2. GOALS OF DATA COMPRESSION

General Objective in data compression:

Given a signal and a channel, find an encoder and decoder that give the "best possible" replica of the signal at the user end or the most reliable recommendation to an action, such as accept a person to a secure area from his/her finger print or deny.

To formulate the task of data compression as a precise problem, we need:

- probabilistic descriptions of signal and channel (parametric model or sample training data),
- possible structural constraints on form of codes (block, sliding block, etc.),
- Quantifiable notion of what "good" or "bad" reconstruction is in the form of quantifiable measurements.

Some important questions:

Mathematical: Quantify what "best" or optimal achievable performance is.

Practical: How do you build systems that are good, if not optimal?

How to measure performance and need to define them?

- Signal-to-Noise Ration (SNR) and Segmented SNR (SegSNR)
- Mean-Square Error (MSE)
- Probability of bit error (*Pe*)
- Information transmission or storage bit rates,
- Complexity of the system designed,
- Cost of coding or design.
- Subjective measures, such as Mean Opinion Scores (MOS).

Shannon Theory

Two publications by C.E. Shannon, "A Mathematical Theory of Communication (1948)" and "Coding Theorems for a Discrete Source with a Fidelity Criterion (1957)" have provided basis for application of probability theory to solving problems in communications in addition to establishing the foundations of the Science of Modern Information Theory.

The primary goal of Shannon's Information Theory is to find the theoretical limits to achievable performance. That is, given a probabilistic model of an information source and a communication channel (for storage or transmission), how reliably can the source signals be communicated to a receiver though the given channel?

Note that it does not tell you how to do it?
It leaves it to smart engineers to come up with good designs!

Key to Shannon's formulation was a family of definitions of the ***information*** content of signals in terms of probabilities. These are:

- Entropy $H(X)$ of a random variable X.

- Entropy rate $\overline{H}(X)$ of a sequence of random numbers (random process) $\{X_n\}$
- Mutual Information $I(X;Y)$ between two random variables X and Y.
- Notion of a ***distortion*** (error) measure $d(X,\hat{X})$ representing the cost of reproducing a signal $\hat{X}$ at the receiver (decoder) instead of a signal X of the source (encoder) so that the average distortion provides a measure of average signal quality and thus the system performance.

Definitions:

Entropy:

- Assume a source S that can generate any of *L* possible messages.
- Let each of these messages has a probability: $P_S = \{p_k; k = 0,1,\cdots,L-1\}$
- Entropy of this source represent the theoretically achievable lower bound on compression Systems and it is defined by:

$$H(S) = \sum_{k=0}^{L-1} p_k.\log_2(1/p_k) \qquad \text{Bits/symbol} \tag{2.1}$$

- Let us assume that a given source has *L* different messages possible (symbol)
- The k^{th} symbol occurs with probability P_k, *k= 0,1,...,L-1*.
- Consider an encoder, which assigns a codeword with l_k bits to a particular symbol s_k.
- Average length of this source coder is simply:

$$L_{ave} = \sum_{k=0}^{L-1} p_k.l_k \qquad \text{Bits/symbol} \tag{2.2}$$

Example 2.1: Consider the following 6 source symbols to be used as messages: *S*={A,B,C,D,E,F} with occurrence probabilities {0.25, 0.20, 0.16, 0.15, 0.13, 0.11}.

Note that sum of probabilities is 1.0 since there are no other messages and possibilities left. One way to represent this symbol set is to assign $\overline{L} = 3-bit$ long codewords to each symbol as it is shown in Table 2.1.

- 8 possible ways to arrange 3 bits: 000, 001,…, 111.
- Chose any 6 out of these combinations.
- Compute the entropy and average length of this coding procedure.

$$H(S) = -\sum_{k=1}^{6} p_k.\log_2(p_k) = 2.5309 \quad bit/symbol$$

- It is easy to see that the difference $\overline{L} - H(S) = 3.0 - 2.5309 = 0.4691\, bits/symbol$ is almost ½ bits per symbol away from the theoretical bound *H(S).*
- Another scheme is to use a variable-length bit assignment according to some nice rule! As also shown un the table. One such assignment procedure is known as Huffman Coding in literature, which will be discussed later.
- The average length of this code is:

$$\overline{L}_{Huffman} = 0.25x2 + 0.20x2 + 0.16x3 + 0.15x3 + 0.13x3 + 0.11x3 = 2.55 \quad bits/symbol$$

- Difference in this case is lowered to:
  $\overline{L} - H(S) = 2.55 - 2.5309 = 0.0191\, bits/symbol$

- This scheme results almost perfect result.

Message ID	Probability	Binary Code	Code Length	Huffmann Code	Code Length
A	0.25	000	3	10	2
B	0.20	001	3	00	2
C	0.16	010	3	111	3
D	0.15	011	3	110	3
E	0.13	100	3	011	3
F	0.11	101	3	010	3

Shannon Source Coding Theorem:

Given a source *S* with entropy *H(S)* then

1. It is possible to encode this source with a distortionless (error free) source coder with an average length L_{ave} provided

$$L_{ave} \geq H(S) \tag{2.3a}$$

2. Conversely, there is no distortionless source coder to encode *S* if

$$L_{ave} < H(S) \tag{2.3b}$$

Channel Capacity:

- Let a source emit symbols $S_0, S_1, \ldots, S_{L-1}$ at a rate R bits per second.
- Transmitted over a channel with a bandwidth B Hz.
- Receiver detects signals coming from the channel with a signal-to-noise ratio SNR= S/N,
- Where *S* and *N* are the signal and noise powers at the input to the receiver, respectively.
- Receiver issues symbols $Y_0, Y_1, \ldots, Y_{K-1}$. (These symbols $\{Y_k\}$ may or may not be identical to the source set $\{S_k\}$ depending upon the nature of the receiver.) Furthermore,
- *L* and *K* may be of different size. (Some codewords might be totally lost in the channel or some codewords might be added by the channel itself.)

If the channel is noiseless then

- L and K are identical and
- Receiver symbols are also same as the source symbols. In this case,
- Reception of some symbol Y_k uniquely determines the source symbol S_k.
- In the noisy channels, however, there is a certain amount of uncertainty regarding the identity of the transmitted symbol when Y_k is received.
- If the information channel has a bandwidth of B Hz. and
- System is designed to operate at a signal-to-noise ratio SNR then

The highest rate we can reliably transmit binary information is called **Capacity of a noisy channe**l and it is defined by:

$$C = B.\log_2(1 + SNR) \qquad \text{Bits/second} \tag{2.4}$$

Shannon Channel Coding Theorem:

Given a channel with a capacity C, then

1. It is possible to transmit symbols from a source emitting at a rate R bits per second with an arbitrarily small probability of error in over this channel if

$$C \geq R \tag{2.5a}$$

2. Conversely, all systems transmitting at a rate R such that

$$C < R \tag{2.5b}$$

are bound to have errors with probability one, i.e, with hundred percent certainty to have errors.

Implication: We can have good source coders, even perfect ones, if the source rate is under the channel capacity. We may not be able design such good coder but that is our problem, not the information theory's! On the other hand, all coders are destined to make errors if they operate at a rate above the channel capacity.

Combination of these theorems shows that in the problem of Point-to-Point communication, the encoder (and decoder) can be decomposed into two separate pieces:

1. Source coding is the conversion of an information source into an efficient binary (or other digital) representation of rate R bits/s, with no regard for the channel except that its capacity is $C > R$ bits/s.

2. Channel coding or error control coding, takes the binary data stream of R bits/s and sends it reliably across the channel. This result is usually called the source/channel separation theorem. This result is not true in general in network communications, e.g., many sources to one receiver (multi-user channel) or one source to many receivers (broadcast channel).

- Source coding reduces the number of bits in order to save on transmission time or storage space. Compression \removes redundancy" to gain efficiency.
- Channel coding typically increases the number of bits or chooses different bits in order to protect against channel errors. Error control "adds redundancy" to permit the detection and correction of errors by looking for violation of known structures of transmitted signals and picking a likely fix.
- Overall communication requires a balance of the above two effects.

1. Consider a bandlimited channel (Bandwidth= W Hz) is used for transmitting signal with a total power S watts.
2. Assume that this channel is subject to additive Gaussian noise with a total noise power $N = 2W.N_0/2 = W.N_0$ watts. Here the noise level (density) is constant in the band of interest as $N_0/2$ watts/Hz.

The capacity of this channel is given by:

$$C = W.\log_2(1 + \frac{S}{W.N_0}) \text{ Bits/second} \tag{2.6}$$

Fundamental Question: What does all this theory have to do with the real world?

- The theorem supports the intuitive practice that good overall systems can be designed by *separately* focusing on the source coding (compression) and channel coding (error control) problems, without concern of the interaction between the two.
- Joint coders are of increasing interest in networks and because they can yield simpler implementations.
- Theory provides benchmarks for comparison for real communication systems. It is impossible to do better than Shannon's bounds (unless you cheat). Thus if you are operating close to the Shannon limit, hardly worth doing any more.
- Most systems of the time fell far short of Shannon. However, few believed that they could ever get near. Over the years, methods evolved to come close to these limits in many applications of both source coding and channel coding.
- Emphasis through first half of Shannon era was on channel coding.
- Compression has historically played a secondary role, partially because of apparently relatively small potential gains and highly nonlinear nature in comparison with error control coding methods.

Question: Why to compress?

Answers:

1.Theory: As systems get better, even small gains get more important.

2.Practice: Sometimes have no choice, must compress or lose all the data and Gains are sometimes significant.

Consider the following simple image compression tasks:

- Low-resolution, TV quality, color video with 512*x*512 pixels/color at 8 bits/pixel, and 3 colors =~ $6\ x 10^6$ bits
- 24 *x* 36 mm (35-mm) negative photograph scanned at 12 μm which is 3000 *x* 2000 pixels/color, 8 bits/pixel, and 3 colors =~144 $x\ 10^6$ bits
- 14 *x* 17 inch radiograph scanned at 70 μm: 5000 *x* 6000 pixels, at resolution 12 bits/pixel =~ 360 $x\ 10^6$ bits. Medium size hospital generates tera bytes each year.
- LANDSAT Thematic Mapper scene: 6000x6000 pixels/spectral band, 8 bits/pixel, and 6 non-thernmal spectral bands =~1:7 $x\ 10^9$ bits.

Therefore, data compression is required for efficient **transmission:**

- to send more data in available bandwidth.
- to send the same data in less bandwidth
- more users on same same bandwidth

and **storage**

- to store more data
- to compress for local storage, and
- to put details on cheaper media

In both cases choice may be to compress or lose, e.g., you are allotted a given storage or bandwidth by an overall systems design, and your data rate is larger than the available storage or transmission capacity.

- Compression is also useful for progressive reconstruction, scalable delivery, browsing.
- It can also speed other signal processing by reducing the number of bits to be crunched (provided we have not lost essential information) or by combining with the compression algorithm, e.g.,
    - Enhancement
    - classification/detection
    - regression/estimation
    - filtering.

Example 2.2: Let us compute entropies of few grayscale (monochromatic, B/W) images as well as plotting histograms (computed probabilities) using a set of images from the Matlab database, which exhibit 256-levels of gray levels (8-bits), 0 being black and 255 white.

```
% Example using entropy.m to plot the histogram and compute entropy  of sample images.
% Note these 8-bit grayscale images are from the Matlab library.
% H is measure in bits per pixel
% Author: K. S. Thyagarajan; Modified by: H. Abut

A= imread('lena.ras');
figure, imshow(A), title('Original Image: Lena');

p=histogram(A);
figure, stem(p), title('Image Histogram')
xlabel('Pixel Value'),ylabel('Relative Frequency')

H=sum(-p .* log10(p+1e-06))/log10(2);
disp('Image Entropy in Bits/pixel: ');
disp(H);
```

Image Entropy in Bits/pixel: 7.7498


Original Image: Lena

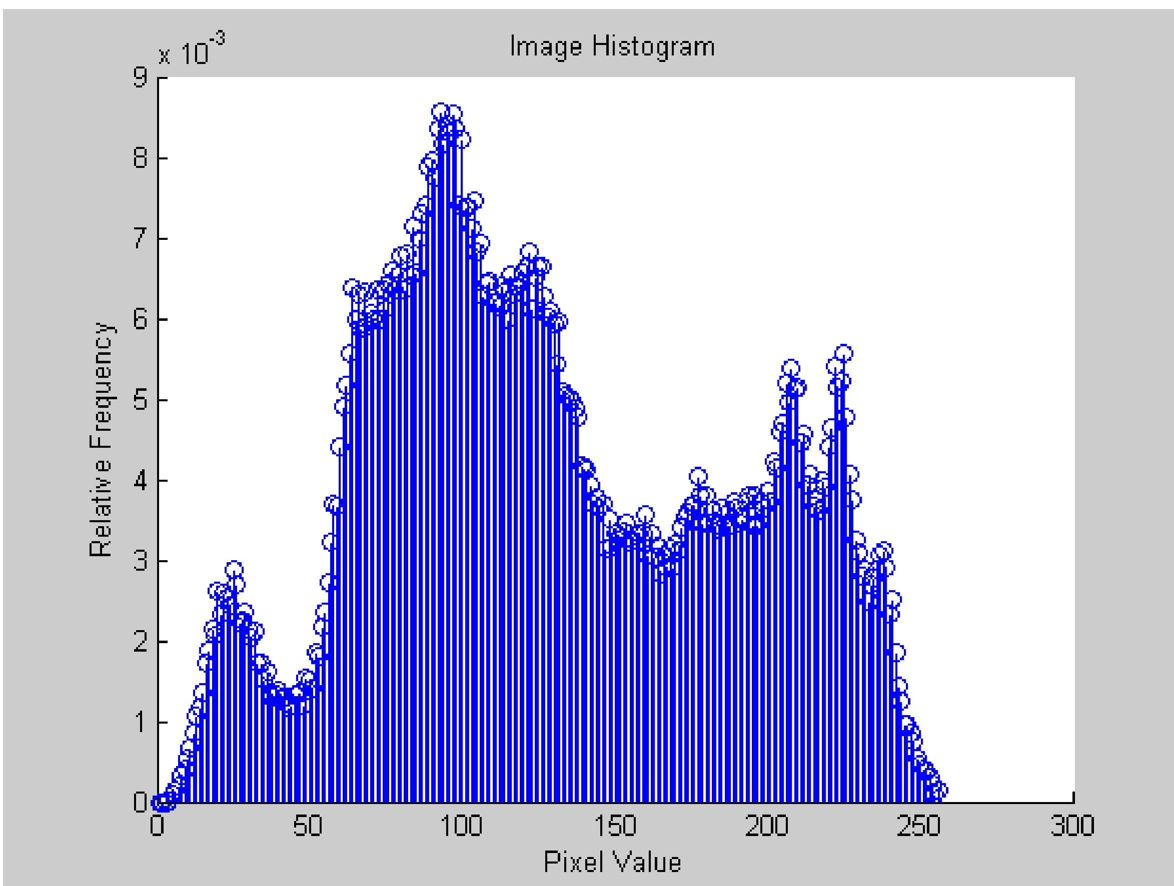


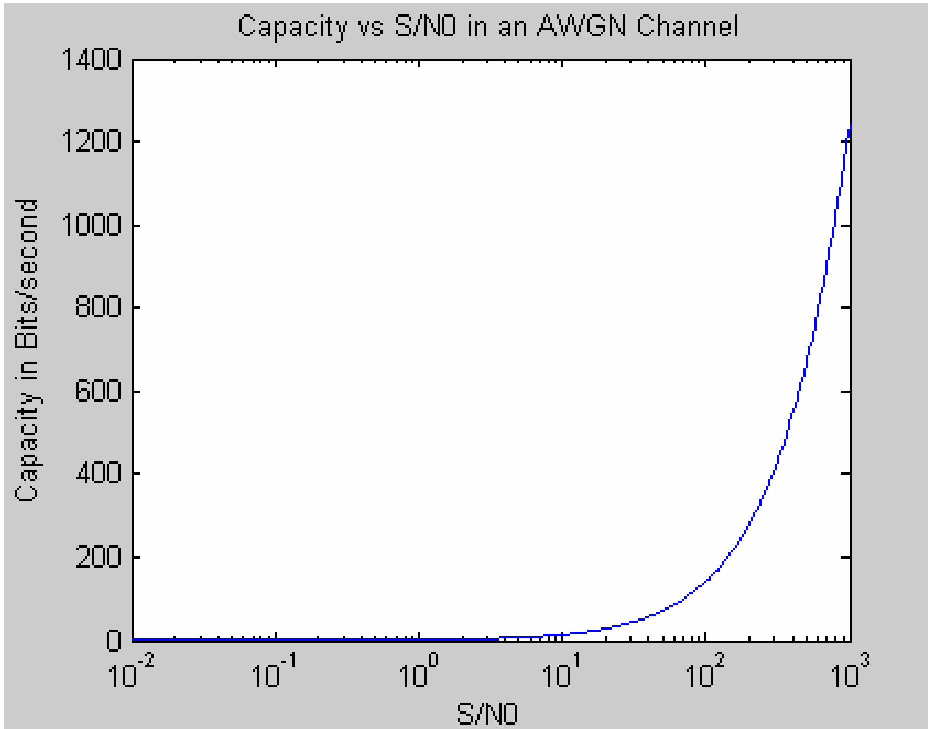
Image Histogram
x 10^-3
Relative Frequency
Pixel Value
0
1
2
3
4
5
6
7
8
9
0
50
100
150
200
250
300

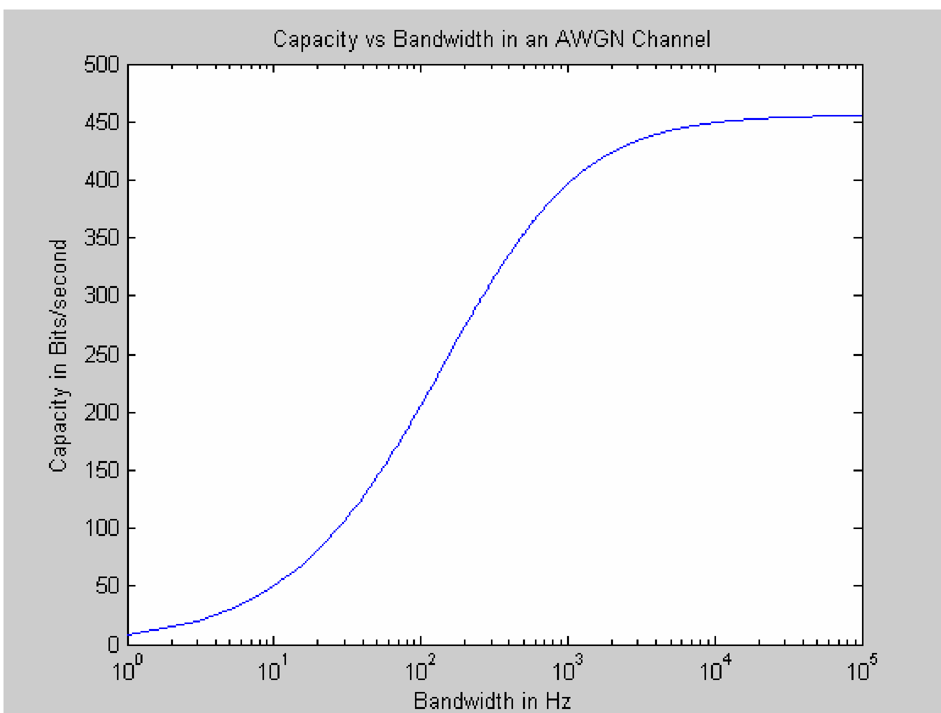
```
function [h,H]=histogram(A,L);
% HISTOGRAM Compute the histogram and cumulative histogram of A.
%    h = HISTOGRAM(A) computes the histogram assuming there are
%    256 integer grey-levels in A.
%    h = HISTOGRAM(A,L) computes the histogram using L integer
%    grey-levels in A.
%    [h,H] = HISTOGRAM(...) returns the histogram h and the
%    cumulative histogram H.
%    h is normalised so that the sum of the histogram is 1.
%    The maximum value of H is 1.
% M. Andrews 9 August 2000; Modified by H. Abut

% Create histogram bins
        if nargin == 1
          Bins=0:255;
        else
          Bins=0:L-1;
        end;

% Compute and normalise histogram
h=hist(A(:),Bins);
h=h/sum(h);

% Compute cumulative histogram
        if nargout == 2
          H=cumsum(h);
        end;
```

Example 2.3: Let us plot the capacity of an additive Gaussian channel with a bandwidth 3000 Hz as a function of $-20\,dB \le S/N_0 \le 30\,dB$. Repeat the experiment for a constant $S/N_0 = 25\,dB$ as a function of the bandwidth.

```
% Example 2.3 Channel Capacity Plots for a Additive Gaussian Noise Channel
% Programmed: Proakis, Salehi, Bauch
% Modified: H. Abut
echo on
bandwidth = 3000                   % Bandwith in Hz.
sovern0 = [-20:0.2:30];            % S/N0 in dB
sovern0 = 10 .^ (sovern0 ./ 10);   % S/N0 in none dB representation

capacity = bandwidth .* log2(1+sovern0/bandwidth);
semilogx(sovern0, capacity); title('Capacity vs S/N0 in an AWGN Channel')
xlabel('S/N0'); ylabel('Capacity in Bits/second');
clear;
w=[1:10,12:2:100,105:5:500,510:10:5000,5025:25:20000,20050:50:100000];
sovern0=25;
sovern0 = 10 .^ (sovern0 ./ 10);
```

```
capacity = w .* log2(1+sovern0 ./ w);
figure; semilogx(w,capacity); title('Capacity vs Bandwidth in an AWGN Channel')
xlabel('Bandwidth in Hz'); ylabel('Capacity in Bits/second');
```





LOSSLESS vs LOSSY COMPRESSION

Overview of Lossless Compression:

It is also known as:

- Noiseless coding
- Lossless coding
- Invertible coding
- Entropy coding
- Data compaction codes.

They can perfectly recover original data (if no storage or transmission bit errors, i.e., noiseless channel).

1. They normally have variable length binary codewords to produce variable numbers of bits per symbol.
2. Only works for ***digital sources***.

Properties of Variable rate (length) systems:

- They can more efficiently trade off quality/distortion (noise) and rate.
- They are generally more complex and costly to build.
- They require buffering to match fixed-rate transmission system requirements.
- They tend to suffer catastrophic transmission error propagation; in other words, once in error then the error grows quickly.

3. But they can provide produce superior rate/distortion (noise) trade-off. For instance, in image compression we can use more bits for edges, fewer for flat areas. Similarly, more bits can be assigned for plosives, fewer for vowels in speech compression.

General idea: Code highly probable symbols into short binary sequences, low probability symbols into long binary sequences, so that average is minimized. Most famous examples:

- **Morse code:** Consider dots and dashes as binary levels “0” and “1”.

  Assign codeword length inversely proportional to letter relative frequencies. In other words, most frequent letter (message) is assigned the smallest number of bits, whereas the least likely message has the longest codeword, such as “e” and “”z” in English alphabet.

  Total number of bits per second would much less than if we have used fixed-rate ASCII representation for alphanumerical data.

- **Huffman code** 1952, employed in UNIX *compact* utility and many standards, known to be optimal under specific constraints. (We will study later.)
- **Run-length codes** popularized by Golomb in early 1960s, used in JPEG standard.
- **Lempel-Ziv(-Welch) codes** 1977,78 in Unix *compress* utility, diskdouble, stuffit, stacker, PKzip, winzip, DOS, GIF.
- **Arithmetic codes** by Fano, Berlecamp, Rissanen, Pasco, Langdon. Example: IBM Q-coder.

Warning: To get average rate down, need to let maximum instantaneous rate grow. This means that can get data expansion instead of compression in the short run.

Typical lossless compression ratios: 2:1 to 4:1

Overview of Lossy Compression:

These schemes are non-invertible and information is always lost. However, they permit more compression.
Examples:

1. **Memoryless, non-predictive compression techniques:**
   - PCM (Shannon (1938), Oliver, Pierce (1948).
   - Sampling + scalar quantization; analog to digital conversion.
   - By introducing loss in a controlled fashion, we could prevent further loss in the channel. Birth of digital communication.

2. **Predictive coding:**
   Developed by Derjavitch, Deloraine, and Van Mierlo (1947), Elias (1950, 1955), Cutler (1952), DeJager (1952).

Idea: Predict next sample based on previous reconstructions (decisions) and code the differences (residual, prediction error). Specific systems:

- Predictive scalar quantization: Differential pulse code modulation (DPCM), delta modulation (DM), adaptive DPCM (ADPCM), sigma-delta modulation ($\Sigma - \Lambda$ *Modulation*).
- Almost all speech coders, in particular, cellular systems, use some form of predictive coding.
- CD players use $\Sigma - \Lambda$ *Modulation* and delta modulation, both forms of predictive codes.
- Most video codecs use frame-to-frame predictive coding: Motion compensation (MC).
- **Optimal quantization** Lloyd (1956) Optimizing PCM. connection of quantization and statistics (clustering)

3. **Transform coding:**

Developed Mathews and Kramer (1956), Huang (1962, 1963, Habibi, Chen (Compression Labs Inc). Dominant image coding (lossy compression) method in ISO and other standards:

- p*64, H.26*, JPEG, JPEG2000
- MPEG I, II, IV, and now VII.
- C-Cubed, CLI, Picture-Tel, and many others
- JPEG is ubiquitous in WWW and use transform coding (DCT) + custom uniform quantizers + runlength coding + Huffman or arithmetic coding..

4. **Linear predictive coding (LPC)**

First very low bit rate speech coding based on sophisticated signal processing and cellular technology, implementable in DSP chips.

- Itakura and Saito (1968), Atal et al. (1971), Markel and Gray (1972-76), NTT, Bell, Signal Technology, TI (Speak and Spell).

5. Vector Quantization (VQ)

Initial development late 1970s early 1980s.

- Very low rate LPC speech coding, Chaffee and Omura (1974-75), Linde, Buzo, Gray, et al. (1978-1980), Adoul et al. (1978), Abut, Gray, et al. (1982-84).
- Low rate image/video coding 1.0 bit/pixel or lower, Hilbert (1977), Yamada, et al. (1980-83), Gersho (1982), Baker (1982-83), Caprio et al. (1978), Menez, et al. (1979).
- Code excited LPC (CELP) basis for the cell phone technology: Steward (1981-82), Schroeder and Atal (1984-85)
- Software-based video: table look-ups instead of computation. Tekalp and others (1986). Quicktime, Sorenson, Indeo, Cell, Supermac Cinepak, Motive, VXtreme.

It is imperative for HDTV systems. (Already part of the Grand-Alliance Proposal for the standard.) It is currently in use in software based video: table lookups instead of computation Apple's QuickTime, Intel's Indeo, Sun's Cell, Supermac Technology's Cinepak, Media Vision's Motive, VXtreme

Various structures:

- Product codes (scalar quantization, gain/shape)
- Successive approximation/ tree-structured
- Predictive and finite-state
- Fractal coding
- Video streaming.

6. Subband/pyramid/wavelet Coding: (late1980s).

- MUSICAM digital audio (European standard)
- EZW (winzip) (Knowles, Shapiro)
- SPHIT (Said and Pearlman)
- CREW (RICOH Calif.),
- JPEG 2000 (Marcellin 2002)
- MPEG 4, MPEG7

7. Properties: (Possibly) in between lossless and lossy: "perceptually lossless" compression:

- what the eye can see (which may depend on context).
- Noise level of the acquisition device.
- What can be squeezed through transmission or storage, i.e., an imperfect picture may be better than no picture at all.
- Loss may be unavoidable, may have to choose between imperfect image or long delays or no data at all.
- However, some loss is not a problem with follow-up studies, archives, R&D, education, entertainment.

8. Fundamental Question: Is lossy compression acceptable?

- Growing evidence suggests that lossy compression even in medicine does not damage diagnosis and may in fact improve it if performed intelligently.
- However, it is NOT acceptable for computer programs, banking, espionage and a number of other security emphasized fields.

Compression application areas:

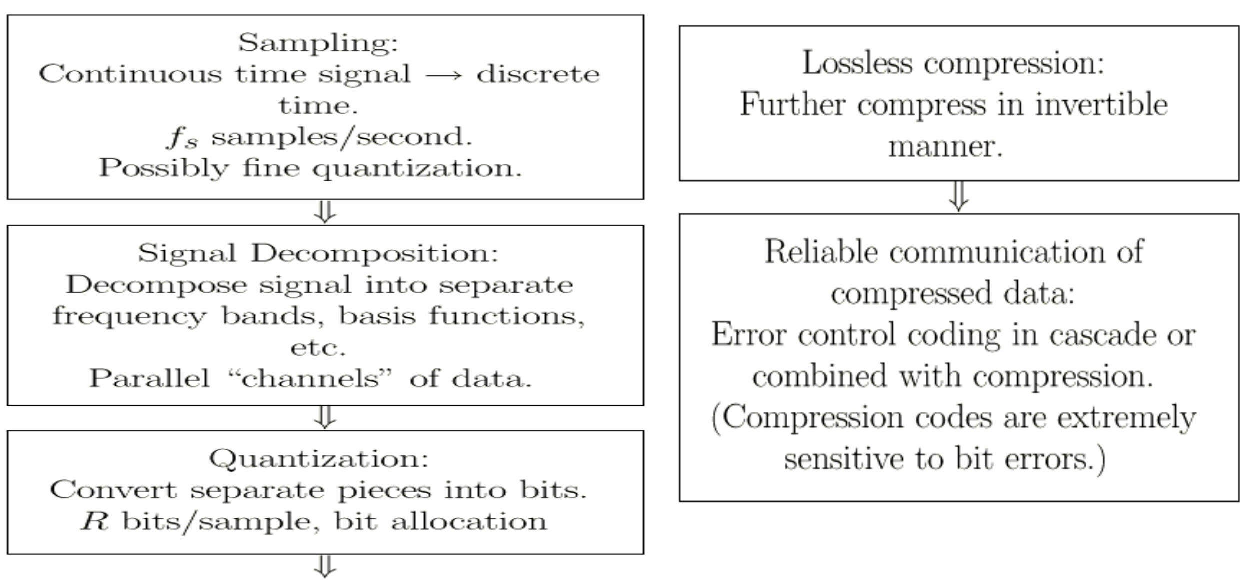
- Teleconferencing, FAX
- Medical records (archival, educational, remote diagnosis)
- Remote sensing, Remote control
- Space communications
- Multimedia, Web
- Binary, gray-scale, color, video, audio.
- Instructional TV on the net.
- Surveillance.
- Classification and feature extraction in person identification, gene clustering, bioinformatics.
- Image processing (texture modeling, synthesis)
- Data mining

Compression Context:

- It is generally part of a general system for data acquisition, transmission, storage, and display within the Shannon point-to-point communication system model.
- It involves A/D and D/A conversions and digital acquisition.
- Quality and utility of "modified" digital images/sounds/video varies depending on the specifics of the application and the bandwidth available.

Components of Typical Encoded Communication System

Transmitter (Encoding, Compression)



Sampling:

It is also known as A/D conversion, high resolution quantization achieved by digitizing amplitude if it is one-dimensional signal or digitizing space and amplitude for 2-D signals, i.e., image and video.

$N \times M$ pixels, $G = 2^r$ gray levels. Number of bits = $N \times M \times r$
Note: This step not necessary for digitally acquired image.

Basic idea: Sample and quantize:

For instance, for an image the process can be formulated by:

$$\{f(x,y);\ x,y \in \mathcal{R}\}$$
$$\Downarrow$$
$$\{f_{n,m} = f(n\Delta x, m\Delta y);\ n,m \in \mathcal{Z}\}$$
$$\Downarrow$$
$$\{q(f_{n,m});\ n,m \in \mathcal{Z}\}$$

Terminology: $f(x,y)$: image intensity (amplitude, brightness, luminance) at location (x,y)

$f_{n,m}$: sampled image value with integer indices (n,m), which is obtained at:

$f(n\Delta x, m\Delta y)$ of the original image with resolution Δx horizontally and Δy vertically.

$q(f_{n,m})$: digitized/quantized value of the image $f(x,y)$

Sampling (Nyquist) Theorem:

If a continuous-time (analog) signal *x(t)* has no frequency components (harmonics) at values greater than a frequency value f_{max} then this signal can be UNIQUELY represented by its equally spaced samples if the sampling frequency F_s is greater than or equal to $2f_{max}$. This is known as the analog-to-digital (A/D) conversion at F_s samples/s. Furthermore, the original continuous signal x(t) can be TOTALLY recovered from its samples x(n) after passing them through an ideal integrator (ideal low-pass filter) with an appropriate bandwidth.

***Signal decomposition* (transformation, mapping):** Decompose image or speech into collection of separate images/speech (bands, coefficients). Examples:

- Fourier, Hadamard, Walsh, Sine, Discrete Cosine (DCT) and Karhunen-Loeve (KL),
- Wavelet and Sub-band (multi-resolution),
- Also: Hotelling, Principal Value Decomposition, Hartley, Fractal (Typically done digitally.)

Why?

Several reasons:

1. Good transforms tend to compact energy into a few coefficients, which allow many to be quantized to zero bits without affecting quality.
2. Good transforms tend to decorrelate (reduce linear dependence) among coefficients, causing scalar quantizers to be more efficient. (folk theorem)
3. Good transforms are effectively expanding the signals in good basis functions. (Mathematical intuition.)
4. The eye and ear tend be sensitive to behavior in the frequency domain, so coding in the frequency domain allows the use o perceptually based distortion measures, e.g., incorporating masking.

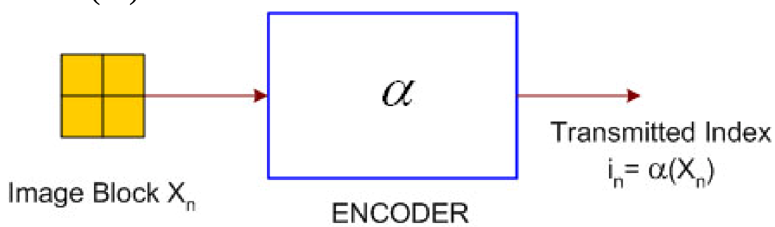
Issues

- Encoder and decoder structures/ Encoder and decoder algorithms,
- Encoder and decoder design and optimization/ Encoder and decoder performance (*R; D*),
- Encoder and decoder complexity/cost,
- Theoretical bounds and limits, and
- Systems

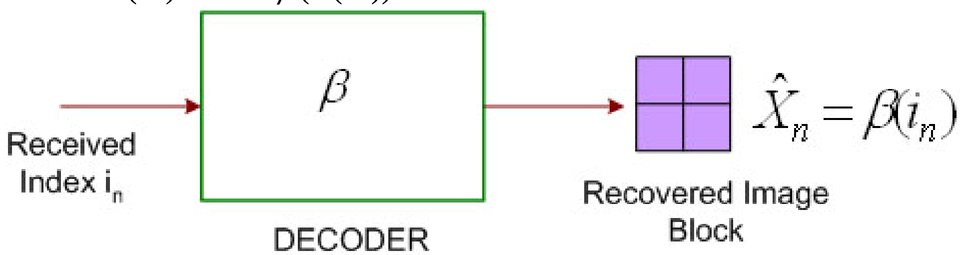
Encoder - Decoder Pair Memoryless Block Coder

Memoryless: We use information from the current and only current sample to code the signal.
Encoding and Decoding Operations:

Encoder: $X \xrightarrow{\alpha} i = \alpha(X)$ (3.1)



Decoder: $\alpha(X) \xrightarrow{\beta} \hat{X} = \beta(\alpha(X))$ (3.2)



Code Length, Rate: Consider a binary vector i composed of $\{0,1\}$ we define the length of a binary vector as:

$$l(i) = \textit{length of the binary string } i$$

Example: $l(0) = 1;$ $l(101) = 3;$ $l(1011000) = 7$

Instantaneous rate of a binary vector i is defined by:

$$r(i) = \frac{i}{k} \qquad \text{bits/symbol} \qquad (3.3)$$

where k is the number of possible binary vectors, symbols.

The average rate (average codeword length) of the encoder applied to the source is defined by:

$$R = E[r(\alpha(X))] \quad \text{bits/symbol} \qquad (3.4)$$

where E stands for average (expected value).
An encoder is "fixed-rate" or "fixed-length" if all the codewords sent to the channel has the same length:

$$l(i) = R.k \quad \textit{for all } i \qquad (3.5)$$

Otherwise, it a "variable-rate" or "variable-length" coder.

Remarks: Selection of fixed or variable rate codes can have important implications in practice.

- Variable rate codes can cost more as they may require data buffering if the encoded data is to be transmitted over a fixed rate channel.
- They are harder to synchronize. Single errors in decoding, lost, or added bits on the channel can have catastrophic effects.
- Buffers can overflow (causing data loss) or underflow (causing wasted time or bandwidth).

- However, variable rate codes can provide superior compression performance. For example, in image compression we can use more bits for edges, fewer for background. In voice compression, more bits for plosives, fewer for vowels.

A source code is **invertible** or **noiseless** or **lossless** if

$$\beta(\alpha(x)) = x \tag{3.6}$$

which is the same as saying, the code is invertible if

$$\beta = \alpha^{-1} \tag{3.7}$$

A code is **lossy** if it is not lossless. In this case, a notion of **distortion *d*** between input vector and reconstructed replica has to be developed and it must be measured to quantify the seriousness of the loss.

Distortion Measure: It measures $d(X,\hat{X})$ loss resulting if an original input X is reproduced as $\hat{X}$ at the decoder. Mathematically, a distortion measure satisfies: $d(X,\hat{X}) \geq 0$. To be useful d should be:

- Easy to compute
- Easy to tract
- Meaningful for perception or application.

Types of Distortion: No single distortion measure accomplishes all of these goals, although the well-known **mean squared-error (MSE)** distortion (distance) is defined by:

$$d(X,\hat{X}) = \left\| X - \hat{X} \right\|^2 = \sum_{l=0}^{k-1} |x_l - \hat{x}_l\ | \tag{3.8}$$

where $X = (x_0, x_1, \cdots x_{k-1})$ and $\hat{X} = (\hat{x}_0, \hat{x}_1, \cdots \hat{x}_{k-1})$ individual vectors. This measure satisfies the first two properties and occasionally accomplishes the third and most difficult.

Weighted or transform versions are used perceptual coding:

$$d(X,\hat{X}) = (X - \hat{X})^* B(X - \hat{X}) \tag{3.9}$$

where "$*$" stands for complex conjugate of a complex vector and B_X is a matrix with some special characteristic, hopefully, correlating with perception. Note that if B_X is an identity matrix then the distortion measure reduces to the mean-square error (MSE) case of (3.8).

Mean absolute difference (MAE) distortion (distance) is defined by:

$$d(X,\hat{X}) = \left\| X - \hat{X} \right\| = \sum_{l=0}^{k-1} |x_l - \hat{x}_l\ | \tag{3.8}$$

where $X = (x_0, x_1, \cdots x_{k-1})$ and $\hat{X} = (\hat{x}_0, \hat{x}_1, \cdots \hat{x}_{k-1})$ individual vectors. This measure more frequently used in image compression due to its computational simplicity and it also satisfies the first two properties.

Hamming Distortion or Distance:

$$d_H(X,\hat{X}) = \begin{cases} 0 & if \;\; X = \hat{X} \\ 1 & f \;\; X \neq \hat{X} \end{cases} \tag{3.10}$$

Average Hamming Distance for vectors:

$$d_H(X,\hat{X}) = \sum_{l=0}^{k-1} d_H(x_l,\hat{x}_l) \tag{3.11}$$

Common assumption: *d* is an **additive** distortion measure. We often normalize distortion with respect to magnitude, power or other factors.

Remarks: If $d(X,\hat{X}) = 0 \; if \;\; and \;\; only \; if \;\; X = \hat{X}$

- This clearly implies that zero distortion is equivalent to *lossless* coding. We normally make this assumption if the source is discrete.
- Performance of a data compression system is measured by the expected values of the distortion $d(X,\hat{X})$ and the associated rate *R*.

Units of rate will be:

- Bits/symbol for binary communication systems.
- Bits/sample for one-dimensional signals such as sampled speech.
- Bits/pixel for still imagery, and
- Bits/second for image sequence coding.

Example 3.1: 1101 and 0101 have a Hamming distance 1
1101 and 0001 have a Hamming distance 2.

The significance of the Hamming distance: When two binary vectors have Hamming distance *d*, then it would take *d* single-bit errors (i.e. inversions of bit values) to convert one into the other.

Example 3.2: Above three vectors corresponds to digits: 13, 5, and 1. The MSE difference for the first pair would be: $(13\text{-}5)^2$=64 and the second one is: $(13\text{-}1)^2$=144. Similarly, MAE values would be: 8 and 12, which would have very little value.

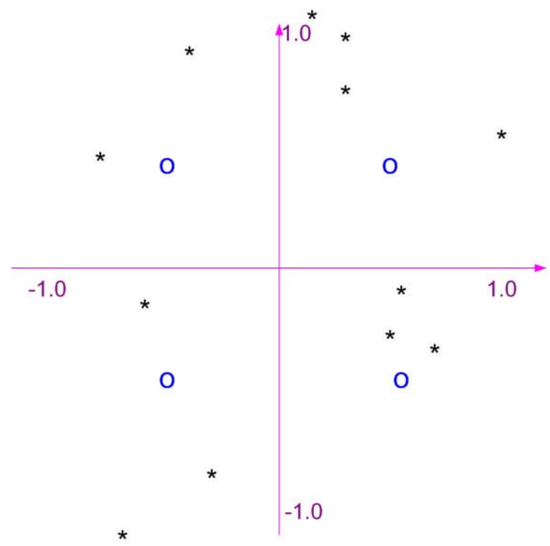
Example 3.3: Let us consider the following 12 vectors (stars) clustered in a 2-D space and suppose that we will represent them with four codewords (circles) located at (0.5,0.5), (-0.5,0.5), (-0.5,-0.5) and (0.5,-0.5). In doing so, we will select the circles in the same quarter-circle where the point lies. Let us compute the average MSE and MAE for this dataset.

Horizontal	-0.37	.63	-.83	-.70	-.28	1.09	.59	.14	.70	.30	.30	.37
Vertical	.98	-.11	.60	-1.21	-.94	.51	.17	1.76	-.35	.79	1.06	-.32
Circle Used	-.5;.5	.5;-.5	-.5;.5	-.5;-.5	-.5;-.5	.5;.5	.5;.5	.5;.5	.5;-.5	.5;.5	.5;.5	.5;-.5
Abs. Diff.	.13;.48	.13;.30	.33;.10	.20;.71	.22;.44	.59;.01	.09;.33	.36;1.26	.2;.15	.2;.29	.2;.56	.13;.18
ΣDiff. Sq.	.2473	0.171	.1189	.5441	.242	.3482	.117	.4777	.0625	.1241	.3536	.0493

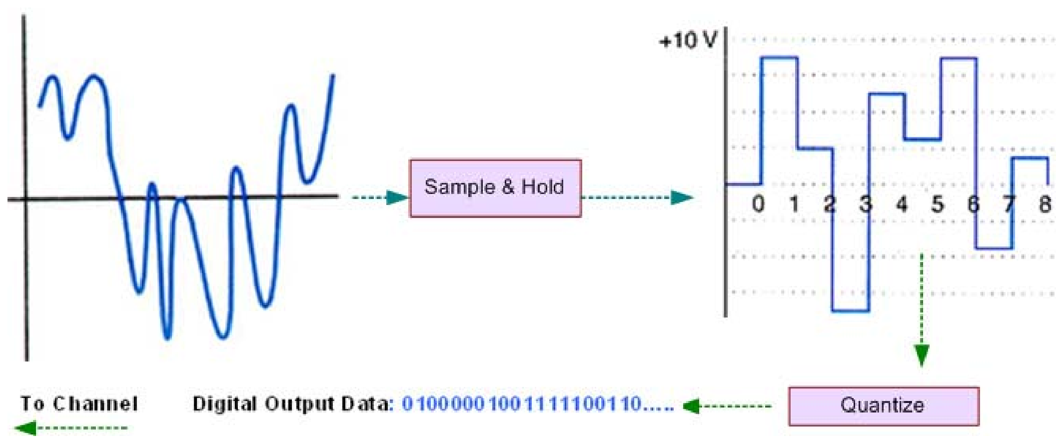
Average Absolute Difference= .13+.48+.33+.39+….+.18/12=0.72 units

MSE=.2473+.171+…+.0493/12=0.238 units

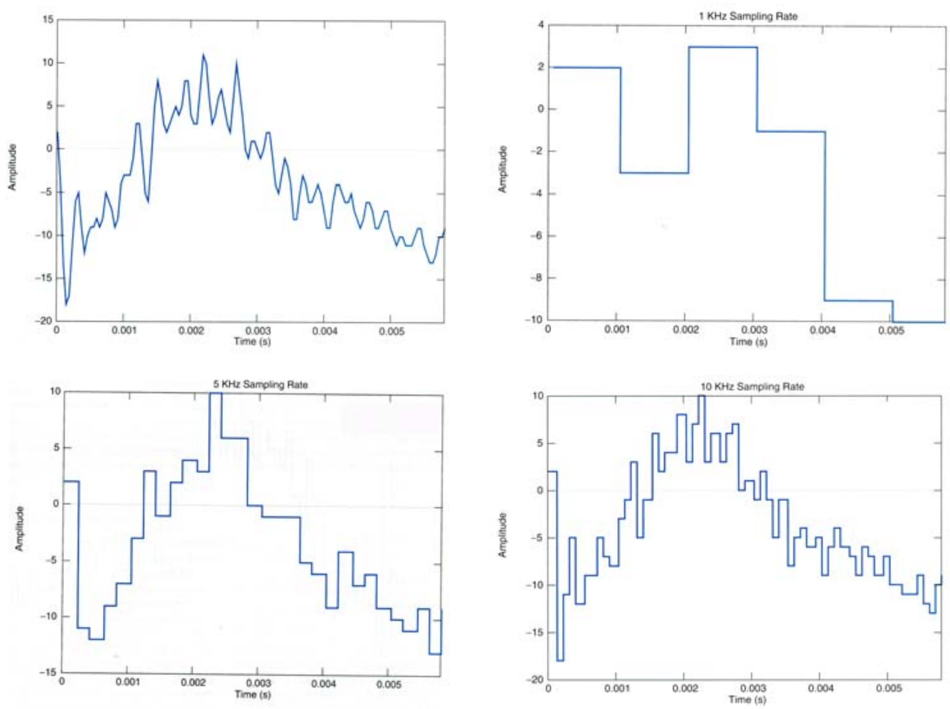
These would have significant if the samples were digitized values of imagery in a unit-size scale.



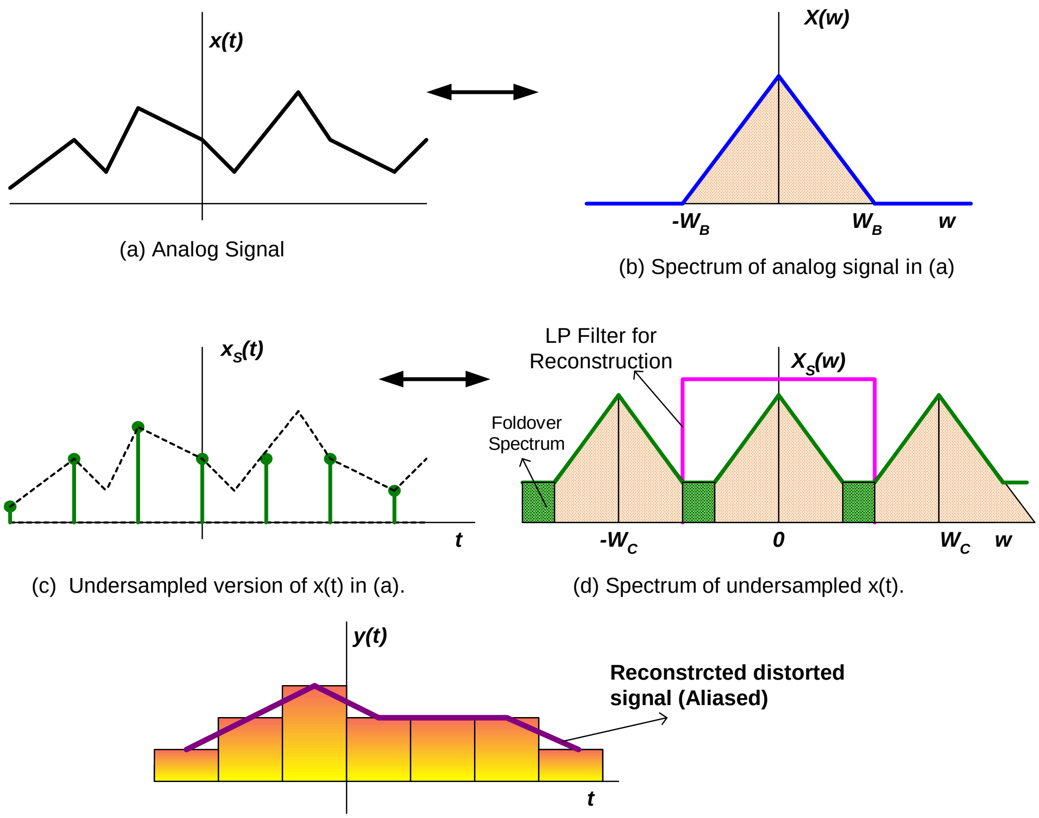
Example: 3.4 Sampling & PCM: From the information processing perspective key issue in the digital communication system is to have proper sampling and PCM rates. If we do not sample at the Nyquist rate or above distortion is unavoidable.



Encoding Stage: The effect of sampling rate on analog signals can be illustrated as shown below, where the first 5.0 ms segment of the sound "I" in the word "Information" spoken by a male speaker. Upper left is the original analog signal and the rest are the versions sampled at 1.0 KHz, 5.0 kHz, and 10.0 kHz, respectively. It is to see that the first two sampling rates have poor tracking capability of the original, whereas, the last figure has an envelope very close to that of the analog version.



Decoding and Reconstruction Stage: It is commonly known as the D/A conversion and achieved by interpolating the discrete samples by use of an ideal Low-Pass filter with a bandwidth *B Hz*. This reconstruction process for an under-sampled (below Nyquist rate) can be illustrated as:



(a) Analog Signal

(b) Spectrum of analog signal in (a)

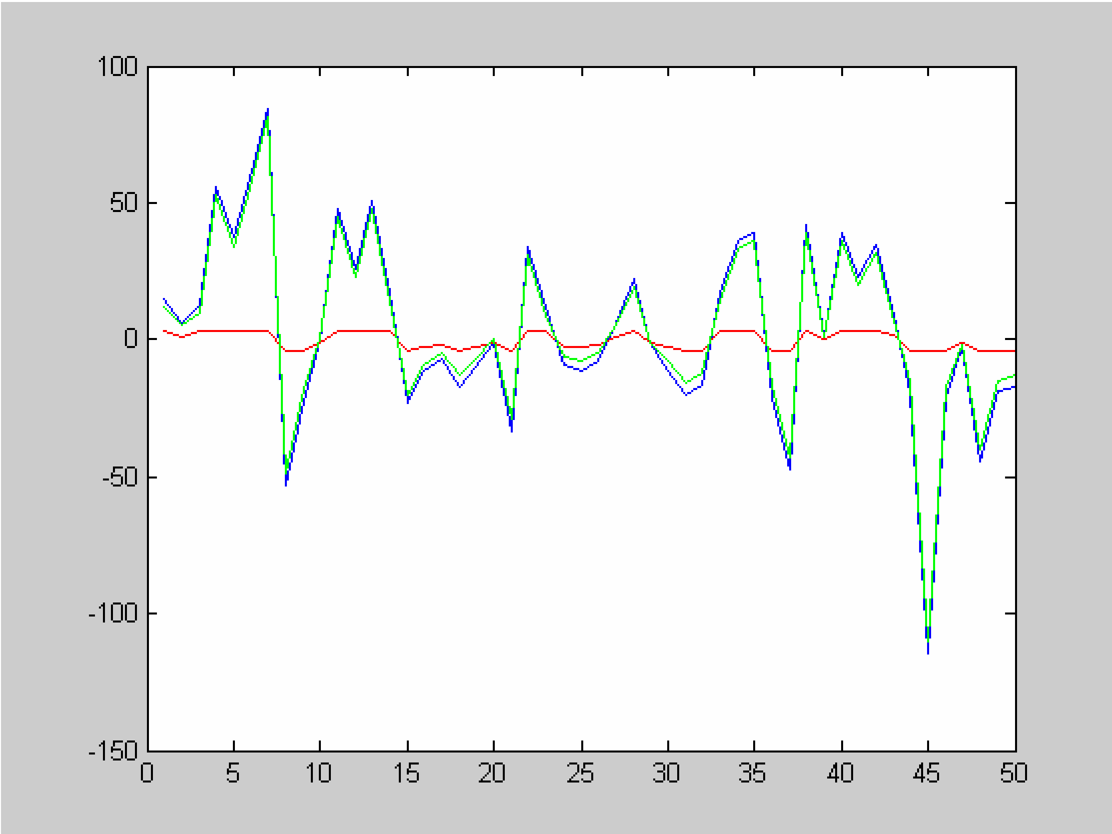
(c) Undersampled version of x(t) in (a).

(d) Spectrum of undersampled x(t).

(e) Distorted output signal y(t).

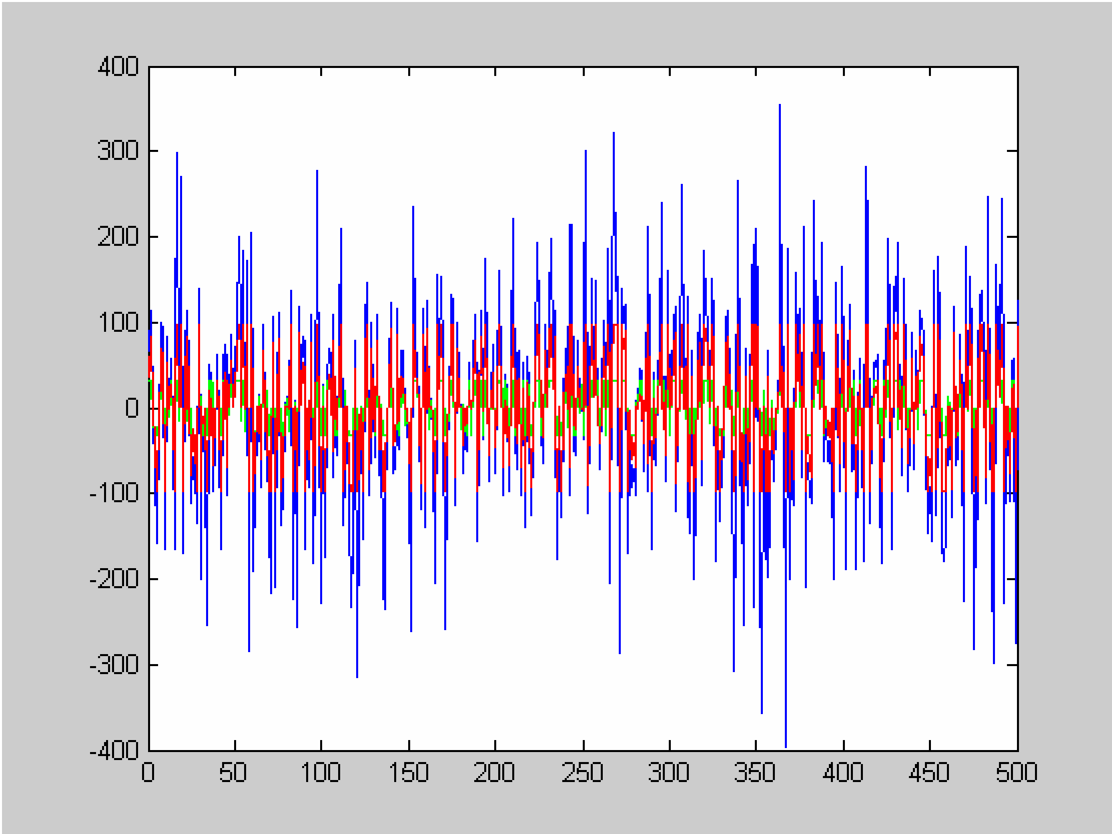
Example: 3.4 Uniform PCM of random numbers

```
% PCM Codec (8-levels) for encoding of 50 random numbers
a=randn(1,50); index=1:50;
a= a .* 32; disp (a)
a_quan=uencode(a,3,16,'signed'); disp (a_quan);
error=int8(a)-a_quan
plot(index,a,'b',index, a_quan,'r',index,error,'g')
```



- As we can see from the matlab output, the error curve (green) is enormous. Terribly poor encoding performance!
- Let us repeat the example for 500 point and 6-bit uniform PCM. The plots clearly indicate a significantly reduced error curve as expected.

```
% Repeat the example for 500 points and 6-bits
a=randn(1,500);
index=1:500;
a= a .* 128; disp (a)
a_quan=uencode(a,6,32,'signed'); disp (a_quan);
error=int8(a)-a_quan

figure;
plot(index,a,'b',index, a_quan,'g',index,error,'r')
```


400
300
200
100
0
-100
-200
-300
-400
0
50
100
150
200
250
300
350
400
450
500