Chapter 5 Image Compression

Introduction

Image compression is a type of data compression applied to digital images, to reduce their cost for storage or transmission. Algorithms may take advantage of visual perception and the statistical properties of image data to provide superior results compared with generic data compression methods which are used for other digital data. It is a type of compression technique that reduces the size of an image file without affecting or degrading its quality to a greater extent.

Image compression is typically performed through an image/data compression algorithm or codec. Typically such codecs/algorithms apply different techniques to reduce the image size, such as by: Specifying all similarly colored pixels by the color name, code and the number of pixels. This way one pixel can correspond to hundreds or thousands of pixels.

Why Compression?

- To reduce the volume of data to be transmitted (text, fax, images)
- To reduce the bandwidth required for transmission and to reduce storage requirements (speech, audio, video)

How is compression possible?

- Redundancy in digital audio, image, and video data
- Properties of human perception
- Digital audio is a series of sample values; image is a rectangular array of pixel values; video is a sequence of images played out at a certain rate
- Neighboring sample values are correlated

Redundancy

- Adjacent audio samples are similar (predictive encoding); samples corresponding to silence (silence removal)
- In digital image, neighboring samples on a scanning line are normally similar (spatial redundancy)
- In digital video, in addition to spatial redundancy, neighboring images in a video sequence may be similar (temporal redundancy)

Compressed version of digital audio, image, video need not represent the original information exactly. Perception sensitivities are different for different signal patterns. Human eye is less sensitive to the higher spatial frequency components than the lower frequencies (transform coding).

Compression algorithms remove redundancy

If more data are used than is strictly necessary, then we say that there is redundancy in the dataset. Data redundancy is not abstract concept but a mathematically quantifiable entity. If n_1 and n_c denote the number of information carrying units in two data sets that represent the same information, the relative data redundancy R_D of the first data set (n_1) can be defined as:

$R_D = 1 - 1/C_R$ (1) $C_R = n_1/n_c$ (2)

Where C_R is compression ratio, defined as:

Where n_1 is the number of information carrying units used in the uncompressed dataset and n_c is the number of units in the compressed dataset. The same units should be used for n_1 and n_c; bits or bytes are typically used.

When $nc<<n1$, $C_R \rightarrow$ large value and $R_D \rightarrow 1$. Larger values of C indicate better compression
When $nc>>n1$, $C_R \rightarrow 0$ and $R_D \rightarrow$ large value. Undesirable Case

Three basic types of redundancy can be identified in a single image:

1) Coding redundancy
2) Interpixel redundancy
3) Psychovisual redundancy

5.1 Coding Redundancy

- A code is a system of symbols used to represent a body of information or set of events.
- Each piece of information or event is assigned a sequence of code symbols called code word. The number of symbols in each code word is called its length. For eg:- Each pixel in a gray scale image is represented by 8-bits binary. (Pixel value = 255, code = 11111111, code length per pixel= 8)
- The code words are ordered in the same way as the intensities that they represent; thus the bit pattern 00000000, corresponding to the value 0, represents the darkest points in an image and the bit pattern 11111111, corresponding to the value 255, represents the brightest points.
- if the size of the code word is larger than is necessary to represent all quantization levels, then we have coding redundancy. An 8-bit coding scheme has the capacity to represent 256 distinct levels of intensity in an image. But if there are only 16 different grey levels in a image , the image exhibits coding redundancy because it could be represented using a 4-bit coding scheme. Coding redundancy can also arise due to the use of fixed-length code words.
- Grey level histogram of an image also can provide a great deal of insight into the construction of codes to reduce the amount of data used to represent it .
- Let us assume, that a discrete random variable r_k in the interval (0,1) represents the grey levels of an image and that each r_k occurs with probability $Pr(r_k)$. Probability can be estimated from the histogram of an image using

$$Pr(r_k) = h_k/n \text{ for } k = 0,1 \ldots\ldots L-1 \qquad (3)$$

Where L is the number of grey levels and h_k is the frequency of occurrence of grey level k (the number of times that the kth grey level appears in the image) and n is the total number of the pixels in the image. If the number of the bits used to represent each value of rk is l(rk), the average number of bits required to represent each pixel is :

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) P_r(r_k) \qquad (4)$$

Huffman Coding:

1. Ranking pixel values in decreasing order of their probability
2. Pair the two values with the lowest probabilities, labeling one of them with 0 and other with 1.
3. Link two symbols with lowest probabilities.
4. Go to step 2 until you generate a single symbol which probability is 1.
5. Trace the coding tree from a root.

- Assigns fewer bits to symbols that appear more often and more bits to the symbols that appear less often
- Efficient when occurrence probabilities vary widely
- Huffman codebook from the set of symbols and their occurring probabilities
- Two properties:
  - generate compact codes
  - prefix property

Problem on Huffman Coding

Consider the symbols with different Probabilities

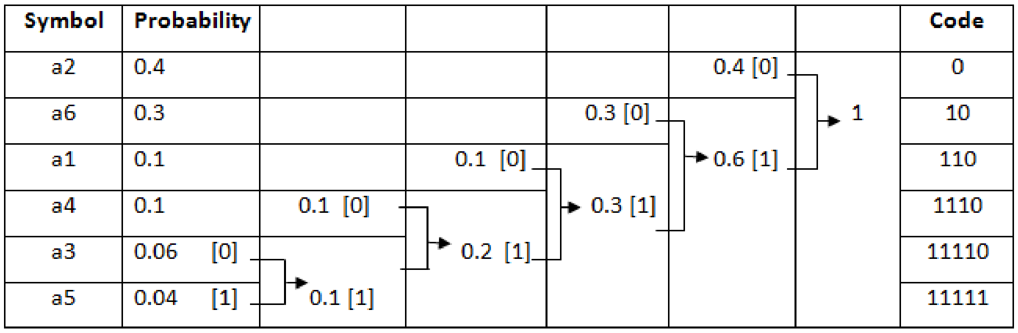
Symbol	Probability
a1	0.1
a2	0.4
a3	0.06
a4	0.1
a5	0.04
a6	0.3

The First step in Huffman's approach =>

Create a series of source reductions by ordering the probabilities. [Descending order]

The Second step =>

Code each reduced source starting with the smallest source and working back to the original source.



Symbol	Probability						Code
a2	0.4				0.4 [0]		0
a6	0.3			0.3 [0]		1	10
a1	0.1		0.1 [0]		0.6 [1]		110
a4	0.1	0.1 [0]		0.3 [1]			1110
a3	0.06 [0]		0.2 [1]				11110
a5	0.04 [1]	0.1 [1]					11111

$L_{avg} = (0.4)1 + (0.3)2 + (0.1)3 + (0.1)4 + (0.06)5 + (0.04)5 = 2.2$ bits/symbol

5.2 Inter-pixel redundancy

Inter-pixel redundancy is due to the correlation between the neighboring pixels in an image. That means neighboring pixels are not statistically independent. The gray levels are not equally probable. To reduce the inter-pixel redundancy the difference between adjacent pixels can be used to represent an image.

The intensity at a pixel may correlate strongly with the intensity value of its neighbors.

Because the value of any given pixel can be reasonably predicted from the value of its neighbors much of the visual contribution of a single pixel to an image is redundant; it could have been guessed on the bases of its neighbor's values.

We can remove redundancy by representing changes in intensity rather than absolute intensity values .For example, the differences between adjacent pixels can be used to represent an image . Transformation of this type is referred to as *mappings*. They are called *reversible* if the original image elements can be reconstructed from the transformed data set.

(a)Run length encoding is one of the examples which also take advantage of inter-pixel redundancy.

A "run: of consecutive pixels whose gray levels are identical is replaced with two values: the length of the run and the gray level of all pixels in the run. Example (50, 50, 50, 50) becomes (4, 50).Especially suited for synthetic images containing large homogeneous regions. The encoding process is effective only if there are sequences of 4 or more repeating characters

Applications – compression of binary images to be faxed.

- Repeated occurrence of the same character is called a run
- Number of repetition is called the length of the run

Example1:

AAABBCDDDD

Encoded: 3A2B1C4D Decoded: AAABBCDDDD

It is supported by most bitmap file formats (BMP, TIFF etc).It performs lossless data compression. Run length coding works by reducing the physical size of repeating string of characters. Generally represented => 2 bytes (First byte : No of character and Second byte : value) AAAAAAAAAAAAAAA => 15A (15 bytes replaced by 2 bytes)

Images with repeating gray value along rows (columns) can be compressed by storing runs of identical grey values. Compression is measured with Compression Ratio (CR)

Compression Ratio (CR) = Original Size / Compressed Size: 1

Consider a 16 character string => 000PPPPPPXXXXAAA => 306P4X3A

CR = 16/8 : 1 => CR = 2 : 1

The various Run Length Coding approaches are represented below.

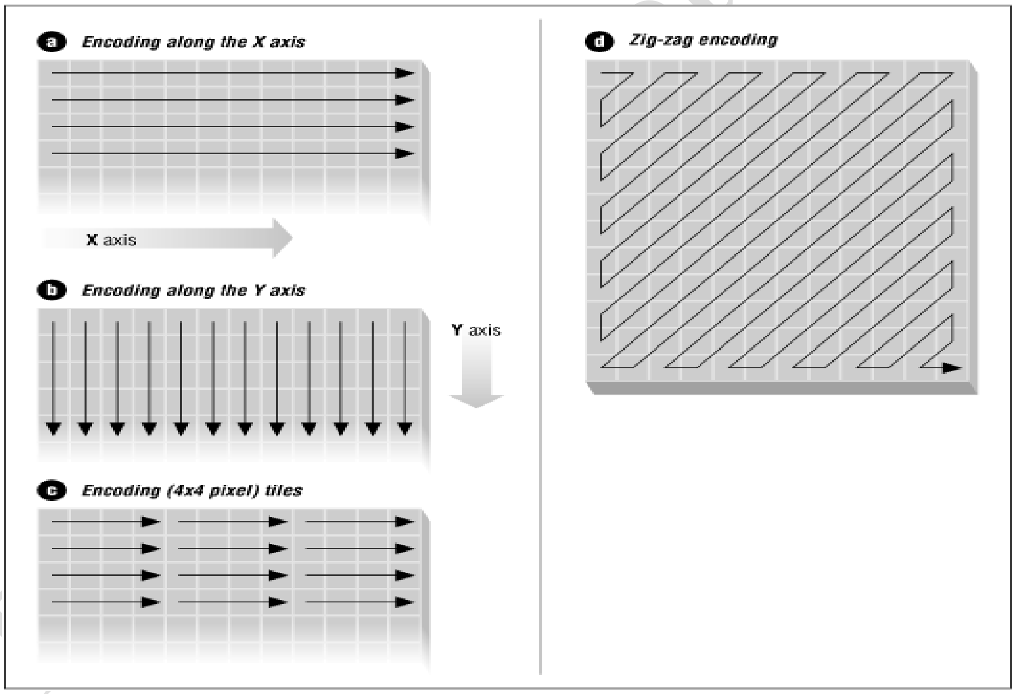

a Encoding along the X axis
X axis
b Encoding along the Y axis
Y axis
c Encoding (4x4 pixel) tiles
d Zig-zag encoding

Figure 5.1 Run Length Coding Approaches

(b)Bit-Plane Coding

An m-bit gray scale image can be converted into m binary images by bit-plane slicing.

The intensities of an m-bit monochrome image can be represented in the form of the base-2 polynomial

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \ldots + a_1 2^1 + a_0 2^0$$

Example:

Let I be the following 2x2 image where the pixels are 3 bits long

101	110
111	011

The corresponding 3 bitplanes are:

1	1	0	1	1	0
1	0	1	1	1	1

An m-bit gray scale image can be converted into m binary images by bit-plane slicing.

These individual images are then encoded using run-length coding.

— Code the bit-planes separately, using RLE (flatten each plane row-wise into a 1D array), Golomb coding, or any other lossless compression technique.

However, a small difference in the gray level of adjacent pixels can cause a disruption of the run of zeroes or ones. Eg: Let us say one pixel has a gray level of 127 and the next pixel has a gray level of 128.

In binary:

127 = 01111111 & 128 = 10000000

Therefore a small change in gray level has decreased the run-lengths in all the bit-planes.

Gray Code

Gray Coded images are free of this problem which affects images which are in binary format.

In gray code the representation of adjacent gray levels will differ only in one bit (unlike binary format where all the bits can change).

Let $g_{m-1}\ldots\ldots g_1 g_0$ represent the gray code representation of a binary number.

Then:

$$g_i = a_i \oplus a_{i+1} \quad 0 \le i \le m-2$$
$$g_{m-1} = a_{m-1}$$

In gray code:

127 = 01000000

128 = 11000000

To convert a binary number $b_1b_2b_3..b_{n-1}b_n$ to its corresponding binary reflected Gray code. Start at the right with the digit b_n. If the b_{n-1} is 1, replace b_n by 1-b_n ; otherwise, leave it unchanged. Then proceed to b_{n-1} . Continue up to the first digit b_1, which is kept the same since it is assumed to be a b_0 =0. The resulting number is the reflected binary Gray code.

Dec	Gray	Binary
0	000	000
1	001	001
2	011	010
3	010	011
4	110	100
5	111	101
6	101	110

7 100 111

Gray Decoding

Decoding a gray coded image The MSB is retained as such, i.e.,

$$a_i = g_i \oplus a_{i+1} \quad 0 \le i \le m-2$$
$$a_{m-1} = g_{m-1}$$

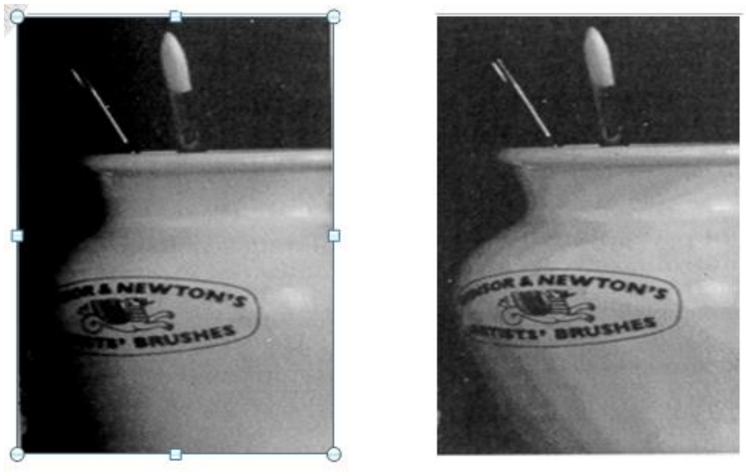
5.3. Psychovisual Redundancy

Human eye does not respond with equal sensitivity to all visual information. Certain information simply has less relative importance than other information in visual processing. This information is said to be *Psychovisual Redundant.*It is associated with real or quantifiable visual information.

Elimination of psychovisually redundant data results in a loss of quantitative information.It is referred as *quantization*. Quantization => Mapping of broad range of input values to a limited number of output values. It is an irreversible operation (Visual Information lost) => Lossy data compression.Consider an 8 bit image with 256 possible gray levels quantized to 16 possible gray levels .

Example First we have a image with 256 possible gray levels . We can apply uniform quantization to four bits or 16 possible levels The resulting compression ratio is 2:1. Note , that false contouring is present in the previously smooth regions of the original image.

The significant improvements possible with quantization that takes advantage of the peculiarities of the human visual system . The method used to produce this result is known as improved gray-scale (IGS) quantization. It recognizes the eye's inherent sensitivity to edges and breaks them up by adding to each pixel a pseudo-random number, which is generated from the order bits of neighboring pixels, before quantizing the result.


a) Original image of 256 gray level b) Uniform quantization to 16 levels

5.4 Image Compression Models

- A typical image compression system consists of Encoder, Channel and Decoder. An input image f(x,y) is fed into the encoder which creates a set of symbols from input data. The decoder should be responsible to decode the encoded representation.
  Encoder - Source Encoder and Channel Encoder.
  Decoder - Source Decoder and Channel Decoder.
- Channel Encoder Increases the noise immunity of the source encoder's output).Channel encoder and decoder play a vital role in the overall encoding and decoding process. Useful channel encoding technique => Hamming Code (Error detection and correction).

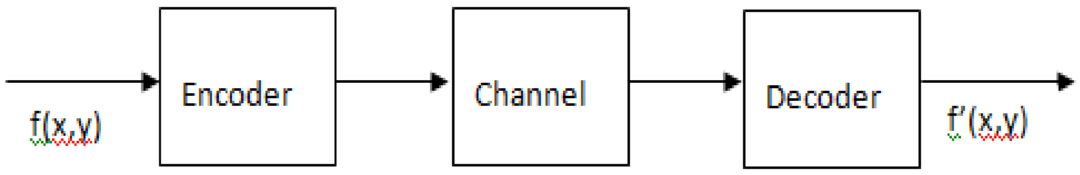


Figure 5.2 A typical image compression system

Source Encode and Decoder

- **First Stage:** the mapper transforms the input data into a format designed to reduce inter-pixel redundancies in the input image. This operation is reversible and may or may not reduce directly the amount of data required to represent the image. Eg. RLE

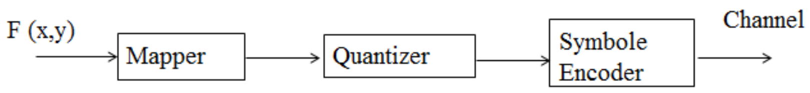


Fig. (a): Source Encoder

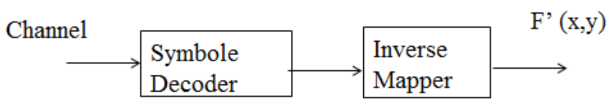
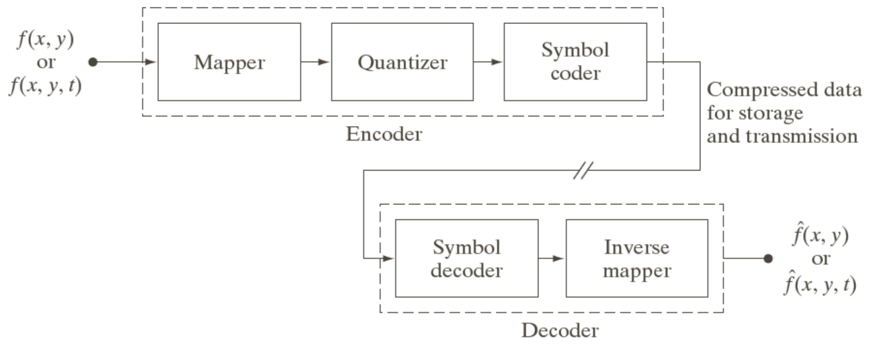


Fig. (b):Source Decoder

- **Second Stage:** Quantizer block reduces the accuracy of the mapper's output in accordance with some pre-established fidelity criterion. This stage reduces the psycho-visual redundancies of the input image.

- **Final Stage:** the symbol coder creates a fixed or variable length code to represent the quantizer output and maps the output in accordance with the code. In most case, VLE code is used.it assigns the shortest code words to the most frequently occurring output values and thus reduces coding redundancies.

- In source encoding process, all 3 operations are not necessarily included in every compression system. No quantizer if there is error free compression. But Source decoder contains only two components, a symbol decoder and an inverse mapper. Because quantizer results in irreversible information loss, an inverse quantize block is not included in the general source decoder model.


$f(x, y)$ or $f(x, y, t)$
Mapper
Quantizer
Symbol coder
Encoder
Compressed data for storage and transmission
Symbol decoder
Inverse mapper
Decoder
$\hat{f}(x, y)$ or $\hat{f}(x, y, t)$

5.5 Lossless and Lossy Compression

Compression refers to a technique where a large file to reduce to smaller sized file and can be decompressed again to the large file. Lossy compression restores the large file to its original form with loss of some data which can be considered as not-noticeable while lossless compression restores the large file to its original form without any loss of data .Following are some of the important differences between Lossy Compression and Lossless Compression.

Sr. No	Aspect	Lossy Compression	Lossless Compression
1	Data Elimination	Lossy compression eliminates those bytes which are considered as not-noticeable.	Lossless compression keeps even those bytes which are not-noticeable.
2	Restoration	After lossy compression, a file cannot be restored to its original form.	After lossless compression, a file can be restored to its original form.
3	Quality	Lossy compression leads to compromise with quality.	No quality degradation happens in lossless compression
4	Size	Lossy compression reduces the size of file to large extent	Lossless compression reduces the size but less as compared to lossy compression.
5	Algorithm used	Transform coding, Discrete Cosine Transform, Discrete Wavelet transform, fractal compression etc.	Run length encoding, Lempel-Ziv-Welch, Huffman Coding, Arithmetic encoding etc.
6	Uses	Lossy compression is used to compress audio, video and images	Lossless compression is used to compress text, images and sound.
7	Capacity	Lossy compression technique has high data holding capacity	Lossless compression has low data holding capacity as compared to lossy compression

5.5.1 Predictive Coding

- It is a statistical estimation procedure where future random variables are predicted from past and present observable variables.
- Image Compression
  - ❖ Lossless Predictive Coding
  - ❖ Lossy Predictive Coding.

Lossless Predictive Coding

- It is based on elimination of inter-pixel redundancy of closely spaced pixels by extracting and coding only the new information.
- The new information => (Actual pixel Value – Predicted Pixel Value).

The basic components of lossless predictive coding system are shown below:

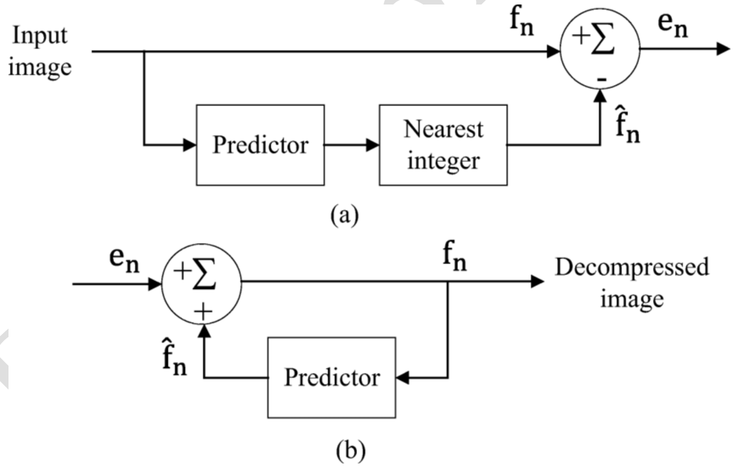


Figure: 5.3 Lossless Predictive Coding Model a) encoder b) decoder.

The encoder and decoder each contain identical predictor. At each successive pixel of input denoted by (f_n) is introduced to the encoder. The predictor generates the anticipated value of that pixel based on some number of past inputs. The output of pixel is rounded to the nearest integer denoted by (f'_n). Prediction error is given by

$e_n = f_n – f'_n$. It is coded using variable-length code

The decoder reconstructs e_n from the received variable length code and performs inverse operation. $f_n = e_n + f'_n$. Prediction is formed by a linear combination of m previous pixels.

$f'_n = \text{round}\left(\sum_{i=1}^{m} \propto_i f_{n-i} \right)$, Where m is the order of the linear predictor ,i for i = 1,2…………m are prediction coefficients.1-D linear predictive coding can be written as

$$f'(x,y) = \text{round}\left[\sum_{i=1}^{m} \propto_i v\ f(x, y-i) \right]$$

Lossy Predictive Coding

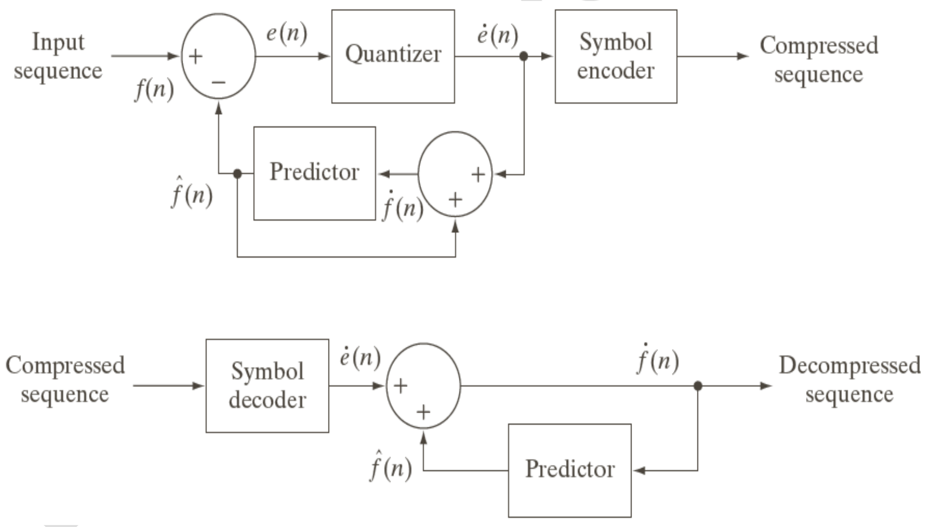
- It is based on the concept of compromising the accuracy of the reconstructed image in exchange for increased compression. The basic components of Lossy predictive coding are shown below.
- 

Figure 5.4 : A Lossy Predictive Coding Model a) encoder b) decoder.

- The Quantizer which absorbs the nearest integer function of the error free encoder is inserted between the symbol encoder and the point at which the prediction error is formed. It maps the prediction error into a limited range of outputs denoted $\dot{e}_n$ which establishes the amount of compression and distortion associated with Lossy prediction coding.

- The error free encoder don't have quantizer so the predictions generated by the encoder and decoder are equivalent. The above fig. shows this accomplished by placing the Lossy encoder's predictor within a feedback loop, where its input denoted by f'_n is generated as a function of past predictions and the corresponding quantized error. The closed loop configuration prevents error build up at the decoder's output.

ie $$f'_n = \dot{e}_n + f''_n$$

Example of Lossy predictive coding: Delta Modulation