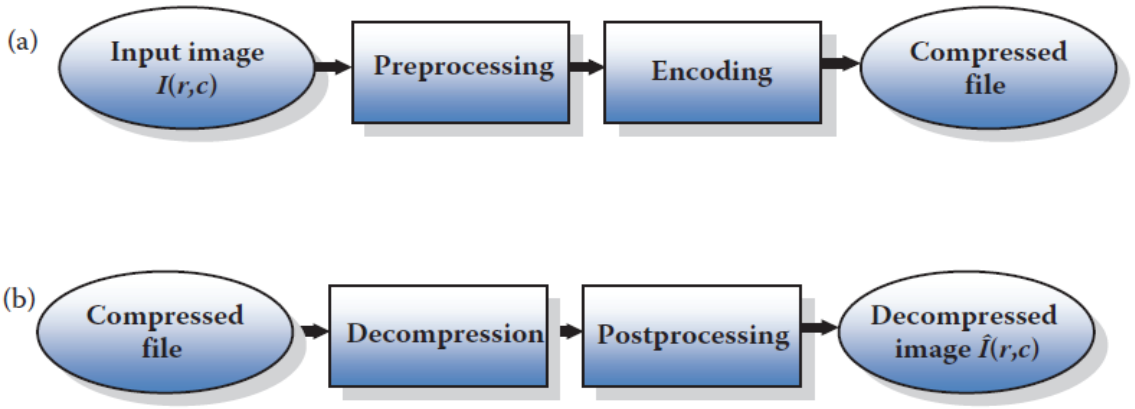
Image compression

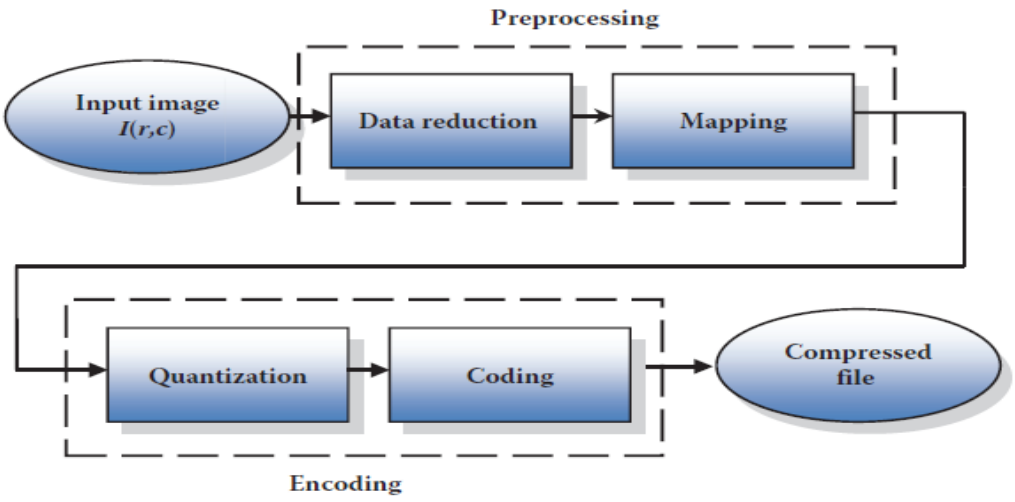
1. Introduction

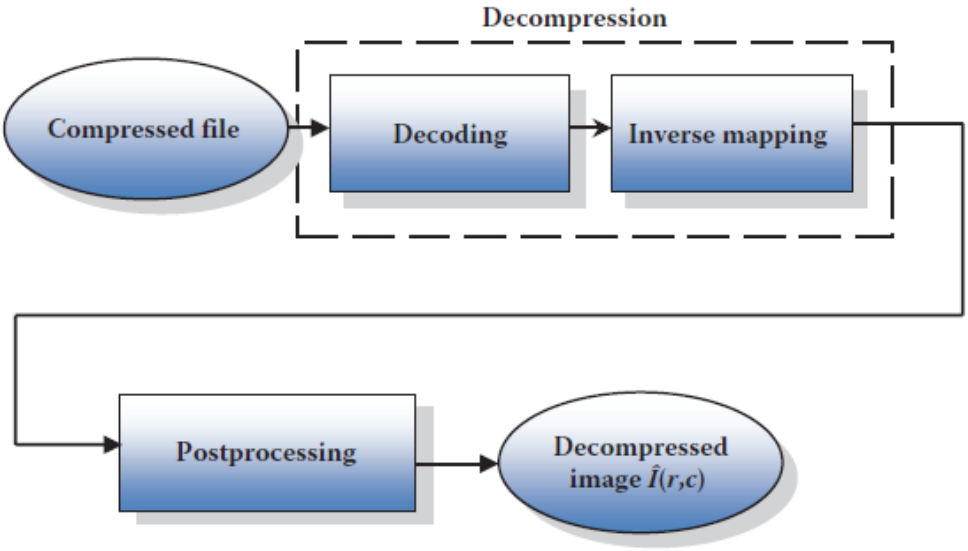
Image compression has been pushed to the forefront of the image-processing field. This is largely a result of the rapid growth in computer power, the corresponding growth in the multimedia market, and the advent of the *World Wide Web*, which makes the Internet easily accessible for every one.

Image data compression, bit rate reduction and data reduction are all terms that mean basically the same thing. In essence the same information is stored or carried using smaller quantity of data or rate of data. Data compression is the process of reduction the amount of data (number of bit) required to represent a given quantity of information . Compression is a special case of a more general technique known as encoding.



Compression system model. (a) Compression, (b) decompression.





Encoding can be described as the process that takes as input a string of symbols, assigns codes for each input symbol and gives as output a string of code . A compression algorithm and its corresponding decompression algorithm are collectively referred to as code. Compression methods are essentially used for image application such as image transmission, video conferencing, facsimile transmission of printed material, and graphics image or transmission of remote sensing images obtained form satellites. An other area for the application of efficient coding is, picture storing in database such as archiving medical images, multi spectral image, fingerprints, and drawing .Data compression is possible because images are extremely data intensive , and contain a large amount of redundancy which can be removed by accomplishing some kind of transform, with a reversible linear phase to de-correlate the image data pixels.

Image compression involves reducing the size of image data file while retaining necessary information .The reduced file is called the compressed file and is used to reconstruct the image , resulting in the decompressed image . The original image , before any compression is performed , is called the uncompressed image file . The ratio of the original, uncompressed image file to the compressed image file is referred to as the compression ratio .

The key to a successful compression comes with the second part of the definition, retaining necessary information. To understand this one must differentiate between data and information. In fact, data are the means by which information is conveyed. Various amounts of data may be used to represent the same amount of information.. If the two individuals use a different number of words to tell the same basic story, two different version of the story is created and at least one includes nonessential data. It is thus said to contain data redundancy.

Data redundancy is a central issue in digital image compression, it is not an abstract concept but mathematically quantifiable entity. In digital image compression, there are three basic data redundancies: *Coding redundancy, Inter pixel redundancy and Psychovisual redundancy* .

Data compression is achieved when one or more of these redundancies are reduced or eliminated. The reduced information in the first two types of redundancy is generally referred as Redundant Information. It is the information that can be obtained from other pieces of information existing in a picture or video sequence. The reduced information in the third type of redundancy is referred as *Irrelevant Information*. It is the information, which is impossible or difficult for the viewer to perceive, or even the information, which has been classified as less important.

There are two types of image compression methods according to the information preserved. The first type is called *loss less* methods because no data are lost, and the original image can be recreated exactly from the compressed data. The second type of the compression methods is called *lossy* because they allow a loss in the actual image data .

2.Type Of Redundancy

The redundancy that is inherent in image data is used to develop compression algorithms. Three primary types of redundancy can be found in images: (*Coding redundancy, Inter pixel redundancy and Psychovisual redundancy*) .

a. Coding Redundancy

It is the redundancy that occurs when the data used to represent the image are not utilized in optimal way. That is , it occurs when the gray level of an image is coded in away that uses more codes than absolutely necessary to represent each gray level. For example, if an 8-bit/pixel image, which allows 256 different intensity level is used to represent a 16 levels, only 4-bit/pixel actually are needed to represent the image. In general, coding redundancy is present when the codes assigned to the set of gray level have not been selected to take the full advantage of the probabilities of gray level .

b. Inter Pixel Redundancy

It occurs because adjacent pixel tend to be highly correlated. This is a result of the fact that in most image the brightness level does not change rapidly, but change gradually. So that adjacent pixel value tend to be relatively close to each other. Therefore, much of the visual contribution of a single pixel to an image is redundant, it could have been guessed on the basis of its neighbor's value. A variety of names including spatial redundancy, geometric redundancy and inter frame redundancy, have been coined to refer to these inter pixel dependencies . Inter pixel redundancy can be divided into two categories, *Spatial redundancy and Temporal redundancy.*

C. Psychovisual Redundancy

It refers to the fact that some information is more important to the human visual system than other types of information .The human perception of information in an image normally does not involve quantitative analysis of every pixel or luminance value in the image. In general the observer searches for distinguishing features such as edges or textural regions and mentally combines them to recognize regions

3.Measure The Efficiency Of The System

Any image compression system has some fundamentals that measure the efficiency of the system. Three measurements are considered: the *compression ratio*, the *entropy* and the *average length.*

3.1.The Compression Ratio

Image compression involves reducing the size of image data files, while retaining necessary information. The reduced file is called the compressed file and is used to reconstruct the image, resulting in the decompressed image. The original image, before any compression is performed, is called the uncompressed image file. The ratio of the original, uncompressed image file and the compressed file is referred to as the compression ratio(CR). The compression ratio is denoted by equation (2-1).

$$\textbf{Compression ratio(CR)} = \frac{\textbf{Uncompressed file size}}{\textbf{Compressed file size}} \qquad \textbf{(2.1)}$$

Ex.1\

The original image is 256 × 256 pixels, single-band (gray scale), 8-bits per pixel. This file is 65,536 bytes (64k). After compression the image file is 6554 bytes. The compression ratio is SIZEU/SIZEC = 65,536/6554 = 9.999 ≈ 10. This can also be written as 10:1.

Ex.2\

Using the preceding example, with a compression ratio of 65,536/6554 bytes, we want to express this as bits per pixel. This is done by first finding the number of pixels in the image: 256 × 256 = 65,536 pixels. We then find the number of bits in the compressed image file: (6,554 bytes)(8 bits/byte) = 52,432 bits. Now we can find the bits per pixel by taking the ratio: 52,432/65,536 = 0.8 bits/pixel.

To transmit an RGB (color) 512 × 512, 24-bit (8-bit per pixel per color) image via modem at 56 (kilo-bits per second), it would take about

$$\frac{(512 \times 512 \text{ pixels})(24 \text{ bits/pixel})}{(56 \times 1024 \text{ bits/second})} \approx 109 \text{ seconds} \approx 1.8 \text{ minutes}$$

3.2. The Entropy

The information of a message or an event to be transmitted is a decreasing function of its probability. This means that the less probable a message, the larger its information, and the more probable a message, the smaller its information. If there are a set of M random events $[\alpha_1, \alpha_2, \alpha_3, \ldots\ldots, \alpha M]$ with probabilities [P1=P (α 1), P_2 = P (α 2)……. P_M =P (α M)], then the information content of the (*ith*) event is called the *self- Information* and is given by:

$I_i = - \text{Log}_2 (P_i)$ in bits/symbol (2-2)

The average of the information content of all possible events, is known as the *Entropy* and is defined as

$$H = - \sum_{i=1}^{M} P_i \text{Log}_2 P_i \quad \text{bits/ symbol} \quad (2\text{-}3)$$

Entropy is a measure of the degree of randomness of the set of random variables. The least random case is when one of the random variable has probability 1 so that the outcome is known in advance and (H = 0). The most random case is when all events are equally likely. In this case [P_1=P_2=……= P_M=1/M] and (H=Log_2 M). In coding application, entropy represents the amount of information associated with the set of coder input values and give a lower bound on the average number of bits required to code those inputs.

Ex.3\

Let L = 8, meaning there are 3 bits/pixel in the original image. Now, let's say the number of pixels at each gray-level value is equal (they have the same probability); that is

P0=P1=.....=P7= 1/8

$$Entropy = -\sum_{i=0}^{7} p_i \log_2(p_i) = -\sum_{i=0}^{7} \frac{1}{8} \log_2\left(\frac{1}{8}\right) = 3$$

This tells us that the theoretical minimum for lossless coding for this image is 3 bits per pixel. In other words, there is no code that will provide better results than the one currently used (called the natural code, since 000_2 = 0, 001_2 = 1, 010_2 = 2, ..., 111_2 = 7). This example illustrates that the image with the most random distribution of gray levels, a uniform distribution, has the highest entropy

Ex.4\

Let $L = 8$, thus we have a natural code with 3 bits per pixel in the original image. Now let's say that the entire image has a gray level of 2, so

$$p_2 = 1, \text{ and } p_0 = p_1 = p_3 = p_4 = p_5 = p_6 = p_7 = 0$$

and the entropy is

$$Entropy = -\sum_{i=0}^{7} p_i \log_2(p_i) = -(1)\log_2(1) + 0 + \ldots + 0 = 0$$

This tells us the theoretical minimum for coding this image is 0 bits per pixel. Why is this? Because the gray-level value is known to be 2. To code the entire image we need only one value, this is called the certain event, it has a probability of 1.

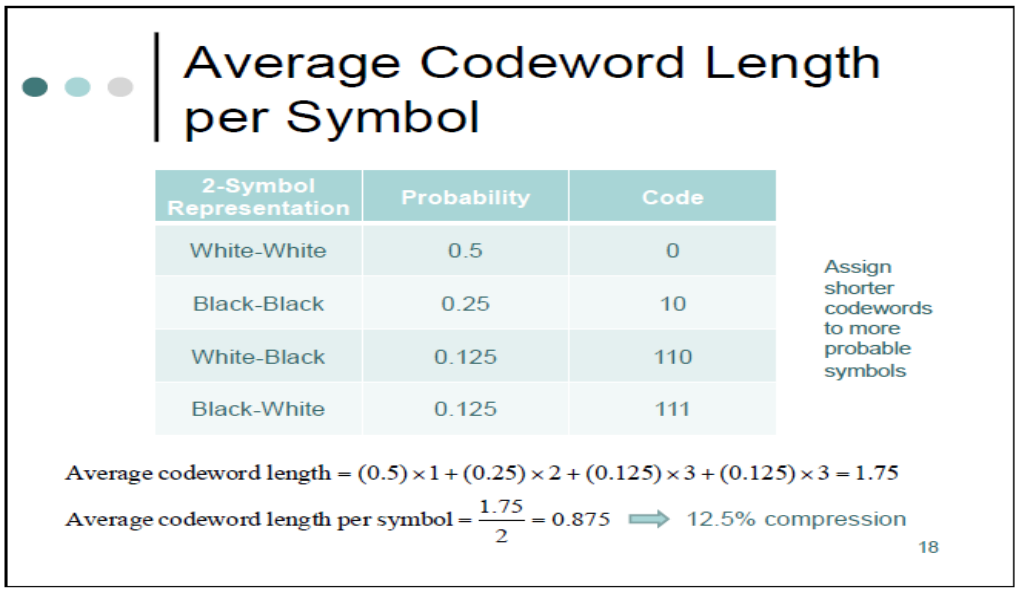
3.3.The Average Length

Let $g_1, g_2 \ldots g_L$ represent the gray levels of an image with probabilities $P_1, P_2 \ldots P_L$. If the code used such as (Huffman code) assigns a number of bits l_i (where l_i is an integer number) to represent (*ith*) gray level, then the average number required is:

$$L_{avg} = \sum_{i=1}^{l} l_i P_i \qquad (2\text{-}4)$$

Where l_i= code length in bits of the ith gray level.

Pi= histogram probability of ith gray level.



4.Image Fidelity Criteria

To evaluate the quality of the reconstructed image result from compression system , some measurement must be adopted .Two types of criteria are used for evaluation image quality :*Objective* and *Subjective.*

4.1.Objective Fidelity Criteria

In the objective fidelity criteria the level of information loss can be expressed as a function of original or input image and the decompressed output image. Digital signals processing and information theory define the amount of error in the reconstructed (decompressed) image by the equation below .

$$\text{Error}(r,c) = \hat{f}(r,c) - f(r,c) \qquad (2\text{-}5)$$

Where the f(r,c) is the original input image and $\hat{f}(r,c)$ is the decompressed output image.

One of the most commonly used objective measures is the root-mean-square error ($\mathbf{e_{rms}}$), which is defined as:

$$e_{rms} = \sqrt{\frac{1}{N^2}\sum_{r=0}^{N-1}\sum_{c=0}^{N-1}[\hat{f}(r,c) - f(r,c)]^2} \qquad (2\text{-}6)$$

A closely related objective fidelity criterion is the mean-square signal to noise ratio (SNR_{RMS}), where SNR considers the decompressed image $\hat{f}(r,c)$ to be the "signal" and the error to be noise and is defined as:

$$SNR_{RMS} = \sqrt{\frac{\sum_{r=0}^{N-1}\sum_{c=0}^{N-1}[\hat{f}(r,c)]^2}{\sum_{r=0}^{N-1}\sum_{c=0}^{N-1}[\hat{f}(r,c) - f(r,c)]^2}} \qquad (2\text{-}7)$$

Another related metric, is the peak signal-to-noise, is defined as

$$(PSNR)_{dB} = 10\text{Log}\,10\,\frac{(L-1)^2}{\frac{1}{N^2}\sum_{r=0}^{N-1}\sum_{c=0}^{N-1}[\hat{f}(r,c) - f(r,c)]^2} \qquad (2\text{-}8)$$

Where L is a number of gray level.

For the ($\mathbf{e_{rms}}$) metric, a smaller value means better image quality, while for SNR and PSNR, a higher value leads to higher image quality. The objective measures are often used in researches because they are easy to generate and offer a simple and convenient mechanism for evaluation of information loss. It is important to note that these metrics are not necessarily correlated to the human perception of an image

4.2. ***Subjective Fidelity Criteria***

Subjective testing is performed by creating a database of images to be tested, gathering a group of people that are representative of the desired population, and then having all the test subjects evaluate the images according to a predefined scoring criterion .The results are then analyzed statistically, typically using the averages and standard deviations as metrics. Subjective fidelity measures can be classified in to three categories. The first types are referred to as "*impairment tests"* where the test subjects score the images in terms of how bad they are. The second types are "*quality tests"* where the test subjects rate the images in term of how good they are. The third types are called "*comparison tests"* where the images are evaluated on side-by-side basis. In order to provide unbiased result, evaluation with subjective measure requires careful selection of the test subjects and carefully designed evaluation experiments.

5. Image Compression : Type And Technique

Depending on the relation between the original image (uncompressed image) and compressed image, there are two types of compression technique introduced. These types are *error free (loss less)* compression techniques and *error inherent (lossy)* compression techniques .

5.1.*Error Free (Loss Less) Compression Techniques*

In numerous applications error free compression is the only acceptable means of data reduction. The archival of medical or business document is such an application, where lossy compression usually is prohibited for legal reasons. The processing of LANDSAT imagery, is another application, where both the use and cost of collecting the data makes any loss undesirable. The resultant compression ratio of monochrome image in loss less compression is about 3:1 . Although this ratio is not high, the advantage of this type of compression is that the decompressed image is the same as the original. Several loss less compression techniques have been developed and implemented, these techniques are:

5.1.1. Huffman Coding

The most popular technique for removing coding redundancy. It was first described by D.A. Huffman in 1952. It is an entropy-coding scheme that uses variable length codes based on the statistical distribution of the input symbols.

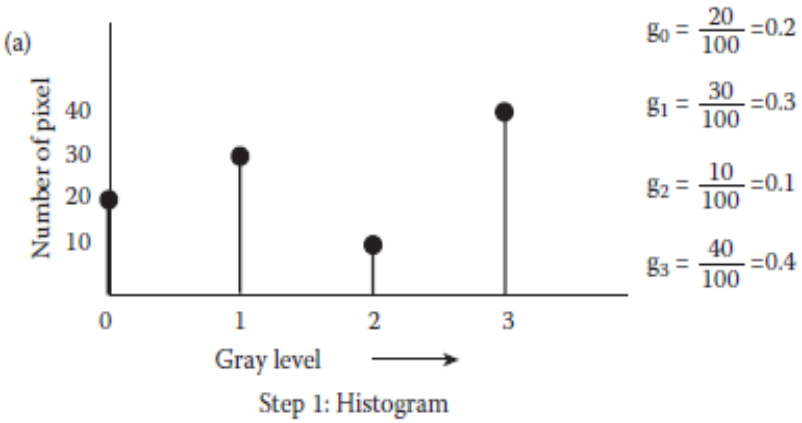
Huffman coding alone will typically reduce the file by [10 to 50% (1.1:1 to 1.5:1)], but this ratio can be improved to 2:1 or 3:1 by preprocessing for irrelevant information removal. In Huffman code the code itself is an instantaneous uniquely decodable block code, it is called a block code. The Huffman code can be constructed with the following procedure.

1- Arrange the probabilities of the gray levels according to their magnitudes.
2- Combine the two smallest probabilities, which results in anode with two branches.
3- If the summation of these two probabilities is equal to one, stop the combining operation.
4- If the summation is not equal to one, then rearrange the reconstructed probabilities in a descending order.
5- Repeat steps 2-4 until the summation of the two probabilities is equal one.
6- Assign (0) to the upper branch and (1) to the lower branch of each node or vice versa until the final code for each probability is obtained.

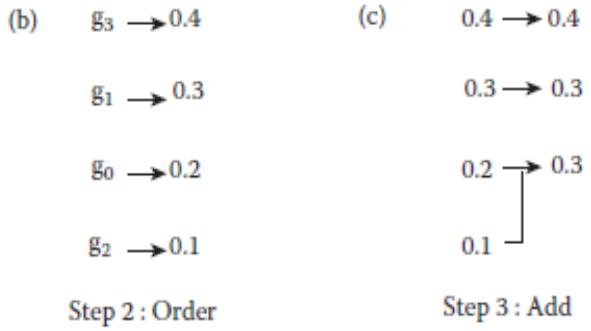
Fig.(2) gives an example to find the Huffman code for eight input probabilities. The first step is to order the input probabilities according to their magnitudes in a descending order. The two smallest probabilities are combined by addition to probabilities which have one fewer probability than the original set, is again ordered in any way (as an example, the 0.04 obtained by combining input probabilities 0.03 and 0.01 could be placed in any two of the bottom step 1 entries). When we get down to two probabilities we stop. As in step 6 of fig (2), code words are generated by starting at the last step and working backward. We assign 0 to the upper branch and 1 to the lower branch of each node. After assigning all the branches, the code for each input probability is obtained as shown in fig. (2).

Symb.	Prob.	step1	step2	step3	step4	step5	step6	code
a3	0.4	0.4	0.4	0.4	0.4	0.4	0.6	1
a1	0.25	0.25	0.25	0.25	0.25	0.35	0.4	01
a8	0.1	0.1	0.1	0.15	0.2	0.25		0000
a6	0.1	0.1	0.1	0.1	0.15			0001
a2	0.07	0.07	0.08	0.1				0011
a4	0.04	0.04	0.07					00100
a7	0.03	0.04						001010
a5	0.01							001011

Figure (2) Huffman coding

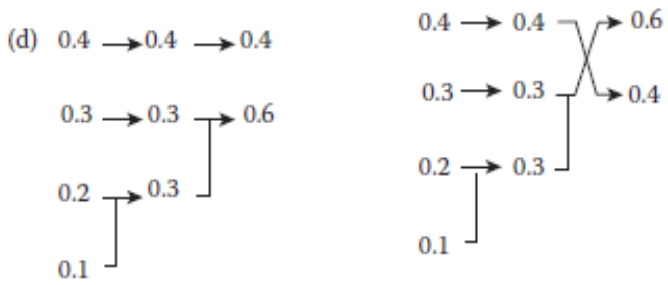


Step 1: Histogram

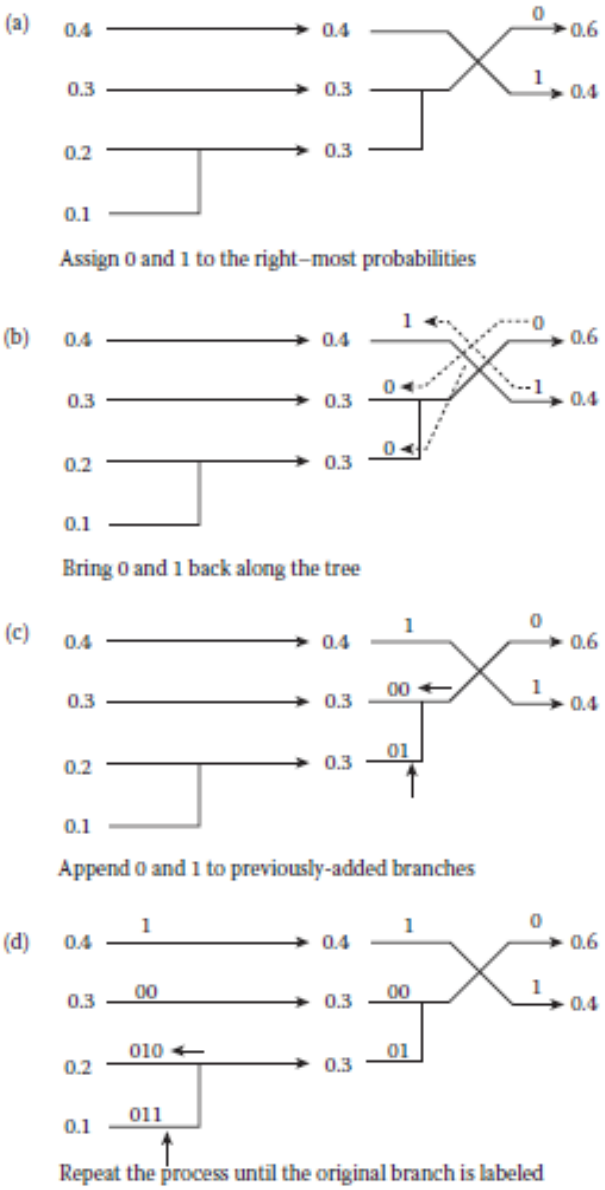


Step 2 : Order

Step 3 : Add



Step 4 : Reorder and add until only two values remain



$$Entropy = -\sum_{i=0}^{3} p_i \log_2(p_i)$$

$$= -[(0.2)\log_2(0.2) + (0.3)\log_2(0.3) + (0.1)\log_2(0.1) + (0.4)\log_2(0.4)]$$

$$\approx 1.846 \text{ bits/pixel}$$

(Note: $\log_2(x)$ can be found by taking $\log_{10}(x)$ and multiplying by 3.322)

$$L_{ave} = \sum_{i=0}^{L-1} l_i \, p_i$$

$$= 3(0.2) + 2(0.3) + 3(0.1) + 1(0.4)$$

$$= 1.9 \text{ bits/pixel} \quad \text{(Average length with Huffman code)}$$

Original Gray Level (Natural Code)	Probability	Huffman code
g_0: 00_2	0.2	010_2
g_1: 01_2	0.3	00_2
g_2: 10_2	0.1	011_2
g_3: 11_2	0.4	1_2

5.1.2. *Run-Length Coding*

Run Length coding (RLC) techniques based on mapping the sequence of image pixels value ($X1$, X_2, X3, -----X_N) along the scan line into a sequence of integer pairs consist of gray level plus run (g_k ,l_k) where g_k denotes the number of adjacent picture elements having the same gray levels . Hence the sequence of integer pairs will be coded instead of the gray level of each pixel. We can either use Horizontal RLC, counting along the rows, or vertical RLC, counting along the columns. Run length can be one-dimensional or two-dimensional. However, in the case of multilevel sources, it has been common practice to run-length encode only the redundancy runs .

The image is an 8×8 binary image, which requires 3 bits for each run-length coded word. In the actual image file are stored 1s and 0s, although upon display the 1s become 255 (white) and the 0s are 0 (black). To apply RLC to this image, using horizontal RLC:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The RLC numbers are
First row: 8
Second row: 0, 4, 4

Third row: 1, 2, 5
Fourth row: 1, 5, 2
Fifth row: 1, 3, 2, 1, 1
Sixth row: 2, 1, 2, 2, 1
Seventh row: 0, 4, 1, 1, 2
Eighth row: 8

Note that in the second and seventh rows, the first RLC number is 0, since we are using the convention that the first number corresponds to the number of zeros in a run.

Given the following 8×8, 4-bit image:

$$\begin{bmatrix} 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 & 12 & 12 & 12 \\ 10 & 10 & 10 & 10 & 10 & 12 & 12 & 12 \\ 0 & 0 & 0 & 10 & 10 & 10 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 & 0 & 0 & 0 \\ 5 & 5 & 5 & 10 & 10 & 9 & 9 & 10 \\ 5 & 5 & 5 & 4 & 4 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The corresponding gray levels pairs are as follows:
First row: 10,8
Second row: 10,5 12,3
Third row: 10,5 12,3
Fourth row: 0,3 10,3 0,2
Fifth row: 5,3 0,5
Sixth row: 5,3 10,2 9,2 10,1
Seventh row: 5,3 4,3 0,2
Eighth row: 0,8
These numbers are then stored in the RLC compressed file as
10,8,10,5,12,3,10,5,12,3,0,3,10,3,0,2,5,3,0,5,5,3,10,2,9,2,10,1,5,3,4,3,0,2,0,8

5.2. Error Inherent (Lossy) Compression Techniques

Lossy compression achieves high compression ratio with complex images. It is based on the concept of increasing the compression ratio while the accuracy of the reconstructed image is decreased, in fact many loosy compression techniques are capable of compressing image more than 20 times with virtually no visible information loss, and [30 to 50] time with minimal degradation.

Lossy techniques are not suitable for computer data, but are popular for audio and video application as they allow greater compression factors then loss-less techniques . Examples of lossy compression techniques are:

5.2.1. Predictive Coding

Predictive coding has been used extensively in image compression. Predictive image coding algorithms are used primarily to exploit the correlation between adjacent pixels. They predict the value of a given pixel based on the values of the surroundings pixels. Due to the correlation property among adjacent pixels in image, the use of a predictor can reduce the amount of information bits to represent image.

This type of lossy image compression technique is not as competitive as transform coding techniques used in modern lossy image compression, because predictive technique has inferior compression ratios and worse reconstructed image quality than those of transform coding .

Differential Pulse Code Modulation (DPCM) is one particular example of predictive coding.

A..*Differential Pulse Code Modulation (DPCM)*

Differential Pulse Code Modulation (DPCM) offers an attractive means of picture coding because of its relative simple realization in hardware. This is especially important for real time application and encoding of high resolution pictures for instance broad band television.

The DPCM compression method is a member of the family of differential encoding compression method. It is based on the well-known fact that neighboring pixels in an image are correlated , so their differences are small. In predictive coding the difference signal between the actual sample and its predicted value is quantized and transmitted. This technique has been used in speech coding, image coding , and in biomedical field.

i..General (DPCM) System

In a general DPCM system , a pixel's gray level is first predicted from the preceding reconstructed pixel's gray level values. The difference between the pixel's gray level value and the predicted value is then quantized. Finally, the quantized difference is encoded and transmitted to the receiver.

A basic DPCM system is shown in Fig.(3). Instead of transmitting the actual samples (S) directly to the receiver as in PCM , the quantized difference (*dq*) between the actual sample(S) and a predicted value ($\hat{S}$) is transmitted. The process consists of finding the difference (*d*) between the present sample value (S) and its predicted value ($\hat{S}$) .The difference signal is in general uncorrelated or almost uncorrelated. When the prediction is well chosen ,the entropy of the difference signal is much smaller than the entropy of the original signal and therefore the number of bits required to transmit or store the difference signal is less. The quantized difference (*dq*) (the quantizer output) is generally corrupted by noise (*eq*) called quantization noise or error. The quantizer output may be expressed as:

$dq = d + eq$ (2-9)

where e_q quantization error .

The predictor input (see fig (2-2)) is :

$$\dot{S} = d_q + \hat{S} = d + e_q + \hat{S}$$

Since:

$$d = S - \hat{S}$$

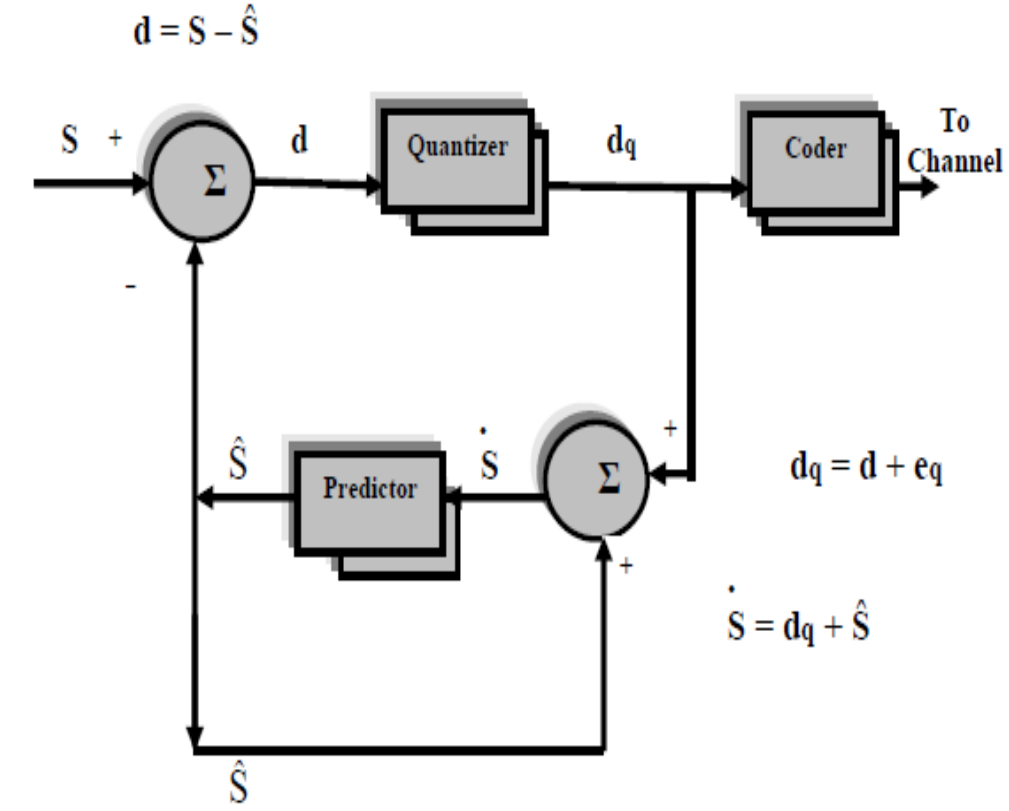
Then:

$$\dot{S} = S + e_q$$

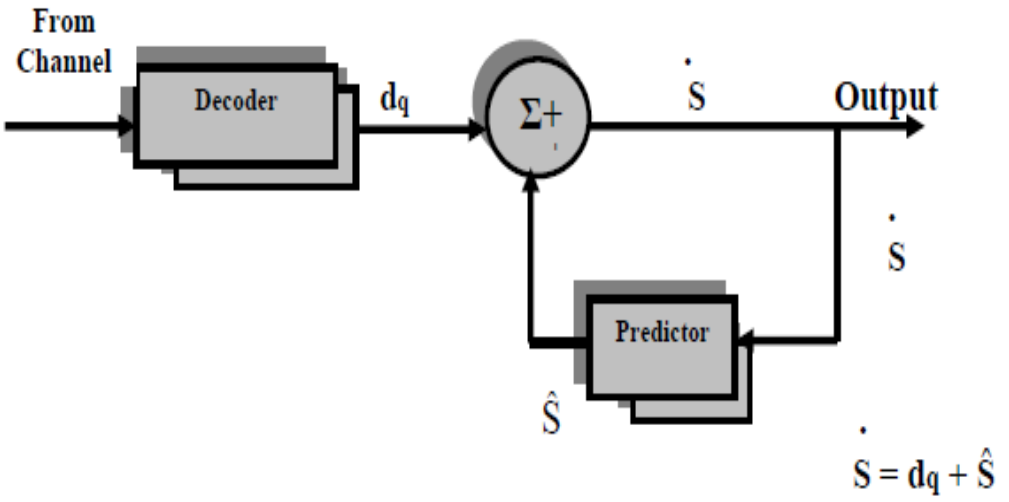
The reconstructed signal ($\dot{S}$) at the predictor input differs from the original input signal (S) by the quantization error (*eq*) . In the absence of quantization noise(*eq* = 0, *dq* = *d* = S – $\hat{S}$) and transmission error, the reconstructed sample ($\dot{S}$) are identical to the original input sample(S).

The predictors at the transmitter and receiver side calculate an estimated value in the same way from previous reconstructed samples . The corresponding receiver output is equal to ($\dot{S}$) which differs from the original input (S) only by the quantization error (*eq*) incurred as a result of quantizing the prediction difference (*d*) .

The best DPCM system design is that system which minimizes a measure of the overall error between the input and the output of the system. However this design procedure is prevented by non-linear characteristic of the quantizer. Therefore the optimization problem is solved by designing the linear predictor ignoring the presence of the quantizer and then the quantizer is designing to match the amplitude distribution of the difference signal.



(a) Transmitter



(b) Receiver

Figure (3). Differential Pulse Code Modulation DPCM)System

ii. <u>Predictor Design</u>

The predictor uses the statistical predictability between pixels to form an estimate of each pixel as a linear combination of previous pixels, where previous pixels is a term that has direct meaning in the context of top-to-bottom, left- to-right scanning, which imposes specific sequence on pixel occurrence .

The optimal predictor is used in most predictive coding applications minimizes the encoders mean-square prediction error . Let S_1, S_2…S_n in fig.(3) represent values of (n) previously scanned pixels that are to be used in the prediction of a value S_0. The pixels employed in the prediction may lie along the same line and/or along previously scanned line of the same image (frame) in this case the technique is called intraframe coding.

Neglecting quantization, a linear estimation of ($\hat{S}o$) is given by:

$$\hat{S}o = a_1S_1 + a_2S_2 + a_3S_3 + \ldots..+a_nS_n \qquad (2\text{-}10)$$

Where the a's are the prediction coefficient. A prediction difference is then defined as:

$$d_0 = S_0 - \hat{S}_0 \qquad (2\text{-}11)$$

The mean-square prediction error (MSE) is defined as :

$$E\{d_0^2\} = E\left\{\left(S_0 - \hat{S}_0\right)^2\right\}$$

The optimal choice for the prediction coefficient are determined by:

$$\frac{\partial E\{d0^2\}}{\partial a_k} = 0, \quad \text{for } k = 0, 1, 2, \ldots.. n$$

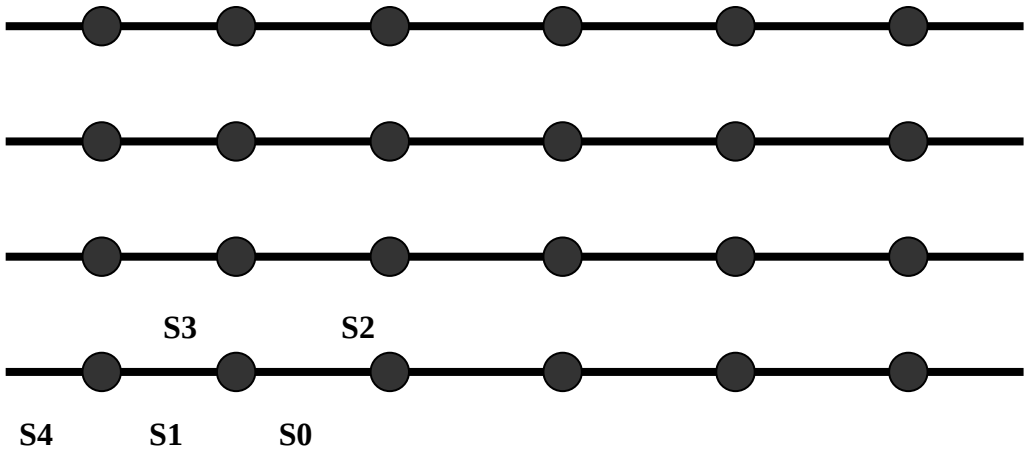


Fig.(4) Location of picture elements (pixels) for prediction of S0

In typical predictive image coding systems the amplitude distribution of the difference signal can be approximated by two-sided Laplacian density function as shown in fig.(4) . Normally the dynamic range of the difference signal is much smaller than the dynamic range of input signal. Since the difference signal amplitude is modeled by Laplacian Density function, a nonuniform quantizer matched to the difference signal statistics is employed .

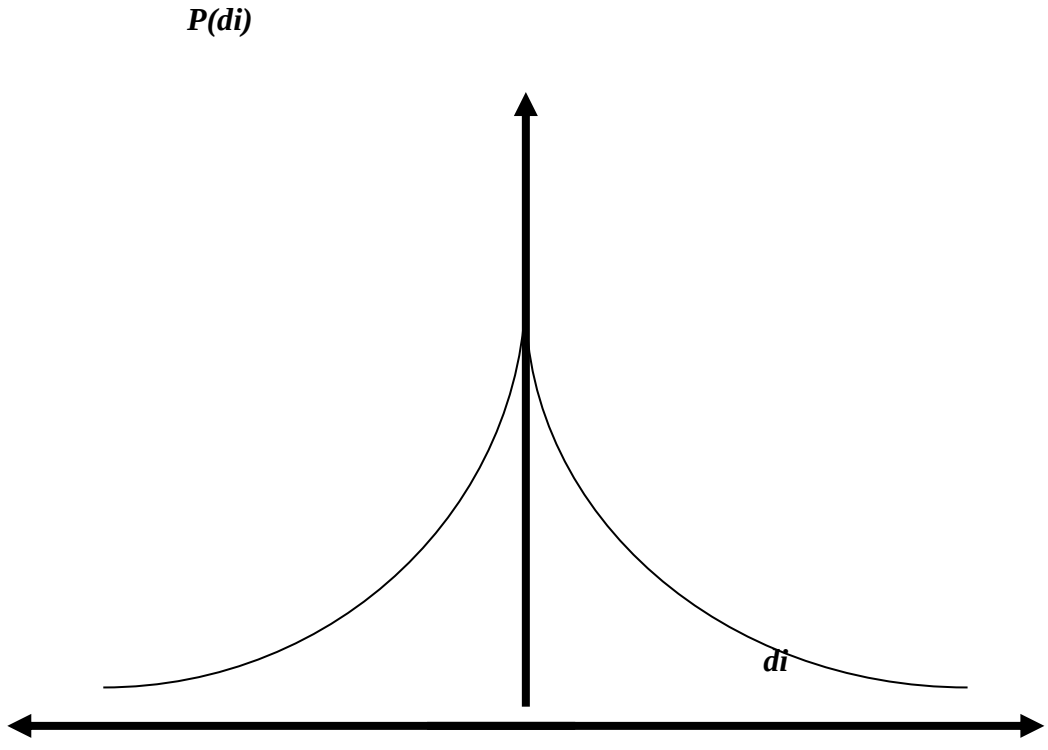


Fig.(5) Probability density function P(d) difference signal

The number of previous pixels employed in the estimate operation is referred to as the *order of the predictor* . Predictor using one pixel is called "*first order predictor*" . A "*second order predictor* " utilizes two pixels and an " *nth order predictor* " would employ *n* previous pixels.

Using more of the previous values in the prediction increases the complexity of the computation for both compression and decompression, and it has been determined that using more than three of the previous value provides no significant improvement in the resulting image.

6. Quantizer Design

Quantization is the process of mapping a continuous set of values into a finite set of values. In the DPCM system the input to the quantizer is the prediction difference signal (prediction error). The difference signal is an analog signal, which must be first converted into a finite number N of discrete values. The quantizer dose this process. The quantizer design problem is to select the best decision and reconstruction level for a particular optimization criterion and input probability density function. There are two types of quantization: Scalar Quantization and Vector Quantization.

Scalar quantization is an example of a lossy compression method ,where it is easy to control the trade-off between compression ratio and the amount of loss. However , because it is so simple, its use is limited to cases where much loss can be tolerated.

A quantizer can be specified by its input partitions and output levels (also called reproduction points) . If the input range is divided into levels of equal spacing, then the quantizer is termed as a *Uniform Quantizer,* and if not , it is termed as a *Non-Uniform Quantizer.* Uniform quantizers perform optimally for signal with uniform distribution. Non-uniform quantizer is specified for non-uniform distribution. It is more complex, but the added complexity is often worthwhile because it can be used to reduce the perceptual effects of the quantization. In DPCM , the value of difference signal(*d*) is compared to a set of decision levels. If the pixel value (*d*) falls between two adjacent decision levels (*di , di+1*) , it is quantized to a fixed reconstruction level(*dqi*) lying in the quantization band.

The quantization problem entails specification of a set of decision
levels di and a set of reconstruction levels *dqi* such that if :

$$di \leq d < di+1$$

Then the pixel value (*d*) is quantized to a reconstruction level (*dqi*) .

The decision and reconstructed levels for gaussion, laplacian and ray Leigh distributions for different N is listed in table (1) .

It is assumed that the distributions have zero mean $m=0$ and unit standard deviation $\delta=1$ which is called normalized levels**.**

size	gaussia		laplacian		Ray leigh	
bits	*di*	*dqi*	*di*	*dqi*	*di*	*dqi*
1-	∞	-0.7979	∞	- 0.7071	0.0000	1.2657
	0.0000	0.7979	0.0000	0.7071	2.0985	2.9313
	∞		∞		∞	
2-	∞	1.5104	∞	- 1.0834	0.0000	0.8079
	0.9816	0.4528	-1.1269	-0.4198	1.2545	1.7010
	0.0000	0.4528	0.0000	0.4198	2.1667	2.6325
	0.8816	1.5104	1.1269	1.0834	3.2465	3.8604
	∞		∞		∞	
3-	∞	2.1519	∞	- 3.0867	0.0000	0.5016
	1.7479	1.3439	-2.3796	- 1.6725	0.7619	1.0222
	1.0500	0.756	-1.2527	- 0.8330	1.2594	1.4966
	0.5005	0.2451	0.5332	-0.2334	1.7327	1.9688
	0.0000	0.2451	0.0000	0.2334	2.2182	2.4675
	0.5005	0.756	0.5332	0.8330	2.7476	3.0277
	1.0500	1.3439	1.2527	1.6725	3.3707	3.7137
	1.7479	2.1519	2.3796	3.0867	4.2124	4.7111
	∞		∞		∞	
4-	- ∞	2.7326	∞	- 4.4311	0.0000	0.3057
	- 2.4008	2.0690	-3.7240	-3.0169	0.4606	0.6156
	- 8435	1.6180	-2.5971	-2.1773	0.7509	0.8863
	-1.4371	-1.2562	-1.8776	-1.5778	1.0310	1.1397
	-1.0993	-0.9423	-1.3444	-1.1110	1.2624	1.3850
	0.7995	-0.6568	-0.9198	-0.7287	1.5064	1.6277
	0.5224	-0.3885	-0.5667	-0.4048	1.7499	1.8721
	0.2588	-0.1284	-0.2664	-0.1240	1.9970	2.1220
	0.0000	0.1284	0.0000	0.1240	2.2517	2.3814
	0.2528	0.3885	0.2644	0.4048	2.5182	2.6550
	0.5224	0.6568	0.5667	0.7287	2.8021	2.9492
	0.7995	0.9423	0.9198	1.1110	3.1110	3.2729
	1.0993	1.2562	1.3444	1.5778	3.4566	3.6403
	1.4371	1.6180	1.8776	2.1773	3.8588	4.0772
	1.8435	2.0690	2.5971	3.0169	4.3579	4.6385
	2.4008	2.7326	3.7240	4.4311	5.0649	5.4913
	∞		∞		∞	

Table (1) Placement of decision & reconstructed level for max quantizer