ECE160 / CMPS182
Multimedia

Lecture 7: Spring 2007

Lossless Compression Algorithms

Quantization and Transmission of Audio

- **Coding of Audio**: Quantization and transformation of data are collectively known as **coding** of the data.

  a) For audio, the μ-law technique for companding audio signals is usually combined with an algorithm that exploits the temporal redundancy present in audio signals.

  b) Differences in signals between the present and a past time can reduce the size of signal values and also concentrate the histogram of pixel values (differences, now) into a much smaller range.

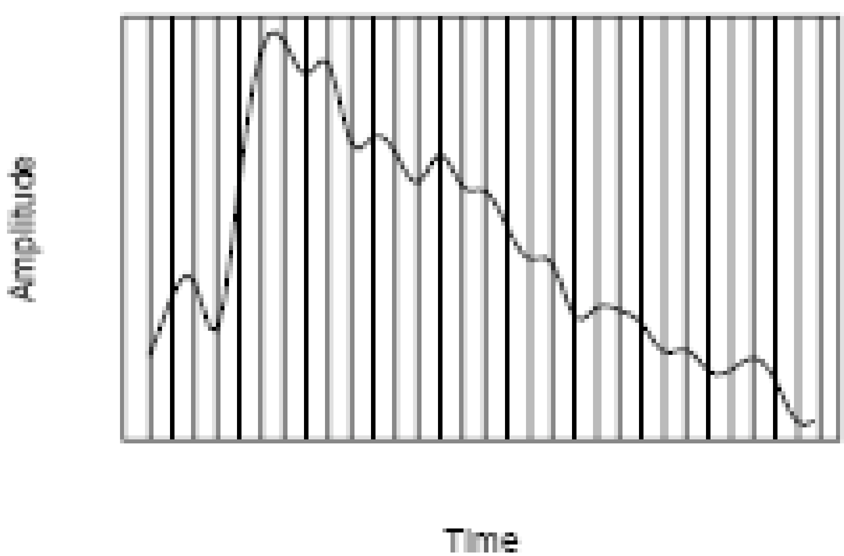
Quantization and Transmission of Audio

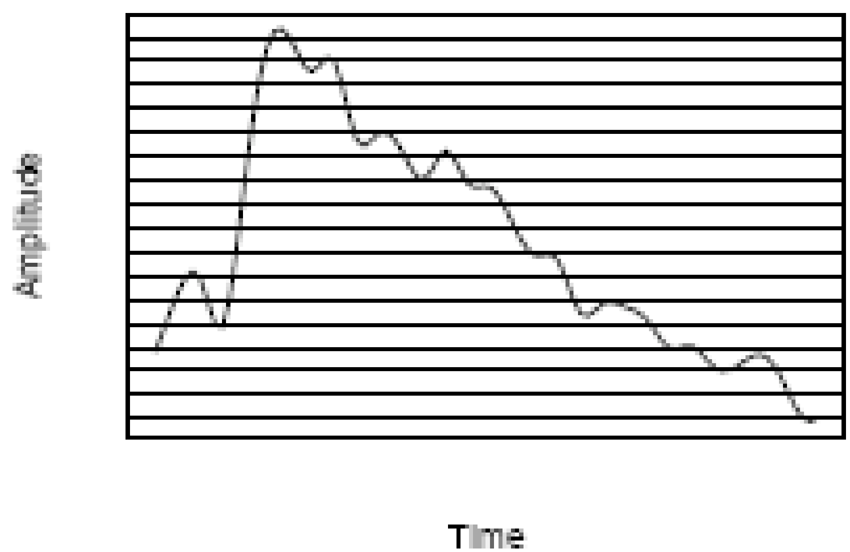
c) The result of reducing the variance of values is that lossless compression methods produce a bitstream with shorter bit lengths for more likely values

- In general, producing quantized sampled output for audio is called **PCM** (Pulse Code Modulation). The differences version is called **DPCM** (and a crude but efficient variant is called **DM**). The adaptive version is called **ADPCM**.

Pulse Code Modulation

- The basic techniques for creating digital signals from analog signals are **sampling** and **quantization**.
- Quantization consists of selecting breakpoints in magnitude, and then re-mapping any value within an interval to one of the representative output levels.





Pulse Code Modulation

a) The set of interval boundaries are called **decision boundaries**, and the representative values are called **reconstruction levels**.

b) The boundaries for quantizer input intervals that will all be mapped into the same output level form a **coder mapping**.

c) The representative values that are the output values from a quantizer are a **decoder mapping**.

- d) Finally, we may wish to **compress** the data, by assigning a bit stream that uses fewer bits for the most prevalent signal values

Pulse Code Modulation

Every compression scheme has three stages:

A. The input data is **transformed** to a new representation that is easier or more efficient to compress.

B. We may introduce **loss** of information. *Quantization* is the main lossy step => we use a limited number of reconstruction levels, fewer than in the original signal.

C. **Coding**. Assign a codeword (thus forming a binary bitstream) to each output level or symbol. This could be a fixed-length code, or a variable length code such as Huffman coding

PCM in Speech Compression

Assuming a bandwidth for speech from about 50 Hz to about 10 kHz, the Nyquist rate would dictate a sampling rate of 20 kHz.

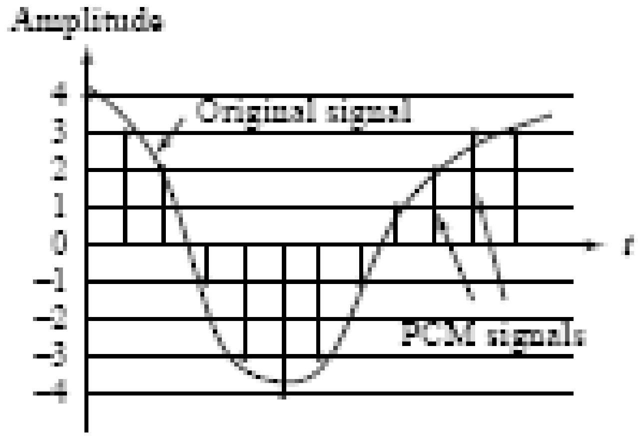
(a) Using uniform quantization without companding, the minimum sample size we could get away with would likely be about 12 bits. For mono speech transmission the bit-rate would be 240 kbps.

(b) With companding, we can reduce the sample size down to about 8 bits with the same perceived level of quality, and thus reduce the bit-rate to 160 kbps.

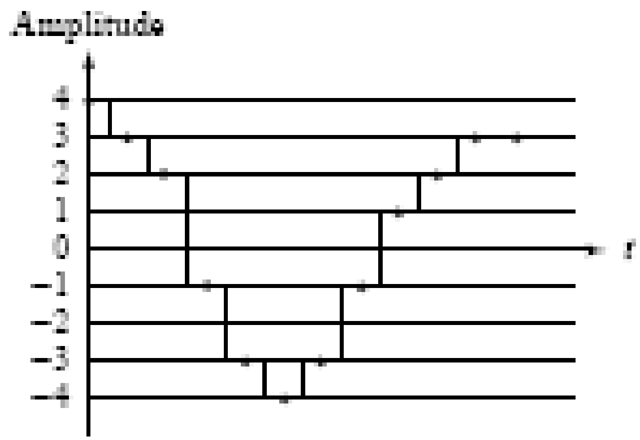
(c) However, the standard approach to telephony in fact assumes that the highest-frequency audio signal we want to reproduce is only about 4 kHz. Therefore the sampling rate is only 8 kHz, and the companded bit-rate thus reduces this to 64 kbps.

(d) Since only sounds up to 4 kHz are to be considered, all other frequency content must be noise. Therefore, we remove this high-frequency content from the analog input signal using a band-limiting filter that blocks out high, as well as very low, frequencies.

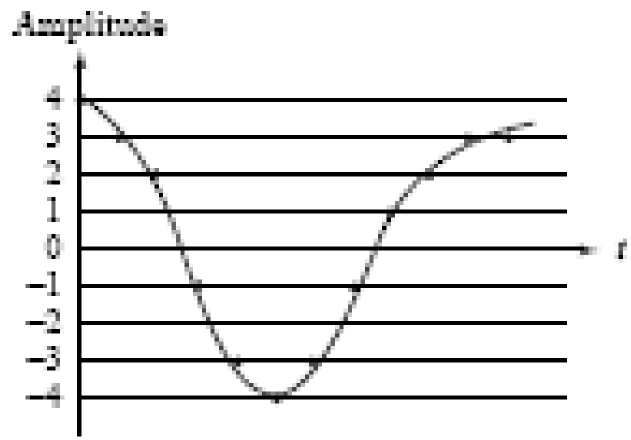
PCM in Speech Compression



(a)



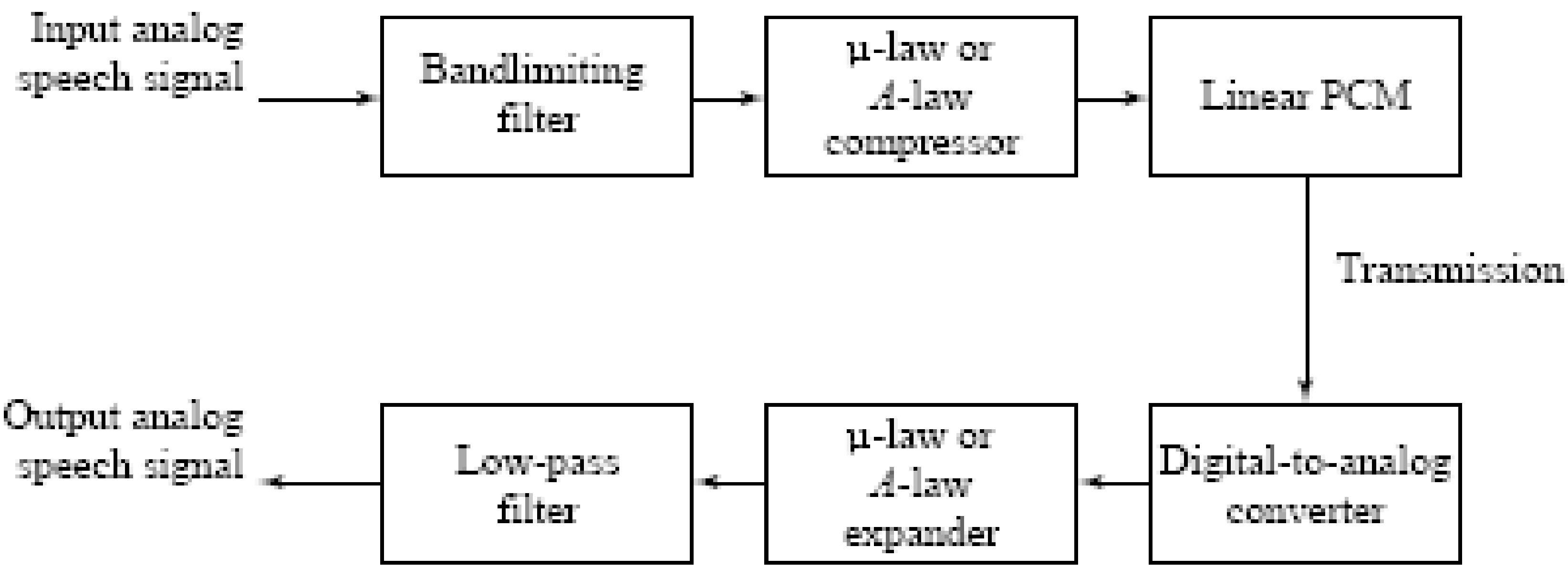
(b)



(c)

PCM in Speech Compression

- The complete scheme for encoding and decoding telephony signals is shown below. As a result of the low-pass filtering, the output becomes smoothed.



Differential Coding of Audio

- Audio is often stored in a form that exploits differences - which are generally smaller numbers, needing fewer bits to store them.

  (a) If a signal has some consistency over time ("temporal redundancy"), the difference signal, subtracting the current sample from the previous one, will have a more peaked histogram, with a maximum near zero.

  (b) For example, as an extreme case the histogram for a linear ramp signal that has constant slope is flat, whereas the histogram for the derivative of the signal (i.e., the differences from sampling point to sampling point) consists of a spike at the slope value.

  (c) If we assign codewords to differences, we can assign short codes to prevalent values and long codewords to rare ones.

Lossless Predictive Coding

- **Predictive coding**: simply means transmitting differences - predict the next sample as being equal to the current sample; send not the sample itself but the difference between previous and next.

  (a) Predictive coding consists of finding differences, and transmitting these using a PCM system.

  (b) Note that differences of integers will be integers. Denote the integer input signal as the set of values f_n. Then we **predict** values $\hat{f}_n$ as simply the previous value, and define the error e_n as the difference between the actual and the predicted signal:

$$\hat{f}_n = f_{n-1}$$

$$e_n = f_n - \hat{f}_n$$

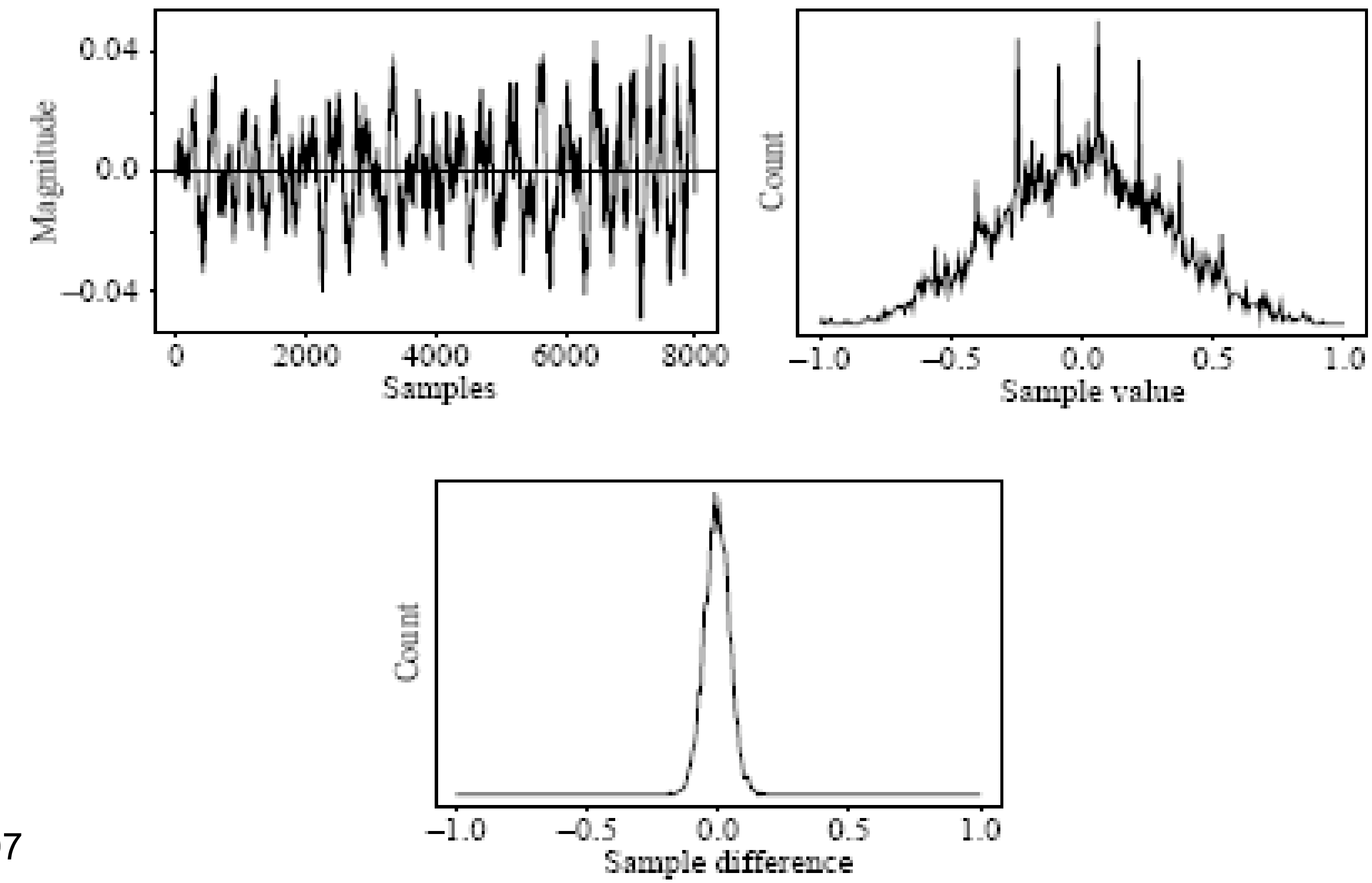
Lossless Predictive Coding

- (c) But it is often the case that some **function** of a few of the previous values, f_{n-1}, f_{n-2}, f_{n-3}, etc., provides a better prediction. Typically, a linear **predictor** function is used:

$$\widehat{f_n} = \sum_{k=1}^{2 \text{ to } 4} a_{n-k} f_{n-k}$$

Differential Coding of Audio

- Differencing concentrates the histogram.
  (a): Digital speech signal.
  (b): Histogram of digital speech signal values.
  (c): Histogram of digital speech signal differences.



Differential Coding of Audio

- One problem: suppose our integer sample values are in the range 0..255. Then differences could be as much as -255..255 - we've increased our **dynamic range** (ratio of maximum to minimum) by a factor of two and need more bits to transmit some differences.

  (a) A clever solution for this: define two new codes, denoted SU and SD, standing for Shift-Up and Shift-Down. Special code values are reserved for these.

  (b) We use codewords for only a limited set of signal differences, say only the range $-15..16$. Differences in the range are coded as is, a value outside the range $-15..16$ is transmitted as a series of shifts, followed by a value that is inside the range $-15..16$.

  (c) For example, 100 is transmitted as: SU, SU, SU, 4.

Lossless predictive coding

- *Lossless predictive coding* - the decoder produces the same signals as the original. As a simple example, suppose we devise a predictor for $\hat{f}_n$ as follows:

$$\hat{f}_n = \lfloor \frac{1}{2}(f_{n-1} + f_{n-2}) \rfloor$$

$$e_n = f_n - \hat{f}_n$$

Lossless predictive coding

- Let's consider an explicit example. Suppose we wish to code the sequence $f_1; f_2; f_3; f_4; f_5$ = 21, 22, 27, 25, 22. For the purposes of the predictor, we'll invent an extra signal value f_0, equal to f_1 = 21, and first transmit this initial value. uncoded:

$$\widehat{f_2} = 21 \, , \, e_2 = 22 - 21 = 1;$$

$$\widehat{f_3} = \lfloor \tfrac{1}{2}(f_2 + f_1) \rfloor = \lfloor \tfrac{1}{2}(22 + 21) \rfloor = 21,$$
$$e_3 = 27 - 21 = 6;$$

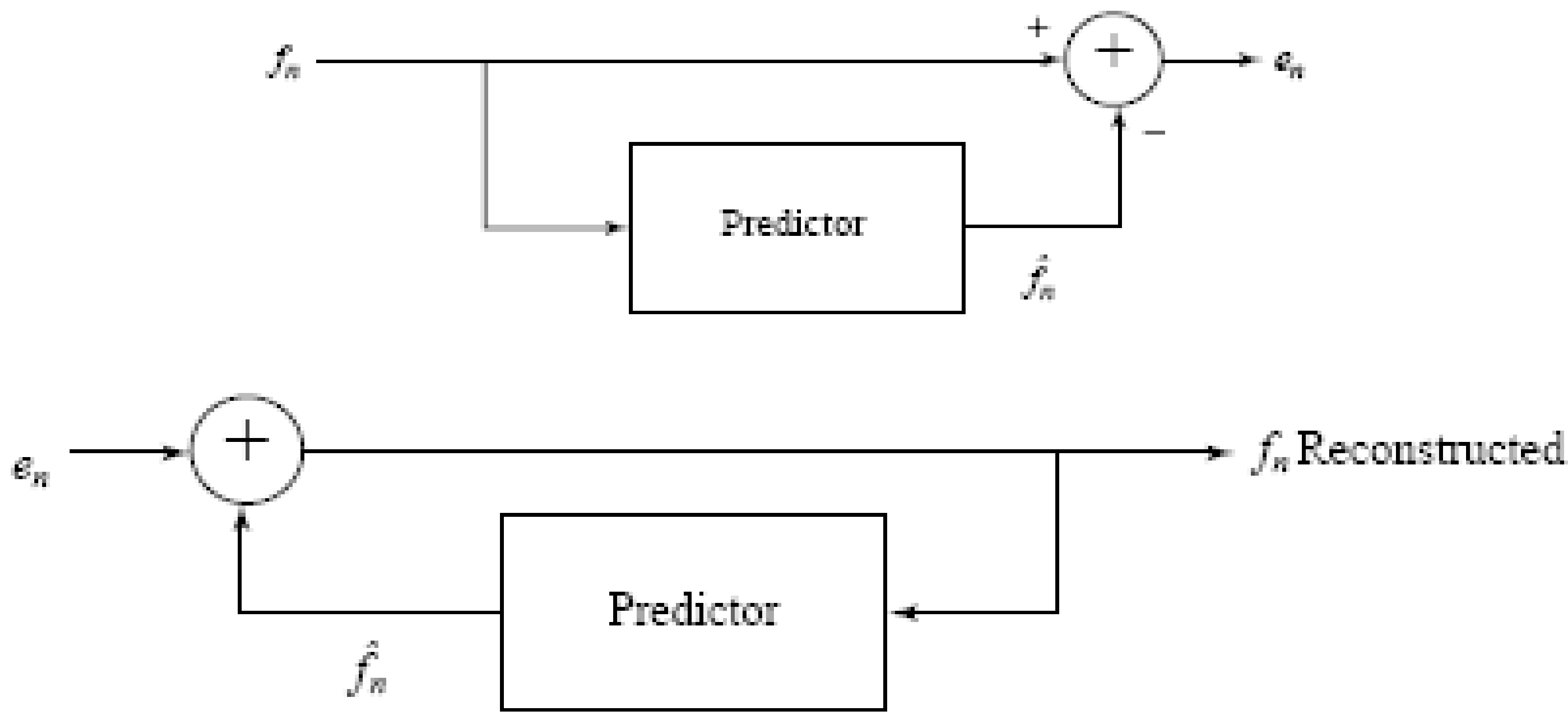
$$\widehat{f_4} = \lfloor \tfrac{1}{2}(f_3 + f_2) \rfloor = \lfloor \tfrac{1}{2}(27 + 22) \rfloor = 24,$$
$$e_4 = 25 - 24 = 1;$$

$$\widehat{f_5} = \lfloor \tfrac{1}{2}(f_4 + f_3) \rfloor = \lfloor \tfrac{1}{2}(25 + 27) \rfloor = 26,$$

$$e_5 = 22 - 26 = -4$$

Lossless predictive coding

- Schematic diagram for Predictive Coding encoder and decoder.



DPCM

- Differential PCM is exactly the same as Predictive Coding, except that it incorporates a *quantizer* step.

  (a) One scheme for analytically determining the best set of quantizer steps, for a non-uniform quantizer, is the **Lloyd-Max** quantizer, which is based on a least-squares minimization of the error term.

  (b) Our nomenclature: signal values:
  f_n - the original signal, $\hat{f}_n$ – the predicted signal, and
  $\tilde{f}_n$ the quantized, reconstructed signal.

  (c) **DPCM**: form the prediction; form an error e_n by subtracting the prediction from the actual signal; then quantize the error to a quantized version, $\tilde{e}_n$.

DPCM

- The set of equations that describe DPCM are

$$\hat{f}_n = function\_of(\tilde{f}_{n-1}, , \tilde{f}_{n-2}, \tilde{f}_{n-3}, \ldots),$$

$$e_n = f_n - \hat{f}_n,$$

$$\tilde{e}_n = Q[e_n],$$

$$\text{transmit } codeword(\tilde{e}_n),$$

$$\text{reconstruct: } \tilde{f}_n = \hat{f}_n + \tilde{e}_n.$$

- Then *codewords* for quantized error values $\tilde{e}_n$ are produced using entropy coding,

DPCM

- The main effect of the coder-decoder process is to produce reconstructed, quantized signal values $\tilde{f}_n = \hat{f}_n + \tilde{e}_n$.
  The **distortion** is the average squared error $[\sum_{n=1}^{N} (\tilde{f}_n - f_n)^2]/N$; one often plots distortion versus the number of bit-levels used. A Lloyd-Max quantizer will do better (have less distortion) than a uniform quantizer.
- For speech, we modify quantization steps adaptively by estimating the mean and variance of a patch of signal values, and shifting quantization steps accordingly for every block of signal values. Starting at time *i* we take a block of *N* values f_n and minimize the quantization error:

$$min \sum_{n=i}^{i+N-1} (f_n - Q[f_n])^2$$

DPCM

- Since signal **differences** are very peaked, we can model them using a Laplacian probability distribution function, which is strongly peaked at zero: it looks like

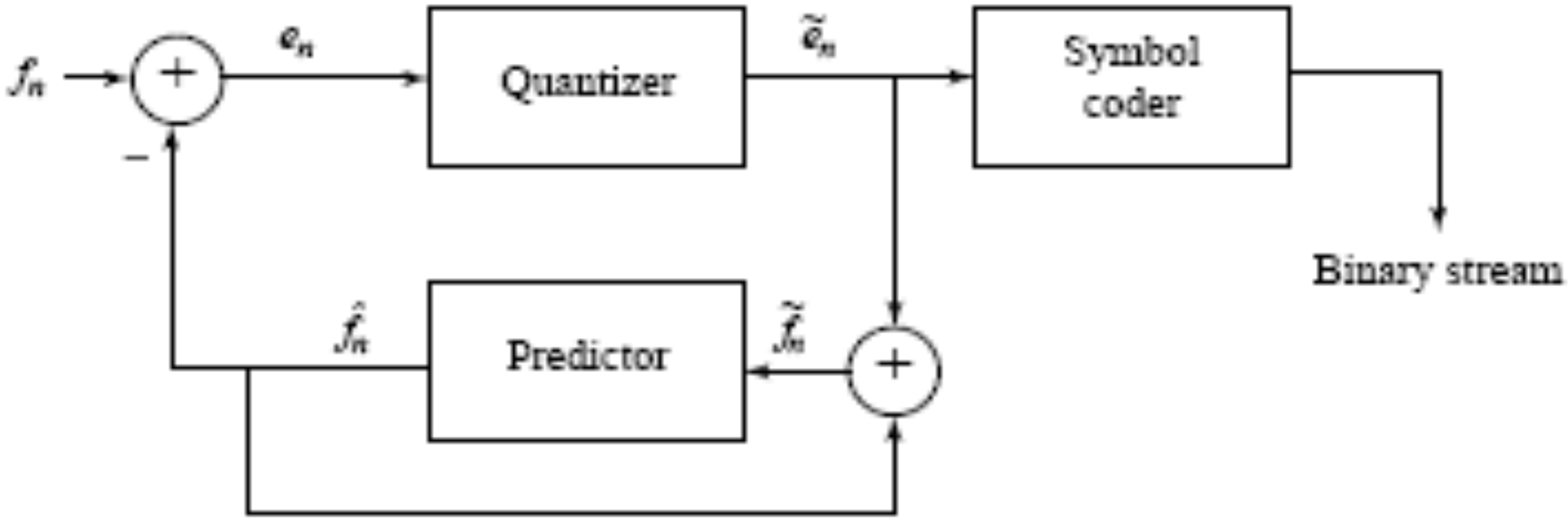
$$l(x) = (1/\sqrt{2\sigma^2})exp(-\sqrt{2}|x|/\sigma)$$

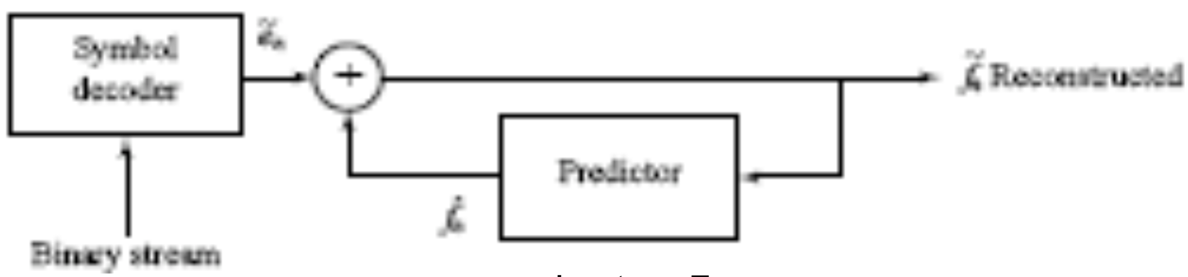
  for variance σ^2.
- So typically one assigns quantization steps for a quantizer with nonuniform steps by assuming signal differences, d_n are drawn from such a distribution and then choose steps to minimize

$$min \sum_{n=i}^{i+N-1} (d_n - Q[d_n])^2 \, l(d_n)$$

DPCM

- Schematic diagram for DPCM encoder and decoder





DM

- **DM** (Delta Modulation): simplied version of DPCM. Often used as a quick AD converter.

  1. **Uniform-Delta DM**: use only a single quantized error value, either positive or negative.

  (a) => a 1-bit coder. Produces coded output that follows the original signal in a staircase fashion. The set of equations is:

$$\hat{f}_n = \tilde{f}_{n-1} ,$$
$$e_n = f_n - \hat{f}_n = f_n - \tilde{f}_{n-1} ,$$

$$\tilde{e}_n = \begin{cases} +k & \text{if } e_n > 0, \text{ where } k \text{ is a constant} \\ -k & otherwise \end{cases}$$

$$\tilde{f}_n = \hat{f}_n + \tilde{e}_n .$$

DM

- DM does not cope well with rapidly changing signals. One approach to mitigating this problem is to simply increase the sampling, perhaps to many times the Nyquist rate.
- **Adaptive DM**: If the slope of the actual signal curve is high, the staircase approximation cannot keep up. For a steep curve, can change the step size *k* adaptively.

  - One scheme for analytically determining the best set of quantizer steps, for a non-uniform quantizer, is *Lloyd-Max*.

ADPCM

- **ADPCM** (Adaptive DPCM) takes the idea of adapting the coder to suit the input much farther. The two pieces that make up a DPCM coder: the quantizer and the predictor.

  1. In Adaptive DM, adapt the quantizer step size to suit the input. In DPCM, we can change the step size as well as decision boundaries, using a non-uniform quantizer.

  We can carry this out in two ways:

  (a) **Forward adaptive quantization**: use the properties of the input signal.

  (b) **Backward adaptive quantizationor**: use the properties of the quantized output. If quantized errors become too large, we should change the non-uniform quantizer.

ADPCM

- We can also **adapt the predictor**, again using forward or backward adaptation. Making the predictor coefficients adaptive is called *Adaptive Predictive Coding* (APC):

  (a) Recall that the predictor is usually taken to be a linear function of previous reconstructed quantized values, $\tilde{f}_n$.

  (b) The number of previous values used is called the “order" of the predictor. For example, if we use M previous values, we need M coefficients a_i , i = 1..M in a predictor

$$\hat{f}_n = \sum_{i=1}^{M} a_i \tilde{f}_{n-i}$$

ADPCM

- However we can get into a difficult situation if we try to change the prediction coefficients, that multiply previous quantized values, because that makes a complicated set of equations to solve for these coefficients:

  (a) Suppose we decide to use a least-squares approach to solving a minimization trying to find the best values of the a_i:

$$min \sum_{n=1}^{N} (f_n - \hat{f}_n)^2$$

- Here we would sum over a large number of samples *fn*, for the current patch of speech, say. But because $\tilde{f}_n$ depends on the quantization we have a difficult problem to solve. As well, we should really be changing the fineness of the quantization at the same time, to suit the signal's changing nature; this makes things problematical.

ADPCM

- Instead, one usually resorts to solving the simpler problem that results from using not $\tilde{f}_n$ in the prediction, but instead simply the signal f_n itself. Explicitly writing in terms of the coefficients *ai*, we wish to solve:

$$min \sum_{n=1}^{N} (f_n - \sum_{i=1}^{M} a_i f_{n-i})^2$$

- Differentiation with respect to each of the a_i, and setting to zero, produces a linear system of M equations that is easy to solve. (The set of equations is called the Wiener-Hopf equations.)

Dolby

- Dolby started by analog coding to hide hiss on cassette tapes. He amplified small sounds before recording and then reduced them (and the hiss) on playback.
- Dolby moved on to multichannel sound for movie widescreens (but no multichannels on the movie film).
- You will encounter Dolby AC-3, (5.1 channels), right front, center, left front, right rear, left rear and subwoofer. Dolby EX (6.1) has a center rear Dolby TrueHD has 8 channels and high quality

ECE160 / CMPS182

Multimedia

Lecture 7: Spring 2007

Lossy Compression Algorithms

Introduction

- For lossless compression algorithms, see the text book – they are easy.
- Lossless compression algorithms do not deliver *compression ratios* that are high enough. Hence, most multimedia compression algorithms are *lossy*.
- What is *lossy compression* ?
  - The compressed data is not the same as the original data, but a close approximation of it.
  - Yields a much higher compression ratio than that of lossless compression.

Distortion Measures

- The three most commonly used distortion measures in image compression are:
- *mean square error* (MSE) σ^2,

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2$$

where x_n, y_n, and N are the input data sequence, reconstructed data sequence, and length of the data sequence respectively.

- *signal to noise ratio* (SNR), in decibel units (dB),

$$SNR = 10\log_{10}\frac{\sigma_x^2}{\sigma_d^2}$$

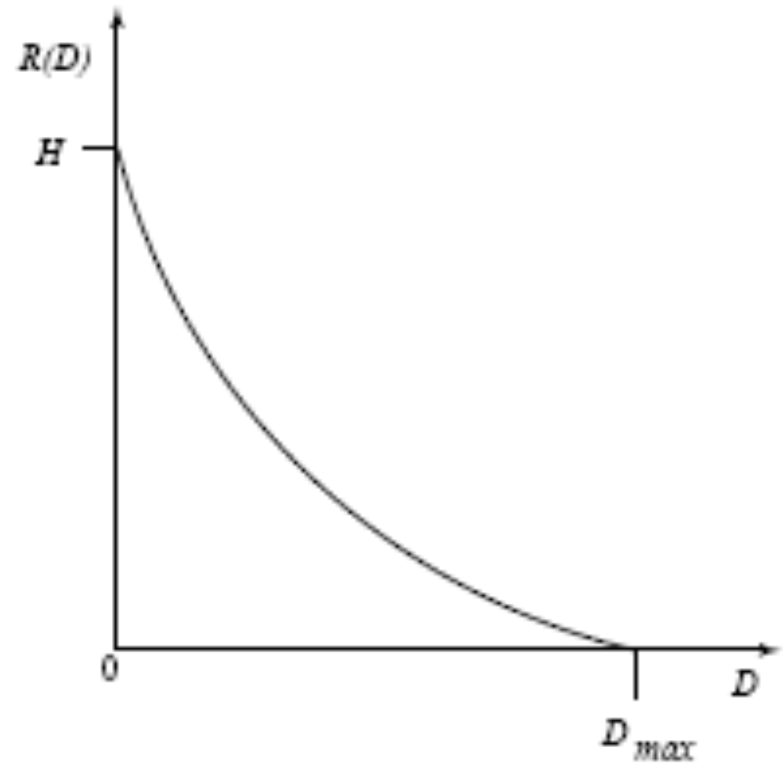
where σ^2 is the average square value of the original data sequence and σ^2 is the MSE.

- *peak signal to noise ratio* (PSNR),

$$PSNR = 10\log_{10}\frac{x_{peak}^2}{\sigma_d^2}$$

Rate-Distortion Theory

- Provides a framework for the study of tradeoffs between Rate and Distortion. A Typical Rate Distortion Function:



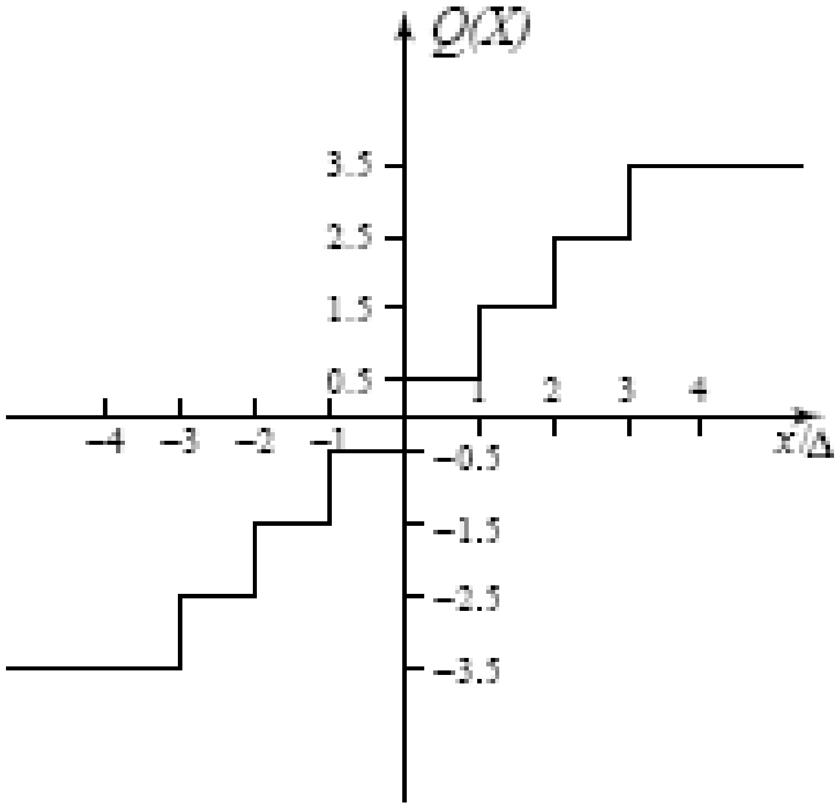
Quantization

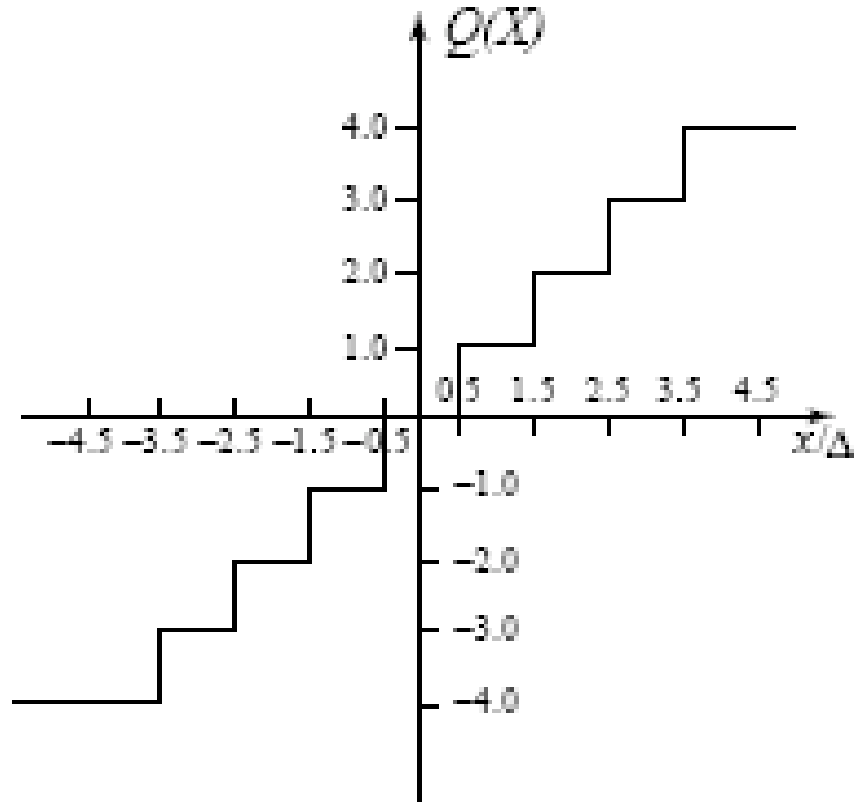
- Reduce the number of distinct output values to a much smaller set.
- Main source of the “loss" in lossy compression.
- Three different forms of quantization.
  - Uniform: midrise and midtread quantizers.
  - Nonuniform: companded quantizer.
  - Vector Quantization.

Uniform Scalar Quantization

- A uniform scalar quantizer partitions the domain of input values into equally spaced intervals, except possibly at the two outer intervals.
  - The output or reconstruction value corresponding to each interval is taken to be the midpoint of the interval.
  - The length of each interval is referred to as the *step size*, denoted by the symbol Δ.
- Two types of uniform scalar quantizers:
  - Midrise quantizers have even number of output levels.
  - Midtread quantizers have odd number of output levels, including zero as one of them

Midrise and Midtread





Uniform Scalar Quantization

- For the special case where $\Delta = 1$, we can simply compute the output values for these quantizers as:

$$Q_{midrise}(x) = \lceil x \rceil - 0{:}5$$
$$Q_{midtread}(x) = \lfloor x + 0{:}5 \rfloor$$

- Performance of an M level quantizer. Let $B = \{b_0, b_1, \ldots, b_M\}$ be the set of decision boundaries and $Y = \{y_1, y_2, \ldots, y_M\}$ be the set of reconstruction or output values.
- Suppose the input is uniformly distributed in the interval $[-X_{max}, X_{max}]$. The rate of the quantizer is:

$$R = \lceil \log_2 M \rceil$$

Companded quantization

- *Companded quantization* is **nonlinear**.
- A *compander* consists of a *compressor function G*, a uniform quantizer, and an *expander function G*−1.
- The two commonly used companders are the μ-law and *A*-law companders.