

Boundary Segments:

This is a type of boundary representation in which a boundary is decomposed into segments. It is used when the boundary has one or more concavities i.e, curves that carry shape information. The decomposition procedure is used to

- (i) Reduce the complexity of the boundary and
- (ii) Simplify the description process

Convex Hull, H:

A set 'A' is said to be **convex** if the straight line joining any two points in 'A' lies entirely within 'A'. The **convex hull**, H of an arbitrary set 'S' is the smallest convex set containing S. The set difference H- S is called **the convex deficiency** of S. The convex hull of the region enclosed by the boundary is used as a tool for efficient decomposition of the boundary.

Procedure:

Consider the boundary of an object 'S' shown in fig a.

- (1) First, the convex deficiency of the set i.e, object S is defined, which is the shaded region in the fig b
- (2) Next, the contour i.e, outline of S is followed and the points at which there is a transition into or out of the convex deficiency are marked.
- (3) These points are the partitioning points that give the segmented boundary. The result obtained is shown in fig.

Boundary Segmentation

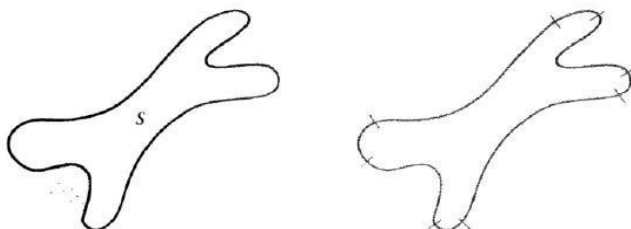


Fig 5.1.1 Representation of Boundary Segments

(Source:Rafael C. Gonzalez, Richard E. Woods, 'Digital Image Processing', Pearson, Third Edition, 2010. - Page – 815)

Advantages:

- The advantage of segmenting a boundary using convex deficiency is that it is independent of the size and orientation of the given region.
- The convex hull and its deficiency can also be used for describing a region and its boundary.
- **Drawback:**
- Factors like digitalization, noise and variations in segmentation may lead to **irregular boundaries** of regions.
- Such boundaries produce convex deficiencies with small, meaningless components scattered randomly throughout the boundary.
- This results in an inefficient decomposition process.

REGIONAL DESCRIPTORS.

Regional descriptors are used to describe image regions, that is regions enclosed by boundaries. In practice, regional descriptors are used with boundary descriptors.

Some important regional descriptors are:

- (1) Simple descriptors
- (2) Texture
- (3) Topological descriptors
- (4) Moments of two-dimensional functions

Simple Descriptors:

The simple descriptors used to describe a region are given below:

(1) Area:

The area of a region is defined as the number of pixels in a region. This is one of the simplest regional descriptors

(2) Perimeter:

The perimeter of a region is defined as the length of its boundary. The perimeter and area used as descriptors when the size of a region is fixed

(3) Compactness:

- The compactness of a region is defined as $(\text{perimeter})^2/\text{area}$.
- Compactness is a dimensionless quantity and therefore it is not sensitive to uniform scale changes.
- It is also insensitive to orientation and thus the error introduced by rotation of a digital region can be avoided.
- Its value is minimum for disk — shaped regions.

(4) Other Measures:

Some other simple measures for describing a region are:

- (i) Mean and median of the gray levels
- (ii) Minimum and maximum gray-level values
- (iii) Number of pixels with values above and below the mean.

TOPOLOGICAL DESCRIPTORS:

Topology is defined as the study of properties of an image which are not affected by any deformation such as stretching, rotation etc.

These properties will change only when there is tearing or joining i.e. folding of the image. Therefore, they are also called rubber — sheet distortions.

Using such properties to describe an image or region is known as the topological description.

Descriptors Used:

Some of the topological descriptors used to describe a region are:

- Number of holes in the region
- Number of connected components of the region etc.

Example:

Consider the region shown in fig. Let its topological descriptor is defined by the number of holes in the region.

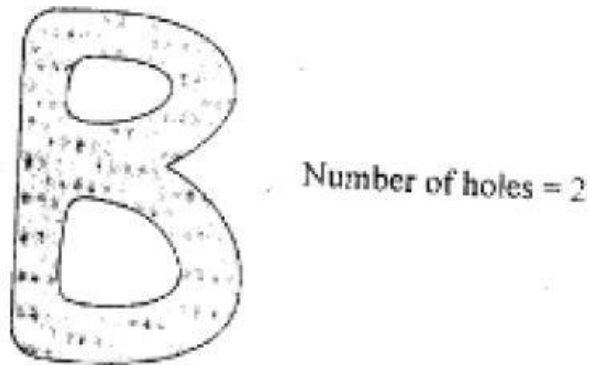


Fig.5.1.2 Example for topological descriptor

(Source: Rafael C. Gonzalez, Richard E. Woods, 'Digital Image Processing', Pearson, Third Edition, 2010. - Page – 823)

This property will not be changed if the figure is stretched or rotated. It will be affected only if the region is torn or folded. Thus, some additional features are provided by the topological descriptors which are useful to characterize the regions.

- The compactness of a region is defined as $(\text{perimeter})^2/\text{area}$.
- Compactness is a dimensionless quantity and therefore it is not sensitive to uniform scale changes.
- It is also insensitive to orientation and thus the error introduced by rotation of a digital region can be avoided.
- Its value is minimum for disk — shaped regions.

(5) Other Measures:

Some other simple measures for describing a region are:

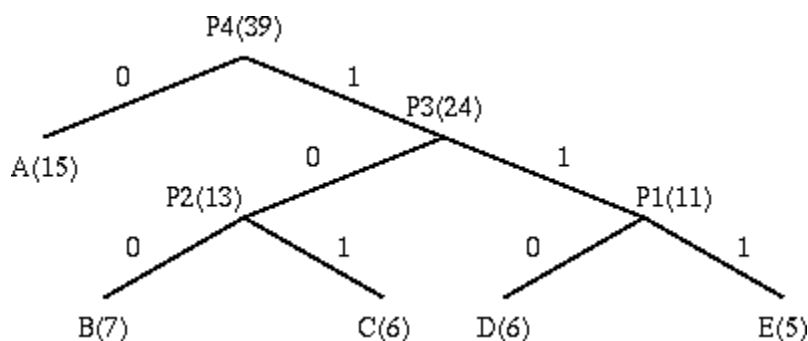
- (i) Mean and median of the gray levels
- (ii) Minimum and maximum gray-level values
- (iii) Number of pixels with values above and below the mean.

Huffman Coding Procedure With an Example.

Huffman coding is based on the frequency of occurrence of a data item (pixel in images). The principle is to use a lower number of bits to encode the data that occurs more frequently. Codes are stored in a *Code Book* which may be constructed for each image or a set of images. In all cases the code book plus encoded data must be transmitted to enable decoding.

A Bottom-Up Approach

1. Initialization: Put all nodes in an OPEN list, keep it sorted at all times (e.g.,ABCDE).
2. Repeat until the OPEN list has only one node left:
 - (a) From OPEN pick two nodes having the lowest frequencies/probabilities, create a parent node of them.
 - (b) Assign the sum of the children's frequencies/probabilities to the parent node and insert it into OPEN.
 - (c) Assign code 0, 1 to the two branches of the tree, and delete the children from OPEN.



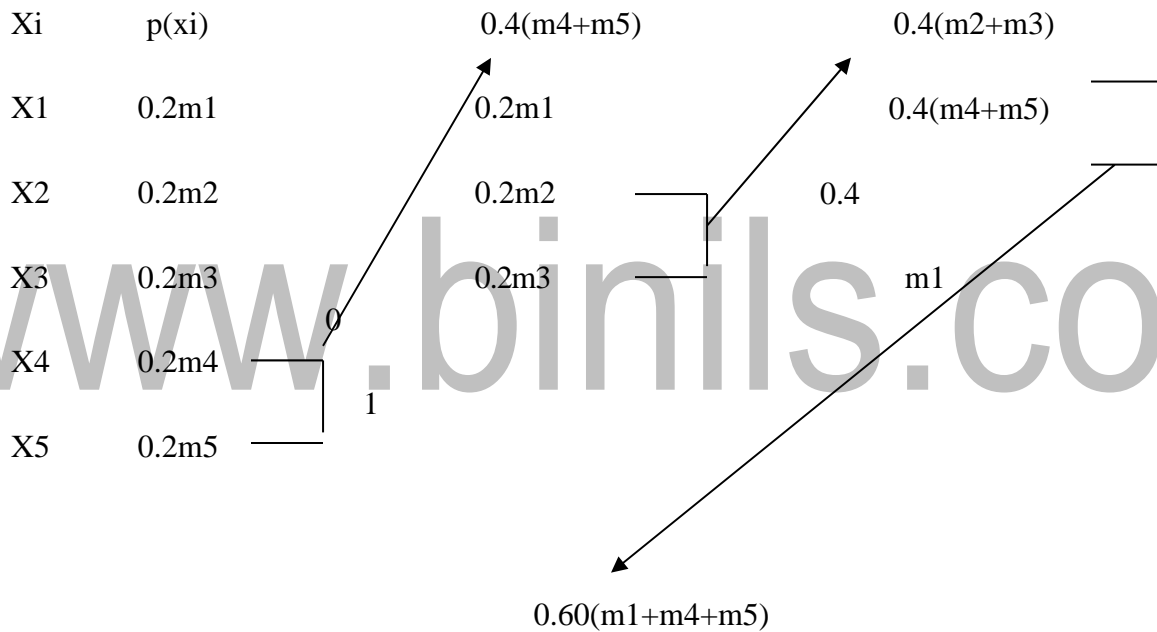
Symbol	Count	$\log(1/p)$	Code	Subtotal (# ofbits)
A	15	1.38	0	15
B	7	2.48	100	21
C	6	2.70	101	18
D	6	2.70	110	18
E	5	2.96	111	15
TOTAL (# of bits):				87

a) **HUFFMANN CODING:**

- i) Order the given symbols in the decreasing probability
- ii) Combine the bottom two or the least probability symbols into a single symbol that replaces in the next source reduction
- iii) Repeat the above two steps until the source reduction is left with two symbols per probability
- iv) Assign '0' to the first symbol and '1' to the second symbol

Problem:

A DMS x has five equally likely symbols $P(x_1) = p(x_2) = P(X_3) = p(x_4) = P(x_5) = 0.2$ construct the Huffman code and calculate the efficiency.



X_i	code	Length
x1	00	2
x2	10	2
x3	11	2
x4	000	3
x5	001	3

$$H(x) = - \sum_{i=1}^5 p(x_i) \log_2 p(x_i)$$

$$= - [(0.2$$

$$\log_2(0.2) \times 5] H(x)$$

$$= 2.32$$

$$L = \sum_{i=1}^5 p(x_i) n_i$$

$$= (0.2)(2+2+2+3+3)$$

$$L = 2.4$$

$$\text{Therefore } \%q = H(x) / L \log_2(M) = 2.32 / 2.4 \log_2(2) = 96.7\%$$

Properties of Huffman coding:

Optimum code for a given data set requires two passes.

1. Code construction complexity $O(N \log N)$.
2. Fast lookup table based implementation.
3. Requires at least one bit per symbol.
4. Average codeword length is within one bit of zero-order entropy (Tighter bounds are known): $H \leq R \leq H + 1 \text{ bit}$
5. Susceptible to bit errors.

ARITHMETIC CODING:

Basic idea:

Map input sequence of symbols into one single codeword

Codeword is determined and updated incrementally with each new symbol (symbol-by-symbol coding)

At any time, the determined codeword uniquely represents all the past occurring symbols. Codeword is represented by a half-open subinterval $[L_c, H) = [0, 1]$. The half-open subinterval gives the set of all code words that can be used to encode the input symbol sequence, which consists of all past input symbols. Any real number within the subinterval $[L_c, H_c]$ can be assigned as the codeword representing the past occurring symbols.

Algorithm:

Let

$S = \{s_0, \dots, s_{(N-1)}\}$ – source alphabet

$p_k = P(s_k)$ – probability of symbol s_k , $0 \leq k \leq (N - 1)$

$[L_c, H_c]$ - Interval assigned to symbol s_k , where $p_k = H_{s_k} - L_{s_k}$

$$L_{s_k} = \sum_{i=0}^{k-1} p_i = P_{k-1}; \quad 0 \leq k \leq (N-1)$$

$$H_{s_k} = \sum_{i=0}^k p_i = P_k; \quad 0 \leq k \leq (N-1)$$

- The subinterval limits can be computed as:

Encoding Procedure:

1. Coding begins by dividing interval $[0,1]$ into N non-overlapping intervals with lengths equal to symbol probabilities p_k
2. $L_c = 0$; $H_c = 1$
3. Calculate code subinterval length: $\text{length} = H_c - L_c$
4. Get next input symbols k
5. Update the code subinterval $L_c = L_c + \text{length} \cdot L_{s_k}$
 $H_c = L_c + \text{length} \cdot H_{s_k}$
6. Repeat from Step 3 until all the input sequence has been encoded

Decoding procedure:

1. $L_c = 0$; $H_c = 1$
2. Calculate code subinterval length: $\text{length} = H_c - L_c$
3. Find symbol subinterval , $0 \leq k \leq (N - 1)$ such that
4. Output symbols k
5. Update the code subinterval

$$L_c = L_c +$$

Length · Lsk

Hc = Lc +

length · Hsk

6. Repeat from Step 2 until last symbol is decoded.

In order to determine when to stop the decoding:

A special end-of-sequence symbol can be added to the source alphabet.

If fixed-length block of symbols are encoded, the decoder can simply

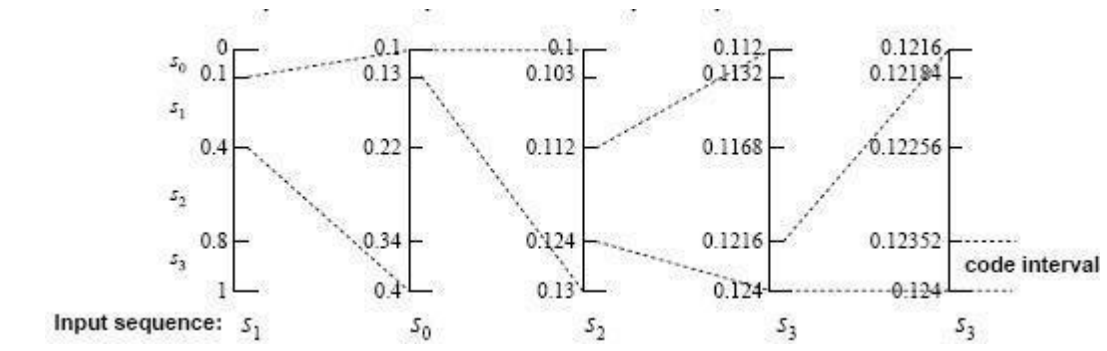
keep a count of the number of decoded symbols

Example:

1. Construct code words using arithmetic coding for the sequence s_0, s_1, s_2, s_3 with probabilities

{0.1, 0.3, 0.4, 0.2}

Source symbol	Probability	Symbol Subinterval $[L_s, H_s)$
s_0	0.1	[0, 0.1)
s_1	0.3	[0.1, 0.4)
s_2	0.4	[0.4, 0.8)
s_3	0.2	[0.8, 1)



Answer:

Iteration # I	Encoded symbol s_k	Code Subinterval $[L_c, H_c)$
1	s_1	[0.1, 0.4)
2	s_0	[0.1, 0.13)
3	s_2	[0.112, 0.124)
4	s_3	[0.1216, 0.124)
5	s_3	[0.12352, 0.124)

www.binils.com

Jpeg Standard

A lossy baseline coding system, adequate for most compression applications

1. An extended coding system for greater compression, higher precision or progressive reconstruction applications
2. A lossless independent coding system for reversible compression

Source: Tutorial Point

Details of JPEG compression Algorithm

- 1) Level shift the original image
- 2) Divide the input image in to 8x8 blocks
- 3) Compute DCT for each block (matrix)(8x8)
- 4) Sort elements of the 8x8 matrix

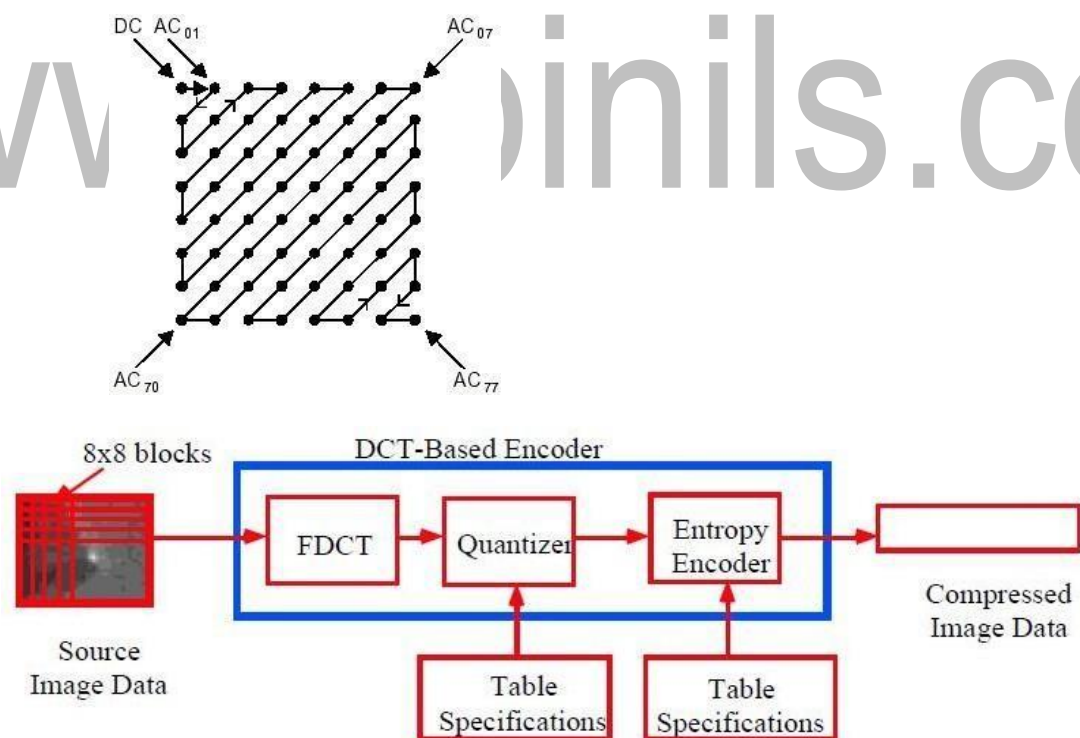


Fig 5.3.1 JPEG Encoder Block Diagram

Details of JPEG Decompression Algorithm

- 1) Compute the reverse order for the output vector
- 2) Perform Huffman decoding next.
- 3) Restore the order of the matrix
- 4) De-normalize the DCT and perform block processing to reconstruct the Original image.
- 5) Level shift back the reconstructed image

JPEG Decoder Block Diagram

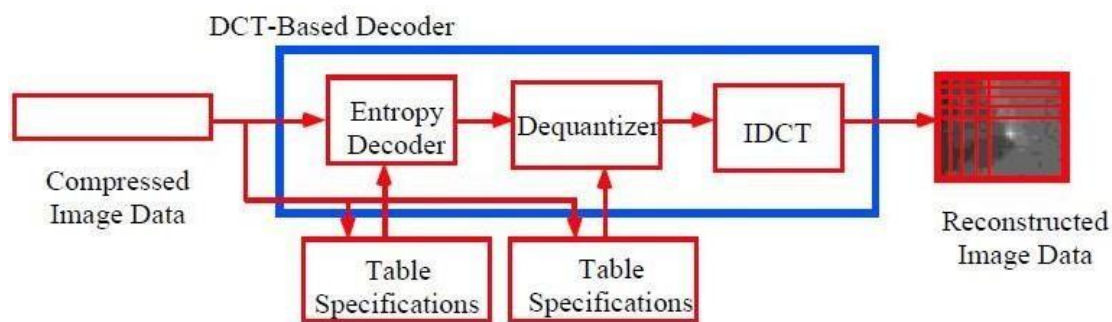


Fig 5.3.2 JPEG decoder Block Diagram

Source: Tutorials point.

2. JPEG2000

JPEG 2000 Encoder Algorithm:

- a. The first step of the encoding process is to DC level shift the pixels of the image by subtracting 2^{m-1} , where 2^m is the number of gray levels of the image
- b. If the input image is a color image, then RGB value is converted into YUV and these components are level shifted individually.
- c. After the image has been level shifted, its components are divided into tiles.
- d. Tiles are rectangular arrays of pixels that contain the same relative portion of all the components. Tile component is the basic unit of the original or reconstructed image.
- e. A wavelet transform is applied on each tile. The tile is decomposed into different resolution levels.

- e. The decomposition levels are made up of sub bands of coefficients that describe the frequency characteristics of local areas of the tile components, rather than across the entire image component.
- f. Thus 1D-DWT of the rows and columns of each tile component is then computed.

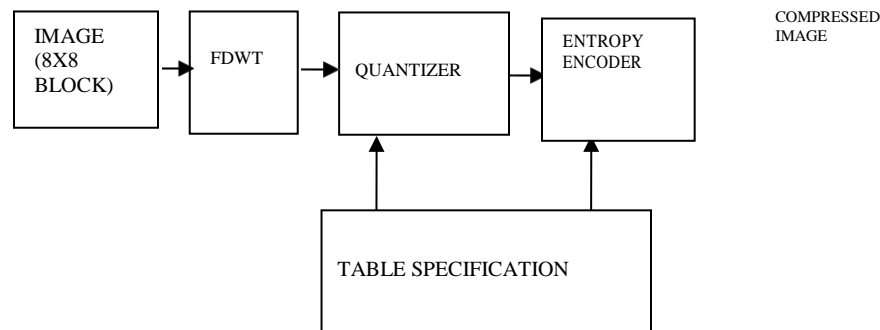


Fig: 5.3.3 Main structure of JPEG2000encoder.

Source: Tutorials point.

- g This involves six sequential lifting and scaling operations.

$$Y(2n+1)=X(2n+1)+\alpha [X(2n)+X(2n+2)],i_o-3\leq 2n+1\leq i_1+3$$

$$Y(2n)=X(2n)+\beta [Y(2n-1)+Y(2n+1)],i_o-2\leq 2n\leq i_1+2$$

$$Y(2n+1)=Y(2n+1)+\gamma [Y(2n)+Y(2n+2)],i_o-1\leq 2n+1\leq i_1+1$$

$$Y(2n)=Y(2n)+\delta [Y(2n-1)+Y(2n+1)],i_o\leq 2n\leq i_1 \quad Y(2n+1)=(-k).Y(2n-1),i_o\leq 2n+1\leq i_1$$

$$Y(2n)=Y(2n)/K,i_o\leq 2n\leq i_1$$

X = input tile component

Y = resulting transform coefficients

io and i1 = represent the position of the tiles component within a component.

JPEG 2000 Decoder Algorithm:

- a. Decode the bit modeled or arithmetically bit streams
- b. Dequantize the coefficients using,

$$R(u, v) = \begin{cases} (q(u, v) + 2^{\frac{M-N(u,v)}{b}}) \cdot \Delta q(u, v) > 0 \\ (q(u, v) - 2^{\frac{M-N(u,v)}{b}}) \cdot \Delta q(u, v) < 0 \\ 0, & \bar{q}(u, v) = 0 \end{cases}$$

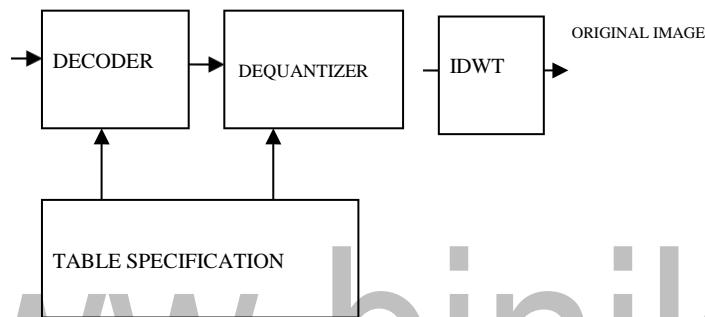


Fig: 5.3.4 Main structure of JPEG 2000 decoder.

Source: Tutorials point.

- a. The dequantized coefficients are then inverse transformed by column and by row using inverse forward wavelet transform filter bank or using the following lifting based operations

b. $X(2n) = k \cdot Y(2n), i_0 - 3 \leq 2n \leq i_1 + 3$

c. $X(2n+1) = (-1/k) \cdot Y(2n+1), i_0 - 2 \leq 2n+1 \leq i_1 + 2$

$$X(2n) = X(2n) - \delta[X(2n-1) + X(2n+1)], i_0 - 3 \leq 2n \leq i_1 + 3$$

$$X(2n+1) = X(2n+1) - \gamma[X(2n) + X(2n+2)], i_0 - 2 \leq 2n+1 \leq i_1 + 2$$

$$X(2n) = X(2n) - \beta[X(2n-1) + X(2n+1)], i_0 - 1 \leq 2n \leq i_1 + 1$$

$$X(2n+1) = X(2n+1) - \alpha[X(2n) + X(2n+2)], i_0 \leq 2n+1 \leq i_1$$

- d. Perform DC level shifting by adding 2^{m-1}

Image Compression and Recognition

Compression Types

Now consider an encoder and a decoder as shown in Fig.. When the encoder receives the original image file, the image file will be converted into a series of binary data, which is called the bit-stream.

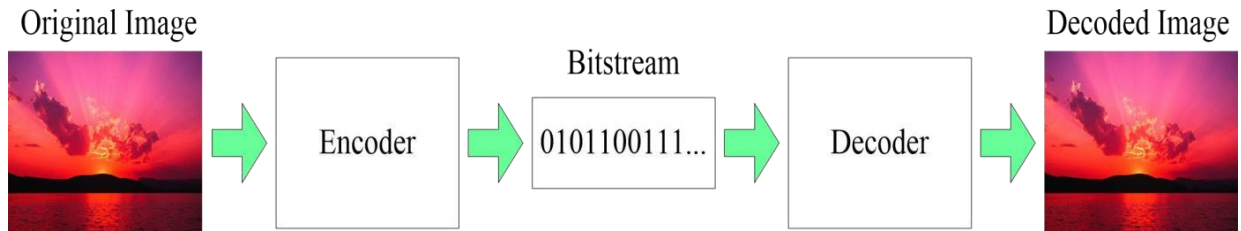


Fig.5.2.1The basic flow of image compression coding

Source: Tutorial point

The decoder then receives the encoded bit-stream and decodes it to form the decoded image. If the total data quantity of the bit-stream is less than the total data quantity of the original image, then this is called image compression.

Lossless compression can recover the exact original data after compression. It is used mainly for compressing database records, spreadsheets or word processing files.

There is no information loss, and the image can be reconstructed exactly the same as the original

Applications: Medical imagery

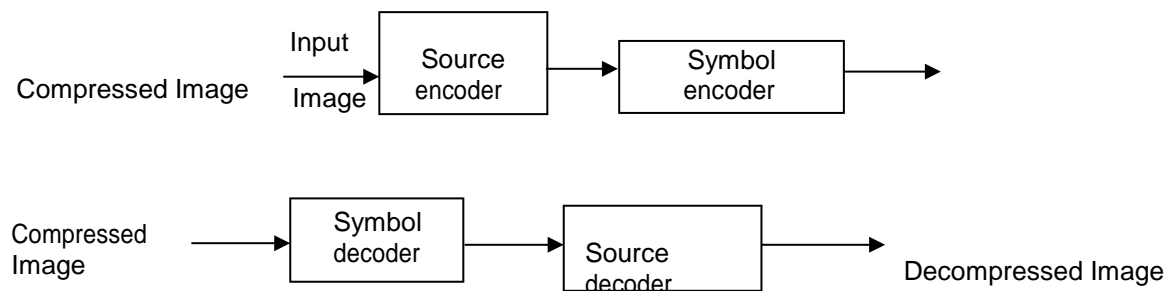


Fig5.2.2: The block diagram of encoder& decoder of Lossless compression

Source: Tutorial point

1. Source encoder is responsible for removing the coding and inter pixel redundancy and psycho visual redundancy.
2. Symbol encoder- This reduces the coding redundancy .This is the final stage of encoding process.

Source decoder- has two components

- a) Symbol decoder- This performs inverse operation of symbol encoder.
- b) Inverse mapping- This performs inverse operation of map.

Lossless Compression technique

Variable length coding (Huffman , arithmetic),

LZW coding, Bit Plane coding, Lossless Predictive coding

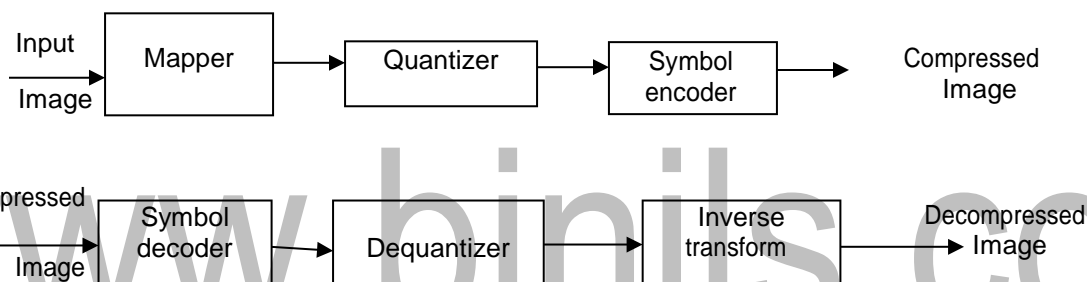


Fig5.2.3 The Block Diagram Of Encoder& Decoder Of Lossy compression

Source: Tutorial point

Lossy compression will result in a certain loss of accuracy in exchange for a Substantial increase in compression.

Lossy compression is more effective when used to compress graphic images

1. Information loss is tolerable
2. Many-to-1 mapping in compression eg. quantization
3. Applications: commercial distribution (DVD) and rate constrained environment

where lossless methods can not provide enough compression ratio

1) Mapper -this transforms the input data into non-visual format. It reduces the interpixel redundancy.

2) Quantizer - It reduces the psycho visual redundancy of the input images .This step is omitted if the system is error free.

3) Symbol encoder- This reduces the coding redundancy .This is the final stage of encoding process.

Source decoder- has two components

a) Symbol decoder- This performs inverse operation of symbol encoder.

b) Inverse mapping- This performs inverse operation of mapper.

Channel decoder-this is omitted if the system is error free.

MPEG COMPRESSION STANDARD.

Video Compression standards:

Video compression is used for compression of moving picture frames.

Based on the applications, video compression standards are grouped into two categories,

(1) Video teleconferencing standards

(2) Multimedia standards

The International Telecommunication Union (ITU) has a number of video conferencing compression standards such as, H.261, H.262, H.263 and H.320

1.H.261 is used to support full motion video transmission over T1 lines

2.H.263 is designed for very low bit rate video in the range of 10 to 30 Kbits/sec

3.H.320 is used for ISND

b.Multimedia Standards

These are used for video on demand, digital HDTV broadcasting and image/video database services.

The principal multimedia compression standards are MPEG-1, MPEG-2 and MPEG-4

1. MPEG-1 is an entertainment quality coding standard for the storage and retrieval of video on digital media such as CDROMs. It supports a bit rate of 1.5 Mbit/s. Thus it is used for compression of low resolution.
2. MPEG-2 is used for cable TV distribution and narrow channel satellite broadcasting with a bit rate of 2 to 10 Mbit/s. Thus is used for higher resolution standards, mainly for studio quality audio and video compression.
3. MPEG-4 provides 5 and 64 Kbit/s for mobile and PSTN and upto 4 Mbit/s for TV and film applications.
4. MPEG -4 provides (a) improved video compression efficiency; (b) Content based interactivity and (c) universal access including increased robustness.

MPEG Algorithm:

MPEG standard consists of both video and audio compression. The MPEG algorithm relies on two basic techniques

Block based motion compensation DCT based compression

1. The video signal is sequence of still picture frames. From the picture, slices are formed in the raster scan order as shown below,
2. For each slice the macro blocks are obtained. The macro block consists of four blocks of 8 x 8 pixels. Thus the micro block has a size of 16 x 16 pixels. If the macro block is in RGB format, then it is converted into Y Cr Cb format, where Cr and Cb represent chrominance signals. Once the sequence of macro blocks is formed, coding can take place.

3. The motion is estimated by predicting the current frame on the basis of certain previous and/or forward frame. The information sent to the decoder consists of the compressed DCT coefficients of the residual block together with the motion vector. There are three types of pictures in MPEG: Intra-frames (I) Predicted frames (P) Bidirectional predicted frames (B)

Figure demonstrates the position of the different types of pictures. Every Nth frame in the video sequence is an I-picture, and every Mth frame a P-picture. Here $N=12$ and $M=4$. The rest of the frames are B-pictures.

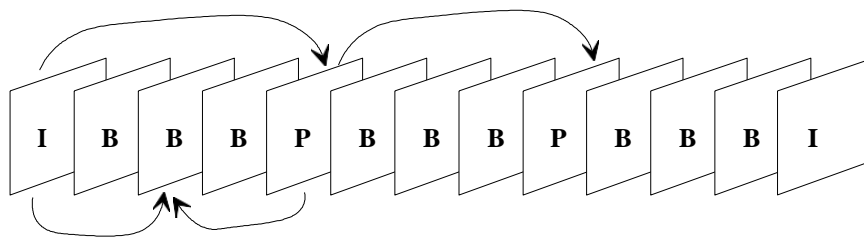


Figure 5.2.4: Interframe coding in MPEG.

Source: Tutorial point

Compression of the picture types:

1. Intra frame (I –frame) are coded as still images by DCT algorithm. They provide access points for random access, but only with moderate compression. It is used as the reference point for the motion estimation.

2. Predicted frame (P-frame) are coded with reference to a past picture. The current frame is predicted on the basis of the previous I- or P-picture. The residual (difference between the prediction and the original picture) is then compressed by DCT.

3. Bidirectional frame (B-frame) is the compressed difference between the current frame and a prediction of it based on the previous I- or P- frame and next P-frame. Thus the decoder must have access to both the past and future reference frames. The encode frames are therefore recorded before transmission and the decoder

reconstructs and displays them in the proper sequence. Bidirectional pictures are never used as reference.

1. After the pictures are divided into 1616 macro blocks, each consisting of four 88 elementary blocks, the DCT of the macro block is calculated whose coefficients are quantized.

The choice of the prediction method is chosen for each macro block separately.

1.The intra-coded blocks are quantized differently from the predicted blocks:

Intra-coded blocks contain information in all frequencies and are quantized differently from the predicted blocks .The predicted blocks, contain mostly high frequencies and can be quantized with coarser quantization tables.

Motion estimation and compensation:

1.The prediction block in the reference frame is not necessarily in the same coordinates than the block in the current frame.

2.Because of motion in the image sequence, the most suitable predictor for the current block may exist anywhere in the reference frame.

3.The motion estimation specifies where the best prediction (best match) is found.

4.The prediction block in the reference frame is not necessarily in the same coordinates than the block in the current frame.

5.Because of motion in the image sequence, the most suitable predictor for the current block may exist anywhere in the reference frame.

6.The motion estimation specifies where the best prediction (best match) is found.

4.Motion compensation consists of calculating the difference between the reference and the current block.

5.Finally the quantized DCT coefficients are then run – length encoded in a zig-zag order and then huffmann coded to compress the data.

6. Also the encoder is designed to generate a bit stream that matches the capacity of

the intended video channel. This is done by using a rate controller which adjusts the quantization parameters as a function of the occupancy of the output buffer.

7. As the buffer becomes fuller the quantization is made coarser, so that fewer bits stream into buffer.

Three basic types of encoded output frames:

- 1. Intra frame or independent frame (I-frame).** An I frame is compressed independently of all previous and future video frames. Of the three possible encoded output frames, it most highly resembles a JPEG encoded image. Moreover, it is the reference point for the motion estimation needed to generate subsequent P- and B-frames.
- 2. Predictive frame (P-frame).** A P-frame is the compressed difference between the current frame and a prediction of its based on the previous I-or P-frame. The difference is formed in the leftmost summer of fig. The prediction is motion compensated and typically involves sliding the decoded block in the lower part of fig. around its immediate neighborhood in the current frame and computing a measure of correlation (such as the sum of the square of the pixel-by-pixel differences). In fact, the process is often carried out in sub pixel increments (such as sliding the sub image $\frac{1}{4}$ pixels at a time), which necessitates interpolating pixel values prior to computing the correlation measure. The computed motion vector is variable length coded and transmitted as an integral part of the encoded data stream. Motion estimation is carried out on the macro block level.
- 3. Bidirectional frame (B- frame).** A B- frame is the compressed difference between the current frame and a prediction of it based on the previous I- or P- frame and next- P- frame. Accordingly, the decoder must have access to both past and future reference frames. The encoded frames are therefore reordered before transmission; the decoder reconstructs and displays them in the proper sequence.

Patterns and Pattern Classes:

A pattern is an arrangement of descriptors,

The name *feature* is used often in the pattern recognition literature to denote a descriptor. A *pattern class* is a family of patterns that share some common properties. Pattern classes are denoted $\omega_1, \omega_2, \dots, \omega_W$, where W is the number of classes. Pattern recognition by machine involves techniques for assigning patterns to their respective classes— automatically and with as little human intervention as possible. Three common pattern arrangements used in practice are vectors (for quantitative descriptions) and strings and trees (for structural descriptions). Pattern vectors are

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

represented by bold lowercase letters, such as \mathbf{x} , \mathbf{y} , and \mathbf{z} , and take the form

where each component, x_i , represents the i th descriptor and n is the total number of such descriptors associated with the pattern. Pattern vectors are represented as columns (that is, $n \times 1$ matrices). Hence a pattern vector can be expressed in the form shown in Eqn. (1) or in the equivalent form $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, where T indicates transposition. The nature of the components of a pattern vector \mathbf{x} depends on the approach used to describe the physical pattern itself. Let us illustrate with an example that is both simple and gives a sense of history in the area of classification of measurements. In our present terminology, each flower is described by two measurements, which leads to a 2-D pattern vector of the form

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

where x_1 and x_2 correspond to petal length and width, respectively. The three pattern classes in this case, denoted ω_1 , ω_2 , and ω_3 , correspond to the varieties *setosa*, *virginica*, and *versicolor*, respectively. Because the petals of flowers vary in width and length, the pattern vectors describing these flowers also will vary, not only between different classes, but also within a class. The above Figure shows length and width measurements for several samples of each type of iris. After a set of measurements has been selected (two in this case), the components of a pattern vector become the entire description of each physical sample. Thus each flower in this case becomes a point in 2-D Euclidean space. We note also that measurements of petal width and length in this case adequately separated the class of *Iris setosa* from the other two but did not separate as successfully the *virginica* and *versicolor* types from each other. This result illustrates the classic *feature selection* problem, in which the degree of class separability depends strongly on the choice of descriptors selected for an application.

RECOGNITION BASED ON MATCHING:

Recognition techniques based on matching represent each class by a prototype pattern vector. An unknown pattern is assigned to the class to which it is closest in terms of a predefined metric. The simplest approach is the minimum distance classifier, which, as its name implies, computes the (Euclidean) distance between the unknown and each of the prototype vectors. It chooses the smallest distance to make a decision. We also discuss an approach based on correlation, which can be formulated directly in terms of images and is quite intuitive.

MINIMUM DISTANCE CLASSIFIER

Suppose that we define the prototype of each pattern class to be the mean vector of

$$m_j = \frac{1}{N_j} \sum_{x \in \omega_j} x_j \quad j = 1, 2, \dots, W.$$

the patterns of that class:

where N_j is the number of pattern vectors from class ω_j and the summation is taken over these vectors. As before, W is the number of pattern classes. One way to determine the class membership of an unknown pattern vector x is to assign it to the class of its closest prototype, as noted previously. Using the Euclidean distance to determine closeness reduces the problem to computing the distance measures.

The decision boundary between classes ω_i and ω_j for a minimum distance classifier is

The surface given by the above equation is the perpendicular bisector of the line segment joining m_i and m_j . For $n = 2$, the perpendicular bi-sector is a line, for $n = 3$ it is a plane, and for $n > 3$ it is called a hyper plane. The two classes, Iris versicolor and Iris setosa, denoted ω_1 and ω_2 , respectively, have sample mean vectors $m_1 = (4.3, 1.3)^T$ and $m_2 = (1.5, 0.3)^T$. The decision functions are

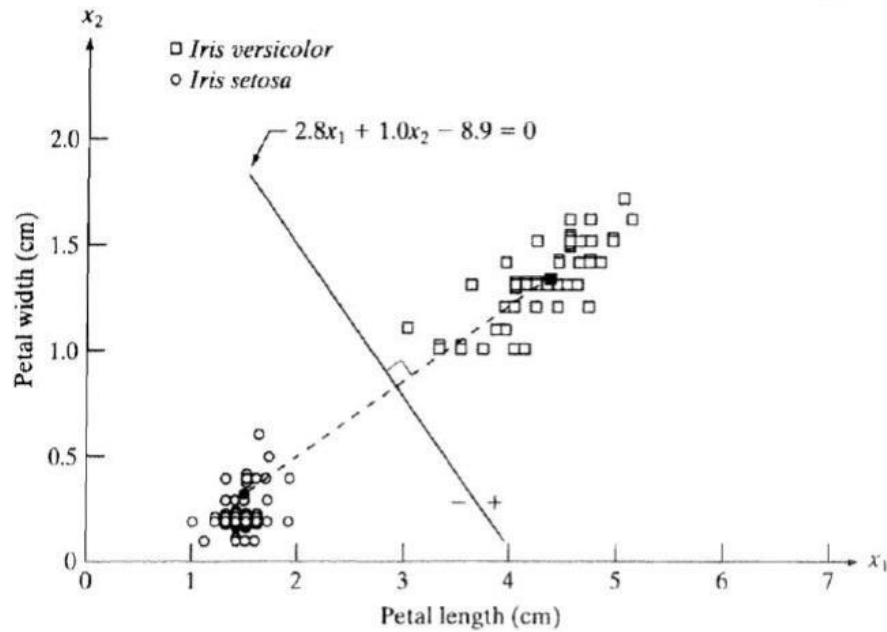


Fig.5.4.1 plot of this boundary

(Source: Rafael C. Gonzalez, Richard E. Woods, 'Digital Image Processing', Pearson, Third Edition, 2010. - Page – 862)

The above figure shows a plot of this boundary (note that the axes are not to the same scale). Substitution of any pattern vector from class ω_1 would yield $d_{12}(x) > 0$.

- Conversely, any pattern from class ω_2 would yield $d_{12}(x) < 0$.
- In other words, given an unknown pattern belonging to one of these two classes, the sign of $d_{12}(x)$ to one of these two classes, the sign of $d_{12}(x)$ would be sufficient to determine the pattern's class membership.