

77230

DATA COMPRESSION TECHNIQUES

A Thesis Submitted to the
Graduate School of Natural and Applied Science of
Dokuz Eylül University
in Partial Fulfillment of the Requirements for
the Degree of Master of Science in Electrical and Electronics Engineering



by
Berna ÖNGEN

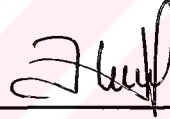
September, 1998
İZMİR

M.Sc. THESIS EXAMINATION RESULT FORM

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



Assist. Prof. Dr. Reyat Yilmaz
(Advisor)



(Committee Member)

Dr. Haldun Sarnel



Dr. Tavit Jenol

(Committee Member)

Approved by the
Graduate School of Natural and Applied Sciences



Prof. Dr. Cahit Helvacı
Director

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor Assis. Prof. Dr. Reyat Yılmaz for his valuable guidance, to my friends for their understanding and helps, for their guidance and to my family for their support.



ABSTRACT

Data compression techniques are developed to increase the efficiency of communication channels and to occupy less space in storage media. This technique helps reducing communication period, enabling more data flow and decreasing the cost of hardware for data storage. Almost in all data transfer and data storage needs, data compression techniques are extensively used. Amongst the many applications, image data compression has great importance due to its high redundant data content.

In this study, different image data compression techniques are investigated and compared with each other for their efficiency and effects on image quality. Among the many data compression techniques, the following ones are chosen to be compared with each other: Run-Length Coding, Discrete Fourier Transform, Discrete Cosine Transform, Vector Quantization, Hierarchical Finite-State Vector Quantization and Wavelet Transform techniques. The basic criteria in comparing these techniques are their compression rates versus their signal to noise ratio.

This study has revealed that at low compression rates Run Length Coding and in high compression rates Vector Quantization and Wavelet Transform are more efficient than the others. As alternative approaches, techniques which are called hybrid Wavelet Transform with Vector Quantization technique and hybrid Wavelet Transform with Hierarchical Finite-State Vector Quantization are proposed and presented. The results show that these combinations have better performance than the each algorithm individually at high compression ratios.

ÖZET

Veri Sıkıştırma teknikleri iletim bandı genişliğinin etkin bir şekilde kullanımı ve verilerin en az belleğe ihtiyaç duyacak şekilde depolanması için geliştirilmiştir. Böylece iletim süresi ve bilgileri saklamak için gerekli donanımın en aza indirilmesi sağlanabilmektedir. Veri Sıkıştırma tekniklerinin birçok uygulama alanları vardır. Bunların içinde görüntü sıkıştırma günümüzde önemi gittikçe artan bir uygulama alanıdır.

Bu tezde uygulamada en çok kullanılan görüntü sıkıştırma yöntemleri incelenerek görüntüler üzerindeki etkileri gösterilmiş ve yöntem karşılaştırmaları yapılmıştır. Run-Length kodlama, Ayırık Fourier Transform, Ayırık Kosinüs Transform, Vektör Kuantalama, Hiyerarşik Vektör Kuantalama, Wavelet Transform teknikleri ile görüntüler sıkıştırılıp, farklı sinyal gürültü oranları ve mutlak hata oranları sıkıştırma oranlarına göre karşılaştırılmıştır. Run-Length Kodlama'nın düşük sıkıştırma oranlarında daha etkili, buna rağmen Vektör Kuantalama, Hiyerarşik Vektör Kuantalama ve Ayırık Kosinüs Transform metotlarının daha yüksek sıkıştırma oranında daha iyi sonuç verdiği görülmüştür.

Yeni sıkıştırma yöntemleri olarak da, önce Wavelet Transform uygulanmış görüntüye, bir uygulamada Vektör Kuantalama metodu diğer uygulamada ise Hiyerarşik Vektör Kuantalama uygulanmıştır. Ortaya çıkan bu melez sıkıştırma tekniklerinin sonucunda elde edilen resimlerin yüksek sıkıştırma oranlarında, diğer sıkıştırma tekniklerine göre orijinal resme daha yakın olduğu görülmüştür.

CONTENTS

	Page
CONTENTS	iv
LIST OF FIGURES.....	vii
LIST OF TABLES	x

Chapter One INTRODUCTION

Introduction	1
--------------------	---

Chapter Two INTRODUCTION TO DIGITAL IMAGE ENHANCEMENT

2.1 An Image Representation	3
2.2 Sampling and Quantization	4
2.3 Spatial Resolution	5
2.4 Brightness Resolution	8
2.5 Color Resolution	10
2.6 Histogram Processing	11
2.7 Image Enhancement	13
2.8 Spatial Filtering	18
2.8.1 Low-Pass Spatial Filtering	19
2.8.2 High-Pass Spatial Filtering	22

Chapter Three

DATA COMPRESSION TECHNIQUES FOR IMAGE

3.1 Introduction to Data Compression	24
3.2 Lossless Compression	27
3.2.1 Run-Length Coding	28
3.2.1.1 The Performance Result of Run-Length Coding	29
3.3 Lossy Compression	32
3.3.1 Discrete Fourier Transform Coding	33
3.3.1.1 The Performance Result of Discrete Fourier Transform Coding	36
3.3.2 Discrete Cosine Transform Coding	39
3.3.2.1 The Performance Result of Discrete Cosine Transform Coding.....	40
3.3.3 The Application of Discrete Cosine Transform : JPEG Compression.....	44
3.3.4 Vector Quantization	46
3.3.4.1 The Vector Quantizer Design	48
3.3.4.2 The Performance Result of Vector Quantization	51
3.3.5 Hierarchical Finite-State Vector Quantization	58
3.3.5.1 Structure Map	59
3.3.5.2 Hierarchical Finite-State Coding	61
3.3.5.3 The Performance Result of Hierarchical Finite-State Vector Quantization.....	62
3.3.6 Wavelet Compression	67
3.3.6.1 Wavelet Transform	68
3.3.6.2 The Performance Result of Wavelet Transform.....	73
3.4 Comparison of the Image Compression Techniques	76

Chapter Four

THE HYBRID COMPRESSION TECHNIQUES

4.1 An Introduction to A Hybrid Wavelet Transform Combined with Vector Quantization.....	78
4.1.1 The performance result of A Hybrid Wavelet Transform and Vector Quantization Technique	79
4.2 An Introduction to Hybrid Wavelet Transform Combined with Hierarchial Finite-State Vector quantization	82
4.2.1 The performance result of Hybrid Wavelet Transform and Hierarchical Finite-State Vector Quantization Technique	83

Chapter Five

CONCLUSION

Conclusion	87
References	89

LIST OF FIGURES

	Page
Figure 2.1 The discrete pixel numbering convention	3
Figure 2.2 A 320x240 digital image showing the location of pixel (35,99)	4
Figure 2.3 An image is converted from a continuous-tone from to a digital form	5
Figure 2.4 The effect of spatial resolution on a sample image	7
Figure 2.5 The effect of brightness resolution on a sample image	9
Figure 2.6 Histograms corresponding to four basic image types	12
Figure 2.7 The “Lightnou” image with 256 gray-level and 320x240 pixels	12
Figure 2.8 The histogram of the “Lightnou” image	13
Figure 2.9 A kind of transform function for image transformation	14
Figure 2.10 The complement function, original image and brightness complement of the image	14
Figure 2.11 The histogram slide function, original image and the image after histogram sliding operation	16
Figure 2.12 The histogram stretch function, original image and the image after histogram stretching operation	17
Figure 2.13 The original images and low-pass filtered images by using 3x3 and 5x5 masks	20
Figure 2.14 The original images and high-pass filtered images by using 3x3mask	22
Figure 3.1 A General Communication System Model, Source Encoder and Source Decoder	25
Figure 3.2 The “Lightnou” image compressed and decompressed by Run- Length coding	30
Figure 3.3 The “Peppers” image compressed and decompressed by Run- Length coding	31

Figure 3.4	The path of Zig-Zag Sequence	35
Figure 3.5	The compression and decompression scheme of Discrete Fourier Transform Coding.	36
Figure 3.6	Compressed and decompressed images by DFT Coding.....	37
Figure 3.7	The compression and decompression scheme of Discrete Cosine Transform Coding.....	39
Figure 3.8	Compressed and decompressed sample images by DCT	41
Figure 3.9	The Block Diagram of JPEG Compression.....	45
Figure 3.10	Encoding and Decoding Scheme of Vector Quantization.....	47
Figure 3.11	The typical tree of codebook structure of Vector Quantization .	50
Figure 3.12	The “Lenna” images by decompressed Vector Quantization in different codebook sizes with 1x2 vector size.....	51
Figure 3.13	The “Lightnou” images by decompressed Vector Quantization in different codebook sizes with 1x2 vector size.....	53
Figure 3.14	The “Building” images by decompressed Vector Quantization in various codebook sizes with 2x2 vector size.....	55
Figure 3.15	The compressed and decompressed “Peppers” image at the same codebook and various codevectors.....	57
Figure 3.16	A general compression scheme of the Hierarchical Finite-State Vector Quantization.....	59
Figure 3.17	A Structure Tree.....	60
Figure 3.18	A Structure Map.....	60
Figure 3.19	The “Tiffany” image and its Structure Map	60
Figure 3.20	The images compressed and decompressed by using HFSVQ at different threshold values and the same codebook design	63
Figure 3.21	The images compressed and decompressed by using HFSVQ at different codebook sizes and the same threshold values.....	64
Figure 3.22	Different kind of continuous wavelet functions.....	68
Figure 3.23	A Haar Wavelet function.....	69
Figure 3.24	Scaled and translated Haar Wavelet functions	70
Figure 3.25	Fast implementation of the wavelet transform.....	72
Figure 3.26	The Lenna image Wavelet decomposition.....	73

Figure 3.27	The decompressed “Madrill” images with Wavelet compression	74
Figure 3.28	The “Room” images compressed and decompressed by Wavelet compression technique	75
Figure 4.1	The Block Diagram of Hybrid Wavelet Transform and Vector Quantization.....	78
Figure 4.2	The “Lenna” compressed images by hybrid Wavelet compression and vector quantization at different codevector size	80
Figure 4.3	The “Peppers” compressed images by hybrid wavelet compression and vector quantization at different codevector size and codebook size	81
Figure 4.4	A general compression scheme of the Hybrid Wavelet Compression and Hierarchical Finite-State Vector Quantization.....	83
Figure 4.5	The “Peppers” image compressed a hybrid wavelet compression and hierarchical finite-state vector quantization method with different configurations	84
Figure 4.6	The “Tiffany” image compressed by using hybrid wavelet compression with hierarchical finite-state vector quantization method by using various configurations.....	85

LIST OF TABLES

	Page
Table 2.1 3x3 Kernel Convolution Coefficients.	19
Table 2.2 Low-pass filter masks of various sizes	20
Table 2.3 High-pass Filter Mask	22
Table 3.1 PSNR results of different images compressed Run-Length Coding at different Threshold.....	29
Table 3.2 PSNR and MSE values for different Threshold levels	32
Table 3.3 The pixel values of the first 8x8 block of “Peppers” image and 64 DFT coefficients of this image.....	38
Table 3.4 Peak signal-to-noise ratio (PSNR) versus compression ratio (bpp) for various images compressed by Discrete Fourier Transform coding.....	38
Table 3.5 The pixel values of the first 8x8 block of “Palm1” image and 64 DCT coefficients of this image.....	43
Table 3.6 The comparison of the Normal Order (N-O) with Zig-Zag Order	44
Table 3.7 Peak Signal-to-noise ratio (PSNR) versus compression ratio (bpp) for DCT compression.....	44
Table 3.8 The codebook of the “Lenna” image shown in Figure 3.12(d)...	54
Table 3.9 The codebook of the “Lenna” image shown in Figure 3.10 (f)...	54
Table 3.10 The peak Signal-to-noise ratio (dB) versus compression ratio (bpp) for Vector Quantization with 1x2 codevector.....	54
Table 3.11 Peak signal-to-noise ratio (PSNR) versus compression ratio (bpp) for Vector Quantization with 2x2 codevector	56
Table 3.12 The codebook of the “Building” image shown in Figure 3.14(d).....	56

Table 3.13	Peak signal-to-noise ratio (PSNR) versus compression ratio (bpp) for Vector Quantization with 4x4 codevector.....	56
Table 3.14	Various codebook and threshold values for each Layer in HFSVQ compression	62
Table 3.15	PSNR values versus compression ratio at different threshold values for various images HFSVQ	65
Table 3.16	PSNR values versus compression ratio at different codebook size and same threshold values for various images for HFSVQ.	66
Table 3.17	PSNR versus compression ratio for images compressed with Wavelet compression	74
Table 3.18	Comparison of Discrete Fourier Transform Coding and Discrete Cosine Transform Coding at same compression ratios.	76
Table 3.19	Comparison of vector quantization with 1x2 codevector and 2x2 codevector at same compression ratio.....	77
Table 4.1	PSNR values versus compression ratio at different codevector and 32 codebook for wavelet compression with VQ.....	82
Table 4.2	PSNR values versus compression ratio at different codevector and 64 codebook for wavelet compression with VQ	82
Table 4.3	PSNR values versus compression ratio at different codebook sizes and same threshold for wavelet compression with HFSVQ.....	86

CHAPTER ONE

INTRODUCTION

Data Compression seeks to reduce the number of bits used to store or transmit information. By using compression techniques, the analog data, which is at very high rate, is converted to relatively low rate discrete data for communication over a digital communication link or storage in a digital memory. The data compression techniques have received increased attention, because recently digital communication and secure communication have become increasingly important. For efficiency and economical reasons, it is necessary to reduce the transmission rate and to limit the amount of data to be stored.

Nowadays, data compression has many applications. One of the most popular applications is image compression. The image data compression techniques are one of the major parts of image processing. These techniques are useful in television, remote sensing via satellite, aircraft, radar, computer communications, document processing, video archiving and transmission.

For example, the Meteosat radiometer produces one image slot (37.5 Mbytes) every half an hour. The required storage capacity for daily image is then 1.8Gbytes[11]. The use of compression methods yielding compression ratios of 8 or higher significantly reduces the storage space and the transmission time of these images. In the specific case of 8 to 1 compression ratio, the gain in efficiency and the economical profit are already obvious. The necessary storage space and the transmission time will indeed be divided by a factor 8 for the same amount of data, and the communication channel capacity will be multiplied by the same factor for the given transmission time and communication line.

Data compression techniques for image can be divided into two major families: lossless and lossy[3]. Lossy data compression on an image accepts a certain loss of accuracy in exchange for greatly increased compression. Lossy data compression on an image is more effective when applied to a graphics image and digitized voice. Most compression techniques can be adjusted to different quality levels, gaining higher accuracy in exchange for less effective compression. Lossless compression consists of those techniques guaranteed to generate an exact duplicate of the input data stream after decompressing. This is the type of compression used when storing database records, spreadsheets or word processing files.

A variety of lossless and lossy compression schemes have been developed nowadays. In the second chapter, firstly, image properties such as histogram, image enhancement techniques and spatial filtering are given. In the third chapter, different lossless and lossy compression techniques are examined. In this thesis, Run-Length Coding [4] as a sample of lossless compression, Discrete Fourier Transform [8], Discrete Cosine Transform[15], Vector Quantization[10], Hierarchical Finite-State Vector Quantization [23] and Wavelet Compression[17] which is very popular in recent years are examined. In the fourth chapter, the hybrid compression techniques are presented.

Discrete Fourier Transform and Discrete Cosine Transform are usually used as a first step for other compression techniques such as JPEG [15]. Wavelet Compression technique is an alternative method which is preferred in recent years to transform coding. In this thesis, hybrid compression techniques are examined as alternative compression methods. The performance results are presented and compared with other compression techniques. The hybrid compression techniques are implemented by combining wavelet compression with vector quantization and wavelet compression with hierarchical Finite-State Vector Quantization. All compression algorithms are written in "C" for windows programming language. The compression results are presented and the results are compared to each other according to their compression ratio, signal to noise ratio and mean square error for different images. And all the performance results are given in the conclusion.

CHAPTER TWO

INTRODUCTION TO DIGITAL IMAGE ENHANCEMENT

2.1 An Image Representation

The term image is a two-dimensional light intensity function, $f(x,y)$. This function defines the value of the brightness of the image at any point x and y spatial coordinates.

A digital image is a function $f(x,y)$ that has been discretized both in spatial coordinates and brightness. A digital image can be considered as a matrix of which row and column indices identify a point in a image. Each element of this digital array is called as a pixel.

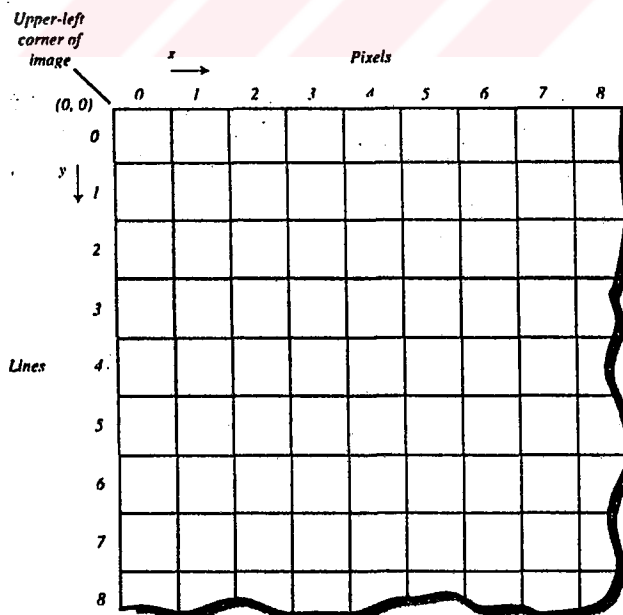


Figure 2.1 The discrete pixel numbering convention.



Figure 2.2 A 320x240 digital image showing the location of pixel (35,99).

The picture shown in Figure 2.2 includes total 76,800 pixels.

Images can have either digital or analog representation. In the digital representation of gray-level images, the image is presented as a two-dimensional array of numbers. Each gray-level is presented by 8 bits (1 byte), then the gray level may have any of 2^8 or 256 possible values. These levels are assigned integer values ranging from 0 to 255, with 0 representing the darkest intensity level and with 255 representing the brightest intensity level. In a colored image, the representation is similar except the number represents three primary colors: red, green and blue at each location of the matrix. For a 24 bit color representation, the number is divided into three 8 bit segments. Each segment represents the intensity of the colour primaries.

In the analog form, images are usually presented as horizontal raster lines. Each line is basically an analog signal carrying the continuous variations of light intensities along a horizontal line in the original scene. For example; images on television sets are displayed through raster scanning.

2.2 Sampling and Quantization

A digital image is composed of discrete points of gray-level tone, rather than continuously varying tones. To obtain a digital image from a continuous-tone image,

it must be divided up into individual points of brightness. Additionally, each point of brightness must be described by a digital data value. The processes of breaking up a continuous tone image and determining digital brightness values are referred to as sampling and quantization that are illustrated in Figure 2.3.

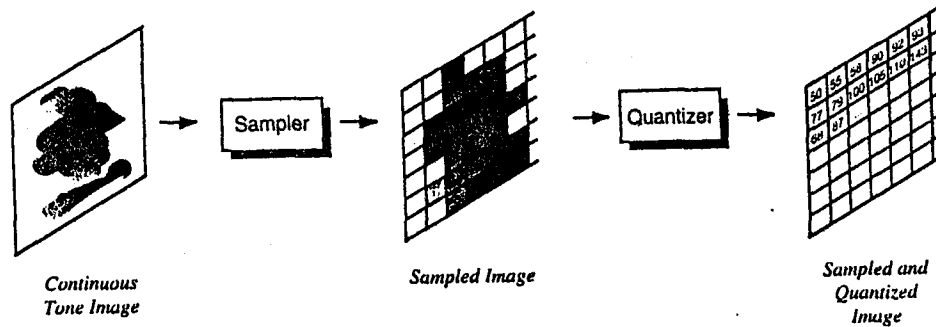


Figure 2.3 An image is converted from a continuous-tone to a digital form.

As a result of this process, the obtained matrix is shown below:

$$\begin{vmatrix}
 f(0,0) & f(0,1) & f(0,2) & \dots & f(0,M-1) & f(0,M) \\
 f(1,0) & f(1,1) & f(1,2) & \dots & f(1,M-1) & f(1,M) \\
 \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\
 f(N-1,0) & f(N-1,1) & f(N-1,2) & \dots & f(N-1,M-1) & f(N-1,M) \\
 f(N,0) & f(N,1) & f(N,2) & \dots & f(N,M-1) & f(N,M)
 \end{vmatrix}$$

The quality of the digital image is directly related to the number of the pixels, M and N . These aspects are known as image resolution. Image resolution is the capability of the digital image to resolve the elements of the original scene.

For digital images, the resolution characteristics can be broken into two primary groups as spatial and brightness resolution (or color resolution for color images).

2.3 Spatial Resolution

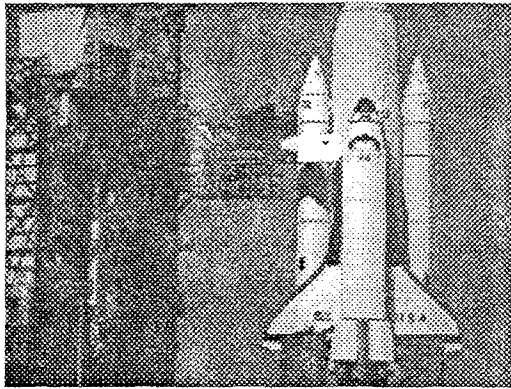
Spatial resolution is described as how many pixels comprise a digital image. The more pixels in the image, the greater its spatial resolution.

The continuous-tone image, like a photograph or video camera image, is broken into enough discrete pixels so that the digitized image contains all the information of the original. This means that the displayed digital image would look identical to the original continuous-tone image to an observer. This criterion holds true, when the digital image is intended for use by human observer.

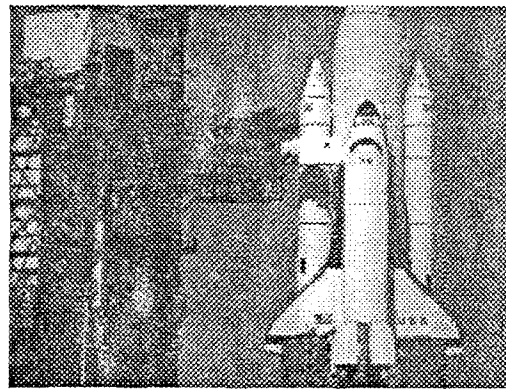
The concept of spatial frequency explains how finely the image is sampled. The required sampling rate must be determined so that the digital image adequately resolves all the spatial details of the original continuous-tone image. Mathematically, the theorem is classical sampling theorem that is used to represent fully the spatial details of an original continuous-tone image. The image is sampled at a rate at least twice as fast as the highest spatial frequency contained in it. This guarantees that both the dark and light portions of the detail are sampled. The highest spatial frequencies that can be contained in the resulting digital image will not exceed one half of the sampling rate. This frequency is named as Nyquist rate.

If sampling occurs at a slower rate than required by the sampling theorem, high spatial frequency details will be missed in the digital image. However, the digital image will appear to have less spatial resolution than the original image. Because, in this digital image, there are not enough pixels to represent all of the spatial details of the original image. On the other hand; if sampling occurs at a rate higher than that required by the sampling theorem, extra pixels will be created.

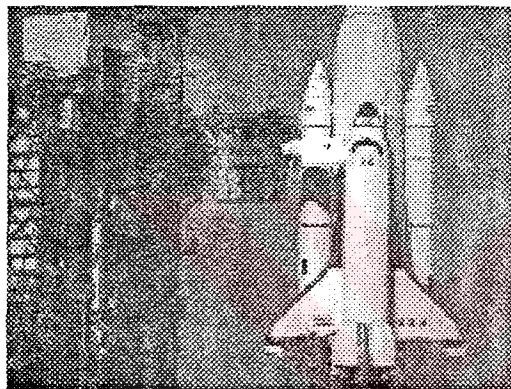
In real-life systems, the sampling rate of a particular image acquisition system is generally fixed. It is not adjusted by the spatial frequency of image. The camera and digitizer are selected to meet the minimum spatial-resolution requirements of the application.



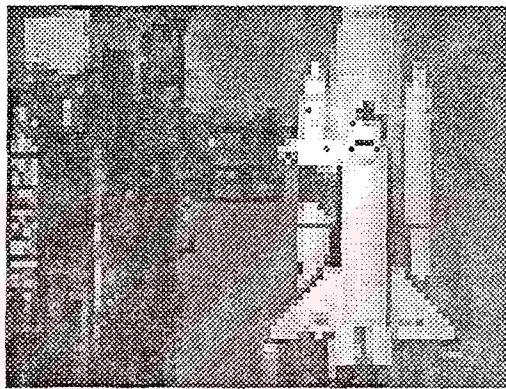
(a) An image with spatial resolution of 640 x 480 pixels.



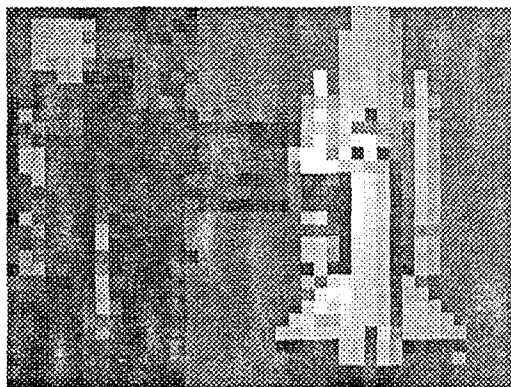
(b) Resolution reduced to $\frac{1}{2}$ in both x and y dimensions: 320 x 240 pixels.



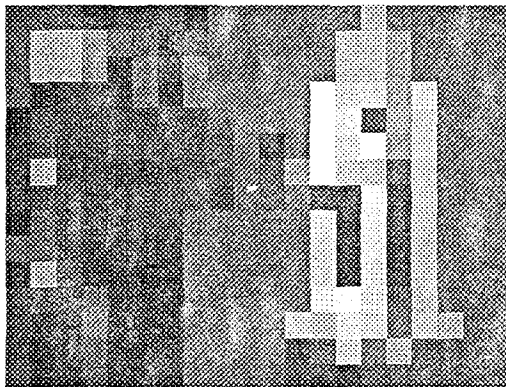
(c) Resolution reduced to $\frac{1}{4}$: 160 x 120 pixels.



(d) Resolution reduced to $\frac{1}{8}$: 80 x 60 pixels.



(e) Resolution reduced to $\frac{1}{16}$: 40 x 30 pixels.



(f) Resolution reduced to $\frac{1}{32}$: 20 x 15 pixels.

Figure 2.4 The effect of spatial resolution on a sample image.

Figure 2.4 shows that the effect of an image sampled at different spatial resolution. As the resolution is increased, the structure of the image becomes more apparent. The image with the least spatial resolution is the one with a resolution of 40x30. This image is virtually unintelligible.

The spatial resolution requirements of an image are based on its intended application. Digital images for use in television production applications require a spatial resolution of about 640x480. Moreover, 320x240 image appears to be virtually identical to the 640x480 version as shown in Figure 2.4 (a) and (b). So, this version can be selected easily for some applications.

As the spatial resolution increases, the number of pixels climbs exponentially. For 640x480 version, the total number of pixels in the image is 307,200. But, for the 160x120 version, the number of the pixels is 19,200. It is important to consider the spatial resolution required by a given application. Selecting the minimum necessary resolution can significantly reduce digital image storage and processing time.

2.4 Brightness Resolution

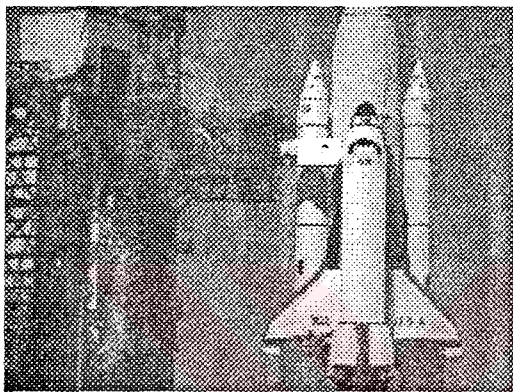
Every pixel in a digital image represents the intensity of the original image at the spatial location where it was sampled. The digital brightness of pixel represents the intensity of the original image.

Intensity refers to the magnitude of light energy actually reflected from a physical scene. The term brightness refers to the measured intensity after it is acquired, sampled, quantized, displayed and observed. The brightness of a pixel accounts for all the effects induced by the entire imaging system.

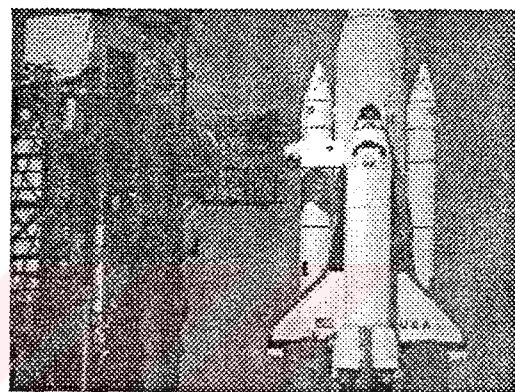
The quantization process converts the continuous-tone intensity to a digital brightness value at the sample points. The accuracy of the digital value is directly dependent upon how many bits are used, the brightness can be converted to one of

eight gray-levels. In this case, gray-level “0” represents black, gray level “1” represents white and gray levels “1” through “6” represent the ascending gray tones between black and white. The eight gray levels comprise what is called the gray-scale, or in this case, the 3-bit gray scale.

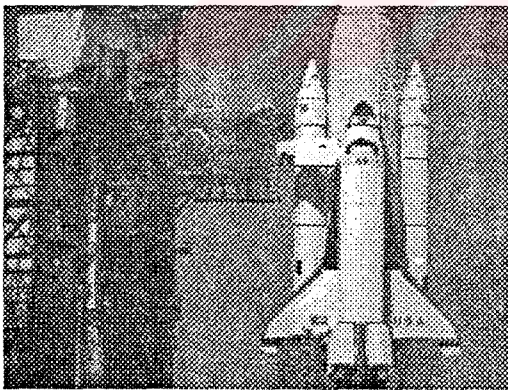
With a 4-bit brightness value, every pixel’s brightness is represented by one of 16-gray levels. And 8-bit brightness value yields a 256 level gray-scale range. Every additional bit used to represent the brightness doubles the range of the gray-scale.



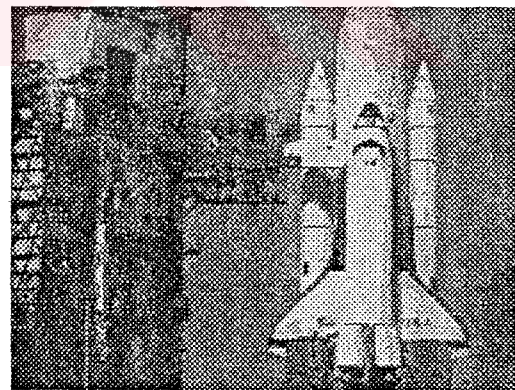
(a) Brightness resolution of 8 bits, 256 gray level.



(b) Brightness resolution of 7 bits, 128 gray level.

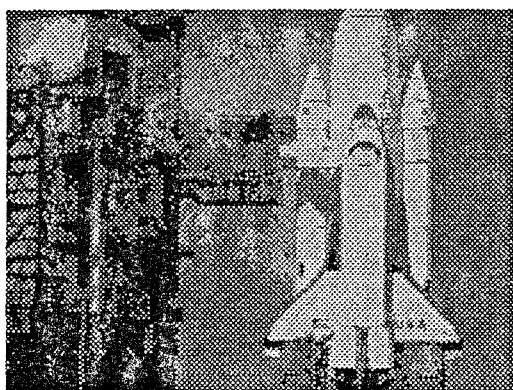


(c) Brightness resolution of 5 bits, 32 gray level.

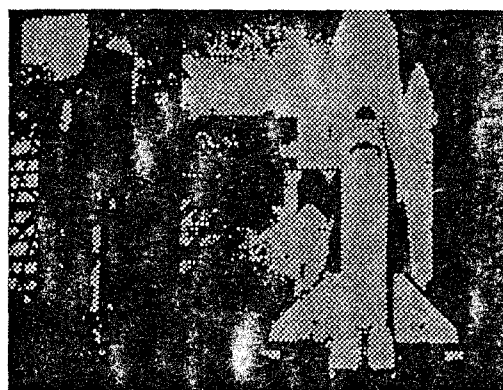


(d) Brightness resolution of 3 bits, 8 gray level.

Figure 2.5 The effect of brightness resolution on a sample image.



(e) Brightness resolution of 2 bits, 4 gray level.



(f) Brightness resolution of 1 bits, 2 gray level.

Figure 2.5 (cont.)

Figure 2.5 shows an image quantized to various brightness resolution. The image quantized to eight bits of brightness resolution appears very natural and continuous. As the brightness resolution decreases, the image appears coarser and mechanical. This effect is known as brightness contouring. Contouring occurs when there are not enough gray levels to represent the actual brightness in the original image adequately. Brightness contouring is the effect of insufficient brightness resolution.

The number of bits used in quantizing an image depends on how the image will be used. 8 bit quantization is the most common way used in many applications and generally sufficient.

2.5 Color Resolution

For color images, the same concepts of sampling, quantization, spatial and brightness resolution are valid. But, instead of a single brightness value, color digital images have pixels that are generally quantized using three brightness components. In displaying color, three independent color emitters are used. These emitters have a unique spectral band of light to generate all colors in the spectrum.

At a color video display screen, if it is a cathode ray tube (CRT), liquid crystal display (LCD) or another type, individual dots of solid colors will be noticed. These dots emit light in the colors of red, green and blue.

All the colors in the spectrum can be created with the primary colors of red, green and blue (RGB). This is called the additive of primary colors that are emitting light.

Digital image processing must handle the red, green and blue components of each pixel of image. Generally, each component must be quantized at a resolution rate equivalent to the brightness resolution used in a gray-scale image.

2.6 Histogram Processing

The brightness characteristics of an image can be displayed with a tool known as the brightness histogram. In general terms, a histogram is a distribution graph of a set of numbers. The brightness histogram is a distribution graph of the gray levels of pixels within a digital image. It provides a graphical representation of how many pixels within an image fall into the various gray level steps. A histogram appears as a graph with brightness on the horizontal axis from 0 to gray scale, and number of pixels on the vertical axis.

The histogram gives a convenient and easy way to read representation of the concentration of pixels versus brightness in an image. Using this graph, it is seen that an image is dark or light and in high or low contrast. Figure 2.6 show histogram graphs for different images.

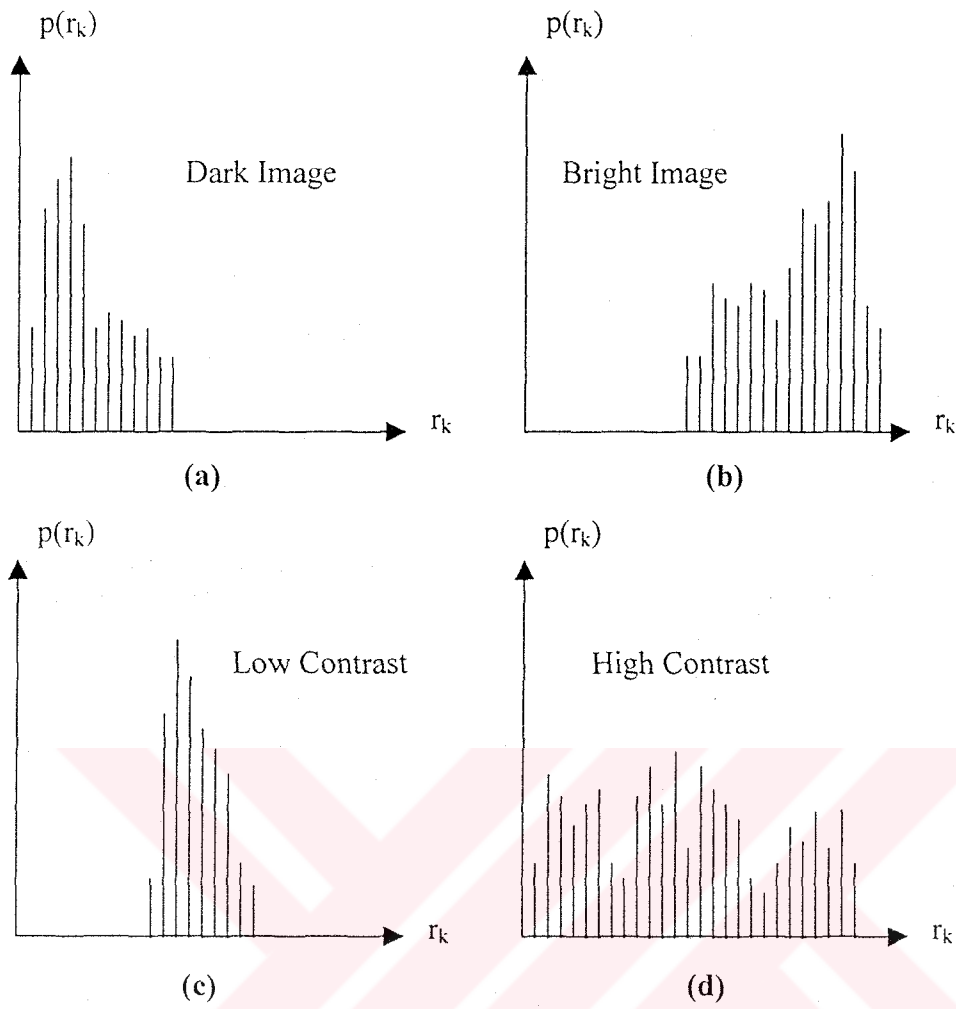


Figure 2.6 Histograms corresponding to four basic image types.

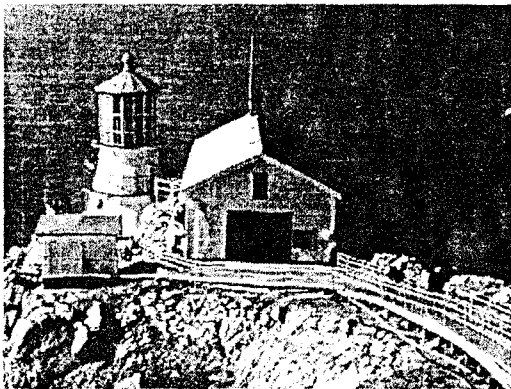


Figure 2.7 The "Lightnou" image with 256 gray level and 320x240 pixels.

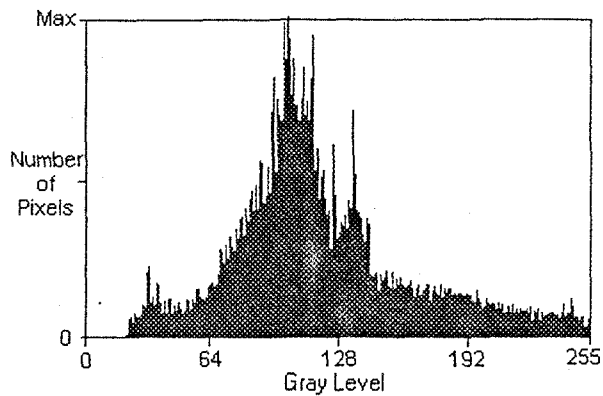


Figure 2.8 The histogram of the “Lightnou” image.

Figure 2.7 and Figure 2.8 show the original image and their histogram, respectively. The histogram for color images are computed and displayed for each color (blue, red and green) components.

2.7 Image Enhancement

The principal objective of enhancement techniques is to process an image so that the result is more suitable than the original image for a specific application.

The approaches for image enhancement fall into two broad categories: spatial domain methods and frequency domain methods. The spatial domain refers to the image plane itself, and is based on direct manipulation of pixels in an image. Frequency domain processing techniques are based on modifying the Fourier Transform of an image.

Enhancement techniques for spatial domain are based on point processing which modifies the gray level of a pixel. The input image is converted to the output image according to mathematical or logical relationship.

For example, negatives of digital images are useful in numerous applications, such as displaying medical images and photographing a screen with monochrome positive film. The negative of digital image is obtained by using the transform function.

$$s = T(r) \quad (2.1)$$

This transfer function expressed in Equation 2.1 is shown in Figure 2.9, where L is the number of the gray levels.

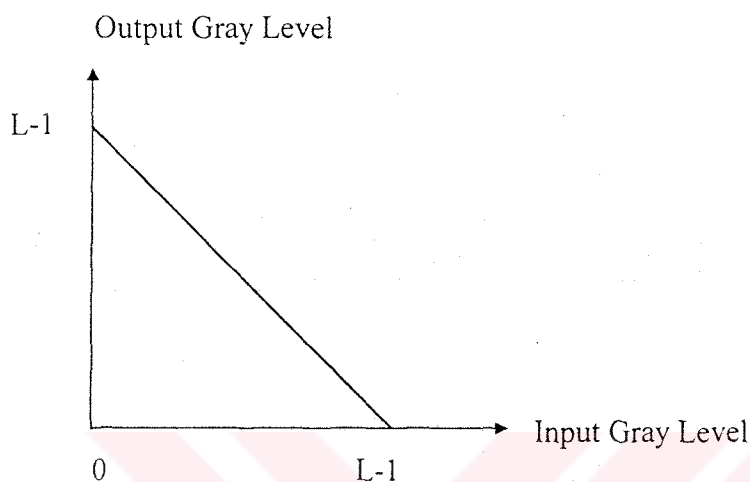
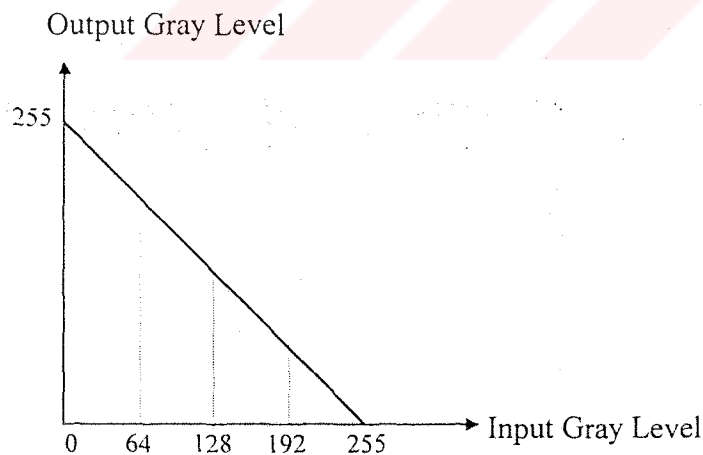


Figure 2.9 A kind of transform function for image transformation.

The original image with 256 gray level, the gray level transformation function and negative of the image are shown in Figure 2.10.



(a) The complement operation mapping function for 256 gray level.

Figure 2.10 The complement function, original image and brightness complement of the image.



(b) The “Room” image with 256 gray level and 256x256 pixels.



(c) The brightness complement of “Room” image.

Figure 2.10 (cont.)

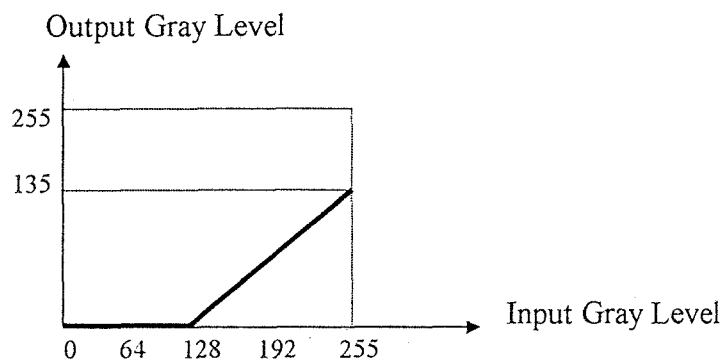
Histogram sliding and histogram stretching operations are also examples for histogram processing. These operations redistribute the brightness in an image and enhance contrast characteristics of images.

By looking at a histogram of an image, the contrast deficiencies can be determined. The most frequently encountered types of histograms are those showing characteristics such as low-contrast/low dynamic range, high contrast/high dynamic range, well-balanced contrast/high dynamic range. An image of poor contrast quality look can be made considerably better by using sliding and stretching operations.

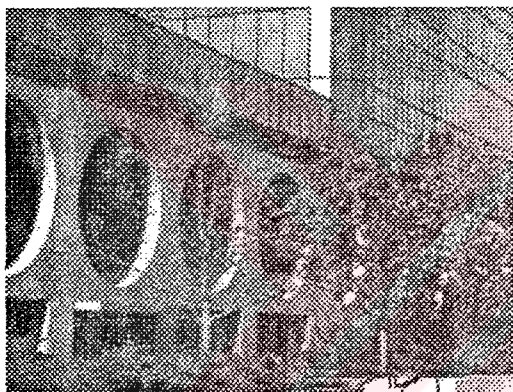
The histogram sliding operation is simply addition or subtraction of a constant brightness value to all pixels in the image. The sliding operation is sometimes referred to addition of offset to the image brightness.

The histogram stretching operation is the multiplication by or division of all pixels by a constant value. The stretching operation is referred to addition of a gain to the image brightness value.

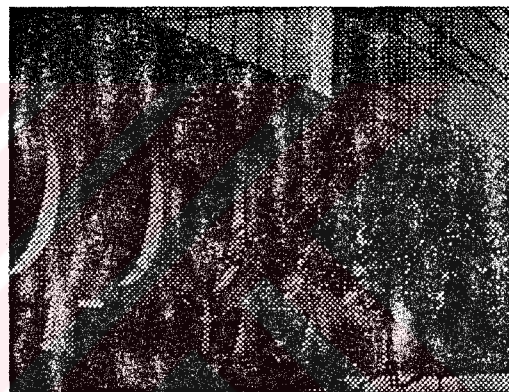
Figure 2.11 shows the original image, the function of the histogram slide mapping and the image after histogram sliding, respectively.



(a) The histogram slide function for 256 gray-level image.



(b) The “Building” image with 256 gray-level and 320x240 pixels.



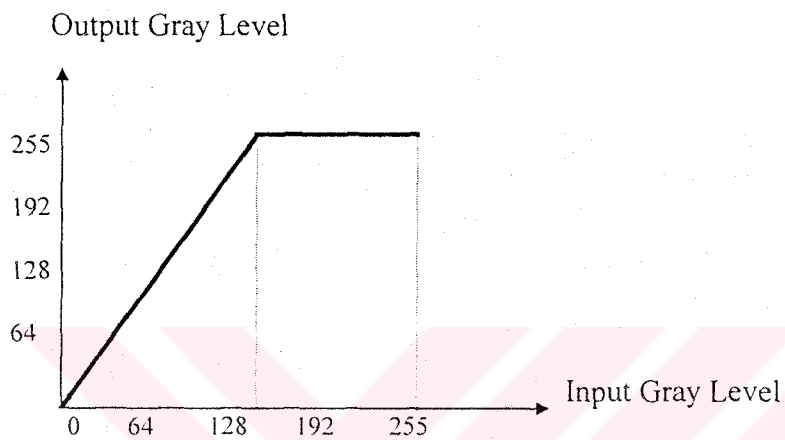
(c) The “Building” image after histogram sliding.

Figure 2.11 The histogram slide function, original image and the image after histogram sliding operation.

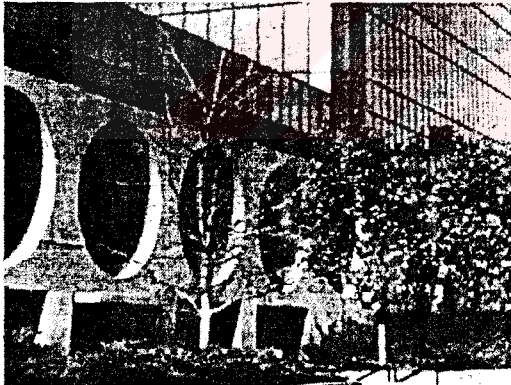
The image that is shown in Figure 2.11(b) is a low contrast image. The histogram of the image ranges from 120 gray level to 250 gray level. By subtracting the brightness value of 120 from all pixels by using a point process, more contrast image than the original image can be obtained. In this study, histogram slide function shown in Figure 2.11(a) was applied to the image and then, as a result of sliding operation, the more contrast image was obtained as shown in Figure 2.11(c).

For the stretching process, the each pixels of the image is multiplied by a specific value. For this example, the gain value is selected as shown in Figure 2.12 (a).

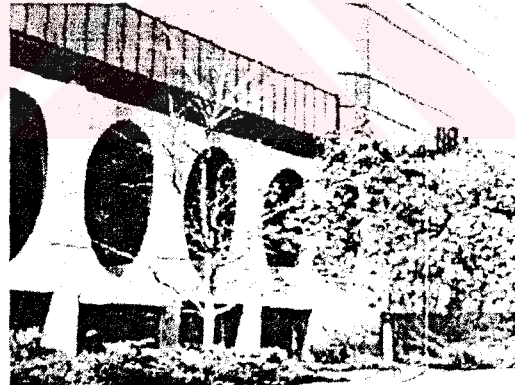
As a result of this operation, the image appears considerably better contrast characteristics. In this study, histogram stretching function shown in Figure 2.12(a) was applied to the image and then, as a result of sliding operation, better contrast image was obtained as shown in Figure 2.11(c).



(a) The histogram stretch function for 256 gray-level.



(b) The "Building" image with 256 gray-level and 320x240 pixels.



(c) The "Building" image after histogram stretch.

Figure 2.12 The histogram stretch function, original image and the image after histogram stretching operation.

2.8 Spatial Filtering

Pixel point processing provides image combinations, image gray-scale alterations and corrections, all of which are important digital image processing tools. However, point operations can not provide the ability to alter spatial scene details with an image. Because, point processes act pixel by pixel by mapping a single corresponding output pixel. The point process does not consider neighboring input pixels in its processing.

Pixel group processing operates on a group of input pixels surrounding a centre pixel. The adjoining pixels provide valuable information about brightness trends in the area. Using these brightness trends, the spatial filtering has been done.

An image is composed of basic frequency components ranging from low frequencies to high frequencies. Where rapid brightness transitions are prevalent, there are high spatial frequencies. Slowly changing brightness transitions represent low spatial frequencies. The highest frequencies in an image are found wherever sharp edges or points are present like a transition from white to black within a one or two pixel distance.

An image can be filtered to accentuate or remove a band of spatial frequencies, such as the high frequencies or low frequencies. These digital image processing operations is known as spatial filtering operations.

The spatial convolution process uses a weighted average of the input pixel and its immediate neighbors to calculate the output pixel brightness value. The group of pixels used in the weighted average calculation is called the Kernel. Kernel dimension is generally square with an odd number of mask values in each dimension. The Kernel can have the dimensions of 3x3 and 5x5 etc. In practice, 3x3 and 5x5 Kernels are used in most spatial filtering operations.

The mechanics of spatial filtering operation is straightforward. In carrying out 3x3 Kernel convolution, 9 convolution coefficients are defined as seen below:

Table 2.1 3x3 Kernel Convolution Coefficients.

a	b	c
d	e	f
g	h	i

This array of coefficients is called the convolution mask. Each pixel in the input is evaluated with its 8 neighbors, using this mask to produce an output pixel value. The pixel and its 8 neighbors are multiplied by their respective convolution coefficients and all of them are summed. The results are placed at the same centre pixel location in the output image. This process occurs pixel by pixel for each pixel in the input image. The equation for the spatial convolution process is

$$\begin{aligned}
 O(x, y) = & aI(x-1, y-1) + bI(x, y-1) + cI(x+1, y-1) + \\
 & dI(x-1, y) + eI(x, y) + fI(x+1, y) + gI(x-1, y+1) + \\
 & hI(x, y+1) + iI(x+1, y+1)
 \end{aligned} \quad (2.2)$$

where implied that each input pixel is processed through the equation.

2.8.1 Low-pass Spatial Filtering

A spatial low-pass filter allows to pass the low frequency components of an image. High frequency components are attenuated and are virtually absent in the output image. A common low-pass convolution mask is composed of all nine coefficients having the value of $1/9$. The 3x3 and 5x5 mask for low-pass filter are shown below.

Table 2.2 Low-pass filter masks of various sizes.

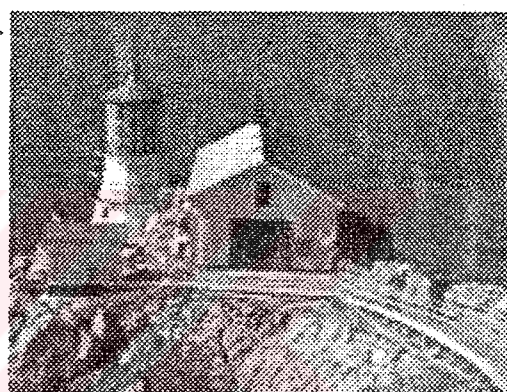
$1/9$	1	1	1
	1	1	1
	1	1	1
	1	1	1

$1/25$	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	1
	1	1	1	1	1

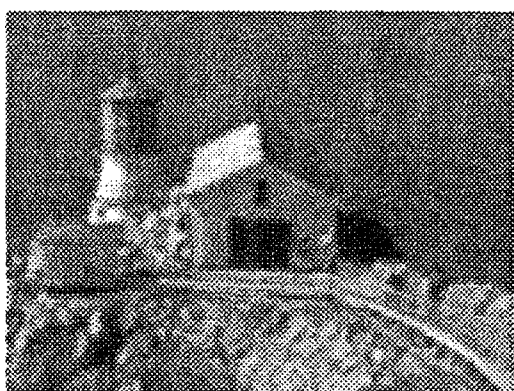
In these study, by using 3x3 and 5x5 low-pass filter mask, the sample images are low-filtered and presented in Figure 2.13.



(a) The "Lightnou" image with 256 gray-level and 320x240 pixels.



(b) Low-pass filtered "Lightnou" image by using 3x3 mask.



(c) Low-pass filtered "Lightnou" image by using 5x5 mask.

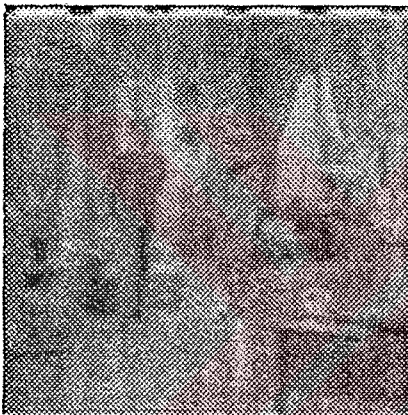
Figure 2.13 The original images and low-pass filtered images by using 3x3 and 5x5 masks.



(d) The “Room” image with 256 gray-level and 256x256 pixels.



(e) Low-pass filtered “Room” image by using 3x3 mask.



(f) Low-pass filtered “Room” image by using 5x5 mask.

Figure 2.13 (cont.)

The visual effect of a low-pass filter is image blurring. This is because the sharp brightness transitions become attenuated to small brightness transitions. If 5x5 mask is used for low-pass filtering, the image becomes more blurred according to 3x3 mask usage.

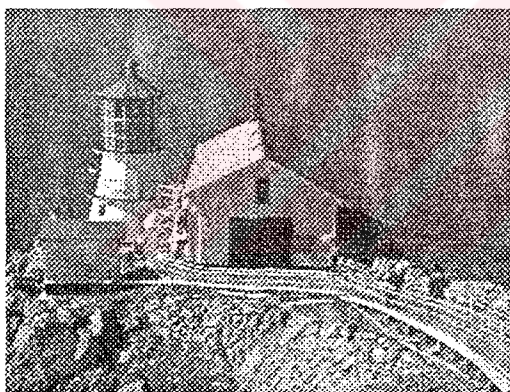
2.8.2.High-pass Spatial Filter

The high-pass filter has the opposite effect of the low-pass filter. It attenuates low frequency spatial components while leaving high frequency components untouched. A common high-pass mask is shown below.

Table 2.3 High-pass Filter Mask.

-1	-1	-1
-1	8	-1
-1	-1	-1

In these study, by using 3x3 high-pass filter mask, the sample images are high-filtered and presented in Figure 2.14.

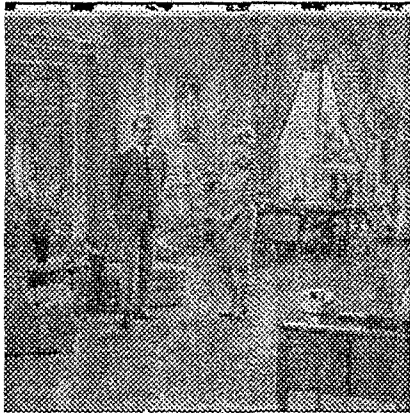


(a) The “Lightnou” image with 256 gray-level and 320x240 pixels.

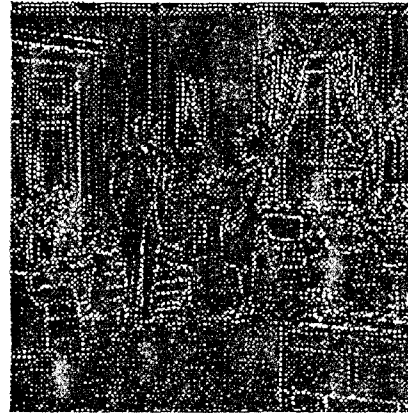


(b)High-pass filtered “Lightnou” image by using 3x3 mask.

Figure 2.14 The original images and high-pass filtered images by using 3x3 mask.



(c) The “Room” image with 256 gray-level and 256x256 pixels.



(d) High-pass filtered “Room” image by using 3x3 mask.

Figure 2.14 (cont.)

CHAPTER THREE

DATA COMPRESSION TECHNIQUES FOR IMAGE

3.1 Introduction to Data Compression

Image compression and decompression operations are used to reduce the data content of a digital image. The goal of these operations is to represent an image with some required quality level. Image compression operations seek to extract essential information from an image, so that the image can be accurately reconstructed. Nonessential information is discarded.

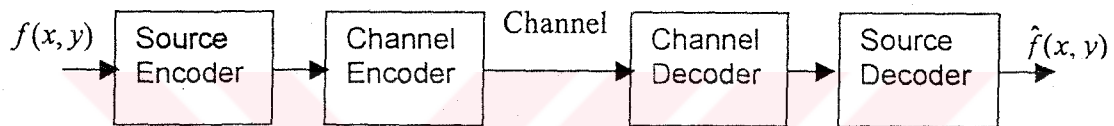
Image storage refers to the electronic storage of an image's data typically on magnetic or other permanent media. Image transmit refers to the electronic transfer of data of an image over a data link.

If the amount of the necessary data to represent an image can be reduced, then the amount of time to transmit can be reduced. In addition, the amount of storage space is reduced. For instance, the compression of image data by a ratio of ten to one will allow the transmission of ten compressed images in the same time as required for one uncompressed image.

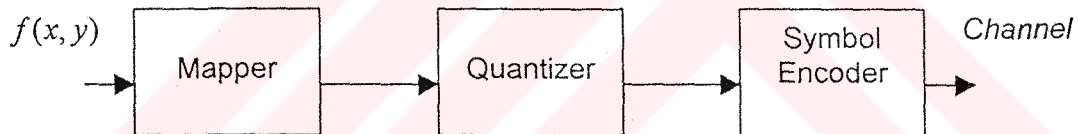
Image compression schemes are divided into two general groups, lossless compression and lossy compression. Lossless image compression preserves the exact

data content of the original image while lossy image compression preserves some specified level of image quality.

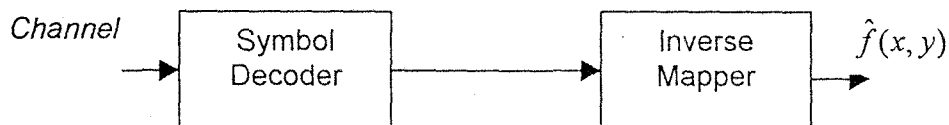
All image data compression schemes involve both compression and decompression operation which is the inverse operation of compression. The compression operation converts the original data into a compressed image data form. The decompression operation converts the compressed image data back to its original uncompressed form. The general communication system model and the source encoder and decoder model in the system are shown in Figure 3.1. The $f(x, y)$ function refers to input data and $\hat{f}(x, y)$ function refers to output data in Figure 3.1.



(a) A General Communication System Model



(b) Source Encoder



(c) Source Decoder

Figure 3.1 A General Communication System Model, Source Encoder and Source Decoder.

Image compression and decompression operations are often called image coding operations, because the process uses data coding methods to represent an image in a new form.

The amount of compression operation is calculated by dividing the data size of the original image by the data size of the compressed image. The result is called as compression ratio. If the value of compression ratio in bit/pixel is small, the high compression ratio is obtained. And, the higher the compression ratio, the smaller the compressed image size has become.

The compression ratio, C_R is

$$C_R = \frac{n_1}{n_2} \text{ pixel / pixel } \text{ or } C_R = \frac{n_2}{n_1} \text{ bit / pixel (bpp)} \quad (3.1)$$

where n_1 is the data size of the original image and n_2 is the data size of the compressed image. The relative data redundancy can be defined as

$$R_D = 1 - \frac{1}{C_R} (\text{pixel / pixel}) \quad (3.2)$$

The another goal of the data compression is also to reduce data redundancy in an image. A practical compression ratio, such as 10 (or 10:1) means that the input data set has 10 information carrying units for every 1 unit in compressed output data set. The corresponding data redundancy of 0.9 implies that 90 percent of the data in the input set is redundant.

In this thesis, the compression results are calculated by using mean square error (MSE) and peak signal-to-noise ratio (PSNR) criteria. These criteria are given in Equation 3.3. and 3.4 [16]. The performance of the compression techniques is obtained according to compression ratio (bpp) and peak signal-to-noise ratio (dB).

The mean square error is

$$\text{MSE} = \frac{1}{M * N} \sum_{x=0}^M \sum_{y=0}^N (f(x,y) - \hat{f}(x,y))^2 \quad (3.3)$$

The peak signal-to-noise ratio is

$$\text{PSNR} = 10 * \log_{10} \left[\frac{(\text{peak} - \text{signal} - \text{value})^2}{\text{MSE}} \right] \quad (3.4)$$

where

peak-signal-value = 255 for an 8 bits/pixel image.

$M*N$ = number of total pixels in image.

$f(x,y), \hat{f}(x,y)$ = value of pixel (x,y) in the original and reconstructed images.

3.2 Lossless Image Compression

When a set of arbitrary digital data is compressed such as text document or numeric accounting data, lossless compression is applied mostly. Because; as a result of decompression, it is desired that the exact original data is reproduced. If the reconstructed data is not exactly same as the original data, the reconstructed data can not be used properly in these kinds of applications. For instance; a text document might have a few missing characters after decompressing operation. Because; close approximation is not good enough for these examples. The type of compression scheme, where the compressed data is decompressed back to its exact original form is called lossless data compression. It is devoid of losses or degradation of the data.

A variety of lossless image compression schemes have been developed. Many of these techniques come directly from the digital data compression and have been merely adapted for use with digital image data. Run-Length coding, Huffman coding are typical examples for lossless image compression techniques.

3.2.1 Run-Length Coding

In lossless image compression, there is an intrinsic limitation to how much an image can be compressed. The entropy of an image is a measure of its information content. If the entropy is high, an image's information tends to be highly unpredictable. An entropy of an image can be computed as the probability of its occurrence. This is displayed as a number of bits necessary to represent that probability. For any random image, the entropy is,

Entropy = Number of pixels in a line x number of lines x number of bits per pixel.

For 640 pixel x 480 line and 8 bit image, the entropy will be 2,457,600. Any one of $2^{2,457,600}$ possible different images can be represented by an image of these dimensions.

Run-Length coding image compression takes advantage of the fact that several nearby pixels in an image will be tended to have the same brightness value. Grouping pixels of identical brightness into single codes can reduce this form of redundancy.

The Run-Length Coding is one of lossless compression techniques, when the threshold is set to zero. For threshold values greater than zero, the RLC is not one of the lossless compression techniques any more. However, higher compression ratios are obtained when the threshold is greater than zero, as a result of lossy Run-Length Coding. So, in this study, the threshold values are chosen greater than zero to compare Run-Length Coding with lossy compression techniques.

The Run-Length scheme works as following: The original image is evaluated by starting at the first pixel at the location (0,0). This pixel is named as reference pixel. For Run-Length coding, a specific threshold is chosen according to compression ratio and application. By looking at reference pixel and its following neighbours across the

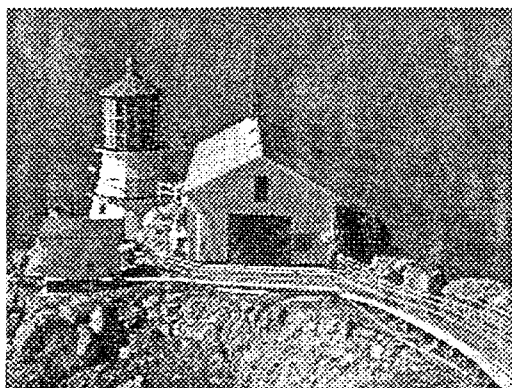
line. the scheme determines how many following pixels are in the same brightness interval that is chosen as threshold. If the next one or more pixels in the sequence are in the threshold intervals according to the reference pixel, they are all represented by a new code. The new code is made up of two values, a brightness value and the number of pixels that are in the same threshold. The process then moves to the next pixel in the same line with a new brightness value and repeats when the end of the line is reached. the process starts again at the start of the next line. The process continues until the entire image is Run-Length coded.

3.2.1.1 The Performance Result of the Run-Length Coding

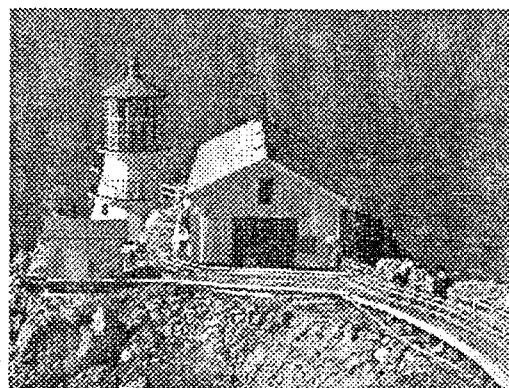
In this study, different kinds of images are compressed and decompressed by using Run-Length coding and the results are obtained. Moreover; various threshold values are chosen to examine the effect of the threshold on compression ratio and peak signal-to-noise ratio.

Table 3.1 PSNR results of different images compressed Run-Length Coding at different Threshold.

Image	Threshold=20		Threshold=35		Threshold=50	
	PSNR (dB)	C _R (bpp)	PSNR (dB)	C _R (bpp)	PSNR (dB)	C _R (bpp)
Palm1	29,94	5,12	24,59	2,63	21,37	1,45
Lightnou	29,81	3,82	24,63	2,28	21,41	1,53
Building	29,56	3,11	24,44	1,58	21,19	0,90
Text	30,00	4,17	25,16	2,38	22,13	1,56
Lenna	30,50	1,74	25,19	0,78	21,82	0,42
Peppers	30,21	1,88	24,91	0,94	21,88	0,61
Mandrill	30,73	5,68	24,57	2,83	21,12	1,40
Room	30,50	2,75	24,67	1,37	21,12	0,79



(a) The “Lightnou” image with 320 pixels x 240 lines and 256 gray-level.



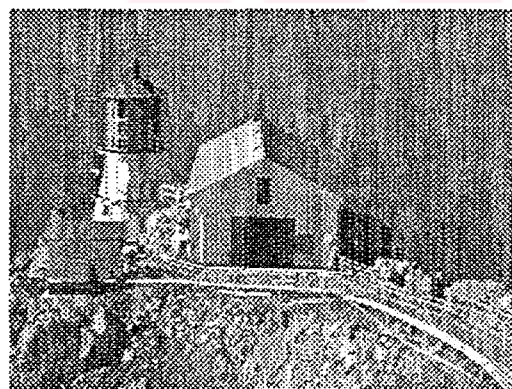
(b) Threshold = 15
SNR = 32,68dB MSE=15
CR = 4,83 bpp (1,65:1)



(c) Threshold = 20
SNR = 29,81dB MSE=68
CR = 3,82 bpp (2,10:1)

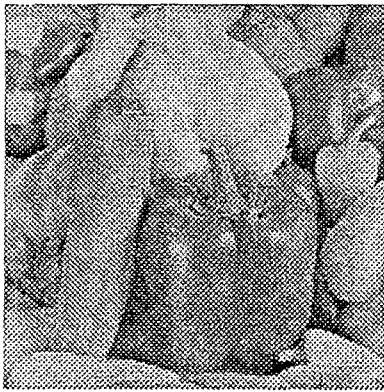


(d) Threshold = 35
SNR = 24,63dB MSE=224
CR = 2,28 bpp (3,51:1)



(e) Threshold = 50
SNR= 21,41dB MSE=470
CR = 1,53 bpp (5,22:1)

Figure 3.2 The “Lightnou” image compressed and decompressed by Run-Length coding.



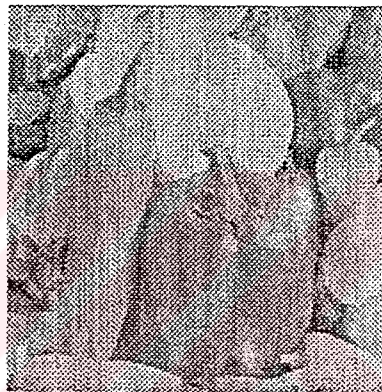
(a) The "Peppers" image with 256x256 pixels and 256 gray-level.



(b) Threshold = 15
SNR= 32,44dB MSE=37
CR = 2,44 bpp (3,28:1)



(c) Threshold = 20
SNR= 30,21dB MSE=62
CR = 1,88 bpp (4,25:1)



(d) Threshold = 35
SNR= 24,91dB MSE=210
CR = 0,94 bpp (8,52:1)



(e) Threshold = 50
SNR= 21,88dB MSE=422
CR = 0,61 bpp (13,18:1)

Figure 3.3 The "Peppers" image compressed and decompressed by Run-Length coding.

The compression results are presented in Table 3.1 and Table 3.2 for various images, and also sample images compressed and decompressed by using Run-Length Coding are given in Figure 3.2 and Figure 3.3. As shown in the tables, if the threshold increases, the compression ratio also increases but the peak signal-to-noise ratio reduces. The optimum compression ratio is chosen for applications' requirement. When the threshold is small, the decompressed image becomes closer to the original image; however, the compression ratio becomes smaller as seen in the figures.

Table 3.2 PSNR and MSE values for different Threshold levels.

Image	PSNR (dB)	MSE	bpp	CR	Threshold	Pixels
Lightnou	36,99	13	6,55	1,22	10	320x240
Lightnou	32,68	15	4,83	1,65	15	320x240
Lightnou	29,81	68	3,82	2,10	20	320x240
Lightnou	24,63	224	2,28	3,51	35	320x240
Lightnou	21,41	470	1,53	5,22	50	320x240
Lightnou	18,45	928	0,95	8,42	70	320x240
Peppers	35,34	19	3,34	2,39	10	256x256
Peppers	32,44	37	2,44	3,28	15	256x256
Peppers	30,21	62	1,88	4,25	20	256x256
Peppers	24,91	210	0,78	10,20	35	256x256
Peppers	21,88	422	0,61	13,18	50	256x256
Peppers	18,90	836	0,36	22,22	70	256x256

3.3 Lossy Image Compression

The lossy image compression techniques reduce the amount of the data required to reconstruct the original image. Although there will be losses in the digital data of the original image, the image maintain its integrity and the level of its quality.

The greatest advantage of the lossy compression techniques is their ability to compress an image to a much smaller data form than the lossless schemes.

For many applications, the degradations introduced by the lossy compression schemes can be easily accepted in exchange for their incredibly high compression ratios. For instance, television broadcasters do not need to maintain absolute image data integrity when transmitting video programmes to viewers.

Many lossy image compression schemes have been developed and some of these methods will be presented in this thesis.

3.3.1 Discrete Fourier Transform Coding

A frequency transform composes an image from its spatial-domain form of brightness into a frequency domain form of fundamental components. Each frequency component has a magnitude and a phase value. Similarly, an inverse frequency transform converts an image from its frequency-domain form back into its spatial domain form. The frequency domain form of an image is also depicted as an image where brightness is represented by the magnitudes of the various fundamental frequency components.

Numerous frequency transforms exist and each has an inverse transform to convert a frequency image back to its original spatial form of brightness. For continuous functions, the common transform used in the applications mostly is the Fourier Transform.

The 2-D Fourier transform and its inverse transform are

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^0 f(x, y) \exp(-j2\pi(ux + vy)) dx dy \quad \text{and}$$

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^0 F(u, v) \exp(j2\pi(ux + vy)) du dv \quad (3.5)$$

The type of Fourier transform, which is used to process digital functions is called Discrete Fourier Transform (DFT).

The 2-D continuous function $f(x,y)$ is discretized into a sequence $f(x_0, y_0), f(x_0 + \Delta x, y_0 + \Delta y), f(x_0 + 2\Delta x, y_0 + 2\Delta y), \dots$ by taking N samples Δx units apart. The 2-D Discrete Fourier Transform of the discretized function is

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp \left[-j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right] \quad (3.6)$$

for $u = 0, 1, 2, \dots, M-1$ and $v = 0, 1, 2, \dots, N-1$.

2-D Inverse Discrete Fourier Transform is

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp \left[j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right] \quad (3.7)$$

for $x = 0, 1, 2, \dots, M-1$ and $y = 0, 1, 2, \dots, N-1$.

When images are sampled in a square array ($M=N$), the 2-D Discrete Fourier Transform pair will be;

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp \left[-j2\pi \left(\frac{ux + vy}{N} \right) \right] \quad (3.8)$$

for $u, v = 0, 1, 2, \dots, N-1$ and

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) \exp \left[j2\pi \left(\frac{ux + vy}{N} \right) \right] \quad (3.9)$$

for $x, y = 0, 1, 2, \dots, N-1$.

In practice, images are typically digitized in square arrays, so Equation 3.8 and 3.9 transform pair is used mostly as the Discrete Fourier Transform.

In the encoding DFT compression, firstly image pixels are grouped into 8x8 block and then for each block, DFT formulation shown in Equation 3.8 is applied. As a result of this calculation, 64 values that are referred as DFT coefficients are obtained for each block. These 64 Discrete Fourier coefficients are arranged in order by Zig-Zag order rule, which is shown in Figure 3.4.

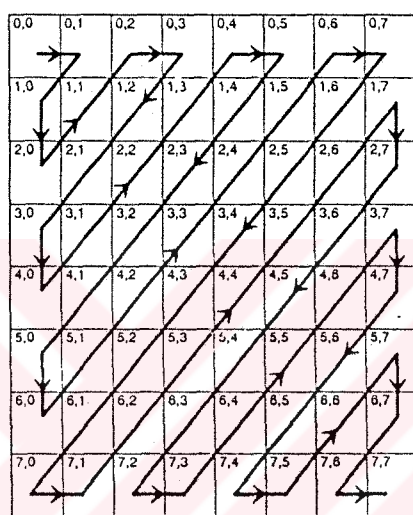


Figure 3.4 The path of Zig-Zag Sequence.

After applying Equation 3.8, complex DFT coefficients occurred. In this thesis, the magnitude of these complex coefficients are used in the decompressing operation. Similarly, the coefficients resulted from Equation 3.9 are also complex numbers. To obtain the pixel values of decompressed images, the magnitude of these complex numbers are determined.

The first coefficient that is at (0,0) location is named DC value. The other coefficients in the block are AC values. The DC coefficient and following few AC coefficients contain low frequency features of an image. The rest of the coefficients

represent the high frequency features of an image. Thus, the image can be applied to low-pass filter according to low frequency coefficients and also be applied to a high-pass filter according to high frequency coefficients. The coefficients used in compression are stored in the memory or transmit along to the channel and the others are set to zero. As a result, the compression ratio can be calculated according to the data in image and the number of stored coefficients. The reconstructed image is obtained by using Inverse Fourier Transform as given in Equation 3.9.

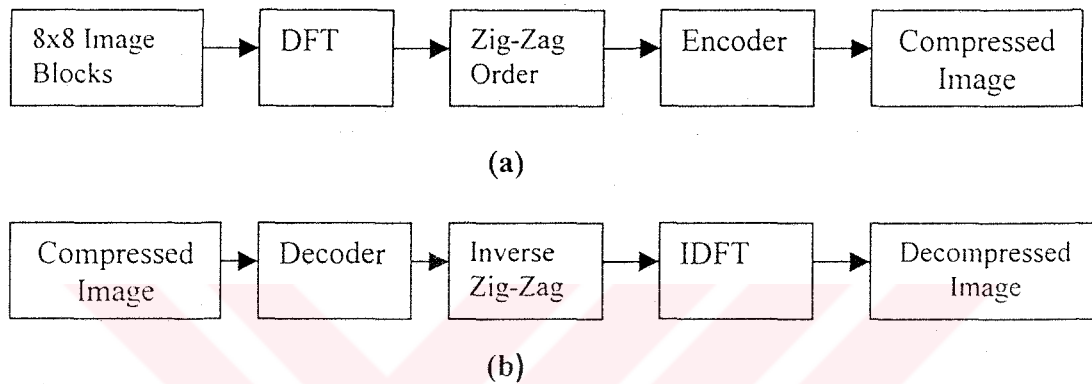
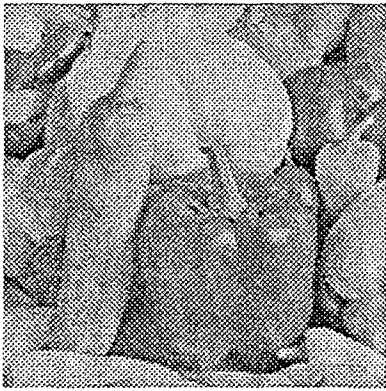


Figure 3.5 The compression and decompression scheme of Discrete Fourier Transform Coding.

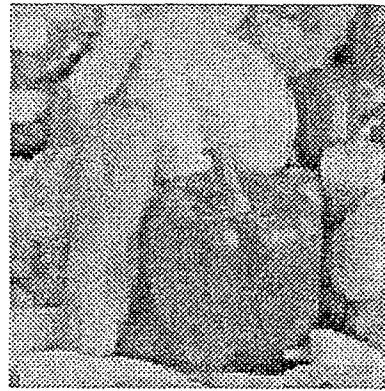
In this thesis, the compression and decompression block diagram for Discrete Fourier Transform Coding presented in Figure 3.5 are used to simulate this compression method and the results are obtained.

3.3.1.1 The Performance Result of Discrete Fourier Transform Coding

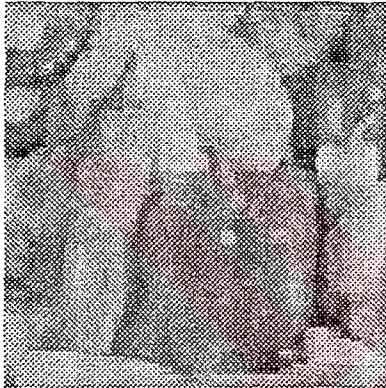
The images are compressed by Discrete Fourier Transform compression technique at different compression ratios and the results are obtained for various images at different compression ratios. The sample images are shown in Figure 3.6 and the all results are given in Table 3.4. As shown in the table, as a result of decreasing compression ratio, the signal-to-noise ratio reduces.



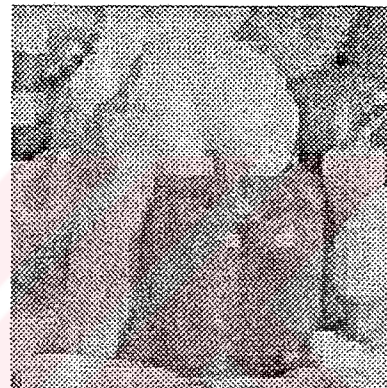
(a) The "Peppers" image with
256 gray-level and 256x256



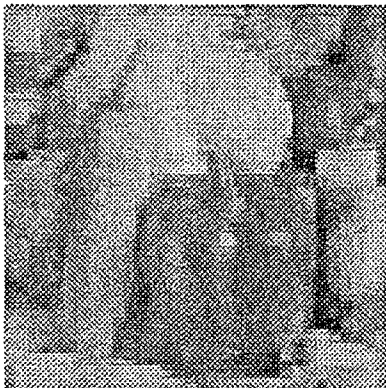
(b) SNR=29,55dB MSE=72
CR = 6 bpp (1,33:1)



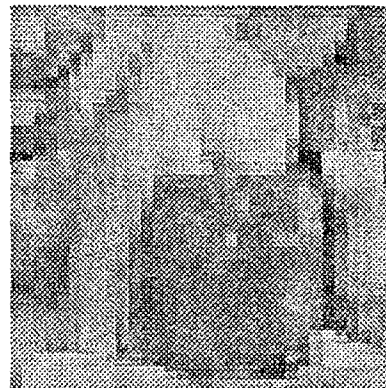
(c) SNR=26,92dB MSE=132
CR = 3 bpp (2,67:1)



(d) SNR=25,82dB MSE=170
CR = 1,5 bpp (5,33:1)



(e) SNR=24,57dB MSE=227
CR = 0,75 bpp (10,66:1)



(f) SNR=22,97dB MSE=328
CR = 0,375 (21,33:1)

Figure 3.6 Compressed and decompressed images by DFT Coding.

The magnitude of first 64 Discrete Fourier Transform coefficients of a sample image are shown in Table 3.3 according to pixel values of image. As shown in Table 3.3, the first coefficient (DC) is the biggest value in the all coefficients.

Table 3.3 The pixel values of the first 8x8 block of “Peppers” image and 64 DFT coefficients of this image.

8x8 block pixel values								8x8 block DFT coefficients							
104	156	156	152	152	140	124	120	1356	69	57	58	55	58	57	69
132	184	184	184	188	184	184	176	31	7	13	9	10	10	9	13
128	180	184	184	184	184	180	176	39	16	8	10	9	7	4	2
96	188	184	184	184	180	180	176	36	4	8	11	9	10	11	14
92	188	184	184	184	180	180	176	43	10	4	4	4	4	4	10
148	184	184	180	180	184	180	172	36	14	11	10	9	11	8	4
112	184	184	180	180	184	180	176	39	2	4	7	9	10	8	16
128	184	180	180	184	184	180	176	31	13	9	10	10	9	13	7

Table 3.4 Peak signal-to-noise ratio (PSNR) versus compression ratio (bpp) for various images compressed by Discrete Fourier Transform coding.

bpp/dB	0,375	0,75	1,125	1,5	2,25	3	4,5
Palm1	20,17	21,48	22,09	22,48	22,96	23,46	24,10
Lightnou	19,95	20,80	21,50	21,86	22,24	22,53	22,88
Lenna	23,93	25,80	26,49	26,73	27,45	27,76	28,54
Peppers	22,17	24,57	25,39	25,83	26,43	26,93	27,52
Tiffany	23,79	24,71	25,90	26,67	27,13	26,09	26,04

In this study, the peak signal-to-noise ratios for various images, which are compressed and decompressed by using Discrete Fourier Transform are obtained for different compression ratios. According to Table 3.4, if the compression ratio increases, the peak signal-to-noise value becomes worse. In Discrete Fourier

Transform, the compression ratio directly depends on the DFT coefficients that are used in the decompression operation.

3.3.2 Discrete Cosine Transform Coding

Discrete Cosine Transform (DCT) has become very popular choice for image data compression in recent years. In the DCT coding, firstly image pixels are grouped into 8x8 block and then Equation 3.10 is applied to each block similar to DFT. As a result of calculation 64 DCT coefficients for each block, the coefficients are arranged in order by Zig-Zag order rule. The coefficients, which are not used in compression are set to zero. The decompressed image is obtained by using inverse Discrete Cosine Transform given in Equation 3.11.

The scheme of the compression and decompression of Discrete Cosine Transform Coding is given in Figure 3.7.

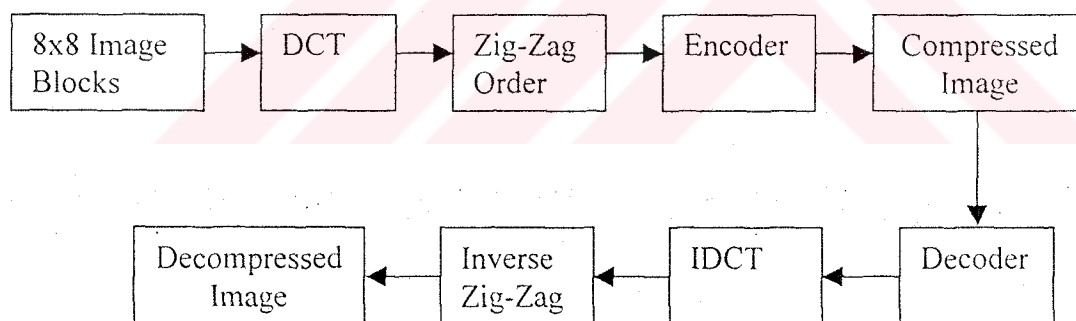


Figure 3.7 The compression and decompression scheme of Discrete Cosine Transform Coding.

The 2-D DCT pair is defined as [15];

$$C(u, v) = \frac{1}{\sqrt{2N}} \alpha(u) \alpha(v) \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} f(x, y) \cos\left[\frac{u\pi(2x+1)}{2N}\right] \cos\left[\frac{v\pi(2y+1)}{2N}\right] \quad (3.10)$$

for $u, v=0, 1, 2, \dots, N-1$,

$$f(x, y) = \frac{1}{\sqrt{2N}} \sum_{v=0}^{N-1} \sum_{u=0}^{N-1} \alpha(u) \alpha(v) C(u, v) \cos\left[\frac{x\pi(2u+1)}{2N}\right] \cos\left[\frac{y\pi(2v+1)}{2N}\right] \quad (3.11)$$

for $x, y=0, 1, 2, \dots, N-1$.

$$\alpha(0) = \frac{1}{\sqrt{2}} ; \quad \alpha(u) = 1 ; \quad \text{for } u=1, 2, 3, \dots, N.$$

$$\alpha(v) = 1 ; \quad \text{for } v=1, 2, 3, \dots, N.$$

In this study, Equation 3.10 is used to obtain the Discrete Cosine Transform coefficients and to decompressed the images, Equation 3.11 is used.

3.3.2.1 The Performance Result of Discrete Cosine Transform Coding

By using Equation 3.10 and 3.11, DCT compression and decompression are simulated. Figure 3.7 and Figure 3.8 show sample images compressed and decompressed by using DCT techniques and Zig-Zag order rule. Moreover; the compression result are obtained for various images at different compression ratios as presented in Table 3.5.



(a) The "Palm1" image with 256 gray-level and 320x240 pixels.



(b) SNR=30,73dB MSE=99
CR = 6 bpp (1,33:1)



(c) SNR=26,14dB MSE=158
CR = 4,5 bpp (1,77:1)



(d) SNR=24,81dB MSE=215
CR = 3 bpp (2,67:1)



(e) SNR=24,59dB MSE=265
CR = 2,25 bpp (3,55:1)



(f) SNR=22,57dB MSE=360
CR = 1,5 bpp (5,34:1)

Figure 3.8 Compressed and decompressed sample images by DCT.



(g) SNR=22,15dB MSE=396
CR = 1,125 (7,11:1)



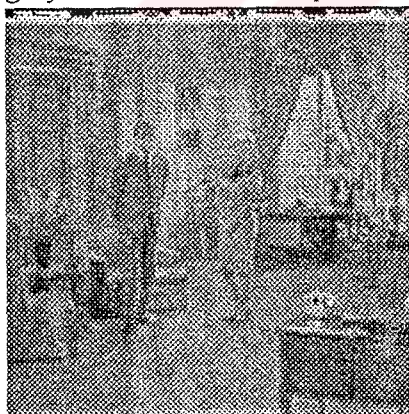
(h) SNR=21,37dB MSE=474
CR = 0,75 bpp (10,67:1)



(i) The "Room" image with 256
gray-level and 256x256 pixels.



(j) SNR=24,20dB MSE=247
CR = 3 bpp (2,67:1)



(k) SNR=22,77dB MSE=224
CR = 2,25 bpp (3,55:1)



(l) SNR=21,29dB MSE=483
CR = 0,75 bpp (10,67:1)

Figure 3.8 (cont.)

The first 64 Discrete Cosine Transform coefficients and pixels values of “Palm1” image are shown in Table 3.5. As shown in the table, the first coefficient (DC) is the biggest value in the all coefficients.

In this study, the peak signal-to-noise ratios of various images, which are compressed and decompressed by using Discrete Cosine Transform, are obtained for different compression ratios. According to the results that are presented in Table 3.6, if the compression ratio increases, the peak signal-to-noise ratio becomes worse. In Discrete Cosine Transform, the compression ratio directly depends on the DCT coefficients that are used in the decompression operation.

Table 3.5 The pixel values of the first 8x8 block of “Palm1” image and 64 DCT coefficients of this image.

8x8 block pixel values								8x8 block DCT coefficients							
218	218	215	216	213	215	214	216	1724	-1	2	4	2	-1	-2	1
217	217	214	214	213	213	215	216	-2	6	5	-4	1	1	2	-2
216	216	214	214	215	213	214	216	1	1	1	1	1	1	0	-2
215	215	215	214	215	214	216	214	4	1	1	1	0	0	1	-1
215	215	215	214	217	216	216	216	0	1	-1	0	0	1	1	0
216	216	214	215	219	218	217	216	1	1	0	1	0	1	0	0
215	215	213	215	218	217	218	215	2	1	0	1	2	-1	1	0
214	214	212	215	219	217	217	215	0	0	1	0	1	0	1	0

The compression results presented below, the coefficients are obtained by using Zig-Zag order. If the coefficients along the x-axis are used to reconstruct the image, the peak signal-to-noise ratio decreases in the same compression ratio as shown in Table 3.6.

Table 3.6 The comparison of the Normal Order (N-O) with Zig-Zag Order.

bpp/dB	1,5 bit/pixel		3 bit/pixel	
	N-O	Z-Z-O	N-O	Z-Z-O
Peppers	24,24	28,84	28,84	32,57
Room	24,38	22,43	26,46	24,20
Palm1	21,05	22,57	23,20	24,81

Table 3.7 Peak Signal-to-noise ratio (PSNR) versus compression ratio (bpp) for DCT compression.

bpp/dB	0,1875	0,375	0,75	1,125	1,5	2,25	3	4,5	6
Palm1	19,61	20,19	21,37	22,15	22,57	24,59	24,81	26,14	28,17
Building	20,36	20,75	19,62	19,36	19,24	19,29	19,33	19,92	20,12
Lightnou	19,15	19,81	19,94	20,34	20,32	20,55	20,70	20,81	20,80
Lenna	23,51	24,08	26,99	28,35	28,89	30,42	31,41	33,36	34,71
Peppers	22,06	24,53	26,06	27,84	28,84	30,57	32,57	34,91	37,34
Madrill	21,15	21,71	22,77	23,45	23,93	24,63	25,48	26,70	28,54
Room	21,15	20,62	21,29	21,38	22,43	22,77	24,20	24,63	30,73
View	20,03	19,52	20,23	21,26	21,69	22,26	23,19	24,22	25,88
Tiffany	22,09	22,22	22,34	23,00	24,03	24,51	27,13	27,34	31,23

3.3.3 Application of Discrete Cosine Transform : JPEG Compression

The Joint Photographic Experts Group (JPEG) standard, established jointly by the ISO/IEC and CCITT organisations, is one of the best important image data compression standard of 1990's [3].

The JPEG image data compression standard handles gray-scale and colour images of varying resolution and size. It is intended to support many industries that need to transport and archive images. The JPEG standard is used in graphics and desktop publishing, medical imaging, colour fax and countless other applications. This standard is commonly used in lossy mode. However; there is also lossless mode with reduced compression performance.

The JPEG specification consists of several parts, including a specification for both lossless and lossy encoding. The JPEG lossy compression algorithm operates in three successive stages shown in Figure 3.9.

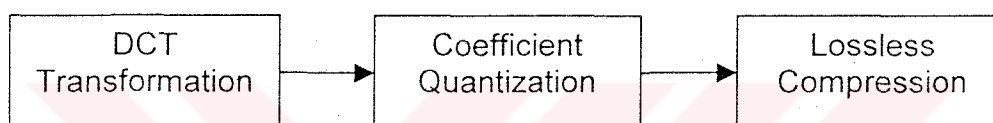


Figure 3.9 The Block Diagram of JPEG Compression

Firstly, an image is transformed to the frequency domain using the Discrete Cosine Transform. Then resulting smaller valued frequency components are discarded, leaving only the longer-valued components. The remaining frequency components are DPCM coded and then Huffman coded.

The JPEG compression scheme is adjustable. For instance, the number of retained frequency components can be changed, producing variable compression ratios and inversely proportional decompressed image quality. The JPEG algorithm can be fine-tuned to meet an application's requirements of compressed image data size and decompressed image quality.

3.3.4 Vector Quantization (VQ)

Image data compressing using vector quantization has attracted the interest of both academia and the data compression industry since the early 1980's. Vector quantization is one of the most powerful tools for audio, speech, image and video compression. The most important features of vector quantization can be summarized as follows; simplicity of the decoder, potential of a fractional bit allocation for the vector components, ability to exploit the statistical correlation between neighboring data in a straight forward manner, reduce the transmission bit rate or the storage data.

An important goal of the design of a vector quantization is the construction of the VQ codebook, since it affects both coding efficiency and implementation complexity of the decoder. Linde, Guzo and Gray have developed a constructive method for VQ codebook design that is a systematic generalization of Lloyd's method I for designing optimum scalar quantizer [1]. The generalized Lloyd algorithm or LGB algorithm generates a locally optimal codebook by iteratively improving an initial codebook with respect to a given training sequence. Typically, the VQ codebook designed using LGB consists of quantization regions with nonregular shapes and sizes, so that the best available code vector can be selected after comparing a given input vector against to all available code vectors in the codebook. LGB methods guarantee a locally optimum codebook relative to the source vectors.

Encoding complexity is a major drawback in the real-time implementation of full-search VQ, since the number of computations required for the selection of the closest code vector, increases exponentially with the vector dimensions and the coding rate. Several methods have been improved to solve the problem of VQ encoding complexity. Typically, these methods involve imposing a certain structure to the VQ codebook so that unconfined access to all effective code vectors is restricted. Tree-

structured VQ, multistage VQ, product-code VQ, classified VQ, and finite-state VQ fall into this category. They have proved to be very successful for both image and video coding applications [21].

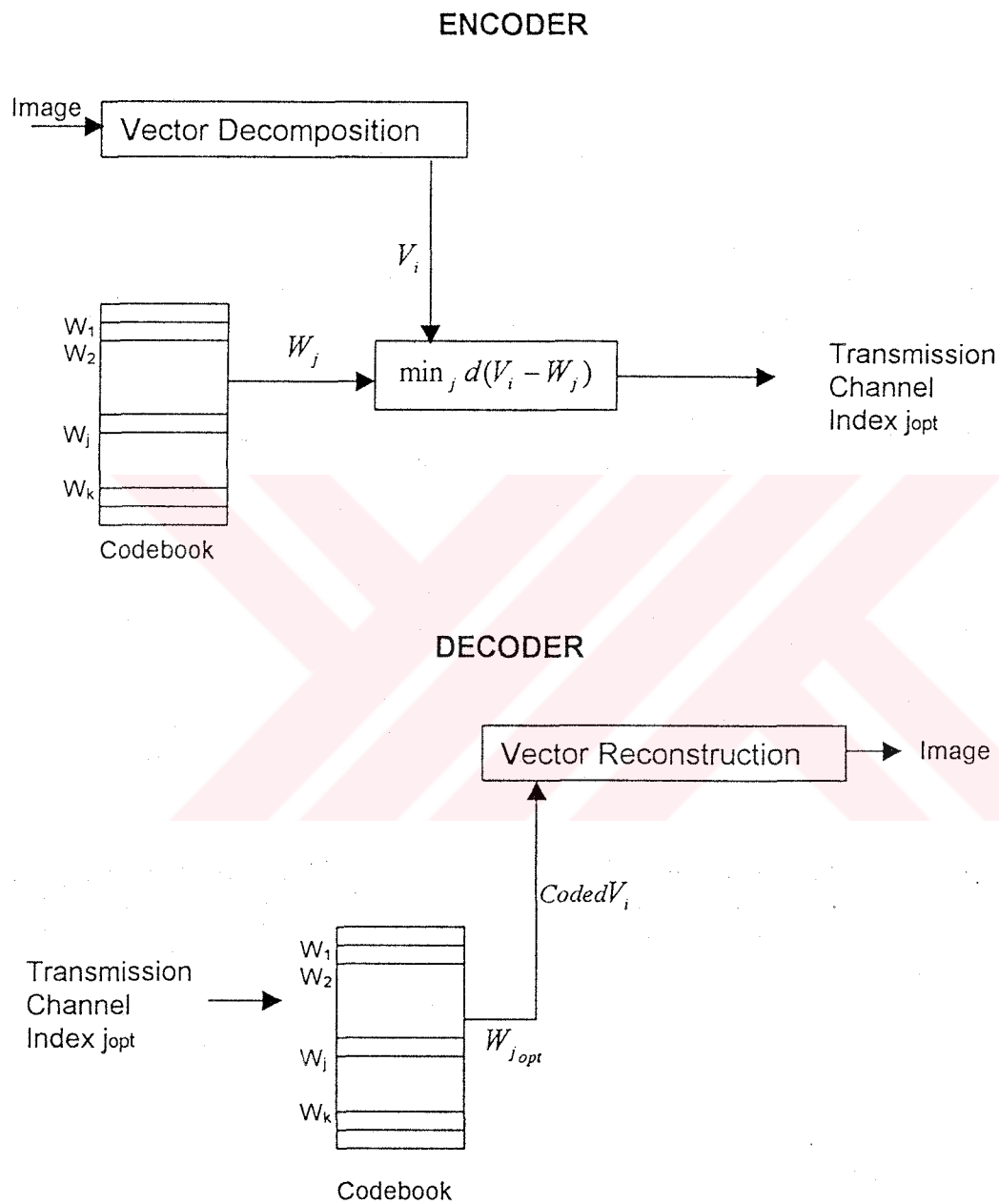


Figure 3.10 Encoding and Decoding Scheme of Vector Quantization.

The encoding and decoding scheme of vector quantization compression technique is shown in Figure 3.10. This scheme is the basis of the other types of vector quantization techniques.

In general, both the codebook size and the vector dimension play very important role in overall performance. If the vector dimension is chosen large, the better potential performance is obtained. However, with increased vector dimension, the required codebook size also increases, and increasing vector sizes add computational complexity. So the most common vector sizes are chosen as 1x2, 2x2 and 4x4-pixel dimension. Different codebook sizes can be used to compressed images for example; 8, 16, 32 etc. The vector size and codebook size is selected according to the application. The more vector sizes and codebook sizes are chosen, the more competitive reconstructed images are obtained.

3.3.4.1. The Vector Quantizer Design

A vector quantizer can be defined as a mapping Q of K -dimensional Euclidean space R_K into finite subset Y of R_K . Thus,

$$Q : R_K \rightarrow Y \quad \text{where } Y = (\hat{x}_i; i=1,2,3,\dots,N)$$

is the set of reproduction vectors and N is the number of vectors in Y . An encoder generates the address of the reproduction vector by using input vector x . And a decoder generates the reproduction vector \hat{x} by using addresses. The distortion measures with $d(x, \hat{x})$ that represent the penalty between x and \hat{x} vectors and the best mapping minimizes $d(x, \hat{x})$. The LGB algorithm and other variations of this algorithm are based on this minimization. One simple distortion measure is the square error distortion for K -dimensional Euclidean space given by Equation 3.12. This equation measures the square of the Euclidean between the vectors.

$$d(x, \hat{x}) = \|x - \hat{x}\|^2 = \sum_{j=0}^{K-1} (x_j - \hat{x}_j)^2 \quad (3.12)$$

The purpose of an optimal vector quantizer designing is to obtain a quantizer consisting of N reproduction vectors, such that it minimizes the expected distortion. Lloyd proposed an iterative nonvariational technique for design of scalar quantizer.

The time averaged square error distortion is given by Equation 3.13.

$$D(x, q(x)) = \frac{1}{N} \sum_{i=0}^{N-1} d(x_i, \hat{x}_i) \quad (3.13)$$

The LGB algorithm for a known distribution training sequence follows these rules:

1) Initialization: Given N = number of levels, a distortion threshold $\epsilon \geq 0.0$, an initial N -level reproduction alphabet \hat{A}_0 , and a training sequence $\{x_j; j = 1, 2, \dots, n\}$. Set $m=0$ and $D_{-1} = \infty$.

2) Given $\hat{A}_m = \{y_i; i=1, 2, \dots, N\}$, find its minimum distortion partition $P(\hat{A}_m) = \{S_i; i=1, 2, \dots, N\}$ of the training sequence: $x_j \in S_i$ if $d(x_j, y_i) \leq d(x_j, y_k)$ for all k .

Compute the resulting average distortion

$$D_m = D[(\hat{A}_m, P(\hat{A}_m))] = \frac{1}{n} \sum_{j=0}^{n-1} \min_{y \in \hat{A}_m} d(x_j, y) \quad (3.14)$$

3) If $(D_{m-1} - D_m)/D_m \leq \epsilon$, halt with \hat{A}_m and $P(\hat{A}_m)$ describing the final quantizer. Otherwise, continue.

4) Find the optimal reproduction alphabet $\hat{x}(P(\hat{A}_m)) = \{\hat{x}(S_i); i = 1, 2, \dots, N\}$ for $P(\hat{A}_m)$. Set $\hat{A}_{m+1} = \hat{x}(P(\hat{A}_m))$. Replace m by $m+1$, and go to 2.

of successively higher rates until achieving an acceptable level of distortion. This method considers an M-level quantizer with $M=2^R$, $R=0,1,\dots$, and continues until it achieves an initial guess for N-level quantizer as follows:

- 1) Initialization: Set $M=1$, and define $\hat{A}_0 = \hat{x}(A)$, which is the centred of the entire alphabet (the centred of the training sequence if a sample distribution is used.)
- 2) Given the reproduction alphabet $\hat{A}_0(M)$ containing M vectors $\{y_i; i=1,\dots,M\}$, split each vector y_i into two close vectors $y_i + \varepsilon$ and $y_i - \varepsilon$, where ε is a fixed perturbation vector. The collection \hat{A} of $\{y_i + \varepsilon, y_i - \varepsilon, i=1,\dots,M\}$ has 2M vectors. Replace M by 2M.
- 3) When M is equal to N, it is set $\hat{A}_0 = \hat{A}(M)$ and the process is halted. \hat{A}_0 is then the initial reproduction alphabet for the N-level quantization algorithm. If not, run the LGB algorithm for an M-level quantizer on $\hat{A}(M)$ and then return to step 2.

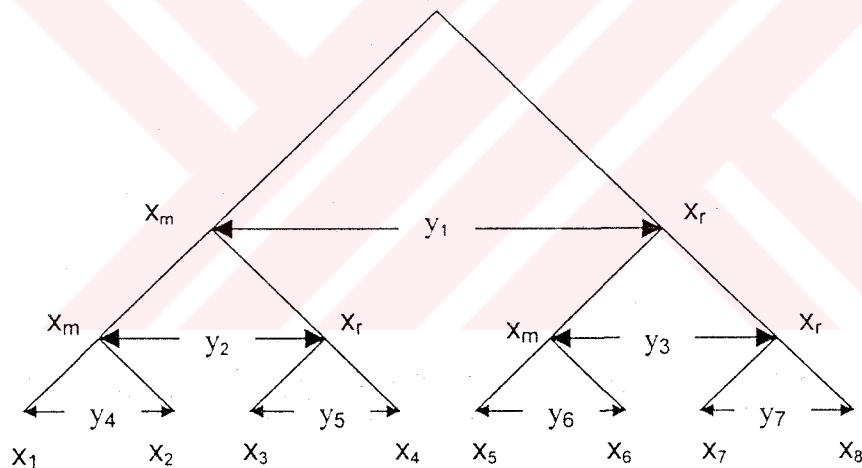


Figure 3.11 The typical tree of codebook structure of Vector Quantization.

The typical tree of the codebook structure of an image for vector quantization is presented in Figure 3.11. This tree structure is an example of 8 codebook design. As a result of these operations, the N codebook or reproduction vector is obtained. Each vector is then compared to a set of codeword vector stored in a ROM, and a codeword address, which is identifying the best match is transmitted. The receiver reconstructs

the image using corresponding templates in plane of the original vectors that are stored in the receiver codebook ROM. The compression ratio for vector quantization is calculated simply as given Equation 3.15.

$$C_R = \frac{\log_2 N_c}{k} \text{ bpp} \quad (3.15)$$

where N_c is the codebook size and k is the number of pixel in selected codevector. For example; if it is used 2x2 codevector and 32 codebook, the compression ratio will be 1,25 bit/pixel.

3.3.4.2 The Performance Result of Vector Quantization

In this thesis, images are compressed and decompressed by using vector quantization technique for different codebook and codevector sizes. The sample images are shown in Figure 3.12 and Figure 3.13 and the obtained results are given in Table 3.10, Table 3.12 and Table 3.13.



(a) The "Lenna" image with 256 gray-level and 256x256 pixels.



(b) SNR=31,41dB MSE=47
CR = 3,5 bpp (2,28:1)

Figure 3.12 The "Lenna" images by decompressed Vector Quantization in different codebook sizes with 1x2 vector size.



(c) SNR=31,32dB MSE=48
CR = 3 bpp (2,66:1)



(d) SNR=31,23dB MSE=49
CR = 2,5 bpp (3,2:1)



(e) SNR=31,32dB MSE=54
CR = 2 bpp (4:1)



(f) SNR = 29,44dB MSE = 74
CR = 1,5 bit/pixel (5,33 : 1)

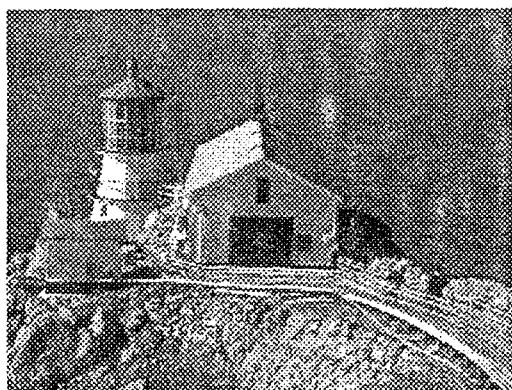


(g) SNR=26,73dB MSE=138
CR = 1 bpp (8:1)

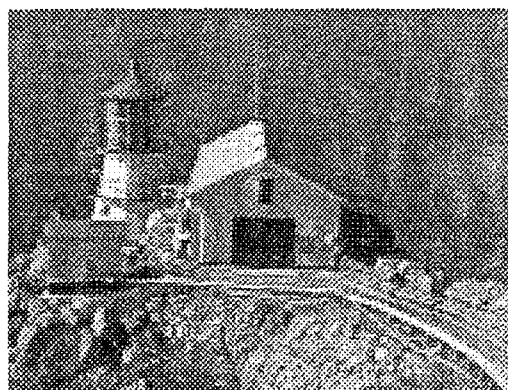


(h) SNR=21,99dB MSE=411
CR = 0,5 bpp (16:1)

Figure 3.12 (cont.)



(a) The "Lightnou" image with 256 gray-level and 320x240 pixels.



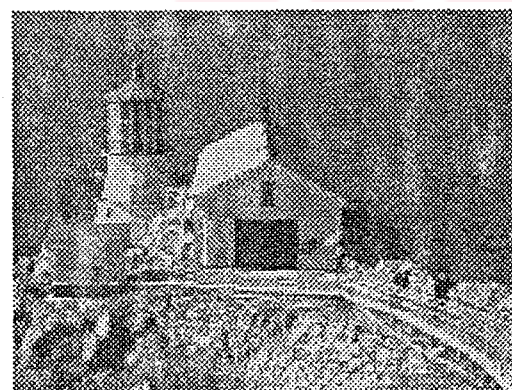
(b) SNR=27,20dB MSE=119
CR = 3 bpp (2,66:1)



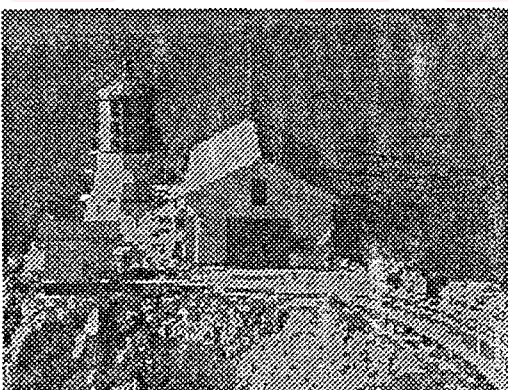
(c) SNR=27,06dB MSE=124
CR = 2,5 bpp (3,2:1)



(d) SNR=26,58dB MSE=249
CR = 2bpp (4:1)



(e) SNR=25,06dB MSE=218
CR = 1,5 bpp (5,33:1)



(f) SNR=22,01dB MSE=409
CR = 1 bpp (8:1)

Figure 3.13 The "Lightnou" images by decompressed Vector Quantization in different codebook sizes with 1x2 vector size.

The codebooks for various sizes are presented Table 3.8 and Table 3.9 to give an example to the codebook type.

Table 3.8 The codebooks of the “Lenna” image shown in Figure 3.12 (d).

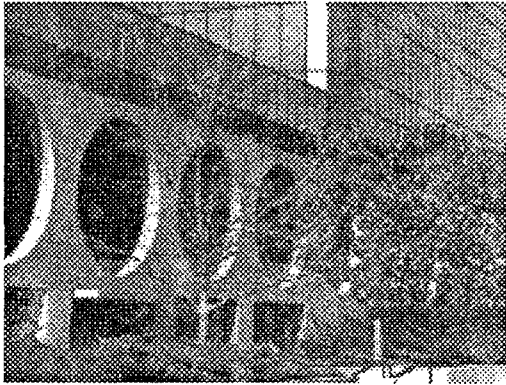
Codebook (x_1, x_2)			Codebook (x_1, x_2)			Codebook (x_1, x_2)			Codebook (x_1, x_2)		
1	53	53	9	79	80	17	104	103	25	133	134
2	59	59	10	82	82	18	106	106	26	141	140
3	62	61	11	85	86	19	110	110	27	149	147
4	64	65	12	88	89	20	114	114	28	155	155
5	67	68	13	91	92	21	117	117	29	162	162
6	70	71	14	94	94	22	121	120	30	170	171
7	73	74	15	97	98	23	124	125	31	179	180
8	76	77	16	100	101	24	129	128	32	191	191

Table 3.9 The codebook of the “Lenna” image shown in Figure 3.10 (f).

Codebook (x_1, x_2)			Codebook (x_1, x_2)			Codebook (x_1, x_2)			Codebook (x_1, x_2)		
1	60	61	3	84	85	5	109	109	7	145	145
2	73	73	4	96	96	6	123	123	8	174	174

Table 3.10 The peak Signal-to-noise ratio (dB) versus compression ratio (bpp) for Vector Quantization with 1x2 codevector.

Codebook	2	4	8	16	32	64	128
bpp/dB	0,5	1	1,5	2	2,5	3	3,5
Palm1	17,52	22,02	25,06	26,58	27,06	27,20	27,25
Lightnou	18,07	22,01	24,75	26,40	27,20	27,38	27,49
Building	17,60	20,98	23,85	26,09	28,22	29,44	29,50
Text	17,32	20,74	23,54	25,28	26,55	26,76	27,13
Lenna	21,99	26,73	29,44	30,81	31,23	31,32	31,41
Peppers	20,20	24,69	28,69	30,81	31,70	31,75	31,80
Madrill	20,39	24,33	26,52	27,34	27,64	27,68	27,68
Room	19,76	23,43	26,43	28,59	29,87	30,00	30,54



(a) The "Lightnou" image with 256 gray-level and 320x240 pixels.



(b) SNR=27,60dB MSE=113
CR = 1,75 bpp (4,57:1)



(c) SNR=27,56dB MSE=114
CR = 1,5 bpp (5,34:1)



(d) SNR=27,45dB MSE=117
CR = 1,25 bpp (6,4:1)



(d) SNR=25,30dB MSE=192
CR = 0,75 bpp (10,67:1)



(e) SNR=22,26dB MSE=386
CR = 0,5 bpp (16:1)

Figure 3.14 The "Building" images by decompressed Vector Quantization in various codebook sizes with 2x2 vector size.

Table 3.11 Peak signal-to-noise ratio (PSNR) versus compression ratio (bpp) for Vector Quantization with 2x2 codevector.

Codebook	2	4	8	16	32	64	128
bpp/dB	0,25	0,5	0,75	1	1,25	1,5	1,75
Palm1	17,40	21,06	23,09	23,95	24,19	24,24	24,26
Lightnou	17,80	21,02	22,87	23,57	23,75	23,79	23,79
Building	17,94	22,26	25,30	27,02	27,45	27,56	27,60
Text	17,65	20,60	23,09	24,05	24,27	24,33	24,33
Lenna	21,88	26,06	28,26	29,27	29,56	29,62	29,38
Peppers	20,00	24,05	26,86	28,13	28,54	28,64	28,45
Madriill	19,77	22,77	24,12	24,53	24,69	24,71	24,71
Room	19,46	22,85	25,21	26,70	27,41	27,52	27,52

Table 3.12 The codebook of the “Building” image shown in Figure 3.14 (d).

Codebook (x_1, x_2, y_1, y_2)			Codebook (x_1, x_2, y_1, y_2)			Codebook (x_1, x_2, y_1, y_2)			Codebook (x_1, x_2, y_1, y_2)		
1	37	37	3	87	86	5	124	124	7	169	168
	36	37		86	86		124	124		168	168
2	63	63	4	105	105	6	142	142	8	221	222
	63	63		106	105		142	142		222	221

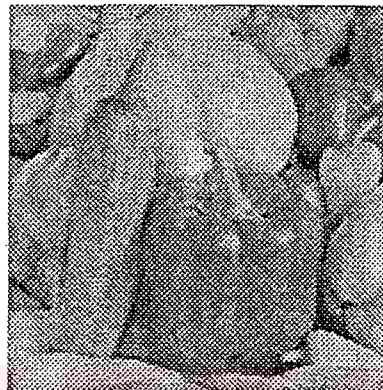
Table 3.13 Peak signal-to-noise ratio (PSNR) versus compression ratio (bpp) for Vector Quantization with 4x4 codevector.

Codebook	64	128	Codebook	64	128
bpp/dB	0,375	0,4375	bpp/dB	0,375	0,4375
Palm1	20,87	20,88	View	21,18	21,19

According to the performance results, the more codebook size is used, the better peak signal-to-noise ratio is obtained and the reconstructed images are more competitive the original images. Moreover; when the less vector size is used, the peak signal-to-noise ratio increases, because; the details in the image can be process better, when the vector size is small.



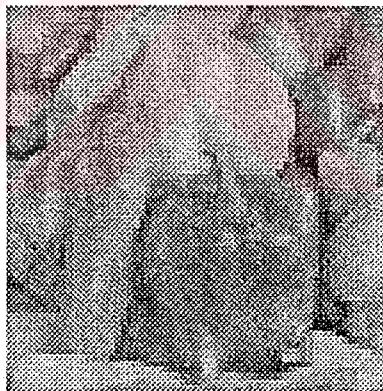
(a) The "Peppers" image with 256 gray-level and 256x256 pixels.



(b) Codebook = 64 Vector Size = 1x2
SNR=31,75dB MSE=
CR =3bpp (2,66:1)



(c) Codebook = 64 Vector Size = 2x2
SNR=28,64dB MSE=89
CR =1,5bpp (5,33:1)



(d) Codebook = 64 Vector Size = 4x4
SNR=20,77dB MSE=254
CR =0,375bpp (21,33:1)

Figure 3.15 The compressed and decompressed "Peppers" image at the same codebook and various codevectors.

3.3.5 Hierarchical Finite-State Vector Quantization (HFSVQ)

Vector quantization is an efficient spatial domain image coding technique at low bit rate, for example, 1bit/pixel. Various improvement of the vector quantization techniques have been developed, in order to achieve successful compression while keeping satisfactory quantities of the reconstructed image. Vector quantization generally causes high frequency quantization errors around sharp edges. The high frequency errors may cause sharp edges blurred, which damages the perceptual quality in the reconstructed images. However, the distortions in the smooth area may not be apparently observed by human eye.

Hierarchical Finite-State Vector Quantization is developed on the basis of the vector quantization techniques. However, in the HSVQ technique, an original image is decomposed into blocks of different sizes, which are assigned into different layers according to their gray-level scale contrast.

The blocks with low contrast, which are located in a smooth region of the image, will have large block sizes and be assigned into higher levels. Fewer codebook sizes are chosen for the representative vectors in these higher layers, since there are strong correlation between adjacent pixels in the smooth regions. For blocks with high contrast where the gray-scales vary dramatically from pixel to pixel, more codebook is used to represent details in an image.

The codes for image coding with the HFSVQ consists of two parts:

1. The first part consists of the structure codes that provide the information of layer assignment of the image blocks. This is named as Structure Map.
2. This other part consists of the local address indexes of the codeword in the codebook.

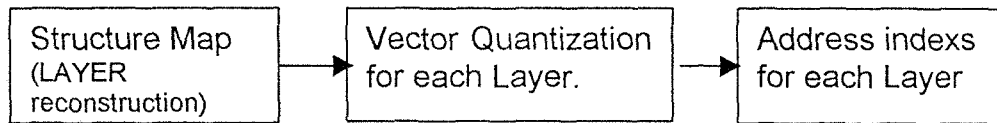


Figure 3.16 A general compression scheme of the Hierarchical Finite-State Vector Quantization.

The HFSVQ scheme is more efficient than conventional VQ techniques, because it is adjusted the reconstructed accuracy in different regions. As a result of the compression, lower bit rate can be obtained than the other methods.

3.3.5.1 Structure Map

Any image can be divided into several regions according to its gray-level contrast for a given threshold. The gray-scale contrast is calculated by horizontal and perpendicular gradient. The horizontal gradient is the average difference between the horizontal pixels and the perpendicular gradient is the average difference between the perpendicular pixels in a block. If the both gradients in a block are smaller than the threshold, this block is accepted in the smooth layer. Otherwise, it is assigned into higher layer.

To reconstruct the structure map of an image, firstly, the whole image is divided into a group of blocks of size 16x16. If the gray-level is lower than the selected threshold, this block is assigned into layer 1 (L_1) that is the smoothest layer. Otherwise, this block can be divided into subblocks of size 8x8. The same threshold is applied this block and if the gray-level is lower than the threshold, it is assigned into layer 2 (L_2), which are fairly smooth regions than L_1 . If it is not, this block again can be divided into subblocks of size 4x4. The same threshold is used and the gradient of this block is smaller than the threshold, it is put into layer 3 (L_3), otherwise, it is assigned into layer 4 (L_4).

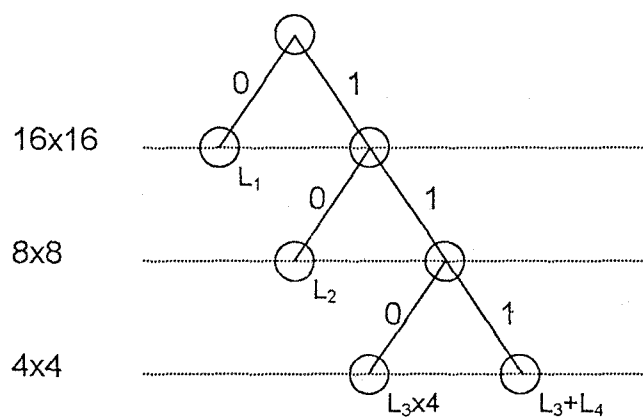


Figure 3.17 A Structure Tree.

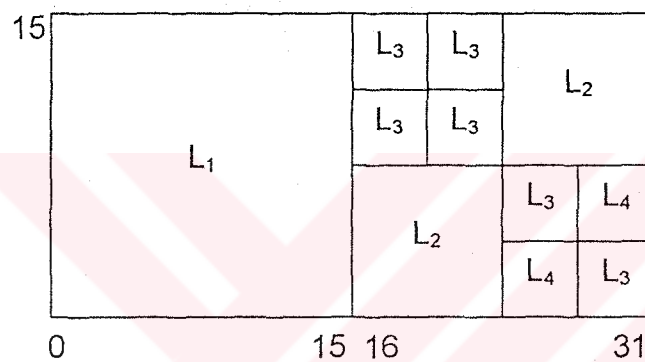


Figure 3.18 A Structure Map.

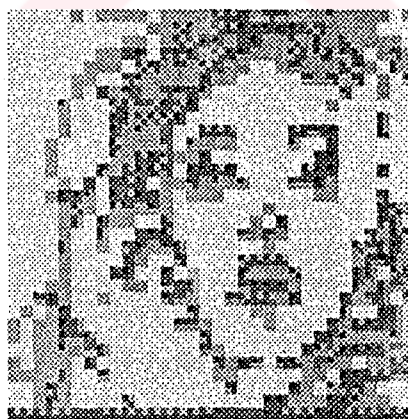


Figure 3.19 The "Tiffany" image and its Structure Map.

3.3.5.2 Hierarchical Finite-State Coding

After obtained a structure map of any image, the vector quantization technique is applied to each layer and the codebook for each layer is obtained individually. Least bits or codebook for Layer1 and most bits or codebook for Layer4 are chosen to represent the original image. As a result, the details in the image are seen clearly.

The compression ratio for four layers is calculated as following:

$$CR = \frac{N_1 \cdot b_1 + N_2 \cdot b_2 + N_3 \cdot b_3 + N_4 \cdot b_4}{N} \frac{1}{\log_2 k} \text{ bpp} \quad (3.16)$$

N : The total number of pixels in the original image.

$N_1 \dots N_4$: The number of pixels for each layer. They are changing according to selected threshold and image.

$b_1 \dots b_4$: The number of bits used for each layer.

k : The number of codevector size.

In this study, four different kind of codebook for each layer and three different threshold values are selected and applied to the various images. Threshold values are selected to 4,5 and 6 respectively, and the effect of the threshold on the compression ratio and signal-to-noise ratio are examined. While simulating the HFSVQ compression method, 2x2 codevector is used for all layers. In addition, to examine the effect of the codebook size on the performance of the compression, various codebook sizes are chosen and the results are obtained.

The various codebook and threshold values for each Layer in HFSVQ technique are presented in Table 3.14. The necessary bit number for each codebook is equal to $bit = \log_2 codebook$.

Table 3.14 Various codebook and threshold values for each Layer in HFSVQ compression.

Codebook size	Layer1	Layer2	Layer3	Layer4	Threshold		
Option -1	2	4	8	16	4	5	6
Option -2	4	4	16	32	4		
Option -3	4	4	32	64	4		
Option -4	8	16	32	64	4		

3.3.5.3 The Performance Result of Hierarchical Finite-State Vector Quantization

The sample images compressed at different threshold values and codebook size are presented in Figure 3.20 and 3.21, Table 3.15 and 3.16.

If the codebook size used for each layer increases, the compression ratio also increases. Because, the compression ratio changes directly the number of bit used for each codebook. The compression ratio also depends on the threshold. Because, if the threshold is chosen small, the number of block in the Layer-4 increases and the total required bit also increases for all image. Additionally, if a small threshold is used to compress, the details of image are made more visible and reconstructed images approximate original images.



(a) The “Tiffany” image with 256 gray-level and 256x256 pixels.



(b) SNR=25,14 MSE=199
CR=0,4775 (16,75:1)
Threshold =4
L1=2 codebook L2=4
L3=8 L4=16

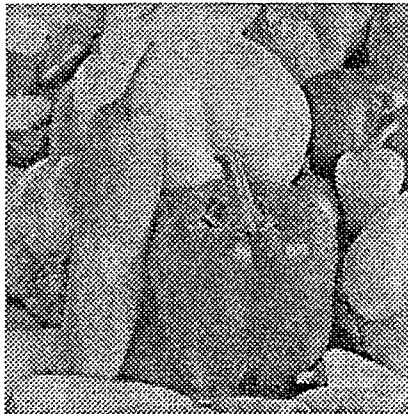


(c) SNR=25,03 MSE=204
CR=0,4325 (18,49:1)
Threshold =5
L1=2 codebook L2=4
L3=8 L4=16

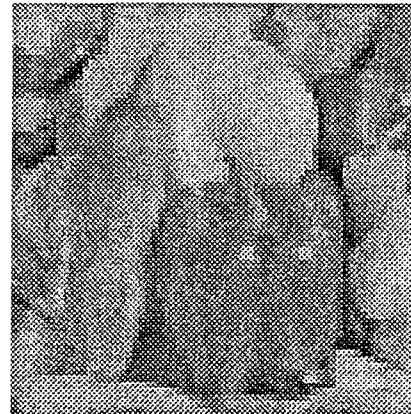


(d) SNR=24,93 MSE=209
CR=0,41 (19,51:1)
Threshold =6
L1=2 codebook L2=4
L3=8 L4=16

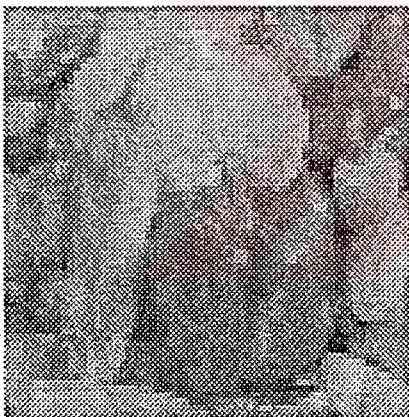
Figure 3.20 The images compressed and decompressed by using HFSVQ at different threshold values and the same codebook design.



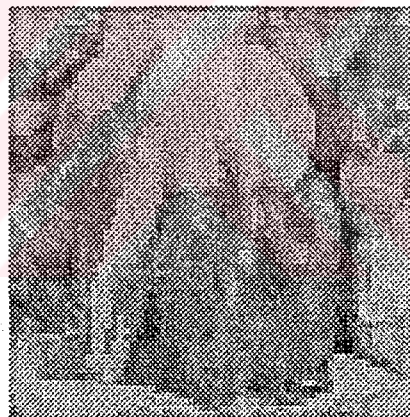
(a) The "Peppers" image with 256 gray-level and 256x256 pixels.



(b) SNR=26,14 MSE=158
 CR=0,775 (10,32:1)
 Threshold =4
 L1=4 codebook L2=4
 L3=32 L4=64



(c) SNR=26,06 MSE=161
 CR=0,69 (11,59:1)
 Threshold =4
 L1=4 codebook L2=4
 L3=16 L4=32



(d) SNR=24,49 MSE=231
 CR=0,5225 (15,31:1)
 Threshold =4
 L1=2 codebook L2=4
 L3=8 L4=16

Figure 3.21 The images compressed and decompressed by using HFSVQ at different codebook sizes and the same threshold values.

Table 3.15 PSNR values versus compression ratio at different threshold values for various images for HFSVQ.

dB/bpp	L1=1; L2=2; L3=3; L4=5 bit Threshold=4;		
	PSNR(dB)	CR(bpp)	Compression Ratio
Lenna	24,85	0,48	16,67
Peppers	24,49	0,5225	15,31
View	21,41	0,67	11,94
Room	23,15	0,5575	14,35
Tiffany	25,14	0,4775	16,75
dB/bpp	L1=1; L2=2; L3=3; L4=5 bit Threshold=5;		
	PSNR(dB)	CR(bpp)	Compression Ratio
Lenna	24,17	0,4325	18,50
Peppers	24,01	0,46	17,39
View	21,33	0,6525	12,26
Room	22,78	0,5125	15,61
Tiffany	25,03	0,4325	18,50
dB/bpp	L1=1; L2=2; L3=3; L4=5 bit Threshold=6;		
	PSNR(dB)	CR(bpp)	Compression Ratio
Lenna	23,79	0,41	19,51
Peppers	23,51	0,4275	18,71
View	21,28	0,6475	12,36
Room	22,48	0,4825	16,58
Tiffany	24,93	0,41	19,51

Table 3.16 PSNR values versus compression ratio at different codebook size and same threshold values for various images for HFSVQ.

dB/bpp	L1=2; L2=2; L3=4; L4=5 bit Threshold=4;		
	PSNR(dB)	CR(bpp)	Compression Ratio
Lenna	26,52	0,65	12,31
Peppers	26,06	0,69	11,59
View	21,64	0,8125	9,85
Room	23,93	0,71	11,27
Tiffany	26,89	0,6575	12,17
dB/bpp	L1=2; L2=2; L3=5; L4=6 bit Threshold=4;		
	PSNR(dB)	CR(bpp)	Compression Ratio
Lenna	26,55	0,72	11,11
Peppers	26,14	0,775	10,32
View	21,69	0,955	8,38
Room	23,98	0,805	9,94
Tiffany	26,93	0,73	10,96
dB/bpp	L1=3; L2=4; L3=5; L4=6 bit Threshold=4;		
	PSNR(dB)	CR(bpp)	Compression Ratio
Lenna	29,15	0,98	8,16
Peppers	29,45	1,0225	7,82
View	25,09	1,17	6,84
Room	26,78	1,0575	7,57
Tiffany	28,09	0,9775	8,18

3.3.6 Wavelet Compression

Considerable interest has arisen in recent years regarding new transform techniques that specifically solve the problems of image compression and other analysis [17]. These techniques are grouped into the headings of multi-resolution analysis time-frequency analysis and wavelet transforms. In this chapter, the wavelet compression on an image will be illustrated.

For many decades, scientists have wanted more appropriate functions than the sinus and cosines, which comprise the bases of Fourier analysis, to approximate choppy signals. The sinus and cosines functions are not suitable for approximating sharp spikes. But with wavelet analysis, it can be used for approximating functions that are contained in finite domain. Wavelet is well suited for approximating data with sharp discontinuities.

The Fourier Transform and Wavelet Transform are both linear operations that generate a data structure. The mathematical properties of the matrices involved in the transform are similar as well. The inverse transform matrix for both the Fast Fourier and Discrete Wavelet Transform is the transpose of the original one.

Result of both transforms can be viewed as a rotation in function space to a different domain. For Fast Fourier Transform, this new domain contains basis functions that are sinus and cosines. For the wavelet transform, this new domain contains more complicated basis functions called mother wavelets.

Wavelet compression is an excellent tool for data compression. For example; the FBI has standardised the use of wavelets in digital fingerprint image compression. The compression ratios are on the order of 20:1, and the difference between the original and the decompressed image is very little.

3.3.6.1 Wavelet Transform

Wavelets are functions that satisfy certain requirements. The name wavelet comes from the requirement that they should integrate to zero waving above and below the x-axis. There are a lot of different wavelet functions as shown in the Figure 3.22.

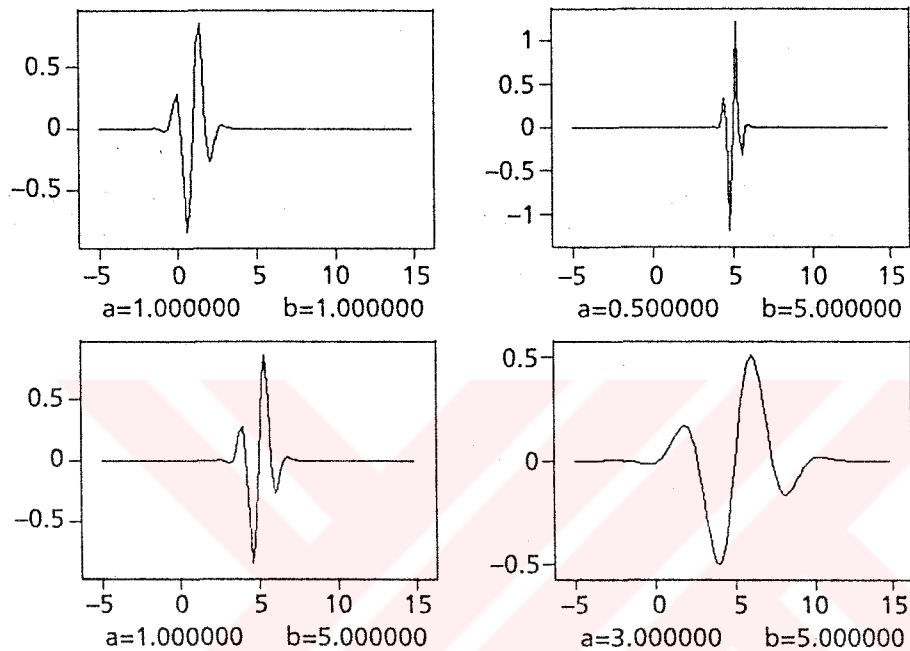


Figure 3.22 Different kind of continuous wavelet functions.

Once mother wavelet $\Psi(x)$ is fixed, and the dilation of the mother wavelet is;

$$\Psi_{(s,t)}(x) = g\left(\frac{x-t}{s}\right) \quad (3.17)$$

where s and t are integers. It is convenient to take special values for t and s in defining the wavelet basis. $s = 2^j$ and $t=k$ where j and k are integers as well. The choice of s and t is called critical sampling.

For a given function of set of sampled data $f(t)$, the Wavelet decomposition can be written as;

$$f(t) = \sum_{i=0}^S \sum_{j=0}^T c_{ij} w_{ij}(t) \quad (3.18)$$

which is a linear combination of the w_{ij} functions, where c_{ij} is Wavelet coefficients.

The basic one-dimensional wavelet transform for continuous function is;

$$W_{f(t)}(j, k) = \int f(t) w_{(j,k)}(t) dt \quad (3.19)$$

This gives the wavelet coefficient at $W(s, t)$, so the calculation will be repeated for each value of s and t . The result is a two dimensional function or a set of two-dimensional data points in the discrete case.

The inverse wavelet transform is used to reconstruct the function from its wavelet representation. This is shown in Equation 3.18 in general case;

$$f(t) = C_0 \iint c_{ij} w_{ij}(t) dj di \quad (3.20)$$

To explain how wavelets work, the simplest and oldest of all wavelets, Haar Wavelet is chosen.

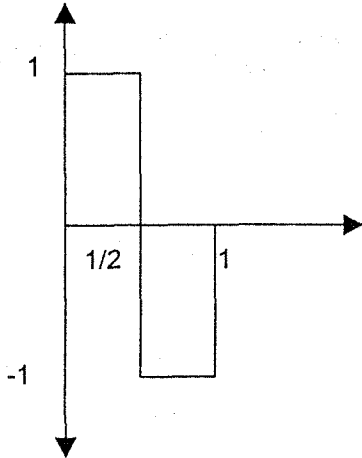


Figure 3.23 A Haar wavelet function.

The scaled and translated Haar wavelet is described as

$$\Psi_{jk}(t) = \Psi_H(2^j t - k) \quad (3.21)$$

The sample of these wavelets can be shown as graphs in Figure 3.24.

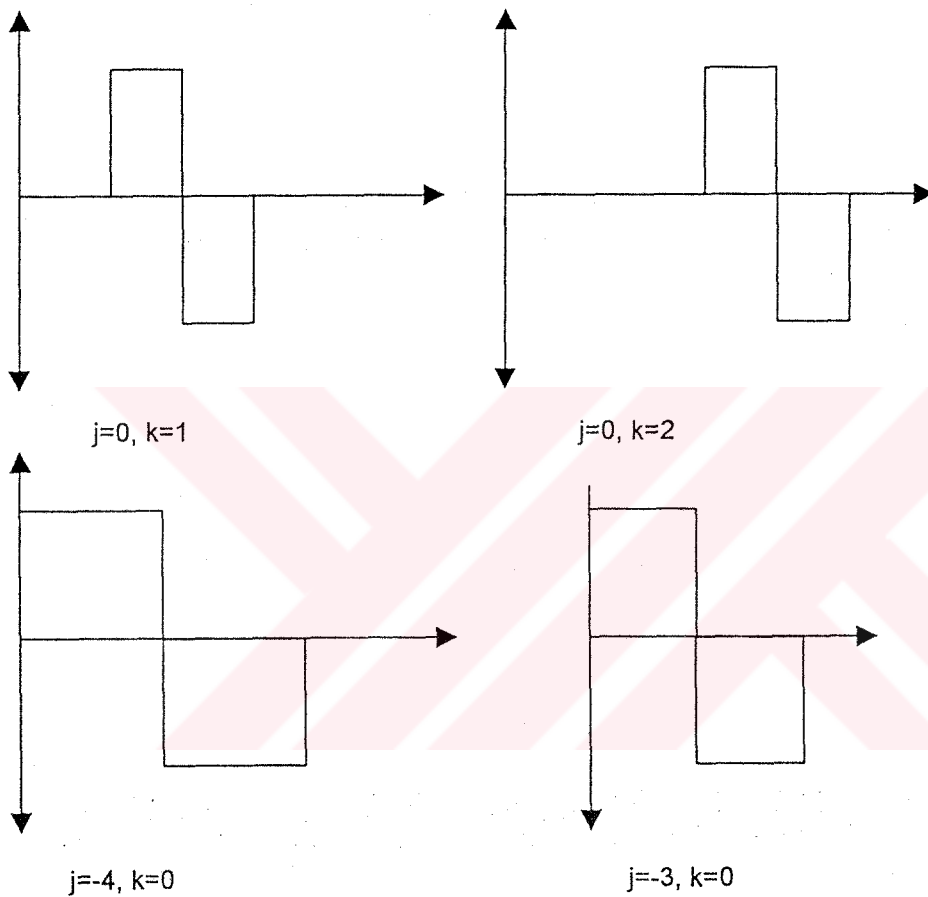


Figure 3.24 Scaled and translated Haar wavelet functions.

It is common knowledge in mathematical circles that a continuous function can be approximated by these Haar functions in a way similar to the use of sine and cosine functions in the Fourier Series approximations. A linear combination of Haar function is constructed, whose sum approximates the required function is:

$$f(t) = \sum_{j=-\infty}^{\infty} A_j \sum_{k=-\infty}^{\infty} B_{jk} \Psi_H(2^j t - k) \quad (3.22)$$

which has the same appearance as Equation 3.20 but with $c_{ij} = A_j B_{jk}$.

By assuming that the highest frequency present is 2^{-1} , which corresponds to a scale of 2. The first step in the determination of the coefficients c_{ij} is to factor out the highest frequencies, at the $j=-1$ level.

For any problem that is to be solved using computer, the bounds of the sums will be finite. For image processing and vision purposes, the function will consist of regularly sampled values, the grid being imposed by the digitisation process.

What is called a wavelet transform with respect to the Haar basis is really the calculation of the values for c_{jk} . According to Equation 3.20, with modifications to Haar basis, these can be written as below:

$$c_{jk} = 2^j \int_{2^{-j}k}^{2^{-j}(k+1)} f(t) \Psi_{jk}(t) dt \quad (3.23)$$

The series in Equation 3.22 can be rewritten as:

$$f(t) = \sum_{j=-\infty}^{-1} \sum_{k=-\infty}^{\infty} c_{jk} \Psi_{jk}(t) \quad (3.24)$$

The coefficients of the highest frequency layer all have $j=-1$; solving Equation 3.24 using Haar wavelets gives:

$$c_{-1,k} = -\frac{1}{2} \int_{2k}^{2k+1} f(t) \Psi_{-1,k}(t) dt = \frac{f(2k) - f(2k+1)}{2} \quad (3.25)$$

Each layer can be computed in this way, one at a time. In this instance, the relation is:

$$c_{j,k} = \frac{1}{2} (a_{j+1,2k} - a_{j+1,2k+1}) \quad (3.26)$$

$$a_{j,k} = \frac{1}{2} (a_{j+1,2k} + a_{j+1,2k+1}) \quad (3.27)$$

$c_{j,k}$ defines the difference between neighbourhood pixels in the image and is named as G function and $a_{j,k}$ is the average of the neighbourhood pixels, named as H function.

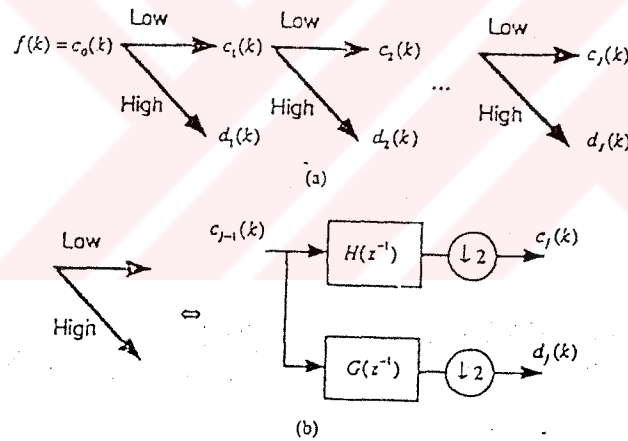


Figure 3.25 Fast implementation of the wavelet transform.

In the Figure 3.24, the fast implementation of the wavelet transform is presented. The first part of the figure shows basic principle of the algorithm is the repetitive split of the sequence $c_{j,k}$ into two halves using the low and high operations. The second part shows the implementation of the low-pass and high-pass operators using filtering and decimation by a factor of two.

Any image can be approximated by a matrix A in which the entries $A_{x,y}$ correspond to intensities of grey in the pixel (x,y) . The process of the image wavelet decomposition goes as follows: On the rows of the matrix A , the filters H and G are applied. Two resulting matrices are obtained: $H_r A$ and $G_r A$, both dimension $2^n \times 2^{n-1}$ (subscript r suggest that the filters are applied on rows of the matrix A).

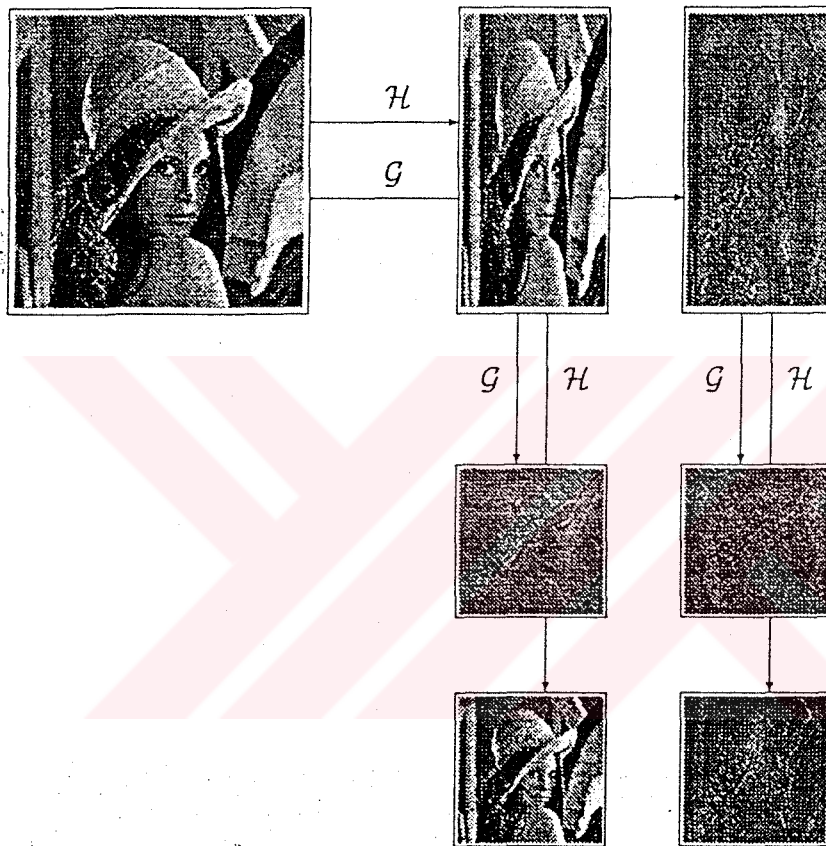


Figure 3.26 Lenna image wavelet decomposition

3.3.6.2 The Performance Result of Wavelet Transform

In this study, sample images are analysed according to the Wavelet transform, and reconstructed images are presented Figure 3.27, Figure 3.28 and Table 3.17. The differences between the images are examined.



(a) The "Madrill" image with 256 gray-level and 256x256 pixels.



(b) SNR=24,76 MSE=217
CR=2bpp (4:1)

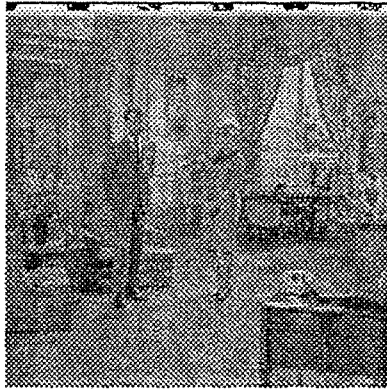


(c) SNR=22,48 MSE=367
CR=0,5bpp (16:1)

Figure 3.27 The decompressed "Madrill" images with Wavelet compression.

Table 3.17 PSNR versus compression ratio for images compressed with Wavelet compression.

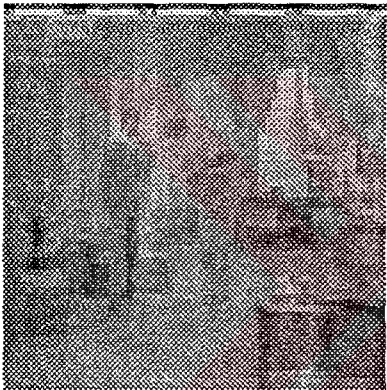
bpp/dB	2bpp	0,5bpp	bpp/dB	2bpp	0,5bpp
Palm1	24,40	21,47	Peppers	28,79	24,97
Lightnou	23,90	21,00	View	22,95	20,40
Building	27,64	23,26	Room	27,72	21,98
Text	24,44	19,61	Madrill	24,77	22,48
Lenna	29,74	26,17	Tifanny	28,26	24,57



(a) The "Room" image with 256 gray-level and 256x256 pixels.



(b) SNR=27,72 MSE=110
CR=2bpp (4:1)



(c) SNR=21,98 MSE=412
CR=0,5bpp (16:1)

Figure 3.28 The "Room" images compressed and decompressed by Wavelet compression technique.

3.4 Comparison of the Data Compression Techniques

The compression performance results are presented for each compression technique individually. According to the given results, the compression techniques are compared to each other.

Table 3.18 Comparison of Discrete Fourier Transform Coding and Discrete Cosine Transform Coding at same compression ratios.

bpp/dB	0,75 bit /pixel		1,5 bit /pixel	
	DFT	DCT	DFT	DCT
Palm1	21,48	21,37	22,48	22,57
Lightnou	20,80	19,94	21,86	20,32
Lenna	25,80	26,99	26,73	28,89
Peppers	24,57	26,06	25,83	28,84
Tiffany	24,71	22,34	26,67	24,03

As shown in Table 3.18, when discrete Fourier transform coding and discrete cosine transform coding are applied to images, the signal-to-noise ratios are close to each other at the same compression ratio. For both techniques, at high compression ratios, good performance results are obtained.

According to the Table 3.19, at high compression ratios vector quantization with 2x2 codevector gives better performance results compared to vector quantization with 1x2 codevector. But vector quantization with 2x2 codevector requires more memory size than the other.

Table 3.19 Comparison of vector quantization with 1x2 codevector and 2x2 codevector at same compression ratio.

bpp/dB	1 bit /pixel		1,5 bit /pixel	
	1x2	2x2	1x2	2x2
Palm1	22,02	23,95	25,06	24,24
Lightnou	22,01	23,57	24,75	23,79
Building	20,98	27,02	23,85	27,56
Text	20,74	24,05	23,54	24,33
Lenna	26,73	29,27	29,44	29,62
Peppers	24,69	28,13	28,69	28,64
Madrill	24,33	24,53	26,52	24,71
Room	23,43	26,70	26,43	27,52

If the DCT and DFT compression techniques are compared to vector quantization, it is seen that the vector quantization has better performance.

By using hierarchical finite-state vector quantization, more high compression ratios are obtained like 0,48bit/pixel, when the same signal-to-noise ratio is used with the other compression techniques.

Wavelet compression technique gives good performance results at low compression ratio for example 2bit/pixel.

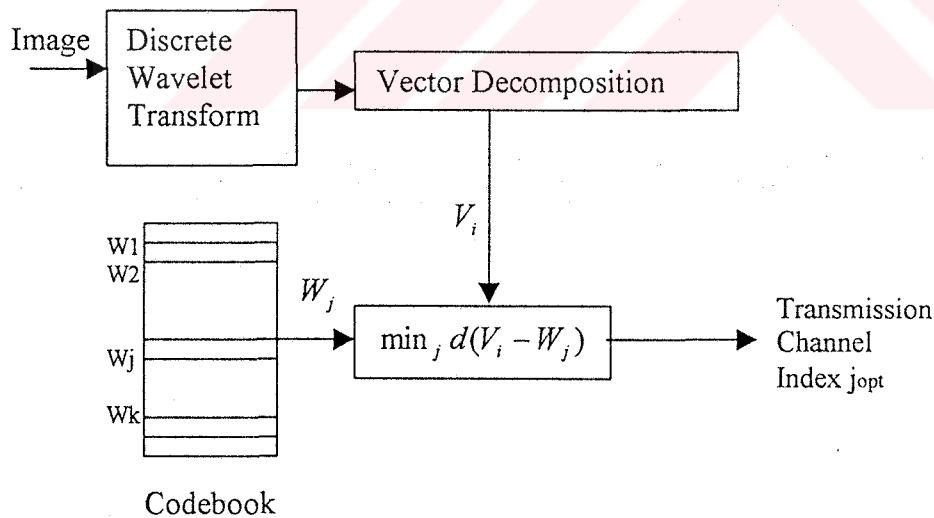
CHAPTER FOUR

THE HYBRID COMPRESSION TECHNIQUES

4.1 An Introduction of Hybrid Wavelet Transform and Vector Quantization Compression Technique

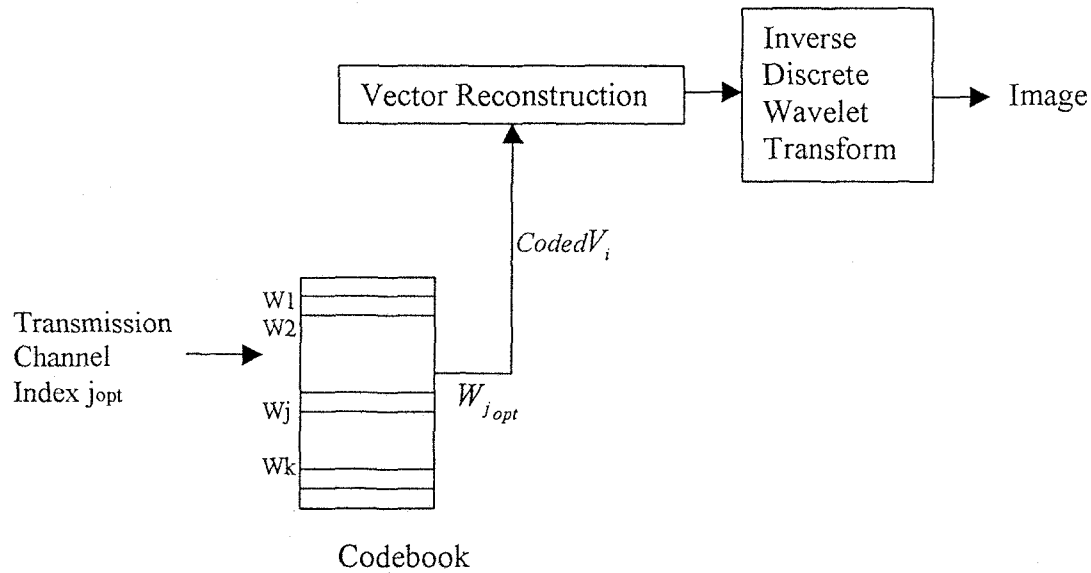
In this thesis, the vector quantization method and wavelet transform methods are individually simulated and also the performance results are given in the previous chapter. In this chapter, the two methods are used together and as a result of the simulation, better performance results are obtained.

The encoder and decoder block diagram for the Hybrid Wavelet Transform and Vector Quantization Compression Technique is shown in Figure 4.1.



(a) Decoder

Figure 4.1 The Block Diagram of Hybrid Wavelet Transform and Vector Quantization.



(b) Decoder

Figure 4.1 (cont.)

By using the decoder and encoder block diagrams of the hybrid wavelet transform and vector quantization compression method is examined and the performance results are obtained.

4.1.1 The performance result of Hybrid Wavelet Transform and Vector Quantization

Firstly, the wavelet compression is applied to the image, and compressed image is obtained. After, vector quantization technique is applied to this image. Some compressed and decompressed images are presented in Figure 4.2 and 4.3. Moreover, the sample images are given in Table 4.1 and 4.2 for different codebook size and codevector size.

As shown in tables, when the small codebook size is used, the high compression ratio is obtained but the signal-to-noise ratio decreases. If the codevector size becomes small, the signal-to-noise ratio increases however; the compression ratio becomes low. The optimum configuration is chosen according to applications.



(a) The "Lenna" image with 256 gray-level and 256x256 pixels.



(b) 32 codebook size
1x2 vector size
4:1 wavelet comp.
SNR = 26,34 MSE = 151
CR = 0,625 bpp (12,8 :1)

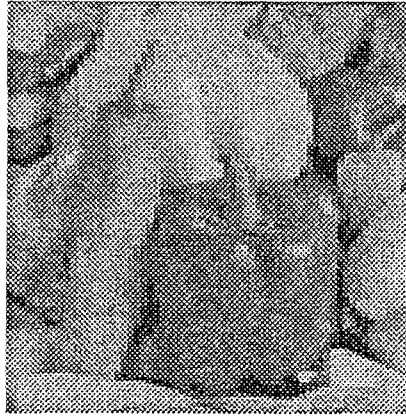


(c) 32 codebook size
2x2 vector size
4:1 wavelet comp.
SNR = 25,68 MSE = 176
CR = 0,3125 bpp (25,6 :1)

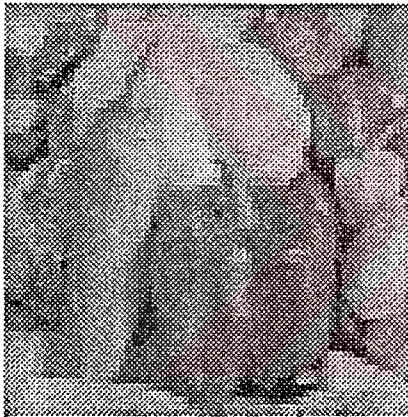
Figure 4.2 The "Lenna" compressed images by hybrid wavelet compression and vector quantization at different codevector size.



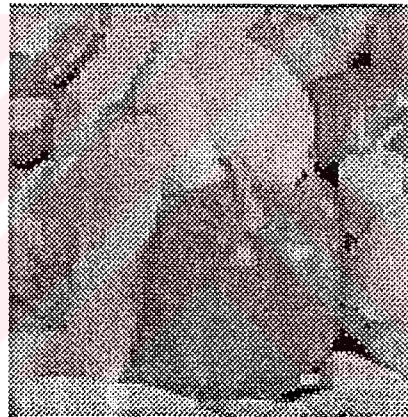
(a) The “Peppers” image with 256 gray-level and 256x256 pixels.



(b) 32 codebook size
1x2 vector size
4:1 wavelet comp.
SNR = 25,93 MSE = 166
CR = 0,625 bpp (12,8 :1)



(c) 32 codebook size
2x2 vector size
4:1 wavelet comp.
SNR = 24,71 MSE = 220
CR = 0,3125 bpp (25,6 :1)



(d) 64 codebook size
2x2 vector size
4:1 wavelet comp.
SNR = 23,61 MSE = 283
CR = 0,375 bpp (21,33 :1)

Figure 4.3 The “Peppers” compressed images by hybrid wavelet compression and vector quantization at different odevector size and codebook size.

Table 4.1 PSNR values versus compression ratio at different codevector and 32 codebook for wavelet compression with VQ.

32 codebook			
dB/bpp	(2x1 codevector) 0,625 bpp	(2x2 codevector) 0,3125 bpp	(4x4 codevector) 0,078 bpp
Lenna	26,34	25,68	21,87
Peppers	25,93	24,71	19,85
View	21,50	20,22	17,94
Tiffany	26,34	18,53	18,13
Madrill	23,20	22,38	20,24

Table 4.2 PSNR values versus compression ratio at different codevector and 64 codebook for wavelet compression with VQ.

64 codebook		
dB/bpp	(2x1 codevector) 0,75 bpp	(2x2 codevector) 0,375 bpp
Lenna	25,73	24,01
Peppers	25,23	23,61
View	21,41	19,44
Tiffany	26,09	19,07
Madrill	22,82	21,23

4.2 An Introduction of Hybrid Wavelet Transform and Hierarchical Finite State Vector Quantization Compression Technique

In this thesis, in the third chapter, the hierarchical finite state vector quantization method is simulated and the performance results are obtained for different codebook size and threshold values. In this chapter, the two methods which are wavelet

transform and hierarchical finite-state vector quantization are combined and at high compression rates, better performance results are obtained as shown in the Table 4.3.

The compression scheme of the Hybrid wavelet compression and hierarchical finite state vector quantization technique is given in Figure 4.4.

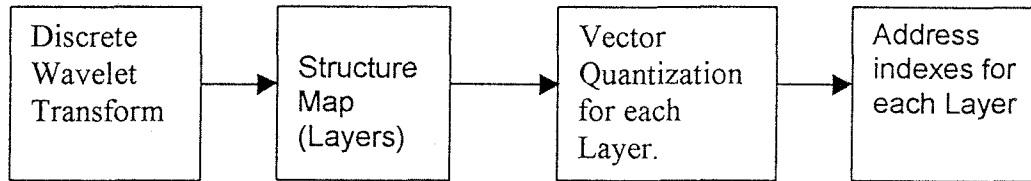


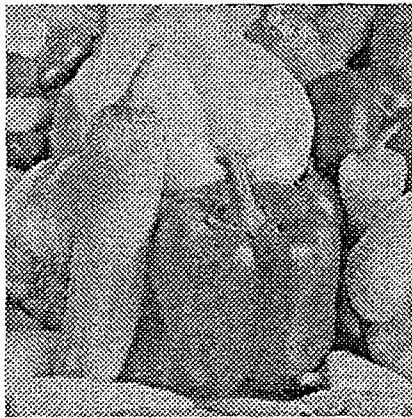
Figure 4.4 A general compression scheme of the Hybrid Wavelet Compression and Hierarchical Finite-State Vector Quantization.

In this study, the hybrid compression method is examined by using the compression block diagram as shown in Figure 4.4, and the performance results are obtained.

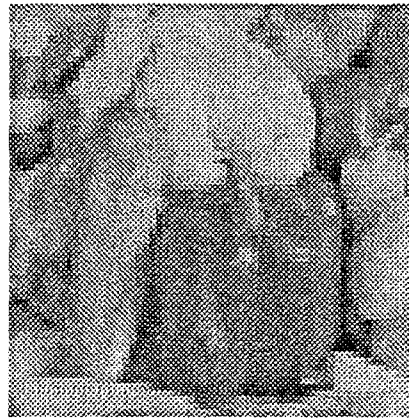
4.2.1 The performance result of Hybrid Wavelet Transform and Hierarchical Finite State Vector Quantization

Firstly, the wavelet compression is applied to the image, and compressed image is obtained. After, Hierarchical Finite State vector quantization compression is applied to this image. In Figure 4.5 and 4.6, compressed and decompressed sample images are presented at different compression ratio. To examine the effect of the codebook size of the compression ratio and signal-to-noise ratio, different codebook size are used. The results are presented in Table 4.3.

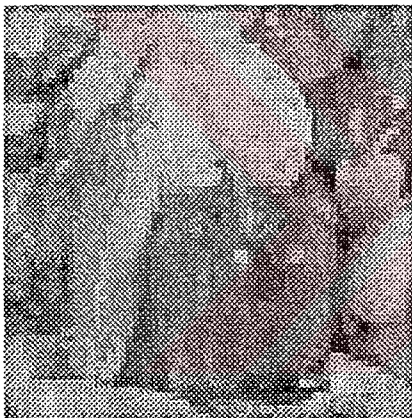
As shown in tables, if small codebook size is used for each layer, the high compression ratio is obtained. However, the signal-to-noise ratio decreases. Four layers are and 2x2 codevector is used for each layer while simulating this hybrid compression technique.



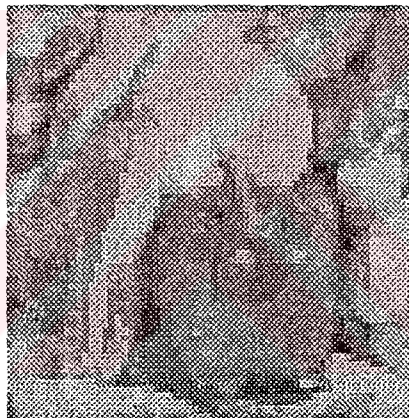
(a) The “Peppers” image with 256 gray-level and 256x256 pixels.



(b) L1=8 codebook L2=16
 L3=32 L4=64
 2x2 vector size 4:1 wavelet
 SNR = 24,75dB MSE = 218
 CR = 0,2553 bpp (31,34:1)



(c) L1=4 codebook L2=4
 L3=16 L4=32
 2x2 vector size 4:1 wavelet
 SNR = 23,54dB MSE =
 CR = 0,1723bpp (46,43:1)



(d) L1=2 codebook L2=4
 L3=8 L4=16
 2x2 vector size 4:1 wavelet
 SNR = 22,83dB MSE =
 CR = 0,1303 bpp (61,40:1)

Figure 4.5 The “Peppers” image compressed a hybrid wavelet compression and hierarchical finite-state vector quantization method with different configurations.



(a) The "Tiffany" image with 256 gray-level and 256x256 pixels.



(b) L1=8 codebook L2=16
 L3=32 L4=64
 2x2 vector size 4:1 wavelet
 SNR = 24,13dB MSE = 169
 CR = 0,2444bpp (32,73:1)



(c) L1=4 codebook L2=4
 L3=16 L4=32
 2x2 vector size 4:1 wavelet
 SNR = 22,47dB MSE = 215
 CR = 0,1642bpp (48,72:1)



(d) L1=2 codebook L2=4
 L3=8 L4=16
 2x2 vector size 4:1 wavelet
 SNR = 23,54dB MSE = 288
 CR = 0,1194bpp (67,0:1)

Figure 4.6 The "Tiffany" image compressed by using hybrid wavelet compression with hierarchical finite-state vector quantization method by using various configurations.

Table 4.3 PSNR values versus compression ratio at different codebook sizes and same threshold for wavelet compression with HFSVQ.

dB/bpp	L1=1; L2=2; L3=3; L4=4 bit Threshold=4; 4:1 wavelet comp.		
	PSNR(dB)	CR(bpp)	Compression Ratio
Lenna	24,24	0,1199	66,72
Peppers	22,83	0,1303	61,40
Room	20,56	0,1392	57,47
Tiffany	23,54	0,1194	67,00
dB/bpp	L1=2; L2=2; L3=4; L4=5 bit Threshold=4; 4:1 wavelet comp.		
	PSNR(dB)	CR(bpp)	Compression Ratio
Lenna	24,81	0,1626	49,20
Peppers	23,54	0,1723	46,43
Room	20,84	0,1773	45,12
Tiffany	22,47	0,1642	48,72
dB/bpp	L1=3; L2=4; L3=5; L4=6 bit Threshold=4; 4:1 wavelet comp.		
	PSNR(dB)	CR(bpp)	Compression Ratio
Lenna	25,85	0,2449	32,67
Peppers	24,75	0,2553	31,34
Room	21,72	0,2642	30,28
Tiffany	24,13	0,2444	32,73

CHAPTER FIVE

CONCLUSION

The principal objective of this thesis is to present data compression techniques for images and to describe the most commonly used compression methods. So, the most popular compression techniques were examined, simulated and the performance results were obtained according to signal to noise ratio and compression ratio. In addition, a hybrid wavelet transform and vector quantization was introduced as a new technique.

If the compression ratio is low, Run-Length Transform becomes more effective by means of mean square error. However, if the compression ratio increases, the mean square error and difference between the reconstructed image and original image increases. As the Run-Length Coding is lossless compression technique, it is convenient to use it on texts or documents then graphics for low rate compression.

The vector quantization and hierarchical finite-state vector quantization methods give better results at high compression ratio for instance 1 bit/pixel. Getting better results with increasing codebook size causes more memory size. Also increasing vector size yields worse performance results, though. So, optimum vector size and optimum codebook size must be chosen according to the desired hardware and compression ratio.

In this study, it is seen that as a result of wavelet transform also get high compression ratio and low mean square error like vector quantization and hierarchical finite-state vector quantization.

In this thesis, hybrid compression methods were examined. After wavelet compression, vector quantization technique was applied to the images, and good results were obtained compared to wavelet transform and vector quantization individually at high compression rates. Additionally, after wavelet compression, hierarchical finite-state vector quantization also was applied to the images and high signal to noise ratios were obtained at high compression rates. The signal to noise ratio versus compression ratio for each compression method were obtained and the performance results were compared to each other.



REFERENCES

- [1] Averbuch, Amir & Lazar, Danny & Israeli, Moshe. (1996). Image Compression Using Wavelet Transform and Multiresolution Decomposition. IEEE Transactions on Image Processing, 5, 4-15
- [2] Awcock, G.J. & Thomas, R. (1996). Applied Image Processing. McGraw Hill International Editions, Singapore.
- [3] Baxes, Gregory A. (1994). Digital Image Processing: Principles and Applications. John Wiley, New York.
- [4] Castleman, Kenneth R. (1996). Digital Image Processing. Prentice Hall, New Jersey.
- [5] Chang, Pao-Chi & Gray, M. Robert & May, Jack. (1987). Fourier Transform Vector Quantization for Speech Coding. IEEE Transaction on Communications, 35, 1059-1067.
- [6] Dionysian, Raffi. (1994). Variable-Precision Arithmetic for Vector Quantization. University of California, Los Angeles.
- [7] Efstratiadis, Serafim N. & Tzovaras, Dimitrios & Strintzis, Michael G. (1996). Hierarchical Partition Priority Wavelet Image Compression. IEEE Transactions on Image Processing, 5, 1111-1123.
- [8] Gonzales, Rafael C. & Richard E. (1992). Digital Image Processing. Addison-Wesley Publishing Company, New York.
- [9] Graps, Amara. (1995). An Introduction to Wavelets. IEEE Computational Science and Engineering, vol 2.
- [10] Gray, Robert M. (1994). Vector Quantization. IEEE ASSP Magazine, pp 4-23.

- [11] Hilton, Michael L. & Jaweth, Bjorn D. & Sengupta, Ayan. (1994). Compressing Still and Moving Images with Wavelet. Multimedia Systems, vol 2.
- [12] Kovacevic, Jelena & Cohen, Albert. (1996). Wavelets: The Mathematical Background. Proceeding of the IEEE.
- [13] Lin, Y. (1989). Digital Image Processing Applications. The Society Bellingham, Los Angeles.
- [14] Nasrabadi, Nasser M. & King, Robert A. (1988). Image Coding Using Vector Quantization: A Review. IEEE Transactions on Communications, 36, 957-971.
- [15] Nelson, Mark & Gailly Jean-Loup. (1995). The Data Compression Book. M&T Books, New York.
- [16] Panchanathan, Sethuraman & Gamaz, Nadja & Jain, Amit. (1996). Image Scalability Using Wavelet Vector Quantization. Journal of Electronic Imaging, 5, 167-175.
- [17] Parker, J.R. (1997). Algorithms for Image Processing and Computer Vision. John Wiley & Sons, Inc., New York.
- [18] Ramakrishnan, Sangeeta & Rose, Kenneth & Gersho, Allen. (1998). Constrained-Storage Vector Quantization with a Universal Codebook. IEEE Transactions on Image Processing, 7, 785-793.
- [19] Ruttiman, Urs E. & Unser, Michael. (1998). Statistical Analysis of Functional MRI Data in the Wavelet Domain. IEEE Transactions on Medical Imaging, 17, 142-154.
- [20] Sid-Ahmed, M. A. (1994). Image Processing Theory, Algorithms, & Architectures. McGraw Hill International Editions, Singapore.
- [21] Silva, Eduardo & Sampson, Demetrios G. & Ghanbari, Mohammed. (1996). A Successive Approximation Vector Quantizer for Wavelet Transform Image Coding. IEEE Transaction on Image Processing, 5, 299-310.

- [22] Sitaram, Vijay S. & Huang, Chien-Min & Israelsen, Paul D. (1994). Efficient Codebooks for Vector Quantization Image Compression with an Adaptive Tree Search Algorithm. IEEE Transactions on Communications, 42, 3027-3033.
- [23] Yu, Ping & Venetsanopoulos, Anastasios N. (1994). Hierarchical Finite-State Vector Quantization for Image Coding. IEEE Transactions on Communications, 42, 3020-3026.

