Multimedia Systems

Part 17

Mahdi Vasighi
www.iasbs.ac.ir/~vasighi

Department of Computer Science and Information Technology,
Institute for Advanced Studies in Basic Sciences, Zanjan, Iran

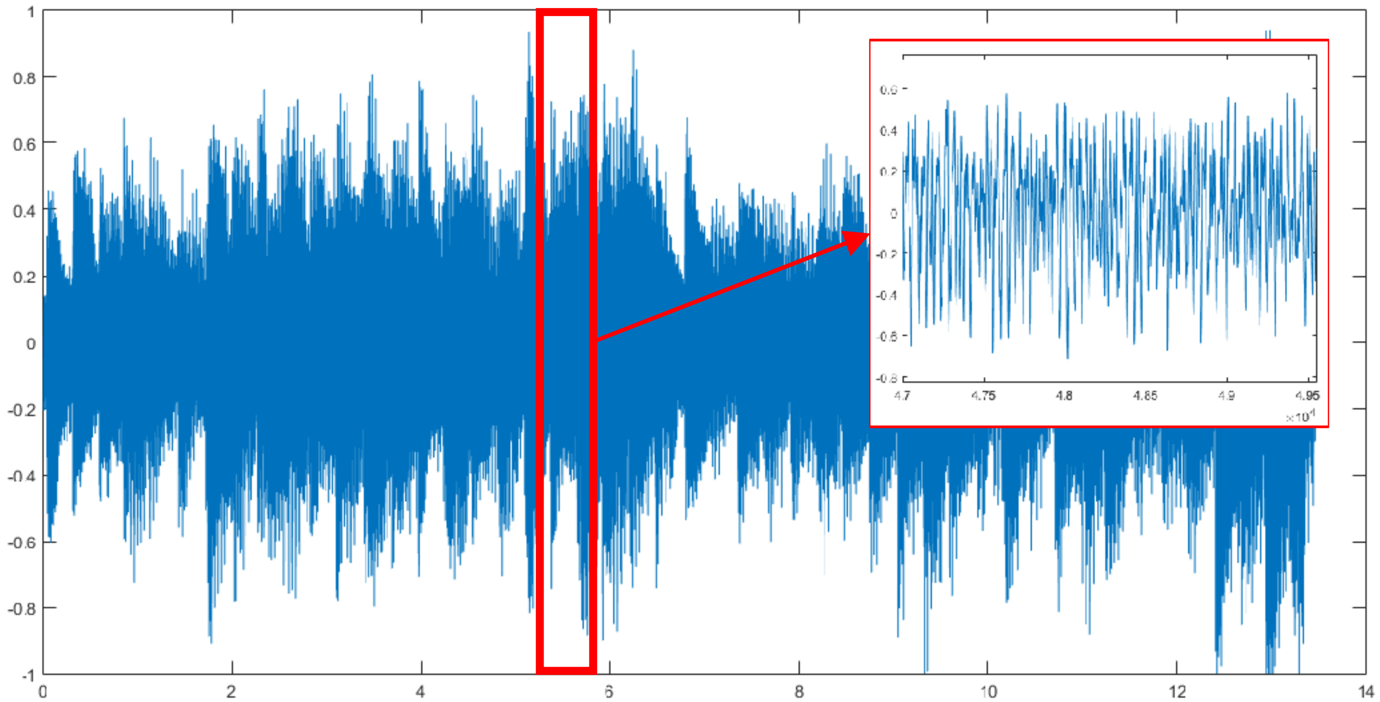
Coding of Audio

Quantization and transformation of data are collectively known as **coding** of the data

- 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.

- **Differences** in signals between the present and a past time can reduce the size of signal values

Coding of Audio

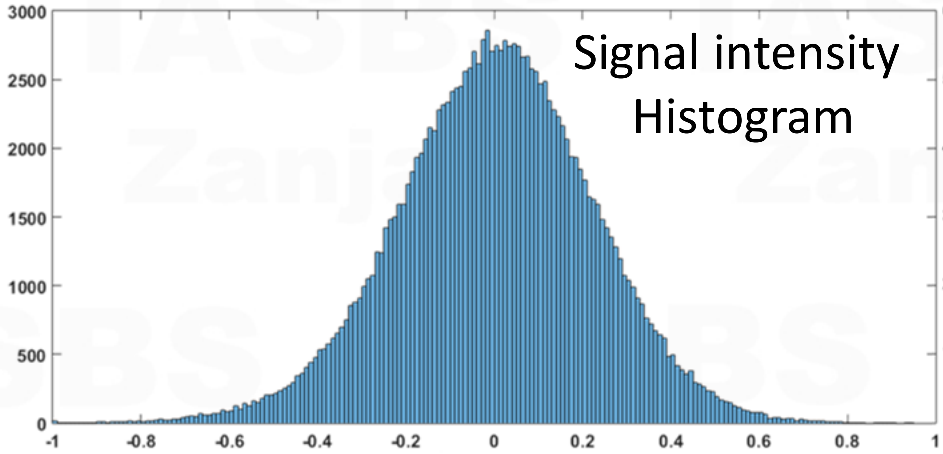


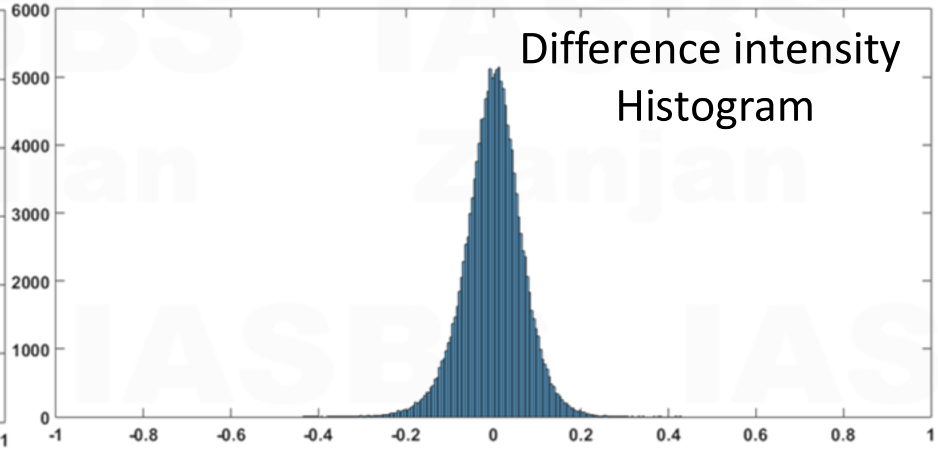
Audio signal

13.4965 seconds
11025 Hz

```
[y,Fs] = audioread('acoustic.wav');
plot((0:length(y)-1)/Fs,y)
Sound(y,Fs)
```

Coding of Audio



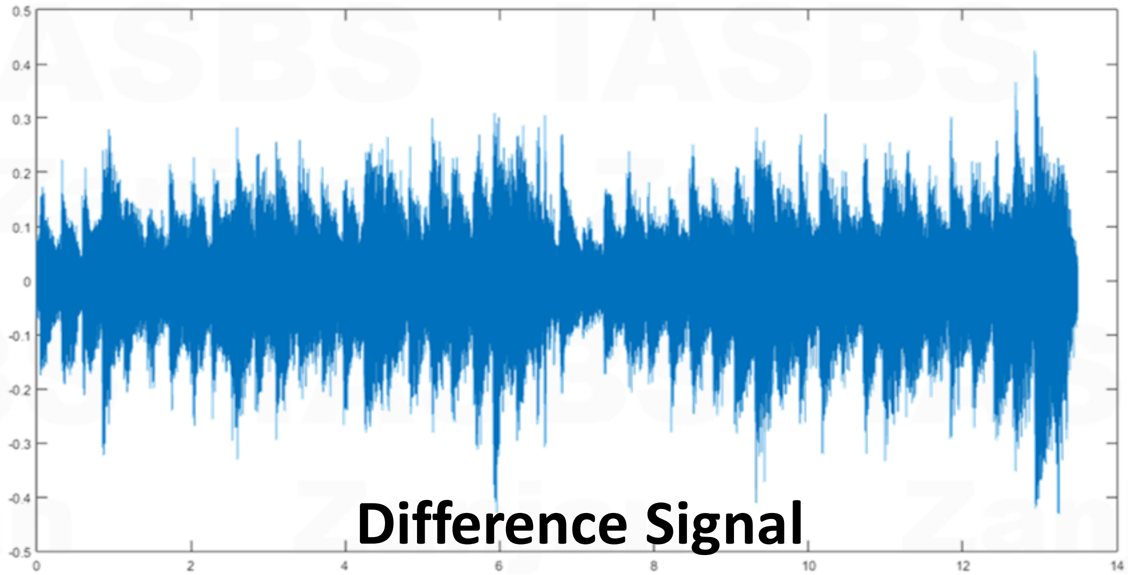



```
histogram(y);
xlim([-1 1])
```

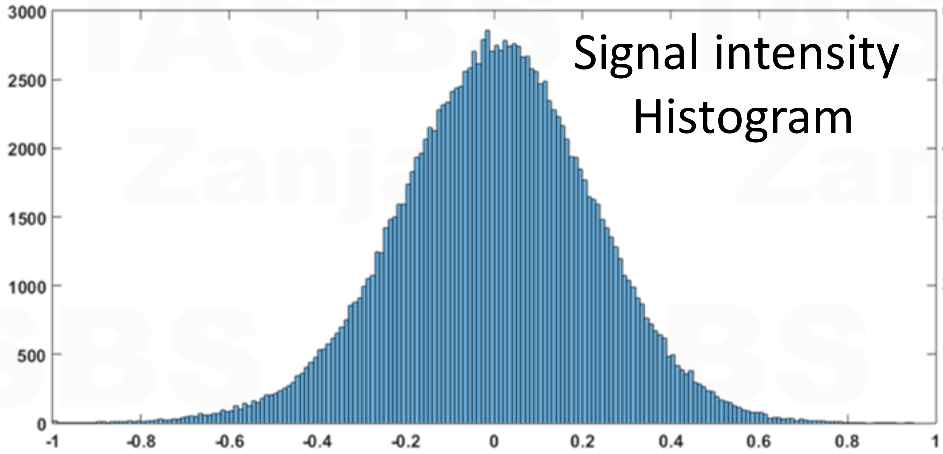
```
dy=diff(y);
```

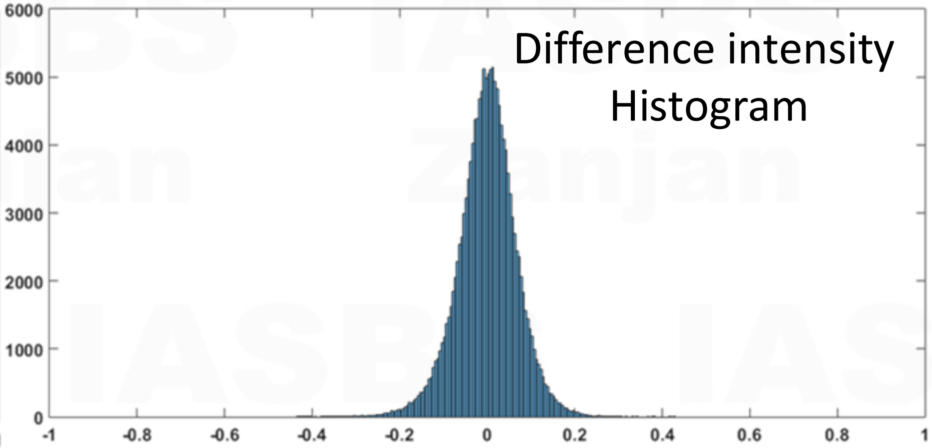
```
histogram(dy);
```

dY = [y(2)-y(1), y(3)-y(2), … ,y(m)-y(m-1)]



Coding of Audio





Differences in signals can

- Effectively reduce the size of signal values
- concentrate the histogram of differences into a much smaller range

- By reducing the variance of values, the lossless compression methods produce a bitstream with shorter bit lengths for more likely values

Coding of Audio

In general, producing quantized sampled output for audio is called **PCM** (Pulse Code Modulation).

The differences version is called

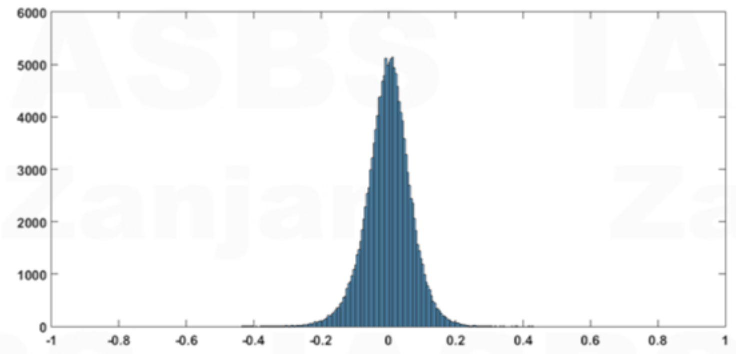
Differential Pulse Code Modulation

DPCM

Coding of Audio

Signal consistency over time
(Temporal redundancy)

more peaked histogram
maximum around zero



if we then go on to assign bitstring codewords to differences, we can assign short codes to prevalent values and long codewords to rarely occurring ones.

Coding of Audio

Consider the signal consists only four symbols s_1, s_2, s_3, s_4.

Standard (fixed-length) coding:

- we would need 2 bits/symbol
- e.g., assign the codewords 00, 01, 10, 11, respectively.
- 2 bits/symbol

Variable-length coding

- The probabilities for the symbols are 1/2 for s_1, 1/4 for s_2, and 1/8 each for s_3 and s_4.
- we assign the codes 0, 10, 110, and 111 for s_1, s_2, s_3 & s_4
- should use the probabilities of the symbols for bits/symbol
- $(1)(1/2) + (2)(1/4) + (3)(1/8) + (3)(1/8) = 1.75$ bits/symbol

Lossless Predictive Coding

Predictive coding consists of finding differences and transmitting them, using a PCM system.

- We predict the next sample as being equal to the current sample and send not the sample itself but the error involved in making this assumption.

- integer signal as the set of values f_n
- Then we predict values $\hat{f}_n$ as simply the previous value!
- We would wish our prediction to be as close as possible to the actual signal.

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

Lossless Predictive Coding

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

some function of a few of the previous values, $f_{n-1}, f_{n-2}, f_{n-3}$, etc., may provide a better prediction of f_n

$$\hat{f}_n = \sum_{k=1}^{2 \text{ to } 4} a_{n-k} f_{n-k}$$

Such a predictor can be followed by a truncating or rounding operation to result in integer values.

- Coefficients can be changed adaptively
- assigns short codewords to frequently occurring symbols

Lossless Predictive Coding

What to do if a particular set of difference values does indeed consist of some exceptional large differences?

defining two new codes to add to our list of difference values!

- Shift-Up code (SU)
- Shift-Down code (SD)

- If SU and SD = 32. only the range of −15 to 16
- a value outside the range −15 to 16 can be transmitted as a series of shifts, followed by a value that is inside the range

- 100 is transmitted as SU, SU, SU, 4

Lossless Predictive Coding

A simple example:

suppose we devise a predictor for $\hat{f}_n$ as follows:

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

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

the error e_n (or a codeword for it) is what is actually transmitted.
Suppose we wish to code the sequence

f_1, f_2, f_3, f_4, f_5 = 21, 22, 27, 25, 22.

An extra signal, **f_0** will be invented equal to **f_1**=21 and first transmitted encoded.

Lossless Predictive Coding

$f_0, f_1, f_2, f_3, f_4, f_5$ = 21, 21, 22, 27, 25, 22.

$$\hat{f}_2 = 21, \quad e_2 = 22 - 21 = 1$$

$$\hat{f}_3 = \lfloor \frac{1}{2}(f_2 + f_1) \rfloor = \lfloor \frac{1}{2}(22 + 21) \rfloor = 21$$
$$e_3 = 27 - 21 = 6$$

$$\hat{f}_4 = \lfloor \frac{1}{2}(f_3 + f_2) \rfloor = \lfloor \frac{1}{2}(27 + 22) \rfloor = 24$$
$$e_4 = 25 - 24 = 1$$

$$\hat{f}_5 = \lfloor \frac{1}{2}(f_4 + f_3) \rfloor = \lfloor \frac{1}{2}(25 + 27) \rfloor = 26$$
$$e_5 = 22 - 26 = -4$$

Lossless Predictive Coding

the predictor is based on f_{n-1}, f_{n-2}, … . Therefore, the predictor must involve a memory.