Image Compression Techniques: A Survey in Lossless and Lossy algorithms.

Article *in* Neurocomputing · March 2018

DOI: 10.1016/j.neucom.2018.02.094

3 authors:

Abir Jaafar Hussain
Liverpool John Moores University
184 PUBLICATIONS **961** CITATIONS
SEE PROFILE

Ali Al-Fayadh
Al-Nahrain University
19 PUBLICATIONS **16** CITATIONS
SEE PROFILE

Naeem Radi
Liverpool John Moores University
27 PUBLICATIONS **80** CITATIONS
SEE PROFILE

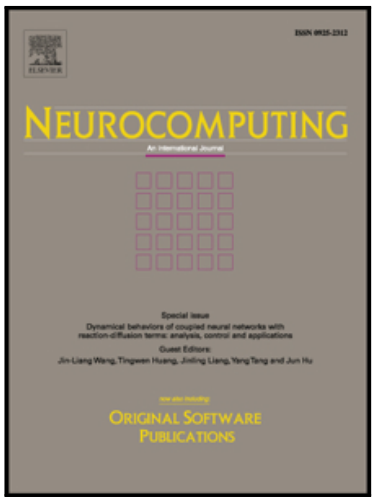


Image Compression Techniques: A Survey in Lossless and Lossy algorithms.

A.J. Hussain , Ali Al-Fayadh , Naeem Radi

PII: S0925-2312(18)30293-5
DOI: 10.1016/j.neucom.2018.02.094
Reference: NEUCOM 19409

To appear in: *Neurocomputing*

Received date: 23 October 2017
Revised date: 10 February 2018
Accepted date: 24 February 2018

Please cite this article as: A.J. Hussain , Ali Al-Fayadh , Naeem Radi , Image Compression Techniques: A Survey in Lossless and Lossy algorithms., *Neurocomputing* (2018), doi: 10.1016/j.neucom.2018.02.094

Image Compression Techniques: A Survey in Lossless and Lossy algorithms.

A. J. Hussain[a,*], Ali Al-Fayadh[b], Naeem Radi[c]

[a]*Applied Computing Research Group,*
Department of Computer Science, Liverpool John Moores University, Byrom Street, Liverpool, L3 3AF, UK

[b]*Al-Nahrain University*
Department of Mathematics and Computer Applications
Baghdad, Iraq

[c]*Al-Khawarizmi international college*
Abu Dhabi, Al Bahia Abu Dhabi, UAE

ABSTRACT

The bandwidth of the communication networks has been increased continuously as results of technological advances. However, the introduction of new services and the expansion of the existing ones have resulted in even higher demand for the bandwidth. This explains the many efforts currently being invested in the area of data compression. The primary goal of these works is to develop techniques of coding information sources such as speech, image and video to reduce the number of bits required to represent a source without significantly degrading its quality.

With the large increase in the generation of digital image data, there has been a correspondingly large increase in research activity in the field of image compression. The goal is to represent an image in the fewest number of bits without losing the essential information content within**.** Images carry three main type of information: *redundant*, *irrelevant*, and *useful*. Redundant information is the deterministic part of the information, which can be reproduced without loss from other information contained in the image. Irrelevant information is the part of information that has enormous details, which are beyond the limit of perceptual significance (i.e., psychovisual redundancy). Useful information, on the other hand, is the part of information, which is neither redundant nor irrelevant. Human usually observes decompressed images. Therefore, their fidelities are subject to the capabilities and limitations of the Human Visual System.

*Corresponding author Tel.: +44(0)1512312458, Fax: +44(0)1512074594
Email: a.hussain@ljmu.ac.uk

This paper provides a survey on various image compression techniques, their limitations, compression rates and highlights current research in medical image compression.

1. INTRODUCTION

Images are important representative objects. They can represent transmitted television or satellite pictures, medical or computer storage pictures and much more [199]. When a two-dimensional light intensity signal is sampled and quantised to create a digital image, a huge amount of data is produced. The size of the digitised picture could be so great that results in impractical storage or transmission requirements. Image compression deals with this problem such that the information required to represent the image is reduced thus making the transmission or storage requirements of images more practical.

The applications of image compression for transmission purposes are limited by real-time considerations. On the other hand, the applications of image compression for storage purposes are less strict. This is because the algorithm is not performed in real time, and therefore buffering is not required to match the output generated at the encoder to the transmission rate of the communication channel [1].

There are two types of compression methods, lossless and lossy image compression. In the former method, the compressed image should be an exact replica of the original image. Lossless image compression has wide applications such as the archival of medical [201-203] or business documents and digital radiography where any loss of information in the original image can result in improper diagnosis [2]. Other applications of lossless compression include the compression of image for camera system [200], the storage and transmission of thermal images captured by Nano-satellite [204] and remote sensing applications such as monitoring forest fires and determining the soil moisture [205]. The latter method is the most common in image compression, where some of the information of the original image will be lost and the purpose of the lossy method is to increase the compression rate on the expense of the accuracy of the reconstructed image. The compressed images include some distortion and the measure of the efficiency of the compression algorithm is considered with respect to the resulting distortion, the data compression ability and the implementation complexity of the algorithm [3], [127], [89]. The applications of lossy image compression include the transmission of images through the web [206], and the construction of image vegetation [207]. Kozhemiakin et. al. [208] indicated that while remote sensing images are widely

compressed using lossless compression to preserve the image quality, small compression ratio is achieved and suggested lossy image compression technique based on discrete cosine transform.

1.1 Digital Representation of Images

An image could be defined as a two-dimensional (2-D) function, $f(x, y)$, where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the brightness, intensity or grey level [4]. Before the image can be processed, it needs to be digitised. In a digital images $f(i, j)$, i and j are integer values as shown in Figure 1.

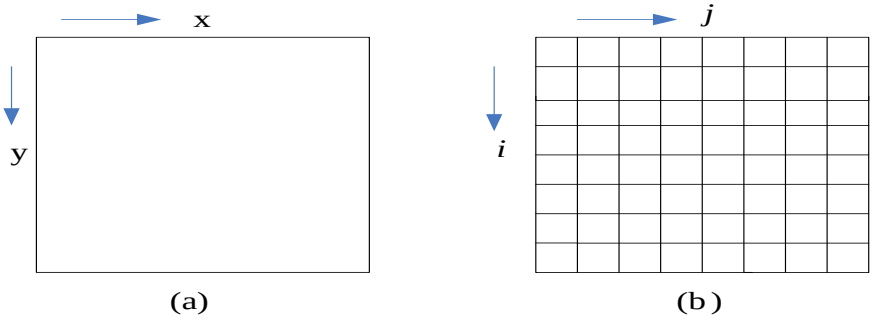
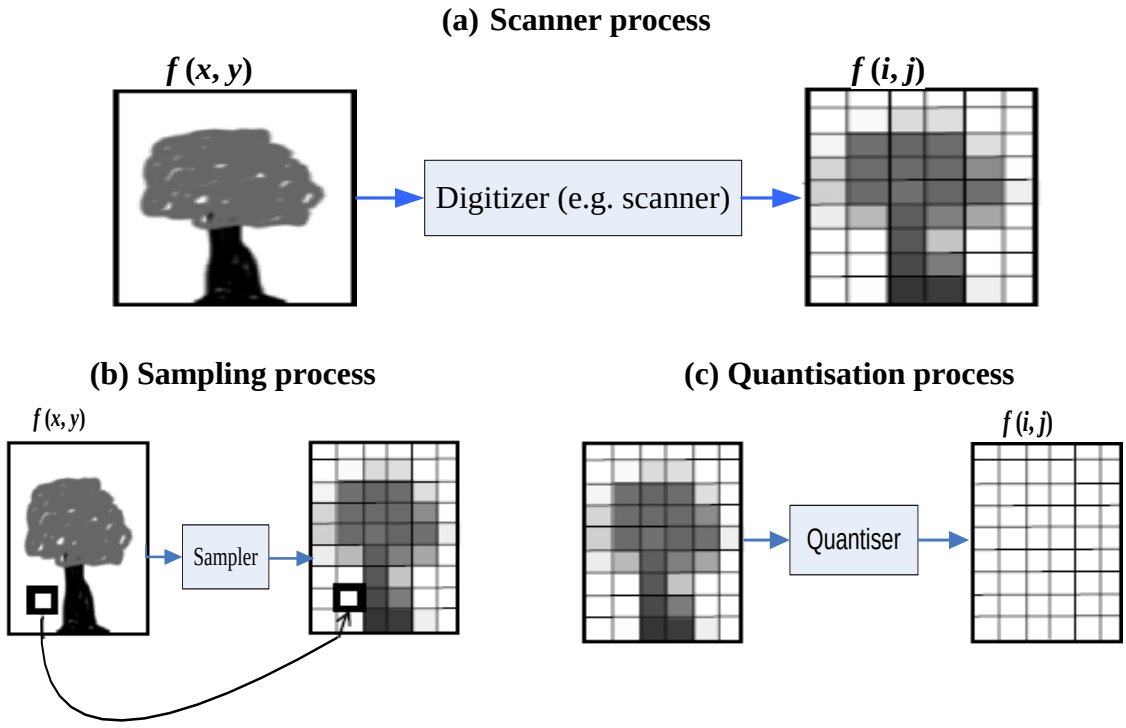


Figure 1: Digital representation of images.

(a) $f(x, y)$ – Continuous. (b) $f(i, j)$ – Digital



The continuous-tone image *f*(*x*,*y*) is scanned and its brightness is measured or sampled at discrete locations to form an element in a rectangular matrix

The sampled brightness values are still continuous. Next, each element in the matrix is quantised (replaced by an integer). Quantised values are called gray levels.

Figure 2: Transform process from continuous-tone image into digital image form.
(a) Scanner process, (b) Sampling process, and (c) Quantisation process

To transform a continuous tone image into digital format, a digitizer is required. The most commonly used digitizers are scanners and digital cameras. The two functions of a digitizer are sampling and quantising. Figure 2 shows the transformation steps.

A *grayscale* image measures light intensity only. Each pixel is a scalar proportional to the brightness. It is represented digitally by a matrix of $M \times N$ pixels. Pixels are typically unsigned 8-bit integer values, ranging from 0 to 255, and therefore allowing for 256 shades of gray. A typical example is given in Figure 3.

Figure 3: A typical greyscale image of resolution 512 × 512.

Colour images can be simulated, according to trichromatic theory, by linear combinations of the three primary colours red, green and blue, resulting in what is known as a colour space. A colour image measures the intensity and chrominance of light. Each colour pixel is a vector of colour components. Common colour spaces are RGB (red, green and blue), HSV (hue, saturation, value), and CMYK (cyan, magenta, yellow, black), which are used in the printing industry [5]. Colour images consist of three matrices representing each pixel's coordinates in some or other colour space. Since each of these matrices consists of 8-bit values, each pixel has 24-bit precision. This is known as 24-bit colour.

For storage purposes, pixel values need to be quantised. The brightness in greyscale images is usually quantised to Z levels, so $f(x, y) \in \{0, 1, \ldots, Z-1\}$. If Z has the form 2^L, the image is referred to as having L *bits per pixel*. Many common greyscale images use 8- bits per pixel, giving 256 distinct grey levels. This is a rough bound on the number of different intensities the human visual system is able to discern [6].
Coding of colour signals is no different in principle from coding of luminance signals and researchers have achieved visual "gain" that colour pictures give in comparison with their monochrome counterparts for very little extra effort (an increase in the bit-rate around 10%) [7].

1.2 Digital Image Compression

Digital image compression is a very popular research topic in the field of multimedia processing.
An image can be compressed due to the following reasons: within a single image, there exists significant correlation or redundancy among neighbouring samples or pixels. This correlation is referred as spatial correlation or redundancy. For data acquired from multiple sensors (such as satellite images), there exists significant correlation or redundancy among samples from these sensors. This correlation or redundancy is called spectral correlation or redundancy. Thus, the compression is achieved by taking advantage of the redundancy that exists in images. If the redundancies are removed prior to compression, a more effective compression can be achieved.

Image compression system consists of two parts: the compressor, and the decompressor. The compressor consists of a preprocessing stage and encoding stage, whereas the decompressor consists of a decoding stage followed by a postprocessing stage. A systematic view of the compression process is depicted in Figure 4**.** Before encoding, pre-processing is performed to prepare the image for the encoding process, and consists of any number of operations that are application specific. After the compressed file has been decoded, postprocessing can be performed to eliminate some of the potentially undesirable artefacts gained due to the compression process.
The compressor part can be further broken down into stages as depicted in Figure 5**.** The first stage in pre-processing is data reduction. The image data can be reduced by grey level and/or spatial quantisation, or can undergo any desired image improvement (for example, noise removal). The second step in processing is the mapping process, which maps the original

image data into another mathematical space where it is easier to compress the data. Next, as part of the encoding process, is the quantisation stage, which takes the potentially continuous data from the mapping stage and puts it in discrete form.

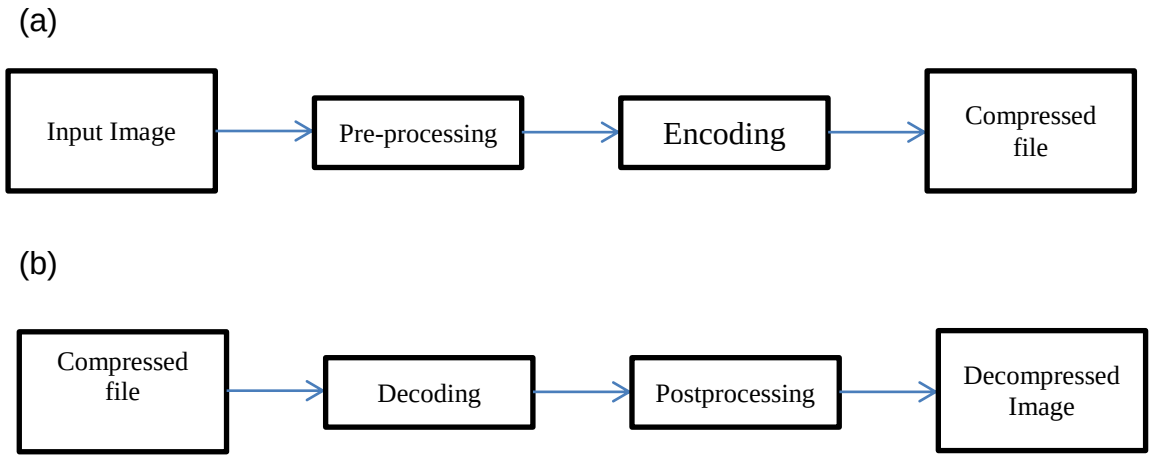


Figure.4: compression system model.

(a) Compression. (b) Decompression [8].

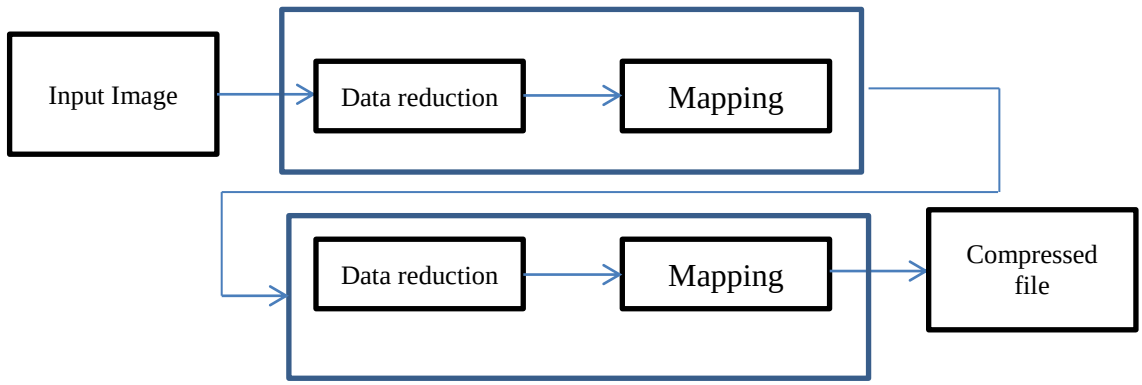


Figure 5: The Compressor [8].

The final stage of encoding involves coding the resulting data. The compression algorithm may consist of all the stages, or it may consist of only one or two of those stages.

The decompressor can be further broken down into the stages shown in Figure 6. The decoding process is divided into two stages. In the first stage, the decoder takes the

compressed file and reverses the original coding by mapping the codes to the original, quantised values. Next, these values are processed by a stage that performs an inverse mapping to reverse the original mapping process. Finally, the image may be postprocessed to enhance the look of the final image.

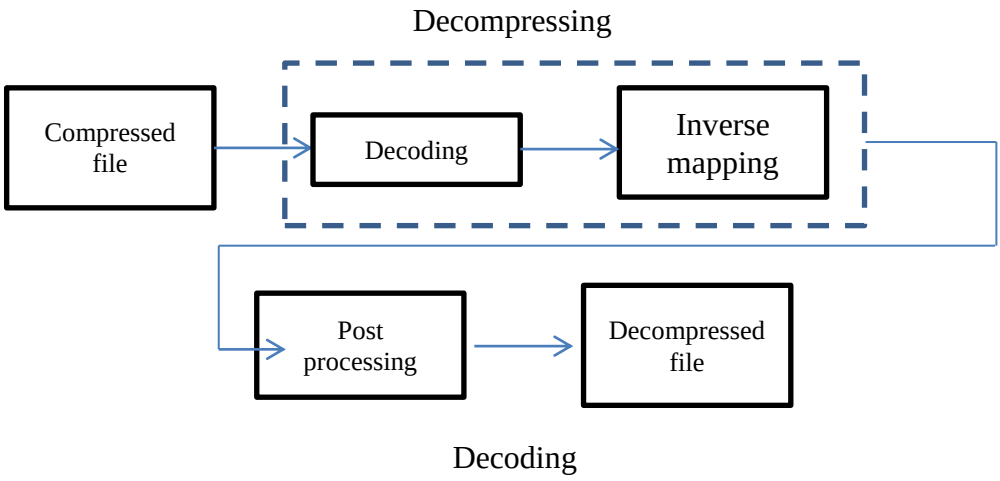


Figure 6: The decompressor [8].

The development of a compression algorithm is highly application specific. During the preprocessing stage of compression, processing such as enhancement, noise removal, or quantisation is applied. The goal of preprocessing is to prepare the image for the encoding process by eliminating any irrelevant information which is defined by the application.

The mapping process is important because image data tends to be highly correlated. If the value of one pixel is known, it is likely that the adjacent pixel value is similar. By finding a mapping equation that decorrelates the data, this type of data redundancy can be removed. It is important to note that the quantisation process is not reversible, so some information may be lost during quantisation. Additionally, since it is not a reversible process, the inverse process does not exist, so it does not appear in the decompression model.

The coding stage of any image compression algorithm is very important. The coder provides a one-to-one mapping; each input is mapped to a unique output by the coder, so it is a reversible process. The code can be an *equal length cod*e, where all the codewords are the same size, or an *unequal length code* with variable length codewords. In most cases, an unequal length code is the most efficient for data compression, but requires more overhead in the coding and decoding stages [8].

The reduced file created by the compression process is called the *compressed file* and is

used to reconstruct the image, resulting in the *decompressed image*. The ratio of the original (uncompressed file) and the compressed file is referred to as the *compression ratio*. The compression ratio is determined according to the following equation:

$$Compression\ Ratio = \frac{Uncompressed\quad file\quad size\,(bytes)}{Compressed\quad file\quad size\ \ (bytes)} \tag{1}$$

Another image compression terminology is the number of *bits per pixel*. For $N \times N$ image, this is determined as follows:

$$Bits\ Per\ Pixel = \frac{Number\quad of\quad bits}{Number\quad of\quad Pixels} \tag{2}$$

1.3 Quality Measures in Image Coding

One of the main problems in lossy compression consists in the assessment of the quality of the compressed image [116-117]. Therefore, in order to determine the efficiency of a compression algorithm, we need to quantify the difference. The difference between the original and the reconstruction image is often called the *distortion* [9]. To determine exactly what information is important, and to be able to measure image quality, we need to define image fidelity criteria. The performance of a compression algorithm will be good if the *distortion* is small [9]. This means that the fidelity or quality of the reconstructed image is high when the difference between the original and the reconstructed image is small.

Fidelity criteria can be divided into two classes; *objective fidelity criteria*, and *subjective fidelity criteria.* The objective fidelity criteria provide us with equations that can be used to measure the amount of error in a processed image in comparison to the original image [8]. The objective criteria are widely used, however they are not necessarily correlated with the human perception of image quality. For example, an image with a low error as determined by an objective measure may actually look much worse than an image with a high error metric.

. These types and degrees of degradation depend on the situation for which the image is being used. Because human responses are difficult to model mathematically, and there is difficulty in objectively reporting the results since they depend on observer's judgment; practical approximate measures of distortion are used to determine the quality of the reconstructed image $g(x, y)$ to its original $f(x, y)$. One of the simplest and most popular method is to use the

difference between *f* and *g*. In its most basic form is the mean square error (*MSE*), and is often represented by the symbol σ_q^2, which is determined according to the following equation [9], [11]:

$$MSE = \sigma_q^2 = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(f(x,y) - g(x,y) \right)^2 \qquad (3)$$

where *f* and *g* are $M \times N$ size images. *MSE* is often called *quantisation error variance* σ_q^2. This is a useful measure as it gives an average value of the energy lose in the lossy compression of the original image *f*.

A human observing two images affected by the same type of degradation will generally judge the one with the smaller *MSE* to be closer to the original. A very small *MSE* can be taken to mean that the image is very close to the original [109], [64], [137]. However the *MSE* has some problems when images with different types of degradation are compared, the one with the smallest *MSE* will not necessarily seem closest to the original. In many applications, the Signal-to Noise- Ratio (*SNR*) is expressed in terms of *MSE*, and is defined as:

$$SNR = 10 \log_{10} \frac{\sigma^2}{\sigma_q^2} = 10 \log_{10} \left[\frac{\frac{1}{M \times N} \sum_{y=1}^{N} \sum_{x=1}^{M} \left(f(x,y) \right)^2}{\frac{1}{M \times N} \sum_{y=1}^{N} \sum_{x=1}^{M} \left(f(x,y) - g(x,y) \right)^2} \right] \qquad (4)$$

where σ^2 is the variance of the original image, and σ_q^2 is the *MSE*. The *SNR* is measured in decibels (dB) and gives a good indication of the ratio of signal to noise reproduction.

A more subjective qualitative measurement of quality is the Peak Signal-to-Noise Ratio (*PSNR*). This measurement is used when there is more interest in the size of the error relative to the peak value of the image than with the size of the error relative to the average squared value of the image. The *PSNR* between two images having 8-bits per pixel is determined as follows:

$$PSNR = 10 \log_{10} \left[\frac{255^2}{MSE} \right] \text{ (dB)} \qquad (5)$$

In general, if *b* is the number of bits per pixel, the *PSNR* is given by:

$$PSNR = 10 \log_{10} \left[\frac{\left(2^b - 1\right)^2}{MSE} \right] \text{ (dB)} \qquad (6)$$

From equations 5 and 6, we can conclude that the *PSNR* measures the strength of the signal relative to the strength of the error that is introduced by the compression. Consequently, a high *PSNR* indicates a higher image quality.

These objective measures are often used because they are easy to generate and seemingly unbiased, but these metrics are not necessarily correlated to our perception of an image. Subjective measures are better than the objective measures for image evaluation, if the goal is to achieve high-quality images as defined by our visual perception [8]. Subjective fidelity measures can be classified into three classes. The first type is referred to as *impairment tests*, where the test subjects score the images in terms of how bad they are. The second type is *quality tests*, where the test subjects rate the images in terms of how good they are. The third type is called *comparison tests*, where the images are evaluated on a side-by-side basis.

2. LOSSLESS COMPRESSION METHODS

2.1 Huffman Coding

Huffman coding is one of the oldest image data compression methods. It was developed by Huffman [14] and is used to reduce coding redundancy without degrading the quality of the reconstructed image. The basic idea of the Huffman coding approach is based on data statistics, and it provides an optimal way of assigning variable length codewords to an alphabet of symbols with a known probability distribution. It represents the symbols of the alphabet by variable code length, depending on their probability of occurrence (the more probable a symbol is, the shorter the code is assigned). Shortest average code length is achieved by Huffman coding, provided that all source symbol probabilities are an exact power of ½. Huffman coding algorithm can be described as follows:

Step 1- List the probabilities of various grey levels (the source symbols) in the image. These probabilities are tabulated in a descending order with the highest probability at the top and the lowest probability at the bottom. Produce a node set by making these probabilities the leaves of a binary tree.

Step 2- Take two nodes with the two lowest probabilities from the set, and generate a new probability representing the sum of these two probabilities. The order of the probabilities is reorganised in a descending order for the proceeding process.

Step 3- Produce a parent node with the new probability, and mark the branch of its top (or left) child node as 1 and the branch of its bottom (or right) child node as 0, respectively.

Step 4- Update the node set by replacing the two nodes with the two lowest probabilities for the newly produced node. If the node set contains only one node, quit. Otherwise go to step 2.

The bit assignment procedure is performed in the backward direction, that is, if the node set contains only one node and the algorithm is stopped, we go backwards and assign to the two joined probabilities in the previous stage the symbol of the next stage plus the binary symbols 0 and 1 assigned to each probability. Such process is repeated until the first stage of the process is achieved. To illustrate the main concept of Huffman coding, consider the following example.

Assume a digital image has seven source symbols. {s_1, s_2, s_3, s_4, s_5, s_6, s_7}, with the following probabilities: {0.40, 0.20, 0.10, 0.08, 0.05, 0.04, 0.03}, which can be coded using a fixed word length of 3 bits/pixel. The procedure for obtaining Huffman code is shown in Figure 7. The probabilities of s_6 and s_7 have the lowest values. Therefore, they are combined with each other to form the new probability of value 0.07. This means that the order of the probabilities will be rearranged. The process is repeated until we achieve the final stage of the process where two probabilities remain (in this example 0.5 and 0.4). The bit assignment procedure is performed in the backward direction as show in Figure 8.

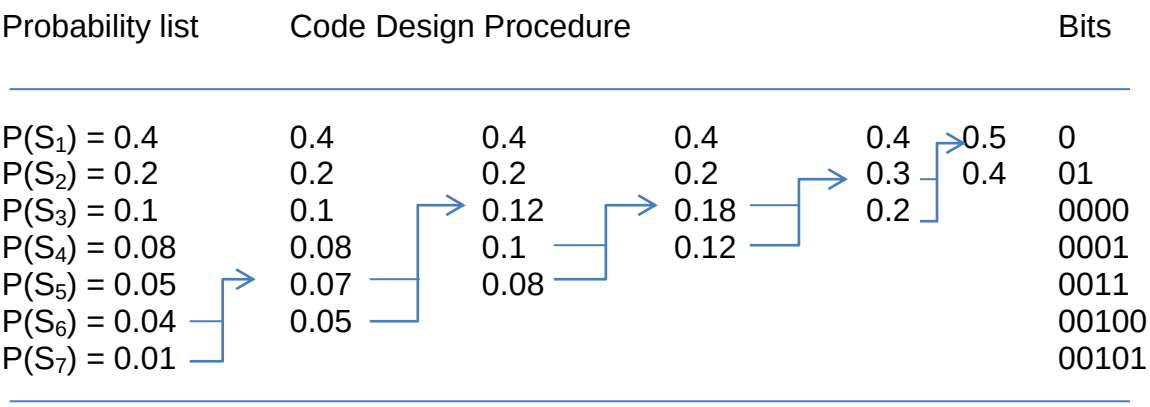


Figure 7: Huffman coding; fixed word length = 3 bits; average word length = 2.07 bits.

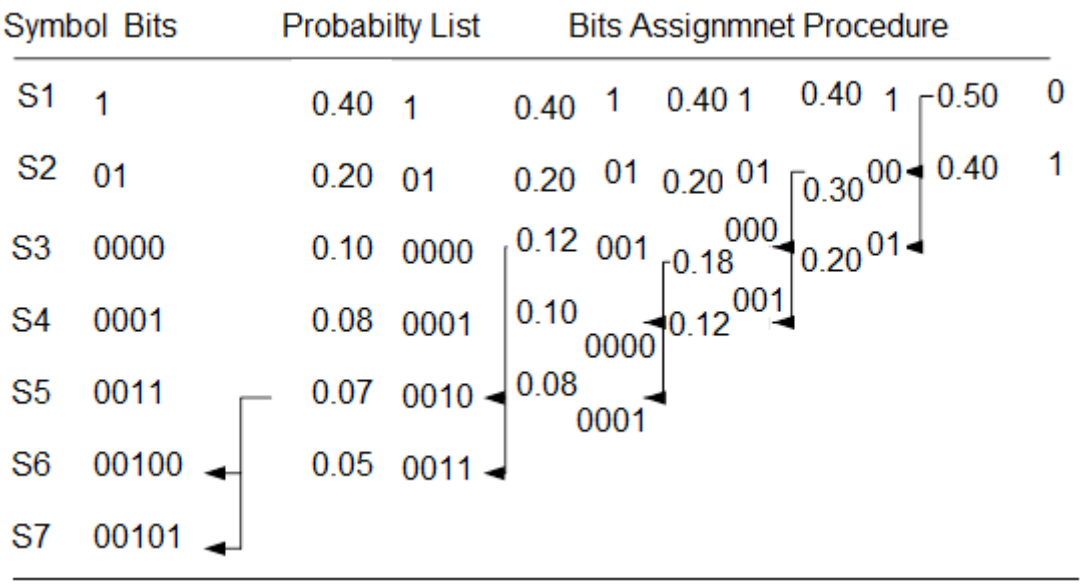


Figure 8: Huffman coding, bit assignment procedure.

The average codewords length assigned to various grey level values is defined as [4]:

$$L_{ave} = \sum_{i=1}^{L} I(r_i)P(r_i) \tag{7}$$

where r_i , i = 1, 2, …, L represent the ith grey level of an L- grey level image, $I(r_i)$ is the number of bits required to represent the grey level, and $P(r_i)$ is its probability. Therefore from Figure 8 the average code length of Huffman codewords in this example is:

$$L_{ave} = (0.4)\,(1) + (0.2)\,(2) + (0.1)\,(4) + (0.08)\,(4) + (0.05)\,(4) + (0.04)\,(5) + (0.03)\,(5)$$

$$L_{ave} = 2.07 \text{ bits}$$

In the above example, it is clear that, the average number of bits will be reduced from 3 bits for a fixed coding to 2.07 bits for variable length coding using Huffman coding. This shows the efficiency of Huffman coding. However, it must be noticed that Huffman coding is optimal only if the probability distribution of a source is known and each source symbol is encoded in integral number of bits. As different images differ in their own probability distribution of symbols, Huffman coding is not necessarily optimal for all images. Nevertheless, Huffman coding is one of the most important compression techniques with applications in JPEG (*Joint Photographic Experts Group*) and other standards.

2.2 Arithmetic Coding

Arithmetic coding is a variable-length coding procedure that, like Huffman coding, is designed to reduce coding redundancy. It is only optimal when all symbols probabilities are an integral power of ½ [7]. Glen and Langdon [15] provided a good introduction to arithmetic coding.

The basic idea in arithmetic coding is to divide the interval between 0 and 1 and to denote the range of the input message, into a number of smaller intervals corresponding to the probabilities of the message symbols. Each probability is represented by a two end interval; the left end is closed while the right end is open. The next input symbol selects one of these intervals, and the procedure is repeated. In this way, the selected interval narrows with every symbol, and at the end, any number inside the final interval can be used to represent the message. Suppose that the current message is specified in the interval [low_{old} , $high_{old}$) and the range of the present incoming symbol is [Q_{a1} , Q_{a2}), this means that the new range of the message is:

$$Low_{new} = low_{old} + (range_{old} \times Q_{a1}) \quad (8)$$

$$high_{new} = low_{old} + (range_{old} \times Q_{a2}) \quad (9)$$

and

$$range_{old} = high_{old} - low_{old} \quad (10)$$

To explain the main concept of arithmetic coding, consider the following example [16]. Assume that the source symbols are {a_1, a_2, a_3, a_4} and the probabilities of these symbols are {0.4, 0.3, 0.2, 0.1}, respectively. The interval [0, 1) can be divided as four subintervals: [0.0, 0.4), [0.4, 0.7), [0.7, 0.9), and [0.9, 1) Table 1 shows source symbols and their probabilities. Consider the message a_2, a_1, a_3, a_4, with the initial probability interval between 0 and 1. The first symbol a_2, in the interval [0.4, 0.7), will scale the initial range of the message to that interval. The next symbol is a_1 which lies in the interval [0.0, 0.4). Therefore, the message interval is now scaled according to Equations 8, 9 and 10 between [0.4, 0.52). Symbol a_3 will scale the message to [0.484, 0.508). The final symbol is a_4 which lies between [0.9, 1.0) and will give the final interval of the message which is [0.5056, 0.508) (refer to Figure 9). The compression output can be any number in the last interval.

Table 1: Source symbols, their probabilities and the initial subintervals.

Symbol	Probability	Initial subinterval
a_1	0.4	[0.0 , 0.4)
a_2	0.3	[0.4 , 0.7)
a_3	0.2	[0.7 , 0.9)
a_4	0.1	[0.9 , 1.0)

If the two ends of the final interval of the message are provided to the decoder, the decoder will determine the first symbol of the message which is a_2 (in our example), since the range of the interval lies completely within the whole range of this symbol. The next process is to perform similar steps to the encoder; this in turn will lead to the next symbol which is a_1. This process is continued until the whole message is retrieved.

To terminate the message and separate it from other incoming messages, end of file symbol known to both ends of the compression system is transmitted.

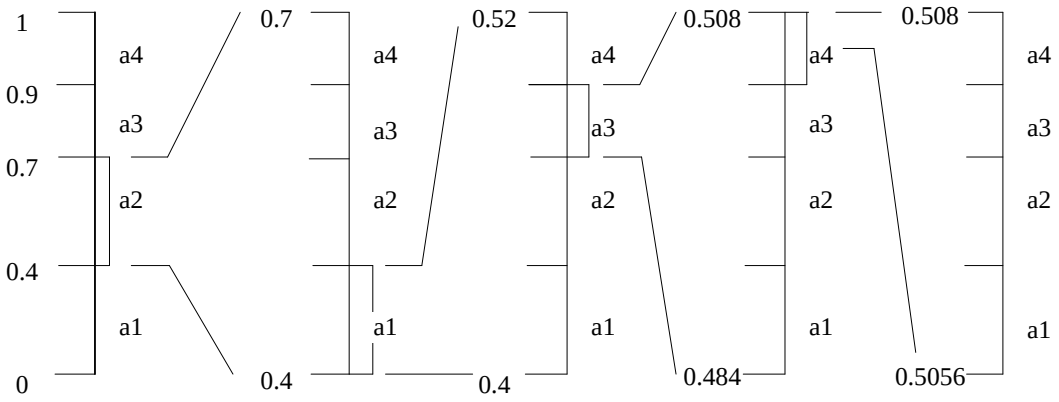


Figure 9: Representation of the arithmetic coding process with the interval scaled up at each stage.

There are several problems in arithmetic coding that can affect its efficiency, some of these problems are:

1. An arithmetic coder produces only one codeword, a real number in the interval [0, 1), for the entire message to be transmitted. We cannot perform decoding process until we received all bits representing this real number.
2. Arithmetic coding is an error sensitive compression scheme. A single bit error can corrupt the entire message.

However, arithmetic coding serves a very important role in imaging standards such as JBIG (*Joint Bi-level Image Experts Group*) and JPEG (*Joint Photographic Experts Group*).

2.3 Lossless Predictive Coding

In lossless predictive image compression approach [2], interpixel redundancies are removed by predicting the current pixel value using closely spaced pixel values and generating new values for coding. The new values represent the error generated from the subtraction of the predicted value from the original value. Figure 10 shows the complete structure of lossless predictive coding system.

(a)

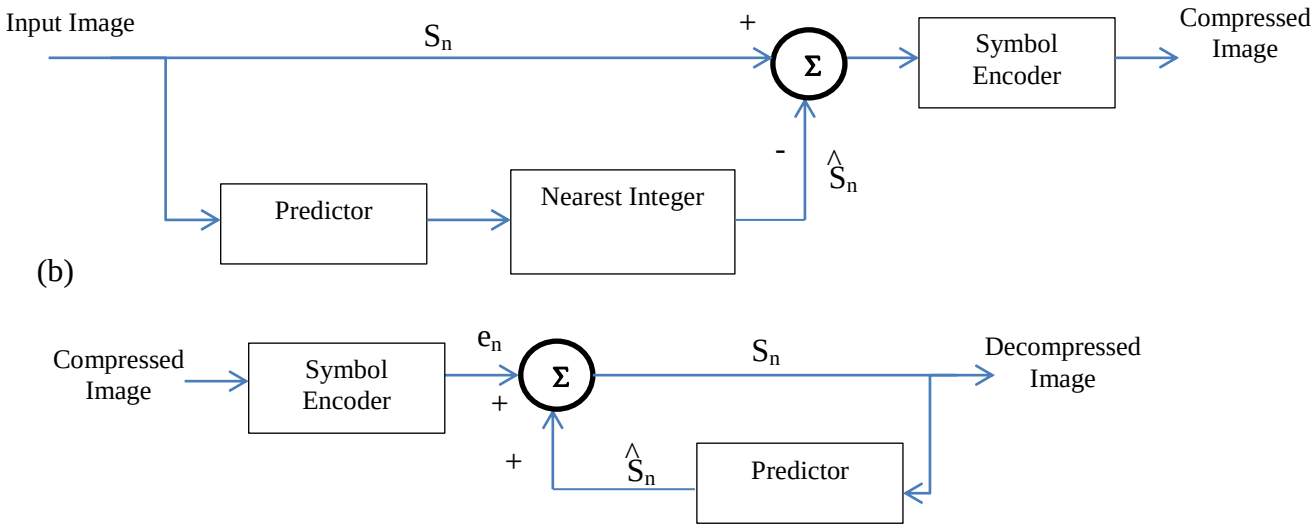


Figure 10. The complete structure of the lossless predictive coding system. (a) The encoder; (b) The decoder [2].

As shown in Figure 10, lossless predictive coding system consists of two parts, the transmitter and the receiver. At the transmitter, the current pixel value is predicted using the closely spaced neighbourhood pixel values. The predicted value is generated using linear weighted combination of the previous pixel values as follows [2]:

$$\hat{S}_n = \text{round}\left[\sum_{i=1}^{m} w_i S_{n-i}\right] \qquad (11)$$

where w_i are the predictor parameters and the linear combination of the previous pixel values is rounded to its nearest integer value. The difference between the original and the predicted signal values:

$$e_n = S_n - \hat{S}_n, \qquad (12)$$

will be transmitted to the symbol encoder in which a variable length coding system is provided to encode the error value.

It may argue that this is a lossy rather than lossless image compression method because of the rounding operation at the transmitter. However, since there is no quantiser, the technique is considered lossless.

At the receiver, the same predictor is provided. The coded error signal is added to the predicted signal $\hat{S}(i)$ to produce the original signal:

$$S_n = e_n + \hat{S}_n \qquad (13)$$

The compression of the image is performed using variable length coding where coding redundancy is removed and the prediction operation provides the elimination of the interpixel redundancy.

3. LOSSY COMPRESSION SCHEMES FOR IMAGE CODING

Unlike other data sources, such as text, or numerical information, in which any errors are obvious, the human eye can compensate for some distortion in images. Such systems are referred as *lossy*. The motivation for lossy compression originates from the inability of lossless algorithms to produce as low bit rates as desired. At the encoder, there exists a quantiser which limits the number of bits required to represent the image. The purpose of the quantiser is to remove psychovisual redundancy. By allowing such 'lossy' compression, the performance of such a compression is no longer capped by the Shannon limit [136], [137]. This means that it becomes possible for lossy compression systems to achieve extremely high compression rates (less than one bit per pixel) [17]. The use of lossy compression is always a trade-off between the bitrate and the image quality. Depending on the application, a significant compression can be achieved if the resulting degradation can be tolerated for that certain application.

There are various approaches to lossy image compression such as *vector quantisation*, *predictive coding* and *transform coding*.

A combined system that uses the properties of two or more of these approaches is known as *hybrid coding*.. The general components of a lossy image compression technique are shown in Figure 11.

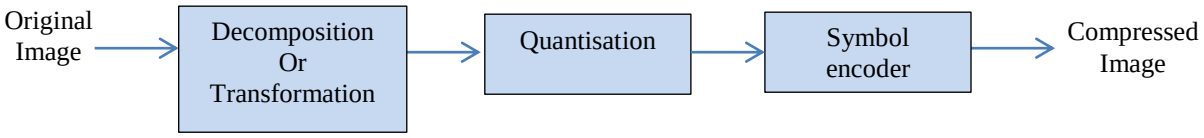


Figure 11: General components of a lossy compression technique.

As shown in Figure 11, the first stage is image decomposition or transformation. The purpose of this stage is to produce a more efficiently coded representation of image data. The second stage is quantisation and it aims to reduce the number of possible output symbols. The type and degree of quantisation has a significant effect on the quality of the reconstructed image and its bitrate. The third stage is symbol encoding, which may or may not be included in the process, depending on the lossy technique used. For example, symbol encoding is usually part of a transform coding method but it is not part of a vector quantization method. Most compression algorithms are either based on transformation first, followed by scalar quantisation and coding; or by direct vector quantization of the original image, skipping the transformation part [132].

3.1 Predictive Coding

A general paradigm in data compression is the concept of prediction. Prediction allows a compact representation of data by encoding the error between the data itself and the information predicted from past observations. If the predictor works well, predicted samples are similar to the actual input and the prediction error is small or negligible.

Pixels in images show a high degree of correlation among their neighbouring samples. This means that there exists a mutual redundancy in the raw data. Predictive coding is used to remove the mutual redundancy. Removing the mutual redundancy by decorrelating the data, much more efficient and better compressed coding of the signal can be obtained.

Predictive coding predicts the next pixel value based on a sequence of reproduced pixels values (previous values) obtained during the scanning of the image and encodes (quantises) the difference between predictive and actual value (the error signal) [8]. The better the prediction, the smaller the transmitted error, hence the better the coding process. If the current reproduced pixel is taken as the sum of the predicted pixel value and the quantised

error value between the current pixel and the predicted pixel, the prediction method is called differential pulse code modulation (DPCM) [113], [130].

There are two different parts in the predictive coding encoder, which are the predictor and the quantiser. The former is used to estimate the next value of the image signal using the previously coded elements, whereas the latter quantises the difference between the predicted value and the original value. The order of the predictor is the number of previous elements used in the prediction. A simplified block diagram of predictive coding system is illustrated in Figure 12.

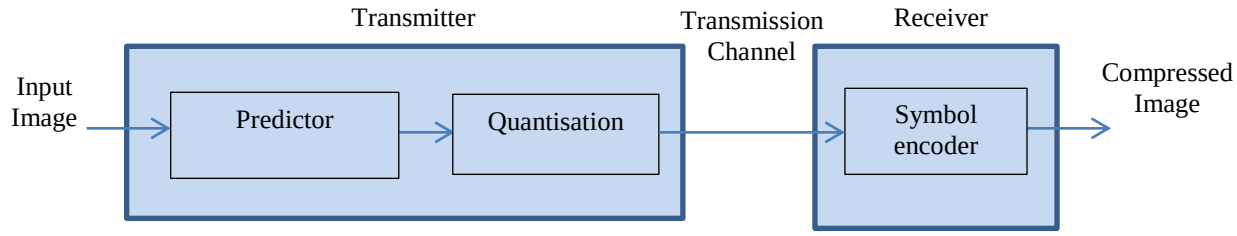


Figure 12: A simplified block diagram of predictive coding system.

The predictors are classified into linear and nonlinear types. For linear predictors, the previous samples are linearly coded to predict the current value, while nonlinear predictors use nonlinear functions for coding the previous samples. Implementation of nonlinear predictors is more complicated than the implementation of linear predictors. However, nonlinear predictors can provide better coding since the correlation among the image pixels is a nonlinear function.

A further classification of the predictors can be done according to the previous pixel values used in the prediction. The prediction can be one dimensional or two dimensional. A one-dimensional predictor uses the previous pixel values in the same row. A two-dimensional predictor uses pixels in the same row and in the previous rows to predict the current pixel value. Figure 13 shows various predictive structures.

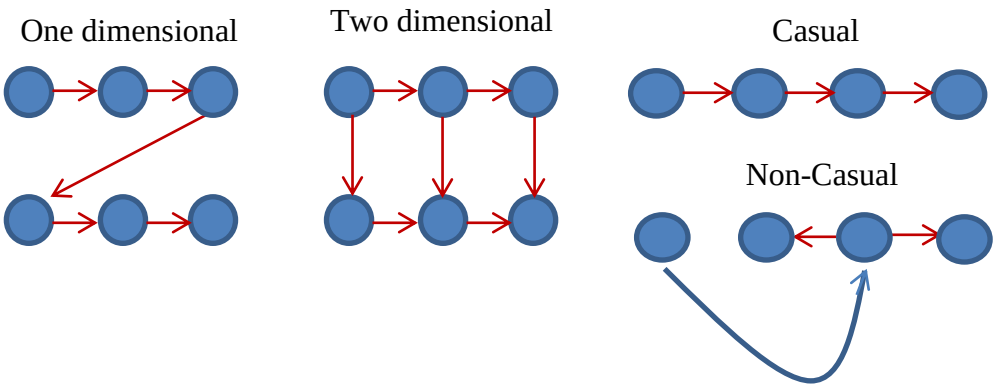


Figure 13: Various predictive structures.

Linear Predictor Structure

The idea behind linear prediction (LP) is that a sample can be approximated by a linear combination of previous samples. Using this approach, the sample *x*(n) at the instant *n* can be predicted as a weighted sum of *p* previous samples. The basic equation of linear prediction is given as follows:

$$x^*(n) = \sum_{k=1}^{p} a_k x(n-k) \tag{14}$$

where $x^*(n)$ is the estimated sample of the actual $x(n)$ value from the linear combination of samples, and a_1, a_2, …, a_p are the prediction coefficients. Linear predictor coefficients are determined such that the coefficients minimise the error between the actual and the estimated signal. The number, *p*, of previous samples decides the order of the model, and the higher number, the more accurate is the prediction. This will however also mean that the computation complexity increases [18]. The prediction error is expressed as:

$$e(n) = x(n) - x^*(n) = x(n) - \sum_{k=1}^{p} a_k x(n-k) \tag{15}$$

and the prediction gain is given by:

$$G_p = \frac{\sigma_x^2}{\sigma_e^2} \tag{16}$$

where σ_x and σ_e are the variances of the input signal and the prediction error signal, respectively.

LP has been extensively studied because it plays an important role in image compression standards such as JPEG due to its compression efficiency and its simplicity, as well as a number of efficient algorithms have been designed to determine the predictor's coefficients [19-24]. In order to minimise the error, the best, or optimal, values of $\{a_k\}$ have to be determined. This can be done by minimising the expectation value of the mean square error between the predictive value and the actual value, i.e. $E[(x(n)-x*(n))^2]$. To satisfy this condition, partial derivatives of the expected value, with respect to each of the coefficients is set to zero.

$$\frac{\partial E\left[\left(x(n)-x^*(n)\right)^2\right]}{\partial a_k}=\frac{\partial E\left[\left(x(n)-(a_1x(n-1)+a_2x(n-2)+...+a_px(n-p))\right)^2\right]}{\partial a_k} \tag{17}$$

$$=-2E\left[\left(x(n)-(a_1x(n-1)+a_2x(n-2)+...+a_px(n-p))\right)x(n-k)\right]=0,\qquad k=1,2,...,p$$

hence,

$$E\left[\left(x(n)-(a_1x(n-1)+a_2x(n-2)+...+a_px(n-p))\right)x(n-k)\right]$$
$$=E[(x(n)-x^*(n))x(n-k)]=0,\qquad k=1,2,...,p \tag{18}$$

Let the covariance of $x(k)$ and $x(j)$ be

$$R_{j,k}=\mathrm{E}\,[x(j)\,x(k)] \tag{19}$$

Then, the following set of equations can be obtained:

$$R_{n,n-k}=a_1R_{n-1,\,n-k}+a_2R_{n-2,\,n-k}+\ldots+a_pR_{n-p,\,n-k},\qquad k=1,2,\ldots,p \tag{20}$$

For optimum predictors, the minimum value of the mean square error can be found as follows:

$$\sigma_e^2=E[x(n)(x(n)-x^*(n))]=E[x(n)x(n)-x(n)x^*(n)]$$
$$=R_{n,n}-(a_1R_{n.n-1}+a_2R_{n,n-2}+...+a_pR_{n,n-p}) \tag{21}$$

where $R_{n,n}$ is the variance, $\boldsymbol{\sigma^2}$, of the input signal.

Consider the implementation of a first-order linear predictor where the following equation is satisfied [16]:

$$x^*(1) = a_1 x(0)\,, \tag{22}$$

and

$$R_{1,0} = a_1 R_{0,0}\,. \tag{23}$$

$R_{0,0}$ is simply the variance of the actual sequence. Therefore, the predictor coefficient, a_1, can be determined according to the following equation:

$$a_1 = \frac{R_{1,0}}{\sigma^2} = \rho_1 \tag{24}$$

where ρ_1 is the inter-element correlation coefficient of the image data in the direction in which the prediction is taken. Thus, from Equation 24, when n = 1, the mean square error is found to be:

$$\sigma_e^2 = R_{1,1} - a_1 R_{1,0} = \sigma^2(1-\rho_1^2) \tag{25}$$

On the other hand, for second order predictors:

$$x^*(2) = a_1 x(1) + a_2 x(0) \tag{26}$$

and

$$R_{2,1} = a_1 R_{1,1} + a_2 R_{0,1}\,,\; R_{2,0} = a_1 R_{1,0} + a_2 R_{0,0} \tag{27}$$

Therefore,

$$\rho_2 = a_1 + a_2\rho_1\,, \quad \rho_3 = a_1\rho_1 + a_2 \tag{28}$$

where $\rho_1 = \frac{R_{0,1}}{\sigma^2} = \frac{R_{1,0}}{\sigma^2}$, $\rho_2 = \frac{R_{2,1}}{\sigma^2}$, *and* $\rho_3 = \frac{R_{2,0}}{\sigma^2}$.

Solving the previous Equations in (28) gives the values of the predictor coefficients:

$$a_1 = \frac{(\rho_2 - \rho_1\rho_3)}{(1-\rho_1^2)}, \quad a_2 = \frac{(\rho_3 - \rho_1\rho_2)}{(1-\rho_1^2)} \tag{29}$$

The same procedure can be followed to implement higher order predictors. Various formulations for efficient computation of the predictor coefficients have been derived. The details of these derivations can be found in [25], [26], and [27].

If the order of the predictor, *p*, is infinite, the error samples can be made completely uncorrelated [28]. On the other hand, using a finite and sufficiently large p^{th} order autoregression process to predict the input signal, error sequences are made uncorrelated [29]. Therefore, increasing the number of samples used for prediction will not improve the estimate value [30]. In image compression, some researchers such as Stott [32] suggested that higher orders predictors are useful to improve the efficiency of the predictive algorithm. However, using more previous values in the predictor increases the complexity of computations for compression and decompression, and it has determined that using more than three of the previous values provides no significant improvement for resulting image [8]. It is also considered that fourth order predictors are sufficient to achieve the required benefit from the scheme [33].

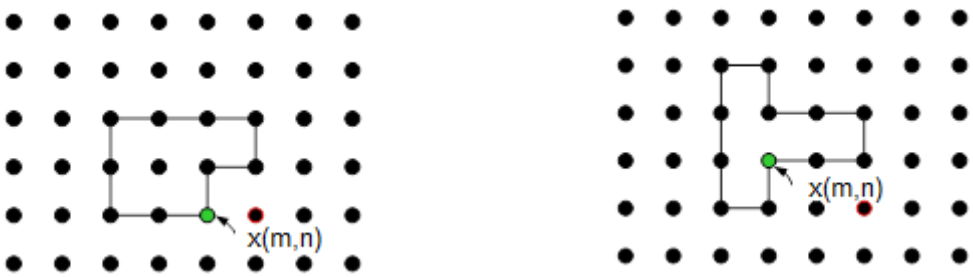
The nonlinear nature of images makes nonlinear predictors more appropriate. Unfortunately, there are not as mathematically tractable as linear predictors. In addition, they are time consuming and usually impossible to design optimal nonlinear predictors [34].

Two-dimensional Prediction

The idea of the DPCM can be extended to the two-dimensional space. In this case, a pixel can be predicted using its adjacent pixels in two dimensions, as in Equation (30)

$$x^*(m,n) = \sum_{(i,j)\in W}\sum a_{i,j}\,x(m-i,n-j) \tag{30}$$

where W is a two-dimensional prediction region and $a_{i,j}$ are the prediction coefficients. Prediction can be causal or non-causal. Examples of causal and non-causal predictions are shown in Figure 14 (a) and 14(b), respectively. In a causal prediction, the prediction of a sample depends only on the previous samples. In non-causal prediction, some future pixels are also used in the prediction.



(a) (b)

Figure 14: Examples of two-dimensional prediction. (a) Causal prediction. (b) Non causal prediction [31] .

It has been shown that for typical images, using more than four nearest pixels for the prediction of a sample cannot increase the prediction gain [30], [31]. Thus, a sample in a two-dimensional DPCM can be predicted as

$$x^*(m,n) = a_1 x(m-1,n-1) + a_2 x(m-1,n) + a_3 x(m,n-1) + a_4 x(m-1,n+1) \quad (31)$$

Maximising the prediction gain requires the minimisation of the error variance. Minimising the error variance, in the special case of a separable correlation function, results in the following relations [31]:

$$a_1 = -\rho_v \rho_h, \qquad a_2 = \rho_v, \quad a_3 = \rho_h, \quad a_4 = 0 \quad (32)$$

where ρ_v and ρ_h are vertical and horizontal correlation coefficients given by

$$\rho_v = \frac{R_{xx}(1,0)}{\sigma^2}, \qquad \rho_h = \frac{R_{xx}(0,1)}{\sigma^2} \quad (33)$$

A separable model for covariance function is defined as

$$R_{xx}(m,n) = \sigma^2 \rho_v^{-|m|} \rho_h^{-|n|} \quad (34)$$

Drawbacks of DPCM

There are three types of degradation are common in a DPCM quantiser design: *granularity*, *slope overload* and *edge-busyness* [35]. Granularity is generated as a result of the step-like nature of the output where the input signal is almost constant. Slope overload happens when

there is a sharp change in the input signal (edges). In this case the quantised output cannot follow the input and a few steps are needed to match the output with the input. Edge-busyness is caused at less sharp edges when the input in the adjacent lines is quantised into different levels. Another drawback of the DPCM is its sensitivity to channel noise and image statistics. Adaptive techniques have been used to compensate these drawbacks.

3.2 Transform Coding

A common approach to image compression is the use of transformations that transform the pixels in the image domain into another domain to produce a set of coefficients, where the representation is more natural and therefore more compact. This representation allows some coefficients to give the bulk of the energy in the image while others are very likely to be very small or zero.

Transform coding is a general scheme for lossy image compression. It uses a reversible, linear transform (such as the Fourier transform) to map the image into a set of coefficients which are then quantised and coded. A good transformation packs as much information as possible into a small number of transform coefficients. Then quantisation selectively eliminates the coefficients that carry the least information. In transform coding approach, an $N \times N$ input image first is divided into a number of $n \times n$ nonoverlapping subimages (blocks), which are then transformed to generate $(N / n)^2$ subimage transform arrays, each of size $n \times n$, and the transform is applied separately to each of these blocks**.**

For a typical image, the correlation between the adjacent pixels is high. Transform coding uses this correlation in order to achieve a high compression ratio. To illustrate this correlation, two consecutive pixels of image "*Lena*" are grouped as a vector (x, y), and the dependency of y on x is presented in Figure 15. It can be shown that most of these points are concentrated near bisector $y = x$ in the dense area.

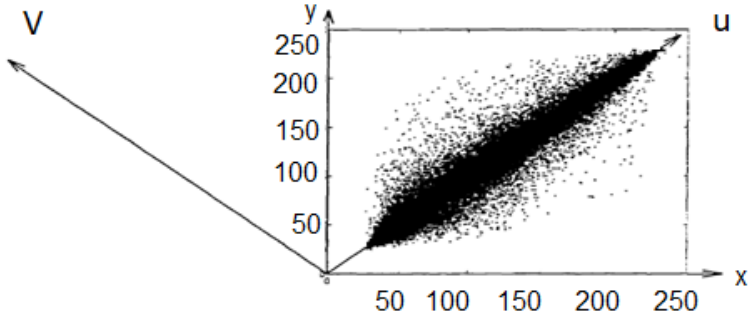
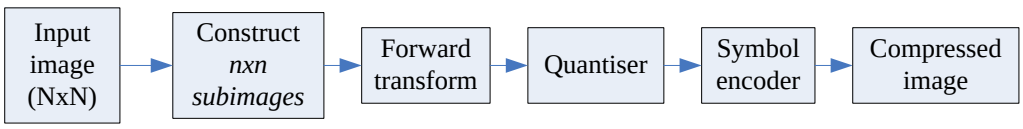


Figure 15: The correlation of adjacent pixels for the image Lena.

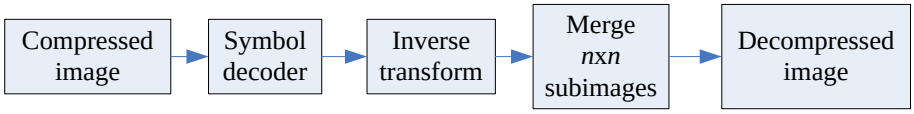
Quantising any two consecutive samples independently results in inefficiency since the quantisation levels for both dimensions are the same. For example, the quantiser allocates the same bit rate to the upper-left as the dense area. However, the probability of a vector being in this area is very low. To improve the quantiser efficiency, after grouping the samples, the coordinate system can be rotated by a certain angle such that one of the axes is placed in the middle of the dense area. In this case, more bits can be allocated to the *u*-axis and less to the *v*-axis (refer to Figure 15). Hence, with the same average bit rate: better precision is achieved. After quantisation and encoding, the inverse of this rotation is carried out in the decoder.

Transform coding achieves high compression due to three mechanisms. First, transform coding is a block technique where a block of data is processed rather than a single element of the image. Second, the quantisation of the transformed coefficients results in removing the correlation defined among the pixels of each subimage. Finally, not all transformed coefficients are quantised and transmitted to the receiver; as a result, high compression rates can be achieved. The implementation of the transform coding system is more complicated than the implementation of the predictive coding system and hence it is not preferable for applications that require less cost and complexity [26].

Figure 16 shows a typical transform coding system. It consists of two parts, the encoder and the decoder. The encoder performs four relatively straightforward operations: subimage decomposition, transformation, quantisation, and coding. Using a suitable form of transformation [37], the values of the grey levels in each block are transformed. Large values which significantly influence the total energy of the system will be quantised, while all other values are set to zero. The decoder implements the inverse sequence of steps (with the exception of the quantisation function) of the encoder.



(a) Encoder



(b) Decoder

Figure 16: A transform coding system. (a) Encoder. (b) Decoder.

Discrete Linear Orthonormal Transforms

The use of orthogonal transforms on digital signals has become popular.

Linear image transformations view a grayscale image as a vector whose components are the pixel intensities of the image. The entire image can be taken as a single vector or it can be divided into blocks as in the transform coding or vector quantisation scheme, and view the blocks as vectors. Let n be the number of pixels in the image (or in the image blocks) so the images (or blocks) are elements of the real vector space R^n. A linear transformation is a linear function $f : R^n \longrightarrow R^n$ of the image space into itself.

The transformation can be represented as a (possibly complex) matrix $\boldsymbol{T}$ operating on column vectors $\boldsymbol{b}$ and $\boldsymbol{c}$:

$$\boldsymbol{c} = \boldsymbol{T}\,\boldsymbol{b} \tag{35}$$

$$\boldsymbol{b} = \boldsymbol{T}^{-1}\boldsymbol{c} \tag{36}$$

where $\boldsymbol{T}$ is an $n \times n$ transform matrix, $\boldsymbol{b}$ represents an image block of size $n \times 1$ and $\boldsymbol{c}$ is the column vector of the transformed coefficients. $\boldsymbol{T}$ is called a *linear transform*.
The transform matrix $\boldsymbol{T}$ is called an *orthonormal* or *unitary transform* if Equation (37) is satisfied [26]:

$$\boldsymbol{T}^{-1} = \boldsymbol{T}^{/} \tag{37}$$

where $\boldsymbol{T}^{/}$ denotes conjugate transpose of the matrix $\boldsymbol{T}$. If the j^{th} column of the transform matrix $\boldsymbol{T}^{/}$ is denoted by $\boldsymbol{t}_j$, then equation (37) is equivalent to

$$t_j^{/} t_i = \delta_{ji} \tag{38}$$

where δ_{ji} is the kronecher delta and is defined as

$$\delta_{ji} = \begin{cases} 1 & if \quad i = j \\ 0 & if \quad i \neq j \end{cases} \tag{39}$$

The vectors $\boldsymbol{t}_j$ are called *orthonormal basis vectors* for the linear unitary transform $\boldsymbol{T}$, and Equation (39) can be written as

$$\boldsymbol{b} = \sum_{j=1}^{N} c_j \boldsymbol{t}_j \tag{40}$$

where c_j is the j^{th} element of $\boldsymbol{c}$, given by

$$c_j = \boldsymbol{t}_j^{/} \boldsymbol{b} \tag{41}$$

In practical applications of transform coding, determining n transformed coefficients requires $n \times n$ multiplications and $n \times (n - 1)$ additions. Therefore, transform coding is a computationally demanding algorithm.

The application of transform coding to images is carried out in a two-dimensional fashion. The image is divided into blocks of 8×8 or 16×16, and the transformation is performed in the horizontal and the vertical directions. The first step of the transform operation is:

$$\boldsymbol{C}_1 = \boldsymbol{T}\,\boldsymbol{B} \tag{42}$$

where $\boldsymbol{T}$ is the transform matrix, and $\boldsymbol{B}$ is the data block.

Transforming the rows of $\boldsymbol{C}_1$ is carried out as follows:

$$\boldsymbol{C} = \boldsymbol{C}_1 \boldsymbol{T}^{\mathrm{T}} = \boldsymbol{T}\,\boldsymbol{B}\,\boldsymbol{T}^{\mathrm{T}} \tag{43}$$

In this case, $\boldsymbol{C}$ contains the final two-dimensional transformed coefficients of the data block $\boldsymbol{B}$. The inverse transformation can be used to reconstruct the original values of the data block matrix. Since we assume an orthonormal transformation, then the inverse of the transform matrix is the same as its transpose matrix.

Basic Transforms for Image Coding

There exist several different transformations that are used in image compression such as Karhunen-Loeve Transform (KLT) [38], Discrete Fourier Transform (DFT) [39], [40]; Discrete Walsh Hadamard Transform (DWHT) [41], [42]; and Discrete Cosine Transform (DCT) [43]. Most of them have been developed in the last 60s and the early 70s. They differ in their energy compaction capability and computational complexity. Other transformations have been developed [44], however we will concentrate on the basic transformations.

Karhunen-Loeve Transform (KLT)

KLT [45] is an orthonormal linear transformation which is a preferred method for approximating a set of vectors or images by a low dimensional subspace. It is the optimum transform in the energy compaction sense; it has the best "input-decorrelating" and "variance-ordering". This means that the method provides the optimal subspace, spanned by the KLT basis, which minimises the *MSE* between the given set of vectors and their projections on the subspace. It creates an autocorrelation matrix from the original signal. From this autocorrelation matrix the orthogonal eigenvectors are found to form the linear basis of the original data signal [46]**.** With these basis vectors, it is possible to represent the original data signal as a linear combination of these basis vectors. The resultant transformed signal has all its linear correlation removed. This transform depends heavily on the statistics of the input samples.

Discrete Fourier Transform (DFT)

In the late 60s, the application of the DFT to image compression was introduced [47], [39]. This is one of the complex transformations used in image coding [48]. The Fourier transform decomposes a complex signal into a weighted sum of a zero frequency term (the DC term which is related to the average value). The equation for the one-dimensional discrete Fourier transform of a sequence $\{f(x), x = 0, 1, \ldots, N - 1\}$ is defined by [44]:

$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x) e^{-j2\pi ux/N}, \quad u = 0,1,\ldots, N-1 \tag{44}$$

The inverse for the one-dimensional DFT is given by:

$$f(x) = \sum_{u=0}^{N-1} F(u) e^{j2\pi ux/N}, \quad x = 0,1,..., N-1 \tag{45}$$

Thus, the transform matrix of the one-dimensional DFT is given by

$$F = \left\{ \frac{1}{N} e^{-j2\pi ux} \right\}_{u,x=0,1,...,N-1} \tag{46}$$

The disadvantages of the DFT is that the difficulty of using this transformation due to the requirement to process both real and imaginary components as shown in Equation 46.

For image application, an $N \times N$ image can be decomposed into a weighted sum of two-dimensional sinusoidal term. The two-dimensional discrete Fourier transform equation is given by:

$$F(u,v) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) e^{\frac{-j2\pi(ux+vy)}{N}}, \quad u,v = 0,1,..., N-1 \tag{47}$$

The inverse of the two-dimensional DFT is given by:

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) e^{\frac{j2\pi(ux+vy)}{N}}, \quad x,y = 0,1,..., N-1 \tag{48}$$

The transform matrix of the two-dimensional DFT is given by

$$F = \left\{ \frac{1}{N^2} e^{\frac{-j2\pi(ux+vy)}{N}} \right\}_{u,v,x,y=0,1,...,N-1} \tag{49}$$

The application of two-dimensional DFT to image arrays produces a two-dimensional spectrum of the data where highly correlated image data have small energy at high spatial frequencies [48]. One important property of the DFT is its *separability* in which the two-dimensional basis image can be decomposed into two product terms. If the basis images are separable, the result can be found by successive application of two, one-dimensional transforms. This can be done by first separating the basis image term into product terms as follows:

$$e^{\frac{-j2\pi(ux+vy)}{N}} = e^{\frac{-j2\pi ux}{N}} e^{\frac{-j2\pi vy}{N}} \tag{50}$$

Equation 50 can be written as

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1} e^{\frac{-j2\pi ux}{N}} \sum_{y=0}^{N-1} f(x,y) e^{\frac{-j2\pi vy}{N}}, \quad u,v = 0,1,...,N-1 \tag{51}$$

One of main reasons that the DFT has become an important tool in signal processing is that it can be implemented using Fast Fourier Transforms (FFT) [31]. With these methods the complexity of the transform operation is reduced from N^2 multiplication/addition operations to ($N \log_2 N$) operations [37]. The problem with the DFT is that it is not an optimal transformation, since it does not diagonalise the covariance matrix. In addition, inverse DFT generates samples which are periodic extension of the first N samples [49], that is,

$$g(x) = f(x), \quad x = 0,1,2,...,N-1 \tag{52}$$

and

$$g(x) \neq f(x), \quad x = N, N+1, N+2,... \tag{53}$$

$$g(x+N) = g(x) \tag{54}$$

This periodicity in DFT causes discontinuities at the beginning and end of each block. This effect can be shown in Figure 17.

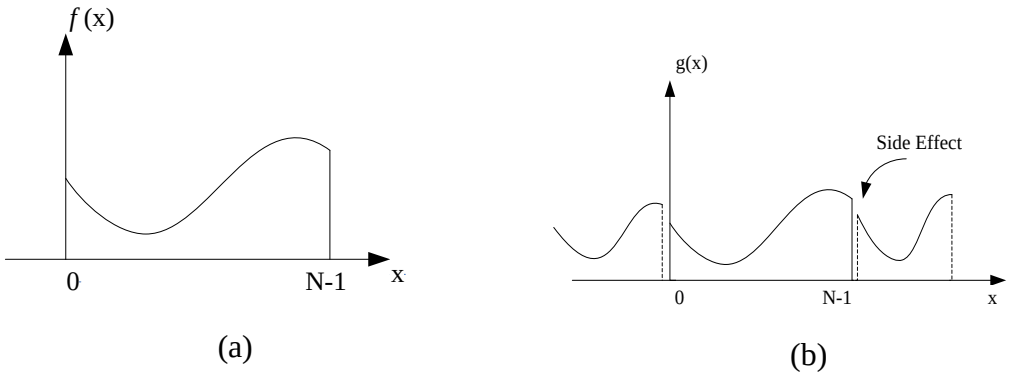


Figure 17: Side effect in DFT.

Discrete Cosine Transform (DCT)

DCT is one of the most efficient transform coding schemes. It was introduced by Ahmed [43] to produce good performance and solve the problem of discontinuity at the ends of the data blocks inherit with the DFT**.**

This transformation is an *orthogonal*, *separable*, and *real* transform which translates the image information from spatial domain to frequency domain to be represented in a more compact form. It is closely related to DFT.. DCT is the basis of many image compression methods. For example, the standard JPEG (Joint Photographers Expert Group) [54], [55], for which DCT is carried out in 8×8 image blocks existed as the main image compression standard.

The equation for the one-dimensional DCT of a sequence $\{f(x), x = 0, 1, \ldots, N-1\}$ that is zero outside the interval $0 \le x \le N\text{-}1$ is defined as:

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left(\frac{(2x+1)u\pi}{2N} \right), \qquad u = 0,1,\ldots,N-1 \tag{55}$$

The inverse transformation is given by

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos\left(\frac{(2x+1)u\pi}{2N} \right), \qquad x = 0,1,\ldots,N-1 \tag{56}$$

$$\alpha(0) = \frac{1}{\sqrt{2}} \quad \textit{and} \quad \alpha(u) = 1 \quad \textit{for all } u \neq 0 \tag{57}$$

One-dimensional DCT basis vectors can be obtained from

$$C_{u,x} = \left\{ \sqrt{\frac{2}{N}} \alpha(u) \cos \frac{(2x+1)u\pi}{2N}, \quad x = 0,1,2,\ldots,N-1 \right\}, \textit{ for } \quad u = 0,1,\ldots,N-1 \tag{58}$$

Since DCT is a separable transform, the two-dimensional DCT and its inverse can be obtained in two steps by successive applications of the one-dimensional DCT and its inverse [55]. For image processing applications, two-dimensional DCT of an $N \times N$ block of pixels, *f(x, y)* can be defined as [37]:

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right) \tag{59}$$

where, *f(x, y)* denotes a two dimensional sequence of *N* x *N* points, *C(u, v)* denotes $N \times N$ points DCT of the block *f(x, y)* and *x , y = 0,1, ...,N-1*; *u ,v = 0,1, ..., N-1*. The inverse DCT is defined as

$$f(x,y) = \alpha(u)\alpha(v)\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u,v)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N} \tag{60}$$

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & for \quad u = 0 \\ \sqrt{\frac{2}{N}} & for \quad u = 1, ..., N-1 \end{cases} \tag{61}$$

The transform matrix of the two-dimensional DCT is given by

$$C_{u,v,x,y} = \left\{ \frac{2}{N}\alpha(u)\alpha(v)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N}, \quad x,y,u,v = 0,1,2,...,N-1 \right\} \tag{62}$$

The coefficient of the DCT matrix with zero frequency (*C* (*0, 0*)) is called the "DC coefficient," while the remaining coefficients are called the "AC coefficients", which reflect variations in grey level values in certain direction at certain rate as shown in Figure 18.

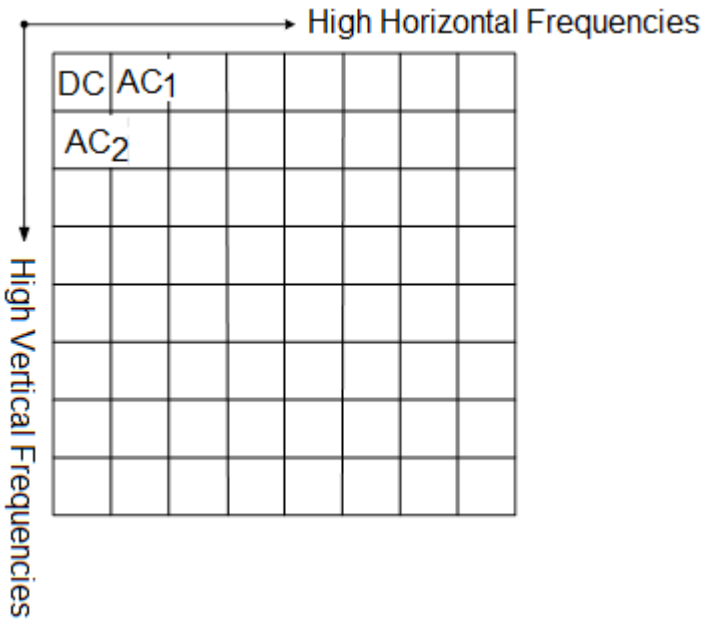


Figure 18: The DCT block structure.

Conveniently, in most image subblocks the energy is concentrated in the DC and low AC frequency bands. That is, the image information (energy) is usually concentrated in the low frequency region (the top left corner). The high frequency region is located in the bottom right corner. This allows for the suppression of the higher AC frequency bands thus allowing the block to be represented by fewer coefficients in its transformed state than when it existed in its original state. One simple observation is that each DCT coefficient *C (u, v)* is a linear combination of all pixel values within the block [50], [175].

DCT has decorrelation property close to that of the KLT for most images. The basis vectors used to represent the original signal or image block are signal or image *independent*. These linear basis functions can be seen in Figure 19 for an 8×8 image block [51].

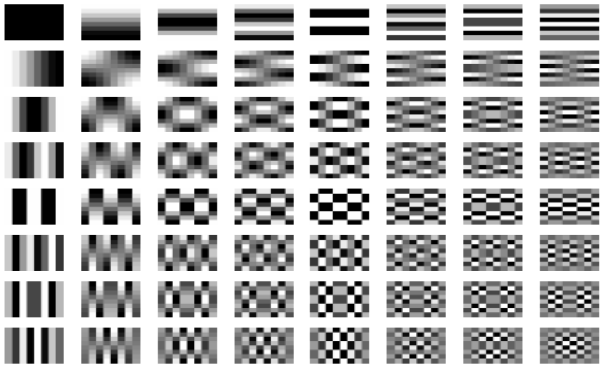


Figure 19: Basis functions of an 8x8 DCT [51].

DCT has an excellent energy compaction property for highly correlated data [52], [53], [173], [174]. These properties make the DCT a popular transform for image coding. However, a disadvantage of DCT is the blocking (or tiling) artifacts that appear in high compression ratios [111]. The same artifacts may appear in other transforms as well, because typically the transforms are implemented on pixel blocks, rather than the whole image. 8×8 DCT coefficients are scanned in a zigzag order as shown in Figure 20, starting from the lowest to highest frequency.

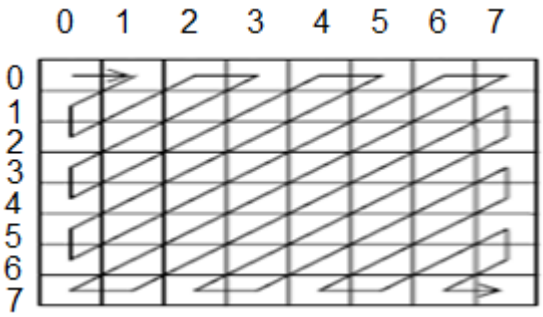


Figure 20: Zigzag scanning of the transformed coefficients [51].

Walsh-Hadamard Transform (WHT)

WHT differs from Fourier and cosine transforms in that the basis functions are not sinusoid. The basis functions are based on square or rectangular waves with peaks ± 1, as shown in Figure 21.

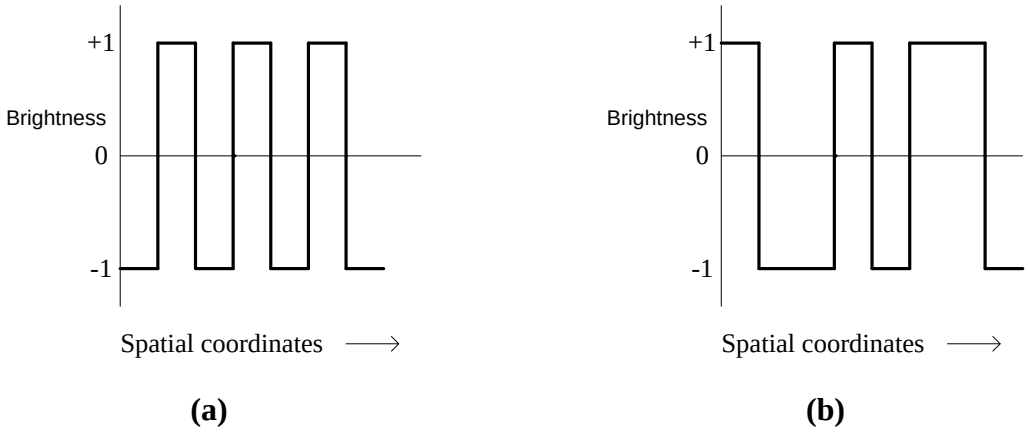


Figure 21: Form of the Walsh-Hadamard basis functions. (a) A square wave. (b) Representation of a rectangular wave. The width of each pulse may vary [8].

The transform is one of the simplest transformations where only addition and subtraction operations are required. The lowest order of the Walsh-Hadamard transform is:

$$\boldsymbol{T}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad (63)$$

By replacing the elements of the transform matrix by the first order matrix itself, higher order Walsh-Hadamard transform matrices can be obtained:

$$\boldsymbol{T}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & -\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \qquad (60)$$

The general expression for the WHT can be defined as follows:

$$\boldsymbol{T} = \frac{1}{\sqrt{N}} H_{2N} \qquad (61)$$

where

$$\boldsymbol{H}_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix} \qquad (62)$$

And $H_1 = 1$ and $N = 2^n$.

The determination of the inverse matrix is further simplified for the WHT because of the symmetry of the transform matrix.

3.4 JPEG Standard

The Joint Photographic Expert Group (JPEG) met initially in 1987 under the auspices of the International Standards Organisation (ISO) to design an optimal still image compression standard. JPEG became a Draft International Standard in 1991 and an International Standard in 1992 [54], [56] and [197]. JPEG is not just one image compression algorithm. There are several modes defined for JPEG [55], including *baseline*, *lossless*, *progressive* and *hierarchical*.

The baseline mode is the most popular one and supports lossy coding only, and can be extended by several options depending on the requirements of the applications. It uses DCT,

Huffman coding and sequential transmission. The extended options include arithmetic coding instead of Huffman coding; and progressive sequential or hierarchical modes. All these options are built on top of the baseline algorithm. There is an independent lossless mode that is not based on DCT but uses *predictive coding* instead [37].

In the baseline mode, the image is divided into 8×8 blocks. DCT is computed over these blocks. DCT coefficients are scaled and truncated in order to reduce the dynamic range of the data. The transformed blocks are quantised with a uniform scalar quantiser, zigzag scanned in which the scaled DCT coefficients are ordered, and entropy coded with Huffman coding. The quantisation step size for each of the 64 DCT coefficients is specified in a quantisation table, which remains the same for all blocks. The DC coefficients of all blocks are coded separately, using a predictive scheme. It refers to this mode simply as JPEG.

The lossless mode is based on a completely different algorithm, which uses a predictive scheme. The prediction is based on the nearest three causal neighbours and seven different predictors are defined (the same one is used for all samples). The prediction error is entropy coded with Huffman coding.

The progressive and hierarchical modes of JPEG are both lossy and differ only in the way the DCT coefficients are coded or computed, when compared to the baseline mode [119]. They allow a reconstruction of a lower quality or lower resolution version of the image, by partially decoding the compressed bitstream. Progressive mode encodes the quantised coefficients by a mixture of spectral selection and successive approximation, while hierarchical mode uses a pyramidal approach to compute the DCT coefficients in a multi-resolution way. In what follows the full details about JPEG image coding will be provided.

Like any transform coding scheme, JPEG is a two–step transform coding algorithm: the first step is lossy and involves a DCT transformation followed by quantisation. This part is used to remove information that is perceptively irrelevant for a human user. The second step involves lossless entropy encoding to eliminate statistical redundancies that could still be present in the transformed representation. Each block is converted into 64 DCT coefficients whose values are uniquely determined by the 64 input pixels.

The DCT coefficients are quantised by a set of uniform scalar quantisers defined in a quantisation table. Different transform coefficients may use quantisers with various step sizes. The step sizes are specified in a quantisation table. The idea of allowing various

coefficients to be quantised differently is to take advantage of the high sensitivity of the human visual system to low frequencies. Lower frequency coefficients are quantised less than higher frequency coefficients. A quantisation table is an 8 × 8 array of quantisation step sizes, with entries from 1 to 255. JPEG does not fix the quantisation table, but it allows the user to specify the table to be used. The table is then included as part of the compressed data. Up to four different quantisation tables may be specified in the baseline JPEG [33]. Tables 2 and 3 show the quantisation tables defined by JPEG for luminance (brightness) and chrominance (colour), respectively [8].

Table 2: Luminance quantisation table.

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

Table 3: Chrominance quantisation table.

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

The quantised DCT coefficients are fed to the entropy coder, which is either a Huffman or an arithmetic coder. When passing the quantised coefficients to the entropy coder, two coding modes can be utilised which are sequential coding (JPEG sequential mode) and progressive coding (JPEG progressive mode). The outputs of the quantisation component are called image descriptors.

3.5 JPEG-2000 Standard

JPEG-2000 [57], [58], [61] is the image compression standard for still image coding developed jointly by the ISO/ITU-T (International Organisation for Standardisation / International Telecommunications Union), to complement the JPEG standard [55] by providing improved compression performance and new functionalities [56]. JPEG-2000 Part I, the core coding algorithm, became an international standard in December 2000 [59] and provides, in a single bit-stream, a wide array of functionalities, such as: progressive transmission by resolution, quality, component (normally: a single colour plane of an image), or location; random access; lossless to lossy compression; and error tolerance. JPEG-2000

algorithm is based upon embedded block coding with optimised truncation [60]. It is based on discrete wavelet transform (DWT) [138], [139], scalar quantisation, context modelling, arithmetic coding and post-compression rate allocation.

JPEG-2000 has been designed to offer compression performance as good as, or better than, conventional JPEG. Reported results comparing JPEG-2000 to conventional JPEG [61] indicate approximately a 2dB improvement in image quality (measured with *PSNR*) at the same bit rate, or alternatively, a 20-30% improvement in compression for the same quality. The superiority of JPEG-2000 is particularly significant at low bit rates (say, compression ratios > 10:1).

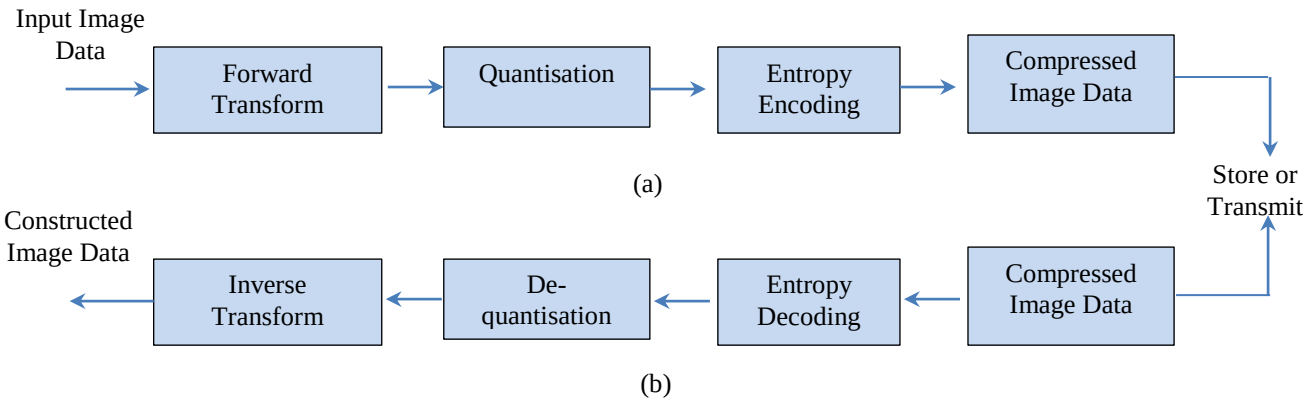


Figure 22: JPEG 2000 Block Diagram. (a) Encoder. (b) Decoder [GON, 04].

Figure 22 shows the block diagram of JPEG-2000 image compression technique. This is similar to every other transform based coding scheme. The transform is first applied on the source image data. The transform coefficients are then quantised and entropy coded, before forming the output. The decoder is just the reverse of the encoder. Unlike other coding schemes, JPEG-2000 can be both lossy and lossless. This depends on the wavelet transform and the quantisation applied

As we can see from Figure 22, the principal difference between the wavelet-based JPEG-2000 system and the DCT-based JPEG system is the omission of the latter subimage processing stages. Because wavelet transforms are both computationally efficient and inherently local (i.e., their basis functions are limited in duration), subdivision of the image into blocks is unnecessary. The removal of subdivision step eliminates the blocking artefact that characterises DCT-based approximations at high compression ratios [4].

JPEG-2000 standard works on image tiles [56]. All operations, including component mixing, wavelet transform, quantisation, and entropy coding, are performed independently for the various tile. The nominal tile dimensions are powers of two, except for those on the boundaries of the image. Tiling is done to reduce memory requirements, and since each tile is reconstructed independently, they can be used to decode specific parts of the image, rather than the whole image. Each tile can be considered as an array of integers in sign-magnitude representation. This array is then described in a number of bit planes. These bit planes are a sequence of binary arrays with one bit from each coefficient of the integer array. The first bit plane contains the most significant bits of all the magnitudes. The second array contains the next most significant bits of all the magnitudes, and so on until the final array, which consists of the least significant bits of all the magnitudes [62], [172].

Before the forward DWT is applied to each tile, all image tiles are DC level shifted by subtracting the same quantity, such as the component depth, from each sample [56].

This process is pictured in Figure 23. These subbands contain coefficients that describe the horizontal and vertical characteristics of the original tile component. All of the wavelets transforms employing JPEG-2000 compression method are fundamentally one-dimensional in nature [63], [136]. Applying one-dimensional transforms in the horizontal and vertical directions forms two-dimensional transform. This results in four smaller image blocks; one with low resolution, one with high vertical resolution and low horizontal resolution, one with low vertical resolution and high horizontal resolution, and one with all high resolution.

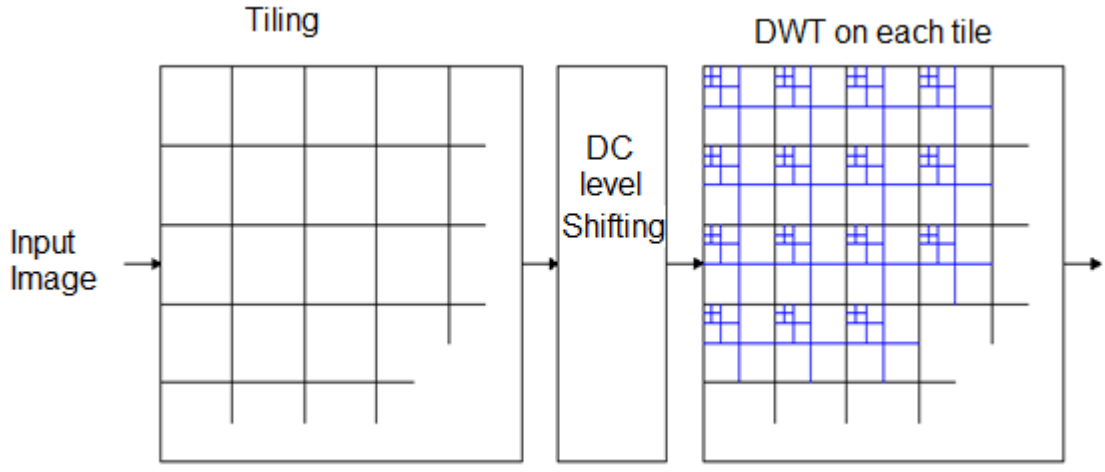


Figure 23: Tiling, DC level shifting, and DWT on each tile [9].

This process of applying the one-dimensional filters in both directions is then repeated a

number of times on the low-resolution image block. This procedure is called dyadic decomposition and is pictured in Figure 24. An example of dyadic decomposition [56], [112] into subbands with the whole image treated as one tile is shown in Figure 25.

One significant feature of JPEG-2000 is the possibility of defining regions of interest (ROI) in an image [65], [138]. These regions of interest are coded with better quality than the rest of the image. This is done by scaling up, or DC shifting, the coefficients so that the bits associated with the regions of interest are placed in higher bit-planes. During the embedded coding process, these bits are then placed first in the bit-stream. Hence, the region of interest will be decoded before the rest of the image.

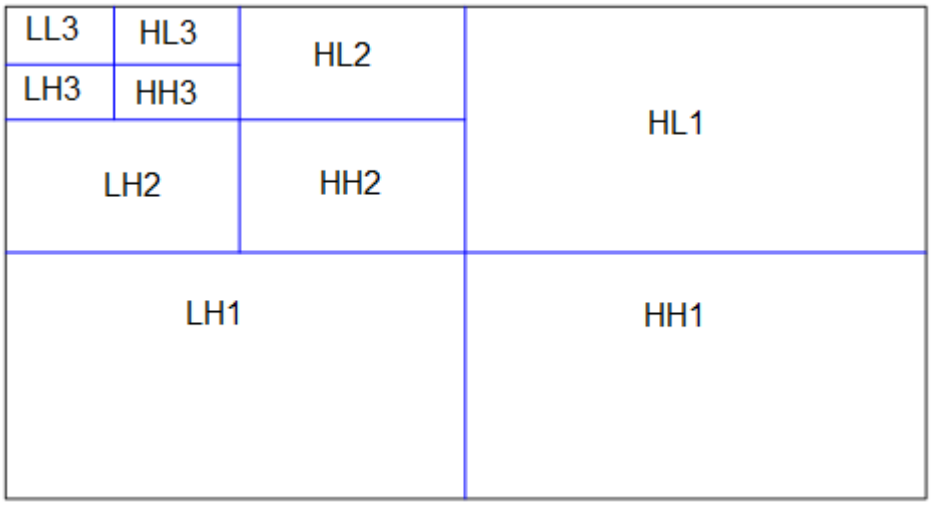


Figure 24: The dyadic decomposition [9].

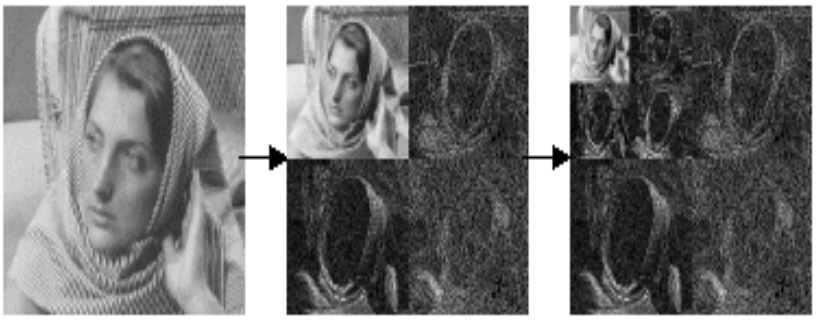


Figure 25: Example of dyadic decomposition [56].

3.5 Singular Value Decomposition

Singular Value Decomposition (SVD) is a highlight of linear algebra. It plays an interesting

fundamental role in many different applications such as digital image processing, dimensionality reduction and image compression [139], [66], [67], [140]. The use of SVD in image compression is motivated by its excellent energy compaction in the least square sense [31], [68], [141]. Consequently, the use of SVD technique in image compression has been widely studied [69-75], [142]. The main disadvantage of the SVD transformation is the need for recalculation for each sub image. To exploit the optimal energy compaction properties of the SVD, most of the effort in designing a more efficient SVD coder is provided into the effective coding of singular values and singular vectors in order to reduce computational cost [76].

SVD is an efficient method to diagonalise a rectangular $m \times n$ matrix A by factorizing it into three matrices U, S, and V, such that:

$$A = USV^T \tag{64}$$

where S is a diagonal $m \times n$ matrix (the same dimensions as A) with elements s_i along the diagonal and *zeros* everywhere else. U and V are orthonormal matrices with sizes $m \times m$ and $n \times n$, respectively.

The matrix U is called the left singular matrix (the columns u_i of U are called the left singular vectors), V is called the right singular matrix (the columns v_i of V are called the right singular vectors), and the diagonal matrix S is the singular value matrix (the diagonal elements s_i of S are called the singular values). The singular vectors form orthonormal bases and lead to the following relationship:

$$Av_i = s_i u_i \tag{65}$$

SVD is an approximation technique reduces any matrix into a smaller invertible and square matrix. Thus, one special feature of SVD is that it can be performed on any real $m \times n$ matrix.

Calculating the SVD consists of finding the eigenvalues and eigenvectors of AA^T and A^TA. The eigenvectors of A^TA make up the columns of V; the eigenvectors of AA^T make up the columns of U. The eigenvalues of A^TA or AA^T are the squares of the singular values for A. The singular values are the diagonal entries of the S matrix and are typically arranged in descending order. Singular values are always real numbers. If the matrix A is a real matrix, then U and V are also real. Equation 65 can be expressed as:

$$A = \sum_{i=1}^{p} u_i s_i v_i^T \tag{66}$$

where u_i and v_i are the i^{th} column vectors of *U* and *V* respectively, s_i are the singular values, and $p = \min\{m, n\}$. If the singular values are ordered so that $s_1 \geq s_2 \ldots \geq s_p$, and if the matrix *A* has a rank $r < p$, then the last $p - r$ singular values are equal to *zero*, and the matrix *A* can be approximated by a matrix A^* with rank *r* (i.e. the SVD becomes A^*) as follows:

$$A^* = \sum_{i=1}^{r} u_i s_i v_i^T \tag{67}$$

Hence, the approximation error matrix E_r is dependent on the performance accuracy of the quantisation and/or truncation by parameter *r*, which can be described as $E_r = A - A^*$.

The 2-*norm* of a matrix may be calculated from the singular values. Therefore the 2-*norm* of the error matrix ($A - A^*$) is equal to the next singular value not used in A^*, that is, the 2-*norm* of approximation error is calculated by:

$$E_r^2 = \left\| A - A^* \right\|_2 = \left\| \sum_{i=1}^{p} u_i s_i v_i^T - \sum_{i=1}^{r} u_i s_i v_i^T \right\|_2 = \left\| \sum_{i=r+1}^{p} u_i s_i v_i^T \right\|_2 = \sum_{i=r+1}^{p} (s_i)^2 \tag{68}$$

As the singular values are in descending order, it can be shown that the error decreases towards *zero* in the 2-*norm* sense.

The property of SVD to provide the closest rank *r* approximation for a matrix *A* as shown in Equation 68 can be used in image processing for compression and noise reduction. By setting the small singular values to *zero*, matrix approximations whose rank equals the number of remaining singular values can be obtained [77].

SVD generally relies on “global” information derived from all the vectors in the dataset, which is more effective for datasets consisting of homogeneously distributed vectors. For databases with heterogeneous distributed vectors, more efficient representation can be generated by subdividing the vectors into various groups characterised by a different set of statistical parameters.

3.6 Vector Quantisation

Vector quantisation (VQ) [9], [78], [115], [133] is an attractive technique for lossy data compression, which has been a key technology for data storage and/or transfer. It is a generalisation of scalar quantisation technique where the numbers of possible pixel values are reduced. The input data consists of *k*-dimensional vectors (*k*-pixel blocks) instead of scalars. VQ is superior to predictive and transform coding because it achieves optimal rate distortion performance subject to constraints on the memory or block length of the observable signal segment being coded [79], [131], [142].

VQ is the process of optimal approximation from an input space to an output space that, in the case of image compression, is itself a subset of the input space.

The input space is not evenly occupied by these vectors [118]. Because of the high correlation between the neighbouring pixel values, some input vectors are very common while others hardly ever appear in real images. For example, completely random patterns of pixels are rarely seen but certain structures (like edges, flat areas, and slopes) are found in almost every image [110], [126], [128], [135], [141]. The efficiency of VQ comes from its role as a pattern matching technique. The vector of samples is a pattern that must be approximated by one of the finite set of prototype patterns. To describe this pattern, one can simply identify the address of the pattern in the dictionary of standard patterns that best approximates it. This dictionary of patterns is called the *codebook*; the patterns in the codebook are called *codewords* or *codevectors*.

VQ is a clustering method, grouping similar vectors (blocks) into one class. It maps features extracted from the sampled input image using pre-processing operations. In the most direct application of VQ to image compression, the source samples (pixels) are blocked into vectors so that each vector describes a small segment, or subblock, of the original image. The simplest way to encode the image with VQ is to independently quantise each input vector. Encoding each vector does not depend on previous encoded vectors ; hence, this technique is called *memoryless* VQ. Spatial correlation between pixels in an individual vector is efficiently exploited. When VQ maps input vectors into a set of codevectors, similar vectors are mapped to the same codevectors in the codebook.

Vector quantiser Q of dimension k and size N is a mapping of a vector (or a “point”) in k-dimensional Euclidean space, R^k, to a finite subset Y of R^k containing N reproduction points, called *codevectors* or *codewords* [9]:

$$Q : R^k \rightarrow Y \tag{69}$$

The finite set $Y = \{\mathbf{y}_{i:}\ i = 1\ldots N\}$, where N is the size of the set Y, is called VQ codebook, and $\mathbf{y}_i$ represents the i^{th} codevector (codeword) in the codebook Y. Associated with each reproduction codevector is a partition of R^k, called a region or cell, $S = \{S_i\ ;\ i = 1, 2, \ldots, N\}$ [98]. The most popular form of vector quantiser is the *Voronoi* or *nearest neighbour vector quantiser* [9], where for each input source vector, $\boldsymbol{x}$, a search is performed to the entire codebook to find the nearest codevector, $\boldsymbol{y}_i$, which has the minimum distance [9].

$$Q(\boldsymbol{x}) = \boldsymbol{y_i} \text{ if } d(\boldsymbol{x}, \boldsymbol{y}_i) < d(\boldsymbol{x}, \boldsymbol{y}_j) \text{ for all } i \neq j \tag{70}$$

where $d(x, y)$ is a distance measure between the vectors, $\boldsymbol{x}$ and $\boldsymbol{y}$. The cells S_i can be described as:

$$S_i = \{\boldsymbol{x} \in R^k : Q(\boldsymbol{x}) = \boldsymbol{y_i}\} \tag{71}$$

with this definition, it follows that

$$\bigcup_{i=1}^{N} S_i = R^k \quad \textit{and} \quad S_i \bigcap S_j = 0; \text{ for all } i \neq j \tag{72}$$

VQ partitions the input space into N nonoverlapping regions so that the input space is completely covered. The regions in a nearest neighbour's vector quantiser are also called *Dirichlet* or *Voronoi regions* [89].

For a codebook of size N, the number of bits needed to uniquely address an individual codevector of dimension n^2 (block of size $n \times n$ pixel), is $\boldsymbol{r} = \mathbf{log_2}\, \boldsymbol{N}$, and the average number of bits needed to describe each pixel is R [9] defined as:

$$R = \frac{\log_2 N}{n^2} \text{ (bit per pixel "bpp")} \tag{73}$$

where R is the bitrate in bits per pixel (bpp) of the vector quantiser.

Vector Quantisation in Image Compression

Vector quantisers are the best quantisers to achieve minimum distortion for a given bitrate and vector dimension. The goal of a vector quantiser based system is to reduce the bitrate and minimise communication channel capacity or digital storage memory requirements while

maintaining the necessary fidelity of the data [78]. Vector quantisation provides many attractive features for image coding applications with high compression ratios [9].

One important feature of vector quantisation is the possibility of achieving high compression ratios with relatively small block sizes [114]. Another important advantage of VQ image compression is its fast decompression using small size lookup table. The decompression process requires the use of the codebook to recreate the image. This type of compression is useful for applications where the images are compressed once and decompressed many times, such as the compression of images displayed on the websites [8].

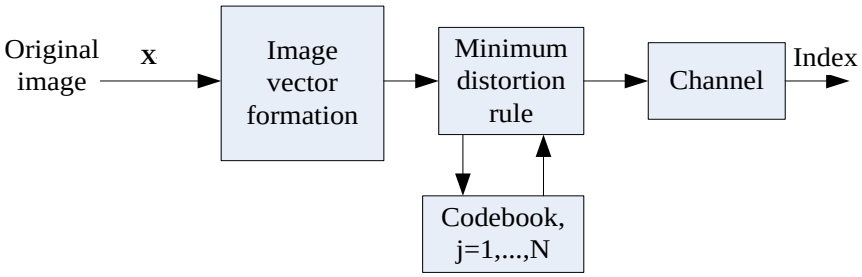


Figure 26: A block diagram of a VQ encoder.

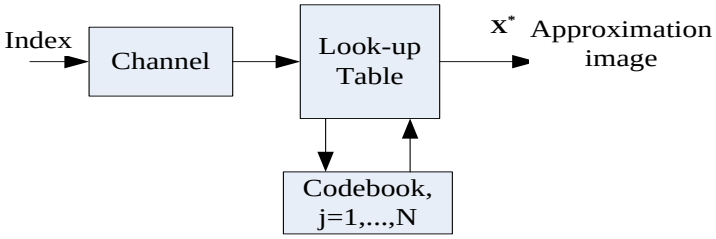


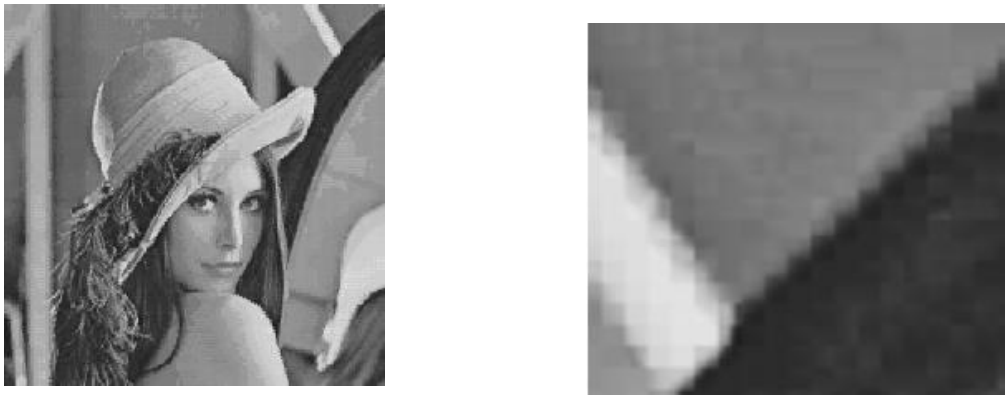
Figure 27: A block diagram of a VQ decoder.

At the decoder, the received index will be used to select the corresponding codevector from the same codebook utilised at the encoder as the reconstructed version of the input vector. Therefore, the codebook is extremely important in reconstructing good quality images at the receiver. The LBG algorithm, which was designed by Linde Buzo and Gray and hence the name, has been used extensively for the reconstruction of the codebook in VQ image coding [80], [150]. Blocks are typically 4×4 pixels or 8×8 pixels, which are transformed into 1-dimentional vectors of 16-dimentioanl or 64-dimensional.

Statistical studies on natural images have shown that there is a little correlation between pixels more than 8 positions apart [51], [81], [151], and as result, most of the correlations are among pixels that are within 4 positions away. Therefore, 4×4 and 8×8 codevectors are

excellent choices from both the bitrate and the correlation-exploitation standpoints. Figures 26 and 27 depict the block diagrams of VQ encoder and decoder, respectively.

Figure 28 shows the image “Lena” quantised at bitrate of 0.5625 bits/pixel and each vector is formed from a block of 4×4 pixels. The codebook trained using various images including the “Lena” image.



(a) (b)

Figure 28: The image “Lena” (512×512) coded using a vector quantiser at 0.52 bpp, and 4×4 blocks: (a) Entire coded image, PSNR = 28.7617; (b) Close-up of an edge region, showing the “staircase effect”.

Practical Limitations of Basic Vector Quantisation

Despite the optimality, basic unconstrained vector quantisers for image coding are limited to small dimension and low bitrates due to their computational complexity and memory requirements growing exponentially. Furthermore, the training of a high bitrate vector quantiser requires a large training set. Mismatch between the trained codebook and an image to be quantised, which is not part of the training set, also leads to degraded quality. Therefore, the training set should be large and representative of the images to be quantised [82], [152].
Various approaches have been proposed in the literature to mitigate the abovementioned limitations of the basic VQ scheme which are concentrated in two directions: to reduce the computational complexity and to generate a better codebook that approaches the global optimal solution. Many methods designed to reduce the computational time, by merging nearest training vector clusters until the desired number of codevectors is obtained or by

reducing the dimension measure in the LBG algorithm [83], [84], [120], [153], [154], [155] and [85]. However, codebooks generated by these methods are slightly degraded even though the computation time is reduced by several times. On the other hand, several methods are proposed to generate better codebooks [86], [156], [87] [30], but these methods need a great deal of time in order to obtain better codebooks.

3.7 Hybrid Coding

Habibi [30] introduced hybrid transform/DPCM coding. The proposed hybrid coding system enjoys the benefits of both transform coding and DPCM and is less vulnerable to their limitations.

In what follows, we will describe one-dimensional hybrid coding. The $N \times N$ image is divided into vertical strips of width M and the one-dimensional transformation is performed as follows:

$$c_i(y) = \sum_{x=1}^{M} b(x,y)t_i(x) \qquad (74)$$

$$\text{for } i = 1,2\ldots.M \text{ and } y = 1,2,\ldots..N$$

where t are the elements of the transformation matrix and b(x,y) is the pixel value at location (x, y). The aim of this transformation is to remove the redundancy between pixels situated in the same row of a strip. Therefore, this is a transformation in the horizontal direction [88]. The vertical redundancy can be removed using DPCM to code the transformed coefficients. A parallel linear DPCM predictor is available at the encoder and used to code the set of coefficients at each transform coordinate column wise [44]. The error signal is defined as:

$$e_i^k = c_i^k - \hat{c}_i^k \qquad (75)$$

where k represents the index of the strip, c_i^k is the transformed value and $\hat{c}_i^k$ is its corresponding predicted value. This error signal is quantised using Lloyd-Max quantiser and transferred to the communication channel. At the receiver, the predicted coefficient signal is added to the quantised error signal to retrieve the original value of the transformed coefficients plus some quantisation error:

$$\tilde{c}_i^k = \hat{c}_i^k + \tilde{e}_i^k, \qquad (76)$$

where $\tilde{e}_i^k$ is the quantised error signal. To obtain the reconstructed value of the image, inverse transformation is performed as follows:

$$\tilde{b}_i^k = T^{-1}\tilde{c}_i^k. \qquad (77)$$

where T^{-1} is the inverse of the transform matrix T. Various transformations can be used in hybrid coding such as Hadamard, Fourier, and Karhunen-Loéve. The one-dimensional hybrid transform/DPCM system is shown in Figure 29.

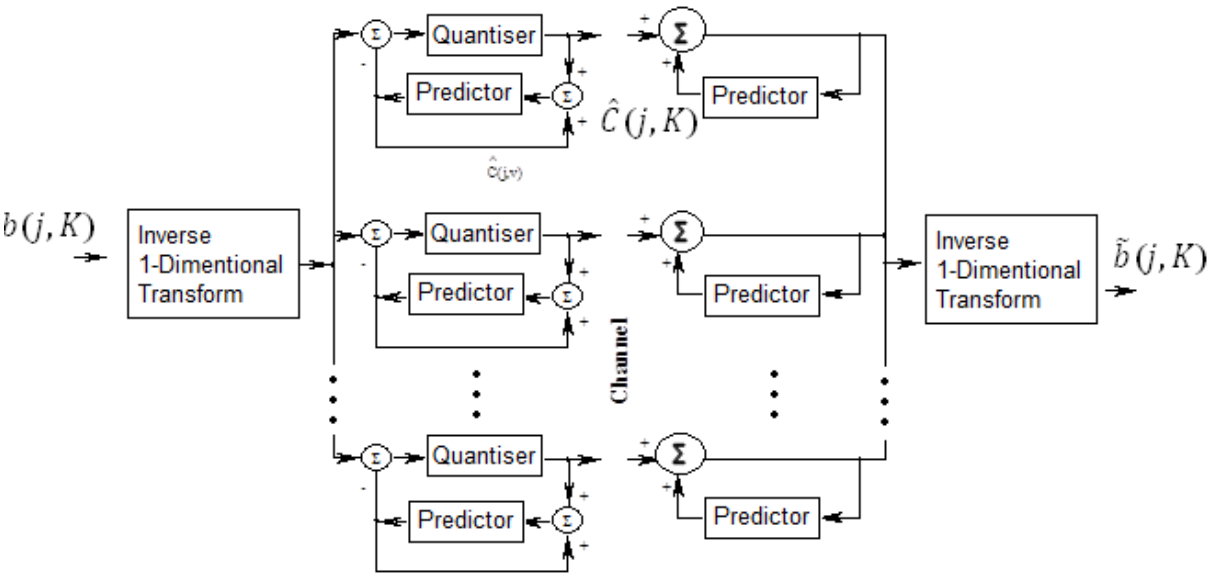


Figure 29. Hybrid coding (one-dimensional transform/DPCM coder).

Habibi [30] extended hybrid coding to two-dimensional transformations followed by a bank of DPCM systems. When the image is divided into large size subimages, the complexity of transform coding is increased. On the other hand, by employing small size subimages, the mathematical complexity is reduced at the expense of the efficiency of the coding. Therefore, a two-dimensional transformation followed by a bank of DPCM systems could improve the performance of the system by removing the correlation in the neighbouring pixels and utilising small block sizes.

Hussain et al. [90] generated hybrid predictive wavelet coding. The system encoder consists of three stages. In the first stage, functional link neural network [91-93] is utilised as predictor structure. The error signals are forwarded to the second stage of the encoder. In this case, a Discrete Wavelet Transform [108] is utilised to transform the error signals into Wavelet coefficients. In the final stage, the transform coefficients of the most significant band will be Entropy encoded using Arithmetic Coding. Figure 30 shows the structure of their encoder.

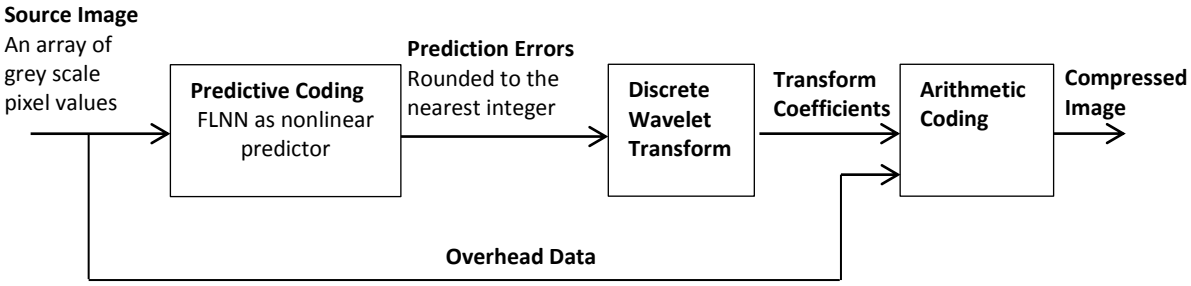


Figure 30: The encoder of the proposed neural network hybrid image compression system.

3.8 Image Compression using Neural Networks

Neural networks are suitable for image compression because of their nonlinearity, massively parallel structure, and training capabilities [143-175]. They have characteristics similar to those of the human visual system, and as a result, they can be applied to the processing of visual information [178] and other applications [195-196].

The parameters of the encoder which are the weights of the neural network need to be trained before encoding. This means that the training sets have to be carefully selected to represent the appropriate class of images for which the compression system is used. Furthermore, for neural networks based image compression systems, we should account for the increase in the bit rate caused by transmitting extra overhead information containing the encoder parameters [70-75].

3.8.1 Basic Image Compression System

Multilayer perceptrons can be used as a complete image compression system. In this case, the compressor and the decompressor (CODEC) are implemented in a single network. The network is trained using backpropagation. The network consists of two layers, the hidden and the output layers. The input neurons work as buffers where the image pixel values are forwarded. The image is divided into nonoverlapped blocks, with the elements of each block forwarded to the input neurons. Hence, the number of input nodes is equal to the size of the image block.

The number of neurons in the hidden layer is less than the number of inputs and corresponds to the compressed ratio, which is the ratio of input to hidden neurons. The outputs of the

hidden neurons are quantised with the same number of bits as the original input pixels and are stored for later use or transmitted to the communication channel. The outputs of the hidden layer are the reconstructed image data. Therefore, the number of neurons in the output layer is equal to the number of inputs [179].

This model is developed by Sonehara et al. [180] where the sigmoid transfer function is used in the hidden and the output layers. The inputs are scaled between 0 and 1 to match the range of the outputs. The simulation results demonstrated poor performances for the testing and the training images.

A similar approach was followed by Sicuranza and Ramponi [181] achieving a SNRs between 19.4 dB and 20.5 dB for the training images and between 18.2 dB and 19.4 dB in the case of generalisation.

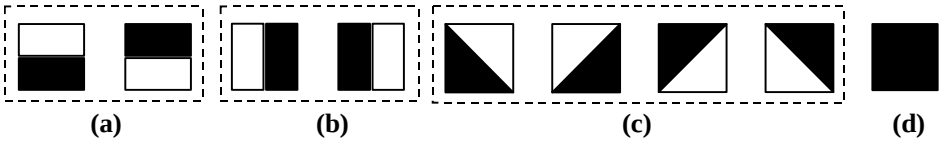


Figure 31: Pattern classes, (a) Horizontal edges, (b) Vertical edges, (c) Diagonal edges, (d) Shade [182].

To further improve this technique, Qiu et al. [182] used several mutilayer perceptrons neural networks to decrease edge degradation in the compressed images. In this case, the image is divided into a number of blocks of size $m \times n$. Then, each block is classified according to its orientation using two directional derivative approximations G_x and G_y determined as follows:

$$G_x = \frac{2}{m \times n}\left[\sum_{i=m/2}^{m-1}\sum_{j=0}^{n-1} b_{ij} - \sum_{i=0}^{(m/2)-1}\sum_{j=0}^{n-1} b_{ij}\right], \tag{78}$$

and

$$G_y = \frac{2}{m \times n}\left[\sum_{i=0}^{m-1}\sum_{j=n/2}^{n-1} b_{ij} - \sum_{i=0}^{m-1}\sum_{j=0}^{(n/2)-1} b_{ij}\right].$$

where b_{ij} is the pixel value of the block. The gradient magnitude and orientation within each block are determined according to:

$$|G| = \sqrt{G_x^2 + G_y^2} \tag{79}$$

and

$$\angle G = \tan^{-1}\left[\frac{G_y}{G_x}\right]$$

The gradient magnitude is then compared to a threshold value. If the gradient magnitude is less than the threshold value, its corresponding block is considered as a shade, otherwise it is a directional block [182].

Nine classes are defined and described as follows: two horizontals, two verticals, four diagonals and one shade as shown in Figure 31. After classification, different neural network structures are used to represent the various orientations and the compression is performed to the residual blocks, which are determined by subtracting the mean value of each block from its individual elements. As a result, in this method the compressed data contains quantised outputs of the hidden layer and the mean value of each block. The simulation results showed improvements over the method selected by Cottrell *et. al.* [183] in which for a bit rate of 1 bit/pixel an average peak signal to noise ratio of 31.79 dB and 30.57 dB are achieved for two training and four test images, respectively.

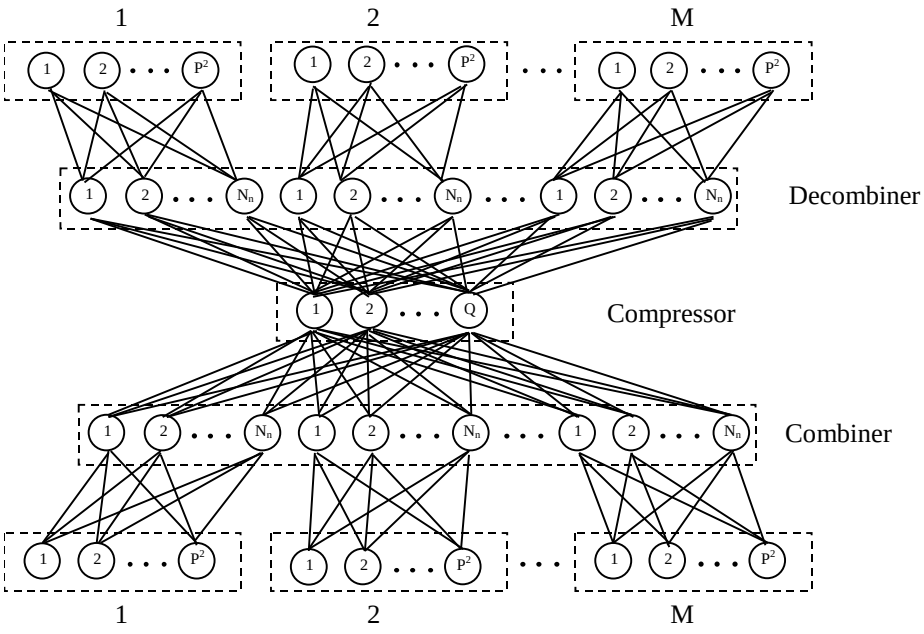


Figure 32: The structure of the hierarchical compression-decompression system [184].

A more complicated CODEC structure is proposed by Namphol and Chin [184], which is called hierarchical neural networks (refer to Figure 32). The system consists of three sections, which are the input-layer, the hidden-layer, and the output-layer sections. At the input layer, there are M blocks corresponding to the number of nonoverlapped blocks of the image. Each block is constructed from P2 nodes, corresponding to the size of each block of the image. The

hidden-layer section is constructed from the combiner, the compressor, and the decombiner. The first structure acts as a multiplexer to the input subblocks and contains less nodes than the number of neurons of the input layer. The second structure has less neurons than the combiner and its outputs represent the compressed data, which are stored for latter processing. The reconstruction of the compressed image involves decombining the stored data using the decombiner which acts as the demultiplexer. The output of the decombiner is forwarded to the output-layer section in which the compressed image data is reconstructed.

Hierarchical neural network is used to compress the Lena and the Mandrill images at a compression rate of 1 bit/pixel using various block sizes. The authors noticed that the performance of the network is enhanced as the block size increased. They suggested that since the network showed learning and generalisation abilities over a wide range of images and robustness when hidden nodes are damaged, it should be considered as an alternative to traditional techniques.

Benbsenisti et al. [185] disagreed with Namphol and Chin because although hierarchical neural networks have better performance than single-stage neural networks, they show potential increase in the complexity of the network structure and required more computational power. Therefore, they suggested the use of single-structure neural networks, utilising normalisation approach, i.e., the image grey levels are normalised according to:

$$x_i(\text{new}) = x_i(\text{old}) - L_i / s, \qquad (80)$$

where xi(new) is the new grey level image value at location i, L_i is a low-pass image, and s is a measure of the speed among the residual grey level differences.

3.8.2 Vector Quantisation

Neural networks are used to design optimal codebooks in vector quantisation. Most of these neural network structures use competitive learning algorithms [186].

Nasrabadi and Feng [187] are the first to use Kohonen self-organising feature maps (SOFMs) in designing optimal codebooks for vector quantisation. Two codebooks are designed for high and low variance blocks with 284 and 64 entries, respectively. The new codebook provides an improvement of 1 dB when compared to a codebook designed using the LBG algorithm and compressed with the same bit rate. Furthermore, SOFMs are relatively easy to implement and require less computational power than the LBG algorithm.

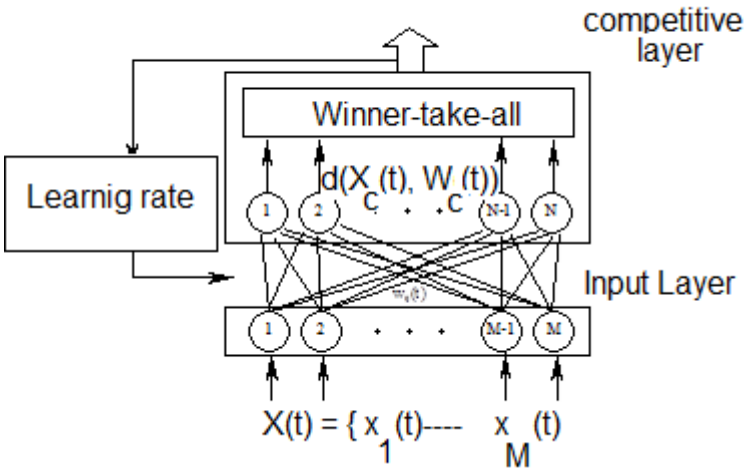


Figure 33: The structure of the FSO neural network [ref. 198].

A similar technique was proposed by Lu and Shin [188], but differs in classifying the edges into horizontal, vertical, and diagonal classes to preserve edge orientations. Therefore, four different codebooks are used instead of two. The edges are classified using multilayer perceptron network trained using a predefined set of binary edge blocks. The authors compared their method with the LBG algorithm in which an improvement of approximately 0.5 dB was achieved with the same compression rate. This result could be further enhanced using nine classes as suggested by Qiu et al. [182].

Although Kohonen self-organising feature maps provided good results when used to design optimal codebook structures, their major problem is that some of the neurons are under utilised. This means that the codebook may contain some untrained vectors, which cannot represent all structure varieties contained in the image. To overcome this problem, Chen et al. [189] proposed frequency sensitive self-organisation (FSO) neural network as shown in Figure 33.

The network consists of a single competitive layer. The inputs are forwarded to the network by the input nodes which distribute the input data X_p to the competitive layer. The distortion between the codevector W_i and the input X_p is calculated at the competitive layer where the winner-take-all operation is performed. The network solves the problem of under-utilisation by including frequency measures.

The training starts by randomly initialising the codevectors, then for each input vector the frequency-sensitivity distortion is determined as follows:

$$FD_i = (1 + \frac{F_i}{F_{thd}}) \sqrt{\sum_{j=1}^{N} (x_{PJ} - w_{ij})^2} \qquad (81)$$

where F_i is the usage frequency of neuron i, and F_{thd} is a frequency threshold value. By introducing the frequency of usage of each neuron, the method prevents under-usage or vastly over-usage. Codebook designed in this way solely is called 1-path codebook. The method starts by designing a 1-path codebook and proceeds by eliminating neurons with small frequencies. Then over-used codevectors are split into two new generated codevectors by adding and subtracting a small constant value vector to produce 2-path codebook. Such process is repeated until the required N-path codebook is designed.

The simulation results indicate that good reconstruction images and fairly high compression ratios can be obtained. Furthermore, for a 4-path or 5-path codebook, an increase in the quality of the compressed image of 1.5 dB can be obtained over 1-path and the Kohonen self-organising feature map codebooks [179].

3.8.3 Predictive Coding

The nonlinear nature of images makes nonlinear predictors more appropriate. Unfortunately, they are time consuming and usually impossible to design optimal nonlinear predictors [190].

Neural networks may provide a solution to this problem since the nonlinearity is embedded in their structures and their training algorithms are simple and tractable.

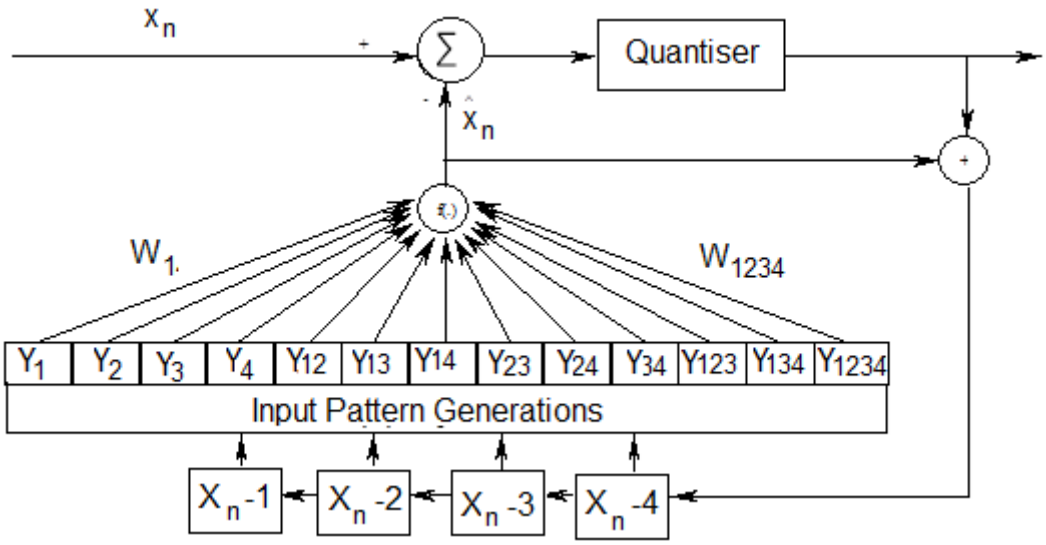


Figure 34: Architecture of practical functional-link type neural network predictor for 4th order generalised 1-D AR source [190].

Y1=Xn-1, Y2=Xn-2, Y3=Xn-3, Y4=Xn-4.
Y12=Xn-1 Xn-2, Y13=Xn-1 Xn-3, Y14=Xn-1 Xn-4, Y23=Xn-2 Xn-3, Y24=Xn-2 Xn-4, Y34=Xn-3 Xn-4.
Y123=Xn-1 Xn-2Xn-3, Y124=Xn-1 Xn-2Xn-4, Y234=Xn-2 Xn-3Xn-4.
Y1234=Xn-1 Xn-2Xn-3Xn-4.

Dianat et al. [191] suggested the use of mutilayer perceptrons as nonlinear predictor structures in DPCM. The network consists of three inputs representing the immediate causal neighbours, a hidden layer of 30 neurons and one output unit for the prediction of the current pixel value. The experiments demonstrated an improvement in the SNR of 4 dB over the optimal linear DPCM predictor. He and Li [192] reported similar results, while stressing the use of the quantised prediction error in the network cost function.

Another neural network structure that is used in predictive image coding system is the functional link neural network proposed by Li and Manikopoulos [193] and is referred to as higher order neural network (refer to Figure 34). The network can appropriately approximate sharply defined structures in images such as edges. The network is first tested to predict a simplified Volterra sequence defined as:

$$V_n = v_n + 0.5 v_n v_{n-1} - 0.5 v_{n-1} v_{n-2} + 0.5 v_{n-2} v_{n-3} v_{n-4}, \qquad (82)$$

where v_n is a sequence of Gaussian random variables with mean 0.429.

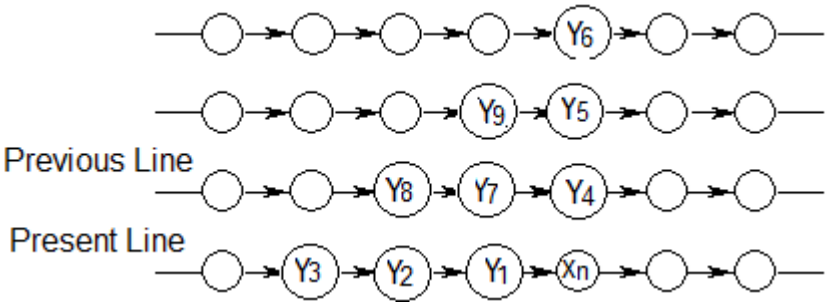


Figure 35: Geometry of prediction schemes for a two-dimensional nonlinear algorithm.

Manikopoulos [194] further enhanced the structure of the network by suggesting the use of adaptive predictive image coding system called neural net adaptive DPCM (NNADPCM) applied to two-dimensional image prediction. In this case, the image is divided into square subblocks of size 32×32. Nine previous values of the image are used to predict the current signal as demonstrated in Figure 35. The network is trained adaptively for each subblock and the trained weights are transmitted to the receiver and used as the initial weights for the successive image subblock. The predicted error is forwarded to a three levels Max quantiser and passed to Huffman coding to achieve an average bit rate and peak SNR of 0.51 bit/pixels and 29.5 dB, respectively.

4. IMAGE COMPRESSION IN MEDICAL IMAGING

Digital images are subject to a wide variety of distortions during acquisition [198]. Medical images face severe requirements for the quality of the reconstructed data. Fidelity loss must be reduced in medical applications to avoid diagnostic errors. Ideally, the decoded image should be identical to the original one, meaning lossless compression. Medical imaging application demands lossless or high-fidelity image compression. This is because compression might affect the diagnostic value of the image and possibly contribute to diagnostic errors. Based on such stringent requirements on medical images, the application of image compression to the medical imaging environment is a challenging problem. Studies have shown that, compression ratios of 3: 1 or 4: 1 with more complex lossless compression are possible [94]. Lossless compression alone is insufficient to attain ratios better than 4:1. Additionally, the increasing volume of data generated by new imaging modalities such as multislice computed tomography (CT) scanners and magnetic resonance imaging (MRI)

justifies the use of lossy compression techniques to decrease the cost of storage and improve the efficiency of transmission over networks for teleradiology or for access to electronic patient records [95].

Amri at al [96] developed image reduction and expansion techniques such as digital watermarking and lossless compression standards JPEG-LS (JLS) for the compression of medical images. They utilised MRI images to achieve good quality image reconstructions. They proposed the combination of the image size and encoding algorithms for lossless image compression.

Independent research groups [95], [97], [98], [99], [100], [11] [101] worked with various lossy compression algorithms indicated that lossy compressed medical images with high fidelity do not affect the diagnostic accuracy in a statistically significant way. Lossy techniques with compression ratios of about 10:1 are commonly adopted for picture archiving and communication system (PACS) for use in primary diagnosis and clinical review, using Digital Imaging and Communication in Medicine (*DICOM*), namely, *DICOM*JPEG or JPEG-2000 compression algorithms [95]. Lossy image compression research in medical imaging has traditionally concentrated on transform methods, especially the two-dimensional DCT [101], [102] or wavelet/subband coding methods [103].

Al-Fayadh et al. [105-107] proposed hybrid classified vector quantizer for the compression of medical images, their technique utilises classified vector quantisation [121-124]. The authors involve a number of neurosurgery specialised consultants experts in the field of medical imaging to judge the visual quality of their reconstructed compressed images, they have also involved a group of novice observers to participate in this visual assessment experiment [125]. Their results indicate that the hybrid technique generates high visual quality images from both the experts and the novice observers' point of views. Their method shows slightly better quality against JPEG-2000 for approximately the same bitrate.

Sophie et al. [104] developed a compression technique for medical images using wavelet transformation, normalization, and prediction. In their paper, they have indicated that their method can produce a good quality image close to the original image for the selected area of interest. The image undergoes 2D wavelet transform in order to find the coefficients. Normalization is then performed to achieve better prediction for each sub-band, separately. The resulted prediction errors are entropy-encoded using arithmetic coding technique.

Vidhya [177] proposed the use of threshold-based medical image compression algorithm. The algorithm can be used for the compression of magnetic resonance imaging, computed tomography and ultrasound images. The technique utilised thresholding coefficient selection and one-level wavelet decomposition achieving a PSNR above 36 dB. The performance of the proposed technique achieves higher values than JPEG and the authors indicated that the use of 8x8 blocks by JPEG could have generated more artefacts. The paper however did not provide a comparison between the proposed technique and JPEG 2000.

Ravichandran et al. [177] used Wavelet Transform for the compression of 3D medical images, their arguments indicate that more medical images are created in hospitals and medical organizations using 2D and 3D. Their simulation results showed that 3D medical images have high frequency patterns, and therefore wavelets technique allows the generation of a better PSNR even at the higher compression ratio than 2D medical images.

5. DISCUSSION AND CONCLUSION

This paper provides a survey into the topic of image compression. The increasing demand for multimedia computing has led to the demand of using digital images. The manipulation, storage and transmission of these images in their raw form is very expensive, it significantly slows the transmission and makes storage costly. With the increasing use of digital imaging comes the need to handle larger volumes of digital image data. Data compression has become an essential part of modern digital communication, video signal processing, and storage systems. Although the bandwidth of communication networks has been increasing continuously, the introduction of new services and the expansion of the existing ones demand an even higher bandwidth. Image data compression is concerned with the minimisation of the volume of data used to represent an image.

Table 4: Comparison of standard Lossless Image Compression Techniques

Technique	Compression Rate	Problems
Arithmetic Coding [7],[15]	Low	The code is only optimal when all symbols probabilities are an integral power of ½
Huffman Coding [14]	Low	Significant mathematical calculation. It determine

		alphabet of symbols with known probability distribution.
Lossless predictive coding [2]	Low	Needs good predictor strcuture

There are two categories of image compression, lossy and lossless. Lossy compression requires the reconstructed image to be an exact replica of the original image. This type of compression is used for medical images constructions in which a loss of information may incur wrong diagnosis. Table 4 compares between standard lossless image compression techniques.

In contrast to error free coding, lossy image compression reduces the accuracy of the coded image in exchange for higher compression ratio. At the encoder, there exists a quantiser which limits the number of bits required to represent the image. The purpose of the quantiser is to remove psychovisual redundancy. There are three standard approaches to lossy image compression which are vector quantisation, predictive coding and transform coding. Combined system that uses the properties of various image compression coding to increase the efficiency is also provided and known as hybrid coding. Table 5 shows basic lossy image compression techniques.

Table 5: Comparison of Standard Lossy Image Compression Techniques

Technique	Compression Rate	Advantages	Problems
Predictive Coding [113], [137]	Low compression rate	Removes Interpixel redundancy.	Low compression rate. Degradation such as granularity, slope overload and edge-busyness [35]. Needs good quantiser and predictor structures sensitivity to channel noise
Transform Coding [48]	High compression rate	Frequency domain technique High compression due to block coding	Affected by the blocking attracts
JPEG [54], [56], [197]	High Compression	Uses Transform coding	Blocking artefact

JPEG 2000 [58]	High Compression	Current Standard Eliminates the blocking artefact Uses block coding to achieve high compression. Uses DWT. Can define ROI	More complicated than JPEG to implement
Vector Quantisation [209], [210]	High compression	Easy implementation of the Decoder High compression Has many improved techniques including classified VQ [211]	Codebook training and construction Edge degradation. Codebook training and construction.

Predictive image compression uses previous values obtained during the scanning of the image to predict the next value along the line. We usually perform the scanning from the left to right and from top to bottom. This order of scanning is performed as a matter of convention.

Transform coding differs from predictive coding by dividing the image into a number of blocks. While predictive coding uses the actual values of the pixels for prediction, transform coding is a frequency domain technique. JPEG and JPEG 2000 standards are based on transform coding.

Vector quantisation has wide areas of application, such as speech coding, speech recognition and image coding. The main concept of vector quantisation is that the image is divided into nonoverlapped blocks which will be used for coding. The blocks of data will be quantised. Each block is transferred into a one-dimensional vector. A lookup table is provided at the encoder where the index of the location of the vector in the lookup table that provides minimum distortion to the input vector will be transmitted to the communication channel or stored.

The application of image compression has widened and its benefits are far from being counted. Image compression has been and continues to be crucial to the growth of multimedia such as the use of digital computers in printing, publishing and video production. In addition, image compression can be used in television or satellite transmission, video conferencing, facsimile transmission of printed material, graphics sensing images obtained from reconnaissance aircraft. Another area where image compression is applicable is where pictures are stored in a database, such as archiving of medical images, multispectral images, fingerprints, and drawings. As a result, image compression is now one of the most commercial applications and demonstrates great potential.

REFERENCES:

[1]. Netravali, A. N. and J. O. Limb. (1980). "Picture coding: A review," Proceedings of the IEEE, 68 (3), pp. 366-406

[2]. Gonzalez, R. C and R. E. Woods. (1992). *Digital image processing*, Addison-Wesley, Reading, pp. 307-411.

[3]. Jain, A.K. (1981). "Image data compression: A review," *Proceedings of the IEEE*, **69 (3)**, pp. 349-389.

[4]. R. C. Gonzalez, R. E. Woods, and S. L. Eddins. Digital Image Processing Using MATLAB. *Pearson Prentice Hall*, 2004.

[5]. R. C. Gonzalez and R. E. Woods. Digital Image Processing. *Addison- Wesley*, 3rd edition, 1992.

[6]. B. Jähne. Digital Image Processing - Concepts, Algorithms and Scientific Applications. *Springer-Verlag*, second edition, 1993.

[7]. R. J. Clarke. Digital Compression of Still Images and Video. *Academic Press, 2nd. Ed., London*, 1996.

[8]. S. E. Umbaugh. Computer Imaging: Digital Image Analysis and Processing. CRC Press, 2005.

[9]. **]** K. Sayood. Introduction to Data Compression. *Academic Press*, 2nd Edition, 2000.

[10]. A. Gersho and R. M. Gray. "Vector Quantization and Signal Compression," *KluwerAcademic Publishers*, 1992.

[11]. P. C. Cosman, K. Oehler, E. A. Riskin, and R. M. Gray, "Using Vector Quantizers for Image Processing", *Proc. IEEE,* Vol.81, No. 9, pp. 1326-1341, 1993.

[12]. R. M. Fano, "The Transmission of Information", *Technical Report* 65, *Research Laboratory of Electronics*, MIT, Cambridge, MA, 1949.

[13]. C. E. Shannon and W. Weaver. The Mathematical Theory of Communication. *University of Illinois Press*, Urbana, 1949.

[14]. D. A. Huffman, "A method for the construction of minimum redundancy codes", Proc. IRE, No. 40, pp. 1098-1101, Sep. 1952.

[15]. G. Glen and J. Langdon, "An Introduction to arithmetic coding," *IBM J. Res. Develop.*, Vol. 28, No. 2, pp. 135-149, 1984.

[16]. A. J. Hussain, "Polynomial Neural Networks and Applications to Image Compression and Signal Prediction," *PhD thesis, the university of Manchester Institute of Science and Technology*, 2000.

[17]. I. H. Witten, T. C. Bell, and A. Moffat. Managing Gigabytes. Compressing and Indexing Documents and Images. *Morgan Kaufmann Publishers*, Inc., 1999.

[18]. U, Gina. Linear Predictive Coding in Voice Conversation. 2004.

[19]. B. Meyer and P. Tischer, "TMW- a New Method for Lossless Image Compression", *International Picture Coding Symposium PCS97 Conference Proc.*

[20]. B. Meyer and P. Tischer, "Extending TMW for Near Lossless Compression of Greyscale Images", *Proc. Data Compression Conference, IEEE Computer Society Press*, pp. 458–470, 1998.

[21]. B. Meyer and P. Tischer, "Glicbawls – Grey Level Image Compression By Adaptive Weighted Least Squares", *Proc. Data Compression Conference, IEEE Computer Society Press*, 503, 2001.

[22]. G. Motta, J. Storer and B. Carpentieri, "Adaptive Linear Prediction Lossless Coding", *Proc. Data Compression Conference, IEEE Computer Society Press*, pp. 491–500, 1999.

[23]. G. Motta, J. Storer, and B. Carpentieri, "Improving Scene Cut Quality for Real–Time Video Decoding", *Proc. Data Compression Conference, IEEE Computer Society Press*, pp. 470–479, 2000.

[24]. X. Wu, K. Barthel and W. Zhang, "Piecewise 2D Autoregression for Predictive Image Coding", *International Conference on Image Processing Conference Proc.*, Vol.3., 1998.

[25]. J. Makoul, "Linear Prediction: A Tutorial Review," *Proc. IEEE*, Vol. 63, pp. 561-580, 1975.

[26]. L. Rabiner and R. Schafer. Digital Processing of Speech Signals. *Prentice-Hall, Inc., Englewood Cliffs, NJ*, 1971.

[27]. J. Picone, "Speech Processing Lecture Notes," Mississippi State University, 1996.

[28]. A. Papoulis. Probability Random Variables and Stochastic Processes. *McGraw hill book company, Inc.*, New York, 1965.

[29]. J. B. O'Neal, "Predictive Quantisation System (Differential Pulse Code Modulation) for the Transmission of Television Signals", *Bell Syst. Techn. J.*, Vol. 45, pp. 689-721,

1966.

[30]. A. Habibi, "Compression of nth-order DPCM Encoder with Linear Transformations and Block Quantisation Techniques", *IEEE Trans. Commun*., Vol. COM-19, No. 6, pp. 948-956, 1971.

[31]. A. K. Jain. Fundamentals of Digital Image Processing. *Prentice Hall,Inc*.,1989.

[32]. J. H. Stott, "Digital Video bit-rate Reduction: DPCM for sub-nyquist pal", *BBC research department*, Report 1987/16, 1987.

[33]. Clarke, R. J. (1995). Digital compression of still images and video. Academic press, London.

[34]. C. N. Manikopoulos, "Neural Network Approach to DPCM System Design for Image Coding", *IEE Proceedings, Vision, Image and Signal Processing*, Vol. 139, No. 5, pp. 501-507, 1994.

[35]. N. Jayant and P. Noll [1984]. Digital Coding of Waveforms: Principles and Applications to Speech and Video. *Prentice–Hall*, 1984.

[36]. A. N. Netravali and B. G. Haskell. Digital Pictures: Representation, Compression, and Standards. 2nd Edition, *Plenum Press, New York*, 1995.

[37]. R. C. Gonzalez, R. E. Woods, and S. L. Eddins. Digital Image Processing Using MATLAB. *Pearson Prentice Hall*, 2004.

[38]. P. A. Wintz, "Transform Picture Coding", *Proceedings IEEE*, Vol. 60, pp. 809-820, July 1972.

[39]. H. C. Anderws and W. K. Pratt, "Fourier Transform Coding of Images", *Hawaii Int. Conf. Sys. Sci*., Western periodicals Company, North Hollywood, California, pp. 677-679, 1968.

[40]. G. B. Anderson and T. S. Huang, "Piecewise Fourier Transformation for Picture Bandwidth Compression", *IEEE Transactions on Communication Technol*., Vol. 19, No. 2, pp.133-140, 1971.

[41]. W. K. Pratt, J. Kane and H. C. Anderws, "Hadamard Transfer Image Coding", *IEEE Proceedings*, Vol. 57, No. 1, pp. 58-68, 1969.

[42]. W. K. Pratt. "Spatial Transform Coding of Colour Images," *IEEE Trans. Commun*. Vol. COM-19, pp. 980-992, 1971.

[43]. N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transform", In: *IEE Transactions on Computers*, Vol. 23, pp. 90-93, 1974.

[44]. W. K. Pratt. Image Transmission Techniques. *Academic Press*, New York, 1979.

[45]. K. Fukunaga. Introduction to Statistical Pattern Recognition. *New York Academic*, 1990.

[46]. H. Anton and C. Rorres. *Elementary Linear Algebra*. 7th Edition, *Anton Textbooks, Inc., New York* 1994.

[47]. H. C. Anderws and W. K. Pratt, "Television Bandwidth Reduction by Encoding Spatial Frequencies", *J. SMPTE,* Vol. 77, pp. 1279-1281, 1968.

[48]. R. J. Clarke. Transfer Coding of Images. *Academic Press, London,* 1985.

[49]. A. V. Oppenheim and R. W. Schafer. Discrete-time Signal Processing. *Englewood Cliffs, NJ: Prentice-Hall,* 1989.

[50]. Li Shang, D.S.Huang, Chun-Hou Zheng, and Zhan-Li Sun, "Noise removal using a novel non-negative sparse coding shrinkage technique," Neurocomputing, vol.69, nos.7-9, pp.874–877, 2006.

[51]. V. Bahaskaran and K. Konstantinides. Image and Video Compression Standards. 2nd Edition, *Kluwer International Series, Boston,* 1997.

[52]. Chun-Hou Zheng, D.S.Huang, Zhan-Li Sun, Michael R. Lyu, and Tat-Ming Lok, "Nonnegative independent component analysis based on minimizing mutual information technique," Neurocomputing, vol.69, nos.7-9, pp.878–883, 2006.

[53]. Zhan-Li Sun, D.S.Huang, and Chun-Hou Zheng, Li Shang, "Optimal selection of time lags for temporal blind source separation based on genetic algorithm," Neurocomputing, vol.69, nos.7-9, pp.884–887, 2006.

[54]. G. K. Wallace, "The JPEG Still Picture Compression Standard", *Comm. of the ACM,* Vol. 34, No.4, pp. 30-44, 1991.

[55]. W B Pennebaker and J Mitchell. JPEG: Still Image Data Compression Standard. *Van Nostrand- Reinhold,* New York, 1993.

[56]. C. Christopoulos, A. Skodras, T. Ebrahimi, "The JPEG2000 Still Image Coding System: An Overview", *IEEE Trans. on Consumer Electronics,* Vol.46, No.4, pp. 1103-1127, November 2000.

[57]. ISO/IEC FDIS 15444-1, "JPEG 2000 Part I Final Draft International Standard", August 2000.

[58]. ISO/IEC JTC 1/SC 29/WG 1, ISO/IEC FCD 15444-1, "Information Technology JPEG 2000 Image Coding System: Core Coding System [WG 1 N 1646]", March 2000, http://www.jpeg.org/FCD15444-1.htm.

[59]. M. Boliek, C. Christopoulos, and E. Majani (Eds), "JPEG 2000 Part 1 Final Draft International Standard," (FDIS15444-1), ISO/IEC JTC1/SC29/WG1, N1855, August 2000.

[60]. D. Taubman, "High Performance Scalable Image Compression With EBCOT*," IEEE*

Transactions on Image Processing, Vol 9, No.7, pp. 1158-1170, July 2000.

[61]. D. Santa-Cruz, T. Ebrahimi, J. Askelof, M. Larsson and C. Christopoulos. "JPEG 2000 Still Image Coding Versus other Standards," ISO/IEC JTC1/SC29/WG1 (ITU-T SG8), July 2000 http://www.jpeg.org/public/wg1n1816.pdf

[62]. Li Shang, D.S.Huang, Ji-Xiang Du, and Chun-Hou Zheng, " Palmprint recognition using FastICA algorithm and radial basis probabilistic neural network," Neurocomputing, vol.69, nos.13-15, pp. 1782-1786, 2006.

[63]. D.S.Huang, Systematic Theory of Neural Networks for Pattern Recognition, Publishing House of Electronic Industry of China, 1996.

[64]. D.S.Huang, "Radial basis probabilistic neural networks: Model and application," International Journal of Pattern Recognition and Artificial Intelligence, 13(7), pp.1083-1101, 1999.

[65]. D.S.Huang, Ji-Xiang Du, "A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks," IEEE Transactions on Neural Networks, vol. 19, no.12, pp 2099-2115, 2008.

[66]. D.S.Huang, W.B.Zhao, "Determining the centers of radial basis probabilistic neural networks by recursive orthogonal least square algorithms," Applied Mathematics and Computation, vol.162, no.1, pp.461-473, 2005.

[67]. W.B.Zhao, D.S.Huang, Ji-Yan Du and Li-Ming Wang, "Genetic optimization of radial basis probabilistic neural networks," International Journal of Pattern Recognition and Artificial Intelligence, vol. 18, no. 8, pp. 1473-1500, 2004.

[68]. D.S.Huang and S.D.Ma,"Linear and nonlinear feedforward neural network classifiers: A comprehensive understanding,"Journal of Intelligent Systems, vol.9, no.1, pp.1-38,1999.

[69]. D.S. Huang, The Study of Data Mining Methods for Gene Expression Profiles, Science Press of China, March 2009.

[70]. Bo Li, D.S.Huang, Chao Wang and Kun-Hong Liu, "Feature extraction using constrained maximum variance mapping," Pattern Recognition, vol.41, no.11, pp. 3287-3294, 2008.

[71]. Zhong-Qiu Zhao, D.S.Huang, "Palmprint recognition with 2DPCA+PCA based on modular neural networks," Neurocomputing, v**ol.71, nos.1-3,** pp. 448-454, 2007.

[72]. D.S.Huang, Jian-Xun Mi, "A new constrained independent component analysis method," IEEE Trans. On Neural Networks, vol.18, no.5, pp. 1532-1535, 2007.

[73]. Zhong-Qiu Zhao, D.S.Huang, "A mended hybrid learning algorithm for radial basis

function neural networks to improve generalization capability," Applied Mathematical Modelling, v**ol.31, no.7,** pp. pp.1271-1281, 2007.

[74]. Ji-Xiang Du, D.S.Huang, Xiao-Feng Wang, Xiao Gu, "Shape recognition based on neural networks trained by differential evolution algorithm," Neurocomputing, v**ol.70, nos.4-6,** pp. 896-903, 2007.

[75]. Ji-Xiang Du, D.S.Huang, Guo-Jun Zhang and Zeng-Fu Wang, "A novel full structure optimization algorithm for radial basis probabilistic neural networks," Neurocomputing, v**ol.70, nos.1-3,** pp. 592-596, 2006.

[76]. P. Drineas, A. Frieze, R. Kannan, S. Vempala and V. Vinay, "Clustering in Large Graphs and Matrices", ACM-SIAM Symposium of Discrete Algorithms 1999.

[77]. D. P. Berrar, W. Dubitzky, M. Granzow, eds., "Singular Value Decomposition and Principal Component Analysis", *In a Practical Approach to Microarray Data Analysis*. Kluwer: Norwell, MA, 91-109. LANL LA – UR – 02 – 4001, 2003.

[78]. R. M. Gray, "Vector quantization," *IEEE ASSP Magazine.* Vol. 1, pp. 4-29, April 1984.

[79]. K. K. Paliwal and V. Ramasubramanian, "Effect of ordering the codebook on the efficiency of the partial distance search algorithm for vector quantisation", *IEEE Trans. Commun*, Vol. 37, No. 5, pp. 538-540, 1989.

[80]. D.S.Huang, "A constructive approach for finding arbitrary roots of polynomials by neural networks," IEEE Transactions on Neural Networks, vol.15, no.2, pp.477-491, 2004.

[81]. D.S.Huang, and Wen Jiang, "A general CPL-AdS methodology for fixing dynamic parameters in dual environments," IEEE Trans. on Systems, Man and Cybernetics - Part B, vol.42, no.5, pp.1489-1500, 2012.

[82]. D.S.Huang, Horace H.S.Ip, Law Ken C K and Zheru Chi, "Zeroing polynomials using modified constrained neural network approach," IEEE Trans. On Neural Networks, vol.16, no.3, pp.721-732, 2005.

[83]. Xiao-Feng Wang, D.S.Huang and Huan Xu, "An efficient local Chan-Vese model for image segmentation," Pattern Recognition, vol. 43, no.3, pp. 603-618, 2010.

[84]. Xiao-Feng Wang, D.S.Huang, "A novel density-based clustering framework by using level set method," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no.11, pp 1515-1531, 2009.

[85]. Bo Li, Chun-Hou Zheng, and D.S.Huang, "Locally linear discriminant embedding: An efficient method for face recognition," Pattern Recognition, vol.41, no.12, pp. 3813-3821, 2008.

[86]. Wen Jiang, D.S.Huang, Shenghong Li, "Random-walk based solution to triple level stochastic point location problem," IEEE Trans. on Cybernetics, vol.46, no.6, pp.1438-1451, 2016.

[87]. Y. Linde, A. H. Buzo, and R. M. Gray, "An Algorithm for Vector Quantizer Design", *IEEE Trans. Commun.*, Vol. COM-28, pp. 84-95, 1980.

[88]. A. N. Netravali and J. O. Limb, "Picture coding: A review", *Proc. IEEE*, Vol. 68, No. 3, pp. 366-406, Mar. 1980.

[89]. A. Gersho and B. Ramamurthi, "Image Coding Using Vector Quantization", In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 428-431, 1982.

[90]. H.-N. Chen and K.-L.Chung, "Improved Adaptive Vector Quantisation Algorithm Using Hybrid Codebook Data Structure", *Real-Time Imaging*, Vol. 11, pp.270-281, 2005.

[91]. K.-L. Chung and J.-Y. Lai, "An Efficient Law-of-cosine-based Search for Vector Quantisation", *Pattern Recognition Letters*, Vol. 25, pp. 1613-1617, 2004.

[92]. Z. Pan, K. Kotani, and T. Ohmi, "An Improved Fast Encoding Algorithm for Vector Quantisation using 2-Pixel-merging Sum Pyramid Data Structure", *Pattern Recognition Letters*, Vol.25, pp.459-468, 2004.

[93]. G. Patane, and M. Russo, "The Enhanced LBG Algorithm", *Neural Networks*, Vol. 14, pp. 1219-1237, 2001.

[94]. P. Veprek, and A. B. Bradley, "An Improved Algorithm for Vector Quantizer Design", *IEEE signal processing letters*, Vol. 7, No.9, pp.250-252, Sep. 2000.

[95]. Netravali, A. N and B. G. Haskell, (1994). *Digital pictures representation, compression and standards*. Plenum press, New York.

[96]. M. Reid, R. Millar, N. B. "Second-generation Image Coding: An Overview," ACM Computing Surveys, 1997.

[97]. Hussain AJ, Al-Jumeily D, Radi N, Lisboa P, Hybrid neural network predictive-wavelet image compression system, Neurocomputing 151, pp. 975-984

[98]. Pao, Y., G. Park, and D. J. Sobajic, (1994). "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, **6**, pp. 163-180.

[99]. Pao, Y. H. and Y. Takefuji, (1992). "Functional-link net computing," *Computer*, **25(5)**, pp. 76-79.

[100]. Igelnik, B. and Y. H. Pao, (1995). "Stochastic choice of basis functions in adaptive function approximation and the functional link net," *IEEE Transactions on neural networks*, **6(6)**, pp. 1320-1329.

[101]. S. M. Perlmutter, *et al*., "Medical Image Compression and Vector Quantization," *Statistical Science*, Vol. 13, No.1, pp. 30-53, 1998.

[102]. D. A. Koff, H. Shulman**, "**An Overview of Digital Compression of Medical Images: Can We Use Lossy Image Compression in Radiology?", *Radiol J*., Vol. 57, No. 4, pp.

211-217, Oct. 2006.

[103]. Amri, H., Khalfallah, A., Gargouri, M. et al. J Sign Process Syst (2017) 87: 203.

[104]. K. Finnis, "A review and comparison of medical image compression", Located at: Atamai, Inc, Calgary, Alberta, August 2004.

[105]. KR. Persons, NJ. Hangiandreou , NT. Charboneau, J. Charboneau, E. James , BR. Douglas , et al, " Evaluation of irreversible JPEG compression for a clinical ultrasound practice. *J Digit Imaging*; 15, pp.15–21, 2002.

[106]. V. Savcenko, BJ. Erickson, PM. Palisson, KR. Persons, A. Manduca, TE. Hartman, et al, "Detection of Subtle Abnormalities on Chest Radiographs after Irreversible Compression", *Radiology*; Vol. 206, pp. 609–616, 1998.

[107]. P. C. Cosman, *et al.*, "Thoracic CT Images: Effect of Lossy Image Compression on Diagnostic Accuracy," *Radiology,* 190, pp. 517-524, Feb. 1994.

[108]. K.K. Chan, S.L. Lou, H.K. Huang, "Full-frame transform compression of CT and MR images", *Radiology*, Vol. 171, pp. 847-851, 1989.

[109]. T. Ishigaki, S. Sakuma, M. Ikeda, Y. Itoh, M. Suzuki, and S. Iwai, "Clinical evaluation of irreversible image compression: Analysis of chest imaging with computer radiography", *Radiology*, Vol. 175, No. 3, pp. 739-743, June 1990.

[110]. P. Saipetch, B. Ho, R. Panwar, M. Ma, and J. Wei, "Applying Wavelet Transforms with Arithmetic Coding to Radiological Image Compression", *IEEE Engineering in Medicine and Biology,* Vol. 14, No. 5, pp. 587-593, Sep.-Oct. 1995.

[111]. P. E. Sophia and J. Anitha (2017) Contextual Medical Image Compression using Normalized Wavelet-Transform Coefficients and Prediction, IETE Journal of Research, 63:5, 671-683

[112]. A. Al-Fayadh, A. J. Hussain, P. Lisboa, and D. Al-Jumeily, "Hybrid Classified Vector Quantisation and Its Application to Image Compression," *IEEE Int. Con. On Computer as a Tool,* EUROCON2007, Sep. 9-12, Warsaw, pp. 157-162, 2007.

[113]. A. Al-Fayadh, A. J. Hussain, P. Lisboa, and D. Al-Jumeily, "A Hybrid Classified Vector Quantisation and Its Application to Image Compression," *IEEE Int. Con*. ICSPC07, Nov. 24-27, Dubai, pp. 125-128, 2007.

[114]. A. Al-Fayadh, A. J. Hussain, P. Lisboa, D. Al-Jumeily and M. Al-Jumaily, "An Adaptive Hybrid Image Compression Method and Its Application to Medical Images," *IEEE Int. Symp. Biomedical Imaging* (ISBI '08), May 14-17, France, 2008.

[115]. M. Antonini, M. Barland, P. Mathieu, I. Daubechies, "Image Coding Using the Wavelet Transform", *IEEE Trans. on Image Processing*, Vol. 1, pp. 205-220, 1992.

[116]. S. Bauer, B. Zovko-Cihlar, M. Grgic, "The Influence of Impairments from Digital Compression of Video Signal on Perceived Picture Quality", *Proceedings of the 3rd International Workshop on Image and Signal Processing*, IWISP'96, Manchester, pp. 245-248, 1996.

[117]. H. S. Chang and K. Kang, "A Compressed Domain Scheme for Classifying Block Edge Patterns," *IEEE Trans. Image Processing*, Vol. 14, No. 2, Feb. 2005.

[118]. T. Chen, H. R. Wu, and B. Qiu, "Adaptive post-filtering of transform coefficients for the reduction of blocking artifacts", IEEE Trans. Circuits and systems for video technology, Vol. 11, 2001.

[119]. P.C Cosman, R.M Gray and M.Vetterli, "Vector Quantization of Image Sub bands: A Survey," IEEE Transactions on image processing, September 14th, 1995.

[120]. C. C. Culter, "Differential quantisation of communication signals", U. S. Pat., No. 2, 605, 361, 1952.

[121]. M. R. El-Sakka and M. S. Kamel, "Adaptive Image Compression Based on Segmentation and Block Classification", *Int. Journal of Imaging Systems and Technology*, Vol.10, No. 1, pp. 33-46, 1999.

[122]. W. H. Equitz, "A New Vector Quantization Clustering Algorithm", *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. 37, No. 10, pp. 1568–1575, Oct.1989.

[123]. A. M. Eskicioglu, " Quality Measurement for Monochrome Compressed Images in the Past 25 years", *IEEE International Conference on Acoustics, Speech and Signal Processing – Proceedings* 2000, Vol. 4, pp. 1907–1910, 2000.

[124]. M. Eskicioglu, P. S. Fisher, "Image Quality Measures and Their Performance", *IEEE Transactions on Communications*, Vol. 43, No. 12, pp. 2959-2965, December 1995.

[125]. R. M. Gray and D. L. Neuhoff, "Quantization", *IEEE Trans. Inform. Theory*, Vol. 44, No. 6, pp. 2325-2383, Oct. 1998.

[126]. S. – MRAK, GRGIĆ, M. – GRGIĆ, M., "Comparison of JPEG Image Coders", *Proc. of the 3rd International Symposium on Video Processing and Multimedia Communications*, VIProm- Com-2001, Zadar, Croatia, pp. 79-85, 2001.

[127]. J. Haber and H. P. Seidel, "Using an Enhanced LBG Algorithm to Reduce the Codebook Error in Vector Quantisation", *In Proc. Comput. Graph. Int.*, pp.99-104, 2000.

[128]. A. J. Hussain, A. Al-Fayadh, D. Al-Jumeily, P. Lisboa and P. Liatsis, " Image Compression Using Improved Hybrid Classified Vector Quantisation", *Electronics Letters*, Vol. 43, No. 7, pp. 385-387, March 2007.

[129]. M. H. Lee, and G. Crebbin, "Classified Vector Quantisation with Variable Block-Size DCT", *IEE Proc. Vis. Image Process*. Vol. 144, No. 1, pp. 39-48, 1994.

[130]. J. W. Kim and S. U. Lee, "A Transform Domain Classified Vector Quantizer for Image Coding", *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 2, No.1 pp.3-14, 1992.

[131]. B. Ramamurthi and A. Gersho, "Classified Vector Quantization of Images", *IEEE Trans. Commun.* COM-34, pp. 1105-1115, 1986.

[132]. ITU, "Methods for the Subjective Assessment of the Quality of Television Pictures", *ITU-R Rec. BT*. 500-7, August 1998.

[133]. N. Keskes, F. Kretz, and H. Maitre, "Statistical Study of Edges in TV Pictures", IEEE *Trans*., Vol. COM-27, pp. 1239-1247, 1979.

[134]. W. Kou. Digital Image Compression: Algorithms and Standards. Kluwer Academic Publishers, 2nd edition, 1995.

[135]. C.-H. Lee and L.-H. Chen, "Novel Image Compression Method Using Edge-Oriented Classifier and Novel Predictive Noiseless Coding Method", *IEE Proc.-Vis. Image Signal Process.* Vol. 144, No. 6, Dec.1997.

[136]. N. W. Lewis and J. A. Allnatt, "Subjective quality of television pictures with multiple impairments", *Electron. Lett*. Vol. 1, pp. 187–188, July 1965.

[137]. S. P. LIoyd, "Least Squares Quantization in PCM", *IEEE Trans. Inform. Theory*, Vol. 1T-28, pp. 129-137, March 1982.

[138]. T. D. Lookabaugh and M. R. Gray, "High-resolution Quantization Theory and The Vector Quantization Advantage", *IEEE Trans. Inform. Theory*, Vol. 35, No. 5, pp. 1020-1033, Sep. 1989.

[139]. J. Makhoul, S. Roucos, and H. Gish, "Vector Quantization in Speech Coding", *Proc. IEEE*, Vol 73, pp. 1551-1588, Nov. 1985.

[140]. N. Nasrabadi and R. King, "Image Coding Using Vector Quantization: A review", *IEEE Trans. Commun*. Vol. COM-36, pp. 957-971, 1988.

[141]. M. Nelson. The Data Compression Book. 2nd Edition, *M and T Books, New York,* New York 10011, 1996.

[142]. M. K. Quweider and J.B. Farison, "Classification of Image Edge Blocks by Orthogonal Projection", *Electron. Lett*. Vol. 33, No. 22, pp.1856-1857, 23rd Oct. 1997.

[143]. C. E. Shannon. "Coding theorems for a discrete source with a fidelity criterion. In *IRE National Convention Record*, Part 4, pp. 142-163, 1959.

[144]. C. E. Shannon, "A Mathematical Theory of Communication", *Bell. Syst. Tech*. J., Vol.

27: pp.379-423, 623-656, 1948.

[145]. J. M. Shapiro, "An Embedded Wavelet Hierarchical Image Coder", Proc. *IEEE Int. Conf. Acoust. Speech, Signal Processing*, Vol. 4, March 1992.

[146]. J. M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", *IEEE Transactions on Signal Processing*, Vol. 41, pp. 3445-3462, 1993.

[147]. O. Virmajoki, P. Franti, T. Kaukoranta, "Practical Methods for Speeding-up The Pairwise Nearest Neighbor Method", Optical Engineering, Vol. 40, No.11, pp. 2495–2504, Nov. 2001.

[148]. C. S. Won, D. K. Park, and S.-J. Park, "Efficient Use of MPEG-7 Edge Histogram Descriptor, *ETRI J.*, Vol. 24, No. 1, pp. 23-30, Feb. 2002.

[149]. K-L. Wu, M-S Yang, "Alternative Learning Vector Quantization", Pattern Recognition, Vol. 39, pp. 351-362, 2006.

[150]. M. D. Adams and R. Ward, "Wavelet Transforms in the JPEG-2000 Standard," in *Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, BC, Canada*, Vol. 1, pp. 160-163, Aug. 2001.

[151]. T. Cornsweet. Visual Perception. *Academic Press, New York*, 1971.

[152]. R. Clark. "An Introduction to JPEG 2000 and Watermarking," **DOI:** 10.1049/ic:20000214

[153]. S. Gilbert. Introduction to Linear Algebra. *Wellesley-Cambridge Press*, 1998.

[154]. R. Kakarala and P. O.Ogunbona, "Singular Analysis Using a Multiresolution Form of the Singular Value Decomposition", *IEEE Trans. Image Processing*, Vol. 10, No.5, pp. 724-735, May 2001.

[155]. H. C. Andrews, C. L. Patterson, " Singular Value Decomposition (SVD) Image Coding", *IEEE Trans. Commun*, pp. 425-432, 1976.

[156]. C. S. M. Goldrick, W. J. Dowling, and A. Bury, "Image Coding Using the Singular Value Decomposition and Vector Quantization", In *Image Processing And Its Applications, IEE*, pp. 296–300, 1995.

[157]. Zhan-Li Sun, D.S.Huang, and Yiu-Ming Cheung, "Extracting nonlinear features for multispectral images by FCMC and KPCA," Digital Signal Processing, vol.15, no.4, 331-346, 2005.

[158]. Zhan-Li Sun, D.S.Huang, Yiu-Ming Cheung, Jiming Liu and Guang-Bin Huang, "Using FCMC, FVS and PCA techniques for feature extraction of multispectral images," IEEE Geoscience and Remote Sensing Letters, vol.2, no.2, pp.108-112, 2005.

[159]. Jian-Xun Mi, D.S.Huang, Bing Wang, Xingjie Zhu, "The nearest-farthest subspace

classification for face recognition," Neurocomputing, vol.113, pp.241-250, 2013.

[160]. Can-Yi Lu, and D.S.Huang, "Optimized projections for sparse representation based classification," Neurocomputing, vol.113, pp.213-219, 2013.

[161]. Yang Zhao, and D.S.Huang, "Completed local binary count for rotation invariant texture classification," IEEE Trans. on Image Processing, vol.21, no.10, pp. 4492 - 4497, 2012.

[162]. Bo Li, Chao Wang and D.S.Huang, "Supervised feature extraction based on orthogonal discriminant projection," Neurocomputing, vol. 73, nos.1-3, pp 191-196, 2009.

[163]. Xiao-Feng Wang, D.S.Huang, Ji-Xiang Du, Huan Xu, Laurent Heutte, "Classification of plant leaf images with complicated background," Applied Mathematics and Computation, vol. 205, no.2, pp 916-926, 2008.

[164]. Xiao-Feng Wang, D.S.Huang, "A novel multi-layer level set method for image segmentation," Journal of Universal Computer Science, vol.14, no.14, pp.2428-2452, 2008.

[165]. P. Waldemar and T. A. Ramstad, "Image Compression Using Singular Value Decomposition with Bit Allocation and Scalar Quantization", In *Proceedings of NORSIG Conference*, pp. 83–86, 1996.

[166]. S.O. Aase, J. H. Husoy and P. Waldemar, "A Critique of SVD-Based Image Coding Systems", *IEEE International Symposium on Circuits and Systems VLSI*, Vol. 4, pp. 13-16, Orlando FL., May 1999.

[167]. A. Dapena and S. Ahalt, "A Hybrid DCT-SVD Image-Coding Algorithm", IEEE Trans. Circuits Syst. Video Technol., Vol. 12, No. 2, pp. 114-121, Feb. 2002.

[168]. M. Tian, Si-Wei Luo, Ling-Zhi Liao, "An Investigation into Using Singular Value Decomposition as a Method of Image Compression", *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference*, Vol. 8, pp.5200-5204, 18-21 Aug. 2005.

[169]. T.-C. Lu, C.-C. Chang, and Y.-L. Lin, "A Content-Based Image Authentication Scheme Based on Singular Value Decomposition", Springer Science + Business Media LLC., Vol. 16, No.3, pp. 506-522, July 2006.

[170]. O. O. Vergara Villegas, P.P. Elias, and V. G. Cruz Sanchez, "Singular Value Decomposition Image Compression System for Automatic Object Recognition", *the IASTED Conference on Advances in Computer Science and Technology*, Puerto Vallarta, Mexico 23-25, Jan. 2006.

[171]. Ji-Xiang Du, D.S.Huang, Xiao-Feng Wang, and Xiao Gu, "Computer-aided plant species identification (CAPSI) based on leaf shape matching technique," Transactions of the Institute of Measurement and Control, vol. 28, no. 3, pp. 275-284, 2006.

[172]. M. W. Marcellin, M. J. Gormish, A. Bilgin, and M. P. Boliek. "An Overview of JPEG-2000," in *Proceedings of 2000 Data Compression Conference*, pp. 523 –544, March.

[173]. T. Bose. Digital Signal and Image Processing. *John Wiley and Sons, Inc*. 2004.

[174]. J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall. MPEG Video Compression Standard. *New York: Chapman and Hall*, 1997.

[175]. B. Shen and I. K. Sethi, "Direct Feature Extraction from Compressed Images", In *Proc. SPIE: Storage & Retrieval for Still Image and Video Databases IV*, Vol. 2670, pp. 404-414, Mar. 1996.

[176]. Vidhya, K, "Medical Image Compression using Adaptive Subband Threshold", Journal of Electrical Engineering & Technology, 11(2), pp. 499-507, 2016.

[177]. D.Ravichandran, M. G. Ahamad, Ashwin Dhivakar, "Performance Analysis of Three-Dimensional Medical Image Compression Based on Discrete Wavelet Transform", Proceedings of the International Conference on Virtual Systems & Multimedia (VSMM), pp. 351-358, 2016.

[178]. Dony, R. D. and S. Haykin, (1995). "Neural network approaches to image compression," *Proceedings of the IEEE*, **83(2)**, pp. 288-303.

[179]. Cramer, C. (1998). "Neural networks for image and video compression: A review," *European journal of operational research*, **108**, pp. 266-282.

[180]. Sonehara, N., M. Kawato, S. Miyake, and K. Nakane, (1989). "Image data compression using neural network model," *IEEE proceedings of the international joint conference on neural networks*, Washington DC, pp. 35-41.

[181]. Sicurana, G. L. and G. Ramponi, (1990). "Artificial neural network for image compression," *Electronic letters*, **26(7)**, pp. 477-479.

[182]. Qiu, G., M. R. Varley, and T. J. Terrell, (1993). "Image compression by edge pattern learning using multilayer perceptrons," *Electronic letters*, **29(7)**, pp. 601-603.

[183]. Cottrell, G. W., P. Munro, and D. Zipser, (1989). "Image compression by back propagation: An example of extensional programming," in models of cognition: a review of cognitive science, N. E. Sharkey, Ed., Norwood, pp. 208-240.

[184]. Namphol, A., S. H. Chin, M. Arozullah, (1996). "Image compression with a hierarchical neural network," *IEEE Transactions on aerospace and electronic systems*, **32(1)**, pp. 326-337.

[185]. Benbenisti, Y., D. Kornreich, H. B. Mitchell, and P. A. Schaeffer, (1997) "A high performance single-structure image compression neural network," *IEEE Transactions on aerospace and electronic systems*, **33(3)**, pp. 1060-1063.

[186]. Rizvi, S. A. and N. M. Nasrabadi, (1999). "Neural networks for image coding: A review," *Proceedings SPIE, Applications of Artificial Neural Networks in Image Processing IV*, San Jose, California, **3647**, pp. 46-57.

[187]. Nasrabadi, N. M., and Y. Feng, (1988). "Vector quantisation of images based upon Kohonen self organization feature maps," *IEEE proceedings of the international joint conference on neural networks*, San Diego, pp. 101-108.

[188]. Lu, C. and Y. H. Shin, (1992). "A neural network based image compression system," *IEEE Transactions on consumer electronics*, **38(1)**, pp. 25-29.

[189]. Chen, O. T., B. J. Sheu, and W. Fang, (1994). "Image compression using self-organization networks," *IEEE Transactions on circuits and systems for video technology*, **4(5)**, pp. 480-489.

[190]. Manikopoulos, C. N. (1992). "Neural network approach to DPCM system design for image coding", *IEE Proceedings-I*, **139 (5)**, pp. 501-507.

[191]. Dianat, S. A., N. M. Nasrabadi, and S. Venkataraman (1991) "A non-linear predictor for differential pulse-code encoder (DPCM) using artificial neural networks", *Proc. ICASSP*, pp. 2793-2796.

[192]. He, Z. and H. Li, (1990). "Nonlinear predictive image coding with a neural networks," *ICASSP Proc.*, pp. 1009-1012.

[193]. Li, J. and C. N. Manikopoulos, (1990), "Nonlinear prediction in image coding with DPCM," *Electronic letters*, **26(17)**, pp. 1357-1359.

[194]. Manikopoulos, C. N. (1992). "Neural network approach to DPCM system design for image coding", *IEE Proceedings-I*, **139 (5)**, pp. 501-507.

[195]. Wenzheng Bao, Zhenhua Huang, Chang-An Yuan and De-Shuang Huang, "Pupylation Sites Prediction with Ensemble Classification Model," Int. J. Data Mining and Bioinformatics, Vol. 18, No. 2, pp 91-104, 2017.

[196]. Bao W, Wang D, Chen Y, De-Shuang Huang, "Classification of Protein Structure Classes on Flexible Neutral Tree." IEEE/ACM Transactions on Computational Biology and Bioinformatics, Volume: 14 Issue: 5,page(s): 1-12,ISSN: 1545-5963.2017.

[197]. Taubman, D., and Michael W. Marcellin. "JPEG2000 Image Compression Fundamentals, Standards and Practice." Journal of Electronic Imaging 11.2 (2013): 286-287.

[198]. Wang, Zhou, et al. "Image quality assessment: from error visibility to structural similarity." IEEE Transactions on Image Processing 13.4 (2004): 600-612.

[199]. Schwarz, Heiko, Detlev Marpe, and Thomas Wiegand. "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard." IEEE Transactions on Circuits and Systems for Video Technology 17.9 (2007): 1103-1120.

[200]. Sengupta, A, Roy D., (2018)."Intellectual Property-Based Lossless Image Compression for Camera Systems", IEEE Consumer Electronics Magazine, Volume 7, Issue 1, pp. 119-124.

[201]. Lucas, L.F.R., Rodrigues, N. M. M., Cruz, L.A.D, and de Faria, S.M.M., (2017) "Lossless Compression of Medical Images Using 3-D Predictors", IEEE Transactions on Medical Imaging, Volume 36, Issue 11, pp. 2250-2260.

[202]. Narmatha, C.; Manimegalai, P.; Manimurugan, S. (2017), "A Lossless Compression Scheme for Grayscale Medical Images Using a P2-Bit Short Technique", Journal of Medical Imaging and Health Informatics, Volume 7, Issue 6, pp. 1196-1204.

[203]. Haddas, S., Coatrieus, G., Cozic, M., and Bouslimi, S. (2017), "Joint Watermarking and Lossless JPEG-LS Compression for Medical Image Security.", Innovation and Research in BioMedical engineering, Volume 38, Issue 4, pp. 198-206.

[204]. Deigant, Y., Akshat, V., Raunak, H., Prinjal, P., and Avi, J., (2017), "A Proposed Method for Lossless Image Compression in Nano-satellite Systems", IEEE Aerospace Conference, Big Sky, MT, 04-11March.

[205]. Rusyn, B., Lysak, Y., Lukenyuk, A., Pohreliuk, L., (2016), "Lossless Image Compression in the Remote Sensing Applications", IEEE First International Conference on Data Stream Mining & Processing, Ukraine, 23-27 August.

[206]. AbuBaker, A., Eshtay, M., AkhoZahia, M., (2016), "Comparison Study of Different Lossy Compression Techniques Applied on Digital Mammogram Images", Volume 7, No. 12, pp. 149-155.

[207]. Hagag, A., Fan, X.P., Abd El-Samie, F.E., (2016), "The Effect of Lossy Compression on Feature Extraction Applied to Satellite Landsat ETM plus Images", Eighth International Conference on Digital Image Processing, Chengdu, China, 20-23 May.

[208]. Kozhemiakin, R., Abramov, S., Lukin, V., Djurovic, B., Djurovic, I., and Vozel, B., (2016), "Lossy Compression of Landsat Multispectral Images", 5th Mediterranean Conference on Embedded Computing, Montenegro, 12-15 June, pp. 104-107.

[209]. Gersho, A. (1982). "On the structure of vector quantizers," *IEEE Trans. Information theory*, **28(2)**, pp. 157-166.

[210]. Gersho, A. (1979). "Asymptotically optimal block quantization," *IEEE Trans. Information theory*, **25(4)**, pp. 373-380.

[211]. Ramamurthi, B. and A. Gersho, (1986). "Classified vector quantisation of images," *IEEE Trans. communications*, **34(11)**, pp. 1105-1115.

Abir Hussain is a Reader (Associate Professor) in Image and Signal Processing and she is the head of the Applied Computing Research Group at the faculty of Engineering and Technology at Liverpool John Moores University. She completed her PhD study at The University of Manchester, UK in 2000 with a thesis title Polynomial Neural Networks for Image and Signal Processing. She has published numerous referred research papers in conferences and Journal in the research areas of Neural Networks, Signal Prediction, Telecommunication Fraud Detection and Image Compression.
She has worked with higher order and recurrent neural networks and their applications to financial, physical, e-health and image compression techniques. She has developed with her research students a number of recurrent neural network architectures. Her research has been published in a number of high esteemed and high impact journals such as the Expert Systems with Applications, PloS ONE, Electronic Letters, Neurocomputing, and Neural Networks and Applications. She is a PhD supervisor and an external examiner for research degrees including PhD and MPhil. She is one of the initiators and chairs of the Development in e-Systems Engineering (DeSE) series, most notably illustrated by the IEEE technically sponsored DeSE International Conference Series.

Ali Al-Fayadh is an associate professor at Al-Nahrain University, Iraq. He has extensive research in the area of image compression in particular for classified image compression techniques and applications to medical images processing.

Naeem Radi is with the Al Khawarizmi University College in the UAE. His main area of research involves the design of image processing algorithms for image compression. Dr. Radi has published many journal and conference papers and reports on many aspects of this research and has also acted as session chair and on program committees for many international conferences. He serves as professional fellowship for BCS and as a professional member for IEEE.