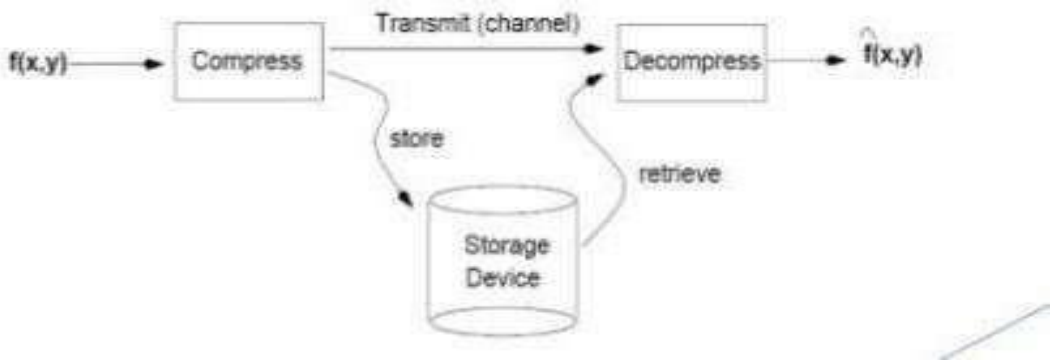
Definition: Image compression deals with reducing the amount of data required to represent a digital image by removing of redundant data.

Images can be represented in digital format in many ways. Encoding the contents of a 2-D image in a raw bitmap (raster) format is usually not economical and may result in very large files. Since raw image representations usually require a large amount of storage space (and proportionally long transmission times in the case of file uploads/ downloads), most image file formats employ some type of compression. The need to save storage space and shorten transmission time, as well as the human visual system tolerance to a modest amount of loss, have been the driving factors behind image compression techniques.

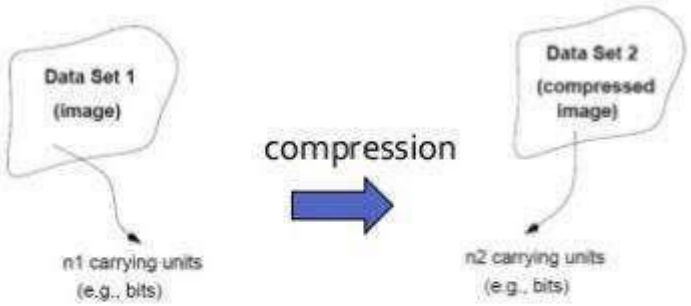
Goal of image compression: The goal of image compression is to reduce the amount of data required to represent a digital image.



Data ≠Information:

- Data and information are not synonymous terms!
- Data is the means by which information is conveyed.
- Datacompressionaimstoreducetheamountofdatarequiredtorepresentagiven quality of information while preserving as much information as possible.
- The same amount of information can be represented by various amount of data. Ex1: You have an extra class after completion of 3.50p.m
  Ex2: Extra class have been scheduled after 7^{th} hour for you.
  Ex3: After 3.50 p.m you should attended extra class.

Definition of compression ratio:



Compression ratic $C_R = \frac{n_1}{n_2}$

Definitions of Data Redundancy:

- Relative data redundancy: $R_D = 1 - \frac{1}{C_R}$

Example:

If $C_R = \frac{10}{1}$, then $R_D = 1 - \frac{1}{10} = 0.9$

(90% of the data in dataset 1 is redundant)

if $n_2 = n_1$, then C_R=1, R_D=0

if $n_2 \ll n_1$, then $C_R \to \infty$, $R_D \to 1$

Coding redundancy:

- Code: a list of symbols (letters, numbers, bitsetc.,)
- Code word: a sequence of symbol used to represent a piece of information or anevent (e.g., graylevels).
- Code word length: number of symbols in each codeword.

Example: (binary code, symbols: 0,1, length: 3)

0: 000	4: 100
1: 001	5: 101
2: 010	6: 110
3: 011	7: 111

N x M image
r_k: k-th gray level
$P(r_k)$: probability of r_k
$l(r_k)$: # of bits for r_k

Expected value:

$$E(X) = \sum_x xP(X = x)$$

Average # of bits: $L_{avg} = E(l(r_k)) = \sum_{k=0}^{L-1} l(r_k)P(r_k)$

Total # of bits: NML_{avg}

Coding Redundancy (con'd)

Case 1: $l(r_k)$ = constant length

Example:

r_k	$p(r_k)$	Code 1	$l(r_k)$
$r_0 = 0$	0.19	000	3
$r_1 = 1/7$	0.25	001	3
$r_2 = 2/7$	0.21	010	3
$r_3 = 3/7$	0.16	011	3
$r_4 = 4/7$	0.08	100	3
$r_5 = 5/7$	0.06	101	3
$r_6 = 6/7$	0.03	110	3
$r_7 = 1$	0.02	111	3

Assume an image with $L = 8$

Assume $l(r_k) = 3$, $L_{avg} = \sum_{k=0}^{7} 3P(r_k) = 3\sum_{k=0}^{7} P(r_k) = 3$ bits

Total number of bits: $3NM$

Case 2: $l(r_k)$ = variable length

Table 6.1 Variable-Length Coding Example

r_k	$p(r_k)$	Code 1	$l(r_k)$	Code 2	$l(r_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	00001	5
$r_6 = 6/7$	0.03	110	3	000001	6
$r_7 = 1$	0.02	111	3	000000	6

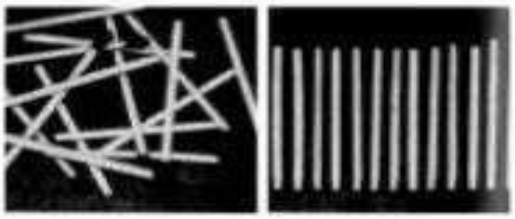
$$L_{avg} = \sum_{k=0}^{7} l(r_k)P(r_k) = 2.7 \text{ bits}$$

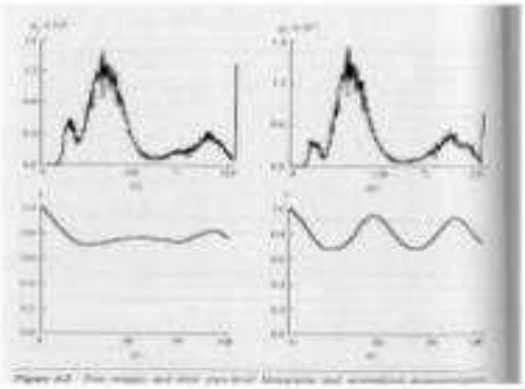
$$C_R = \frac{3}{2.7} = 1.11 \text{ (about 10\%)}$$

$$R_D = 1 - \frac{1}{1.11} = 0.099$$

Interpixel redundancy

- Interpixel redundancy implies that pixel values are correlated (i.e., a pixel value can be reasonably predicted by its neighbors).





$$f(x) \circ g(x) = \int_{-\infty}^{\infty} f(x) g(x+a) da$$

autocorrelation: f(x)=g(x)

Interpixel redundancy (cont'd)

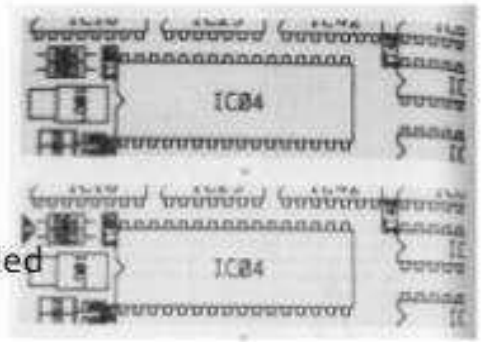
To reduce interpixel redundancy, the data must be transformed in another format (i.e., using a transformation)

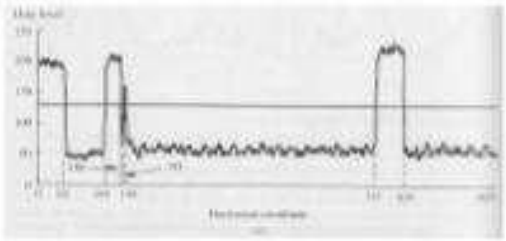
- e.g., thresholding, DFT, DWT, etc.

Example:

origin
al

thresholded





Run-length encoding:

(1,63) (0,87) (1,37) (0,5) (1,4) (0, 556) (1,62) (0,210)

Using 11 bits/pair,

88 bits are required (compared to 1024 ?!)

Psychovisual redundancy

The human eye does not respond with equal sensitivity to all visual information.

It is more sensitive to the lower frequencies than to the higher frequencies in the visual spectrum.

Idea: discard data that is perceptually insignificant!

Fidelity Criteria

$$\hat{f}(x,y) = f(x,y) + e(x,y)$$

- How close $f(x,y)$ $\hat{f}(x,y)$?

- Criteria
  - Subjective: based on human observers
  - Objective: mathematically defined criteria

Subjective Fidelity Criteria

Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

Objective Fidelity Criteria

- Root mean square error (RMS)

$$e_{rms} = \sqrt{\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(\hat{f}(x,y) - f(x,y))^2}$$

- Mean-square

$$SNR_{ms} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(\hat{f}(x,y))^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(\hat{f}(x,y) - f(x,y))^2}$$

COMPRESSION METHODS OF IMAGES:

Compression methods can be ***lossy***,

When a tolerable degree of deterioration in the visual quality of the resulting image is acceptable, or ***lossless***,

When the image is encoded in its full quality. The overall results of the compression process, both in terms of storage savings – usually expressed numerically in terms of compression ratio (CR) or bits per pixel (bpp) – as well as resulting quality loss (for the case of lossy techniques) may vary depending on the technique, format, options (such as the quality setting for JPEG), and the image contents.

As a general guideline, ***lossy compression*** should be used for general purpose photographic images.

Whereas ***lossless compression*** should be preferred when dealing with line art, technical drawings, cartoons, etc. or images in which no loss of detail may be tolerable (most notably, space images and medical images).

Fundamentals of visual data compression

The general problem of image compression is to reduce the amount of data required to represent a digital image or video and the underlying basis of the reduction process is the removal of redundant data. Mathematically, visual data compression typically involves

transforming (encoding) a 2-D pixel array into a statistically uncorrelated data set. This transformation is applied prior to storage or transmission. At some later time, the compressed image is decompressed to reconstruct the original image information (preserving or lossless techniques) or an approximation of it (lossy techniques).

Redundancy

Data compression is the process of reducing the amount of data required to represent a given quantity of information. Different amounts of data might be used to communicate the same amount of information. If the same information can be represented using different amounts of data, it is reasonable to believe that the representation that requires more data contains what is technically called ***data redundancy.***

Image compression and coding techniques explore three types of redundancies: ***coding*** redundancy, ***interpixel***(spatial) redundancy, and *psychovisual* redundancy. The way each of them is explored is briefly described below.

- **Coding redundancy:** consists in using variable-length code words selected as to match the statistics of the original source, in this case, the image itself or a processed version of its pixel values. This type of coding is always reversible and usually implemented using look-up tables (LUTs). Examples of image coding schemes that explore coding redundancy are the Huffman codes and the arithmetic coding technique.
- **Interpixel redundancy:** this type of redundancy – sometimes called spatial redundancy, inter frame redundancy, or geometric redundancy – exploits the fact that an image very often contains strongly correlated pixels, in other words, large regions whose pixel values are the same or almost the same. This redundancy can be explored in several ways, one of which is by predicting a pixel value based on the values of its neighboring pixels. In order to do so, the original 2-D array of pixels is usually mapped into a different format, e.g., an array of differences between adjacent pixels. If the original image pixels can be reconstructed from the transformed data set the mapping is said to be reversible. Examples of compression techniques that explore the interpixel redundancy include: Constant Area Coding (CAC), (1-D or 2-D) Run-Length Encoding (RLE) techniques, and many predictive coding algorithms such as Differential Pulse Code Modulation (DPCM).
- **Psycho visual redundancy: many experiments on the psychophysical aspects of human** vision have proven that the human eye does not respond with equal sensitivity to all incoming visual information; some pieces of information are more important than others. The

knowledge of which particular types of information are more or less relevant to the final human user have led to image and video compression techniques that aim at eliminating or reducing any amount of data that is psycho visually redundant. The end result of applying these techniques is a compressed image file, whose size and quality are smaller than the original information, but whose resulting quality is still acceptable for the application at hand. The loss of quality that ensues as a byproduct of such techniques is frequently called *quantization,* as to indicate that a wider range of input values is normally mapped into a narrower range of output values thorough an irreversible process. In order to establish the nature and extent of information loss, different fidelity criteria (some objective such as root mean square (RMS) error, some subjective, such as pair wise comparison of two images encoded with different quality settings) can be used. Most of the image coding algorithms in use today exploit this type of redundancy, such as the Discrete Cosine Transform (DCT) based algorithm at the heart of the JPEG encoding standard.

IMAGE COMPRESSION AND CODING MODELS

Figure shows a general image compression model. It consists of a source encoder, a channel encoder, the storage or transmission media (also referred to as *channel)*, a channel decoder, and a source decoder. The source encoder reduces or eliminates any redundancies in the input image, which usually leads to bit savings. Source encoding techniques are the primary focus of this discussion. The channel encoder increase noise immunity of source encoder's output, usually adding extra bits to achieve its goals. If the channel is noise-free, the channel encoder and decoder may be omitted. At the receiver's side, the channel and source decoder perform the opposite functions and ultimately recover (an approximation of) the original image.

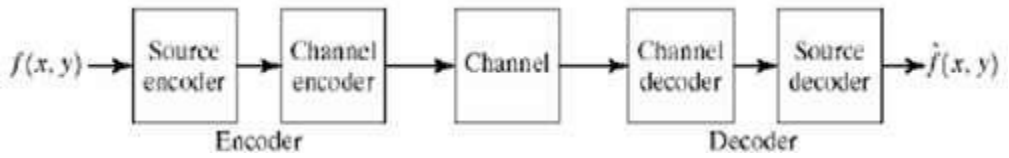


Figure Image compression model

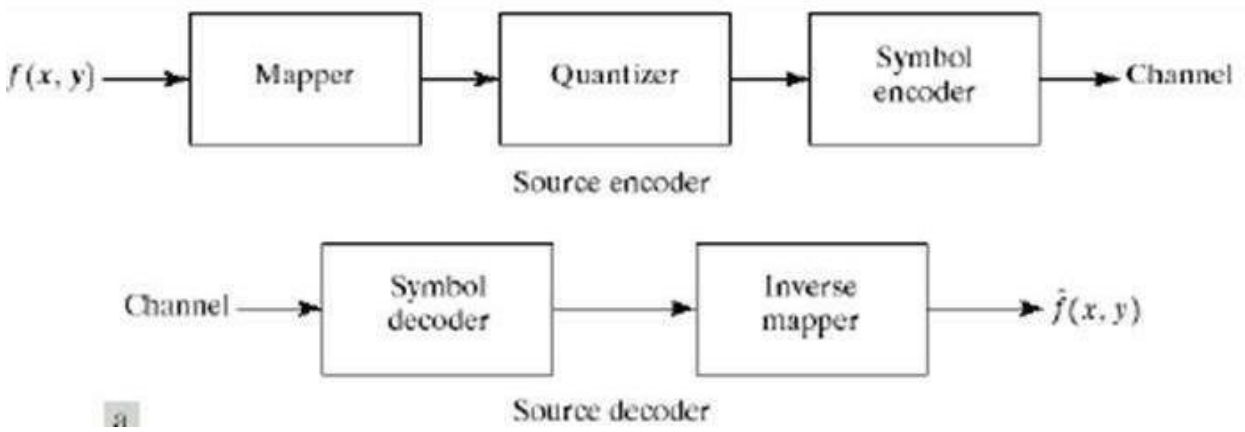


Figure shows the source encoder in further detail. Its main components are:

- **Mapper:** transforms the input data into a (usually nonvisual) format designed to reduce interpixel redundancies in the input image. This operation is generally reversible and may or may not directly reduce the amount of data required to represent the image.

- **Quantizer:** reduces the accuracy of the mapper's output in accordance with some pre-established fidelity criterion. Reduces the psychovisual redundancies of the input image. This operation is not reversible and must be omitted if lossless compression is desired.

- **Symbol (entropy) encoder:** creates a fixed- or variable-length code to represent the quantizer's output and maps the output in accordance with the code. In most cases, a variable-length code is used. This operation is reversible.

Error-free compression

Error-free compression techniques usually rely on entropy-based encoding algorithms. The concept of entropy is mathematically described in equation (1):

Where:

a_j is a symbol produced by the information source

$P(a_j)$ is the probability of that symbol

J is the total number of different symbols

$H(z)$ is the entropy of the source.

The concept of entropy provides an upper bound on how much compression can be achieved, given the probability distribution of the source. In other words, it establishes a theoretical limit on the amount of lossless compression that can be achieved using entropy encoding techniques alone.

Variable Length Coding (VLC)

Most entropy-based encoding techniques rely on assigning variable-length codewords to each symbol, whereas the most likely symbols are assigned shorter codewords. In the case of image coding, the symbols may be raw pixel values or the numerical values obtained at the output of the mapper stage (e.g., differences between consecutive pixels, run-lengths, etc.). The most popular entropy-based encoding technique is the Huffman code. It provides the least amount of information units (bits) per source symbol. It is described in more detail in a separate short article.

Run-length encoding (RLE)

RLE is one of the simplest data compression techniques. It consists of replacing a sequence (run) of identical symbols by a pair containing the symbol and the run length. It is used as the primary compression technique in the 1-D CCITT Group 3 fax standard and in conjunction with other techniques in the JPEG image compression standard (described in a separate short article).

Differential coding

Differential coding techniques explore the interpixel redundancy in digital images. The basic idea consists of applying a simple difference operator to neighboring pixels to calculate a difference image, whose values are likely to follow within a much narrower range than the original gray-level range. As a consequence of this narrower distribution – and consequently reduced entropy – Huffman coding or other VLC schemes will produce shorter code words for the difference image.

Predictive coding

Predictive coding techniques constitute another example of exploration of interpixel

redundancy, in which the basic idea is to encode only the new information in each pixel. This new information is usually defined as the difference between the actual and the predicted value of that pixel.

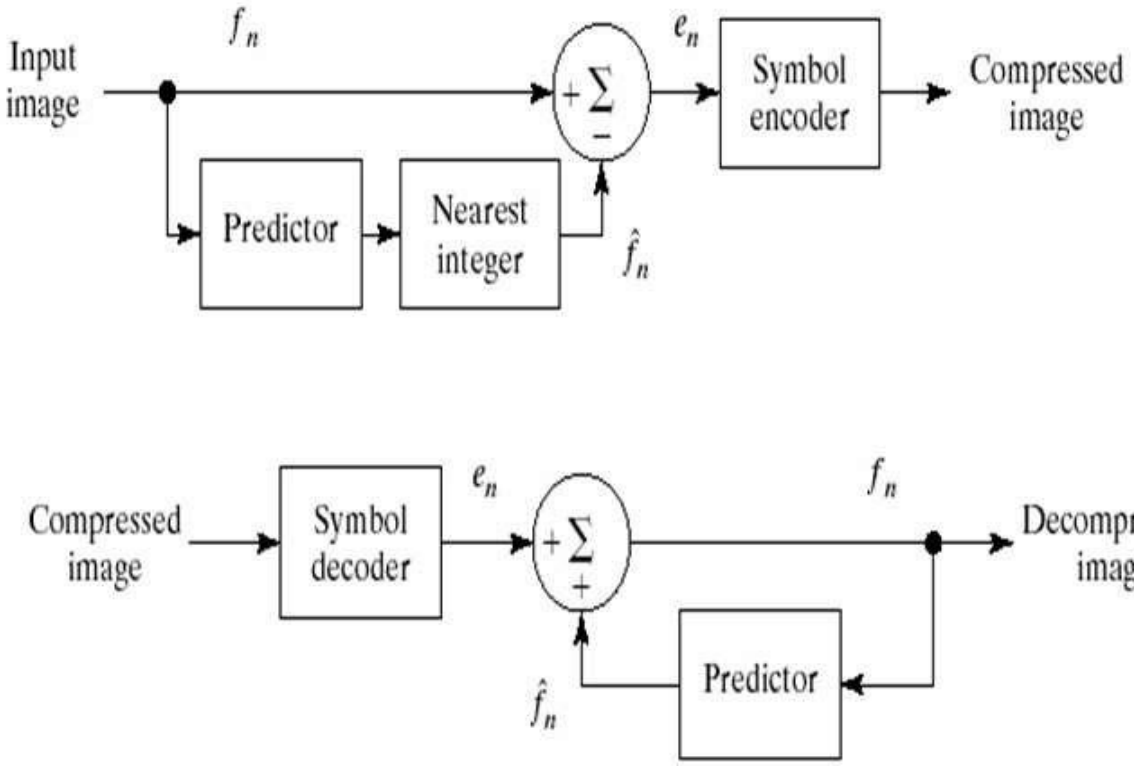


Figure 3 shows the main blocks of a lossless predictive encoder. The key component is the predictor, whose function is to generate an estimated (predicted) value for each pixel from the input image based on previous pixel values. The predictor's output is rounded to the nearest integer and compared with the actual pixel value: the difference between the two – called *prediction error* – is then encoded by a VLC encoder. Since prediction errors are likely to be smaller than the original pixel values, the VLC encoder will likely generate shorter code words.

There are several local, global, and adaptive prediction algorithms in the literature. In most cases, the predicted pixel value is a linear combination of previous pixels.

Dictionary-based coding

Dictionary-based coding techniques are based on the idea of incrementally building a dictionary (table) while receiving the data. Unlike VLC techniques, dictionary-based techniques use fixed-length code words to represent variable-length strings of symbols that commonly occur together. Consequently, there is no need to calculate, store, or transmit the

probability distribution of the source, which makes these algorithms extremely convenient and popular. The best-known variant of dictionary-based coding algorithms is the LZW (Lempel-Ziv-Welch) encoding scheme, used in popular multimedia file formats such as GIF, TIFF, andPDF.

Lossy compression

Lossy compression techniques deliberately introduce a certain amount of distortion to the encoded image, exploring the psychovisual redundancies of the original image. These techniques must find an appropriate balance between the amount of error (loss) and the resulting bit savings.

Quantization

The quantization stage is at the core of any lossy image encoding algorithm. Quantization, in at the encoder side, means partitioning of the input data range into a smaller set of values. There are two main types of quantizers: scalar quantizers and vector quantizers. A scalar quantizer partitions the domain of input values into a smaller number of intervals. If the output intervals are equally spaced, which is the simplest way todo it, the process is called *uniform scalar quantization;* otherwise, for reasons usually related to minimization of total distortion, it is called *non uniform scalar quantization.* One of the most popular non uniform quantizers is the Lloyd-Max quantizer. Vector quantization (VQ) techniques extend the basic principles of scalar quantization to multiple dimensions. Because of its fast lookup capabilities at the decoder side, VQ-based coding schemes are particularly attractive to multimedia applications.

Transform coding

The techniques discussed so far work directly on the pixel values and are usually called *spatial domain techniques.* Transform coding techniques use a reversible, linear mathematical transform to map the pixel values onto a set of coefficients, which are thenquantized and encoded. The key factor behind the success of transform-based coding schemes many of the resulting coefficients for most natural images have small magnitudes and can be quantized (or discarded altogether) without causing significant distortion in the decoded image. Different mathematical transforms, such as Fourier (DFT), Walsh-Hadamard (WHT), and Karhunen- Loeve (KLT), have been considered for the task. For compression purposes, the higher the capability of compressing information in fewer coefficients, the better the transform; for that reason, the Discrete Cosine Transform (DCT) has become the most widely used transform

coding technique.

Wavelet coding

Wavelet coding techniques are also based on the idea that the coefficients of a transform that decorrelates the pixels of an image can be coded more efficiently than the original pixels themselves. The main difference between wavelet coding and DCT-based coding (Figure 4) is the omission of the first stage. Because wavelet transforms are capable of representing an input signal with multiple levels of resolution, and yet maintain the useful compaction properties of the DCT, the subdivision of the input image into smaller sub images is no longer necessary. Wavelet coding has been at the core of the latest image compression standards, most notably JPEG 2000, which is discussed in a separate short article.

Image compression standards

Work on international standards for image compression started in the late 1970s with the CCITT (currently ITU-T) need to standardize binary image compression algorithms for Group 3 facsimile communications. Since then, many other committees and standards have been formed to produce *de jure* standards (such as JPEG), while several commercially successful initiatives have effectively become *de facto* standards (such as GIF). Image compression standards bring about many benefits, such as: (1) easier exchange of image files between different devices and applications; (2) reuse of existing hardware and software for a wider array of products; (3) existence of benchmarks and reference data sets for new and alternative developments.

Binary image compression standards

Work on binary image compression standards was initially motivated by CCITT Group 3 and 4 facsimile standards. The Group 3 standard uses a non-adaptive, 1-D RLE technique in which the last K-1 lines of each group of K lines (for K = 2 or 4) are optionally coded in a 2- D manner, using the *Modified Relative Element Address Designate* (MREAD) algorithm. The Group 4 standard uses only the MREAD coding algorithm. Both classes of algorithms are non- adaptive and were optimized for a set of eight test images, containing a mix of representative documents, which sometimes resulted in data expansion when applied to different types of documents (e.g., half-tone images).. The Joint Bilevel Image Group (JBIG)– a joint committee of the ITU-T and ISO – has addressed these limitations and proposed two new standards (JBIG and JBIG2) which can be used to compress binary and gray-scale images of up to 6 gray-coded

bits/pixel.

Continuous tone still image compression standards

For photograph quality images (both grayscale and color), different standards have been proposed, mostly based on lossy compression techniques. The most popular standard in this category, by far, is the JPEG standard, a lossy, DCT-based coding algorithm. Despite its great popularity and adoption, ranging from digital cameras to the World Wide Web, certain limitations of the original JPEG algorithm have motivated the recent development of two alternative standards, JPEG 2000 and JPEG-LS (lossless). JPEG, JPEG 2000, and JPEG-LS are described in separate short articles.

Encode each pixel ignoring their inter-pixel dependencies. Among methods are:

1. ***Entropy Coding:*** Every block of an image is entropy encoded based upon the Pk's within a block. This produces variable length code for each block depending on spatial activities within the blocks.
2. ***Run-Length Encoding:*** Scan the image horizontally or vertically and while scanning assign a group of pixel with the same intensity into a pair (gi ,li) where gi is the intensity and li is the length of the "run". This method can also be used for detecting edges and boundaries of an object. It is mostly used for images with a small number of gray levels and is not effective for highly textured images.

Example 1: Consider the following 8×8 image.

4	4	4	4	4	4	4	0
4	5	5	5	5	5	4	0
4	5	6	6	6	5	4	0
4	5	6	7	6	5	4	0
4	5	6	6	6	5	4	0
4	5	5	5	5	5	4	0
4	4	4	4	4	4	4	0
4	4	4	4	4	4	4	0

The run-length codes using vertical (continuous top-down) scanning mode are:

(4,9)	(5,5)	(4,3)	(5,1)	(6,3)
(5,1)	(4,3)	(5,1)	(6,1)	(7,1)
(6,1)	(5,1)	(4,3)	(5,1)	(6,3)
(5,1)	(4,3)	(5,5)	(4,10)	(0,8)

i.e. total of 20 pairs $= 40$ numbers. The horizontal scanning would lead to 34 pairs $= 68$ numbers, which is more than the actual number of pixels (i.e. 64).

Example 2: Let the transition probabilities for run-length encoding of a binary image (0:black

and 1:white) be p0 = P(0/1) and p1 = P(1/0). Assuming all runs are independent, find
(a) average run lengths, (b) entropies ofwhite and black runs, and (c) compression ratio. Solution:

A run of length $l \geq 1$ can be represented by a Geometric random variable (Grv) X_i with PMF $P(X_i = l) = p_i (1-p_i)^{t-1}$ with i = 0,1 which corresponds to happening of 1st occurrences of 0 or 1 after *l* independent trials. (Note that (1-P(0/1)) = P(1/1) and(1-P(1/0)) = P(0/0)) and Thus, for the average we have

$$\mu_{X_i} = \sum_{l=1}^{\infty} lP(X_i = l) = \sum_{l=1}^{\infty} lp_i(1-p_i)^{l-1}$$

which using series $\sum_{n=1}^{\infty} na^{n-1} = \frac{1}{(1-a)^2}$ reduces to $\mu_{X_i} = \frac{1}{p_i}$. The entropy is given by

$$\begin{aligned} H_{X_i} &= -\sum_{l=1}^{\infty} P(X_i = l) log_2 P(X_i = l) \\ &= -p_i \sum_{l=1}^{\infty} (1-p_i)^{l-1} [log_2 p_i + (l-1) log_2 (1-p_i)] \end{aligned}$$

Using the same series formula, we get

$$H_{X_i} = -\frac{1}{p_i}[p_i log_2 p_i + (1-p_i) log_2 (1-p_i)]$$

. The achievable compression ratio is

$$C = \frac{H_{X_0} + H_{X_1}}{\mu_{X_0} + \mu_{X_1}} = \frac{H_{X_0} P_0}{\mu_{X_0}} + \frac{H_{X_1} P_1}{\mu_{X_1}}$$

where $P_i = \frac{p_i}{p_0 + p_1}$ are the *a priori* probabilities of black and white pixels.

Huffman Encoding Algorithm*:* It consists of the following steps.

1. Arrange symbols with probability Pk's in a decreasing order and consider them as "leaf nodes" of a tree.
2. Merge two nodes with smallest probability to form a new node whose probability is the sum of the two merged nodes. Go to Step 1 and repeat until only two nodes are left ("root nodes"). 3 Arbitrarily assign 1's and 0's to each pair of branches merging into a node. 4 Read sequentially from root node to the leaf nodes to form the associated code for each symbol.

Example 3: For the same image in the previous example, which requires 3 bits/pixel using standard PCM we can arrange the table .

Gray levels	# occurrences	P_k	C_k	β_k	$P_k\beta_k$	$-P_k \log_2 P_k$
0	8	0.125	0000	4	0.5	0.375
1	0	0	-	0	-	-
2	0	0	-	0	-	-
3	0	0	-	0	-	-
4	31	0.484	1	1	0.484	0.507
5	16	0.25	01	2	0.5	0.5
6	8	0.125	001	3	0.375	0.375
7	1	0.016	0001	4	0.64	0.095
	64	1			R	H

Codewords C_ks are obtained by constructing the binary tree as in Fig. 5.

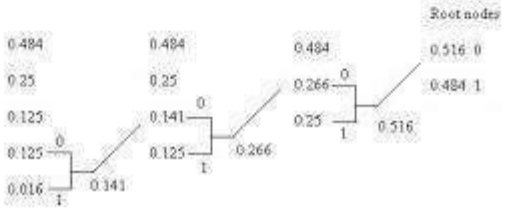


Fig. Tree structure for Huffman Encoding

Note that in this case, we have

$$R = \sum_{k=1}^{8} \beta_k P_k = 1.923 \text{ bits/pixel}$$

$$H = -\sum_{k=1}^{8} P_k \log_2 P_k = 1.852 \text{ bits/pixel}$$

Thus,

$$1.852 \leq R = 1.923 \leq H + \frac{1}{L} = 1.977$$

i.e., an average of 2 bits/pixel (instead of 3 bits/pixel using PCM) can be used to code the image. However, the drawback of the standard Huffman encoding method is that the codes have variable lengths.

PREDICTIVE ENCODING:

Idea: Remove mutual redundancy among successive pixels in a region of support (ROS) or neighborhood and encode only the new information. This mehtod is based upon linear prediction. Let us start with 1-D linear predictors. An N^{th} order linear prediction of x(n) based on N previous samples is generated using a 1-D autoregressive (AR)model.

$$\hat{x}(n) = a_1 x(n-1) + a_2 x(n-2) + \cdots + a_N x(n-N)$$

a_is are model coefficients determined based on some sample signals. Now instead of encoding x(n) the prediction error.

$$e(n) = x(n) - \hat{x}(n)$$

Is encoded as it requires substantially small number of bits. Then, at the receiver we reconstruct x(n) using the previous encoded values x(n-k) and the encoded error signal, i.e.,

$$x(n) = \hat{x}(n) + e(n)$$

This method is also referred to as differential PCM (DPCM).

Minimum Variance Prediction

The predictor

$$\hat{x}(n) = \sum_{i=1}^{N} a_i x(n-i)$$

is the best N^{th} order linear mean-squared predictor of $x(n)$, which minimizes the MSE

$$\epsilon = E\left[\left(x(n) - \hat{x}(n)\right)^2\right]$$

This minimization wrt a_k's results in the following "orthogonal property"

$$\frac{\partial \epsilon}{\partial a_k} = -2E\left[\left(x(n) - \hat{x}(n)\right) x(n-k)\right] = 0, \qquad 1 \le k \le N$$

which leads to the normal equation

$$r_{xx}(k) - \sum_{i=1}^{N} a_i r_{xx}(k-i) = \sigma_e^2 \delta(k), \qquad 0 \le k \le N$$

where $r_{xx}(k)$ is the autocorrelation of the data $x(n)$ and σ_e^2 is the variance of the driving process $e(n)$.

To understand the need for compact image representation, consider the amount of data required to represent a **2** hour standard Definition (SD) using 720 x 480 x 24 bit pixel arrays.

A video is a sequence of video frames where each frame is full color still image. Because video player must display the frames sequentially at rates near 30 fps. Standard definition data must be accessed 30fps x (720 x 480) ppf x 3bpp = 31,104,000 bps.

fps: frames per second, ***ppf***: pixels per frame, ***bpp***: bytes per pixel, ***bps***: bytes per second.

Thus a 2 hour movie consists of : = 31,104,000 bps x (60^2) sph x 2hrs where sph is second per hour = 2.24×10^{11} bytes = 224 GB of data. TWENTY SEVEN 8.5 GB dual layer DVD's are needed to store it.

To put 2 hours movie on a single DVD, each frame must be compressed by a factor of around 26.3.

The compression must be even higher for HD, where image resolution reaches 1920 x 1080 x 24 bits per image.

Webpage images & High-resolution digital camera photos also are compressed to save storage space & reduce transmission time.

Residential Internet connection delivers data at speeds ranging from 56kbps (conventional phone line) to more than 12 mbps (broadband).

Time required to transmit a small 128 x 128 x 24 bit full color image over this range of speed is from 7.0 to 0.03 sec.

Compression can reduce the transmission time by a factor of around 2 to 10 or more.
Similarly, number of uncompressed full color images that an 8 Megapixel digital camera can store on a 1GB Memory card can be increased.

Data compression: It refers to the process of reducing the amount of data required to represent a given quantity of information.

Data Vs Information:

Data and information is not the same thing; data are the means by which information is conveyed.

Because various amounts of data can be used to represent the same amount of information, representations that contain irrelevant or repeated information are said to contain redundant.
In today's multimedia wireless communication, major issue is bandwidth needed to satisfy real time transmission of image data. Compression is one of the good solutions to address this issue.
Transform based compression algorithms are widely used in the field of compression,

because of their de-correlation and other properties, useful in compression. In this paper, comparative study of compression methods is done based on their types. This paper addresses the issue of importance of transform in image compression and selecting particular transform for image compression. A comparative study of performance of a variety of different image transforms is done base on compression ratio, entropy and time factor.

THE FLOW OF IMAGE COMPRESSION CODING:

Image compression coding is to store the image into bit-stream as compact as possible and to display the decoded image in the monitor as exact as possible. Now consider an encoder and a decoder as shown in Fig. 1.3. 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. 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. The full compression flow is as shown in Fig.

Fig. 1.3 The basic flow of image compression coding

The compression ratio is defined as follows:

$$Cr = \frac{n1}{n2}, \qquad (1.2)$$

where $n1$ is the data rate of original image and $n2$ is that of the encoded bit-stream.

In order to evaluate the performance of the image compression coding, it is necessary to define a measurement that can estimate the difference between the original image and the decoded image. Two common used measurements are the Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR), which are defined in (1.3) and (1.4), respectively. f(x,y) is the pixel value of the original image, and f'(x,y)is the pixel value of the decoded image. Most image compression systems are designed to minimize the MSE and maximize the PSNR.

$$MSE = \sqrt{\frac{\sum_{x=0}^{W-1}\sum_{y=0}^{H-1}[f(x,y) - f'(x,y)]^2}{WH}} \qquad (1.3)$$

$$PSNR = 20\log_{10}\frac{255}{MSE} \qquad (1.4)$$

The general encoding architecture of image compression system is shown is Fig. 1.4. The fundamental theory and concept of each functional block will be introduced in the following sections.

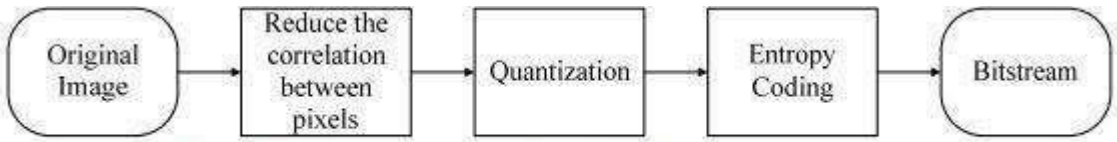


Fig. 1.4 The general encoding flow of image compression

Reduce the Correlation between Pixels

The reason is that the correlation between one pixel and its neighbor pixels is very high, or we can say that the values of one pixel and its adjacent pixels are very similar. Once the correlation between the pixels is reduced, we can take advantage of the statistical characteristics and the variable length coding theory to reduce the storage quantity. This is the most important part of the image compression algorithm; there are a lot of relevant processing methods being proposed. The best-known methods are as follows:

- Predictive Coding: Predictive Coding such as DPCM (Differential Pulse Code Modulation) is a lossless coding method, which means that the decoded image and the original image have the same value for every corresponding element.
- Orthogonal Transform: Karhunen-Loeve Transform (KLT) and Discrete Cosine Transform (DCT) are the two most well-known orthogonal transforms. The DCT- based image compression standard such as JPEG is a lossy coding method that will result in some loss of details and unrecoverable distortion.
- Subband Coding: Subband Coding such as Discrete Wavelet Transform (DWT) is also a lossy coding method. The objective of subband coding is to divide the spectrum of one image into the low pass and the high pass components. JPEG 2000 is a 2- dimension DWT based image compression standard.

QUANTIZATION

The objective of quantization is to reduce the precision and to achieve higher compression ratio. For instance, the original image uses 8 bits to store one element for every pixel; if we use less bits such as 6 bits to save the information of the image, then the storage quantity will be reduced, and the image can be compressed. The shortcoming of quantization is that it is a lossy operation, which will result into loss of precision and unrecoverable distortion. The image compression standards such as JPEG and JPEG 2000 have their own quantization

methods, and the details of relevant theory will be introduced in the chapter 2.

ENTROPY CODING

The main objective of entropy coding is to achieve less average length of the image. Entropy coding assigns codewords to the corresponding symbols according to the probability of the symbols. In general, the entropy encoders are used to compress the data by replacing symbols represented by equal-length codes with the code words whose length is inverse proportional to corresponding probability. The entropy encoder of JPEG and JPEG 2000 will also be introduced.

IMAGE COMPRESSION STANDARD:

In this chapter, we will introduce the fundamental theory of two well-known image compression standards –JPEG and JPEG 2000.

JPEG – JOINT PICTURE EXPERT GROUP

Fig. 2.1 and 2.2 shows the Encoder and Decoder model of JPEG. We will introduce the operation and fundamental theory of each block in the following sections.

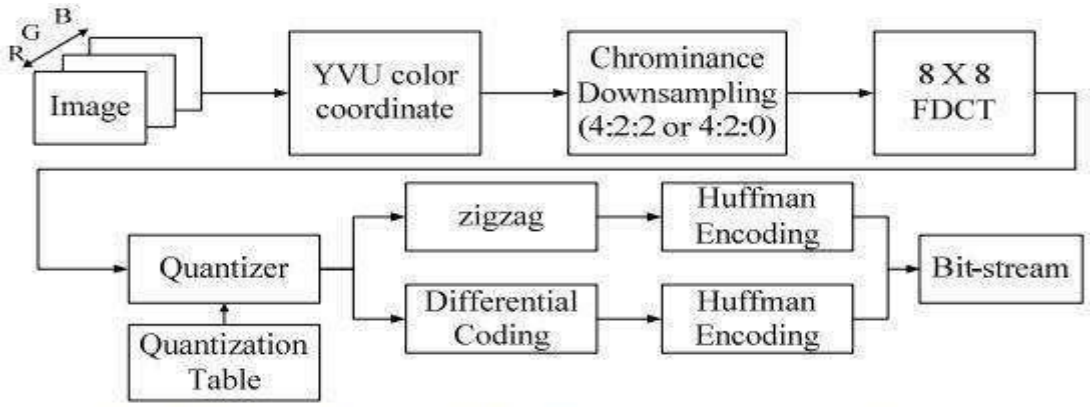


Fig. 2.1 The Encoder model of JPEG compression standard

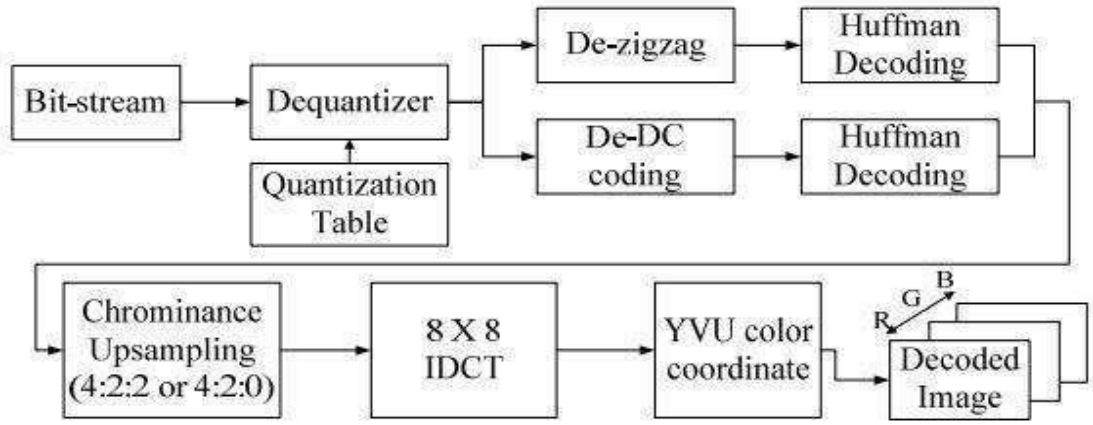


Fig. 2.2 The Decoder model of JPEG compression standard

DISCRETE COSINE TRANSFORM

The next step after color coordinate conversion is to divide the three color components of the image into many 8×8 blocks. The mathematical definition of the Forward DCT and the Inverse DCT are as follows:

Forward DCT

$$F(u,v)=\frac{2}{N}C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right]$$

for $u=0,\ldots,N-1$ and $v=0,\ldots,N-1$ **(2.1)**

where $N=8$ and $C(k)=\begin{cases}1/\sqrt{2} & \text{for } k=0\\ 1 & \text{otherwise}\end{cases}$

Inverse DCT

$$f(x,y)=\frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}C(u)C(v)F(u,v)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right] \quad \textbf{(2.2)}$$

for $x=0,\ldots,N-1$ and $y=0,\ldots,N-1$ where $N=8$

The f(x,y) is the value of each pixel in the selected 8×8 block, and the F(u,v) is the DCT coefficient after transformation. The transformation of the 8×8 block is also a 8×8 block composed of F(u,v).

The DCT is closely related to the DFT. Both of them taking a set of points from the spatial domain and transform them into an equivalent representation in the frequency domain. However, why DCT is more appropriate for image compression than DFT. The two main reasons are:

1. The DCT can concentrate the energy of the transformed signal in low frequency, whereas the DFT can not. According to Parseval's theorem, the energy is the same in the spatial domain and in the frequency domain. Because the human eyes are less sensitive to the low frequency component, we can focus on the low frequency component and reduce the contribution of the high frequency component after taking DCT.
2. For image compression, the DCT can reduce the blocking effect than the DFT.

After transformation, the element in the upper most left corresponding to zero frequency in both directions is the "DC coefficient" and the rest are called "AC coefficients."

Quantization in JPEG:

Quantization is the step where we actually throw away data. The DCT is a lossless procedure. The data can be precisely recovered through the IDCT (this isn't entirely true because in reality no physical implementation can compute with perfect accuracy). During Quantization every coefficients in the 8×8 DCT matrix is divided by a corresponding quantization value. The quantized coefficient is defined in (2.3), and the reverse the process can be achieved by the (2.4).

$$F(u,v)_{Quantization} = round\left(\frac{F(u,v)}{Q(u,v)}\right) \qquad (2.3)$$

$$F(u,v)_{deQ} = F(u,v)_{Quantization} \times Q(u,v) \qquad (2.4)$$

The goal of quantization is to reduce most of the less important high frequency DCT coefficients to zero, the more zeros we generate the better the image will compress. The matrix Q generally has lower numbers in the upper left direction and large numbers in the lower right direction. Though the high-frequency components are removed, the IDCT still can obtain an approximate matrix which is close to the original 8×8 block matrix. The JPEG committee has recommended certain Q matrix that work well and the performance is close to the optimal

$$Q_Y = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix} \qquad (2.5)$$

condition, the Q matrix for luminance and chrominance components is defined in (2.5) and(2.6)

$$Q_C = \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix} \qquad (2.6)$$

ZIGZAG SCAN:

After quantization, the DC coefficient is treated separately from the 63 AC coefficients. The DC coefficient is a measure of the average value of the original 64 image samples. Because there is usually strong correlation between the DC coefficients of adjacent 8×8 blocks, the

quantized DC coefficient is encoded as the difference from the DC term of the previous block. This special treatment is worthwhile, as DC coefficients frequently contain a significant fraction of the total image energy. The other 63 entries are the AC components. They are treated separately from the DC coefficients in the entropy coding process.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

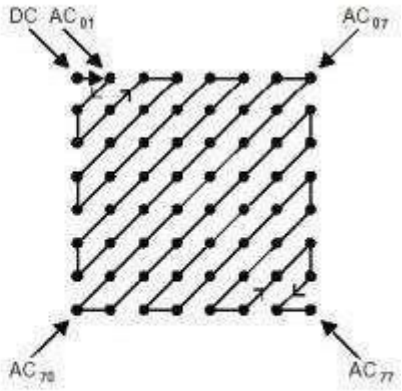
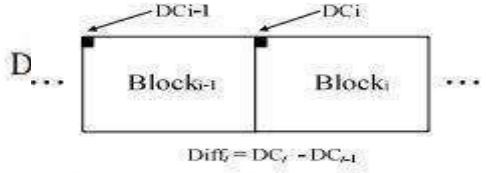


Fig. 2.3 The zigzag scan order

Entropy Coding in JPEG Differential Coding:

The mathematical representation of the differential coding is:



(2.7)

Fig. 2.4 Differential Coding

We set DC0 = 0. DC of the current block DCi will be equal to DCi-1 + Diffi . Therefore, in the JPEG file, the first coefficient is actually the difference of DCs. Then the difference is encoded with Huffman coding algorithm together with the encoding of AC coefficients.