National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Canada

Scalable Image Compression for Database Storage and Transmission Applications

Amit Jain, B.Tech

A THESIS
submitted to the School of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the Degree of
MASTER OF APPLIED SCIENCE
in
Electrical Engineering

Ottawa-Carleton Institute of Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
University of Ottawa
OTTAWA, ONTARIO K1N 6N5

ISBN 0-612-00607-7

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

To my Grandparents,

I hereby declare that I am the sole author of this thesis.

I authorize the University of Ottawa to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Amit Jain

I further authorize the University of Ottawa to reproduce this thesis by photocopying or other means, in total or part, at the request of other institutions or individuals for the purpose of scholarly research.

Amit Jain

Acknowledgements

First, I would like to thank my supervisor Dr.S.Panchanathan for introducing me to the field of image and video compression and his support and encouragement during my thesis work.

I would like to thank all the past and present members of the Multimedia Communications Research Laboratory, especially Rakesh Gandhi, Xiping Wang and Omid Fatemi for their help and co-operation.

My special thanks are due to all the support staff members of Electrical Engineering for their help, especially Michele Roy, Suzanne St-Michel and Lucette Lepage.

I am truly grateful to my family for their consistent support, without which this work would not have been possible.

I am also thankful to the Canadian Institute of Telecommunications Research (CITR) and the Natural Sciences and Engineering Research Council (NSERC) of Canada for their financial support.

Abstract

Many image and video applications are targeted to a multiuser environment and therefore require a flexible compression algorithm to cope with the varying storage, computational and display resources of the devices connected to the system. In addition to the basic feature of image compression, many interactive applications require special features such as the capability of browsing vast global databases to locate an retrieve an image of interest. On the other hand, some applications involve transmission of images over low bandwidth channels or a dual priority transmission medium where the low priority channel may be subject to a loss of bandwidth. These applications require the image to be reconstructed at the best possible quality using the available bandwidth.

Scalable image compression provides the capability of image reconstruction in different sizes and picture qualities. The image can be simultaneously displayed at multiple receivers, each with a different resolution. For interactive applications, the image may be first displayed at a smaller size and/or lower picture quality. Based on the viewer's request, this image can be further enhanced until it is reconstructed at the full size and picture quality. For applications involving image transmission over low bandwidth channels, scalable image compression reduces the size of the compressed bitstream so that a low quality version of the image can be reconstructed at the receiver in real time.

In this thesis, we first define the concept of scalability and describe its three types, namely SNR, spatial and temporal scalability. This is followed by a review of the various techniques reported in the literature for achieving scalable image compression. We then investigate the algorithms for scalable image compression within the framework of JPEG standard. An extension to the hierarchical mode algorithm of JPEG is presented which provides an improved coding performance. This is followed by evaluation of the performance of these algorithms for image transmission over ATM networks.

JPEG provides an efficient technique for achieving SNR scalability. However, JPEG algorithms do not provide a good coding performance for achieving spatial scalability at low bit rates. We propose a high performance algorithm based on wavelet transform and vector quantization for achieving spatial scalability. This algorithm ensures partial decodability of VQ labels by using multiple codebooks for image reconstruction, one for each spatial resolution. Our simulations indicate that the proposed algorithm provides sptial scalability at lower bit rates compared to the hierarchical mode of JPEG.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Image compression is becoming increasingly popular in storage and transmission applications. Examples include videoconferencing, multimedia databases, telepresence, virtual museum, interactive television, mobile multimedia communications, etc. The goal of image compression is to reduce the bit rate for transmission or storage while maintaining an acceptable quality in the reconstructed images. A number of image coding techniques have been developed for this purpose which exploit the psychovisual as well as the statistical redundancies in the image data to provide compression.

1.1 Scalable Image Compression

Image and video applications are often targeted to a multiuser environment where the image data may be accessed by a large number of display and transmission devices, each with a different resolution or quality requirement, storage and processing capabilities. This requires a flexible compression algorithm to cope with the varying computational and display resources of each device or system. In addition to the basic feature of image compression, many interactive applications require special features, such as the capability

of browsing large image databases to locate and retrieve an image of interest. Hence, the compression algorithm should not only provide good reconstruction but should also be capable of extracting the gross structural information of the images at a higher compression (i.e. lower bit rates). On the other hand, many applications involve transmission of images over low bandwidth channels (e.g. mobile communications) or a dual-priority transmission medium (e.g. ATM networks) where the low priority channel may be subject to a loss of bandwidth. These applications require the image to be reconstructed at the best possible quality using the available bandwidth. This necessitates a compression algorithm which not only reorganizes the image data but is also able to match the size of the compressed bitstream to the available bandwidth for real time transmission.

Scalable image compression provides the capability of image reconstruction in different sizes and picture qualities by reorganizing the image data into multiple layers. In other words, the image can be simultaneously displayed at multiple receivers, each with a different resolution. For interactive (database) applications, the image may be first displayed at a low spatial resolution and/or picture quality. Based on the viewer's request or the application requirements, this image is further enhanced until it is reconstructed at the full size and picture quality. In the case of applications involving image transmission over low bandwidth channels, scalable image compression reduces the size of the compressed bitstream by reorganization of the image data so that a low quality version of the image can be reconstructed at the receiver in real time. Similarly, scalable compression provides a graceful degradation in performance for image transmission over dual priority channels in the event of loss of bandwidth in the lower priority channel.

1.2 Investigated approach

Recently, the Joint Photographic Experts Group (JPEG) of the International Standards Organization (ISO) has proposed the JPEG standard for still image compression. JPEG

provides a framework for high quality compression and reconstruction of continuous-tone grayscale and color images for a wide range of applications. It specifies a toolkit of algorithms for converting the raw image data into compressed form and the corresponding decompression algorithms for reconstructing the image. JPEG has four modes of operation, namely baseline sequential, losless compression, progressive coding and hierarchical coding. The baseline sequential algorithm provides the basic feature of Discrete Cosine Transform (DCT) based lossy image compression, while the lossless compression uses a Differential Pulse Code Modulation (DPCM) based algorithm to achieve perfect image reconstruction. The progressive coding mode proposes two different techniques for achieving quality (SNR) scalability, where SNR scalability refers to image reconstruction in progressively improving picture qualities. On the other hand, the hierarchical coding mode uses a pyramidal coding structure to provide spatial scalability, where spatial scalability refers to image representation in different sizes or spatial resolutions.

In this thesis, the problem of scalable image compression is addressed from two perspectives. We first investigate the JPEG standard and evaluate the algorithms proposed in the progressive and hierarchical modes. We then present extensions to the hierarchical mode to achieve an improved compression performance. This is followed by the evaluation of these algorithms for scalable image transmission over broadband networks using asynchronous transfer mode (ATM).

We note that the compression algorithm of the hierarchical mode of JPEG does not provide a good performance at low bit rates. A high performance algorithm based on the wavelet transform and vector quantization has been reported in the literature for low bit rate image compression. Although, wavelet transform provides a scalable image representation, vector quantization is not scalable. As a first step towards making this algorithm scalable, a new technique is proposed in this thesis to achieve spatial scalability using vector quantization. This technique does not require any overhead bits to provide a scalable image bitstream. We then apply this technique to the combined wavelet transform and vector quantization based

compression algorithm. The proposed algorithm not only achieves a high compression performance, but also provides spatial scalability without requiring any overhead bits.

1.3 Thesis Organization

The thesis is organized as follows: Chapter 2 reviews the basic concepts of source coding theory and presents some of the lossy and lossless image compression techniques reported in the literature. This follows a review of the JPEG standard for still image compression, vector quantization and wavelet transform based image coding. Chapter 3 defines the concept of scalability and describes its three types. This is followed by a review of the various spatial and transform domain techniques for achieving scalable image compression. Chapter 4 presents scalable image compression algorithms within the framework of JPEG standard. Chapter 5 introduces two new techniques based on vector quantization for scalable image compression. Chapter 6 presents the conclusions and suggestions for further research.

1.4 Main Contributions

The main contributions of this thesis are summarized below:

- Investigate and evaluate the scalable image compression techniques within the framework of JPEG standard.
- Present an extension to the JPEG hierarchical mode algorithm which provides an improved coding performance.
- Evaluate the performance of JPEG algorithms for scalable image transmission over ATM networks.
- Propose an algorithm for achieving spatial scalability using vector quantization.
- Propose a high performance technique based on wavelet transform and vector

quantization for scalable image compression.

Chapter 2

Review of Image Compression

In this chapter, we first review the basic concepts of source coding theory as applied to image compression. This is followed by a brief review of lossless and lossy image coding techniques, such as Huffman coding, run-length coding, predictive coding and transform coding. We next review the JPEG standard for still image compression followed by vector quantization and wavelet transform based image coding.

2.1 Source Coding Theory

Source coding theory deals with the compression of data generated by an information source that emits a sequence of symbols chosen from a finite alphabet. Entropy and rate distortion are fundamental concepts in source coding theory. Entropy provides a useful measure of information contained in the source data, as it determines the minimum average bit rate required for perfect reconstruction of the source symbols. Rate distortion theory provides a lower bound on the average bit rate for a given distortion in the reconstructed symbols. The details of entropy and rate distortion theory, as applied to image compression, are now presented.

2.1.1 Entropy

Let us consider an information source X generating images of size $N \times N$ pixels, where each pixel is quantized to K gray levels. A total of $K^{N \times N}$ possible image patterns can be generated by such a source. Let the probability of a specific image pattern be denoted by $p(\mathbf{x})$ where

$$\mathbf{x} = \{x_{ij}\} \quad ; \quad i, j = 0, 1, \ldots, N-1, \tag{2.1}$$

where x_{ij} is the $(i,j)th$ element of $\mathbf{x}$. The average information of the source is specified by its entropy [1] which is defined as:

$$H(X) = -\frac{1}{N \times N} \sum_{all\ \mathbf{x}} p(\mathbf{x}) \log_2 p(\mathbf{x}) \quad bits\ /\ pixel \tag{2.2}$$

It has been shown [2] that the source entropy is bounded by 0 and $\log_2 K$, that is,

$$0 \leq H(X) \leq \log_2 K \tag{2.3}$$

The left side equality holds if all probabilities $p(\mathbf{x})$ except one are zero, in which case the source is totally predictable. The right side equality holds when every source symbol $\mathbf{x}$ has an equal probability. The redundancy of the source is defined as the difference between the maximum possible value of the source entropy ($\log_2 K$) and and its actual value $H(X)$.

If every pixel in the image is statistically independent of the others, then the source probability $p(\mathbf{x})$ can be expressed as:

$$p(\mathbf{x}) = \prod_{i,j=0}^{N-1} p_{ij}(x_{ij}) \tag{2.4}$$

Here, p_{ij} represents the probability that the pixel X_{ij} of the image source X, has a value equal to x_{ij}. In this case $H(X)$ can be expressed as,

$$H(X) = -\frac{1}{N \times N} \sum_{i,j=0}^{N-1} \sum_{all\ x_{ij}} p_{ij}(x_{ij}) \log_2 p_{ij}(x_{ij}) \tag{2.5}$$

In practice, the statistical information of an image, $p(\mathbf{x})$, is however not easily measured or modeled and therefore, the true entropy of the image is, in general, very difficult to obtain.

Hence, a simpler measure, the first order entropy $H^1(X)$, which is defined on a pixel-by-pixel basis is often used. $H^1(X)$ can be defined as:

$$H^1(X) = -\sum_{k=0}^{N-1} P_k \log_2 P_k \tag{2.6}$$

where P_k is the probability of the occurence of the kth gray level value. If the pixels of the image are identically and independently distributed (i.i.d), that is:

$$p_{ij}(x_{ij}) = p_{uv}(x_{uv}), \quad for \quad x_{ij} = x_{uv} \tag{2.7}$$

then the entropy $H(X)$ of the image is equal to the first order entropy, $H^1(X)$.

$$H(X) = H^1(X) = -\sum_{k=0}^{N-1} P_k \log_2 P_k \tag{2.8}$$

The first order entropy is often referred to as *memoryless entropy* as it provides the minimum bit rate required for lossless reproduction of an image without exploiting the correlation between pixels.

2.1.2 Rate Distortion function

For lossy coding, the rate distortion function (RDF) determines the minimum distortion that can be achieved for a given average bit rate. RDF, therefore, provides an upper bound on the performance of practical coders [3]. RDF is a measure of transmission of information from the source to the receiver. The average mutual information $I_{NxN}(X,\hat{X})$ between the source and the receiver is defined as:

$$I_{NxN} = \sum_{all\ x,\hat{x}} p(x)\, p(\hat{x}/x) \log_2 \frac{p(\hat{x}/x)}{\sum_{all\ x} p(x) p(\hat{x}/x)} \tag{2.9}$$

This is a measure of the statistical dependence between the source output X and the decoded output $\hat{X}$. Since $p(x)\, p(\hat{x}/x) = p(x,\hat{x})$, it follows that

$$\sum_{all\ x} p(x)\, p(\hat{x}/x) = \sum_{all\ x} p(x,\hat{x}) = p(\hat{x}) \tag{2.10}$$

and therefore,

$$I_{NxN} = \frac{1}{N \times N} \sum_{all\ x,\hat{x}} p(x,\hat{x})\ \log_2 \frac{p(\hat{x}/x)}{p(\hat{x})} \tag{2.11}$$

It can be shown that

$$I_{NxN} = H(X) - H(X/\hat{X}) \tag{2.12}$$

where $H(X/\hat{X})$ is the entropy of the source X, given the decoder output $\hat{X}$. Hence the mutual information between X and $\hat{X}$ is the difference in the uncertainty at the source X, and the uncertainty at the source X given the decoder output $\hat{X}$. For lossless encoding, the average mutual information is given by:

$$I_{NxN} = H(X) - H(X/\hat{X}) = H(X) \tag{2.13}$$

This implies that for lossless coding, the least average bit rate is $H(X)$. On the other hand, if there is no information transmission between the encoder and the decoder, the mutual information $I_{NxN}(X,\hat{X})$ is equal to 0. Therefore, the average mutual information is lower bounded by 0 and upper bounded by the source entropy, i.e.

$$0 \le I_{NxN} \le H(X) \tag{2.14}$$

If we define a distance or distortion measure $d(x,\hat{x})$ between x and $\hat{x}$, the average distortion per pixel can be expressed as:

$$D = \frac{1}{N \times N} E[d(X,\hat{X})] = \frac{1}{N \times N} \sum_{all\ x,\hat{x}} d(x,\hat{x})\ p(x)\ p(\hat{x}/x) \tag{2.15}$$

The $N \times N$ block rate distortion function $R_{NxN}(D)$ [4] is defined as the minimum average mutual information between X and $\hat{X}$ subject to the constraint of a fixed average distortion D,

$$R_{NxN}(D) = \inf_{all\ p(\hat{x}/x)} I_{NxN}(X,\hat{X}) \tag{2.16}$$

It can be observed from equations (2.9) and (2.16) that for a given source, the minimization can only be over the conditional probability $p(\hat{x}/x)$ as the source symbol probabilities $p(x)$ are fixed.The rate distortion function $R(D)$ [4] is simply the limiting value of the $N \times N$ block rate distortion function $R_{NxN}(D)$ as the block size $N \times N \rightarrow \infty$,

$$R(D) = \lim_{NxN \rightarrow \infty} R_{NxN}(D) \tag{2.17}$$

It is to be noted that the distortion D and the mutual information $I_{NxN}(X,\hat{X})$ depend on the type of source coding technique used. However, there is a minimum I_{NxN} that is needed to ensure that the average distortion (of reconstruction) does not exceed the specified upper limit D. This minimum value is equal to $R(D)$. For exact reconstruction (D=0), the bit rate required is equal to the source entropy $H(X)$,

$$R(0) = H(X) \tag{2.18}$$

2.1.3 Distortion Measures

Source coding schemes are generally evaluated by using a distortion measure, which is essentially a cost function $d(x,\hat{x})$, for reproducing the input x by an ouput $\hat{x}$. The performance of a coding system can now be evaluated using the average distortion introduced by the coding system, $E[d(X,\hat{X})]$,

$$E[d(X,\hat{X})] = \sum_{all\ x,\hat{x}} d(x,\hat{x})\ p(x,\hat{x}) \tag{2.19}$$

In order to be valuable for both design and comparison purposes, a distortion measure must be:

(a) computable in real time for efficient evaluation;

(b) subjectively meaningful, so that it correlates well with human visual observations;

(c) tractable for proper mathematical analysis.

One of the most widely used distortion measures in image coding is the Peak Signal to Noise Ratio (PSNR) [5], which is defined as:

$$Peak\ Signal\ to\ Noise\ Ratio\ (PSNR) = 10\ \log_{10}\left(\frac{Peak\ Signal\ Value^2}{Mean\ Square\ Error}\right) \tag{2.20}$$

where,

$$Mean\ Square\ Error\ (MSE) = \frac{1}{N \times N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(x_{ij} - \hat{x}_{ij})^2 \tag{2.21}$$

The other distortion measures used in literature are the Normalised Mean Square Error (NMSE), Signal to Noise Ratio (SNR) and Mean Absolute Error (MAE) which are defined as follows:

$$NMSE = \frac{[\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(x_{ij} - \hat{x}_{ij})^2]}{[\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(x_{ij})^2]} \tag{2.22}$$

$$SNR = 10 \log_{10}(\frac{1}{NMSE}) \tag{2.23}$$

$$MAE = \frac{1}{N \times N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(|x_{ij} - \hat{x}_{ij}|) \tag{2.24}$$

where $(|x_{ij} - \hat{x}_{ij}|)$ is the absolute difference between the original image pixel x_{ij} and the reconstructed image pixel $\hat{x}_{ij}$.

All distortion measures have their relative advantages and disadvantages and none of them correlate extremely well with the subjective ratings. For example, two reconstructed images with the same PSNR values may receive very different subjective evaluations. In order to obtain a higher correlation between the objective and subjective evaluations, several new distortion measures have been recently proposed [6,7] and this field continues to be an active research area.

2.2 Lossless Compression

A high degree of correlation exists between the neighbouring pixels in natural images. Lossless compression techniques exploit this statistical redundancy in such a way that the entire process is reversible i.e. the original image is exactly recovered. There is considerable interest in lossless techniques, especially in applications which require very high fidelity reconstructed images. In this thesis, a combination of run-length and huffman coding techniques have been used for lossless encoding. This section presents a brief review of the huffman and run-length coding techniques.

2.2.1 Huffman Coding

According to Shannon's first theorem (for lossless compression), the average bit rate R for encoding an information source is lower bounded by the source entropy H. Huffman [8] has provided a practical method for lossless compression and his technique generally results in a bit rate very close to the Shannon's limit. Huffman technique designs a variable length code (VLC) for each source symbol such that the number of bits in the code is approximately inversely proportional to the probability of that symbol's occurence. Huffman codes are instantaneously decodable and their efficiency is defined as:

$$Efficiency = \frac{H}{R} \times 100\% \qquad (2.25)$$

An efficiency of 100% is obtained when the probabilities of the source symbols are negative powers of 2.

In order to build a Huffman code table for a given source, the symbols are arranged in the decreasing order of their probabilities i.e. (0.4, 0.25, 0.15, 0.15, 0.05), as shown in Fig.2.1(a). The original source is reduced in multiple stages by combining the two least probable symbols at each stage until a source with only two symbols is obtained. These two symbols are assigned huffman codes '0' and '1', respectively. The codewords for the previous reduced stage are found by appending a '0' and '1' to the least probable codeword (symbol) of the current stage as shown in Fig.2.1(b). This process is continued until the Huffman codes for the original source symbols are found.

The design of Huffman code tree depends upon the source statistics. For a source with time-varying statistics, a new code tree has to be built periodically for efficient coding. This problem is tackled by using an adaptive Huffman technique in which the encoder keeps track of the probability of the symbols emitted by the source. Both the transmitter and the receiver start with the same set of Huffman codes and modify them using the same algorithm. This ensures perfect reconstruction of the symbols without transmission of any side information.

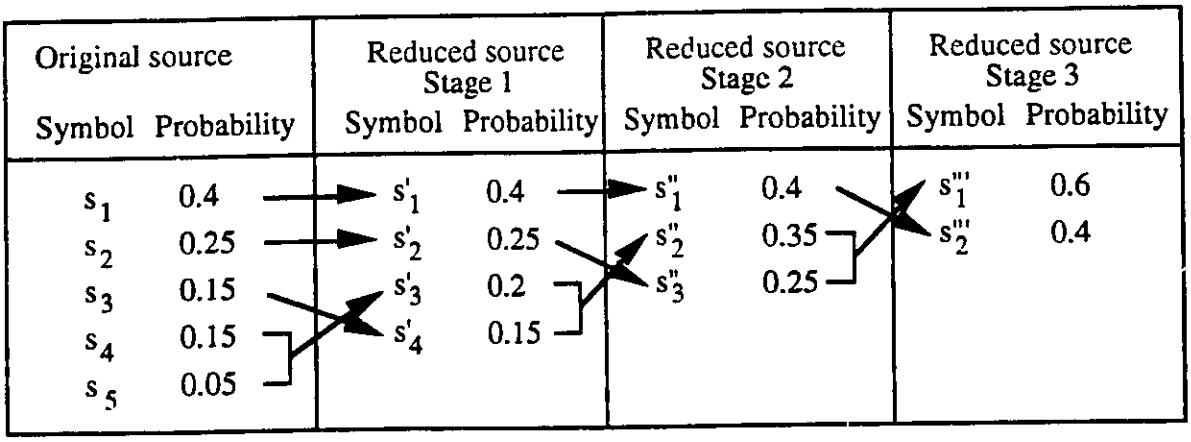


Figure 2.1(a): Source reduction process

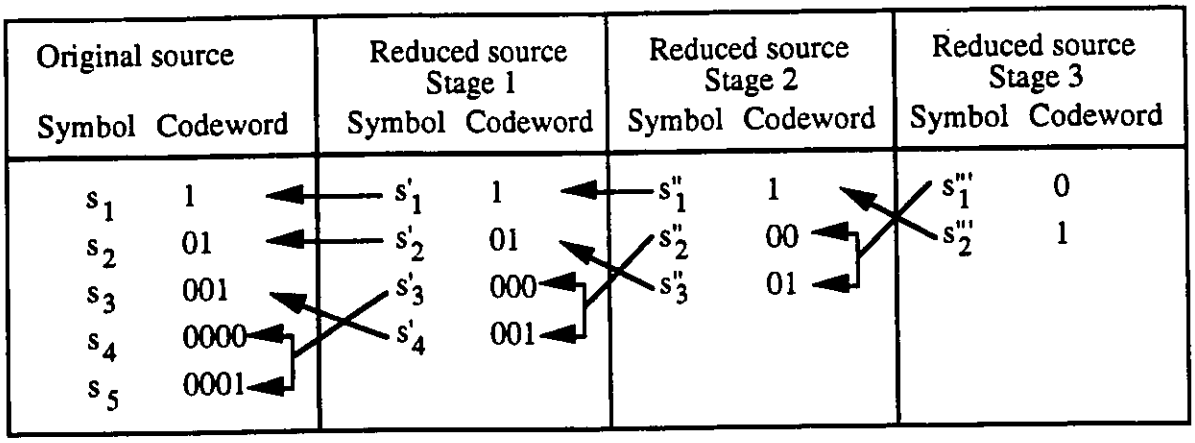


Figure 2.1(b): Huffman code construction

2.2.2 Run-length Coding

Run-length coding (RLC) is another type of lossless coding technique, which is useful for sources having identical consecutive symbols. It is extensively used in facsimile transmission. In the case of images, RLC defines a *run* as a sequence of consecutive pixels with identical values along a fixed direction, for example the horizontal/vertical scan lines. Significant bit rate reductions can be achieved for long *runs* as seen in binary images like text, graphics, etc. RLC is not very efficient when applied directly to images having high details. However, significant compression may be achieved by first splitting the images

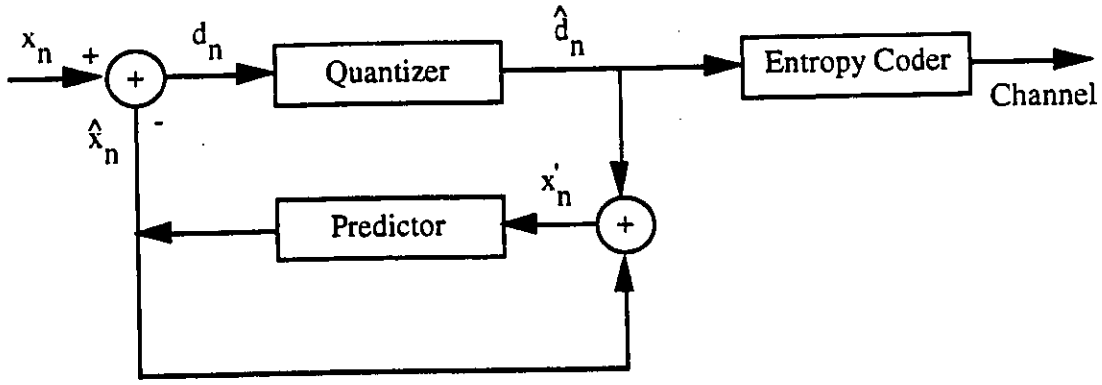
into a set of bit planes which are then individually run-length coded. RLC can also be used to provide high compression in transform coders. In order to achieve this, the transform coefficients are first quantized which results in long *runs* of zero valued coefficients. We can then use run-length coding for efficiently compressing the quantized coefficients [9]. Run-length coding can be extended to two dimensions by defining a *connected area* to be a contiguous group of pixels having identical values. To compress an image using two dimensional RLC, only the values which specify the *connected area* and its intensity are stored/transmitted.

2.3 Lossy Compression

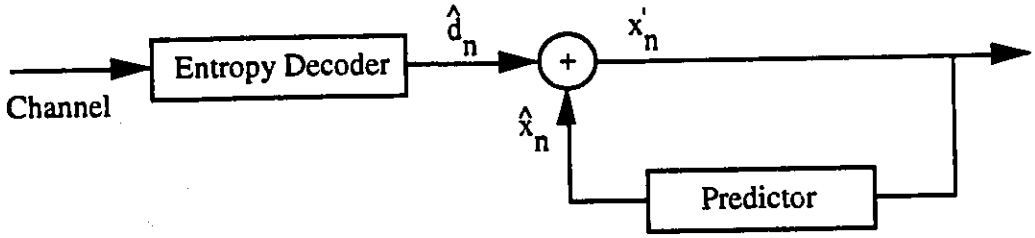
The lossless compression techniques usually result in low compression ratios. In practice, lossy compression techniques are used in applications requiring high compression (i.e. lower bit rates). In lossy compression, the objective is to reduce the bit rate subject to some constraints on the image quality. Lossy compression techniques can be classified into three categories: predictive coding, transform coding and subband coding. We will review predictive and transform coding, as both techniques have been used in this thesis. The two techniques have relative advantages and limitations. Predictive coding techniques provide real-time compression and have lower complexity and memory requirements. However, their compression ratios are relatively low and they result in a loss of subjective image quality at high compression. Transform coding achieves higher compression ratios for a given distortion value. It achieves a better subjective image quality by distributing the reconstruction error over a large image region. However, they have higher computational complexity and memory requirements. Another promising technique for low bit rate (lossy) image compression is vector quantization (VQ) which will be reviewed in detail in section 2.5.

2.3.1. Predictive Coding

Predictive coding exploits the redundancy related to the predictability and smoothness in the data. For example, an image having a constant gray level value can be fully predicted from the value of its first pixel. In images having multiple gray levels, the value of an image pixel may be used to predict the values of the neighboring pixels. *Differential pulse code modulation (DPCM)* is the basic compression scheme used in predictive coding techniques.



A DPCM Coder



A DPCM Decoder

Figure 2.2: Block diagram of a DPCM coder and decoder

DPCM

The block diagram of a DPCM system is shown in Fig.2.2. In DPCM, the data to be compressed is first predicted from the neighborhood data samples using a fixed algorithm.

The predicted estimate $\hat{x}_n$ is then subtracted from the actual data value x_n and the difference $d_n = x_n - \hat{x}_n$ is quantized, entropy coded and transmitted. The quantized difference value is in turn used to predict the next data sample x_{n+1}. In general, both linear and non-linear techniques can be used for prediction. In linear predictive coding (LPC) technique [2], the predicted value $\hat{x}_n$ is calculated as follows:

$$\hat{x}_n = \sum_i a_i x_{n-i} \tag{2.26}$$

where, a_i ; $i = 0, 1, 2, \ldots$ are called LPC coefficients. The predictor can be optimized to minimize the variance or the energy of the prediction error d_n. The optimal set of LPC coefficients $a_{i,opt}$ can be obtained using the following equation:

$$\sum_i a_{i,opt} R(j-i) = R(j) \tag{2.27}$$

where $R(j)$; $j = 0, \pm 1, \pm 2, \ldots$ is the autocorrelation function of the input data samples. The step size for quantizing the prediction difference d_n can be adjusted depending on the available data rate and the MSE requirements. A special case of DPCM is *Delta Modulation* where the quantizer output can assume only two possible levels. Delta Modulation quantizes the positive and negative differences as $+\Delta$ and $-\Delta$, respectively where Δ is a fixed ouput level.

Adaptive DPCM

DPCM provides significant compression by efficiently exploiting the correlation between the adjacent data samples. However, it is sensitive to changes in the data statistics. For non-stationary data, such as images, a fixed predictor and quantizer may not always provide a good quality in the reconstructed data. Two kinds of adaptive techniques can be used to match the DPCM coder to the variations in the input data statistics: an adaptive predictor with a fixed quantizer or an adaptive quantizer with a fixed predictor[10]. In these adaptive approaches, the parameters involved in designing the predictor and quantizer have

to be periodically updated with changing statistics of the data. For example, an adaptive predictor has to periodically recompute the autocorrelation function $R(j)$, which is in turn is used to calculate the optimal set of LPC coefficients $a_{i,opt}$ according to equation (2.27).

2.3.2. Transform Coding

In transform coding, the image data is transformed from spatial to frequency domain by the application of an orthogonal transform T,

$$Q = T X T' \tag{2.28}$$

where, $T' = T^{-1}$. The block diagram of a transform coding scheme is shown in Fig. 2.3.

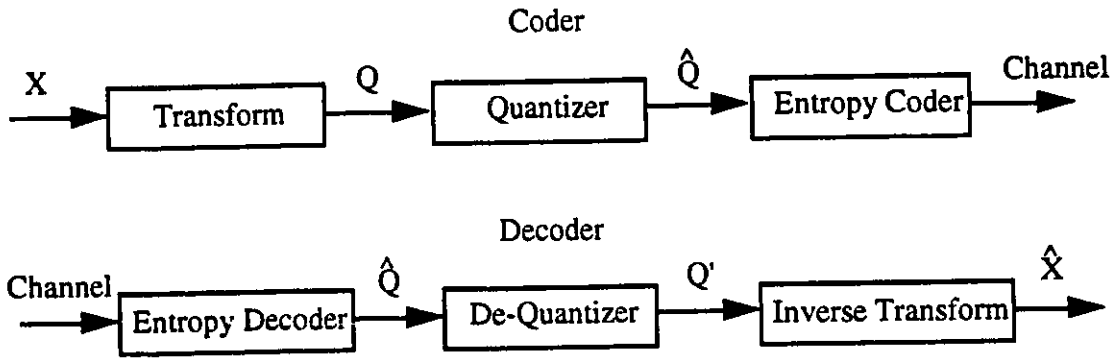


Figure 2.3: Block diagram of a transform coding scheme,
X= Original Image, $\hat{X}$ = reconstructed image

Transforms decorrelate the energy of the image pixels into coefficients Q_{ij}; $i,j = 0,\ldots,N-1$. In general, the low frequency coefficients contain most of the crucial information required for image reconstruction. In order to achieve compression, the coder quantizes the transform coefficients using small step sizes for the visually important low frequency coefficients. The quantized coefficients are then entropy coded using variable number of bits, specified by a bit allocation map. The decoder performs

decompression followed by de-quantization to recover the coefficients Q'_{ij}, which are then inverse transformed to reconstruct the coded image $\hat{X}$.

$$\hat{X} = T' Q' T \tag{2.29}$$

Transformation

A transform is *optimum* if it completely decorrelates the image data i.e. the image energy is packed into minimum number of coefficients. An optimal transform minimizes the geometric mean of the variances of coefficients which corresponds to the minimization of reconstruction error variance in transform domain [2]. A well known optimal transform is the Karhunen-Loeve transform (KLT). However, its implementation involves a high computational overhead. In addition, the KLT transform kernel is image dependent. Therefore, in practice, image independent sub-optimal transforms such as discrete sine, cosine and fourier transforms are used which have a low computational complexity compared to KLT. Among all the sub-optimal transforms, the rate distortion performance of discrete cosine transform (DCT) is closest to KLT [11]. DCT has been adopted as the compression technique in image and video coding standards, such as JPEG, MPEG and H.261. Details of DCT are presented below.

DCT decomposes the original data into a set of orthogonal basis functions. For image compression applications, the forward 2-D DCT of an $N \times N$ block of pixels is often defined as [11]

$$F(u,v) = \frac{16\, C(u)\, C(v)}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) \cos\left[\frac{(2i+1)u\pi}{2N}\right] \cos\left[\frac{(2j+1)v\pi}{2N}\right] \tag{2.30}$$

and the correponding inverse 2-D DCT is defined as

$$f(i,j) = \frac{16}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)\, C(v)\, F(u,v) \cos\left[\frac{(2i+1)u\pi}{2N}\right] \cos\left[\frac{(2j+1)v\pi}{2N}\right] \tag{2.31}$$

where, $C(x) = \begin{cases} 1/\sqrt{2} & for \quad x = 0 \\ 1 & otherwise \end{cases}$;

$f(i,j)$ = value of pixel (i, j) in the image block ;

$F(u,v)$ = DCT coefficient with frequency indices (u,v) ; $u, \ v = 0,1,...,N-1$.

The cosine terms in the equation (2.30) represent the 2-D transform kernel (or basis function) of DCT and are linearly separable. It can be shown that for a first order Markov source model, the DCT basis functions become identical to the (image dependent) KLT basis functions as the adjacent pixel correlation coefficient approaches unity [12]. Natural images generally exhibit high pixel-to-pixel correlation and therefore DCT provides a compression performance virtually indistinguishable from KLT. In addition, DCT has a fast implementation like DFT, with a computational complexity of $O(N \log N)$ for an N-point transform. Unlike DFT, DCT avoids generation of spurious spectral components resulting in a higher compression efficiency.

We note that KLT, DCT and DFT are orthogonal transforms. Orthogonal transforms have the following interesting properties:

1. The average energy of the transform coefficients is equal to the average energy of the input image pixels, that is,

$$\frac{1}{N \times N} \sum_{i,j=0}^{N-1} \sigma_{ij}^2 = \frac{1}{N \times N} \sum_{i,j=0}^{N-1} \sigma_{x,ij}^2 = \sigma_x^2 \qquad (2.32)$$

Here, $\sigma_{ij}^2 = E(Q_{ij}^2); \quad i,j = 0,...,N-1$ is the second moment of the $(i,j)th$ transform coefficient and $\sigma_{x,ij}^2 = E(X_{ij}^2); \quad i,j = 0,...,N-1$ is the second moment of the $(i,j)th$ image pixel.

2. The average reconstruction error variance is equal to the average quantization error variance in the transform domain, that is,

$$\sigma_r^2 = \frac{1}{N \times N} \sum_{i,j=0}^{N-1} \sigma_{r,ij}^2 = \frac{1}{N \times N} \sum_{i,j=0}^{N-1} \sigma_{q,ij}^2 = \sigma_q^2 \qquad (2.33)$$

Here, $\sigma_{r,ij}^2 = E((X_{ij} - \hat{X}_{ij})^2)$; $i,j = 0,...,N-1$ is the reconstruction error variance of the $(i,j)th$ image pixel and $\sigma_{q,ij}^2 = E((Q_{ij} - \hat{Q}_{ij})^2)$; $i,j = 0,...,N-1$ is the quantization error variance of the $(i,j)th$ transform coefficient.

It can be concluded from these properties that for orthogonal transforms, the spatial domain analysis will be completely reflected in the transform domain in terms of the average values of the second moment.

Quantization

The transform coefficients are quantized individually. For uniform quantization, the quantization error variance σ_q^2 and the coefficient variance are related as follows[2]:

$$\sigma_q^2 = \varepsilon_q^2 \sigma_x^2 = \varepsilon^2 2^{-2B} \sigma_x^2 \qquad (2.34)$$

Here $\varepsilon_q^2 = \varepsilon^2 2^{-2B}$ is the quantizer performance factor which depends on the probability density function (PDF) of the transform coefficients and the number of quantization levels. The parameter ε^2 is a variable correction factor that takes into account the performance of a practical quantizer and B is the number of bits assigned to the quantizer.

Bit Allocation

In order to obtain significant compression, the bit allocation procedure in a transform coding technique generally requires optimization. In other words, the transform coefficients should be assigned an unequal number of bits corresponding to their contribution to error variance in the spatial domain. Let B_{ij} be the number of bits allocated to the $(i,j)th$ transform coefficient. The avearge number of bits per coefficient, B, can be expressed as:

$$B = \frac{1}{N \times N} \sum_{i,j=0}^{N-1} B_{ij} \qquad (2.35)$$

An optimum bit allocation is given by,

$$B_{ij} = B + \frac{1}{2}\log_2 \varepsilon_{ij}^2 \sigma_{ij}^2 - \frac{1}{N \times N} \sum_{k,l=0}^{N-1} \frac{1}{2}\log_2 \varepsilon_{kl}^2 \sigma_{kl}^2 \qquad (2.36)$$

$$= B + \frac{1}{2}\log_2 \frac{\varepsilon_{ij}^2 \sigma_{ij}^2}{[\prod_{k,l=0}^{N-1} \varepsilon_{kl}^2 \sigma_{kl}^2]^{1/(N \times N)}}; \quad i,j = 0,1,\ldots,N-1 \qquad (2.37)$$

It should be observed that both the coefficient variance, σ_{ij}^2, and the quantizer performance factor, ε_{ij}^2, are proportional to the number of quantizer levels, $2^{B_{ij}}$. In the case of images, transform based coding (e.g. baseline mode of JPEG) provides a good image quality at bit rates ranging from 0.7 to 1.5 bits per pixel. However, at medium to low bit rates, it produces blocking artifacts resulting in poor subjective quality.

Other transform coding techniques

Recently, subband coding [13-15] has emerged as a powerful approach for image compression. In subband coding, an image is first filtered to create a set of sub-images or subbands, each of which contains a limited range of spatial frequencies. The different subbands can be downsampled due to their small bandwidths as compared to the original image. Fig.2.4 (a) shows the decomposition of an image into 16 subbands using a balanced binary tree structure. Each subband is coded individually. Compression is

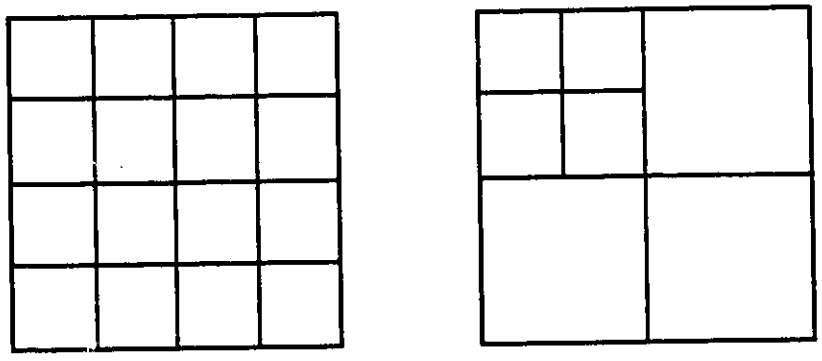


Figure 2.4: (a) A two level subband decomposition of an image using a balanced binary tree (b) A two level wavelet decomposition of an image.

achieved by allocating different number of bits to the subbands based on their visual importance. The image is reconstructed by upsampling the decoded subbands and combining them using an appropriate set of filters. Subband coding is generally implemented using quadrature mirror filters (QMFs) for alias-free reconstruction. A special case of the subband coding technique is wavelet transform. Wavelet theory [30-33] provides multi-resolution / multi-frequency representation of nonstationary signals, such as images. It provides signal localization in both space and frequency domains, resulting in sub-images which are relatively more stationary and hence easier to code. Fig. 2.4 (b) shows a two level wavelet decomposition of an image resulting in 7 subbands. Details of wavelet based image coding will be presented in section 2.6.

2.4 JPEG Standard for Still Image Compression

Recently, the Joint Photographic Experts Group, of International Standards Organization (ISO), has proposed the JPEG standard for still image compression. The JPEG standard [9, 16, 17] provides a framework for high quality compression and reconstruction of continuous-tone grayscale and color images for a wide range of applications. JPEG specifies:

- A toolkit of algorithms for converting the raw image data into a compressed form.
- Decompression algorithms for reconstructing the image from the compressed data.
- Guidelines for practical implementations of these algorithms.
- An interchange format for exchanging the compressed image data between different application environments.

However, the standard does not specify many parametric details such as color space designation (e.g. RGB or YUV), aspect ratio (width to height ratio of an image pixel), etc. which are generally application dependent. The compression algorithms are executed in the

JPEG coder while the decompression algorithms are executed by the individual JPEG decoders.

The JPEG standard has four modes of operation, each providing a unique functionality for a specific class of applications:

(a) Baseline Sequential

(b) Lossless Compression

(c) Progressive coding

(d) Hierarchical coding.

The baseline sequential mode provides the basic feature of (DCT based) lossy image compression. The lossless compression mode uses a DPCM based algorithm followed by entropy coding to achieve an exact image reconstruction and is targeted towards applications which require very high image quality, such as medical imaging. The progressive mode provides the capability of reconstructing an image in progressively improving picture qualities, using a modification of the baseline mode algorithm. On the other hand, the hierarchical mode reconstructs the image in different sizes, using either the baseline compression algorithm or lossless compression or both. We now present the details of the baseline sequential and lossless compression modes. The progressive and hierarchical modes will be discussed in chapter 3.

2.4.1 Baseline Sequential Mode

The compression algorithm used in the baseline sequential mode first partitions the original image into non-overlapping blocks of 8x8 pixels as shown in Fig.2.5. In order to decrease the average energy of the image pixels, a value of 2^{p-1} is subtracted from each image pixel $f(i,j)$, where p is the number of bits required to represent each pixel value (i.e. a value of 128 is subtracted for an 8 bits/pixel image). This is generally referred to as a *level shift*

operation. Each block of 8x8 pixels is then transformed using DCT into an array of 8x8 coefficients (equation 2.38).

$$F(u,v) = \frac{C(u)\,C(v)}{4}\sum_{i=0}^{7}\sum_{j=0}^{7} f(i,j)\,\cos\left[\frac{(2i+1)u\pi}{16}\right]\cos\left[\frac{(2j+1)v\pi}{16}\right] \qquad (2.38)$$

where, $C(x) = \begin{cases} 1/\sqrt{2} & for \quad x = 0 \\ 1 & otherwise \end{cases}$;

$f(i,j)$ = level shifted value of pixel (i, j) in the image block ;

$F(u,v)$ = DCT coefficient with frequency indices (u,v) ; $u,\ v = 0,1,....,7$.

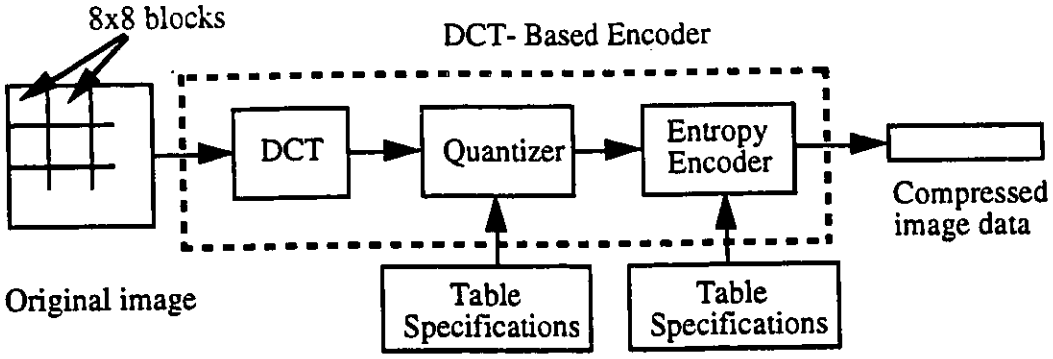


Figure 2.5: Baseline JPEG Encoder

The first coefficient (0,0) of every block is called the DC coefficient, while the rest of the coefficients are called AC coefficients. The DC coefficient corresponds to the mean of the pixels in an image block. DCT concentrates most of the perceptually important information into the DC and a few (low frequency) AC coefficients as shown in Fig.2.6. In other words, the higher frequency coefficients contain relatively less crucial details. All the coefficients are quantized using a uniform midstep quantizer (equation 2.39) and rounded to the nearest integer as shown in Fig.2.7.

$$C(u,v) = \left\lfloor \frac{F(u,v) + (Q(u,v))/2}{Q(u,v)} \right\rfloor \qquad (2.39)$$

where, $Q(u,v)$ = quantization step size for coefficient (u,v) ;

$C(u,v)$ = rounded value of the quantized coefficient ;

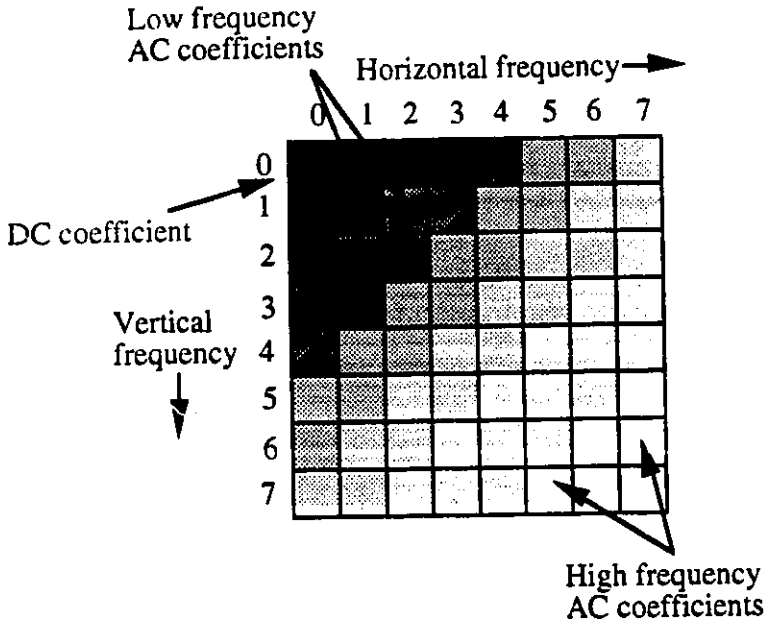


Figure 2.6: A DCT transformed 8x8 pixel block.

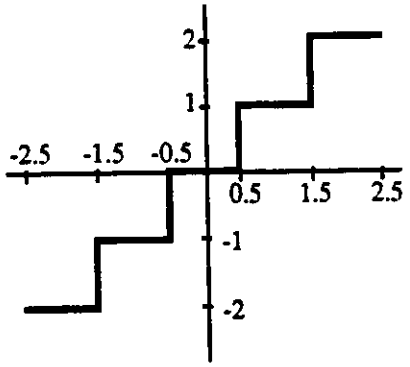


Figure 2.7: A uniform midstep quantizer

Quantization reduces the coefficient magnitude and increases the number of coefficients with zero value. We note that quantization results in a loss of image information which may appear as distortions in the reconstructed image. The quantization step sizes are not fixed in the JPEG standard. In other words, each application can have its own quantization tables which is usually designed to provide the best possible reconstructed image quality. A sample quantization table has been suggested in the standard, based on the statistics from a number of test images. In order to obtain a good subjective image quality, the DC and the

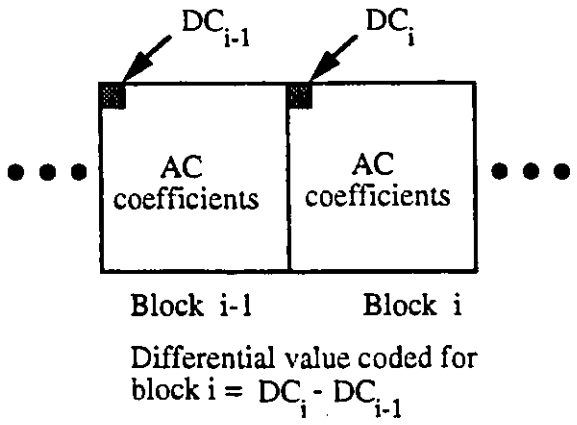


Figure 2.8: Differential DC coding

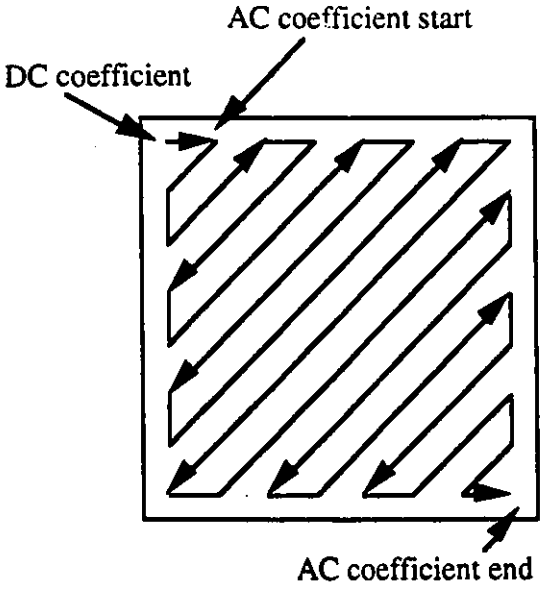


Figure 2.9: Zig-zag scanning of DCT coefficients

low frequency AC coefficients are typically quantized using small step sizes while the high frequency AC coefficients are quantized using larger step sizes. Hence, there is a tradeoff between the quantization step size (i.e.image quality) and the compression achieved: smaller the step size, better the image quality and smaller the compression ratio. The correlation between the DC coefficients of adjacent blocks are exploited using DPCM to achieve further compresion (Fig.2.8).

The DCT coefficients of each block are reordered in a 1-D sequence using a zig-zag scan as shown in Fig.2.9. This scheme generates long runs of zero value coefficients (corresponding to the higher frequency AC cofficients) in most of the images. The zig-zag ordered coefficients are compressed using either huffman or arithmetic coding. In this thesis, huffman coding has been used which is implemented by storing/transmitting a code

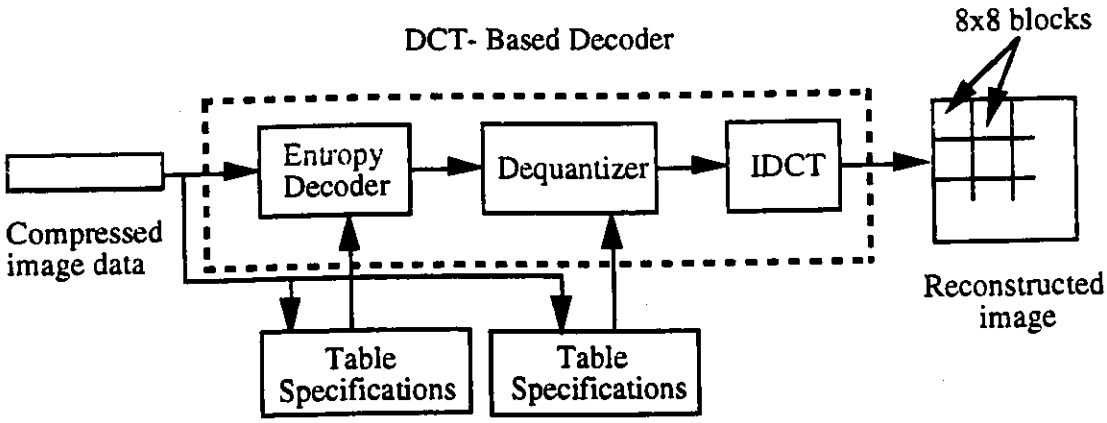


Figure 2.10: Baseline JPEG Decoder

for each DC and non-zero AC coefficient indicating their magnitude and position in the zig-zag order. Finally, the image blocks are raster scanned to generate the image bitstream. Once again, the huffman codes are not fixed in the JPEG standard. Each application can design its own huffman tables to provide the maximum possible compression. The standard only specifes the syntax for describing huffman tables in the interchange format. The baseline compression algorithm (BSA) is allowed to use a maximum of two DC and two AC Huffman tables. We note that an extended sequential algorithm (ESA) has been defined in the standard for special applications. It is similar to the BSA except that a maximum of eight tables are allowed: four each for DC and AC coefficients. Four sample tables (two each for DC and AC coefficients) have been provided in the standard.

The image is reconstructed by performing the decompression operations in the reverse order (Fig. 2.10). Each block of 8x8 pixels is hence transformed back to spatial domain using the inverse discrete cosine transform (IDCT) (equation 2.40).

$$f'(i,j) = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C(u)\, C(v)\, F'(u,v) \cos\left[\frac{(2i+1)u\pi}{16}\right] \cos\left[\frac{(2j+1)v\pi}{16}\right] \quad (2.40)$$

where, $F'(u,v)$ = dequantized DCT coefficient with indices (u,v); $u,v = 0, 1, \ldots 7$.

$f'(i,j) = (i,j)$th pixel in the reconstructed image block; $i,j = 0, 1, \ldots, 7$.

Finally, an inverse *level shift* operation is performed by adding 2^{p-1} to each pixel value. We note that the baseline sequential algorithm can only be used to reconstruct the image in its original size at a specific image quality.

2.4.2 Lossless Compression

We recall from section 2.4 that the lossless compression mode of JPEG uses a DPCM based algorithm for compressing the image. Each pixel x is predicted from its neighbourhood pixels a,b and c as shown in Fig.2.11. The prediction difference is coded using either huffman or arithmetic coding. The standard allows seven schemes for prediction:

Scheme No.	Prediction
0	No prediction
1	Px = Ra
2	Px = Rb
3	Px = Rc
4	Px = Ra + Rb - Rc
5	Px = Ra + ((Rb-Rc)/2)
6	Px = Rb + ((Ra-Rc)/2)
7	Px = (Ra + Rb)/2

Here, Px is the predicted value of pixel x and Ra, Rb and Rc are the reconstructed values of

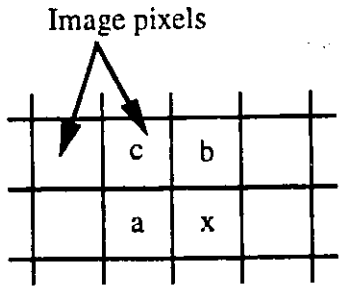


Figure 2.11: Prediction of pixel x from its neighborhood pixels a, b and c.

pixels a,b and c, respectively. Scheme 0 is used only in the hierarchical mode of operation as will be discussed in section 4.2. Schemes 1, 2 and 3 use one-dimensional prediction, while schemes 4, 5, 6 and 7 use two-dimensional prediction. The pixels in the first row and first column of the image are coded using schemes 1 (horizontal prediction) and 2 (vertical prediction), respectively. A prediction value of 2^{p-1} is used for coding the first image pixel, where p is the number of bits required to represent each pixel value.

2.5 Vector Quantization for Image Compression

We recall from section 2.3.2 that the transform based coding produces blocking artifacts resulting in poor subjective quality, at low bit rates. Vector quantization (VQ) is a powerful technique for low bit rate image compression [18, 19]. According to Shannon's rate distortion theory, a better performance can always be achieved by coding vectors instead of scalars. VQ can be defined as a mapping Q of K-dimensional Euclidean space R^k into a finite subset Y, of K-dimensional vectors, called the codebook.

$$Q \; : \; R^k \rightarrow Y \tag{2.41}$$

More specifically, the vector quantizer partitions the space R^k into N non-overlapping regions P_i; $i = 1,\ldots,N$, where N = number of vectors in the codebook Y. Each region P_i is assigned a unique reproduction vector, called the codeword, drawn from the set Y. All

the vectors belonging to the region P_i are now represented by $Q(P_i) = Y_i$. The block diagram of a vector quantizer is shown in Fig.2.12.

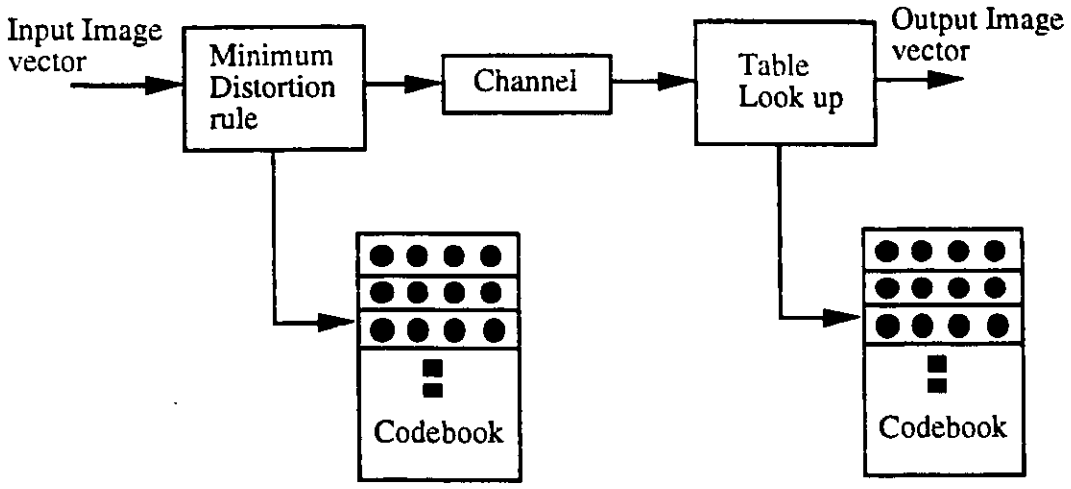


Figure 2.12: Block Diagram of a Vector Quantizer

If $d(X,Y_i)$ is a distortion measure which represents the error when X is reproduced by Y_i, then the optimal quantizer is the one which minimizes $d(X,Y_i)$ over all the quantizers. The quantizer performance is measured in terms of the average distortion introduced:

$$\begin{aligned} D &= E\{d(X,Y_i)\} \\ &= \sum_{i=1}^{N} p(X \in P_i) E\{d(X,Y_i) | X \in P_i\} \end{aligned} \tag{2.42}$$

It follows from equation (2.42) that the optimal quantizer must satisfy the following two conditions:

1. A vector X should be assigned a reproduction vector Y_i based on the nearest neighbour rule, that is,

$$Q(X) = Y_i \quad iff \quad d(X,Y_i) < d(X,Y_j) \quad for \quad all \quad j \tag{2.43}$$

2. The codewords $\{Y_i; i = 1,2,...,N\}$ are selected such that the average distortion in each region is minimized (i.e. the codewords are the centroids of the region they represent).

For image compression, vector quantization consists of the following steps: vector formation, codebook design and quantization.

Vector Formation

The first step in VQ is vector formation. The input image is partitioned into two-dimensional blocks of equal or variable sizes. The key features of each block are extracted which are then arranged into one-dimensional vectors. Various vector formation schemes have been proposed in the literature [21-25]. The simplest scheme is to form the vectors from original pixel values of the image blocks [21]. More sophisticated schemes will be presented in section 2.6.

LBG Algorithm

An algorithm for codebook design has been proposed by Linde *et. al.*, based on the two conditions for optimality, referred to as the generalized Lloyd or the LBG algorithm [20]. The algorithm starts with an intial codebook and assigns each training vector to its nearest neighbour codeword. Each codeword is then modified to minimize its distortion relative to the vectors assigned to it. The last two steps are repeated iteratively until the change in distortion between two successive iterations is sufficiently small. The algorithm is described as follows:

1. Given an initial codebook, $CB_0 = \{Y_i; i = 1,2,....,N\}$, a threshold $\varepsilon \geq 0$, and a training set $\{X_i; i = 1,2,...,M\}$, set m to 0 and D_{-1} to ∞.

2. Assign each input vector to its nearest neighbour codeword,

$$X_i \in Y_{j,m} \quad iff \quad d(X_i, Y_{j,m}) \leq d(X_i, Y_{k,m}) \quad for \quad all \quad j \neq k \tag{2.44}$$

3. Find CB_{m+1}, by computing the centroids of the training vectors assigned to each codeword,

$$Y_{i,m+1} = \frac{1}{K_{i,m}} \sum_{X_j \in Y_{i,m}} X_j \quad ; \quad i = 1,2,\ldots,N \tag{2.45}$$

where $K_{i,m}$ is the number of vectors assigned to $Y_{i,m}$, and

$$\sum_i K_{i,m} = M \quad for \quad all \quad m \tag{2.46}$$

4. Compute the average distortion,

$$D_m = \frac{1}{M} \sum_{i=1}^{N} \sum_{X_j \in P_i, j=1}^{K_{i,m}} D(X_j, Y_{i,m}) \tag{2.47}$$

if the normalized difference between D_m and D_{m-1}, then stop; else go to step 2.

One possible approach to initialize the codebook CB_0 is to select N evenly spaced vectors. Alternatively, a binary splitting algorithm can also be used [20], where the centroid of each partition of the training set is split into two codewords. The LBG algorithm is first applied to yield a codebook of two codewords. Each codeword is then split into two codevectors to yield a codebook of four codewords. This procedure is repeated until N codewords are generated.

Quantization

The image vectors are quantized by searching the codebook to locate the best-match codeword for each input vector. Compression is achieved by transmitting the index (label) of the codeword for each vector. The image is reconstructed at the decoder by a simple table look-up operation, using the labels as an address to a table containing the codewords. The codebooks can be broadly classified into two classes: universal and image adaptive, resulting in universal VQ (UVQ) and image adaptive VQ (IAVQ), respectively. In UVQ, both the transmitter and the receiver have a copy of the same codebook, and therefore no codebook transmission is required [26]. This results in a substantial bit rate reduction. The codebook is generated from a large set of training vectors selected from different types of images. For a codebook of size N, the bit rate is given by,

$$R = \frac{(\log_2 N)}{L} \quad bits / pixel \tag{2.48}$$

where L is the vector dimension. In IAVQ, the codebook is generated from one or more images to be compressed and is transmitted to the receiver as a side information. IAVQ codebooks match the image statistics better resulting in a superior image quality. However, this technique has an overhead in terms of the bit rate for codebook transmission and computational complexity for codebook generation. In this thesis, UVQ has been used as it has lower computational complexity compared to IAVQ.

UVQ requires a codebook of large size to ensure a good quality reconstructed image. This increases the bit rate for label transmission as well as the coding complexity. UVQ may not provide a good coding performance for certain types of input images. Recently, some techniques have been proposed to decrease the codebook size. Examples include classified VQ, predictive VQ and finite state VQ. In classified VQ (CVQ) [27, 28], the training vectors are classified into a finite set of classes based on perceptually important features, such as the edges. A separate codebook is generated for each class. During quantization process, a classifier determines the class to which each input vector belongs. The input vector is then encoded using the appropriate codebook. Predictive VQ (PVQ), techniques employ a predictor followed by vector quantizer. The input vector is predicted from the previously quantized vectors, and the prediction error is quantized. Smaller codebooks are usually required for error vectors. Another approach used to reduce the bit rate and the search complexity is finite state VQ (FSVQ) [29]. An FSVQ encoder consists of a finite set of states, each associated with a small subcodebook. The state of the encoder is determined from the previously encoded vectors and the input vector is quantized using the corresponding codebook.

2.6 Image Coding using Discrete Wavelet transform

We recall from section 2.3.2 that wavelet transform [30-33] is a special case of the more general subband coding technique. Wavelet theory provides a multi-resolution / multi-frequency representation of nonstationary signals, such as images. Wavelet transform decomposes an image into a set of sub-images with excellent signal localization in both the spatial and frequency domains. The sub-images are more stationary compared to the original image and hence easier to code. Wavelets provide a superior subjective image quality by taking into account the characteristics of human visual system (HVS).

The basic concept of one-dimensional wavelet transform is to represent a signal $f(t)$ as superposition of a set of basis functions called wavelets [30, 33]. In order to achieve this, the prototype or mother wavelet function $\psi(t)$ is scaled and translated along the t axis to create a family of basis functions:

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k); \quad j,k \in Z \tag{2.49}$$

Here, j is an integer measure of scaling (compression or dilation) of the wavelet function and is analogous to the concept of frequency, and k denotes the location of the function on the t axis. The function $f(t)$ can now be expressed as:

$$f(t) = \sum_{j,k=-\infty}^{\infty} a_{j,k}\psi_{j,k}(t) \tag{2.50}$$

where $\{a_{j,k}\}; \quad j,k \in Z$, are the discrete wavelet transform (DWT) coefficients of $f(t)$.

Multiresolution Representation

The power of the wavelet theory lies in its ability to represent a signal at different resolutions with excellent time-scale localization [30]. This property, when applied to images, provides lower resolutions of high subjective quality. In multiresolution analysis, the function $f(t)$ is represented at lower resolutions using the concept of scaling function.

For this purpose, a family of scaling functions is generated from the basic scaling function $\phi(t)$:

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k); \quad j,k \in Z \tag{2.51}$$

where $\phi_{j,k}(t)$ is obtained by both scaling and translating the variable t in $\phi(t)$. The function $f(t)$ can now be expressed as:

$$f(t) = \sum_{k=-\infty}^{\infty} c(0,k)\phi_{0,k}(t) + \sum_{j=0}^{\infty}\sum_{k=-\infty}^{\infty} d(j,k)\psi_{j,k}(t) \tag{2.52}$$

The first term of equation (2.52) provides a reduced resolution representation of the function $f(t)$. The second term contains the difference information between the original function and its reduced resolution. Here, $c(0,k)$ and $d(j,k)$ are the scaling and DWT coefficients of the function, respectively. To obtain $f(t)$ at still lower resolution, the first term can be further expanded as:

$$\sum_{k=-\infty}^{\infty} c(0,k)\phi_{0,k}(t) = \sum_{k=-\infty}^{\infty} c(-1,k)\phi_{-1,k}(t) + \sum_{k=-\infty}^{\infty} d(-1,k)\psi_{-1,k}(t) \tag{2.53}$$

where the first term represents the function at half the previous resolution. In practice, we deal only with the coefficients and not with ϕ and ψ functions. If $\phi_{j,k}(t)$ and $\psi_{j,k}(t)$ are orthonormal, then the jth level c and d coefficients can be derived from $(j+1)$th level c coefficients as follows:

$$c(j,k) = \sum_m h(m-2k)c(j+1,m) \tag{2.54}$$

$$d(j,k) = \sum_m g(m-2k)c(j+1,m) \tag{2.55}$$

where $h(m)$ and $g(m)$ are low pass and high pass FIR filters, respectively. In other words, level j coefficients are obtained by FIR filtering level $j+1$ scaling coefficients followed by downsampling. The decomposition of a signal into lower resolutions is generally referred to as the analysis process and consequently, $h(m)$ and $g(m)$ are together referred to as analysis filters. The level j scaling coefficients correspond to a lower resolution and can be further decomposed (Fig. 2.13) until the signal has been represented at the lowest resolution desired. If the function $f(t)$ is sampled above Nyquist rate, then its sampled values serve as a good approximation for the scaling coefficients at the highest

resolution. On the other hand, the scaling coefficients at lower levels represent the samples of the signal at the corresponding resolutions.

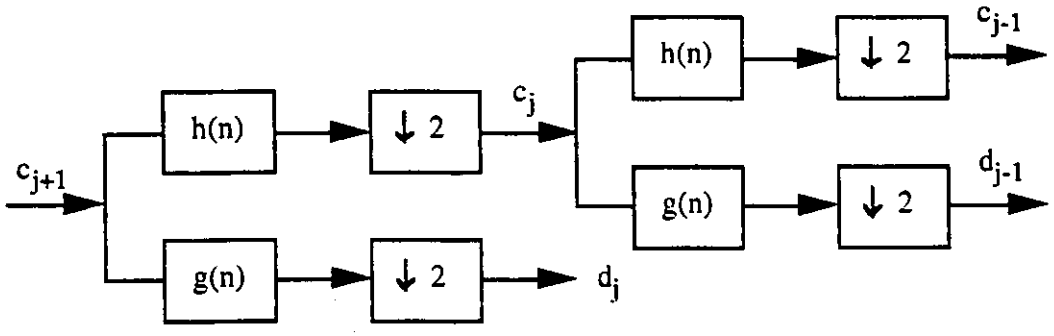


Figure 2.13: Decomposition of a signal into lower resolution using analysis filters

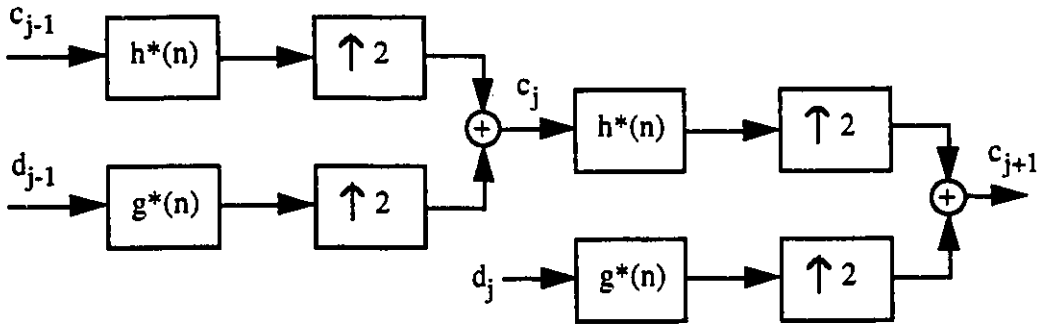


Figure 2.14: Reconstruction of a signal from its lower resolution using synthesis filters

It is possible to reconstruct the signal at a higher resolution using the lower resolution coefficients. The scaling coefficients at $(j+1)$th level can be obtained from the jth level coefficients as follows:

$$c(j+1,k) = \sum_m h^*(k-2m)c(j,m) + \sum_m g^*(k-2m)d(j,m) \qquad (2.56)$$

where $h^*(m)$ and $g^*(m)$ are the low pass and high pass FIR filters, respectively. To evaluate equation (2.56), the level j coefficients $c(j,k)$ and $d(j,k)$ are first upsampled by inserting zeros in between the samples. The upsampled coefficients are then passed through their respective FIR reconstruction filters, $h^*(m)$ and $g^*(m)$, and added to obtain the

$(j+1)$th level coefficients. The reconstruction of a signal at a higher resolution from its lower resolution coefficients is often referred to as synthesis process and consequently, $h^*(m)$ and $g^*(m)$ are together referred to as synthesis filters. The reconstruction process can be recursively continued (Fig. 2.14) until the signal has been obtained in its original resolution.

In image processing applications, a two-dimensional separable wavelet transform is implemented independently, first in the horizontal direction and then in the vertical direction [34]. This results in orientation selective high pass sub-images and a low pass sub-image, each of which is half the original size in both the directions as shown in Fig.2.15 (a). The LL sub-image contains low frequency information in both directions, and is a reduced size version of the original image. The other three sub-images (LH, HL, HH) contain high frequency information along the horizontal, vertical and diagonal directions,

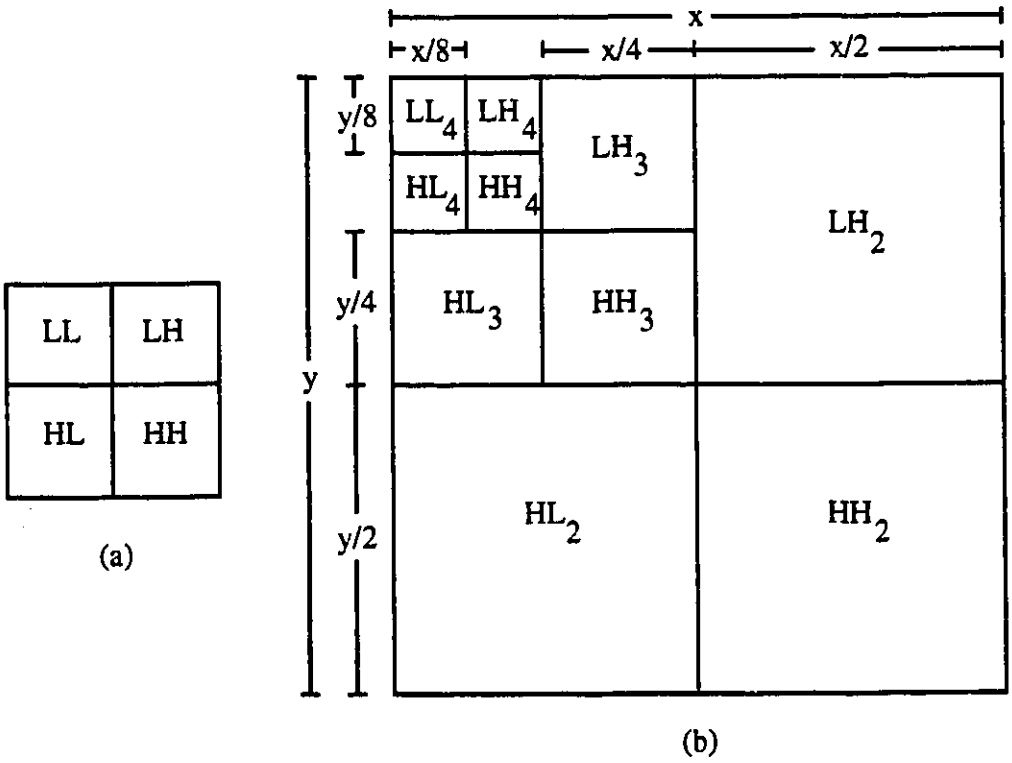


Figure 2.15: (a) Wavelet transform decomposition of an image into 4 sub-images (b) A three level image decomposition .

respectively. The LL sub-image can be further decomposed by successive application of wavelet transform until the smallest size image is obtained. For example, a 3-level image decomposition is shown in Fig.2.15(b), where the LL_4 sub-image generally corresponds to an iconic image. The LL_4, LH_4, HL_4 and HH_4 sub-images can be recombined to reconstruct the image at the next higher resolution LL_3. This process can be repeated to reconstruct the image at LL_2 and LL_1 (full size) resolutions.

The multiresolution sub-images obtained using the wavelet transform are more stationary compared to the original image signal and are hence easier to compress. Recently, some encouraging results have been reported in the literature for wavelet based image compression. Antonini *et. al.* [34] have proposed a multiresolution vector quantization scheme, to compress the different sub-images individually. In their technique, a universal sub-codebook is generated for each resolution level and orientation (horizontal, vertical and diagonal) using a number of test images. The training set comprises of vectors belonging to different images corresponding to the resolution and orientation under consideration. Each sub-codebook is generated by the LBG algorithm using binary splitting technique. A multiresolution codebook can be obtained by assembling the resulting sub-codebooks

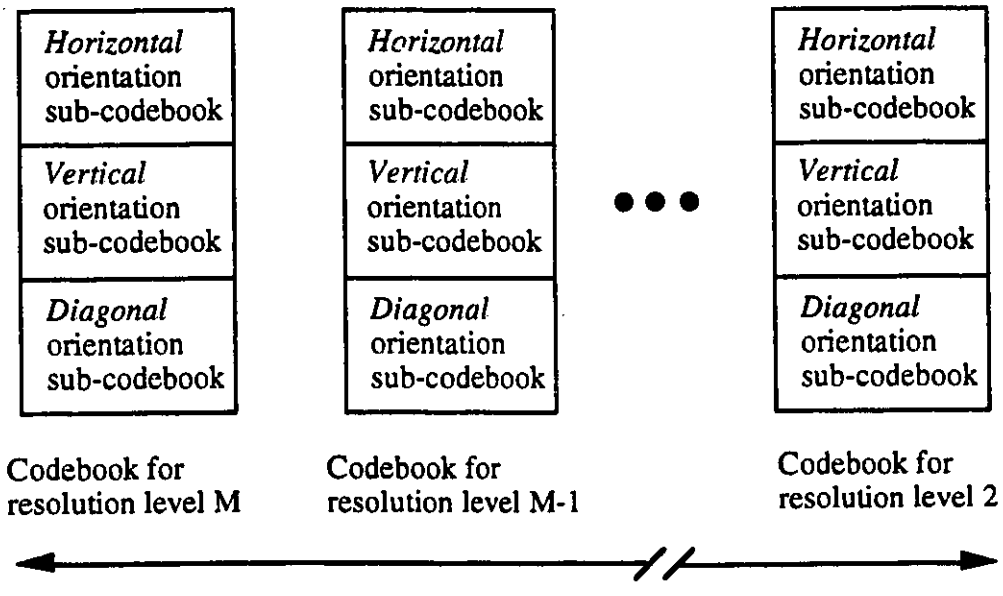


Figure 2.16: A multiresolution codebook

corresponding to different orientations at all resolution levels, as shown in Fig. 2.16. Each sub-codebook provides a low reconstruction error and contains few codewords, which speeds up the searching process for the best-match codeword. The coding complexity is also low as only the appropriate sub-codebook is searched for each input vector. In addition, it also results in a good subjective quality of the reconstructed image. One disadvantage of their technique is that a high bit rate is required to reconstruct the high pass sub-images, which may contain edges and other high frequency information.

Lewis and Knowles [35] have proposed a non-uniform tree structured hierarchical coding scheme for individually quantizing the coefficients in the wavelet sub-images. Their scheme uses a HVS based quantization model to reconstruct images of high subjective quality. They have also proposed a 3-D wavelet transform based technique for video compression.

Banham and Sullivan [36] have introduced a quadtree based technique for coding wavelet transformed images. This technique exploits the correlation between the different orientation sub-images at all resolution levels resulting in efficient compression. The coefficients in each sub-image are predicted using the corresponding coefficients in the higher scale sub-image with the same orientation. The prediction difference is scalar quantized using non-uniform Laplacian quantizers.

Wang *et. al.* [37] have used non-linear interpolative vector quantization (NIVQ) to achieve futher improvements in wavelet based image coding. Their technique forms 64-D image vectors by combining the coefficients in corresponding locations in all the sub-images. Since, the lower scale sub-images represent the image at lower resolutions, vector quantization is performed using only the first 16 elements of each vector corresponding to the lower scale sub-images. This technique not only provides improved coding performance by exploiting the correlation between the wavelet sub-images, but also reduces the computational complexity using NIVQ. Details of this technique will be presented in section 5.2.

2.7 Summary

In this chapter, we first reviewed the fundamental concepts of source coding theory, namely entropy and rate distortion, as applied to image compression. This was followed by a review of lossless and lossy compression techniques, such as huffman coding, run-length coding and DCT. We then presented the details of the JPEG standard which provides a framework for still image compression. Section 2.5 reviewed vector quantization, a powerful technique for low bit rate image compression. Finally, we reviewed wavelet based image coding techniques in section 2.6. In the next chapter, we define the concept of scalability and review the various scalable image compression techniques.

Chapter 3

Review of Scalable Image Compression

In this chapter, we first define the concept of scalability and then describe the three types, namely SNR, spatial and temporal scalabilities. We note that in the context of still images, only the first two types of scalabilities are applicable. We then describe a more general form of scalability called the bit rate scalability. This is followed by a review of the various spatial and tranform domain techniques for achieving scalable image compression.

3.1 Definition of Scalability

In general, each image and video compression application has specific functional requirements and quality constraints. For example, an image browsing application generally requires the capability of displaying images in different sizes and/or picture qualities. Similarly, simultaneous transmission of NTSC and HDTV also requires the capability of displaying video signals in multiple spatial resolutions. Though the two applications require similar features, their quality requirements can be quite different. The transmission application requires high subjective quality at all spatial resolutions, while the browsing application may require a high quality image only at the full spatial resolution. In addition,

some of the features of an application may be optional and may depend on the availability of resources such as, CPU processing power, transmission bandwidth and display resolution. These resources generally vary from one system to another and in some cases may impose constraints even on the basic features required by the applications. Therefore, the encoding and decoding algorithms should be flexible enough to provide a satisfactory performance, by efficiently using the available resources. This extra flexibility is referred to as scalability. In other words, scalability is a property of the image/video data representation that allows applications to provide certain features [38-42]. The three types of scalabilities are defined below.

3.1.1 SNR Scalability

SNR (quality) scalability is a generic feature referring to the representation of images in different qualities or Signal to Noise (SNR) resolutions, at a fixed size. SNR scalability has sometimes been termed as amplitude scaling [38], and more often as progressive image transmission (PIT) [43-45]. SNR scalability makes possible an approximate reconstruction of the image using only a small portion of the compressed bitstream for quick recognition. Based on user's demand, the reconstructed image is then progressively improved, by adding more information from the bitstream in single/multiple iterations, until the image is of acceptable quality. This feature is especially useful in interactive applications, such as image browsing, where an image database is searched based on a particular criterion. The retrieval process can be terminated if the intermediate version of the image is of satisfactory quality or if the image is found to be of no interest. This results in the saving of average decompression time per image and transmission bandwidth (in the case of remote databases) thereby providing *effective* compression due to user termination. In addition, SNR scalability is also useful in transmitting images over low bandwidth channels, such as telephone lines, where the available transmission bandwidth may be limited. These

applications generally require the image to be reconstructed in an acceptable (coarse) resolution using only a small portion of the entire bitstream. However, a typical compression algorithm may provide a (high quality) reconstruction of only a small image region using an approximately equal number of bits.

In order to provide SNR scalability, the compression algorithm should have the following characteristics:

1. The image data should be reorganized in multiple layers in the decreasing order of importance. The first layer corresponds to the reconstructed image at the lowest possible SNR resolution, which is improved by gradually adding the subsequent layers.
2. The first few approximations of the reconstructed image should have a very low bit rate and should provide sufficient information for image recognition.
3. Good coding performance.
4. If required, it should be possible to exactly reconstruct the original image (lossless compression).
5. A fast decompression algorithm, since browsing images in a database implies one time compression and mutiple decompressions.

3.1.2 Spatial Scalability

Spatial scalability refers to the feature of image representation in different sizes or spatial resolutions [39-41]. This feature provides the capability of reconstructing images in lower spatial resolutions using only a small portion of the compressed bitstream. Like SNR scalability, spatial scalability is also very useful in image browsing applications. The images in the database are first displayed at the lowest resolution, corresponding to the size of an icon. Based on user's demand, the selected image can then be zoomed to higher spatial resolutions by adding more information from subsequent portions of the image bitstream,

until the image is reconstructed in full spatial resolution. Alternatively, spatial scalability can be used to simultaneously display an image at different spatial resolutions. This feature is very useful in a multiuser environment where the compressed bitstream is available to a number of display devices with varying resolution requirements, e.g. a videophone, an NTSC video monitor and an HDTV.

The basic characteristics required in a compression scheme for efficiently implementing spatial scalability are:

1. Hierachical reorganization of the image data where the highest level of the hierarchy corresponds to lowest spatial resolution of the image.
2. Low spatial resolutions of high quality.

The remaining requirements are similar to those for SNR scalability, such as good coding performance, lossless reconstruction (if possible) and a fast decompression algorithm.

3.1.3 Temporal Scalability

Temporal scalability refers to a hierarchical representation of video such that each level of the hierarchy has a different frame rate [40, 41]. Temporal scalability is applicable only to video and has applications related to migration of current NTSC and HDTV formats having low temporal resolutions (frame rate) to future progressive HDTV having higher temporal resolution. This is achieved by using a layered coding scheme to accomodate both interlaced and progressive video.

We note that the three types of scalabilities can be combined in different ways (resulting in a layered coding structure), depending on the application requirements. For example, the SNR and spatial scalabilities can be combined together to provide a flexible image browsing system, where each spatial resolution of the image can be displayed at multiple SNR resolutions. Fig.3.1 illustrates a particular combination of the SNR, spatial and temporal scalabilities for a sequence of image frames. Here, the four layers may corresponds to a

videophone with a low frame rate, an NTSC resolution TV, an HDTV and a super high definition HDTV, respectively. In this thesis, only SNR and spatial scalabilities (as applicable to still images) are investigated.

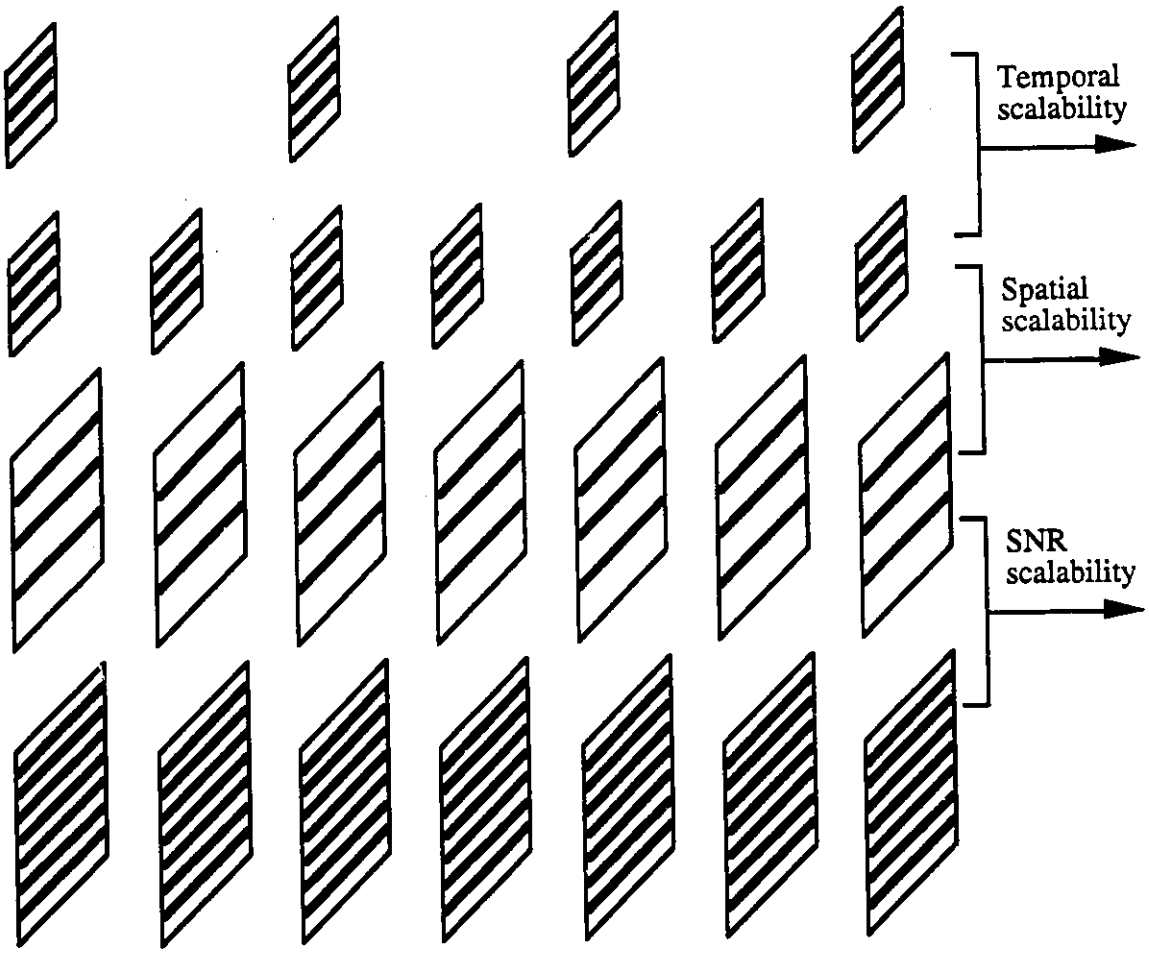


Figure 3.1: Illustrations of the three types of scalabilities.

Bit rate scalability is a more comprehensive concept and refers to the scaling of the image bitstream such that the image sequence can be reconstructed at a lower quality and/or size and/or frame rate using a fixed portion of the compressed bitstream. Bit rate scalability is useful in ensuring a graceful degradation in the system performance in the event of resource constraints. Two important resources whose loss may degrade the system performance are: transmission bandwidth and CPU processing power [42]. For example, loss of

transmission bandwidth may occur in applications which use packet switching techniques, such as asynchronous transfer mode (ATM) for image/video transmission. In ATM, the effective channel capacity is not known in advance as the bandwidth is statistically allocated to a large number of data sources on a competetive basis. In addition, the image data may be lost in the event of cell losses due to network congestion. As a consequence the decoder should be able to decode and display the transmitted image(s) from the partially received image bitstream. Another factor affecting the system performance is that the image/video application has to compete with several other processes for the CPU resource if it is executed on a workstation with a multitasking environment. This is especially applicable in desktop video conferencing applications with multiple participants. Moreover, the same application may be executed on platforms with different processing powers. The coding and decoding processes are therefore influenced by a compuational bandwidth which in turn limits the number of bits decoded. Hence, bit rate scalability should provide a satisfactory performance by facilitating the reconstruction of image/video in the best possible SNR and/or spatial and/or temporal resolutions using only the available portion of the image/video bitstream. In other words, there should be a graceful degradation in performance in the event of loss of system resources.

3.2 SNR Scalability

In order to achieve SNR scalability, the image data is first reorganized into a fixed number of levels in either the spatial or transform domain. This is followed by compression and/or transmission of the various image levels in the decreasing order of their importance. In the spatial domain, the image can be reconstructed progressively either on a pixel by pixel basis, as in the bit plane and subsampling techniques, or by operating on larger data units obtained by preprocessing the image, as in residual error vector quantization. The transform domain techniques can be classified into coefficient scanning, bit slicing and S-transform. We now

review some of the spatial domain techniques which will be followed by a review of the transform domain techniques.

3.2.1 Spatial domain Techniques

Bit planes

A digital image quantized to k bits/pixel resolution can be represented as a set of k bit planes, where the kth bit plane corresponds to the kth bit of all the image pixels. Each bit plane is equal to the original image in size, as shown in Fig.3.2. SNR scalability is achieved by transmitting the bit planes successively starting with the most significant bit plane and ending with the least significant one. At the decoder, the image is reconstructed by gradually refining the gray levels of image pixels as more bit planes are received. This is a very simple technique and is capable of lossless reconstruction if all the bit planes are used. However, it achieves very low compression.

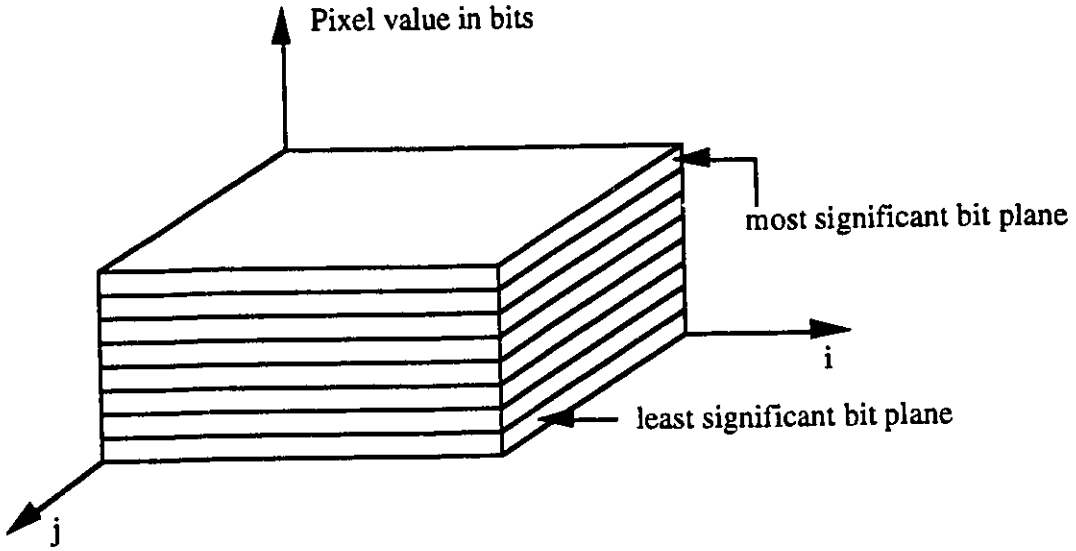


Figure 3.2: Slicing an 8 bit image into bit planes

Subsampling

Another simple spatial technique is the progressive subsampling approach. In this method, the image pixels are subsampled and then transmitted progressively. At the decoder, the missing pixels are interpolated using a variety of techniques such as bicubic splines, bilinear interpolation, cubic convolution etc. However, this approach does not yield a good performance [43].

Residual error vector quantization

Residual error vector quantization (REVQ) essentially employs a multi-stage vector quantization scheme [44]. In the first stage, the input image is vector quantized using a small size codebook and the corresponding labels are transmitted. A residual error image is formed by subtracting the reconstructed image of first stage from the original image. This forms the input to the second stage and is vector quantized using a separate codebook. The corresponding labels are used to improve the quality of the previously reconstructed image. Proceeding in a similar fashion, a residual error image is calculated at each level and vector quantized. Separate codebooks are used for each stage, as there is little correlation between the residual error images. A final stage with entropy coding is required for lossless reconstruction of the image. This technique achieves a higher coding performance compared to the bitplane and subsampling techniques because of the following two reasons. First, vector quantization provides a better performance than scalar quantization, as observed in section 2.5. Secondly, due to the use of iterative feedback the residual error images are decorrelated, which in turn results in a higher compression. The main drawback of REVQ is its computational overhead.

Tree-structured vector quantization

Another VQ based technique for achieving SNR scalability is tree-structured vector quantization (TSVQ) [45]. TSVQ uses multiple codebooks of different sizes which are organized in a tree-structure. Each level of the tree has only one codebook and corresponds to a particular SNR resolution. For the special case of a balanced binary tree (Fig.3.3), the size of the codebook increases by a factor of two at every level of the tree. In addition, each codeword of an intermediate level codebook has two children codewords, which belong to the next level codebook. The input image vectors are first quantized using an intermediate level codebook and the corresponding indices are transmitted to reconstruct the image at a low SNR resolution. Each vector in the reconstructed image is then progressively refined using additional bits for its index, which specify the reproduction codeword in the next level codebook. The image can thus be gradually improved until it is reconstructed at the highest SNR resolution. We note that this technique provides lossy compression. Lossless compression is achieved by transmitting the final residual error image (at the highest SNR resolution). We note that TSVQ has a lower computational complexity compared to normal VQ but it may result in a sub-optimal codebook at the highest resolution.

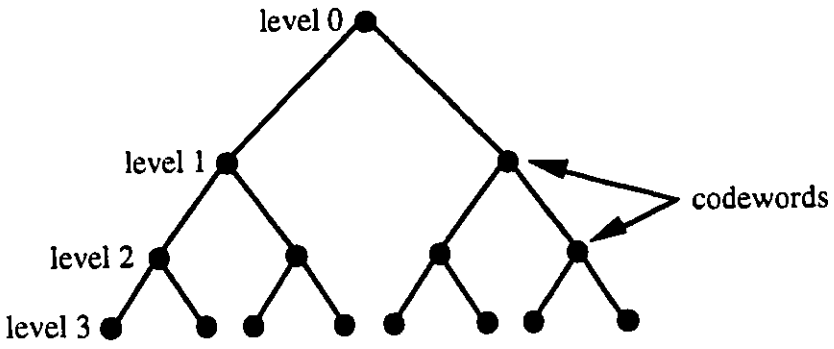


Figure 3.3: A tree-structured codebook for TSVQ

Nonuniform spatial pyramids

In general, the information contained in a natural image is not distributed uniformly over all its pixels. Nonuniform spatial pyramids exploit this feature of the images to yield high compression. In order to achieve this, a set of thresholds, $T_i : T_i > T_{i+1}$, are first chosen [51]. The original image is decomposed into four sub-images of equal size if the difference between the maximum and minimum values of the image pixels is greater than the first threshold, T_1. Each sub-image is recursively decomposed in a similar fashion, using threshold T_1, until a sub-image of size 1 pixel is reached. This forms a nonuniform pyramid corresponding to an approximation of the image. The same procedure is now repeated for the next threshold T_2. As $T_2 < T_1$, the resolution of the pyramid produced with threshold T_2, is higher than that produced with T_1. Furthermore, the lower resolution pyramid is contained in the higher resolution pyramid. We note that this process can be continued with the smaller values of thresholds. SNR scalability is achieved by reconstructing the image starting with the lowest resolution pyramid. This technique makes image identification easy, as high detail areas are reconstructed in the first few approximations. However, the compression ratios are low and there is a high computational overhead. In addition, no proper method exists for optimizing the threshold values.

In general, pyramidal techniques are more suitable for achieving spatial scalability and will be discussed in detail in section 3.3.1.

3.2.2 Transform domain Techniques

In addition to spatial domain, SNR scalability can also be achieved in the transform domain. We recall from section 2.3.2 that an image can be decomposed into a spectrum of decorrelated coefficients using an orthogonal transform. The magnitude of a coefficient corresponds to the amount of energy contributed by its spectral (or basis) function in the

image. The energy of most of the natural images is concentrated in the lower order coefficients. For example, in discrete cosine transform (DCT), the first (DC) coefficient corresponds to the mean value of all the pixels in an image block. Therefore, it is possible to reconstruct a reasonable approximation of the original image using only a few low order coefficients. This image can be further improved by the addition of higher order coefficients. As the coefficients are quantized in most of the transforms, the highest resolution image is lossy and a lossless reconstruction is possible only if a residual error image is transmitted. The transform domain techniques can be classified into three categories: band partitioning, bit slicing and S-transform.

Band Partitioning

In the band partitioning technique, the transform coefficients of the image are partitioned into a number of groups or bands by scanning them in a prespecified order. All the coefficient bands are compressed individually and stored/transmitted in the decreasing order of their importance. The image can be reconstructed in the lowest SNR resolution by decompressing and inverse transforming the first (and the most significant) band. In order to obtain higher SNR resolutions, the coefficients contained in the subsequent bands are inverse transformed and added to the lower resolution image. This process is continued until all the bands have been decoded to reconstruct the image at the highest SNR resolution.

In this approach, it is very important to choose the order in which the coefficients are scanned as it directly affects the formation of bands which in turn determines the image quality at different SNR resolutions. One popular method is to reorder all the AC coefficients in terms of the Normalized Mean Square Error (NMSE) of the reconstructed images. For this purpose, the reconstructed image for each AC coefficient is formed by inverse transforming only the DC and the corresponding AC coefficient. Another possible method is to reorder the coefficients according to their entropy [53]. We note that the

progressive mode of JPEG uses a fixed ordering of DCT coefficients. For this purpose, the 2-D array of coefficients is first converted into a 1-D array by scanning them in a zig-zag fashion (as in the baseline mode of JPEG discussed in section 2.4.1). We note that the coefficient energy generally decreases monotonically along the zig-zag scan. Details of the progressive mode will be presented in section 4.1. Other scan patterns reported in the literature include peano scan, peano-hilbert scan and similar space filling curves, etc [55]. These fixed scan patterns have been designed for a broad range of applications and may not be the best for a specific type of image.

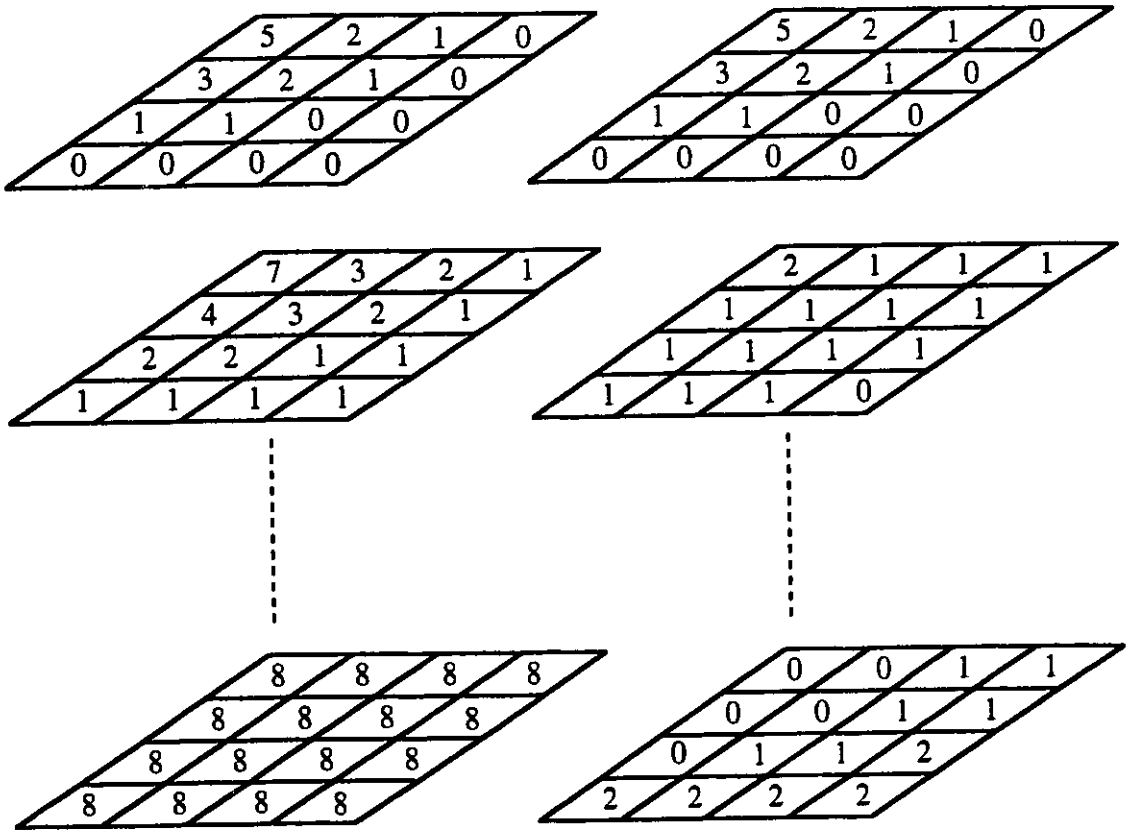


Figure 3.4: Bit maps (left) and incremental bit maps (right) for a block of size 4x4 pixels. Incremental bit rate = 1 bit/pixel.

Bit Slicing

Tzou and Elnahas [53] have described a bit slicing technique for achieving progressive improvement in the quality of the reconstructed image. In their technique, the transform coefficients are scalar quantized in a number of stages by using a set of incremental bit maps. Fig.3.4 shows an example of the bit maps, where the incremental bit rate per stage is 1 bit/coefficient. The total number of bits assigned to each coefficient is equal to the number of bits per pixel (in spatial domain). The sequence of bits allocated to each coefficient can be read from the corresponding location in the bit maps. This scheme can be viewed as a slicing of the full bit assignment map of the coefficients. The scheme has two disadvantages: first, it does not achieve any compression as each coefficient is eventually quantized at the same rate as the original image; second, each coefficient is assigned an equal number of bits i.e. bit allocation is not optimized.

S-Transform

The S-Transform is a hierarchical approach for image representation [56, 57]. It can be used for achieving both SNR and spatial scalabilities: the SNR scalability implementation has been discussed in this section, while that for sptial scalability will be presented in section 3.3.2. The input image is first partitioned into spatially nonoverlapping blocks of size 2 x 2 pixels. Each block of pixels is then transformed into 4 coefficients using Hadamard transform (equation 3.1).

$$d_0 = \frac{1}{4}(x_1 + x_2 + x_3 + x_4)\,, \quad d_1 = \frac{1}{4}(x_1 + x_2 - x_3 - x_4)\,,$$
$$d_2 = \frac{1}{4}(x_1 - x_2 - x_3 + x_4)\,, \quad d_3 = x_1 - x_2 + x_3 - x_4\,, \tag{3.1}$$

where,

d_0 = DC coefficient, d_1, d_2, d_3 = AC coefficients,

x_1, x_2, x_3, x_4 = pixel values in an image block.

The first (DC) coefficient, d_0, is the average of the four pixels in the block, while the other three, d_1, d_2 and d_3, are high frequency (AC) coefficients. A sub-image having half the dimensions of the original image can be obtained by assembling the DC coefficients of all the image blocks. Similar sub-images can be obtained for the three AC coefficients. The DC coefficient sub-image can be further decomposed by succesive application of Hadamard transform to form a pyramidal structure. In the limiting case, the DC sub-image contains only one pixel. SNR scalability is achieved by first displaying the smallest DC sub-image, upsampled to the size of original image. This image is improved by adding information from the corresponding AC sub-images to obtain next SNR resolution. This process is repeated until the image is reconstructed at the highest resolution.

S-Transform decomposes the image into a hierarchy of sub-images without providing any compression. It is necessary to use some compression technique on the sub-images in order to achieve a good coding performance. We note that the operation of S-transform is very similar to wavelet transform as both of them decompose the image to form a sequence of sub-images arranged in a pyramidal structure.

3.3 Spatial Scalability

A simple approach for reconstructing the original image at small spatial resolutions is to subsample the full size reconstructed image to the required size. However, this strategy is wasteful in terms of transmission and computational bandwidths. Therefore, the image data is generally organized in a hierarchical fashion, such that each level of the hierarchy corresponds to a specific spatial resolution of the image. We recall from the previous section that a pyramidal structure reorganizes the image data in a hierarchy of increasing spatial resolutions. As a result, almost all techniques for achieving spatial scalability employ pyramidal data structures. We now review some of these techniques.

3.3.1 Spatial domain techniques

Spatial pyramids

Pyramid data structures provide a natural way of representing images in a hierarchical fashion. A pyramid data structure reorganizes the input image X as a sequence of matrices, $\{X_i\}$; $i = 0,1,...,N-1$ such that $\{X_{i-1}\}$ is a reduced resolution version of the $\{X_i\}$ image. In the limiting sense, X_0 is a single pixel matrix, while X_{N-1} corresponds to the original image. Since each level of the pyramid represents the image at a different spatial resolution, it is possible to achieve spatial scalability by transmitting the images in the pyramid, starting from the topmost level.

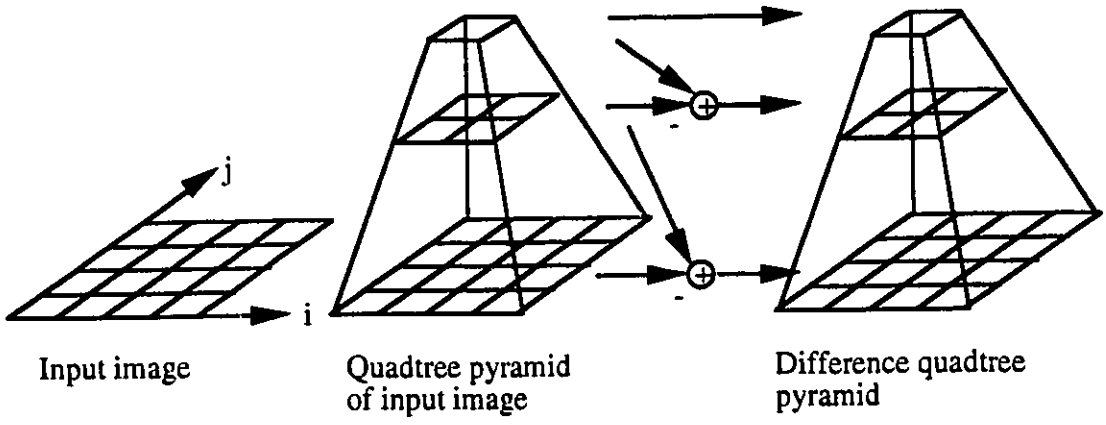


Figure 3.5: Formation of a uniform quadtree pyramid

The simplest pyramid is a quadtree (fig.3.5) which is constructed as follows: Given an image of size $2^n \times 2^n$, a pyramid data can be formed by successively operating on the neighbouring 2 x 2 pixels [46-50]. In other words, the value of a pixel in $\{X_{i-1}\}$ is a function of the corresponding 2 x 2 pixels in $\{X_i\}$. The function can be the mean, minimum, maximum, median, sum, etc. As each pixel at level $i-1$ is associated with four siblings at a lower level i, this data structure is called a uniform quadtree. An example is a quadtree

formed using truncated mean in which the pixels at all levels are truncated to integer values. It can be seen that such a data structure results in a 33.3% additional overhead for representing images. Many techniques have been proposed in the literature to reorganize the levels of the pyramid for minimizing redundant information. The different levels are then individually compressed and stored/transmitted. We now review some of these techniques.

Hierarchical interpolation

The simplest pyramid is obtained by a strict subsampling of the image. Here, each parent node belonging to level $k-1$ of the pyramid resembles one of its four siblings at level k. In order to achieve compression, a difference pyramid is formed to store the difference between the parent node and the remaining three siblings, at each level of the multiresolution pyramid (Fig.3.5). The topmost level of the multiresolution pyramid corresponds to the lowest resolution image. Reconstruction of higher spatial resolutions requires interpolation of the missing pixels. This is performed by recursively adding the difference values to the parent nodes until an exact reconstruction of the image is achieved. There are two disadvantages with this scheme. First, the use of subsampling without any prefiltering results in aliasing artifacts in the lower resolution images. Secondly, the subsampled pixels may not be a good representatives of their respective image regions.

Difference pyramid vector quantization

Wang and Goldberg [58] have suggested a quadtree based approach for progressive image transmission (i.e. SNR scalability). Their technique achieves SNR scalability by first reconstructing the image at different spatial resolutions and is therefore more suitable for achieving spatial scalability. In this technique, a mean quadtree is first formed by partitioning the original image into blocks of 2 x 2 pixels and computing the average pixel

value of each block. From this mean pyramid, a difference pyramid is formed. The difference pyramid has a lower first order entropy than the mean pyramid and can therefore be compressed more efficiently. Each level of the difference pyramid is then vector quantized using a separate codebook. The receiver reconstructs the lowest resolution image and all the subsequent levels of the difference pyramid using their corresponding labels and codebooks. The higher resolutions of the image are obtained by recursive addition of the difference pyramid levels to the lowest resolution image. This is a highly efficient technique as compared to hierarchical interpolation.

Knowlton's technique

Knowlton has introduced a novel approach to form a spatial domain pyramid [36]. His technique uses a reversible transformation, to map pixels of the original image into a new set of values, resulting in a data structure which resembles mean pyramid. This transformation maps k-bit values of adjacent pairs of pixels into a k-bit composite value and a k-bit differentiator. The composite and the differentiator values are roughly equal to the mean of the two pixels and their difference biased with the middle gray scale value (128 for an image with 256 gray levels), respectively. This mapping can be easily performed using a simple algorithm or a look-up table [60]. Different levels of the pyramid are obtained by recursive application of the mapping process, alternating between the horizontal and vertical pairs. The number of bits required by the differentiators is equal to the number of bits in the original image, and therefore this technique does not increase the size of image data during pyramid formation. As the differentiator values have a non-uniform distribution, they can be efficiently compressed using variable-length codes. The image is hierarchically reconstructed at different spatial resolutions using the composite value at the top of the pyramid and all the differentiators. A disadvantage of Knowlton's technique is that at lower

levels of the pyramid, the composite values do not adequately represent the image, resulting in an inferior image quality.

Filtered pyramids

Burt and Adelson have proposed a general method for formation of pyramids [48]. In their approach, the original image $X = G_0$ is first low pass filtered by convolving it with a Gaussian-like weighting function. The low pass filtered gaussian image, G_1, is then subtracted from G_0 to produce a difference image L_0. Due to its low pass characteristics, the image G_1 can be subsampled by a factor of two in both directions without losing much information. On the other hand, L_0 is largely a decorrelated image and can therefore be compressed using fewer bits. This results in an overall reduction in the data to be stored/transmitted. The image G_1 can be further decomposed by succesive low pass filtering and downsampling of the Gaussian images resulting in the formation of a Gaussian pyramid. A Laplacian pyramid can be similarly obtained by assembling all the difference images L_i. The Laplacian pyramid is then encoded using techniques such as run-length coding, vector quantization, etc. The image is reconstructed in the reverse order of decomposition using the lowest resolution Gaussian image and all the Laplacian images. The size of the image is increased by a factor of two in both directions at each step of reconstruction by adding the corresponding level of the Laplacian pyramid to the (upsampled) lower resolution image.

A variation of the Burt and Adelson's technique is the filtered pyramid obtained using Quadrature Mirror Filters (QMF) [14, 15]. For this purpose, a 2-D QMF is constructed by applying a 1-D QMF successively in the horizontal and vertical directions, respectively. This process decomposes the image into four subbands producing one Gaussian (low pass filtered in both directions) and three Laplacian images. The Gaussian image can be further decomposed by recursive application of 2-D QMFs resulting in a Gaussian and a Laplacian

pyramid, respectively. Each level of the Laplacian pyramid consists of three images, corresponding to the high frequency contents in one or both directions. The image is reconstructed in the reverse order of decomposition to provide different spatial resolutions. We recall from section 2.6 that wavelet decomposition can be easily performed with QMF filters. However, QMFs are a small subset of the wavelet theory which is more general in its nature and scope.

3.3.2 Transform domain techniques

We recall from section 3.2.2 that the energy of most natural images is concentrated in the lower order transform coefficients. As a result, smaller spatial resolutions typically contain lower spatial frequencies of the original image. Therefore, it is possible to reconstruct an image in smaller sizes by reorganizing the transform coefficients in a pyramidal data structure to obtain a hierarchical image representation. Many spatial domain pyramidal techniques, such as filtered pyramids, are often used in conjunction with the transform domain techniques to provide a good coding performance. For example, a high compression ratio can be obtained in Burt and Adelson's technique by compressing the Gaussian and Laplacian pyramids using JPEG-DCT. In almost all the cases, the image is reconstructed at the lowest spatial resolution by decompressing the bitstream corresponding to level *0* of the pyramid. This image is then zoomed to a higher spatial resolution by upsampling and its quality is improved by adding information from the decompressed bitstream of the corresponding level. The entire process is repeated until the image is reconstructed at the highest spatial resolution.

Once again due to coefficient quantization, the full size reconstructed image is lossy and hence a lossless reconstruction is possible only if the residual error image is stored/transmitted. We now review some of the transform domain techniques proposed in the literature.

S-Transform

We recall from section 3.2.2 that the S-Transform can be used for achieving both SNR and spatial scalabilities. We now discuss its application for implementing spatial scalability. S-transform hierachically decomposes the input image into a sequence of low pass and high pass sub-images organized in a pyramidal structure. The low pass sub-images represent the original image in different spatial resolutions. The various sub-images are individually compressed using techniques such as runlength coding, vector quantization, etc.

We recall from section 2.6 that wavelet transform decomposes the image into a pyramid of sub-images similar to the S-transform. However, the lower resolution images obtained with wavelet transform are of superior subjective quality.

Spatial domain pyramid

Hanamura *et. al.* [39] have proposed a hierachical coding scheme for scalable video. As a special case, their coding scheme can also be used for scalable compression of still images. They have introduced a downsampling filter (Fig.3.6) for constructing a spatial domain pyramid which results in visually pleasing images at lower spatial resolutions. A complimentary filter for upsampling the low resolution images has also been introduced. The pixel values of the downsampled and upsampled images in Fig.3.6 are calculated using the following equations:

$$d = \frac{1}{64}[(u_1 + u_4 + u_{13} + u_{16}) - 5\,(u_2 + u_3 + u_5 + u_8 + u_9 + u_{12} + u_{14} + u_{15}) + 25\,(u_6 + u_7 + u_{10} + u_{11})] \qquad (3.2\text{ a})$$

$$u = \frac{1}{6}[d_1 + 4d_3 + d_2] \qquad (3.2\text{ b})$$

where, d = value of a pixel in the downsampled image,

u = value of a pixel in the upsampled image.

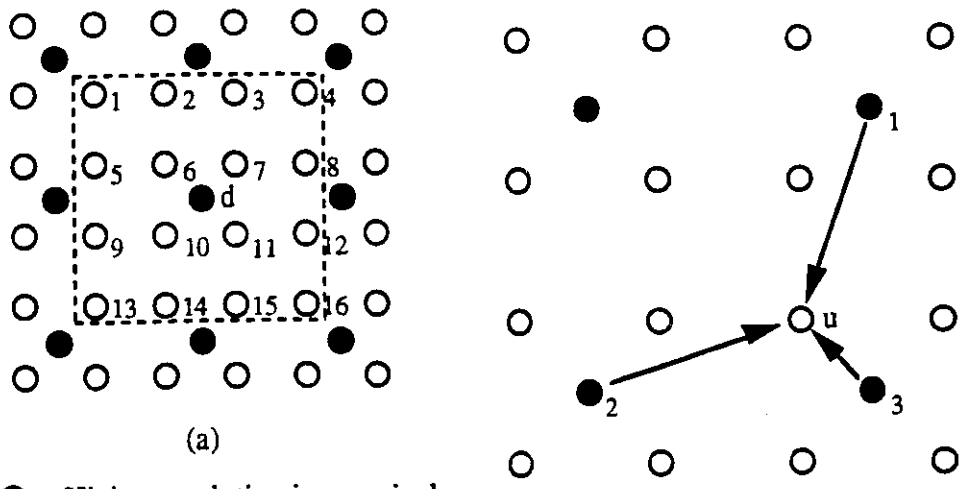


Figure 3.6: Hanamura's technique (a) Downsampling filter (b) Upsampling filter

A difference pyramid is formed by computing the difference between the upsampled images and the corresponding downsampled versions of the original. For still images, all the levels of the difference pyramid are individually compressed using JPEG-DCT.

Frequency domain pyramid

Gonzales and Viscito [38] have proposed to construct a multi-resolution pyramid in the DCT coefficient domain for scalable video coding. As a special case, their coding scheme can also be used for scalable compression of still images. In their technique, the original image is first filtered and downsampled to form a quadtree. For a four level quadtree, the images at 1st, 2nd and 3rd levels are partitioned into non-overlapping blocks of sizes 1x1, 2x2 and 4x4 pixels, respectively. The original image present at the lowest (4th) level of the quadtree is partitioned into blocks of size 8 x 8 pixels. The image blocks at all the levels are then transformed using n x n DCT where n = no. of pixels per dimension in a block . For

efficient compression, the DCT coefficients at each layer are used to predict the corresponding low frequency coefficients in the next higher layer, as shown in Fig.3.7. All the coefficients are then quantized. The predicted coefficents together with the new (non-predicted) coefficients are then scanned in a zig-zag fashion and runlength and huffman coded using the JPEG (MPEG-1 for video) tables.

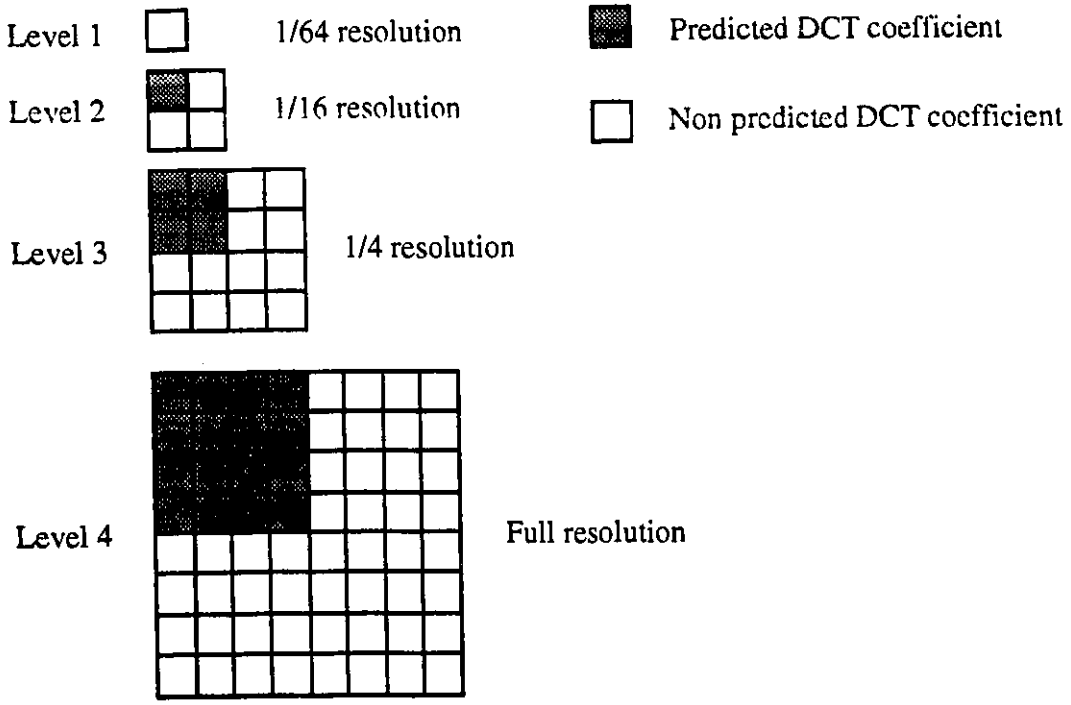


Figure 3.7: Predictive coding of DCT coefficients in a pyramidal fashion

One key difference between the frequency and spatial pyramids is that frequency pyramids use DCTs of smaller size in the lower layers, while spatial pyramids generally use 8 x 8 DCT as recommended in the JPEG (and MPEG-1) standard(s). A detailed performance comparison between the n x n DCT , n = 2, 4 and 8 x 8 DCT operations shows that the former are more efficient in terms of execution time which is a clear advantage for software encoding and decoding. However, 2 x 2 and 4 x 4 DCT encoding requires more side

information in the lower layers as compared to 8 x 8 DCT which may result in a lower compression performance.

3.4 Summary

In this chapter, we first defined the concept of scalability and described its three types. In the context of still images, only SNR and spatial scalabilities are applicable. A more comprehensive concept called bit rate scalability was then defined. This followed with a review of the various spatial and transform domain techniques for achieving SNR and spatial scalabilities.

In the case of SNR scalability, direct spatial domain techniques do not provide good performance unless powerful techniques such as vector quantization are used. Transform domain techniques offer high compression efficiency since most of the energy in natural images is concentrated in a few low order coefficients. Some of the techniques for SNR scalability in transform domain include band partitioning, bit slicing and S-transform.

Spatial scalability is generally implemented by reorganizing the image data in a pyramidal data structure where each level of the pyramid corresponds to a specific spatial resolution of the image. Difference pyramid vector quantization is a powerful technique for achieving spatial scalability. Filtered pyramids (e.g. Burt and Adelson's method) require a spatial or transform domain technique for achieving compression. Most of the high performance transform domain techniques for achieving spatial scalability are DCT based.

In the next chapter, we will investigate the implementation of SNR and spatial scalabilities within the framework of JPEG standard which employs DCT as the (lossy) compression technique.

Chapter 4

JPEG based Scalable Image Compression

In this chapter, we present scalable image compression algorithms within the framework of JPEG standard. We recall from section 2.4 that the JPEG standard has four modes of operation:

1. Baseline sequential
2. Lossless compression
3. Progressive coding mode
4. Hierarchical coding mode

The baseline sequential and lossless compression modes provide the basic feature of lossy and lossless compression, respectively. In this chapter, we first present the progressive coding mode, which provides SNR scalability using two different techniques. We then present the hierarchical coding mode which provides spatial scalability using a pyramidal approach. This follows with the implementation details and simulation results of the progressive and hierarchical modes for performance evaluation and comparison. A high-performance wavelet based technique is then proposed for enhancing the performance of hierarchical mode. Finally, we investigate and compare the performance of transmitting

images over ATM networks compressed using the baseline, progressive and hierarchical modes, respectively.

4.1 Progressive coding mode

In the progressive (DCT-based) mode, the image data (i.e. quantized DCT coefficients) is decomposed into several layers. The higher and lower layers correspond to high priority and low priority information, respectively. Each layer is compressed and stored separately. A first approximation image is reconstructed using only the highest layer. This image is progressively improved by gradually adding the information from the lower layers until a full quality image is obtained. The standard has proposed two techniques for decomposing the DCT coefficients: (1) Spectral selection and (2) Successive approximation.

4.1.1 Spectral selection

In spectral selection (SNR-SS), the DCT coefficients are partitioned into a number of bands according to their frequencies and each band is compressed separately using the enhanced sequential algorithm (ESA) (refer section 2.4). The first band is generally reserved for the DC coefficient. The AC coefficients (zig-zag scanned) are split amongst the remaining available bands as shown in Fig.4.1.

Each band is defined by a header which specifies the indices corresponding to its first and last coefficients. A coarse quality image is obtained by first decoding the DC band. In order to improve the quality of this image, the AC bands are gradually decoded (by computing their inverse DCT), and added to the coarse resolution image. This is possible due to linearity of DCT. Hence, spectral selection reconstructs the image in different qualities (SNR resolutions) thereby providing SNR scalability.

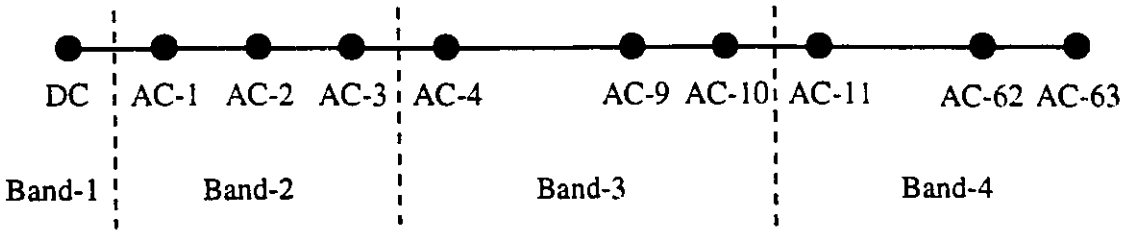


Figure 4.1: Spectral selection technique of JPEG-Progressive mode

4.1.2 Successive approximation

In successive approximation (SNR-SA), all the (quantized) DCT coefficients are first encoded at a reduced precision. In other words, the coefficients are sliced into a number of bit planes as shown in Fig.4.2, which are then individually compressed. The most significant bit (MSB)

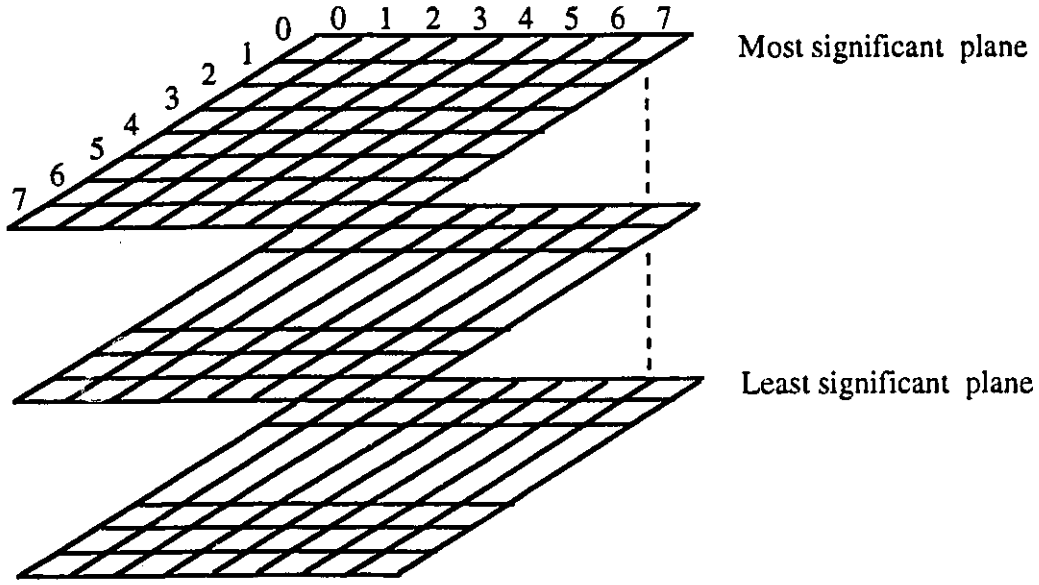


Figure 4.2: Successive approximation technique of JPEG-Progressive mode

plane contains the coefficients at a reduced precision and is compressed using ESA. Each lower bit plane improves the coefficient precision (by 1 bit), and is compressed using a modified version of ESA to achieve high compression. In the successive approximation technique, the first approximation image, reconstructed from only the MSB plane, has both low and high frequency information which results in a reasonable reconstruction of the edges at low bit rates. However, there are distortions or artifacts which are eliminated by gradually adding the information from the lower bit planes. Thus, we obtain images in progressively improving qualities (i.e. SNR scalability).

We note that in both the spectral selection and successive approximation techniques, a buffer is required between the quantizer and the huffman coder to store the DCT coefficients as the compression is performed in multiple passes. These two techniques can be combined to provide a more flexible implementation of SNR scalability: for example, spectral selection within successive approximation.

4.2 Hierarchical coding mode

The hierarchical mode of JPEG standard proposes a pyramidal structure for encoding different spatial resolutions of the image (Fig.4.3). The spatial resolution of the image increases from top level (I_0) to the bottom level (I_{N-1}) of the pyramid. Spatial scalability is achieved by first displaying the lowest resolution image which is gradually zoomed, based on user demand, to progressively higher resolutions (as required in image browsing). Alternatively, the image may be displayed simultaneously on multiple monitors, each with a different spatial resolution (as in simultaneous transmission of NTSC and HDTV). However, the standard does not specify the downsampling filters to be used for obtaining the progressively lower size images. This is due to the fact that the JPEG standard has been developed for a large spectrum of applications, each of which may have its own performance requirements and processing capability. In order to achieve interoperability

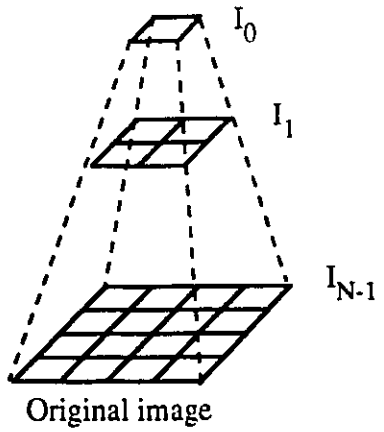


Figure 4.3: Pyramidal structure used in JPEG hierarchical mode.

between different application environments, a simple mean interpolation filter has been specified in the standard to upsample lower resolution images by a factor of two in both the horizontal and vertical directions. The upsampling filter facilitates computation of the difference images as shown in Fig.4.4. The image is first reduced to the smallest size I_0 (corresponding to an icon for a browsing application or to the resolution of low end receivers for transmission applications) by successively downsampling the original image I_{N-1}. The iconic image I_0 is then compressed and the resulting bitstream is decoded to reconstruct the image at the lowest spatial resolution, R_0. To transmit the image at the next spatial resolution I_1 in the differential form, R_0 is first upsampled (by a factor of two in both directions) to the next higher size image, U_0. A difference image D_1 is calculated by computing pixelwise difference between I_1 and U_0. D_1 is compressed and the resulting bitstream is decoded to obtain the reconstructed difference image D_1'. The image at the next spatial resolution, R_1 is obtained by adding U_0 to D_1'. Proceeding in a similar fashion, R_1 is upsampled to the next higher size image U_1. To improve the quality of U_1, the image D_2 ($= I_2 - U_1$), representing the required enhancement information, is compressed and transmitted. The entire procedure is repeated until the final difference image (D_{N-1}) has been encoded.

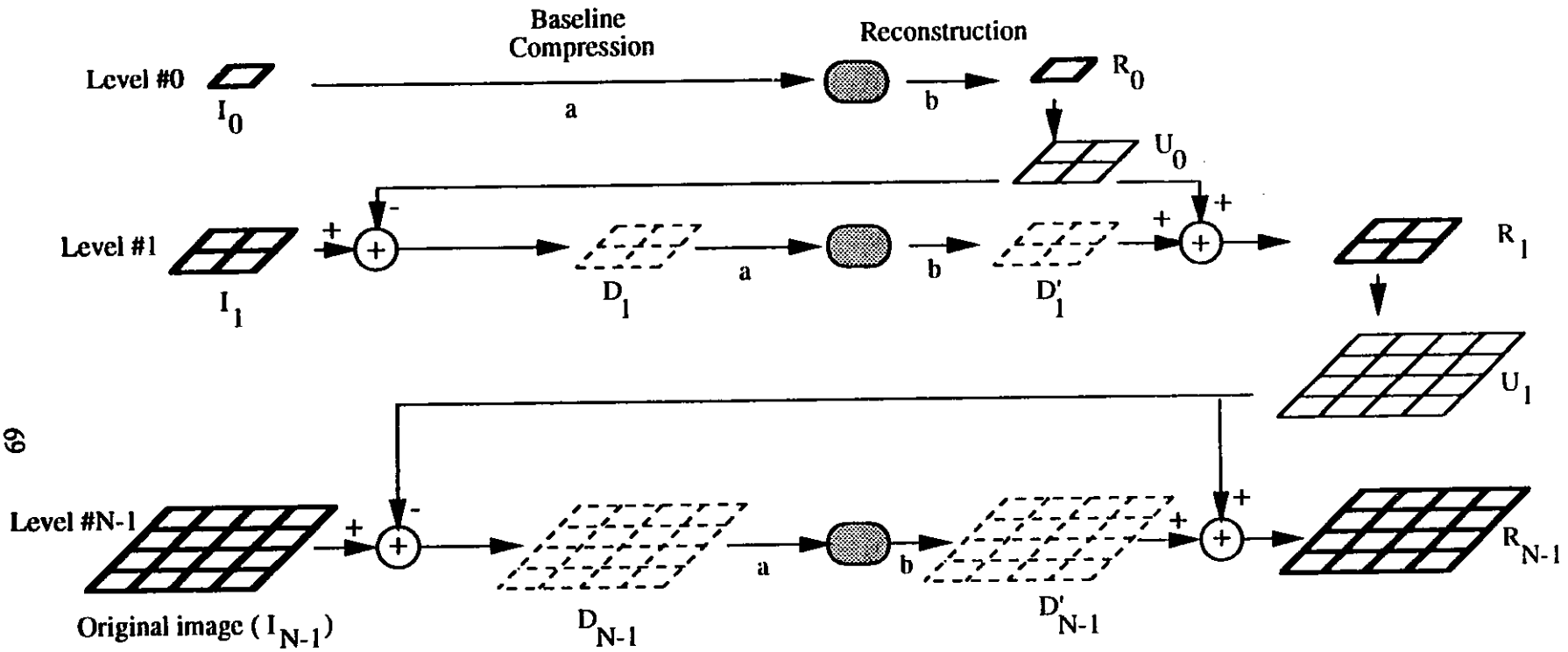


Figure 4.4: Implementation of JPEG- hierarchical Mode. I = downsampled image, D= difference image, D' = reconstructed difference image, R = reconstructed image, U = upsampled image; a= baseline compression, b= baseline reconstruction.

Images of different sizes may be displayed by first reconstructing the smallest (iconic) image R_0. Bitstreams corresponding to the difference images (D_i) are then decoded and added to the respective upsampled images (U_i) until the full resolution image R_{N-1} is obtained. We note that the JPEG hierarchical encoder has to use both downsampling and upsampling filters and perform both compression and decompression operations, respectively. This is necessary to reconstruct the image at all spatial resolutions R_i which are in turn used to compute the difference images D_i. On the other hand, the JPEG decoder reconstructs the images R_i by using only decompression and upsampling operations.

The lowest resolution image I_0 and the difference images D_i are compressed using either (1) baseline sequential, or (2) progressive (DCT based) coding mode. The baseline sequential mode provides a fixed picture quality at every spatial resolution, whereas the progressive mode can be used to gradually improve the picture quality at each spatial resolution (i.e. SNR scalability within spatial scalability). We note that the pixel values of the difference images are usually centered around zero. Therefore no *level shift* operation is performed before calculating the DCT coefficients of the difference images. As a further modification, the DC coefficients are compressed without using DPCM. This is due to insignificant correlation between the DC coefficients of adjacent blocks in the difference images. In order to achieve an exact reconstruction at the highest spatial resolution, the final difference image may be compressed using the lossless mode. For applications which require perfect reconstruction at all spatial resolutions, the lowest resolution image I_0 and the difference images D_i are compressed using the lossless mode.

4.3 Simulation Results and Discussions

In this section, we first present the detailed implementation of the SNR and spatial scalabilities using the progressive and hierarchical coding modes of JPEG, respectively. Simulations have been performed on two test images: "Lena" and "Baboon", each of size

Figure 4.5(a): The original Lena image

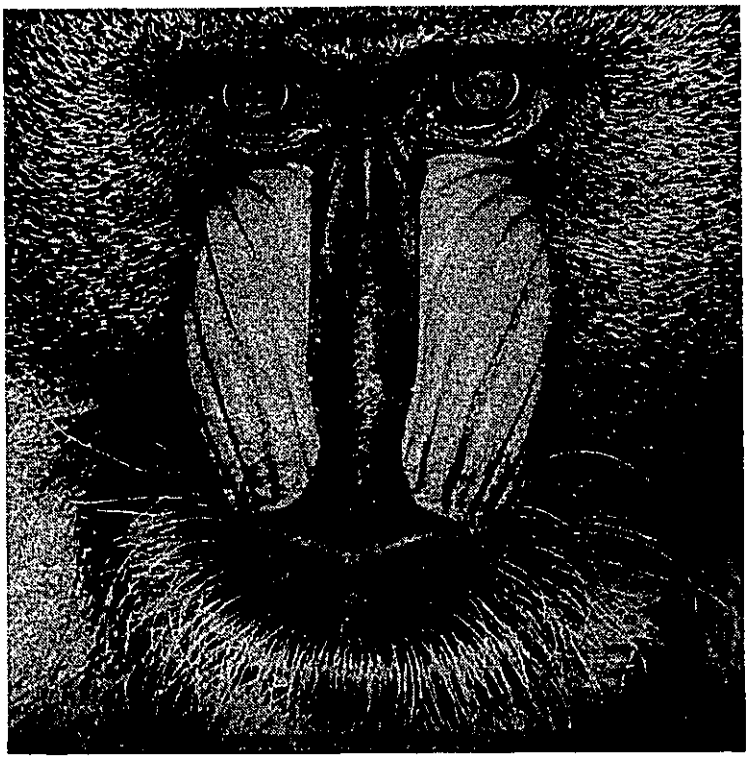


Figure 4.5(b): The original Baboon image

Figure 4.6(a): Baseline JPEG (BSA) encoded Lena image reconstructed at 0.715 bits per pixel

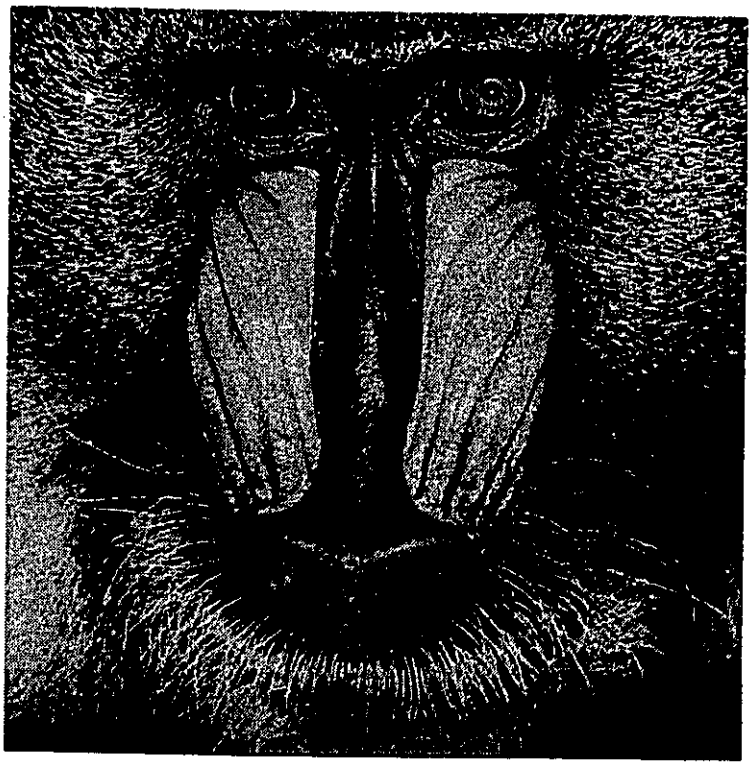


Figure 4.6(b): Baseline JPEG (BSA) encoded Baboon image reconstructed at 1.391 bits per pixel

Table 4.1: Performance Comparison of Progressive mode techniques with Baseline JPEG (BSA).

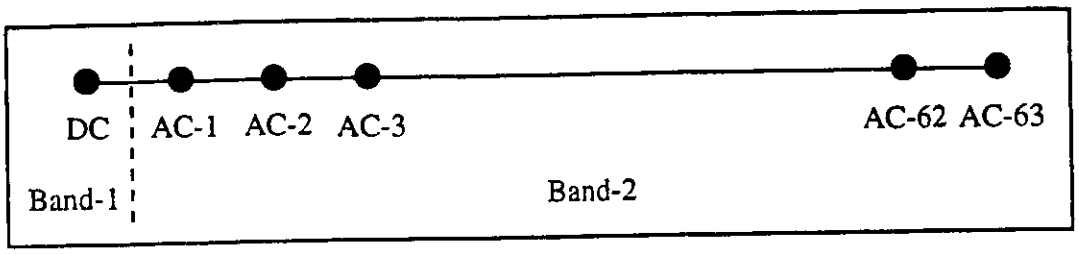
Algorithm	Lena		Baboon	
	Bit rate (bits per pixel)	PSNR (dB)	Bit rate (bits per pixel)	PSNR (dB)
Baseline JPEG (BSA)	0.092	22.62	0.090	19.7
	0.430	29.0	0.451	21.73
	0.523	30.47	0.876	23.9
	0.620	32.28	1.16	25.81
	0.715	34.44	1.391	28.23
Spectral Selection				
(a) SNR-SS-I	0.092	22.62	0.090	19.7
	0.715	34.44	1.391	28.23
(b) SNR-SS-II	0.092	22.62	0.090	19.7
	0.372	28.53	0.445	21.73
	0.729	34.44	1.395	28.23
Successive Approximation				
(a) SNR-SA-I	0.541	30.89	1.036	23.85
	0.820	34.44	1.594	28.23
(b) SNR-SA-II	0.480	25.48	0.891	18.31
	0.616	30.89	1.184	23.85
	0.908	34.44	1.782	28.23

512 x 512 pixels quantized to 8 bits/pixel. The original images and the BSA reconstructed images are shown in Figs.4.5 and 4.6, respectively. We note that (a) and (b) parts of all the image figures correspond to "Lena" and "Baboon" images, respectively. The quantization tables suggested in JPEG standard have been used in all the simulations. The coding performance has been evaluated and compared with the BSA using both objective and subjective criteria. The objective image quality has been evaluated using Peak Signal to noise ratio (PSNR) (refer section 2.1.3).

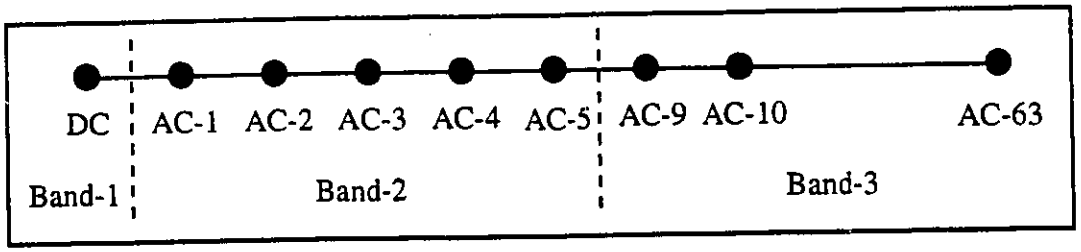
4.3.1 SNR scalability

We recall from section 4.1 that the progressive mode provides SNR scalability by decomposing the DCT coefficients into several layers (of different visual importance) using the spectral selection (SNR-SS) and successive approximation (SNR-SA) techniques. In SNR-SS, two sets of experiments have been performed in which the DCT coefficients have been partitioned into two (SNR-SS-I) and three (SNR-SS-II) bands, respectively. In both the experiments, the first band is reserved only for the DC coefficient. In the first experiment, SNR-SS-I, the second band contains all the AC coefficients (Fig.4.7a). In the second experiment, SNR-SS-II, the second band contains the first 5 AC coefficients (in the zig-zag scan) while the remaining coefficients are contained in the third band (Fig.4.7b). The simulation results for both the experiments have been tabulated in Table 4.1. It can be seen that the spectral selection approach provides SNR scalability (for the same PSNR values) with (almost) no increase in the bit rate compared to baseline JPEG algorithm. Figs.4.8 and 4.9 show the images reconstructed using only the DC and DC+first 5 AC coefficients, respectively for subjective evaluation and comparison with the BSA reconstructed images (Fig.4.6). It can be observed that the first approximation image reconstructed using only the DC coefficients suffers from extreme blocking effects.

However, the image quality improves substantially with the progressive addition of the AC coefficients. Note that the full quality image is exactly the same as in BSA (Fig.4.6).



(a)



(b)

Figure 4.7: (a) SNR-SS-I (b) SNR-SS-II

In the SNR-SA technique, two sets of experiments have been performed. In the first experiment, SNR-SA-I, the AC coefficients have been sliced into an MSB plane of 7 bits and an LSB plane of 1 bit (Fig.4.10a) . In the second experiment, SNR-SA-II, the MSB plane consists of 6 bits, and the remaining 2 bits have been sliced into two bit planes (1 bit each), resulting in a total of 3 bit planes (Fig.4.10b). We note that in both the experiments, DC coefficients have been compressed at full precision and stored in the MSB plane. Table 4.1 compares the performance of the two experiments with the BSA. The images reconstructed using only 6 and 7 MSBs are shown in Figs.4.11 and 4.12, respectively. It can be seen from Table 4.1 that the successive approximation technique results in a higher bit rate compared to the BSA. Most of the bit rate is contributed by the LSB planes which are difficult to compress due to random distribution of values. However, it can be seen

Figure 4.8(a): Lena reconstructed using only DC coefficients (SNR-SS) at 0.092 bits per pixel

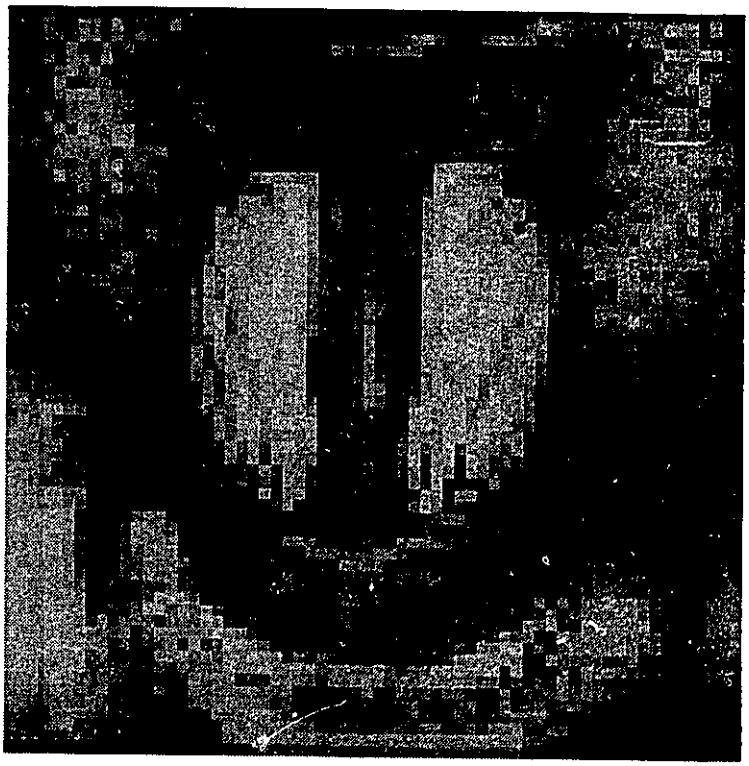


Figure 4.8(b): Baboon reconstructed using only DC coefficients (SNR-SS) at 0.09 bits per pixel

Figure 4.9(a): Lena reconstructed using DC + first 5 AC coefficients (SNR-SS-II) at 0.372 bits per pixel

Figure 4.9(b): Baboon reconstructed using DC + first 5 AC coefficients (SNR-SS-II) at 0.445 bits per pixel

from the images that this technique results in good subjective image quality at intermediate SNR levels, due to reasonable reconstruction of the edges and other high frequency information.

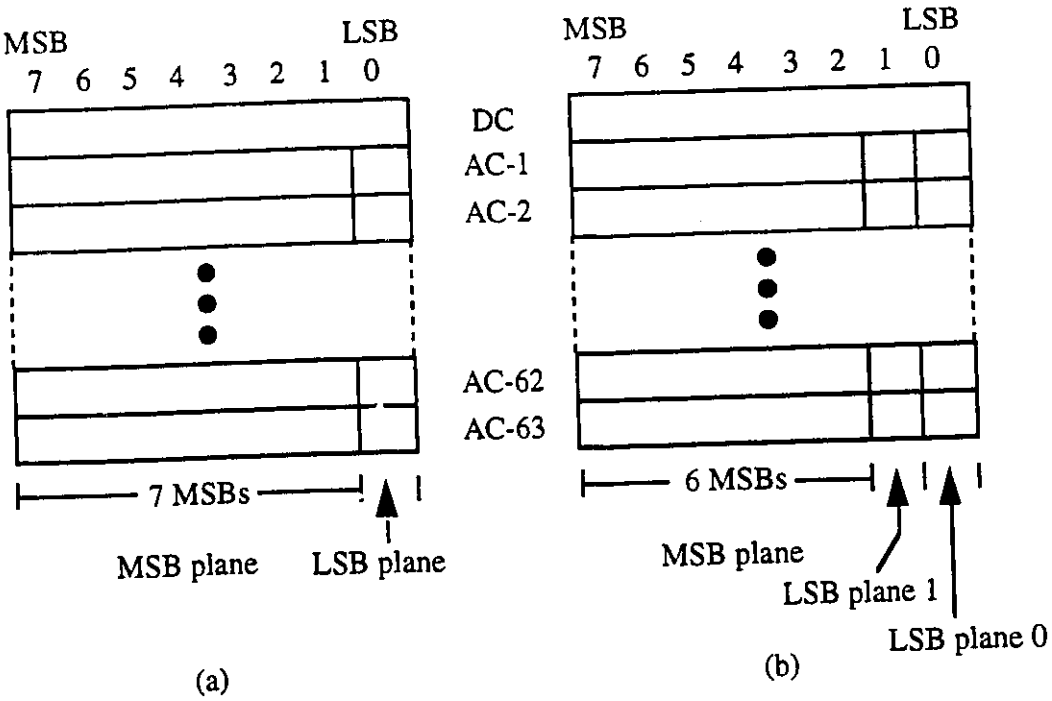


Figure 4.10: (a) SNR-SA-I (b) SNR-SA-II

In summary, the spectral selection technique provides SNR scalability at no additional cost, while the successive approximation technique requires a higher bit rate compared to BSA. However, successive approximation results in good reconstruction of edges and other high frequency information.

4.3.2 Spatial scalability

We recall from section 4.2 that the hierarchical mode does not specify the downsampling filters to obtain smaller size images. In our simulations, a simple mean filter has been used to reduce the image to half the original size in both the dimensions by averaging each block

Figure 4.11(a): Lena reconstructed using only 6 MSBs of DCT coefficients (SNR-SA-II) at 0.480 bits per pixel

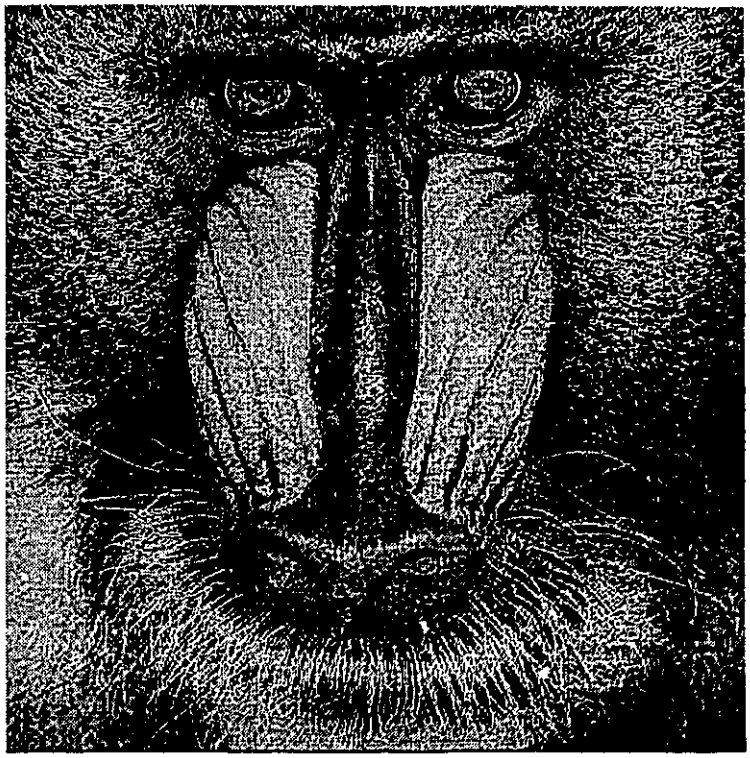


Figure 4.11(b): Baboon reconstructed using only 6 MSBs of DCT coefficients (SNR-SA-II) at 0.891 bits per pixel

Figure 4.12(a): Lena reconstructed using only 7 MSBs of DCT coefficients (SNR-SA-II) at 0.616 bits per pixel

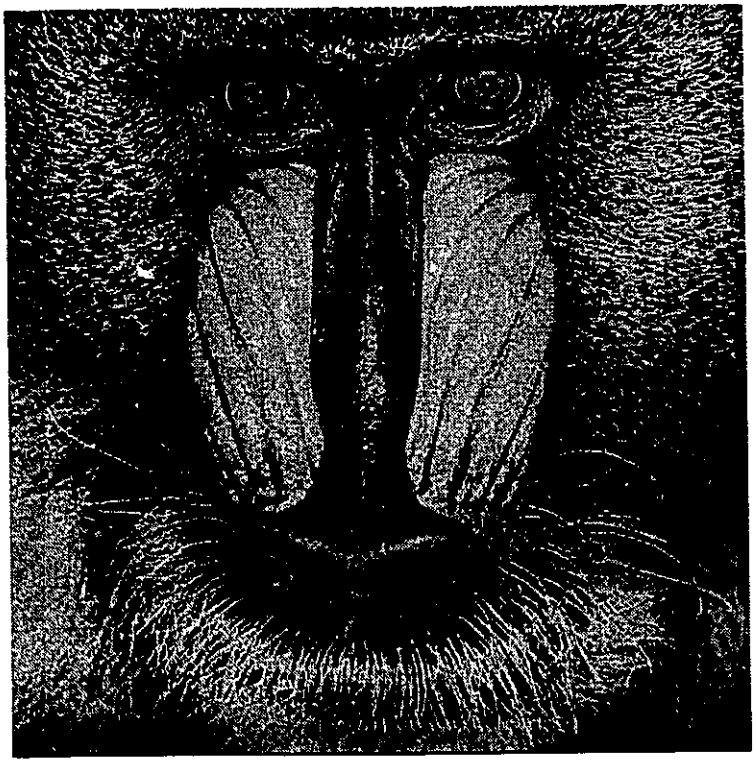


Figure 4.12(b): Baboon reconstructed using only 7 MSBs of DCT coefficients (SNR-SA-II) at 1.184 bits per pixel

(i)

(ii)

(iii)

(iv)

Figure 4.13(a): Lena reconstructed at (i) 64 x 64, (ii) 128 x 128, (iii) 256 x 256, and (iv) 512 x 512 pixel resolutions using the SPS-MF technique.

(i)

(ii)

(iii)

(iv)

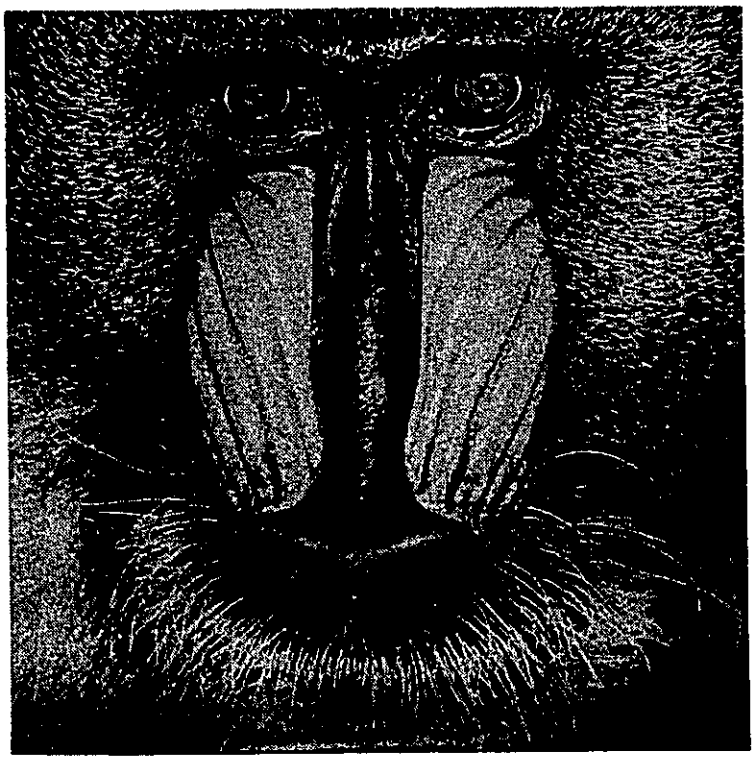
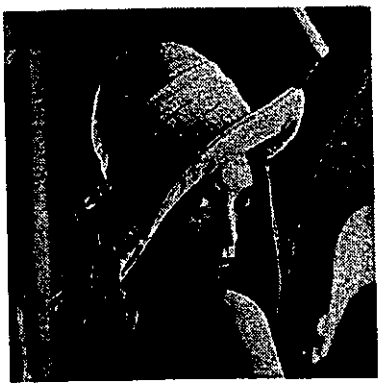


Figure 4.13(b): Baboon reconstructed at (i) 64 x 64, (ii) 128 x 128, (iii) 256 x 256, and (iv) 512 x 512 pixel resolutions using the SPS-MF technique.

(i)

(ii)

(iii)



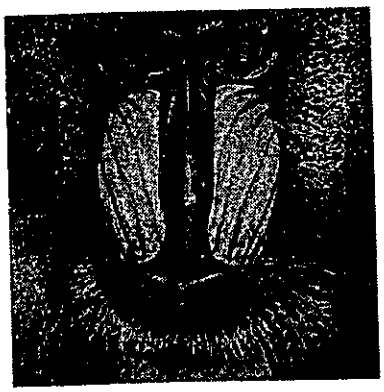
(iv)

Figure 4.14(a): Lena reconstructed at (i) 64 x 64, (ii) 128 x 128, (iii) 256 x 256, and (iv) 512 x 512 pixel resolutions using the SPS-WF technique.

(i)

(ii)

(iii)



(iv)

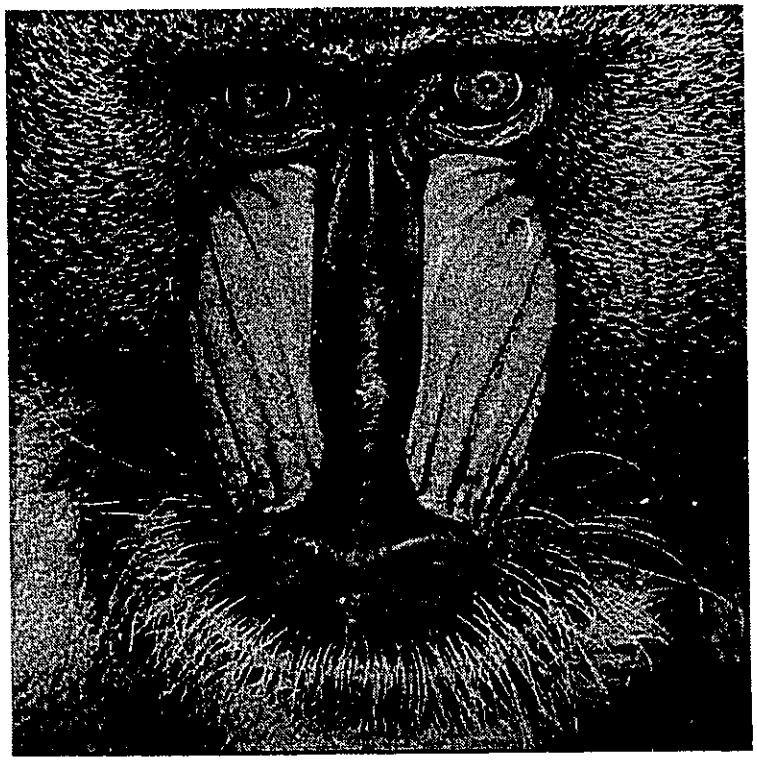


Figure 4.14(b): Baboon reconstructed at (i) 64 x 64, (ii) 128 x 128, (iii) 256 x 256, and (iv) 512 x 512 pixel resolutions using the SPS-WF technique.

of 2 x 2 pixels (SPS-MF). The proposed mean filter approach has been used to implement the JPEG hierarchical mode to achieve four levels of spatial scalability corresponding to image sizes of 64x64, 128x128, 256x256 and 512x512 pixels, respectively. The reconstructed images are shown in Fig.4.13. This approach has two drawbacks. First, a mean downsampling filter does not perform very well in extracting the features from the full size image and hence results in poor quality. Secondly, the difference images D_i may still have a significant amount of information (i.e.high entropy) leading to a poor compression performance. We now present a wavelet transform based technique for achieving spatial scalability (within the framework of hierarchical mode) which results in a substantial performance improvement over SPS-MF.

4.4 Enhancements to Hierarchical mode using Wavelet transform

We recall from section 2.6 that wavelet transform provides a multi-resolution / multi-frequency representation by decomposing an image into four sub-images. The LL sub-image is a reduced size version of the original image. The original image can be further reduced to smaller spatial resolutions by successive application of the wavelet transform on LL sub-images.

We propose to address the drawbacks of the mean downsampling filter by using wavelet transform for obtaining the image in lower spatial resolutions (SPS-WF). In other words, the JPEG coder obtains the images $I_0, I_1, ..., I_{N-2}$ by successive application of wavelet transform on the original image I_{N-1}. The various spatial levels can now be coded in a differential manner as specified in the hierarchical mode of JPEG. We have used biorthogonal wavelet bases [s] to perform a three level decomposition on the test images. For a full size image of 512 x 512 pixels, the lower spatial resolutions correspond to sizes of 256 x 256, 128 x 128, and 64 x 64 pixels, respectively. The images reconstructed with

this wavelet based technique are shown in Fig.4.14. It can be seen that at lower spatial levels, this technique provides a superior reconstructed image quality than those obtained with the mean filter approach. This confirms the excellent feature extraction capability of the wavelet transform. In addition, the full size images obtained with the wavelet scheme require a lower bit rate than the mean filter approach, (hence higher compression) for the same subjective quality of reconstructed images. Most importantly, the wavelet based technique does not require any modifications to existing JPEG decoders.

We note that in the hierarchical mode algorithm, the DCT coefficients of the difference images D_i can be quantized using different quantization tables which should be designed to extract maximum possible information from each image. Since no standard quantization tables have been reported in the literature for difference images, the sample quantization table suggested in the JPEG standard has been used at different quality factors in our simulations to quantize the difference images. In addition, the difference images at the lower spatial levels can be reconstructed using only the first few DCT coefficients. As a result, the difference images at the subsequent spatial levels will have a higher information content. The overall compression performance can be improved by using fewer DCT coefficients (which in turn are coarsely quantized) to reconstruct I_0 and smaller D_i's, while using all the 8 x 8 coefficients (quantized at high quality factors) for the remaining D_i's. This may result in some loss in the quality of smaller size images (R_i) without affecting the full resolution image. However, for applications which require high subjective quality at all spatial levels, such as simultaneous transmission of NTSC and HDTV, the smaller difference images should be reconstructed using all the coefficients (quantized at high quality factors) to preserve high frequency information. The loss in compression performance can be minimized by coarsely quantizing the difference images at higher spatial levels which have low information content.

It is difficult to objectively compare the performance of SPS-MF and SPS-WF at lower spatial resolutions, as the original image is available only at the highest spatial resolution.

Table 4.2: JPEG Hierarchical mode. Performance comparison of the two spatial scalability techniques: SPS-MF and SPS-WF

Lena

Size of the reconstructed image before upsampling to 512 x 512 pixels	SPS-MF		SPS-WF	
	bit rate (bits per pixel)	PSNR (dB)	bit rate (bits per pixel)	PSNR (dB)
64 x 64	0.039	20.43	0.039	23.76
128 x 128	0.146	23.8	0.161	27.46
256 x 256	0.403	28.46	0.424	31.15
512 x 512	0.897	33.72	0.846	33.67

Baboon

Size of the reconstructed image before upsampling to 512 x 512 pixels	SPS-MF		SPS-WF	
	bit rate (bits per pixel)	PSNR (dB)	bit rate (bits per pixel)	PSNR (dB)
64 x 64	0.034	19.07	0.036	19.72
128 x 128	0.162	20.08	0.196	20.94
256 x 256	0.579	22.08	0.673	23.19
512 x 512	1.649	27.48	1.64	27.51

Figure 4.15(a) -(i): Lena reconstructed at 64 x 64 pixel resolution using SPS-MF technique followed by upsampling to 512 x 512 pixels.

Figure 4.15(b) -(i): Baboon reconstructed at 64 x 64 pixel resolution using SPS-MF technique followed by upsampling to 512 x 512 pixels.

Figure 4.15(a) -(ii): Lena reconstructed at 128 x 128 pixel resolution using SPS-MF technique followed by upsampling to 512 x 512 pixels.

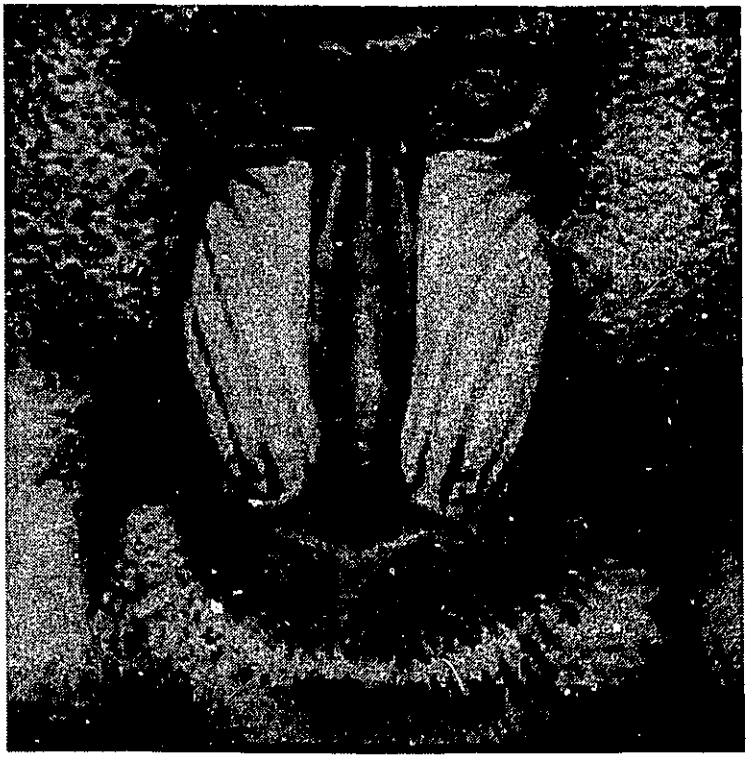


Figure 4.15(b) -(ii): Baboon reconstructed at 128 x 128 pixel resolution using SPS-MF technique followed by upsampling to 512 x 512 pixels.

Figure 4.15(a) -(iii): Lena reconstructed at 256 x 256 pixel resolution using SPS-MF technique followed by upsampling to 512 x 512 pixels.

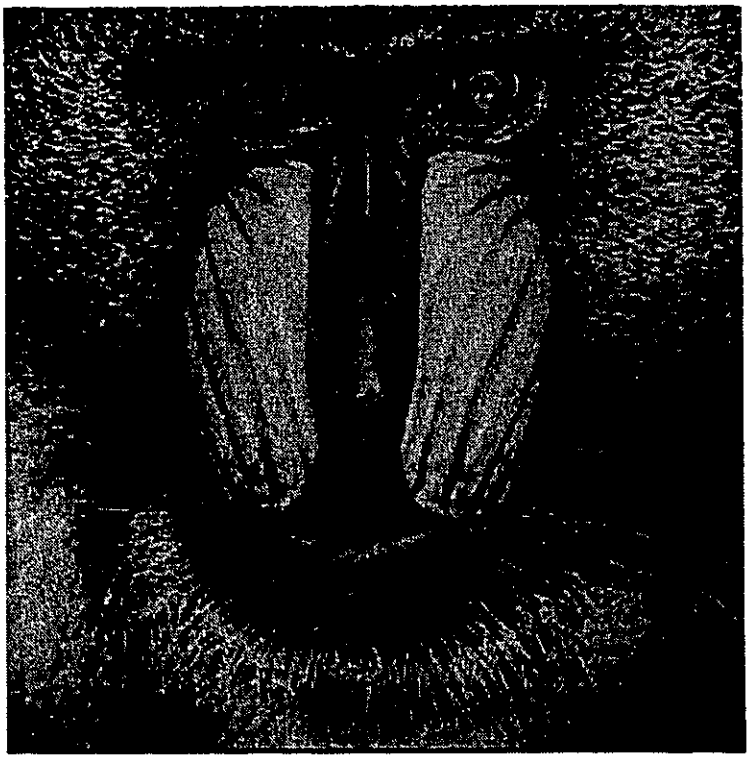


Figure 4.15(b) -(iii): Baboon reconstructed at 256 x 256 pixel resolution using SPS-MF technique followed by upsampling to 512 x 512 pixels.

Figure 4.16(a) -(i): Lena reconstructed at 64 x 64 pixel resolution using SPS-WF technique followed by upsampling to 512 x 512 pixels.

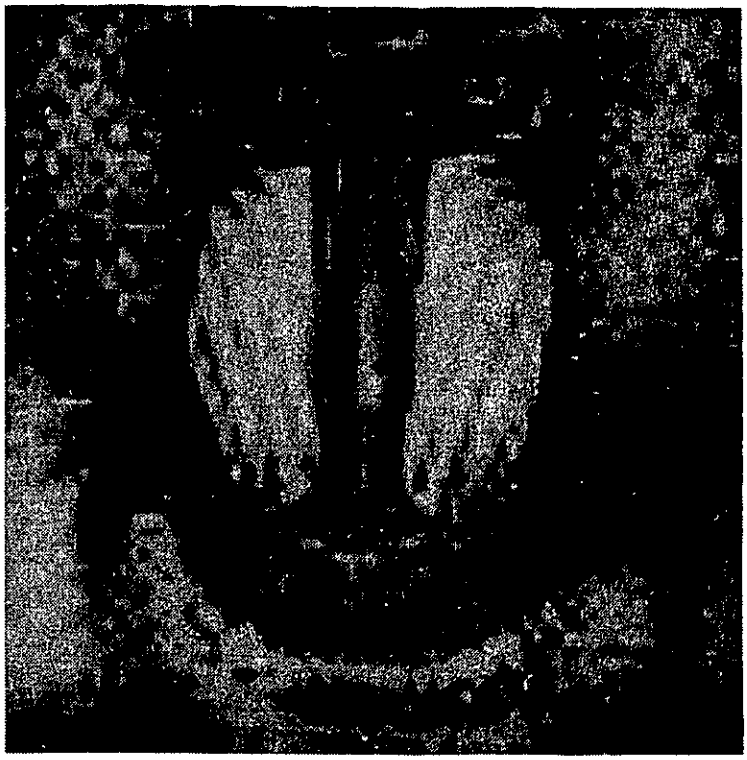


Figure 4.16(b) -(i): Baboon reconstructed at 64 x 64 pixel resolution using SPS-WF technique followed by upsampling to 512 x 512 pixels.

Figure 4.16(a) -(ii): Lena reconstructed at 128 x 128 pixel resolution using SPS-WF technique followed by upsampling to 512 x 512 pixels.

Figure 4.16(b) -(ii): Baboon reconstructed at 128 x 128 pixel resolution using SPS-WF technique followed by upsampling to 512 x 512 pixels.

Figure 4.16(a) -(iii): Lena reconstructed at 256 x 256 pixel resolution using SPS-WF technique followed by upsampling to 512 x 512 pixels.

Figure 4.16(b) -(iii): Baboon reconstructed at 256 x 256 pixel resolution using SPS-WF technique followed by upsampling to 512 x 512 pixels.

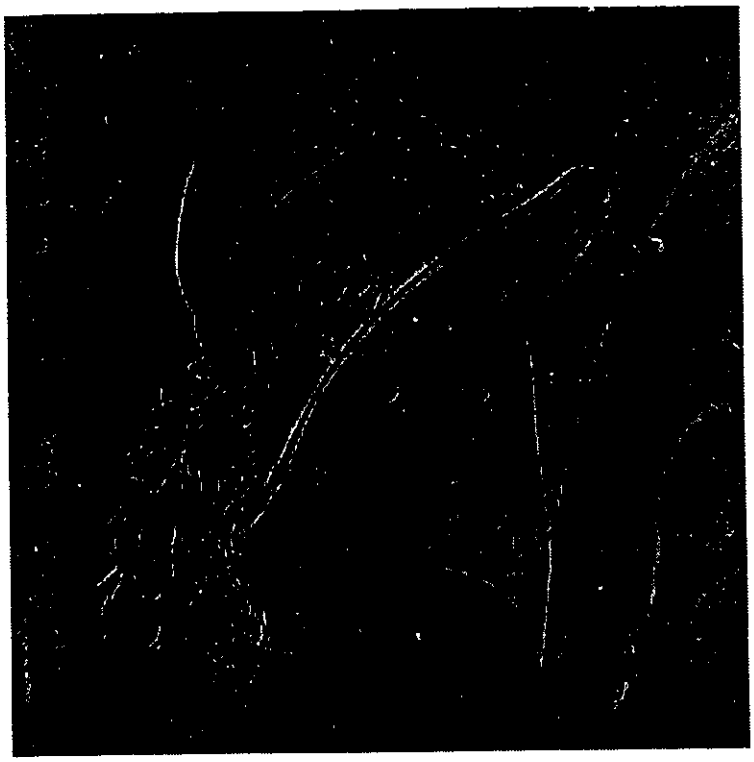


Figure 4.17(a): Difference between the original and SPS-MF reconstructed 256 x 256 pixels Lena image (upsampled to 512 x 512 pixels).

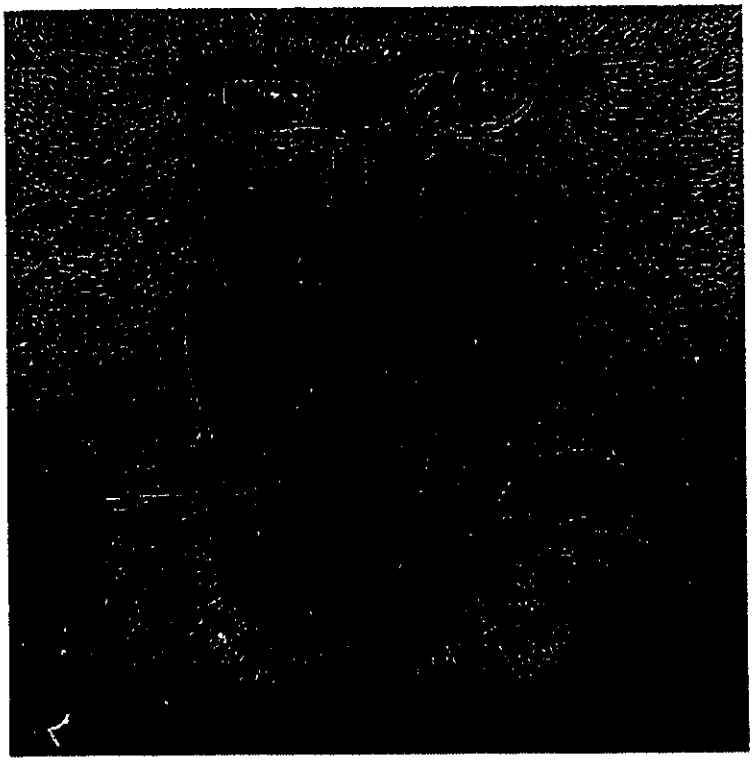


Figure 4.17(b): Difference between the original and SPS-MF reconstructed 256 x 256 pixels Baboon image (upsampled to 512 x 512 pixels).

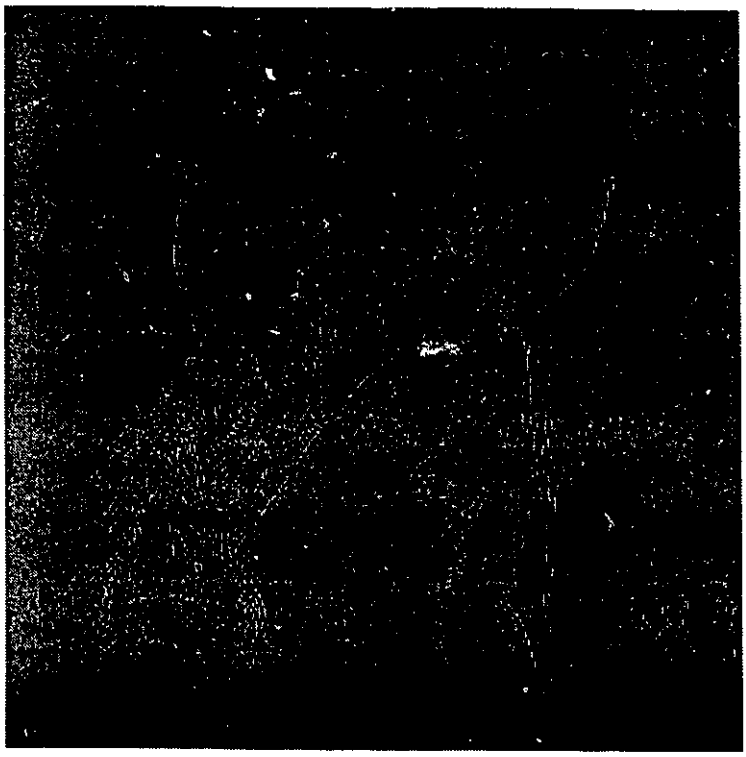


Figure 4.18(a): Difference between the original and SPS-WF reconstructed 256 x 256 pixels Lena image (upsampled to 512 x 512 pixels).

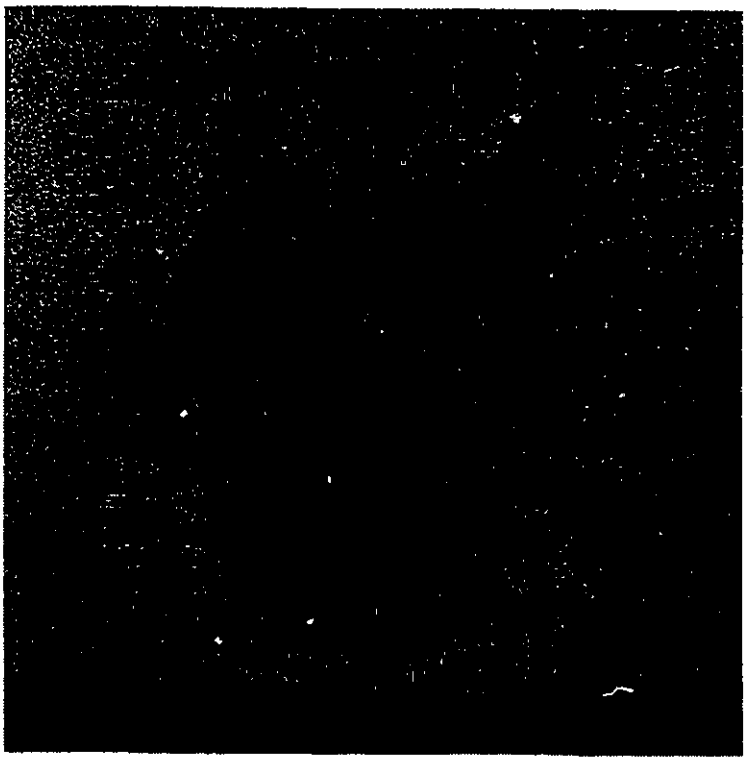


Figure 4.18(b): Difference between the original and SPS-WF reconstructed 256 x 256 pixels Baboon image (upsampled to 512 x 512 pixels).

this wavelet based technique are shown in Fig.4.14. It can be seen that at lower spatial levels, this technique provides a superior reconstructed image quality than those obtained with the mean filter approach. This confirms the excellent feature extraction capability of the wavelet transform. In addition, the full size images obtained with the wavelet scheme require a lower bit rate than the mean filter approach, (hence higher compression) for the same subjective quality of reconstructed images. Most importantly, the wavelet based technique does not require any modifications to existing JPEG decoders.

We note that in the hierarchical mode algorithm, the DCT coefficients of the difference images D_i can be quantized using different quantization tables which should be designed to extract maximum possible information from each image. Since no standard quantization tables have been reported in the literature for difference images, the sample quantization table suggested in the JPEG standard has been used at different quality factors in our simulations to quantize the difference images. In addition, the difference images at the lower spatial levels can be reconstructed using only the first few DCT coefficients. As a result, the difference images at the subsequent spatial levels will have a higher information content. The overall compression performance can be improved by using fewer DCT coefficients (which in turn are coarsely quantized) to reconstruct I_0 and smaller D_i's, while using all the 8 x 8 coefficients (quantized at high quality factors) for the remaining D_i's. This may result in some loss in the quality of smaller size images (R_i) without affecting the full resolution image. However, for applications which require high subjective quality at all spatial levels, such as simultaneous transmission of NTSC and HDTV, the smaller difference images should be reconstructed using all the coefficients (quantized at high quality factors) to preserve high frequency information. The loss in compression performance can be minimized by coarsely quantizing the difference images at higher spatial levels which have low information content.

It is difficult to objectively compare the performance of SPS-MF and SPS-WF at lower spatial resolutions, as the original image is available only at the highest spatial resolution.

Therefore for a proper comparison, the low resolution reconstructed images (R_j) have been upsampled to the full size using the mean interpolation filter. The upsampled images for the two techniques are shown in Figs.4.15 and 4.16, respectively and their PSNR values are tabulated in Table 4.2. The bit rates shown in Table 4.2 correspond to the spatial size of 512 x 512 pixels. It can be seen that the SPS-WF images consistently have a superior subjective quality and a higher PSNR compared to SPS-MF at comparable bit rates. Figs.4.17 and 4.18 show the the difference between the original test images and the 256 x 256 images (upsampled to 512 x 512 pixels) reconstructed using SPS-MF and SPS-WF, respectively. It can be seen that the difference images of SPS-MF contain important visual details, while the SPS-WF difference images contain very low information content. In other words, the SPS-WF images upsampled from the 256 x 256 pixels size are almost indistinguishable from the originals.

4.5 Scalable image transmission

Scalable image compression is not only crucial for image browsing applications but is also useful for image transmission applications. We recall from chapter 3 and sections 4.1 and 4.2 that scalable image compression is usually performed by partitioning the image data into several layers containing information of different visual importance and therefore different priority. This feature of scalable image compression is especially useful for image transmission over networks which employ dual-priority transmission mechanisms and ensure secure delivery of only the high priority data [61-65].

Recently, the advent of broadband-ISDN (B-ISDN) networks has facilitated the development of image/video based services and applications such as, video conferencing, video-on-demand, telepresence, multimedia communications, etc [66-74]. Asynchronous transmission mode (ATM) is gradually gaining acceptance as the preferred means of transmission over B-ISDN. This section investigates the performance of scalable image

transmission over ATM networks. We now briefly review some of the key features of the ATM.

4.5.1 Review of ATM Networks

ATM is a fast packet switching technique [68, 70, 71]. It was primarily developed to support broadband applications which have a variety of transmission rates and quality of service requirements. ATM uses a asynchronous time division multiplexing technique to multiplex the data streams from different information sources. Due to statistical multiplexing, several sources may share a high transmission rate link of capacity less than the sum of their peak arrival rates. This makes it particularly suitable for bursty traffic sources, such as video where a significant multiplexing gain can be achieved through statistical bandwidth assignment. Another advantage of ATM is that it provides a high degree of integration in both transmission and switching functions.

Although ATM is a packet switching (connection-less) technique, it also has some features of connection oriented techniques. In order to transfer data between two network nodes, the source data is first organized into packets (cells) of size 48 bytes each. A virtual connection is established between the source and destination nodes prior to the actual data transfer. Each cell has a 5 byte header which contains the routing information of the this connection in addition to data security and control information. All the cells follow the same route from the source to destination node.

One of the problems with ATM is that the cells have a variable transmission delay, although they are delivered in sequence [69]. This may pose a problem in delay sensitive services, such as those involving voice and video which do not tolerate high transmission delays. A more severe problem with ATM is that there may be a congestion in the network if the total bandwidth demand by different source nodes exceeds the available bandwidth. Effective congestion control strategies are required to guarantee a minimum quality of service to different types of applications supported by the network. Currently, ATM supports cells

with two levels of priority. In the event of a congestion on the network, the low priority cells are discarded to reduce the network load [73, 74]. However, this may result in an unacceptable picture quality for image transmission applications. In the case of scalable images, the high priority data (containing important image information) provide a graceful degradation in performance in the event of loss of low priority data.

4.5.2 Scalable image transmission over ATM networks

We recall from sections 4.1 and 4.2 that the progressive and hierarchical modes of JPEG provide scalable image compression by reorganizing the image data into multiple layers using different approaches. In this section, we investigate scalable image transmission over ATM networks. The baseline (BSA) mode has been used for comparative evaluation of the performance of progressive and hierarchical modes.

The baseline mode of JPEG (refer section 2.4) can only be used to reconstruct the image in its original size at a specific image quality (SNR resolution). In order to packetize BSA compressed images, the bitstreams of the image blocks are sequentially packetized into ATM cells. Since BSA does not partition the image into high priority and low priority data, all the cells are of equal importance (i.e. equal priority) and are therefore categorized as low priority cells.

For transmitting images compressed using the spectral selection (SNR-SS) technique of progressive mode, the low frequency bands are packetized into high priority cells, while the less important high frequency bands are transported in the low priority cells. Such a packetization scheme ensures that in the event of a network congestion and/or cell losses, the visually important low frequency coefficients are not lost during transmission resulting in a graceful degradation in the quality of the received image. Similarly, for images compressed using the successive approximation (SNR-SA) technique, the most significant bit planes are packetized into high priority cells while the least significant bit planes are

transported in low priority cells. We note that the two progressive mode techniques (SS and SA) can be combined together. By suitably packetizing the resulting bitstream into high and low priority cells, it may be possible to minimize degradation in the image quality at a given cell loss rate.

In the case of hierarchical mode, the bitstreams of the smallest size image I_0 (refer Fig.4.4) and the difference images D_i corresponding to the smaller spatial resolutions contain the most significant visual information and are therefore packetized in the high priority cells. Portions of the visually important low frequency coefficients of D_i, corresponding to higher spatial resolutions, may also be packetized in the high priority cells based on the availability of bandwidth. The bits corresponding to the remaining coefficients of the difference images are packetized in the low priority cells. The low resolution receivers display the image after decoding the bitstream corresponding to the smallest size image. The high resolution receivers first decode the bitstreams of the smallest image and then upsample it by a factor of two in both directions using a mean interpolation filter. The upsampled image is further enhanced by adding the corresponding difference image contained in the high and low priority cells. This process is repeated till the image size matches the receiver resolution. In the event of cell losses, the image quality of the low resolution receivers (requiring only high priority cells) would be unaffected. However, due to the loss of some of the enhancement information contained in the low priority cells, the image quality of the high resolution receivers will be gracefully degraded depending on the cell loss rate.

4.5.3 Simulations and discussions

Simulations have been performed using the "Lena" and "Baboon" test images (Fig.4.5) of size 512x512 pixels quantized to 8 bits/pixel. Three sets of simulations have been performed in which the test image is compressed using the BSA, SNR-SS and SPS-WF techniques, respectively. We have chosen the spectral selection technique for investigating

the performance of the progressive mode as it provides a higher compression compared to the successive approximation technique. In the case of hierarchical mode, we have selected the SPS-WF technique as it provides a superior subjective image quality compared to SPS-MF (refer section 4.4).

The compressed bitstreams of the three techniques have been packetized into ATM cells, each of size 48 bytes. In the event of a network congestion, all the high priority cells are transmitted securely to the receiver, whereas some of the low priority cells may have to be discarded to reduce the network load. Two sets of experiments have been performed for each technique to study the effect of the loss of low priority cells at low and high loss rates, respectively. We have used BSA for a comparative evaluation of the performance of the SNR-SS and SPS-WF techniques. The objective image quality has been measured using the Peak Signal to Noise Ratio (PSNR).

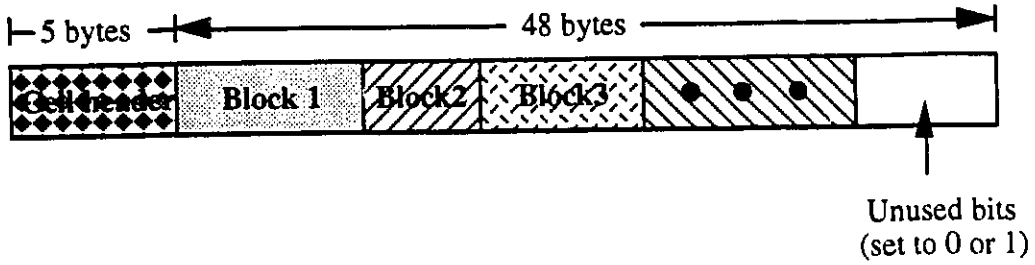


Figure 4.19: Packetization of a BSA compressed image into ATM cells of size 48 bytes each.

In the case of BSA, the compressed bitstream has been packetized without splitting any image block so that all the bits of a block belong to the same cell. In other words, any unused bits of a cell are either set to 0 or to 1 as shown in Fig.4.19. A total of 462 and 947 cells are required for the "Lena" and "Baboon" images, respectively. The "Baboon" image requires almost twice as many cells as "Lena" because it contains a large amount of high frequency information. We recall that all the cells are of low priority. To study the

performance of BSA in the event of network congestion, we have simulated cell losses by randomly discarding some of the cells. For the "Lena" image, a loss of 10 (Expt.I) and 40 (Expt.II) cells has been simulated in two sets of experiments, corresponding to a loss of 2.2% and 8.7%, respectively. Similarly, for the "Baboon" image, a loss of 30 (Expt.I) and 70 (Expt.II) cells has been simulated in two sets of experiments, corresponding to a loss of 3.2% and 7.4%, respectively. The simulation results for the two images have been tabulated in Tables 4.3 and 4.4, respectively. It can be observed from the reconstructed images (shown in Figs.4.20 and 4.21) that the BSA results in a significant performance degradation in the event of cell losses. The reconstructed images have a poor image quality as losing a cell implies losing all the information about the image blocks it contains. In addition, the quality of subsequent blocks also suffers due to predictive coding of DC coefficients used in the BSA. Zhu *et.al.*[75] have proposed block interleaving and coefficient segmentation techniques to minimize the effects of cell losses. Block interleaving requires modifications to the BSA mode and is useful only because BSA does not partition the image into high and low priority data. On the other hand, the progressive and hierarchical modes of JPEG compress the image by first decomposing it into data sets of different priorities and therefore do not require block interleaving. Coefficient segmentation is a special case implementation of the SNR-SS technique.

In SNR-SS, the DCT coefficients have been partitioned into two bands: the first containing the DC and the first 2 AC coefficients, and the second containing all the remaining AC coefficients. For the "Lena" image, the two bands have been packetized into 153 high priority and 306 low priority cells, respectively. On the other hand, "Baboon" image requires 163 high priority and 765 low priority cells, respectively. In order to have a meaningful comparison, the total number of cells used for image reconstruction should be same in both the BSA and SNR-SS in the event of cell losses. We have therefore simulated (random) losses of 7 (Expt.I) and 37 (Expt.II) low priority cells for the "Lena" image in two sets of experiments, respectively. Similarly for the "Baboon" image, a loss of 11

Table 4.3: Performance comparison of the BSA, SNR-SS, and SPS-WF techniques for the transmission of Lena image over ATM networks.

JPEG mode	Total number of cells required		Number of Low priority cells lost	Cells used for reconstruction		PSNR (dB) of the full size/full quality reconstructed image
	High priority	Low priority		High priority	Low priority	
BSA						
No losses	-	462	0	-	462	33.94
Expt. I	-	462	10	-	452	21.18
Expt. II	-	462	40	-	422	16.1
SNR-SS						
No losses	153	306	0	153	306	33.94
Expt. I	153	306	7	153	299	33.5
Expt. II	153	306	37	153	269	31.82
SPS-WF						
No losses	135	389	0	135	389	33.82
Expt. I	135	389	72	135	317	31.91
Expt. II	135	389	102	135	287	31.42

Table 4.4: Performance comparison of the BSA, SNR-SS and SPS-WF techniques for the transmission of Baboon image over ATM networks.

JPEG mode	Total number of cells required		Number of Low priority cells lost	Cells used for reconstruction		PSNR (dB) of the full size/full quality reconstructed image
	High priority	Low priority		High priority	Low priority	
BSA						
No losses	-	947	0	-	947	27.53
Expt. I	-	947	30	-	917	19.7
Expt. II	-	947	70	-	877	17.81
SNR-SS						
No losses	163	765	0	163	765	27.53
Expt. I	163	765	11	163	754	27.28
Expt. II	163	765	51	163	714	26.48
SPS-WF						
No losses	188	882	0	188	882	27.51
Expt. I	188	882	153	188	729	25.35
Expt. II	188	882	193	188	689	24.95

Figure 4.20(a): BSA encoded Lena image reconstructed using only 452 low priority ATM cells after a loss 10 cells (Expt. I).

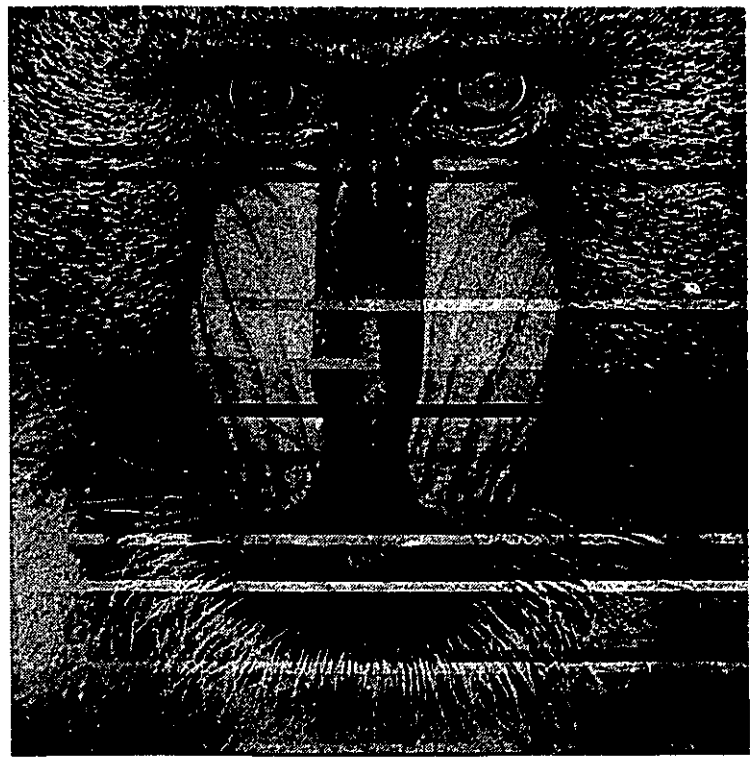


Figure 4.20(b): BSA encoded Baboon image reconstructed using only 917 low priority ATM cells after a loss of 30 cells (Expt. I).

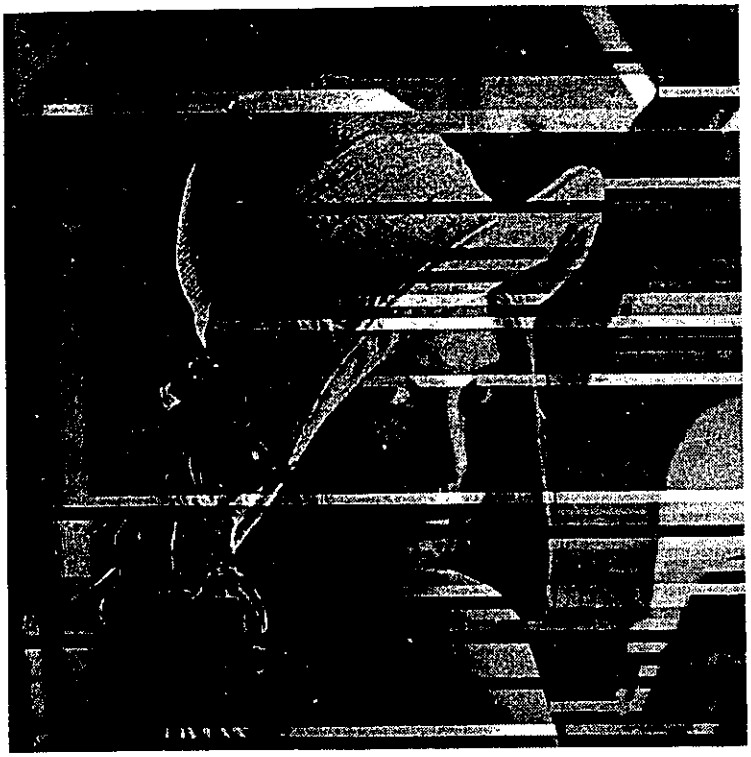


Figure 4.21(a): BSA encoded Lena image reconstructed using only 422 low priority ATM cells after a loss 40 cells (Expt. II).

Figure 4.21(b): BSA encoded Baboon image reconstructed using only 877 low priority ATM cells after a loss of 70 cells (Expt. II).

Figure 4.22(a): SNR-SS encoded Lena image reconstructed using only 153 high priority and 299 low priority ATM cells after a loss of 7 cells (Expt. I).

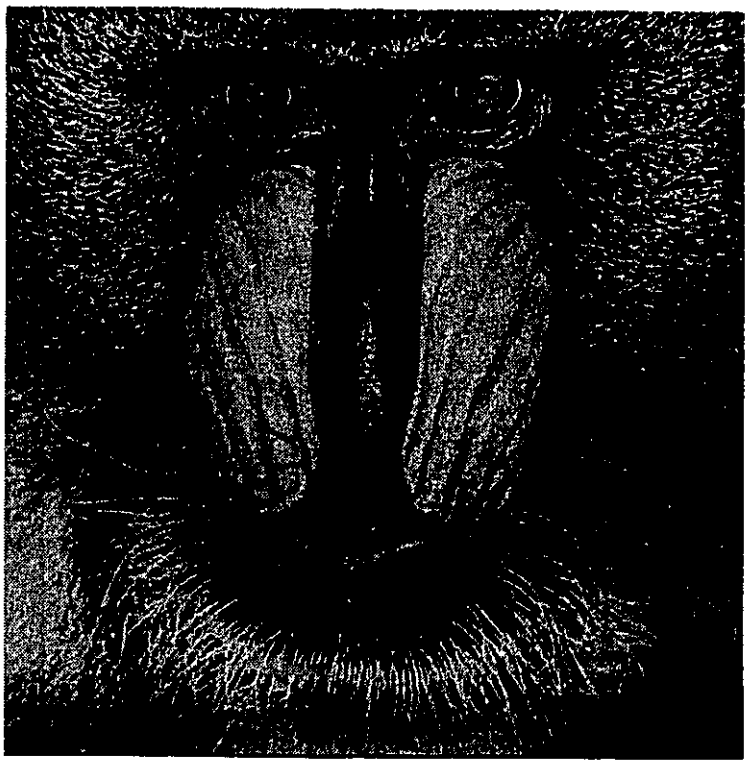


Figure 4.22(b): SNR-SS encoded Baboon image reconstructed using only 163 high priority and 754 low priority ATM cells after a loss of 11 cells (Expt. I).

Figure 4.23(a): SNR-SS encoded Lena image reconstructed using only 153 high priority and 269 low priority ATM cells after a loss of 37 cells (Expt. II).

Figure 4.23(b): SNR-SS encoded Baboon image reconstructed using only 163 high priority and 714 low priority ATM cells after a loss of 51 cells (Expt. II).

Figure 4.24(a): SPS-WF encoded Lena image reconstructed using only 135 high priority and 317 low priority ATM cells after a loss of 72 cells (Expt. I).

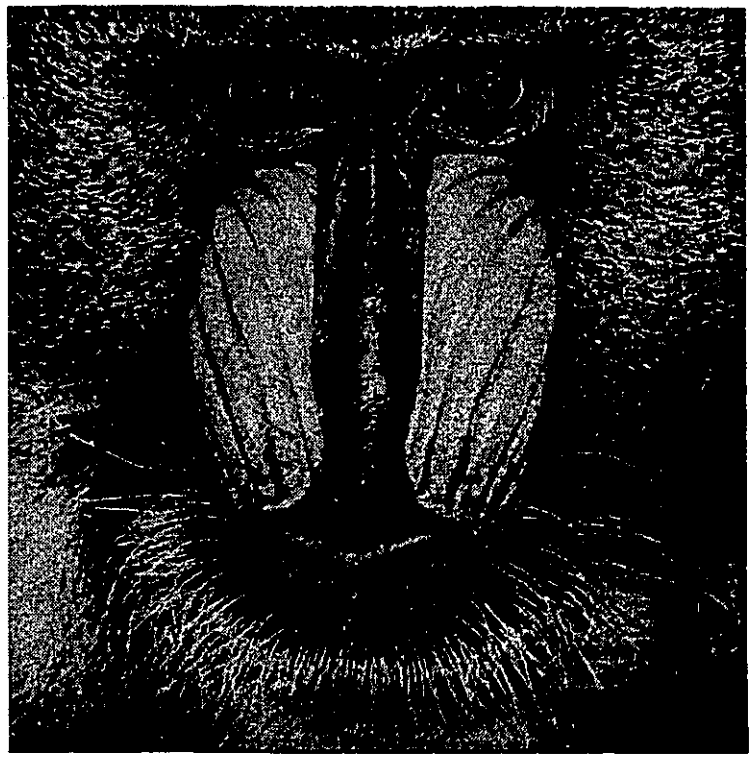


Figure 4.24(b): SPS-WF encoded Baboon image reconstructed using only 188 high priority and 729 low priority ATM cells after a loss of 153 cells (Expt. I).

Figure 4.25(a): SPS-WF encoded Lena image reconstructed using only 135 high priority and 287 low priority ATM cells after a loss of 102 cells (Expt. II).

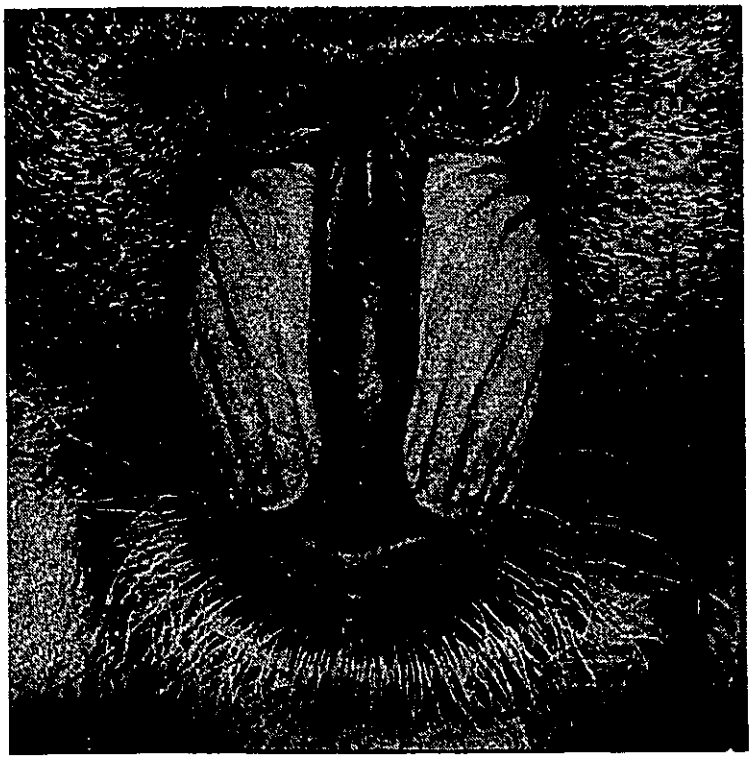


Figure 4.25(b): SPS-WF encoded Baboon image reconstructed using only 188 high priority and 689 low priority ATM cells after a loss of 193 cells (Expt. II).

(Expt.I) and 51 (Expt.II) low priority cells has been simulated in the two experiments, respectively. Figs.4.22 and 4.23 show the corresponding reconstructed images. It can be seen that the spectral selection technique results in a substantial performance improvement over the BSA at low as well as high loss rates. The distortions due to cell losses are confined to few image blocks. In addition, SNR-SS also provides the feature of reconstructing images in different SNR resolutions. Interestingly, SNR-SS requires a total of 153+306=459 cells in comparison to 462 cells of BSA for the "Lena" image. Similarly, for the "Baboon" image SNR-SS requires a total of 163+765= 928 cells as comapred to 947 cells for BSA. This is due to efficient utilization of cells by SNR-SS.

We have implemented the hierarchical mode for two spatial levels corresponding to the sizes of 128x128 and 512x512 pixels, respectively. For the "Lena" image, the (compressed) bitstreams of the 128x128 image and the 512x512 pixels difference image are packetized into 135 high priority and 389 low priority cells, respectively. On the other hand, "Baboon" image requires 188 high priority and 882 low priority cells. We recall from section 2 that the full size image is reconstructed by first upsampling the reconstructed 128x128 pixel image which is then further improved by adding the difference image information from the low priority cells. Once again for fair comparison, we have simulated (random) losses of 72 (Expt.I) and 102 (Expt.II) low priority cells for the "Lena" image (in two sets of experiments, respectively) so that the total number of cells used for image reconstruction are same as before. For the "Baboon" image, a loss of 153 (Expt.I) and 193 (Expt.II) low priority cells has been simulated. The corresponding reconstructed images are shown in Figs.4.24 and 4.25, respectively. It can be observed that in the event of cell losses, SNR-SS technique results in blocking artifacts due to missing high information in some image blocks. However, the loss in image quality may not be perceptible at low cell loss rates. On the other hand, SPS-WF evenly distributes the distortions due to cell losses over large image regions. Therefore at low cell loss rates, SNR-SS provides a superior

subjective quality. On the other hand, SPS-WF results in a more graceful performance degradation at high cell loss rates.

4.6 Summary

In this chapter, we presented scalable image compression algorithms within the framework of JPEG standard. The progressive and hierachical modes of JPEG provide SNR and spatial scalabilities, respectively. Among the two techniques proposed by the standard for progressive mode, the spectral selection technique provides a superior coding performance compared to successive approximation. The hierarchical mode uses a pyramidal structure for encoding the different spatial resolutions of the image. However, the standard does not specify downsampling filters for obtaining the original image in smaller sizes. We used a simple mean downsampling filter to reduce the image to half the original size in both the directions. However, this filter does not perform well in extracting the image features resulting in poor subjective quality. The performance of the hiearchical mode was then enhanced by using wavelet transform for downsampling the image. Wavelet transform provides high quality lower spatial resolutions of the image. Finally, we investigated the performance of transmitting scalable images over dual priority transmission medium such as ATM networks. In this section, we established that the progresssive and hierarchical modes of JPEG provide a graceful degradation in performance compared to the baseline mode, in the event of cell losses. This is due to reorganization of the image information into high and low priority data.

In conclusion, we note that although JPEG provides an efficient technique (spectral selection) for achieving SNR scalability, however its hierachical mode requires an overhead in terms of the bit rate for providing spatial scalability. This overhead increases with the number of spatial resolutions. In the next chapter, we present novel techniques based on

vector quantization and wavelet transform which provide a higher coding performance than the JPEG based techniques.

Chapter 5

Novel techniques for Scalable Image Compression

In chapter 4, we investigated the performance of scalable image compression techniques within the framework of JPEG standard. The progressive mode of JPEG provides an efficient technique for achieving SNR scalability, namely spectral selection. However, the hierarchical mode of JPEG does not provide a good coding performance at low bit rates. In this chapter, we propose a novel technique for scalable image compression, which provides a superior coding performance at low bit rates. This technique is based on the wavelet transform and vector quantization (WVQ).

We recall from section 2.6 that wavelet transform provides a multi-resolution / multi-frequency representation by decomposing an image into four sub-images. The LL sub-image is a reduced size version of the original image. The original image can be further reduced to smaller spatial resolutions by successive application of wavelet transform on the LL sub-images. Several techniques have been discussed in section 2.6 to compress a wavelet transformed image. However, they do not exploit the correlation between the wavelet sub-images resulting in lower compression. The proposed WVQ algorithm provides significantly higher compression by efficiently exploiting the correlation between the wavelet sub-images. The details of WVQ follows.

5.1 Wavelet based Vector Quantization

In the WVQ algorithm, the input image is first decomposed into a three level pyramid structure (corresponding to 10 sub-images) by successive application of wavelet transform as shown in Fig.2.15. The wavelet sub-images are then compressed using vector quantization (VQ). For this purpose, 64-D vectors are formed by combining the corresponding blocks in the 10 sub-images as shown in Fig.5.1. Here, a block of size 2^{3-m} (m= 1, 2, 3) is used for each orientation sub-image (namely, horizontal, vertical and diagonal) at the m^{th} level. This ensures that each sub-image contains the same number of blocks. A 64-D codebook is then generated from a large set of training vectors (drawn from a number of test images) using the LBG algorithm. Each vector in the wavelet transformed image is represented by its best-match codeword in this codebook and the corresponding label is transmitted. At the receiver, the input wavelet transformed image is reconstructed using the labels by a simple table lookup operation into the codebook. An inverse wavelet transform is then applied on this image to obtain a (lossy) reconstruction of the original image.

The computational complexity of the LBG algorithm for codebook generation increases exponentially with the dimensionality of the vector making real-time compression impractical. Recently, Xiping *et al* [37] have proposed a non-linear interpolative vector quantization (NIVQ) based technique to reduce the computational complexity of codebook generation in the WVQ algorithm. In their technique, a 16-D feature vector is generated from each 64-D vector in the wavelet transformed image. Due to wavelet decomposition of the image, the features of each 64-D vector are efficiently captured in the first 16 elements (corresponding to the lower resolution sub-images) which are therefore used as the feature vector. A 16-D codebook is generated from the feature vectors using the LBG algorithm. This codebook partitions the feature vectors into 16-D subspaces. We note that there is a 1-1 mapping between the feature vectors and the full size (64-D) image vectors. A 64-D

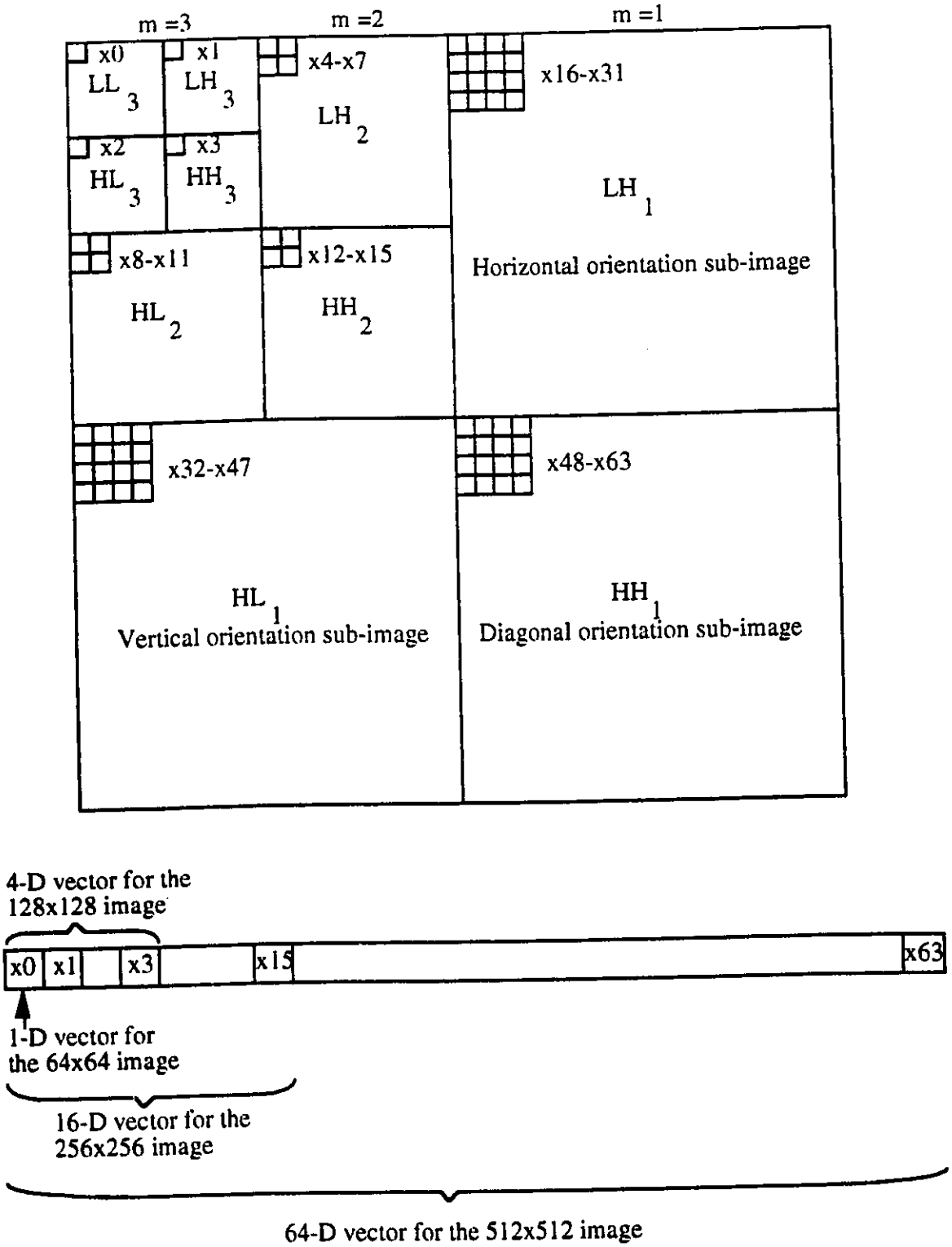


Figure 5.1 Vector formation for applying the SVQ scheme to a three level wavelet decomposed image.

codebook is generated by clustering the full size vectors in the same groups in which their feature vectors were partitioned. In other words, each 64-D codeword is computed by calculating the mean of the group of (64-D) vectors it represents. The 64-D codebook is thus generated by projecting the 16-D subspaces into the 64-D vector space as shown in Fig.5.2.

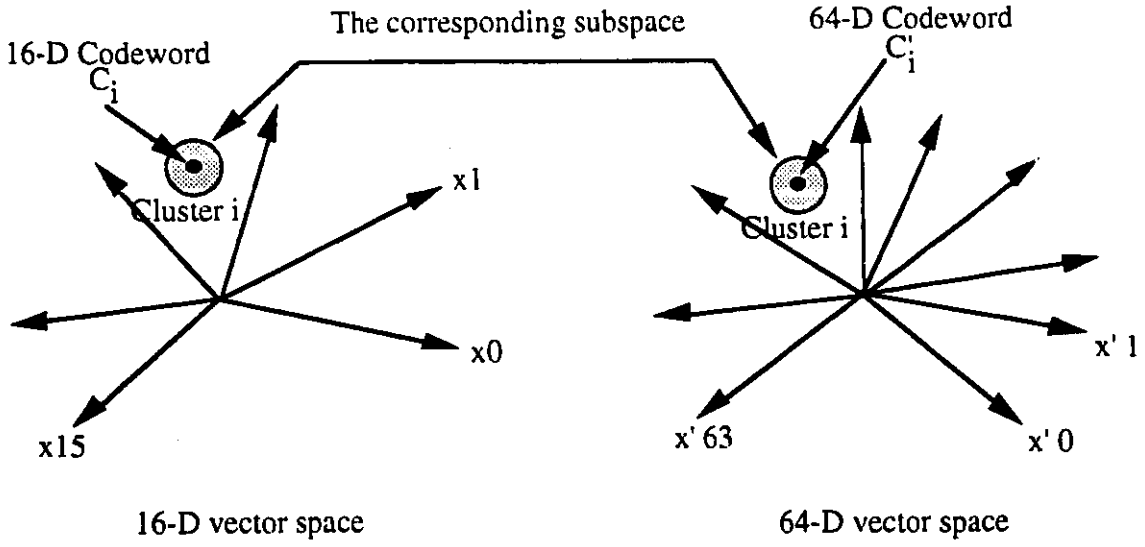


Figure 5.2: Projection of subspaces from 16-D to 64-D vector space.

For universal vector quantization, this codebook is assumed to be present at the decoder. The encoder extracts the 16-D feature vectors from the input 64-D image vectors which are vector quantized using the 16-D codebook and the corresponding labels are transmitted. At the receiver, the input (wavelet transformed) image is reconstructed by using these labels as an index for a table lookup operation from the 64-D codebook (Fig.5.3). WVQ algorithm when implemented using the NIVQ technique provides an excellent coding performance at a significantly lower computational complexity. Although WVQ is an efficient technique, its label bitstream is not scalable. In other words, it is not possible to reconstruct the image in a lower SNR and/or spatial resolution by using only the first few bits of the VQ labels. The reason is that vector quantization is inherently not scalable, although, wavelet transform provides a scalable image representation. We propose a novel technique for scalable image

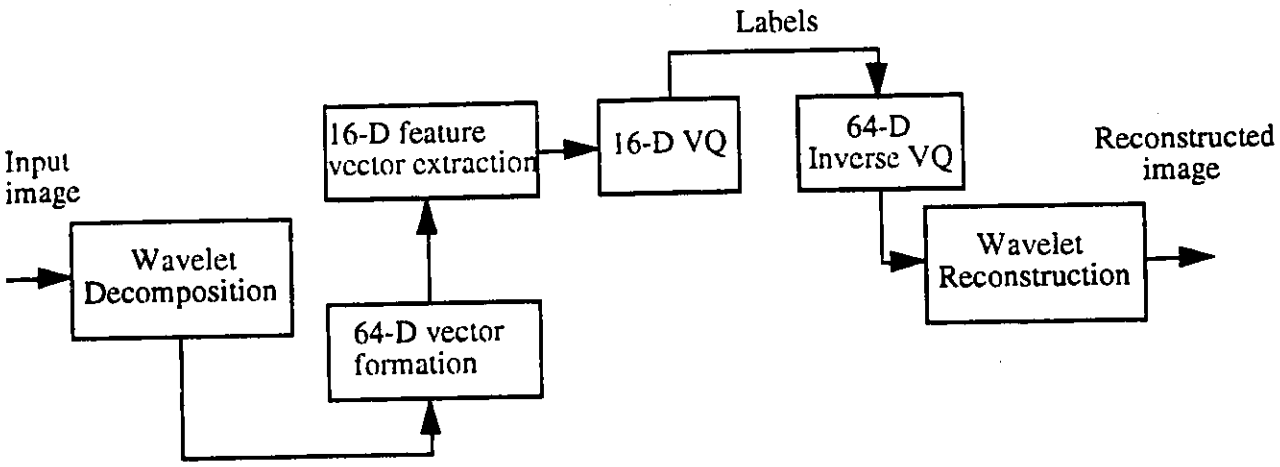


Figure 5.3: Application of NIVQ technique to the WVQ algorithm.

compression using vector quantization. This technique will be later applied to the WVQ algorithm to obtain high quality scalable images with high compression.

5.2 Vector quantization for scalable image compression

Vector quantization (VQ) has been known to provide a good coding performance at low bit rates (refer section 2.5). We recall from section 3.2.1 that Residual Error Vector Quantization (REVQ) achieves SNR scalability by iterative vector quantization of the residual error images. REVQ requires multiple codebooks, one for each residual error image. Since there is little correlation between the residual error images, there is no correlation between their labels. In other words, the label bitstream is not scalable. Unlike REVQ, tree-structured VQ (TSVQ) provides scalable labels for achieving SNR scalability. TSVQ uses multiple codebooks of different sizes which are organized in a binary tree structure (refer section 3.2.1). Fig.5.4 illustrates an example of the TSVQ technique. A low quality version of the image is first reconstructed by decoding only the first 4 MSBs of the VQ labels which serve as an index to a codebook of size 16. The picture quality of the image

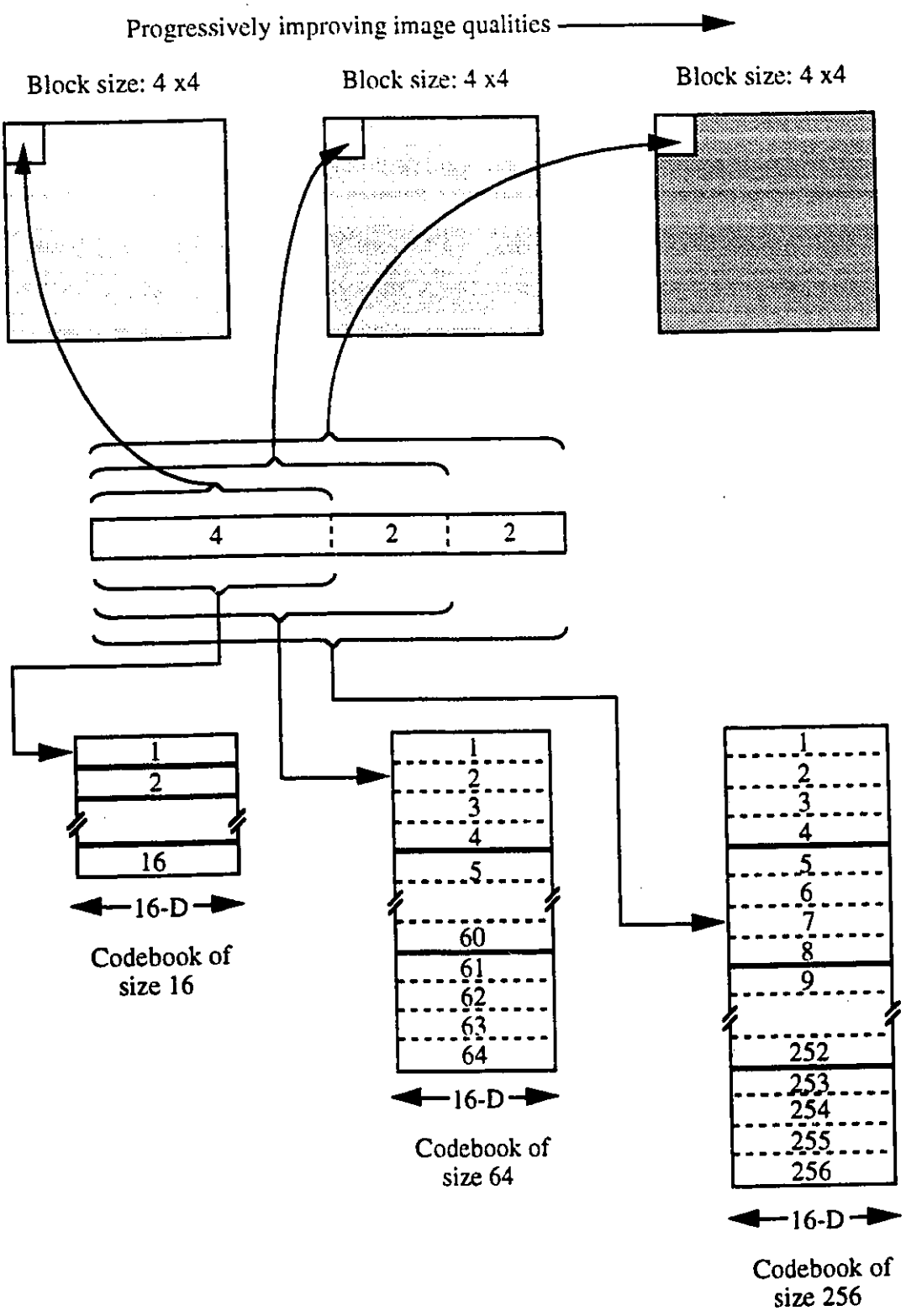


Figure 5.4: Gradual decoding of scalable labels in TSVQ technique for image reconstruction in progressively improving qualities.

is then improved by using an additional 2 MSBs for a table lookup operation into a (larger) codebook of size 64. This permits reconstruction of the image vectors (blocks) with a greater accuracy. The full quality image is obtained by using all the 8 label bits as an address to a (still larger) codebook of size 256. TSVQ may result in a sub-optimal compression performance for the full quality image. This is due to generation of codebooks in a tree structured fashion, where each children codeword (at level $i+1$) is trained using only those image vectors which were represented by its parent codeword (at level i). We note that neither REVQ, nor TSVQ can be used for providing spatial scalability.

In the next section, we propose a novel technique based on VQ for achieving spatial scalability. This technique provides scalable labels such that a smaller size image can be reconstructed by decoding only few MSBs of each label. The lower resolution image can be further enhanced in size by progressively decoding the remaining label bits. The details of the technique follows.

5.2.1 Spatial scalability using vector quantization

A VQ based technique for achieving spatial scalability should not only ensure partial decodability of VQ labels for reconstructing the images at smaller sizes, but should also provide a good quality image at the full spatial resolution. Fig.5.5 illustrates these features for a full size image (I_1) of 512 x 512 pixels. We propose a new technique, called SVQ, which provides these features. SVQ requires separate codebooks, one for each spatial resolution. For example, in Fig.5.5, the image is first reconstructed at a resolution of 128 x 128 pixels (I_3) by decoding only the first 5 MSBs of the VQ labels which serve as an index for a table lookup operation in a small size codebook, U_3. The spatial resolution of image I_3 can be further enhanced by using an additional 2 MSBs of the labels for a table lookup operation in codebook U_2 to reconstruct the image at a size of 256 x 256 pixels (I_2). The full size image (I_1) of resolution 512 x 512 pixels is obtained by using all the 8 bits of each label

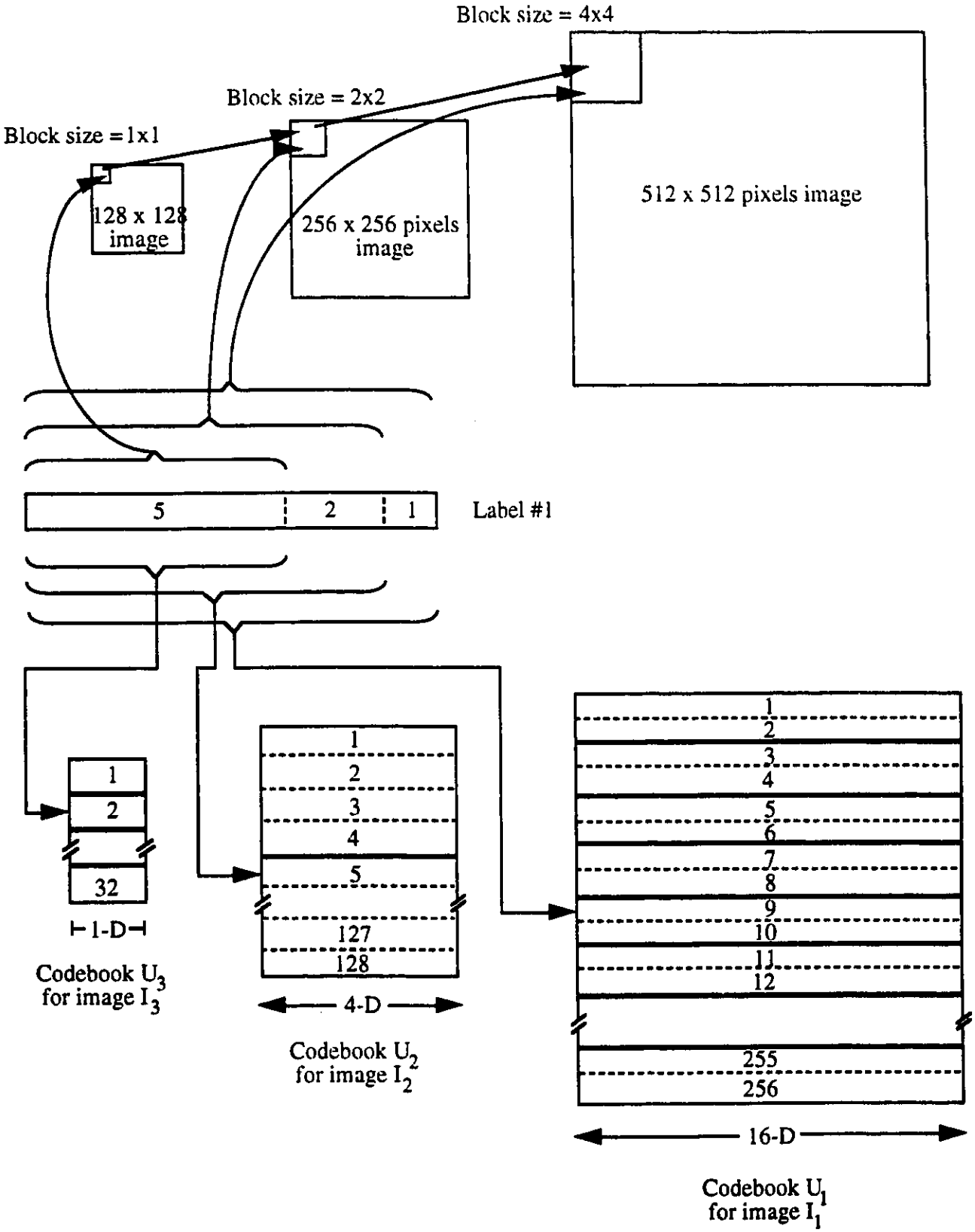


Figure 5.5: Features expected from the SVQ technique.

as an address to codebook U_1. We note that U_3, U_2 and U_1 are codebooks of dimensions, 1, 4 and 16, respectively. Universal codebooks are generally used for all the resolutions to avoid the overhead of codebook transmission which results in substantial bit rate reduction. To begin with, a universal codebook U_1 of size N_1 (N_1 is usually a power of 2 e.g. 256, 512, etc.) is generated for vector quantizing the original image I_1 having a resolution of M_1 x M_1 pixels. We note that $N_1 = 256$ and $M_1 = 512$ in Fig.5.5. The codebook U_1 is generated from a large number of test images using the LBG algorithm with binary splitting technique. This ensures a reconstructed image of good quality at the highest resolution, unlike the TSVQ scheme which may result in a sub-optimal performance due to tree structured codebook generation. In order to generate U_1, the test images are partitioned into non-overlapping blocks of size B_1 x B_1 pixels ($B_1 = 4$ in Fig.5.5) which are then reorganized as vectors of dimension B_1^2 (refer section 2.5). The binary splitting technique generates the codebook in multiple stages. First, a codebook containing a single codeword is obtained either by random initialization or by computation of the mean of all the test image vectors. At each stage, the size of the codebook is doubled by splitting each codeword into two children codewords. This is achieved by introducing small distortions (perturbations) in the parent codeword in opposite directions resulting in two children codewords. The new codebook is refined by using the LBG algorithm. The entire procedure consisting of codebook splitting and refining is repeated until the codebook has the required number of codewords. The binary splitting technique for codebook generation results in reasonably similar neighboring codewords (i.e. codewords with adjacent labels). In other words, these vectors are likely to be represented by the same parent codeword in the previous stage codebook. Once the codebook U_1 is generated, each image vector in I_1 is vector quantized by searching the codebook to locate the best-match codeword and the corresponding label is transmitted. Although the entire label bitstream is now available at the decoder, it is not possible to reconstruct the image in smaller sizes until the corresponding codebooks are also generated. The codebooks for the lower spatial resolutions are generated by merging the

codewords for full size image through a special process which is detailed in the next section. We note that good quality codebooks, U_i (i >1), can be obtained for the lower resolution images, if the codebook U_1 is generated using the binary splitting technique.

Codebook generation for lower resolution images

We now describe the procedure for generating the codebook U_2 to reconstruct the image at the next lower spatial resolution, I_2. We note that the procedure is similar for codebook generation at other spatial resolutions (ex.I_3). In the example in Fig.5.5, the codebooks U_2 is generated by first downsampling the test images (by a factor of two in both directions) to obtain the lower resolution images of size 256 x 256 pixels. We note that I_2 has to be reconstructed using (the MSBs of) each label in the bitstream, therefore it is mandatory that I_2 consist of the same number of blocks (vectors) as I_1. This implies that vectors in I_2 should be one fourth the dimension of vectors in I_1. The downsampled test images are therefore partitioned into blocks of size 2 x 2 pixels, each of which is reorganized as a 4-D vector for generating the codebook U_2. On the other hand, if the image I_1 consisted of blocks of size 8 x 8 pixels (i.e. 64-D vectors), then I_2 would consist of blocks of size 4 x 4 pixels (i.e. 16-D vectors) and consequently codebook U_2 would have 16-D codewords.
Let the size of codebook U_2 be denoted by N_2 (N_2 is also a power of 2 like N_1). The codebook U_2 should have fewer codewords than codebook U_1 (i.e. $N_2 < N_1$) since the image I_2 has to be reconstructed by decoding only the first few MSBs of each label. For example, a codebook of size (N_2 =) 2^7 = 128 codewords is required if only the first 7 MSBs of the labels are used as shown in Fig.5.5. We recall that the VQ labels should be partially decodable to reconstruct the image I_2. In other words, the first 7 MSBs of the labels should provide the address of codewords in U_2 to reconstruct the image vectors at a resolution of 2 x 2 pixels. In addition, it should be possible to enhance the resolution of each image vector to 4 x 4 pixels using all the 8 bits of the labels for a table lookup operation

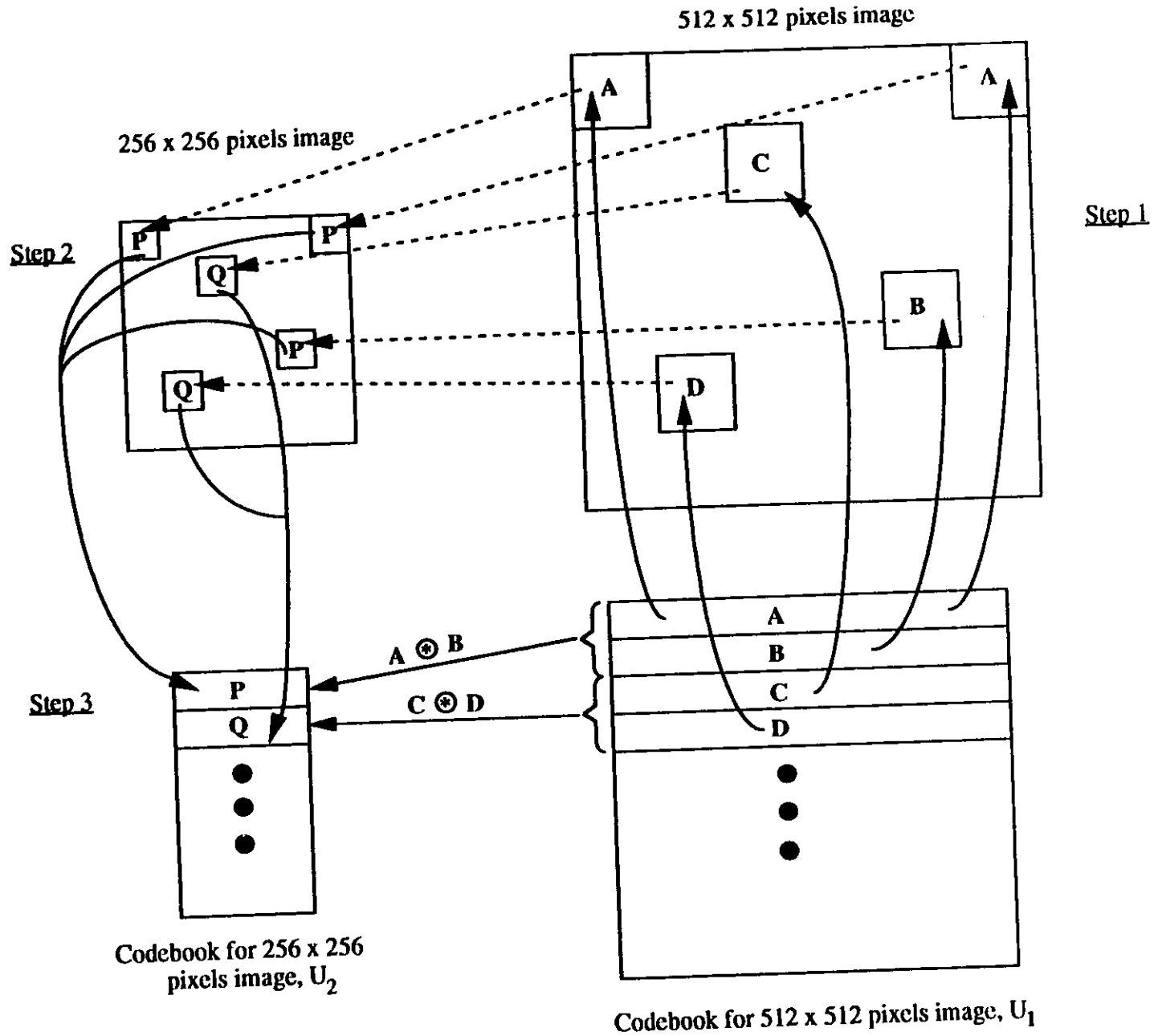


Figure 5.6: Formation of a codebook for lower size image in SVQ technique by merging the codewords of full size image. The merging operation is indicated by ⊛.

into the codebook U_1. In order to ensure that the SVQ technique provides this feature, the codebook U_2 is generated by merging the codewords in U_1 through a special process. Fig.5.6 illustrates this process for merging two 16-D codewords into a single 4-D codeword. In order to generate the first codeword of U_2, **P** (label = 0), the codewords **A** (label = 0) and **B** (label =1) are merged as follows:

Step 1. First, all the blocks (16-D vectors) represented by codewords **A** and **B** in the original test images (resolution = 512 x 512 pixels) are marked.

Step 2. Next, the 4-D vectors located in the corresponding positions in the downsampled test images (resolution = 256 x 256 pixels) are marked.

Step 3. The codeword **P** is computed by calculating the mean of the vectors marked in the downsampled test images.

Here, the label bits of codeword **P** constitute the first 7 MSBs of the labels of both the codewords, **A** and **B**. The next codeword of codebook U_2, **Q** (label =1), is similarly computed by merging codewords **C** (label =2) and **D** (label =3) of U_1. Codeword **P** will now be used to represent all those vectors in image I_2, whose corresponding vectors in image I_1 were represented either by codeword **A** or codeword **B**. Since **A** and **B** are reasonably close neighbours (due to binary splitting technique), the codeword **P** provides a good reconstruction performance for image I_2. We note that it is possible to generate smaller U_2 codebooks (i.e. $N_1/N_2 > 2$) by merging more than 2 consecutive codewords of U_1 into a single U_2 codeword following the above steps in a similar fashion. The codebook U_3, for reconstructing the image at the next lower spatial resolution I_3 (128 x 128 pixels) can be obtained similarly by merging the codewords of U_2. We note that the above procedure can be generalized for any size image I_1 and block sizes $B_1 \times B_1$.

5.3 Wavelet based scalable vector quantization

We now apply the SVQ technique to WVQ algorithm (WSVQ) for achieving spatial

scalability (in the wavelet transform domain) and provide a high compression performance. In WSVQ, the original image is first decomposed into a 3 level pyramid structure (refer Fig.5.1). In order to jointly compress the sub-images, 64-D vectors are formed as in WVQ, which are then vector quantized using a pre-generated codebook U_1'. The codebook U_1' is generated from a large number of test images using the LBG algorithm which can be combined with the NIVQ technique to reduce the computational complexity. The full size image, I_1' (resolution = 512 x 512 pixels), is reconstructed by using all the bits of the VQ labels, each of which serves as an index to the codebook U_1'. Since the image I_1' is reconstructed in the transform domain, an inverse wavelet transform is applied to obtain a spatial domain reconstruction. We note that due to wavelet decomposition, the image at the next lower resolution (= 256 x 256 pixels), I_2', can be reconstructed using only the first 16 elements (corresponding to lower resolution sub-images) of each 64-D vector in the full size image. Therefore, a 16-D codebook U_2' is required for reconstructing the image I_2', whose dimensions correspond to the first 16 elements of codewords belonging to U_1'. Since I_2' has to be reconstructed by decoding only the first few MSBs of each label, U_2' should be smaller in size than U_1'. The codebook U_2' is designed by merging the codewords of U_1' using the SVQ technique as shown in Fig.5.7. For example, codeword **P** (label =0) in U_2' is generated by merging codewords **A** (label =0) and **B** (label =1) of U_1'. For this purpose, all the vectors represented by codewords **A** and **B** in the (wavelet decomposed) test images are marked. Codeword **P** is computed by calculating the mean of all the marked vectors for only the first 16 elements. Codeword **P** will now be used to reconstruct all those (16-D) vectors in image I_2' whose full size (64-D) vectors in I_1' were represented by either of the two codewords, **A** or **B**. This ensures partial decodability of VQ labels to obtain the image in lower sizes or spatial resolutions. Since the images I_1' and I_2' are reconstructed in the transform domain, an inverse wavelet transform is applied to obtain corresponding spatial domain reconstructions. We note that the WSVQ algorithm can be further extended like SVQ to obtain the image at a resolution of 128 x 128 pixels. In addition, the algorithm can

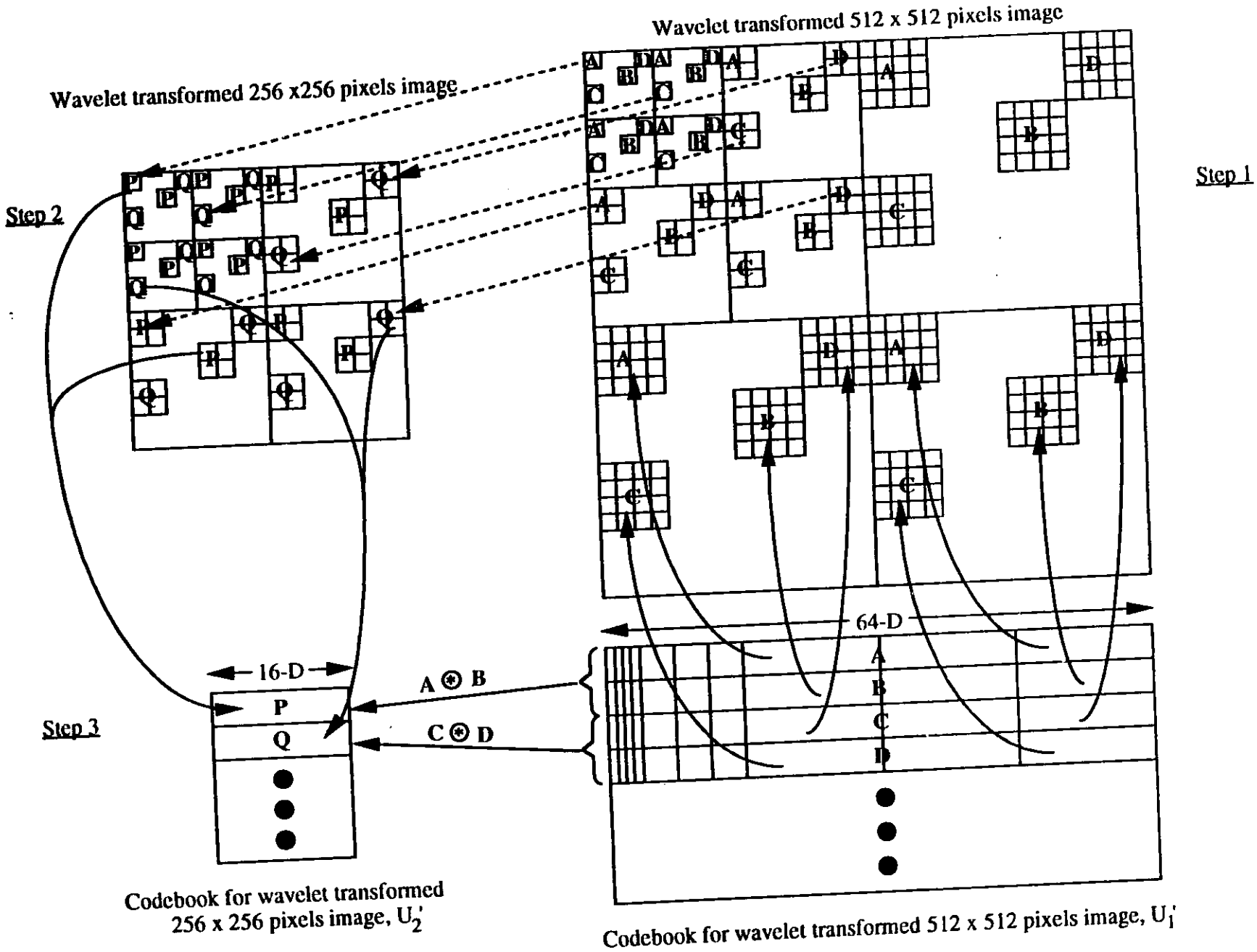


Figure 5.7: Application of SVQ technique to wavelet transformed images. The merging operation is indicated by ⊛.

be generalized for full size image I_1' of different spatial resolutions.

5.4 Simulation Results

Simulations have been performed for both the SVQ and WSVQ techniques on the "Lena" and "Baboon" test images (Fig.4.5) of size 512 x 512 pixels quantized to 8 bits/pixel. We note that (a) and (b) parts of all the image figures correspond to "Lena" and "Baboon" images, respectively. The images are reconstructed at resolutions of 64 x 64, 128 x 128, 256 x 256 and 512 x 512 pixels, respectively for both the techniques. Details of the simulations follows.

In order to apply the SVQ technique, a 16-D universal codebook U_1 of size 256 is first generated from a training set of 10 images, each of size 512 x 512 pixels. For this purpose, all the images are first partitioned into non-overlapping blocks of size 4 x 4 pixels which are reorganized as 16-D vectors. The codebook U_1 is generated using the LBG algorithm on the training vectors with the binary splitting technique. The original "Lena" and "Baboon" images are vector quantized using codebook U_1 and are reconstructed at the full spatial resolution I_1 (512 x 512 pixels) by decoding all the 8 (= $\log_2 256$) label bits of each VQ label. In order to reconstruct the test images at lower spatial resolutions, the training images are then downsampled to resolutions of 256 x 256 and 128 x 128 pixels by successive application of biorthogonal wavelet transform. To obtain the images at a resolution of 256 x 256 pixels (I_2), a 4-D universal codebook U_2 of size 128 (= 2^7) is generated by merging the codewords of U_1 using SVQ. A 1-D codebook U_3 of size 32 (= 2^5) is similarly generated by merging the codewords of U_2 (using SVQ) for reconstructing the images at a size of 128 x 128 pixels (I_3). The test images are then obtained at I_3 and I_2 resolutions by using only the first 5 and 7 MSBs of the labels, which serve as an index to codebooks U_3 and U_2, respectively for a table lookup operation. An iconic image of size 64 x 64 pixels (I_4) is obtained by reconstructing only the alternate 1-D vectors, in both the horizontal and vertical

directions, in the I_3 image. This operation is equivalent to subsampling the 128 x 128 pixels image. The reconstructed images are shown in Fig.5.8.

In order to generate universal codebooks for the WSVQ algorithm, the images in the training set are first decomposed into a 3-level pyramid structure by successive application of the wavelet transform. This is followed by formation of 64-D vectors by combining the corresponding blocks in the wavelet sub-images as shown in Fig.5.1. A 64-D codebook U_1' of size 4096 is then generated for the full size image using the NIVQ technique. For this purpose, we have used the first 16 elements of each (64-D) vector as the feature vector. The (wavelet decomposed) test images are reconstructed at the full spatial resolution I_1' (512 x 512 pixels) using all the 12 (= $\log_2 4096$) bits of each VQ label. In order to reconstruct the test images at a resolution of 256 x 256 pixels (I_2'), a 16-D universal codebook U_2' of size 1024 (= 2^{10}) is generated by merging the codewords of U_1' using WSVQ. A 4-D codebook U_3' of size 256 (= 2^8) is similarly generated by merging the codewords of U_2' (using WSVQ) for reconstructing the images at a size of 128 x 128 pixels (I_3'). Finally, a 1-D codebook U_4' of size 64 (= 2^6) is obtained by merging the codewords of U_3' for image reconstruction at 64 x 64 pixel resolution (I_4'). The test images are then obtained at the I_4', I_3' and I_2' resolutions by using only the first 6, 8 and 10 MSBs of theVQ labels, which serve as an index to codebooks U_4', U_3' and U_2', respectively for a table lookup operation. An inverse wavelet transform is applied on the I_1', I_2' and I_3' images to obtain corresponding spatial domain reconstructions. The reconstructed images are shown in Fig.5.9.

We will now compare the performance of SVQ and WSVQ with that of SPS-MF and SPS-WF techniques (refer chapter 4). As the original image is available only at the highest spatial resolution, therefore for a proper comparison, the low resolution reconstructed images have been upsampled to the full size (i.e. 512 x 512 pixels) using the mean interpolation filter. The upsampled images for the SVQ and WSVQ techniques are shown in Figs.5.10 and 5.11, respectively and their PSNR values are tabulated in Table 5.1. The bit rates shown in

(i)

(ii)

(iii)

(iv)

Figure 5.8(a): Lena reconstructed at (i) 64 x 64, (ii) 128 x 128, (iii) 256 x 256, and (iv) 512 x 512 pixel resolutions using the SVQ technique.

(i)

(ii)

(iii)

(iv)

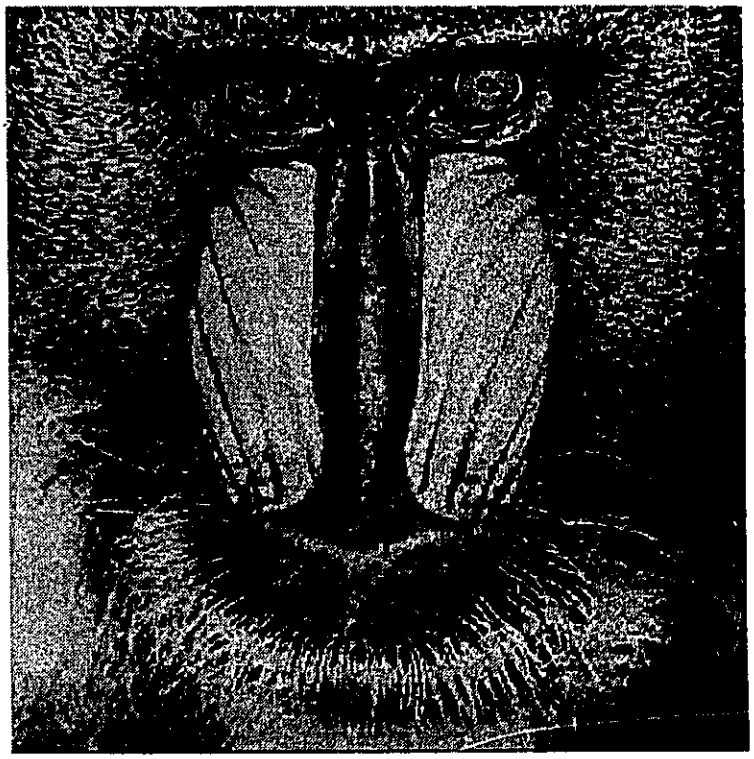


Figure 5.8(b): Baboon reconstructed at (i) 64 x 64, (ii) 128 x 128, (iii) 256 x 256, and (iv) 512 x 512 pixel resolutions using the SVQ technique.

(i)

(ii)

(iii)

(iv)

Figure 5.9(a): Lena reconstructed at (i) 64 x 64, (ii) 128 x 128, (iii) 256 x 256, and (iv) 512 x 512 pixel resolutions using the WSVQ technique.

(i)

(ii)

(iii)

(iv)

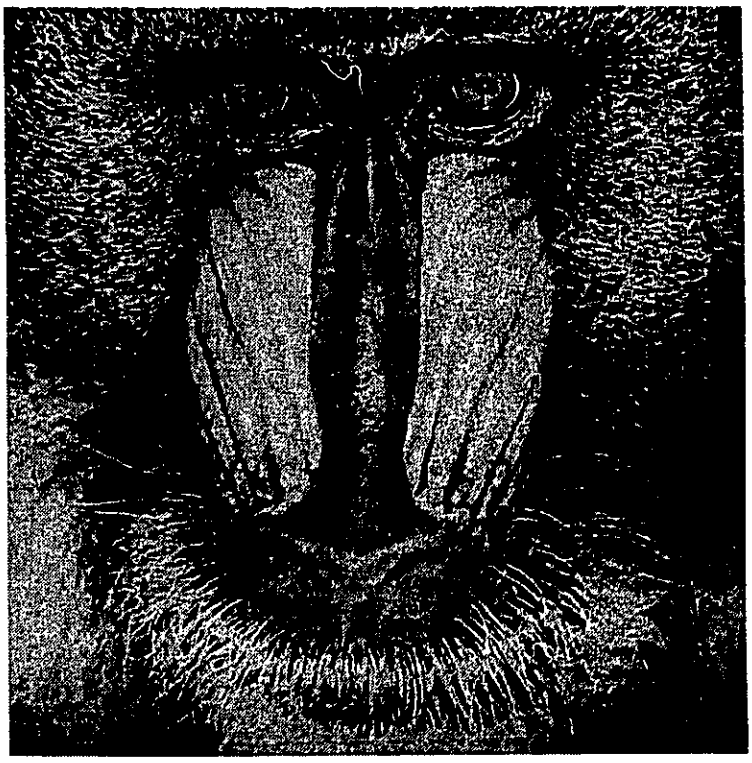


Figure 5.9(b): Baboon reconstructed at (i) 64 x 64, (ii) 128 x 128, (iii) 256 x 256, and (iv) 512 x 512 pixel resolutions using the WSVQ technique.

Figure 5.10(a) -(i): Lena reconstructed at 64 x 64 pixel resolution using SVQ technique followed by upsampling to 512 x 512 pixels.

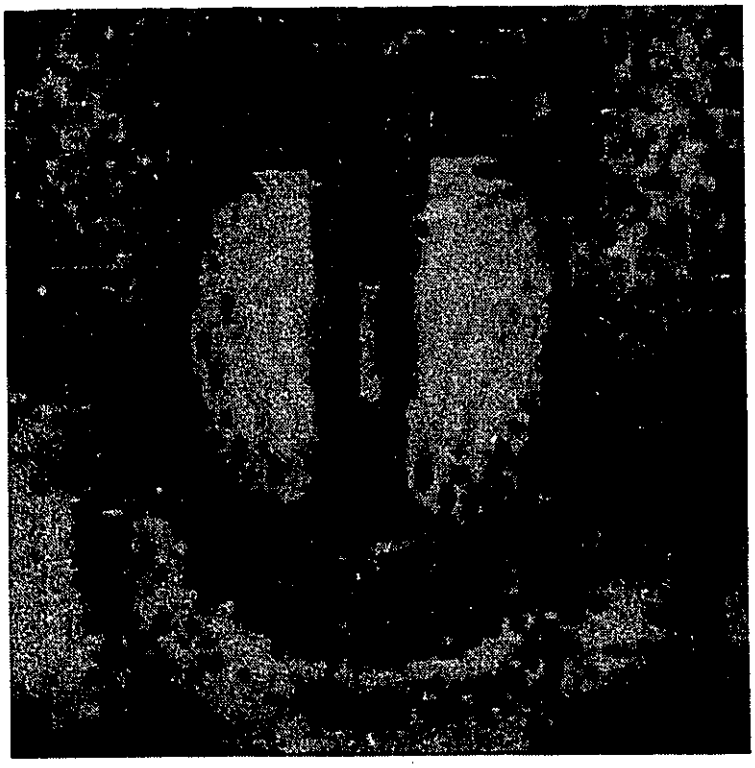


Figure 5.10(b) -(i): Baboon reconstructed at 64 x 64 pixel resolution using SVQ technique followed by upsampling to 512 x 512 pixels.

Figure 5.10(a) -(ii): Lena reconstructed at 128 x 128 pixel resolution using SVQ technique followed by upsampling to 512 x 512 pixels.

Figure 5.10(b) -(ii): Baboon reconstructed at 128 x 128 pixel resolution using SVQ technique followed by upsampling to 512 x 512 pixels.

Figure 5.10(a) -(iii): Lena reconstructed at 256 x 256 pixel resolution using SVQ technique followed by upsampling to 512 x 512 pixels.

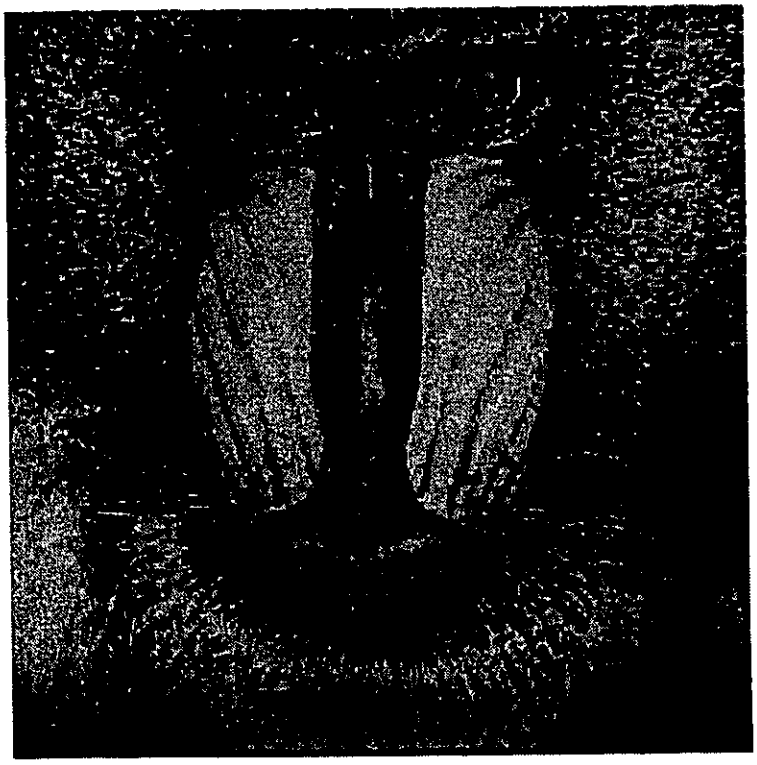


Figure 5.10(b) -(iii): Baboon reconstructed at 256 x 256 pixel resolution using SVQ technique followed by upsampling to 512 x 512 pixels.

Figure 5.11(a) -(i): Lena reconstructed at 64 x 64 pixel resolution using WSVQ technique followed by upsampling to 512 x 512 pixels.

Figure 5.11(b) -(i): Baboon reconstructed at 64 x 64 pixel resolution using WSVQ technique followed by upsampling to 512 x 512 pixels.

Figure 5.11(a) -(ii): Lena reconstructed at 128 x 128 pixel resolution using WSVQ technique followed by upsampling to 512 x 512 pixels.

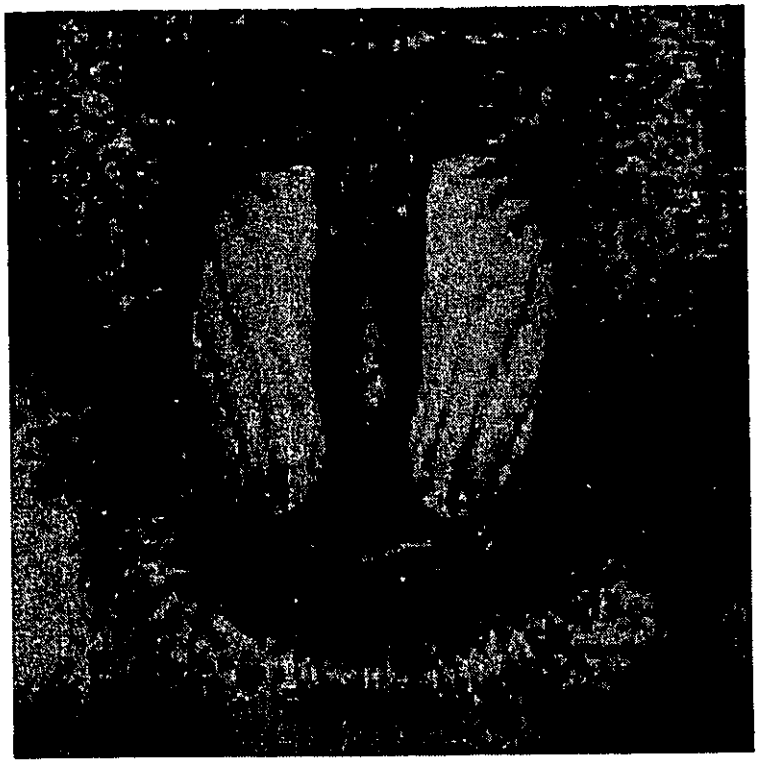


Figure 5.11(b) -(ii): Baboon reconstructed at 128 x 128 pixel resolution using WSVQ technique followed by upsampling to 512 x 512 pixels.

Figure 5.11(a) -(iii): Lena reconstructed at 256 x 256 pixel resolution using WSVQ technique followed by upsampling to 512 x 512 pixels.

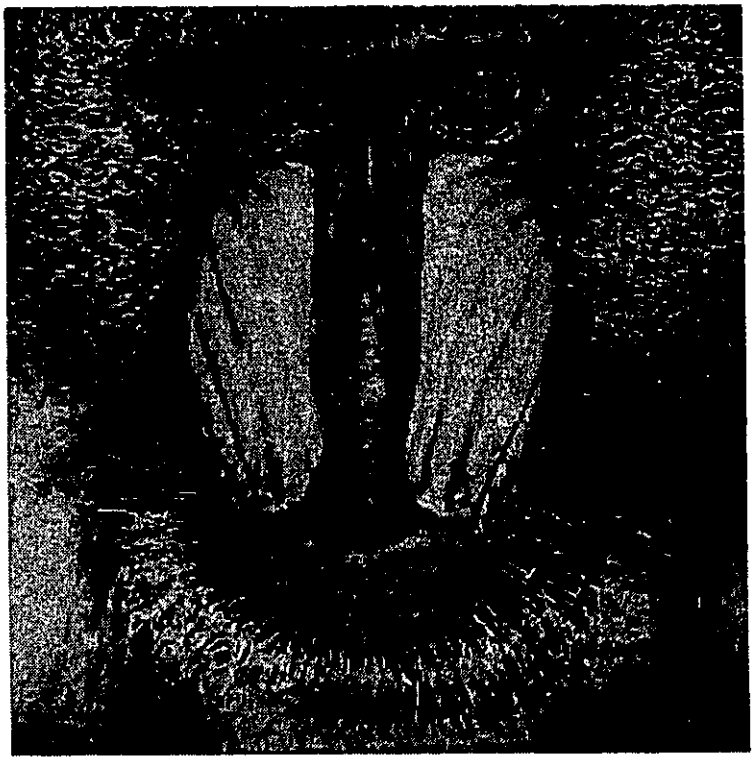


Figure 5.11(b) -(iii): Baboon reconstructed at 256 x 256 pixel resolution using WSVQ technique followed by upsampling to 512 x 512 pixels.

Table 5.1: Performance comparison of the SVQ and WSVQ techniques for achieving spatial scalability

SVQ

Size of the reconstructed image before upsampling to 512 x 512 pixels	Codebook size	Bit rate (bits per pixel)	PSNR of the upsampled images (dB)	
			Lena	Baboon
64 x 64	32	0.078	22.45	18.88
128 x 128	32	0.313	24.35	19.41
256 x 256	128	0.438	27.87	21.51
512 x 512	256	0.5	29.05	23.1

WSVQ

Size of the reconstructed image before upsampling to 512 x 512 pixels	Codebook size	Bit rate (bits per pixel)	PSNR of the upsampled images (dB)	
			Lena	Baboon
64 x 64	32	0.078	22.4	19.36
128 x 128	256	0.125	25.24	19.97
256 x 256	1024	0.156	27.66	21.3
512 x 512	4096	0.188	29.85	24.16

(i)

(ii)

(iii)

Figure 5.12(a): Lena reconstructed at (i) 128 x 128, (ii) 256 x 256, and (iii) 512 x 512 pixel resolutions using the WSVQ+JPEG technique.

(i)

(ii)

(iii)

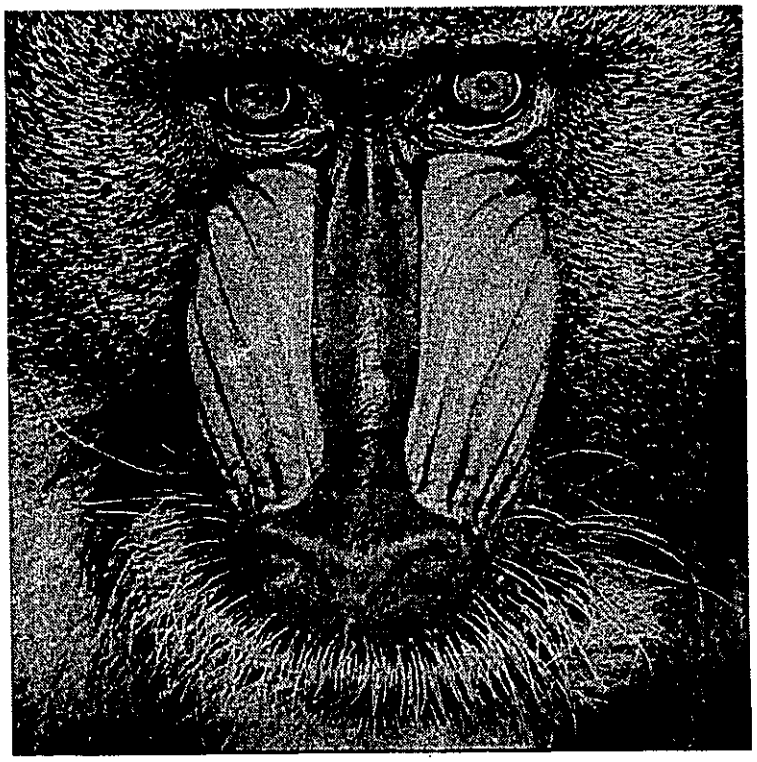


Figure 5.12(b): Baboon reconstructed at (i) 128 x 128, (ii) 256 x 256, and (iii) 512 x 512 pixel resolutions using the WSVQ+JPEG technique.

Figure 5.13(a) -(i): Lena reconstructed at 128 x 128 pixel resolution using WSVQ+JPEG technique followed by upsampling to 512 x 512 pixels.

Figure 5.13(b) -(i): Baboon reconstructed at 128 x 128 pixel resolution using WSVQ+JPEG technique followed by upsampling to 512 x 512 pixels.

Figure 5.13(a) -(ii): Lena reconstructed at 256 x 256 pixel resolution using WSVQ+JPEG technique followed by upsampling to 512 x 512 pixels.

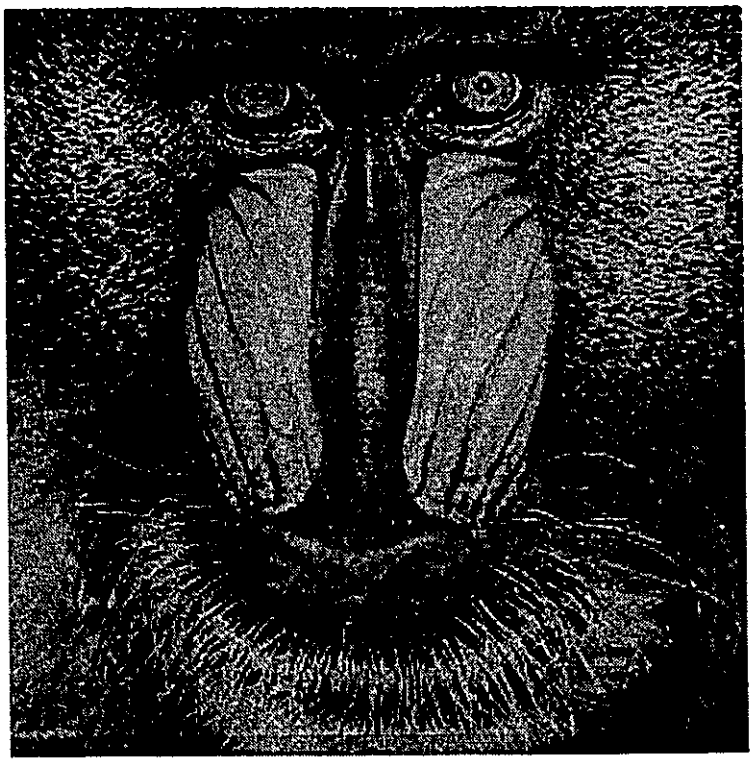


Figure 5.13(b) -(ii): Baboon reconstructed at 256 x 256 pixel resolution using WSVQ+JPEG technique followed by upsampling to 512 x 512 pixels.

Table 5.1 correspond to the spatial size of 512 x 512 pixels. It can be seen that the SVQ technique achieves lower compression compared to the other three techniques. On the other hand, WSVQ algorithm provides an excellent performance at almost all resolutions. Since the PSNR values of WSVQ technique do not directly correspond to those of SPS-WF technique, we have combined the WSVQ technique with the baseline JPEG algorithm. The difference D_i (i = 1, 2, 3) between the WSVQ reconstructed (spatial domain) images, I_1', I_2' and I_3' and the corresponding downsampled version of the original test images is compressed using the baseline JPEG algorithm. The resulting bitstreams are decompressed to reconstruct the difference images D_i^*, which are then added to the respective I_i' images to obtain (a high quality) small size reconstructions of the test images, I_i^*. Fig.5.12 shows I_i^* for both the "Lena" and "Baboon" test images. To obtain a meaningful comparison with the SPS-WF technique, the I_i^* images have been upsampled to 512 x 512 pixels using the mean interpolation filter. The upsampled images are shown in Fig.5.13 and their bit rates are tabulated in Table 5.2. It can be seen that the proposed WSVQ+JPEG technique clearly outperforms the SPS-WF technique, specially for the higher resolution images.

Table 5.2: Performance of the WSVQ+ JPEG algorithm.

Size of the reconstructed image before upsampling to 512 x 512 pixels	Lena		Baboon	
	Bit rate (bits per pixel)	PSNR (dB)	Bit rate (bits per pixel)	PSNR (dB)
128 x 128	0.168	26.54	0.201	20.6
256 x 256	0.425	31.33	0.66	23.34
512 x 512	0.734	34.53	1.188	29.89

5.5 Summary

In this chapter, we presented a novel algorithm based on wavelet transform and vector quantization (WSVQ) for achieving spatial scalability. Although, wavelet transform provides a scalable image representation, vector quantization is not scalable. As a first step towards developing the WSVQ algorithm, a new technique, SVQ, was proposed to achieve spatial scalability using vector quantization. SVQ ensures partial decodability of VQ labels by using different codebooks for image reconstruction, one for each spatial resolution. In contrast to the TSVQ technique for achieving SNR scalability, SVQ provides a good quality image at the full spatial resolution. Most importantly, SVQ does not require any overhead bits to provide a scalable bitstream.

We then applied the SVQ technique to the WVQ algorithm resulting in a high performance algorithm, WSVQ. The proposed algorithm not provides a high compression performance but also provides spatial scalability without requiring any overhead bits. In addition, the computational complexity of WSVQ can be substantially reduced by using NIVQ technique. WSVQ clearly outperforms the hierarchical mode of JPEG, when combined with the Baseline JPEG algorithm.

Chapter 6

Conclusions and Future work

6.1 Conclusions

In this thesis, we first investigated and evaluated the algorithms of the JPEG standard for scalable image compression. JPEG has four modes of operation, namely baseline sequential, lossless compression, progressive coding and hierarchical coding. The baseline sequential algorithm provides the basic feature of DCT based lossy image compression, while the lossless compression uses a DPCM based algorithm to achieve perfect image reconstruction. The progressive and hierarchical coding modes provide SNR and spatial scalability, respectively. Among the two techniques proposed by the standard for progressive mode, the spectral selection technique provides a superior coding performance compared to successive approximation. The hierarchical mode uses a pyramidal structure for encoding the different spatial resolutions of the image. However, the standard does not specify downsampling filters for obtaining the original image in smaller sizes. We used a simple mean filter to downsample the image. This filter does not perform well in extracting the image features resulting in poor subjective quality. The performance of the hiearchical mode was enhanced by using wavelet transform for downsampling the image. The proposed wavelet based hierarchical JPEG coder provides a superior subjective image

quality due to excellent feature extraction capability of wavelet transform. This technique does not require any modifications to the existing JPEG decoders.

We then investigated the performance of transmitting scalable images over ATM networks which employ a dual priority transmission mechanism. In order to transfer the image between two network nodes, the (compressed) image data is packetized into high and low priority cells, each having a payload size of 48 bytes. In the event of network congestion, some of the low priority cells may have to be discarded to reduce the network load. The progresssive and hierarchical modes of JPEG provide a graceful degradation in performance compared to the baseline sequential mode in event of cell losses. This is because the latter two modes provide scalable image compression by reorganizing the image data into several layers containing information of different visual importance and therefore different priority.

The hierarchical mode of JPEG does not provide a good compression performance at low bit rates. A high performance algorithm based on wavelet transform and vector quantization has been reported in the literature for low bit rate image compression. Although, wavelet transform provides scalable image representation, vector quantization is not scalable. As a first step towards making this algorithm scalable, a new technique was proposed in this thesis to achieve spatial scalability using vector quantization. This technique does not require any overhead bits to provide a scalable image bitstream. We then applied this technique to the combined wavelet and vector quantization based compression algorithm. Our simulation results confirmed that the proposed algorithm not only achieves a high compression performance, but also provides spatial scalability without requiring any overhead bits.

6.2 Future Work

The extension of the proposed algorithms for scalable compression of color images is an area of future work. Secondly, the JPEG standard does not specify many parametric details such as the color space designation (e.g.YUV or RGB), pixel aspect ratio, etc. which are generally application dependent. It is of considerable importance to study the effect of these parameters on the subjective image quality for implementation in various applications.

Scalable image compression is very useful for image transmission over dual priority channels such as ATM networks. In this thesis, images have been reconstructed using only the available high and low priority information in the event of loss of some of the low priority data. However, it may be possible to heuristically interpolate most of the missing low priority data by exploiting intercorrelation between the image data. Development of the algorithms for retrieval of the lost data is an interesting area of future research. Efficient algorithms can be designed for color images due to high degree of correlation between the three color components.

WSVQ is a high performance scalable image compression algorithm, but it may be possible to obtain further reductions in the bit rate and improve the quality of reconstructed images by designing separate codebooks for different image features, such as edges, textures, etc. Although the computational complexity of WSVQ can be greatly reduced using the NIVQ technique, however it still requires special purpose architectures for real time implementation. Improvement of the WSVQ algorithm and development of a codec for its real time implementation are areas for further research.

All of the algorithms discussed in this thesis are for scalable compression of still images. Recently, digital video is gaining importance. Extensions of these algorithms for scalable video compression is an area of considerable interest. We note that the JPEG standard algorithms can be extended to provide scalable video compression within the framework of proposed MPEG-II (Moving Pictures Experts Group) video compression standard. A

related field of research is scalable video transmission over high bandwidth dual priority channels such as ATM networks.

Finally, integration of scalable images into multimedia documents for realtime database applications opens up many new areas of future research. In particular, some of the interesting issues are indexing and editing of images and its synchronization with other media types such as voice and text for a jitter-free playback.

Bibliography

[1] R.G.Gallager, *Information Theory and Reliable Communication*, John Wiley, New York, 1968.

[2] N.S.Jayant and Peter Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.

[3] C.E.Shannon, "Coding Theorems for a Discrete Source with a Fidelity Criterion", *IRE National Convention Record*, Part 4, pp.142-163, 1959.

[4] L.D.Davisson, "Rate-Distortion Theory and Applications", *Proceedings of the IEEE*, pp.156-164, July 1972.

[5] A.K.Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[6] J.O.Limb, "Distortion Criteria of the Human Viewer", *IEEE Transactions on Systems, Man and Cybernatics*, pp.778-793, December 1979.

[7] J.A.Saghri, P.S.Cheatham and A.Habibi, "Image Quality Measure based on a Human Visual System Model", *Journal of Optical Engineering*, pp.813-818, July 1989.

[8] D.A.Huffman, "A Method for the Construction of Minimum-redundancy Codes", *Proceedings of the I.R.E.*, pp.1098-1101, September 1952.

[9] G.K.Wallace, "The JPEG Still Picture Compression Standard", *Comunications of the ACM*, vol.34, no.4, pp.30-45, April 1991.

[10] A.Habibi, "Survey of Adaptive Image Coding Techniques", *IEEE Trnsactions on Communications*, vol. COM-25, no.11, pp.1275-1284, November 1977.

[11] K.R.Rao and P.Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*, Academic Press, London, 1990.

[12] R.J.Clarke, *Transform Coding of Images*, Academic Press, London, 1985.

[13] J.W.Woods and S.D.O'Neil, "Subband Coding of Images", *IEEE Trans. Acous. Speech Signal Processing*, ASSP-34(5), 1278-1288, 1986.

[14] P.P.Vaidyanathan, "Quadrature Mirror Filter Banks, M-band Extensions and Perfect -reconstruction Techniques", *IEEE ASSP Magazine*, 4(3), 4-20, 1987.

[15] C.R.Galand and H.J.Nussbaumer, "New Quadrature Mirror Filter Banks", in *Proc. ICASSP*, pp.291-294, 1980.

[16] Joint Photographics Experts Group, ISO/IEC/JTC1/SC2/WG8, "JPEG Draft International Standard", 1991.

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

[18] R.M.Gray, "Vector Quantization", *IEEE ASSP Magazine*, pp.4-29, April 1984.

[19] N.M.Nasrabadi and R.A.King, "Image Coding using Vector Quantization: A Review", *IEEE Trans. on Communications*, pp.957-971, August 1988.

[20] Y.Linde, A.Buzo, R.Gray, "An Algorithm for Vector Quantizer Design", *IEEE Transactions on Communications*, pp.84-95, January 1980.

[21] A.Gersho and B.Ramamurthi, "Image Coding using Vector Quantization", *IEEE International Conf. on Accous., Speech and Signal Processing*, pp.428-431, May 1982.

[22] D.S.Kim and S.U.Lee, "Image Vector Quantization based on Classification in the DCT Domain", *IEEE Trans. on Communications*, vol.39, no.4, pp.549-556, April 1991.

[23] I.Dinstein, K.Rose and A.Heiman, "Variable Block Size Transform Image Coder", *IEEE Trans. on Communications*, vol.38, no.11, pp.2073-2078, November 1990.

[24] S.Gupta and A.Gersho, "Image Vector Quantization with Block Adaptive Scalar Prediction", *SPIE Visual Communications and Image Processing: Visual Communications*, vol.1605, pp.179-189, 1991.

[25] T.Murakami, K.Asai and E.Yamazaki, "Image Sequence Coding", *Electronics Letters*, vol.18, no.23, pp.1005-1006, November 1982.

[26] M.Goldberg, P.Boucher, S.Shlien, "Image Compression using Adaptive Vector Quantization", *IEEE Trans. on Communications*, vol.COM-34, pp.180-187, February 1986.

[27] B.Ramamurthi and A.Gersho, "Classified Vector Quantization of Images", *IEEE Trans. on Communications*, vol.COM-34, no.11, pp.1105-1115, 1986.

[28] J.Kim and S.Lee, "A Transform Domain Classified Vector Quantizer for Image Coding", *IEEE Trans. on Circuits and Systems for Video Technology*, vol.2, no.1, pp.3-14, 1992.

[29] R.Aravind and A.Gersho, "Low-rate Image Coding with Finite-state Vector Quantization", *IEEE Conference on Accous. Speech and Signal Processing*, pp.4.3.1-4.3.4, 1986.

[30] S.Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Trans. on Pattern Anal. and Mach. Intel.*, vol.11, no.7, July 1989.

[31] I.Daubechies, "Orthonormal Bases for Compactly Supported Wavelets", *Communications of Pure and Applied Mathematics*, pp.909-996, November 1988.

[32] I.Daubechies, "Orthonormal Bases of Wavelets with Finite Support - Connection with Discrete Filters", Proc. of International Conference on Wavelets, pp.38-66, 1987.

[33] I.Daubechies, "The Wavelet Transform, Time-frequency Localization and Signal Analysis", *IEEE Trans. on Information Theory*, vol.36, no.5, pp.961-1005, May 1990.

[34] M.Antonini, M.Barlaud, P.Mathieu and I.Daubechies, "Image Coding using Wavelet Transform", *IEEE Trans. on Image Processing*, vol.1, no.2, April 1992.

[35] A.S.Lewis and G.Knowles, "Image Compression using 2-D Wavelet Transforms", *IEEE Trans. on Image Processing*, vol.1, no.2, April 1992.

[36] M.R.Banham and B.J.Sullivan, "A Wavelet Transform Image Coding Technique with a Quadrature Structure", Publisher.

[37] Xiping Wang and S.Panchanthan, "Wavelet Transform Coding using NIVQ", *Visual Communications and Image Processing '93, SPIE*, vol.2094, part 2, pp.999-1009, Cambridge, Massachussetts, Novemeber 1993.

[38] C.Gonzales and E.Viscito, "Flexibly Scalable Digital Video Coding", *Signal Processing: Image Communication*, vol.5, pp.5-20, Elsevier Publishers, 1993.

[39] T.Hanamura, W.Kameyama and H.Tominaga, "Hierarchical Coding Scheme of Video Signal with Scalability and Compatability", *Signal Processing: Image Communication*, vol.5, pp.159-184, Elsevier Publishers, 1993.

[40] A.Puri and A.Wong, "Spatial Domain Resolution Scalable Video Coding", *Visual Communications and Image Processing '93, SPIE*, vol.2094, part 1, pp.718-729-, Cambridge, Massachussetts, Novemeber 1993.

[41] T.Chiang and D.Anastassiou, "Hierarchical Coding of Digital Television", *IEEE Communications Magazine*, pp.38-45, May 1994.

[42] B.Girod, "Scalable Video for Multimedia Workstations", *Computer & Graphics*, vol.17, no.3, pp.269-276, 1993.

[43] A.Sanz, C.Munoz and N.Garcia, "Approximation Quality Improvement Techniques in Progressive Image Transmission", *IEEE Journal on Selected Areas in Communications*, vol.SAC-2, pp.339-373, March 1984.

[44] L.Wang and M.Goldberg, "Lossless Progressive Image Transmission by Residual Vector Quantization", *Proceedings of the Thirteenth Biennial Symposium on Communications*, pp.c.1.9-c.1.12, Kingston, Ont., June 1986.

[45] M.I.Szean, M.Rabbani and P.W.Jones, "Progressive Transmission of Images using a Prediction/Residual Encoding Approach", *Opt.Eng.*, vol.28, no.5, pp.556-564, 1989.

[46] L.Wang and M.Goldberg, "Progressive Image Transmission using Vector Quantization on Images in Pyramid Form", *IEEE Trans. Comm.*, vol. Com-37, Dec. 1989.

[47] L.Wang, "Ph.D. Thesis", *Dept. of Electrical Engg., University of Ottawa*, Ottawa, Canada, Jan.1988.

[48] P.J.Burt and E.H.Adelson, "The Laplacian Pyramid as a Compact Image Code", *IEEE Transactions on Communications*, vol. COM-31, pp.532-540, April 1983.

[49] J.Naor and S.Peleg, "Hierarchical Image Representation for Compression, Filtering and Normalization", *Pattern Recognition Letters*, pp.43-46, October 1983.

[50] A.Tran, K.M.Liu, K.H.Tzou and E.B.Vogel, "An Efficient Pyramid Image Coding System", *IEEE International Conference on ASSP*, Dallas, Texas, pp.744-747, April 1987.

[51] H.M.Dreizen, "Content-driven Progressive Transmission of Grey-scale Images", *IEEE Transactions on Communications*, vol.COM-35, pp.289-296, March 1987.

[52] W.D.Hoffman and D.E.Troxel, "Making Progressive Transmission Adaptive", *IEEE Transactions on Communications*, vol.COM-34, pp.806-813, August 1986.

[53] K.H.Tzou and S.E.Elnahas, "An Optimum Progressive Transmission and Reconstruction Scheme for Transformed Image", *Proceedings of IEEE International Conference on Comm.*, Toronto, Ont., pp.413-418, June 1986.

[54] K.N.Ngan, "Image Display Techniques using Cosine Transform", *IEEE Transactions on ASSP*, vol.ASSP-32, pp.173-177, February 1984.

[55] Benoit B. Mandelbrot, "The Fractal Geometry of Nature", *Publisher*, 1983.

[56] T.H.Wendler and D.Meyer-Ebrecht, "Proposed Standard for Variable Format Picture Processing and a Codec Approach to Match Diverse Imaging Devices",

SPIE, vol.318, part 1, Picture Archiving and Communications Systems (PACS) for Medical Applications, pp.298-305, 1982.

[57] P.Boucher and M.Goldberg, "Transform Image Coding by Vector Quantization", *Ninth Symp. on Signal Proc. and App.*, pp.629-633, Nice, France, May 1983.

[58] L.Wang and M.Goldberg, "Progressive Image Transmission by Multistage Vector Quantization on Images in Pyramid Form", *Proc. IEEE Intl. Conf. on Communications*, Toronto, Ont., pp.419-423, June 1986.

[59] K.Knowlton, "Progressive Transmission of Grey-scale and Binary Pictures by Simple, Efficient and Lossless Encoding Scheme", *Proceedings of IEEE*, vol.68, pp.885-896, July 1980.

[60] F.S.Hill, S.Walker and F.Gao, "Interactive Image Query System using Progressive Transmission", *Comput. & Graphics*, vol.17, no.3, pp.323-333, 1983.

[61] S.Tubaro, "A Two Layers Video Coding scheme for ATM Networks", *Signal Processing: Image Communication*, vol.3, no.2-3, pp.179-195, June 1991.

[62] G.Morrison and D.Beaumont, "Two-layer Video Coding for ATM Networks", *Signal Processing: Image Communication*, vol.3, no.2-3, pp.129-141, June 1991.

[63] M.Ghanbari, "An Adapted H.261 Two-Layer Video Codec for ATM Networks", *IEEE Trans. on Communications*, vol.40, no.9, pp.1481-1490, Sept.1992.

[64] Y.C.Chen, K.Sayood and D.J.Nelson, "A Robust Coding Scheme for Packet Video", *IEEE Trans. on Communications*, vol.40, no.9, pp.1491-1501, Sept.1992.

[65] C.Guillemot and R.Ansari, "Layered Coding Schemes for Video Transmission on ATM Networks", *Journal of Visual Comm. and Image Representation*, vo.5, no.1, pp.62-74, March 1994.

[66] S.L.Sutherland and J.Burgin, "B-ISDN Interworking", *IEEE Comunications Magazine*, vol.31, no.8, pp.60-63, August 1993.

[67] ITU-draft Recommendation Q.93B, "B-ISDN User-Network Interface Layer 3 Specification for Basic Call/Bearer Control", May 1993.

[68] J.S.Turner, "Managing Bandwidth in ATM Networks with Bursty Traffic", *IEEE Network*, vol.6, no.5, pp.50-58, Sept.1992.

[69] V.S.Frost and M.T.Mullen, "Dynamic Bandwidth allocation for B-ISDN Based on End-to-End Delay Estimates", *Proc. IEEE Int. Conf. Commun.*, vol.1, pp.225-231, Chicago, June 1992.

[70] H.Saito, "Call Admission Control in an ATM Network using Upper Bound of Cell Loss Probability", *IEEE Trans. on Commun.*, vol.40, no.9, pp.1512-1521, Sept.1992.

[71] Special Issue on Teletraffic Analysis of ATM Systems, *IEEE JSAC*, April 1991.

[72] Special Issue on B-ISDN High Performance Transport, *IEEE Communications Magazine*, Sept.1991.

[73] Special Issue on Congestion Control in High-speed Packet Switched Networks", *IEEE JSAC*, Spet.1991.

[74] Special Issue on Congestion Control in High-speed Networks", *IEEE Communications Magazine*, Oct.1991.

[75] Q.F.Zhu, Y.Wang and L.Shaw, "Coding and Cell-Loss Recovery in DCT-Based Packet Video", *IEEE Trans. on CSVT*, vol.3, no.3, June 1993.

[76] A.Gersho, "Optimal Nonlinear Interpolative Vector Quantization", *IEEE Transactions on Communications*, vol.38, no.9, September 1990.