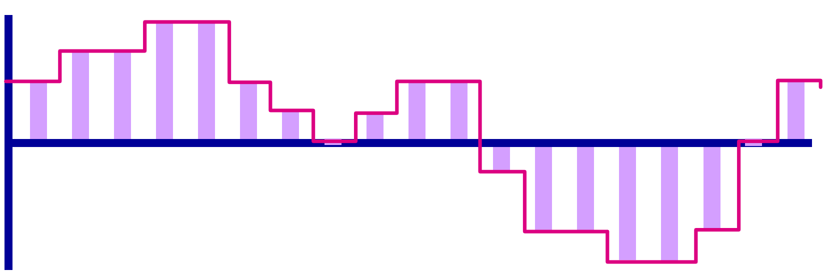
Digital Audio

Sampling & quantization

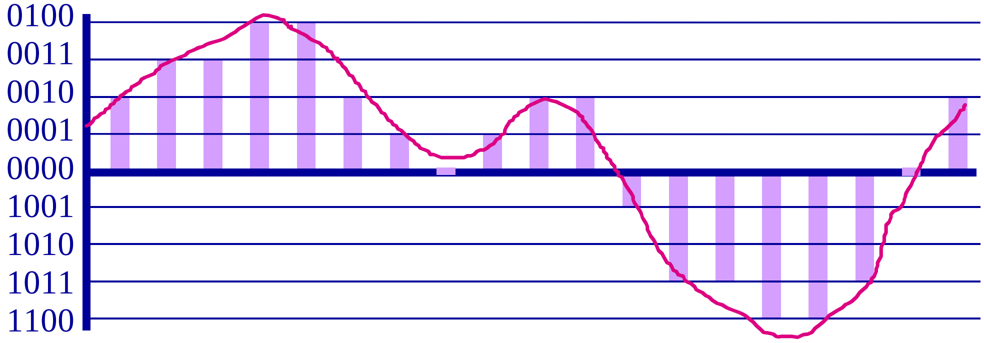
- Sampled input



- Decoded output

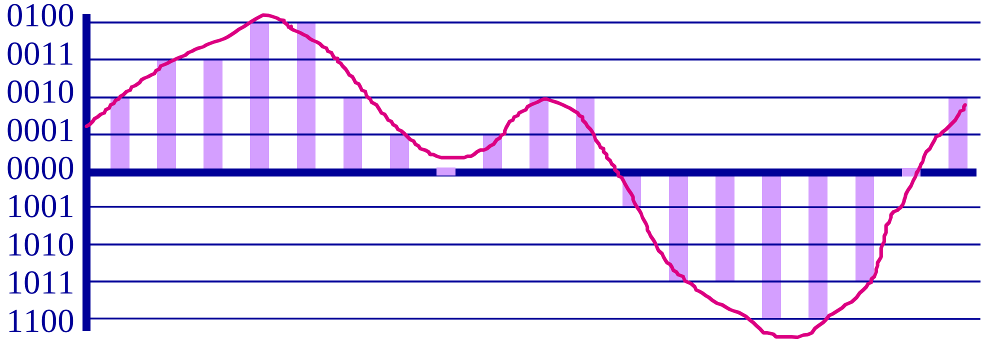


Audio Compression
Basics



- Generic techniques
  - » Difference encodings
  - » Vector quantization
- Audio-specific techniques
  - » Transform & psycho-acoustical based encoding
- Speech-specific techniques
  - » Speech synthesis

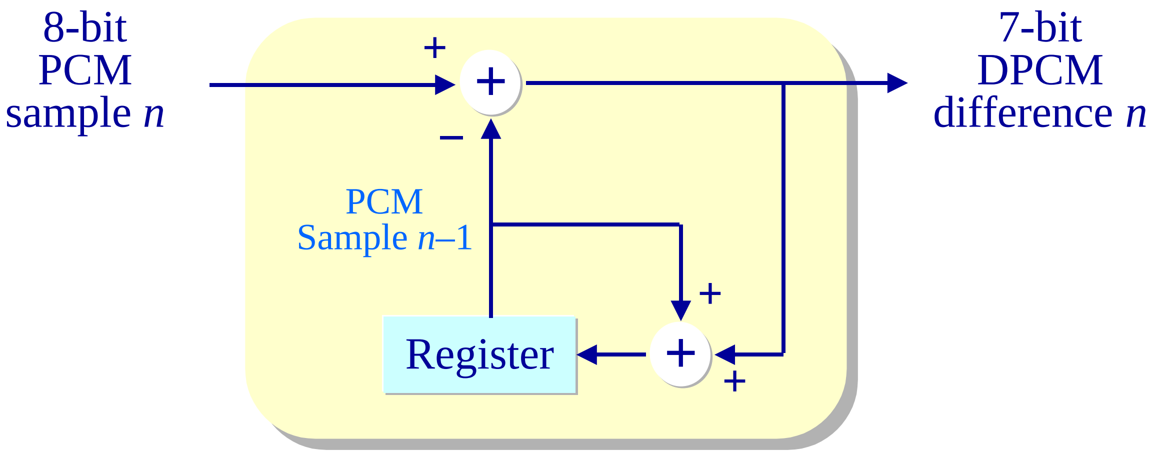
Audio Compression
Difference encodings



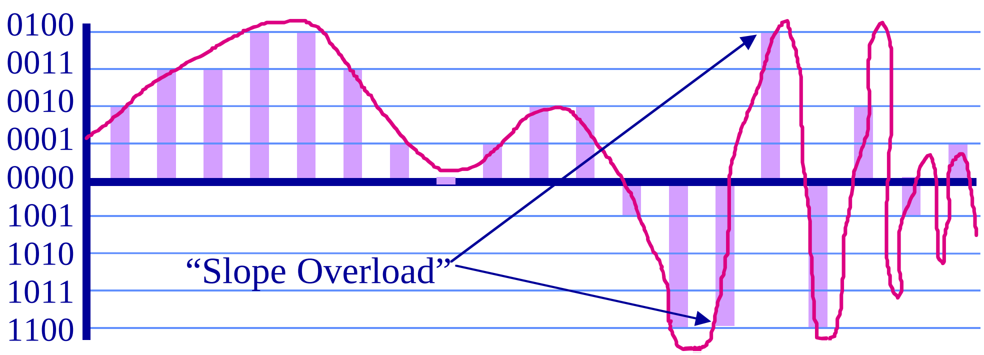
- Differential-PCM (DPCM)
  - » Exploit temporal redundancy in samples
  - » Difference between 2 *x*-bit samples can be represented with significantly fewer than *x*-bits
  - » Transmit the *difference* (rather than the sample)
    - ❖ 8-bit audio sampled at 8 kHz audio can be reduced to 56 kb/s (7 bits/sample)

Differencing Compression
DPCM Implementation

- Simple to implement



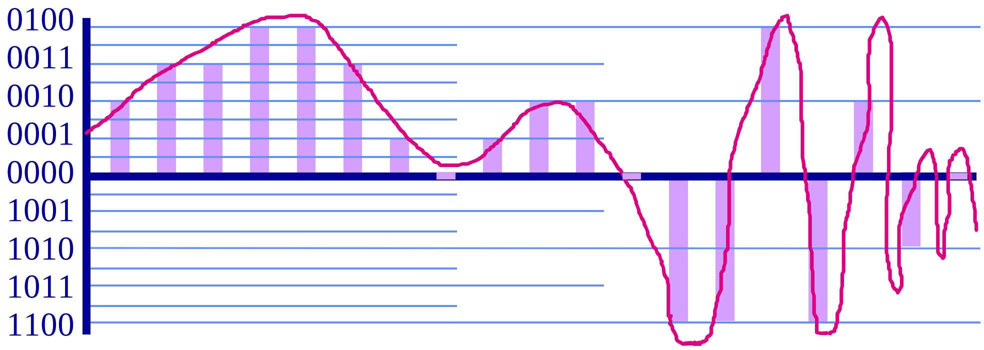
Differencing Compression
The slope overload problem



- Differences in high frequency signals near the Nyquist frequency cannot be represented with a smaller number of bits!
  - » Error introduced leads to severe distortion in the higher frequencies

Differencing Compression

Adaptive differential PCM (ADPCM)

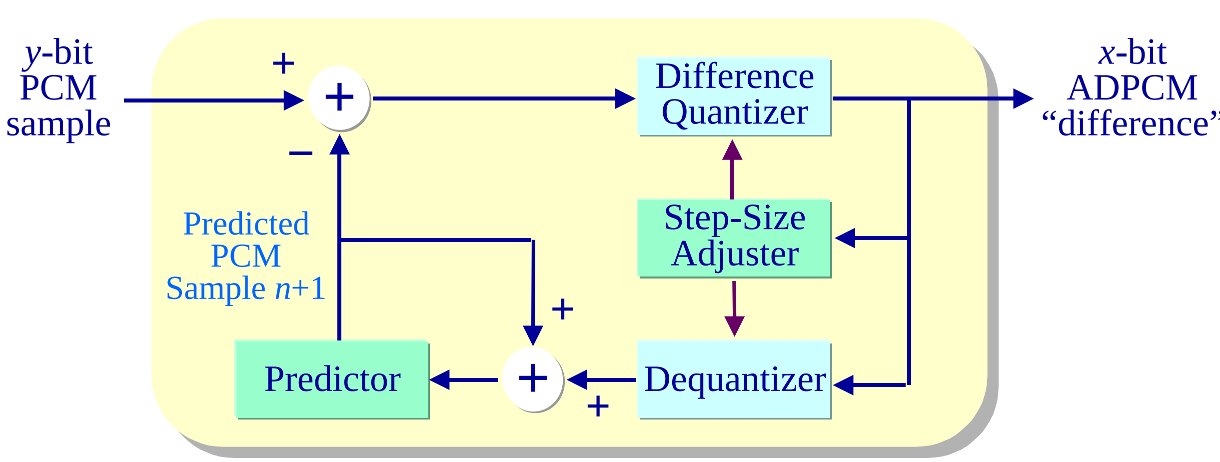


- Use a larger step-size to encode differences between high-frequency samples & a smaller step-size for differences between low-frequency samples
- Use previous sample values to estimate changes in the signal in the near future

Differencing Compression

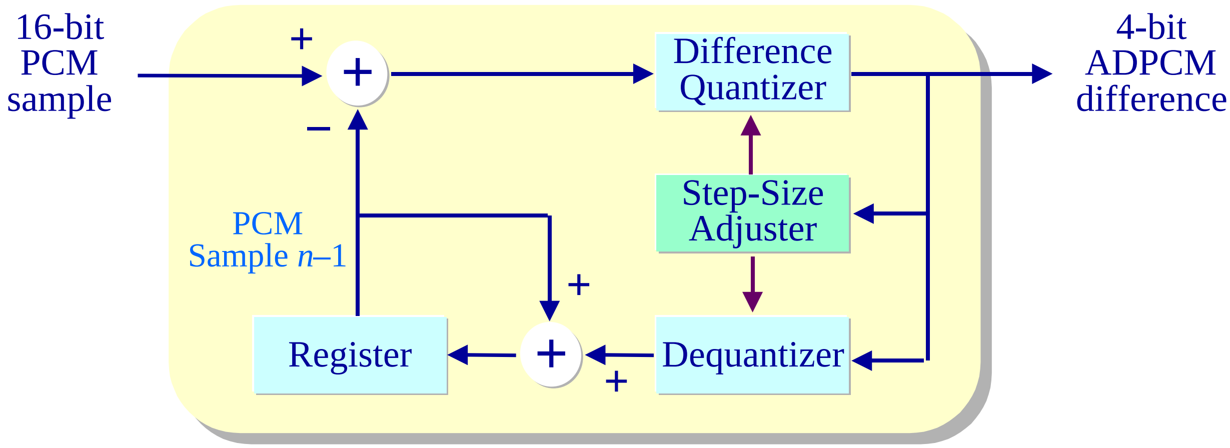
Adaptive differential PCM (ADPCM)

- To ensure differences are always small...
  - » Adaptively change the step-size (quanta)
  - » (Adaptively) attempt to predict next sample value



Differencing Compression

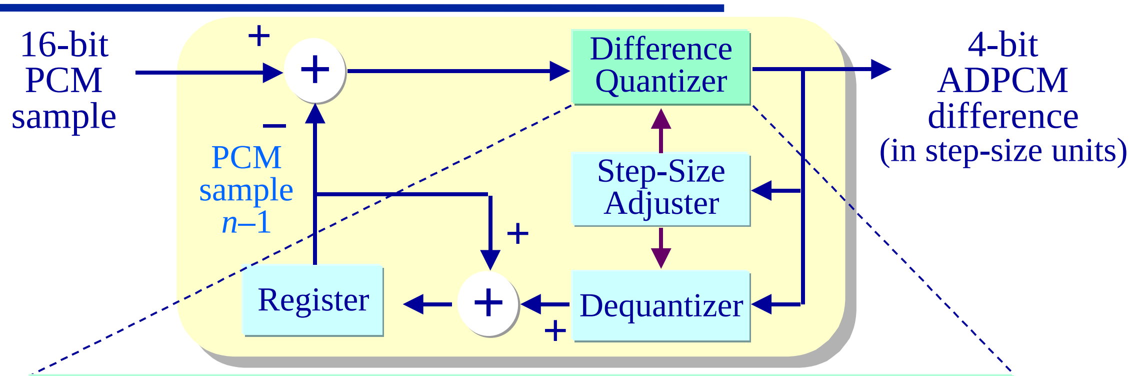
IMA's proposed ADPCM



- Predictor is not adaptive and simply uses the last sample value
- Quantization step-size increases logarithmically with signal frequency

IMA ADPCM

Difference quantization

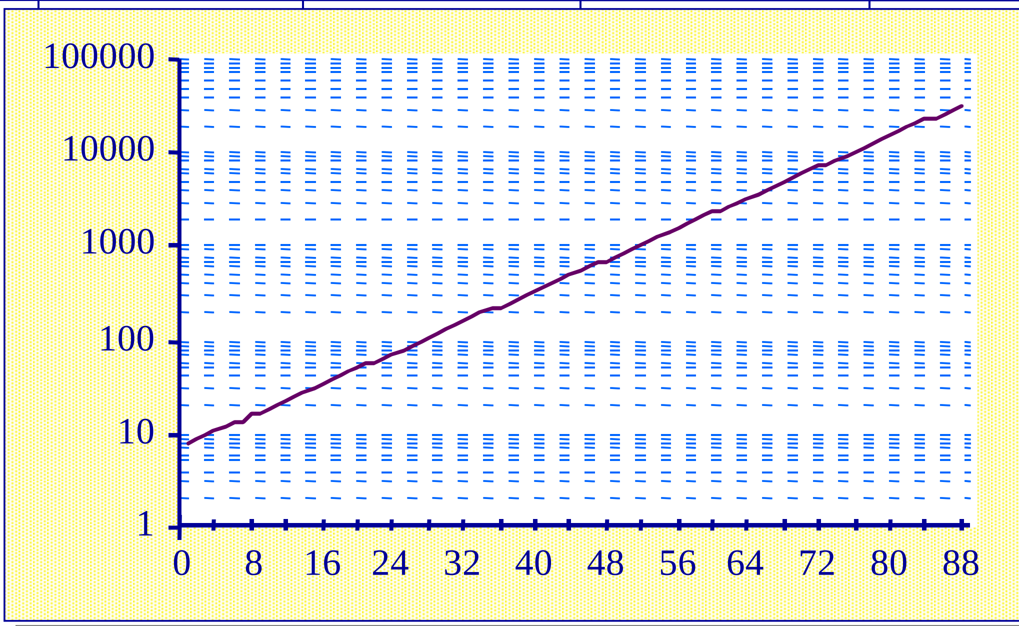


Quantization	Quantizer Output	Step-Size Multiples
difference < ¼ *step_size*	000	0.0
¼ *step_size* ≤ *difference* < ½ *step_size*	001	0.25
½ *step_size* ≤ *difference* < ¾ *step_size*	010	0.50
¾ *step_size* ≤ *difference* < *step_size*	011	0.75
step_size ≤ *difference* < 5⁄4 *step_size*	100	1.0
5⁄4 *step_size* ≤ *difference* < 3⁄2 *step_size*	101	1.25
3⁄2 *step_size* ≤ *difference* < 7⁄4 *step_size*	110	1.5
7⁄4 *step_size* ≤ *difference*	111	1.75

IMA ADPCM

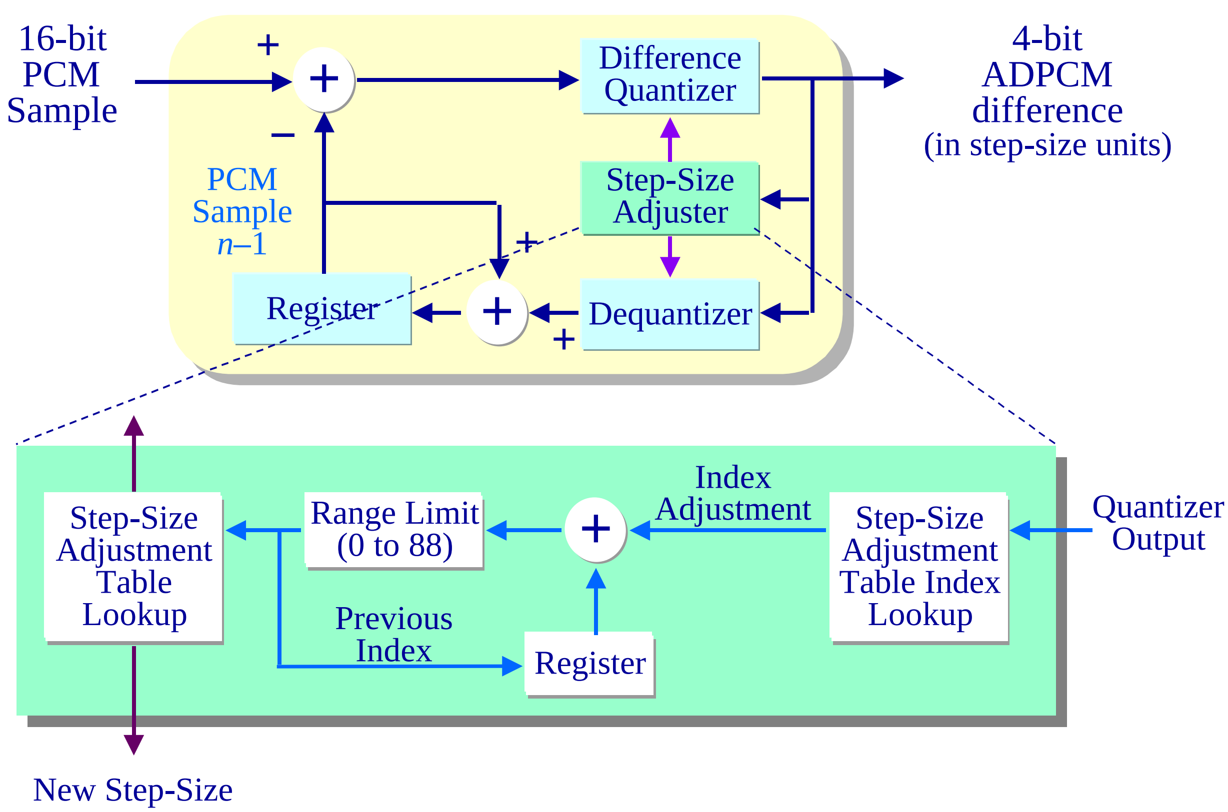
Step-size adjustment table

Index	Step Size	Index	Step Size	Index	Step Size	Index	Step Size	Index	Step Size
0	7								7132
1	[illegible]								7845
2									8630
3									9493
4	[illegible]								0442
5	12								1487
6	13								2635
7	14								3899
8	16								5289
9	17								6818
10	19								8500
11	21								20350
12	23								22358
13	25								24623
14	28								27086
15	31	33	173	51	963	69	5358	87	29794
16	34	34	190	52	1060	70	5894		32767
17	37	35	209	53	1166	71	6484		



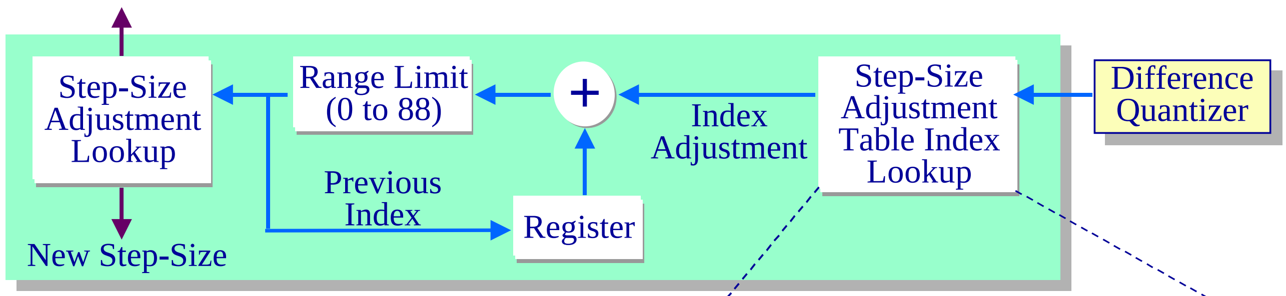
IMA ADPCM

Adaptive step-size selection



IMA ADPCM

Adaptive step-size selection

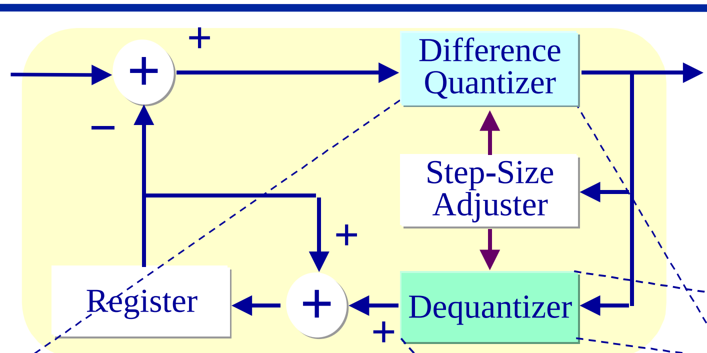


Quantization
difference < ¼ *step_size*
¼ *step_size* ≤ *difference* < ½ *step_size*
½ *step_size* ≤ *difference* < ¾ *step_size*
¾ *step_size* ≤ *difference* < *step_size*
step_size ≤ *difference* < 5⁄4 *step_size*
5⁄4 *step_size* ≤ *difference* < 3⁄2 *step_size*
3⁄2 *step_size* ≤ *difference* < 7⁄4 *step_size*
7⁄4 *step_size* ≤ *difference*

Quantizer Output	Step-Size-Adjustment-Table Adjustment
000	-1
001	-1
010	-1
011	-1
100	2
101	4
110	6
111	8

IMA ADPCM

Dequantization

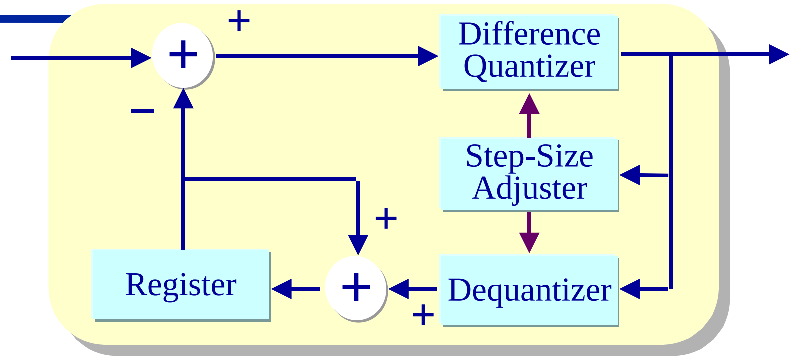


Input	Difference	Step Size	Quantizer output	Step-size multiplier	Reconstituted difference	Predicted value
X	Δ	Step	Q	M	Δ	Decode
150		7				
155	5	7				

Quantization	Quantizer Output
difference < ¼ *step_size*	000
¼ *step_size* ≤ *difference* < ½ *step_size*	001
½ *step_size* ≤ *difference* < ¾ *step_size*	010
¾ *step_size* ≤ *difference* < *step_size*	011
step_size ≤ *difference* < 5⁄4 *step_size*	100
5⁄4 *step_size* ≤ *difference* < 3⁄2 *step_size*	101
3⁄2 *step_size* ≤ *difference* < 7⁄4 *step_size*	110
7⁄4 *step_size* ≤ *difference*	111

Quantizer Output	Reconstituted Difference
000	0
001	0.25 x *step_size*
010	0.50 x *step_size*
011	0.75 x *step_size*
100	1.0 x *step_size*
101	1.25 x *step_size*
110	1.5 x *step_size*
111	1.75 x *step_size*

IMA ADPCM
Example

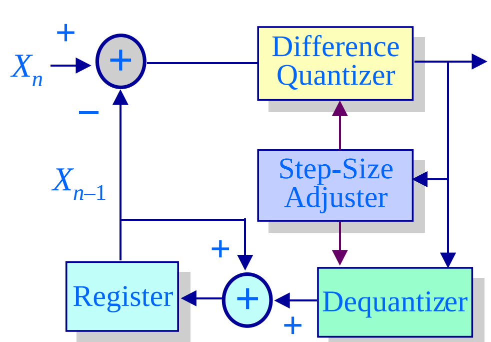


Quantization	Quantizer Output	Step-Size Table Adjustment	Step-Size Multiplier
$sample < \frac{1}{4}\ step\_size$	000	-1	0.0
$\frac{1}{4}\ step\_size \le sample < \frac{1}{2}\ step\_size$	001	-1	0.25
$\frac{1}{2}\ step\_size \le sample < \frac{3}{4}\ step\_size$	010	-1	0.50
$\frac{3}{4}\ step\_size \le sample < step\_size$	011	-1	0.75
$step\_size \le sample < \frac{5}{4}\ step\_size$	100	2	1.0
$\frac{5}{4}\ step\_size \le sample < \frac{3}{2}\ step\_size$	101	4	1.25
$\frac{3}{2}\ step\_size \le sample < \frac{7}{4}\ step\_size$	110	6	1.5
$\frac{7}{4}\ step\_size \le sample$	111	8	1.75

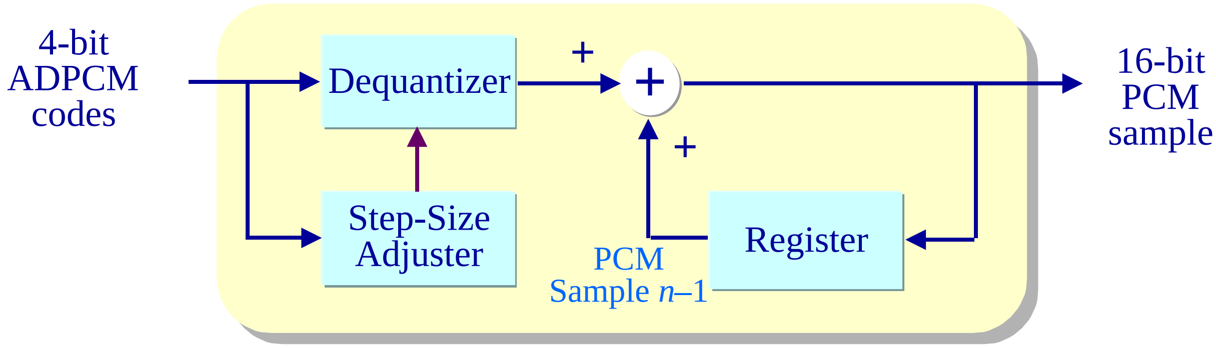
IMA ADPCM
Example

Input, Difference, Step Size, Quantizer output, Index Adjustment, SS adj. table index, Step-size multiplier, Reconstituted difference, Predicted value

X	Δ	Step	Q	Adj	I	M	Δ	Decode
150		7			0			150
155	5	7	010	-1	0	0.5	3.5	154
167								
170								
250								
250								
250								
250								
200								
200								
200								
200								
200								
200								
200								

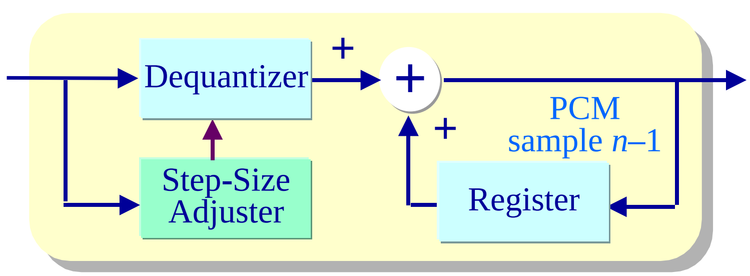


IMA ADPCM
Decoding



- Reconstitute the difference between samples and sum
  - » While adaptively adjusting the step-size
- What happens if code-word(s) are lost in the network?
  - » How does the decoder re-synchronize with the coder?

Differencing Compression
Networking considerations



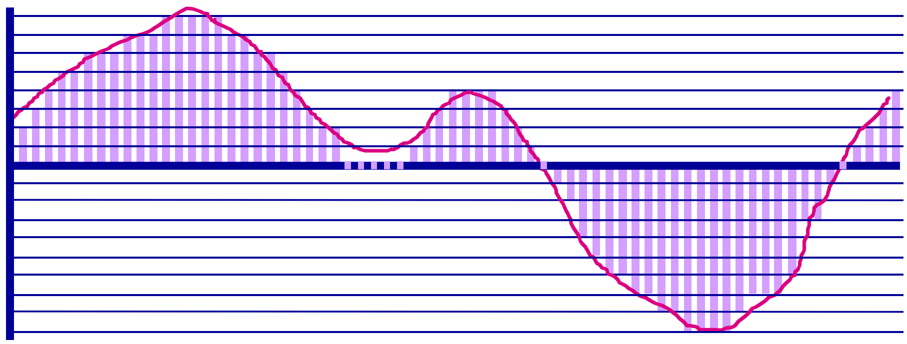
- The IMA codec is reasonably robust to errors
  - » An interval with a low-level signal will correct any step-size error

Quantization
difference $<$ ¼ *step_size*
¼ *step_size* $\le$ *difference* $<$ ½ *step_size*
½ *step_size* $\le$ *difference* $<$ ¾ *step_size*
¾ *step_size* $\le$ *difference* $<$ *step_size*
step_size $\le$ *difference* $<$ 5⁄4 *step_size*
5⁄4 *step_size* $\le$ *difference* $<$ 3⁄2 *step_size*
3⁄2 *step_size* $\le$ *difference* $<$ 7⁄4 *step_size*
7⁄4 *step_size* $\le$ *difference*

Quantizer Output	Step-Size-Adjustment-Table Adjustment
000	-1
001	-1
010	-1
011	-1
100	2
101	4
110	6
111	8

Differencing Compression

Delta modulation

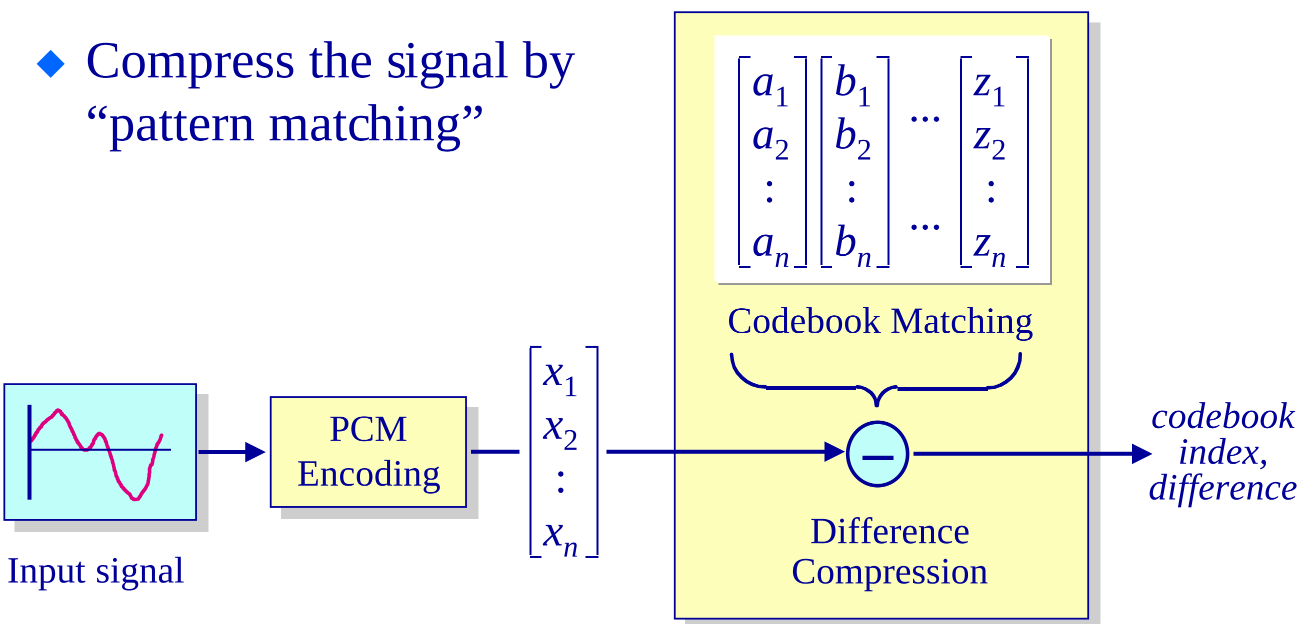


- Use a single bit to encode the difference between consecutive samples
  - » samples assumed to never change by more than 1 quantum
- Requires a much higher sampling rate (and small quantum) to work
  - » slope overload is much more serious

Other Generic Compression Schemes

Vector quantization

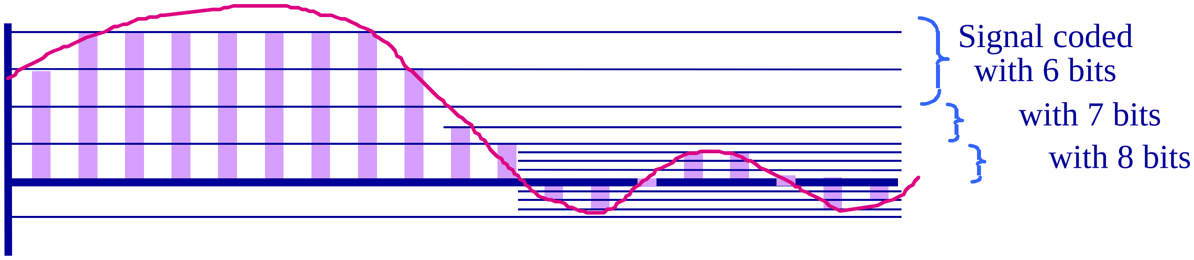
- Compress the signal by "pattern matching"



Vector Quantization

Audio-Specific Compression Techniques

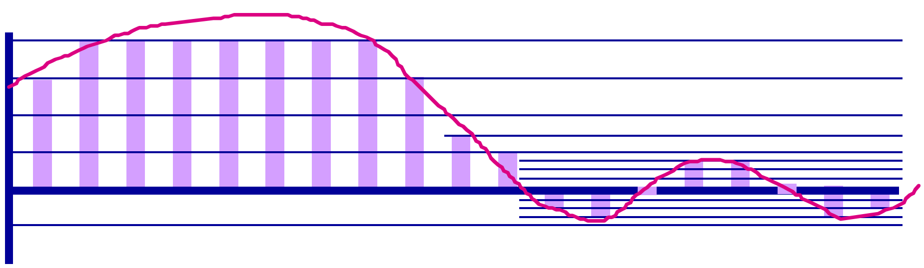
μ-Law companding (ITU Rec. G.711)



- Non-linear quantization of the signal's amplitude
  - » Quantization step-size decreases logarithmically with signal level
  - » Low-amplitude samples represented with greater accuracy (more bits) than high-amplitude samples
  - » Humans are less sensitive to changes in "loud" sounds than "quiet" sounds

Audio-Specific Compression Techniques

μ-Law companding

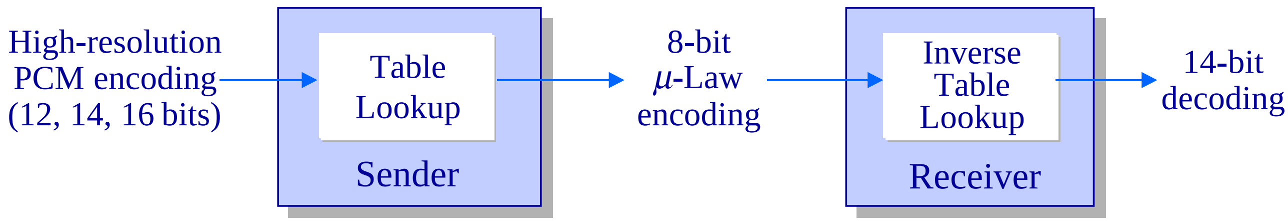


- Provides 14-bit quality (dynamic range) with an 8-bit encoding
- Used in North American & Japanese ISDN voice service
- Simple to compute encoding

$$f(x) = 127 \times \mathrm{sign}(x) \times \frac{\ln(1 + \mu|x|)}{\ln(1 + \mu)} \qquad (x \text{ normalized to } [-1, 1])$$

μ-Law Companding

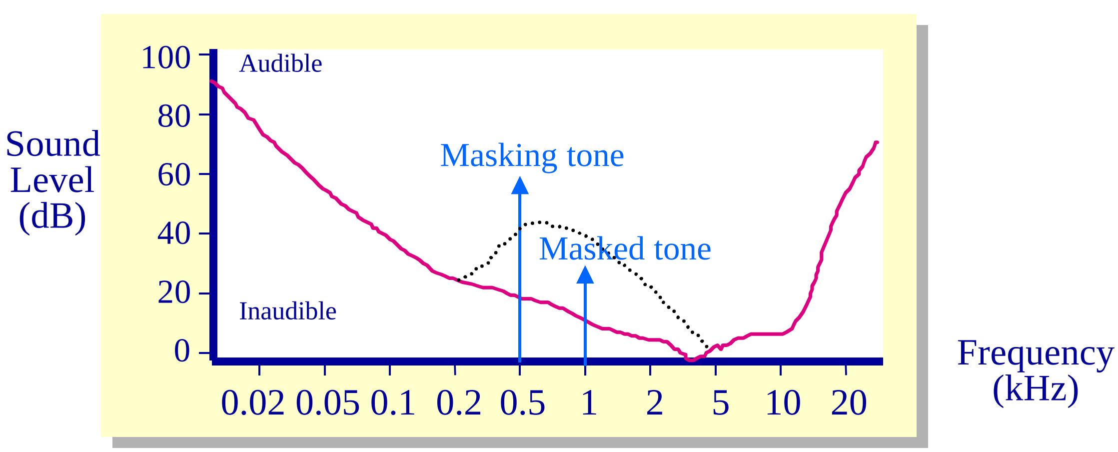
Example



Input Amplitude	Step Size	Segment	Quantization	Code Value	Decode Amplitude
0-1	1		0000	0	0
1-3	2	000	0001	1	2
...			...	...	...
29-31			1111	15	30
31-35	4	001	0000	16	33
...			...	...	...
91-95			1111	31	93
95-103	8	010	0000	32	99
...			...	...	...
215-223			1111	47	219
223-239	16	011	0000	48	231
...			...	...	...
463-479			1111	63	471
...			...	...	...

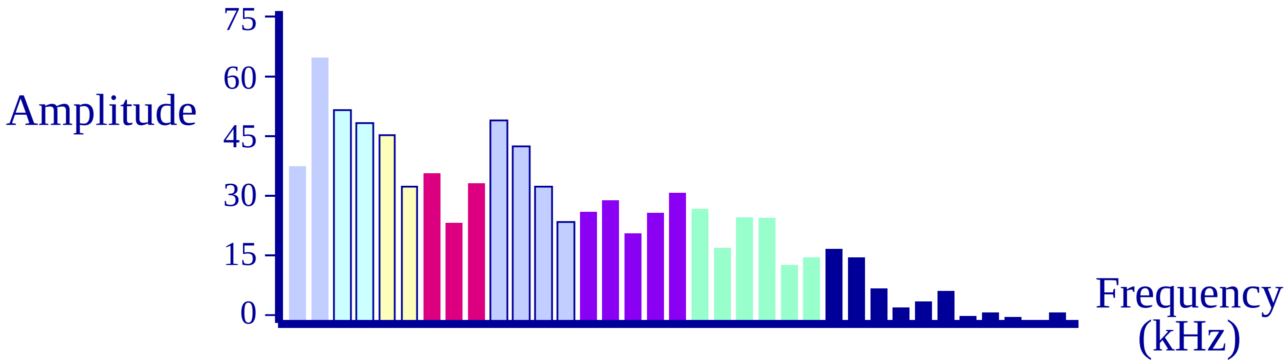
Audio-Specific Compression Techniques

Auditory masking



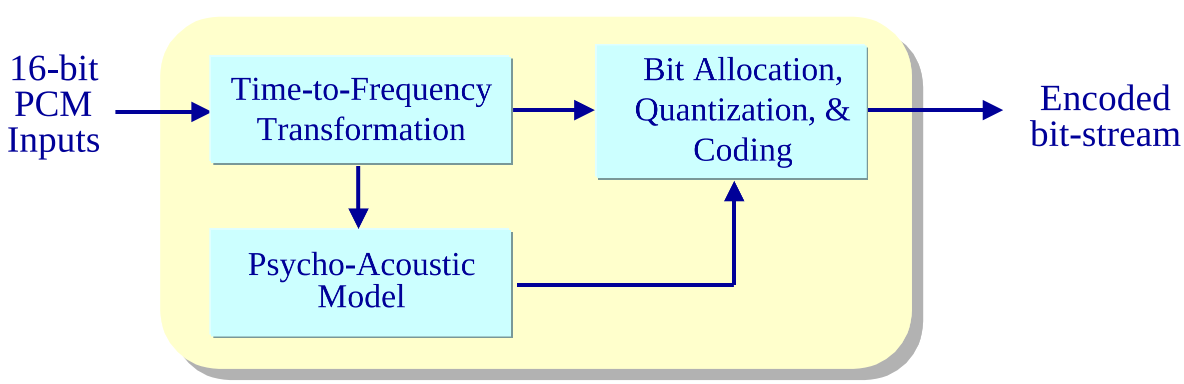
- The presence of tones at certain frequencies makes us unable to perceive tones at other "nearby" frequencies
  - » Humans cannot distinguish between tones within 100 Hz at low frequencies and 4 kHz at high frequencies

Audio-Specific Compression Techniques
Subband coding



- Code signal based on degree of masking present in the signal
- Example: MPEG audio compression (Layer I)
  - » Stereo 16-bit audio sampled at 48 kHz (1.5 Mbps) compressed to 256 kbps (6:1)
  - » Perceptually lossless

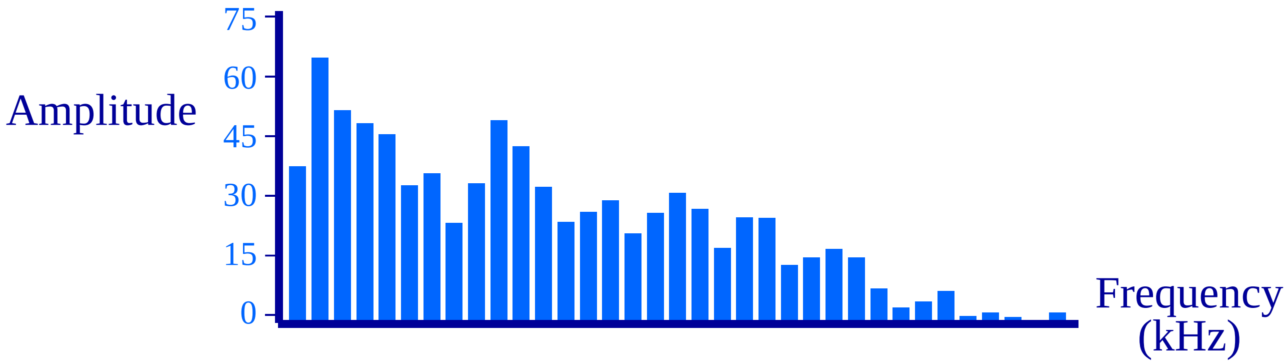
Subband Coding
MPEG Audio



- Transform signal into the frequency domain
- Partition spectrum into 32 uniform subbands
- Sample from each subband according to a psycho-acoustic model
  - » Quantize samples according to the audibility of noise within the band & degree to which the band is masked by other bands

Audio-Specific Compression Techniques

Human speech-based coding (“Vo-coding”)



- Formant — frequency maxima & minima in the spectrum of the speech signal

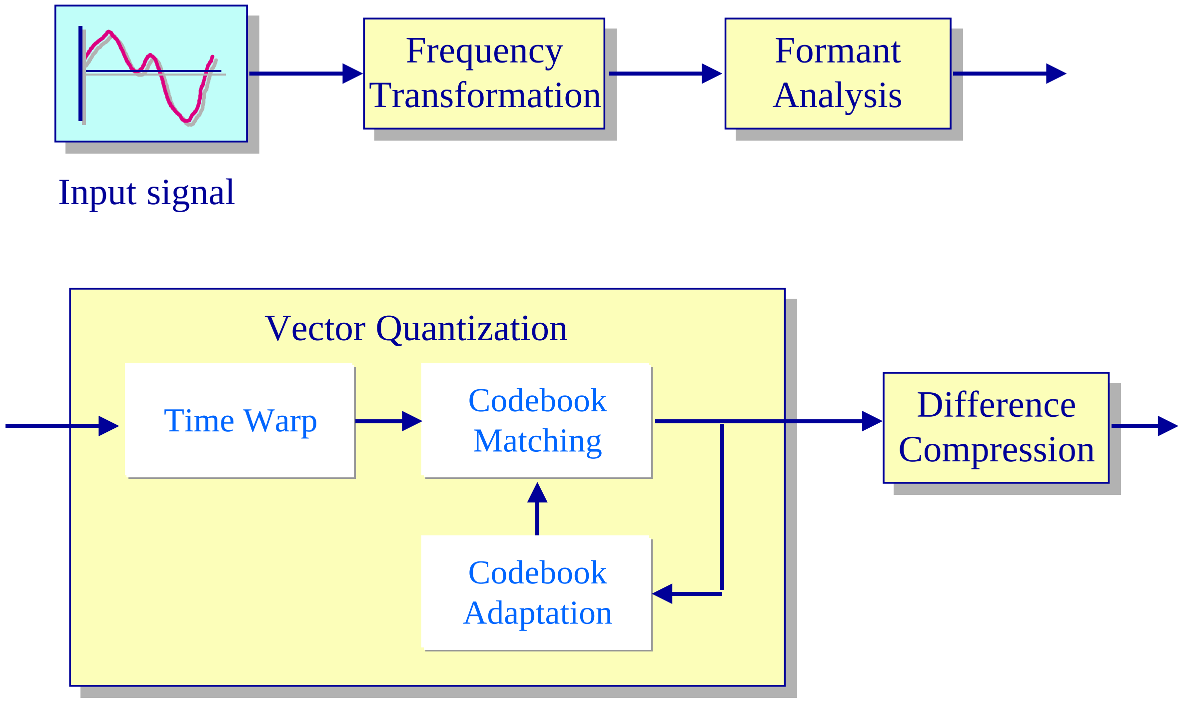
Audio-Specific Compression Techniques

Vocoding — Linear Predictive Coding

- Artificially generate speech via formant synthesis
  - A mathematical simulation of the vocal tract as a series of bandpass filters
  - Code and transmit filter coefficients

- Standards
  - Regular Pulse Excited Linear Predictive Coder (RPE-LPC)
    - digital cellular standard GSM 6.1 (13 kbps)
  - Code Excited Linear Predictive Coder (CELP)
    - US Federal Standard 1016 (4.8 kbps)
  - Linear Predictive Coder (LPC)
    - US Federal Standard 1015 (2.4 kbps)

Vocoding Example
Code Excited Linear Predictive Coder (CELP)



The Audio Data Type
Summary

- Audio is more important than video
  - » And fundamentally less structured
- Generic compression principles
  - » Elimination of temporal redundancy
  - » Exploit commonly occurring patterns
- Audio-specific techniques
  - » Exploit human's insensitivity to changes in amplitude
  - » Exploit auditory masking
- Conclusions
  - » Never make quality assessments based on bit-rate alone!