RD OPTIMIZED PROGRESSIVE IMAGE CODING USING JPEG

By

Jaehan In

B. Sc. (Electrical Engineering) The Johns Hopkins University, U.S.A.

M. Sc. (Electrical Engineering) The Johns Hopkins University, U.S.A.

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

March 1998

Department of Electrical and Computer Engineering
The University of British Columbia
2356 Main Mall
Vancouver, Canada
V6T 1Z4

Date: 4/22/98

Abstract

The JPEG standard allows four modes of operation. They are the hierarchical (HJPEG), progressive (PJPEG), sequential (SJPEG), and lossless modes[1]. The HJPEG and PJPEG modes inherently support progressive image coding. In HJPEG, an image is decomposed into subimages of different resolution, each of which is then coded using one of the other three modes of JPEG. Progressiveness within a resolution in HJPEG can be achieved when each subimage is coded using PJPEG. An image coded using PJPEG consists of scans, each of which contributes to a portion of the reconstructed image quality. While SJPEG yields essentially the same level of compression performance for most encoder implementations, the performance of PJPEG depends highly upon the designed encoder structure. This is due to the flexibility the standard leaves open in designing PJPEG encoders.

In this thesis, an efficient progressive image coding algorithm is developed that is compliant with the JPEG still image compression standard. The JPEG-compliant progressive image encoder is a HJPEG encoder that employs a rate-distortion optimized PJPEG encoding algorithm for each image resolution. Our encoder outperforms an optimized SJPEG encoder in terms of compression efficiency, substantially at low and high bit rates. Moreover, unlike existing JPEG compliant encoders, our encoder can achieve precise rate control for each fixed resolution. Such good compression performance at low bit rates and precise rate control are two highly desired features currently sought for the emerging JPEG-2000 standard.

[1] Lossless JPEG algorithms are rarely used since their performance levels are significantly lower than those of other lossless image compression algorithms, and are therefore not widely used.

Table of Contents

List of Tables

List of Figures

Acknowledgment

I would like to thank my supervisor, Dr. Faouzi Kossentini, for his supervision and advice throughout this research. His generous encouragement and support made this work possible. I would also like to thank my colleague, Shahram Shirani, for his contribution to a part of this research. I also thank my colleagues in the Signal Processing and Multimedia Laboratory for the kind friendship they provided me. Working with them was a pleasure. Personally, I would like to thank my parents for their encouragement and especially my wife, Gyunghwan, for her patience and support.

Glossary

JPEG: Joint Photographic Experts Group.

HJPEG: Hierarchical mode of JPEG.

PJPEG: Progressive mode of JPEG.

SJPEG: Sequential mode of JPEG.

PIT: Progressive image transmission.

RD: Rate-distortion.

HPJPEG: Hierarchically embedded progressive JPEG.

DCT: Discrete cosine transform.

FDCT: Forward discrete cosine transform.

IDCT: Inverse discrete cosine transform.

DPCM: Differential pulse code modulation.

VL: Variable length.

EOB: End of block.

Scan: A single pass through the data for one or more image components in an image.

Chapter 1

Introduction

Images usually contain so much data that many bits are still needed to represent them even after compression. Therefore transmitting whole compressed image data at once over a low bandwidth channel is often inefficient, and is not adequate for applications such as fast image browsing and video-teleconferencing [1, 2]. Under such circumstances, immediate, if not full, reconstruction of the compressed images is essential. Progressive image coding allows images to be progressively coded and transmitted, i.e., only a portion of the information about the original image is first coded and transmitted, and, upon the receiver's request, more bits representing the rest of the information are transmitted.

Progressive image coding can be achieved in two ways: 1) quality progressive image coding, and 2) hierarchical image coding. In quality progressive image coding, a rough approximation about the full image is obtained after decoding a few bits, and, as more bits are decoded, the image reproduction quality is gradually improved. Hierarchical image coding, also referred to as layered image coding, allows the transmitter and the receiver to control the resolution of the encoded/decoded images. In this coding algorithm, original images are often reduced from their original sizes by downsampling or averaging, coded, and transmitted. Later, higher and original resolutions of the input image can be reconstructed by decoding a relatively small number of bits [3, 4]. Both of the above coding algorithms can be combined to provide the receiver with maximum progressiveness control.

JPEG (Joint Photographic Experts Group) is the current image compression standard for still, continuous-tone, monochrome and color images based on discrete cosine transform (DCT). Among the four principal modes provided by JPEG, HJPEG and PJPEG support progressive image transmission (PIT).

Various algorithms using DCT have been developed for PIT. [5, 6, 7, 8, 9]. In PIT, prioritization of the bits to be transmitted takes an important role and is done in several different ways. In [5], the DCT coefficients are sorted in a decreasing order of their ensemble variance. This order is then modified according to a human visual system model. Another method suggested in [8] employs the magnitude of the DCT coefficients as a parameter in prioritization. The DCT coefficients are rounded to the same number of bits and are then sorted so that the larger coefficients with their positions are encoded and transmitted first, followed by the smaller ones.

The above described methods achieve PIT by exploiting various properties of the DCT coefficients. However, it is generally difficult to directly apply them to a JPEG framework, where many constraints must be satisfied during the encoding/decoding process. Although PIT using JPEG was discussed in some papers [6, 10], the performance levels of the corresponding implementations were quite limited. In [6] and [10], only the progressive mode of JPEG was discussed allowing only quality control. In [3, 4], hierarchical mode of JPEG using SJPEG as a submode was implemented allowing only spatial progressiveness.

In this thesis, an efficient RD optimized JPEG encoder is presented which supports spatial and quality progressive image encoding, simultaneously. The new encoder employs the progressive mode of JPEG as a submode of its hierarchical mode. Our encoding algorithm produces downsampled versions of the original image, which are then encoded, producing an ordered sequence of scans that can be used to progressively encode/decode an image, achieving precise RD control and yielding relatively good compression performance at all bit rates, including the very low bit rates.

Our encoding algorithm is compliant with the JPEG standard, in the sense that existing decoders can be used without modification although their reproduction quality levels are significantly higher using our algorithm.

In Chapter 2, we provide a description of JPEG, in particular its three popular modes of operation. In Chapter 3, a rate-distortion (RD) optimized PJPEG compliant coding algorithm is devised. In Chapter 4, rate-control within PJPEG is enabled taking into account all the syntax constraints of a PJPEG encoder. Finally, in Chapter 5, the RD optimized PJPEG algorithm discussed in Chapter 3 is used as a part of a hierarchically embedded progressive JPEG (HPJPEG) encoder. The resulting encoder is compliant with the JPEG standard. Finally, conclusions are presented in Chapter 6.

Chapter 2

Background

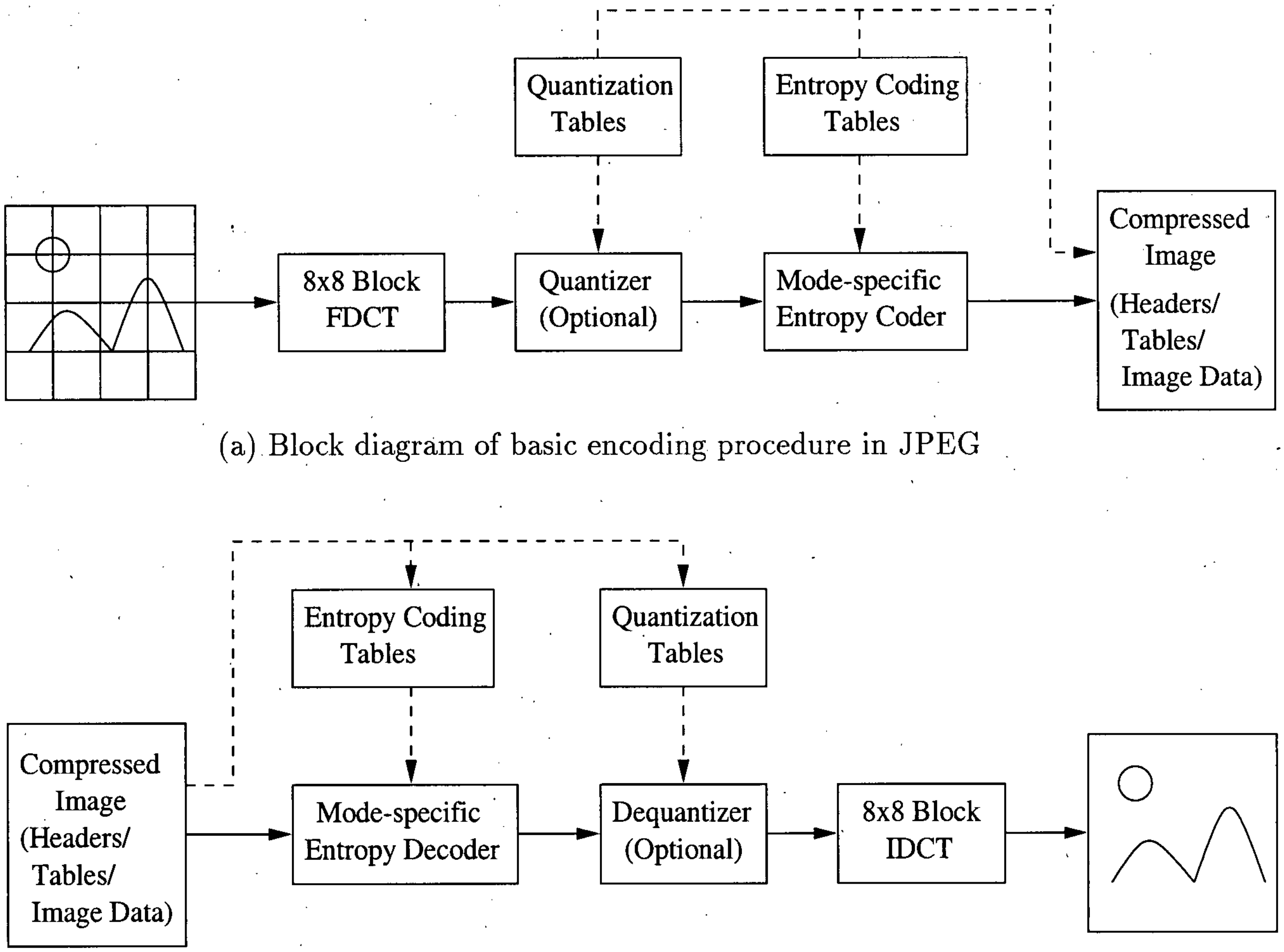
In the following sections, sequential DCT-based mode of JPEG, progressive DCT-mode of JPEG and hierarchical mode of JPEG are discussed in detail, following the common basic operations between the two modes.

2.1 Basics of DCT-Based JPEG

All DCT-based modes of operation in JPEG follow the same basic coding procedure shown in Figure 2.1. First, an input image is partitioned into 8×8 blocks. The first element of each 8×8 DCT block is called the DC coefficient and the rest are called the AC coefficients. Each 8×8 block is then transformed using the forward discrete cosine transform (FDCT) and optionally quantized. The DCT coefficients are then prepared for entropy encoding. The JPEG standard allows two algorithms for entropy coding, one Huffman-based and the other one based on arithmetic coding. Huffman-based coding is widely used because it is simple and publically available, whereas arithmetic coding (although more compression efficient) is more complex and is mostly patented. Therefore, we only consider Huffman-based JPEG modes in the rest of this thesis.

Finally, necessary information for decoding such as headers (image dimension and sample precision) and tables (quantization and Huffman tables) as well as the encoded data are stored in the compressed bit stream.

Likewise, at the decoder, the received bit streams are entropy decoded to provide 8×8 blocks of DCT coefficients, which are then dequantized and transformed using inverse



(a) Block diagram of basic encoding procedure in JPEG

(b) Block diagram of basic decoding procedure in JPEG

Figure 2.1: Basic coding procedure for DCT-based JPEG.

DCT (IDCT) to yield the reconstructed image. The FDCT and IDCT equations are given by

$$S(v,u) = \frac{C(v)}{2}\frac{C(u)}{2}\sum_{y=0}^{7}\sum_{x=0}^{7} s(y,x)cos[(2x+1)u\pi/16]cos[(2y+1)v\pi/16] \qquad (2.1)$$

and

$$s(y,x) = \sum_{v=0}^{7}\frac{C(v)}{2}\sum_{u=0}^{7}\frac{C(u)}{2}S(v,u)cos[(2x+1)u\pi/16]cos[(2y+1)v\pi/16], \qquad (2.2)$$

where $C(u) = 1/\sqrt{2}$ for $u = 0$, $C(u) = 1$ for $u > 0$, $C(v) = 1/\sqrt{2}$ for $v = 0$, $C(v) = 1$ for $u > 0$, $s(y,x)$ is the pixel value of an input 8×8 block and $S(v,u)$ is the DCT coefficient of an 8×8 DCT block. The 2-D DCT can be computed by 1-D DCT of the rows followed by 1-D DCT of the columns. Although 2-D algorithms are available, they are much more complex. In this thesis, the 1-D DCT algorithm described in [11] is implemented in integer arithmetic.

2.2 Sequential DCT-Based JPEG

In the sequential DCT-based mode the image is compressed in a single scan. All the 8×8 DCT blocks in an image are entropy encoded in a raster scan order. Two statistical models are used in the DCT-based sequential mode. One applies to the coding of DC differences generated by the DPCM model and the other applies to the coding of AC coefficients. The DC coefficient of the 8×8 block is always coded first. The DC is coded using a DPCM model in which the prediction is the DC coefficient of the most recently coded block. Table 2.1 shows the encoding process of the DC coefficients. The difference between the DC value and the predicted value is fed to the entropy coder. The Huffman

DC values	8	9	8	-6	-8	-3	3	3	..
DPCM	0	1	-1	-14	-2	5	6	0	..
Size	0	1	1	4	2	3	3	0	..
Appended bits	-	1	0	0001	01	101	110	-	..

Table 2.1: Encoding of DC coefficients using DPCM model

statistical model for coding of DPCM differences segments the difference values into a set of (approximately) logarithmically increasing magnitudes (shown as Size in Table 2.1) categories. Each of the difference categories is assigned a variable length (VL) code. For example, the DC values 9 and 8, and -3 and 3 in Table 2.1 have the same VL code since they are in the same magnitude categories.

Except for zero differences, however, the difference category VL codes do not fully describe the difference. Therefore, immediately following the VL codes for non-zero difference categories, additional bits are appended to the code stream to identify the sign and fully specify the magnitude of the difference. Positive differences are represented in binary representation using the minimum number of bits, and negative differences are represented by the complement of the positive numbers of the same magnitude. Therefore, the most significant bits of the positive and the negative differences will be always 0 and 1, respectively.

The AC coefficients are scanned in zigzag sequence order as shown in Figure 2.2. All of the DCT coefficients are then entropy encoded and output as part of the compressed image data. The encoding process for the AC coefficients is illustrated in Table 2.2.

In order to obtain coding efficiencies for the AC coefficients approaching the entropy, the VL coder aggregates zero coefficients into runs of zeros. However, even this strategy is not quite good enough, so the VL coder statistical model uses symbols that combine the run of zeros with magnitude categories for nonzero coefficients that terminate the

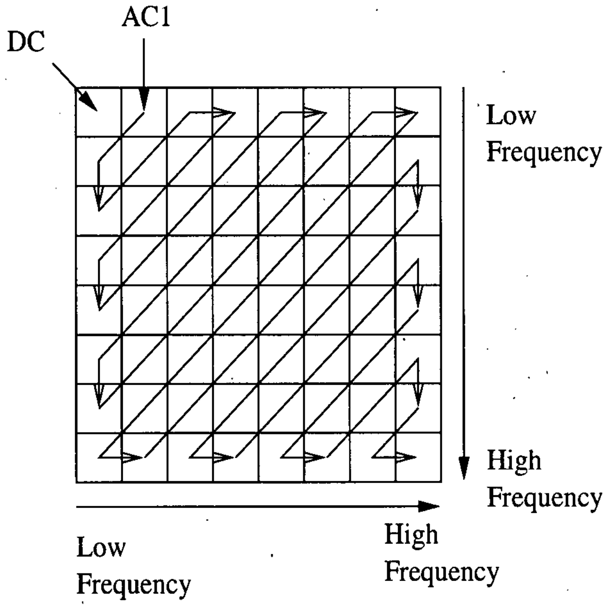


Figure 2.2: Encoding of AC coefficients in zigzag order in an 8 × 8 block .

AC index	1	2	3	4	5	6	7	8	9	10	..	62	63
AC values	0	0	0	0	-14	0	0	1	0	0	..	0	0
Zero run-length counter	1	2	3	4		1	2		1	2	..	54	EOB
Size					4			1					
Appended bits					0001			1					

Table 2.2: Encoding of AC coefficients

runs [2]. These magnitude categories increase logarithmically in exactly the same way as the DPCM difference categories, and the coding strategy is therefore very similar. For example, the first five AC coefficients in Table 2.2 are encoded by a VL code representing four runs of zeros and the size four. These codes are followed by appended bits that fully specify the sign and magnitude of the nonzero coefficient in exactly the same way as in the DC encoding process. Consequently, except for very long runs of zeros, VL codes are assigned to all possible combinations of runs of zeros and magnitude categories.

Because high frequency AC coefficients are mostly small in magnitude, they are mapped to zero after quantization. This is directly incorporated into the AC statistical model by means of a special symbol called the end-of-block (EOB). EOB means that the rest of the coefficients in the block are zero. The last 54 AC coefficients in Table 2.2 are encoded using EOB codes.

There are two types of sequential DCT-based JPEG. A simpler form of the sequential DCT-based mode of operation is called the "baseline JPEG." It represents a minimum capability that must be present in all DCT-based JPEG coders [12]. Only 8-bit input precision and two sets of Huffman tables (each set includes one DC table and one AC table) for entropy coding are allowed in baseline JPEG. A sequential DCT-based JPEG that has capabilities beyond the baseline JPEG is called "extended sequential JPEG," which allows 12-bit input precision and 4 sets of Huffman tables.

The JPEG standard includes a set of "typical" Huffman and quantization tables based on the results of several experiments using many images. While these tables are used in many implementations and yield reasonable performance, an image can be scanned prior to quantization and entropy encoding to generate image-specific quantization and optimized VL coding tables, respectively. The implementations using these optimal tables (although more complex) result in further compression performance gains, and are called "optimized JPEG" implementations [2, 12].

2.3 Progressive DCT-Based JPEG

In the progressive JPEG (PJPEG) mode, 8×8 blocks are formed in the same order, DCT transformed and (optionally) quantized to a specific number of bits. The quantized DCT coefficients are then partially encoded in multiple scans, each of which corresponds to a few bits of one or more DCT coefficients, and therefore represents a portion of the image being encoded/decoded. The DC coefficients are the average values of the 8×8 blocks, and an image containing uniform gray-level blocks is obtained when just the DC coefficients are decoded. The AC coefficients in the first few rows and columns of the 8×8 coefficients represent well the vertical and horizontal edges in the image, respectively. The high frequency AC coefficients represent fine, often non-existent, details of an image. The DCT coefficients, each of which contributing differently to improving the image quality, can be divided into many groups or decomposed into bits as specified in the PJPEG mode, such that the reconstructed image quality increases as more groups or bits of the DCT coefficients are received and decoded.

Two different procedures are defined in PJPEG by which the quantized coefficients may be partially encoded within a scan. In one procedure, called spectral selection, the DCT coefficients from the zigzag sequence in each 8×8 block are segmented into frequency bands of variable lengths. Each frequency band is then encoded in one scan. In the other procedure, called successive approximation, the DCT coefficients need not be encoded to their full accuracy in each scan. The precision of the DCT coefficients of the 8×8 blocks is reduced by dividing them by a power of two (or shifting right their binary representations) before encoding such coefficients. The received DCT coefficients are decoded and multiplied by the same power of two (or shifting left their binary representations) before the IDCT operation. The precision of the DCT coefficients is increased by one bit in each of the subsequent scans. These two procedures can be used

DC values	7	8	10	-5	7	-9	5	0	..
After point transform (>>1)	3	4	5	-3	3	-5	2	0	..
DPCM	3	1	1	-8	6	-8	7	-2	..
Size	2	1	1	4	3	4	3	2	..
Appended bits	11	1	1	0111	110	0111	111	01	..

Table 2.3: First DC scan (first 9 bits) using successive approximation

independently or combined in a flexible manner to yield many combinations [1, 2, 4]. Since they involve essentially the same operation, spectral selection and the first scan in successive approximation are discussed together in the next subsection. This is followed by a detailed description of the subsequent successive approximation scans.

2.3.1 Spectral Selection and Successive Approximation First Scan

The encoding methods for spectral selection and for the first scan of successive approximation are the same as that of the sequential JPEG mode, except that in PJPEG, the DC and AC coefficients are always encoded separately in different scans. Moreover, the end-of-block marker used in the sequential mode is extended to become a series of end-of-band markers in the PJPEG mode. When encoding the first DC scan, the precision of all the DC coefficients is reduced by a point-transform (a right-shift). Table 2.3 illustrates the encoding process of the first DC scan, where the precision of the DC coefficients is reduced by one bit. DPCM differences are then obtained from the DC values of reduced magnitude, and the minimum number of bits required to represent the difference is found (Size in Table 2.3) and variable length encoded, followed by the actual difference in binary representation. Negative differences are represented in the same way as in the sequential JPEG mode.

The first AC scan is also encoded in a similar manner as in the sequential mode.

AC index	1	2	3	4	5	6	7	8	9	10	..	62	63
AC values	0	1	7	-5	2	0	-4	0	1	0	..	0	0
After point transform (>>1)	0	0	3	-2	1	0	-2	0	0	0	..	0	0
Zero run-length counter	1	2	0	0	0	1	0	EOB0					
Size	-	-	2	2	1	-	2						
Appended bits	-	-	11	01	1	-	01						

Table 2.4: First AC scan (first 9 bits) using successive approximation

DC values	7	8	10	-5	7	-9	5	0	..
Previously coded ×2	6	8	10	-6	6	-10	4	0	..
Appended bit (correction bit)	1	0	0	1	1	1	1	0	..

Table 2.5: Subsequent DC scan (last bit) using successive approximation

The precision is reduced by a point-transform (division by a power of two). Table 2.4 illustrates the procedure in the case where the precision of the AC coefficients is reduced by 1 bit. Consecutive coefficients of zero magnitude are counted until a nonzero coefficient is encountered, and the minimum number of bits required to represent the nonzero coefficient is obtained. The combination of the run of zeros and the magnitude of the nonzero coefficient is then variable length encoded, followed by the actual coefficient in binary form. Negative numbers are handled in the same way as when encoding the DC scan.

2.3.2 Successive Approximation Subsequent Scans

Both of the subsequent DC and AC scans in successive approximation increase the precision of the DCT coefficients by one bit at a time. In the subsequent DC scans, the encoding method is rather simple: the codewords are simply the bit values themselves as shown in Table 2.5. In the subsequent AC scans, the bits are coded in three different

AC index	1	2	3	4	5	6	7	8	9	10	..	62	63
AC values	0	1	7	-5	2	0	-4	0	1	0	..	0	0
Previously coded ×2	0	0	6	-4	2	0	-4	0	0	0	..	0	0
Zero run-length counter	1		0	0	0	1	0	1	-	EOB0			
Size	-	1	-	-	-	-	-	-	1				
Appended bits (correction bits)	-	-	1	1	0	-	0	-	-				

Table 2.6: Subsequent AC scan (last bit) using successive approximation

ways: partial run-length coding, nonzero coding, and appending of correction bits. Table 2.6 illustrates the encoding procedure of the subsequent AC scans. The **0**s are counted until a new **1** is encountered. A bit can have a new value of **1** when it converts the decoded value in the previous scans from zero to a nonzero value. The run-length and the size (equal to 1) of the new **1** bit is variable length encoded, followed by a sign bit (**0** for negative and **1** for positive). The **0**s to be added to the nonzero decoded values from the previous scans are skipped when counting the **0**s, i.e., the run-length coding of zeros is partial. Such **0**s, as well as the nonzero bits to be added to the nonzero decoded value, are appended to the variable length codeword and sign bit.

Based on the encoding process described above, it is clear that PJPEG imposes a few constraints on the progressive encoder's structure. Some are generic and are identical to those imposed on the sequential JPEG encoder's structure, and others are imposed mainly to reduce the complexity of PJPEG compliant decoders. Such constraints, which must be satisfied by our proposed PJPEG design algorithm, are stated below:

- The first scan to be encoded/decoded must contain only DC information. Such a scan represents all DC coefficients in spectral selection, and it represents a group of the most significant bits of the DC coefficients in successive approximation.

- The DC and AC coefficients cannot be encoded in the same scan.

- The coefficients in the same spectral band must be consecutive, i.e., their locations can be described by just two points: 1) start of band and 2) end of band.
- Only the first scan of any DCT coefficient can represent more than one precision bit, and the subsequent scans must represent only one precision bit at a time.
- Encoding/decoding of a bit of a coefficient requires that the preceding bits of that coefficient be encoded/decoded in advance.

2.4 Hierarchical JPEG

HJPEG provides the other JPEG modes of operations with a high level structure for a progressive encoding of an image at multiple spatial resolutions. Between progressive stages, spatial resolution of the reconstructed image increases. Figure 2.3 shows a block diagram of HJPEG.

Before any encoding is done, the original image is downsampled by a desired number of multiples of 2 in each dimension. The first stage (lowest resolution image) can then be encoded using one of the progressive, sequential, or lossless JPEG modes of operations. This encoded image is transmitted to the receiver, but, at the same time, is decoded, upsampled and lowpass filtered to become the prediction for the second stage. In the second stage, the corresponding difference image is encoded and transmitted. Again, at the encoder, this difference image is decoded, added to the predicted image from the first stage, upsampled and lowpass filtered to become the prediction for the third stage. Likewise, the difference image obtained in the third stage is encoded and transmitted. After each stage, the receiver is able to progressively reconstruct the image with increasing spatial resolution following the same upsampling and filtering procedure used at the encoder.

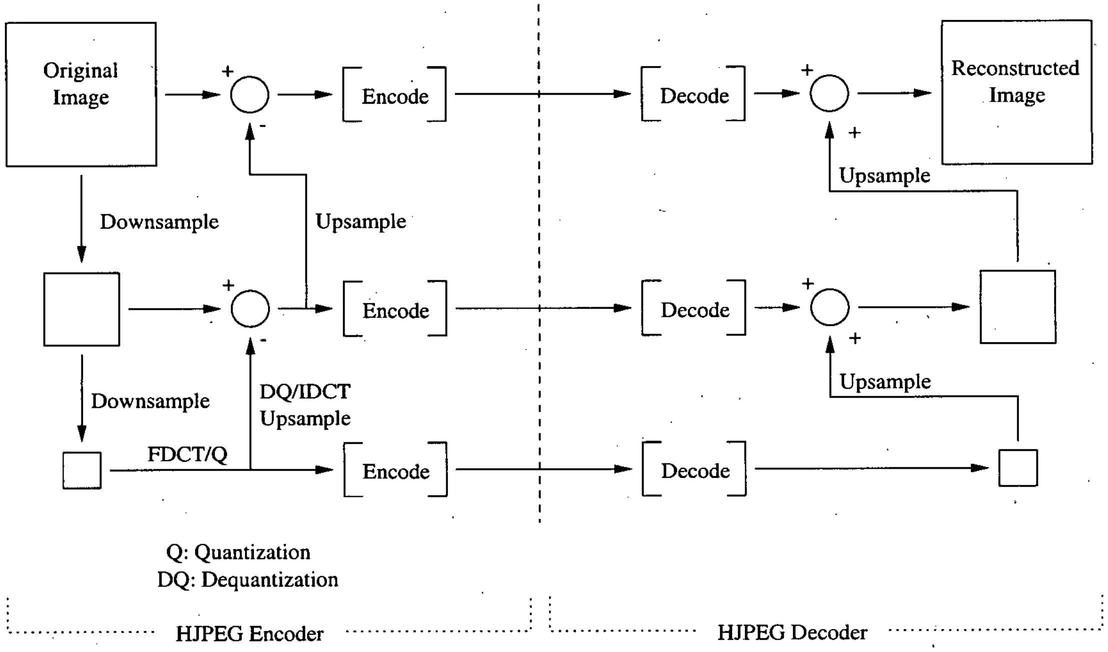


Figure 2.3: HJPEG encoding/decoding procedure. One of the progressive, sequential, or lossless encoding modes can be used in the [Encode] steps, and corresponding decoding modes can be used in the [Decode] steps.

Chapter 3

Rate Distortion Optimized PJPEG

Understanding the relationship between the DCT coefficients and the reconstruction image quality can help us greatly design an efficient and effective PJPEG compliant encoder. For example, as discussed in Section 2.3, only the first few rows and columns of the 8×8 blocks can be transmitted for the images whose content involve mostly vertical and horizontal edges, respectively. However, ordering of 64 DCT coefficients in an 8×8 block according to subjective importance is a very difficult task, and such algorithm requires information about the image content to be available prior to the compression process. Moreover, the bit rate usually supported by typical low-bandwidth channels is generally so low that only the first few most significant bits of a single DCT coefficient in 8×8 blocks can be transmitted. To address the above problems, all the bits of the 64 DCT coefficients in an 8×8 block are here assigned importance levels that involve the size (i.e., bit rate) of the resulting scan and the scan's contribution to reducing the reconstruction distortion, expressed in terms of the sum of squared errors. More specifically, each bit in a DCT coefficient is assigned a distortion-rate ratio, whose numerator is the reduction in reconstruction distortion, and denominator is the scan's size in bytes. The resulting value is then used to associate with each bit a priority level. Taking into account PJPEG's general encoder structure and specific constraints, the prioritized bits are then grouped, achieving higher compression efficiency and lower computational requirements. Details about each of the above steps are presented next.

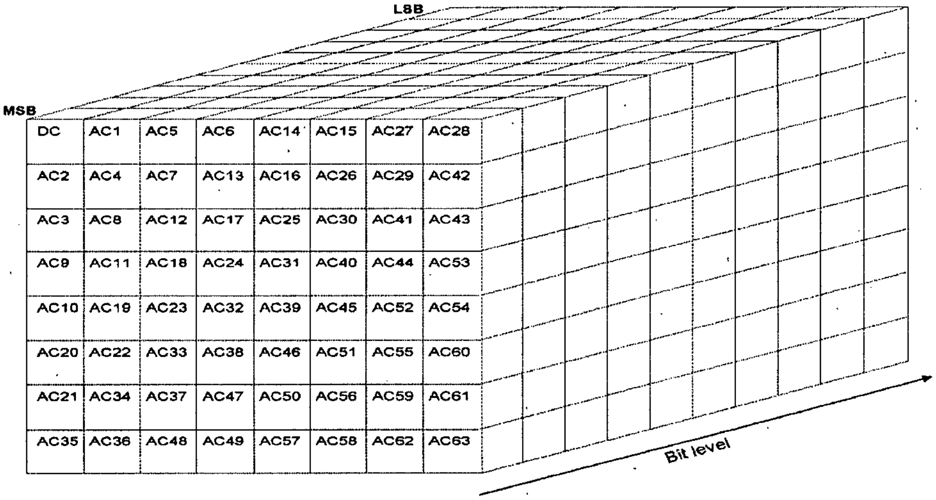

LSB
MSB
DC AC1 AC5 AC6 AC14 AC15 AC27 AC28
AC2 AC4 AC7 AC13 AC16 AC26 AC29 AC42
AC3 AC8 AC12 AC17 AC25 AC30 AC41 AC43
AC9 AC11 AC18 AC24 AC31 AC40 AC44 AC53
AC10 AC19 AC23 AC32 AC39 AC45 AC52 AC54
AC20 AC22 AC33 AC38 AC46 AC51 AC55 AC60
AC21 AC34 AC37 AC47 AC50 AC56 AC59 AC61
AC35 AC36 AC48 AC49 AC57 AC58 AC62 AC63
Bit level

Figure 3.4: An $8 \times 8 \times 10$ parallelepiped.

3.1 Prioritization of Bits

Assuming 8-bit input precision, each DCT coefficient will have 11-bit precision, 10 bits for magnitude and one bit for sign[1]. The sign bit is not considered separately when prioritizing bits since it is always encoded with the first non-zero most significant bit as explained in Section 2.3. Excluding the sign bit for the moment, an 8×8 DCT coefficient block can be considered as a parallelepiped which contains $8 \times 8 \times 10$ bits. Figure 3.4 shows an $8 \times 8 \times 10$ parallelepiped. The decoder assumes parallelepipeds of all **0**s corresponding to 8×8 blocks of zero-valued coefficients. The **0**s are replaced by the actual values of the bits as they are decoded. Therefore, to quantify the effect of each bit on the quality of the decoded image, the distortion associated with each bit being considered **0** is computed based on the fact that more significant bits are more important

[1] JPEG also allows 12-bit input precision as described in Chapter 2. Our algorithm can be easily extended to support 12-bit input precision since the input precision is simply a parameter that can take on many values.

than less significant bits. For example, if the tenth bit (left-most or most significant bit) of the DC coefficient is equal to **1** in value, and the decoder assumes it to be **0**, the resulting distortion[2] (or squared error) will be equal to $(2^9)^2$. When the value of a bit is **0**, its associated distortion will be equal to zero regardless of its position, as the decoder has already assumed all the bits to have a **0** value, and decoding a **0** bit will not reduce the reconstruction distortion. In the PJPEG mode of operation, each bit of a parallelepiped should be transmitted along with the bits at the same position of all the parallelepipeds in an image. Thus, the overall distortion reduction value should be equal to the sum of the distortion reduction values associated with the bits at the same position in all the parallelepipeds in the image, that is, the overall distortion reduction ΔD_i associated with the ith bit should be equal to

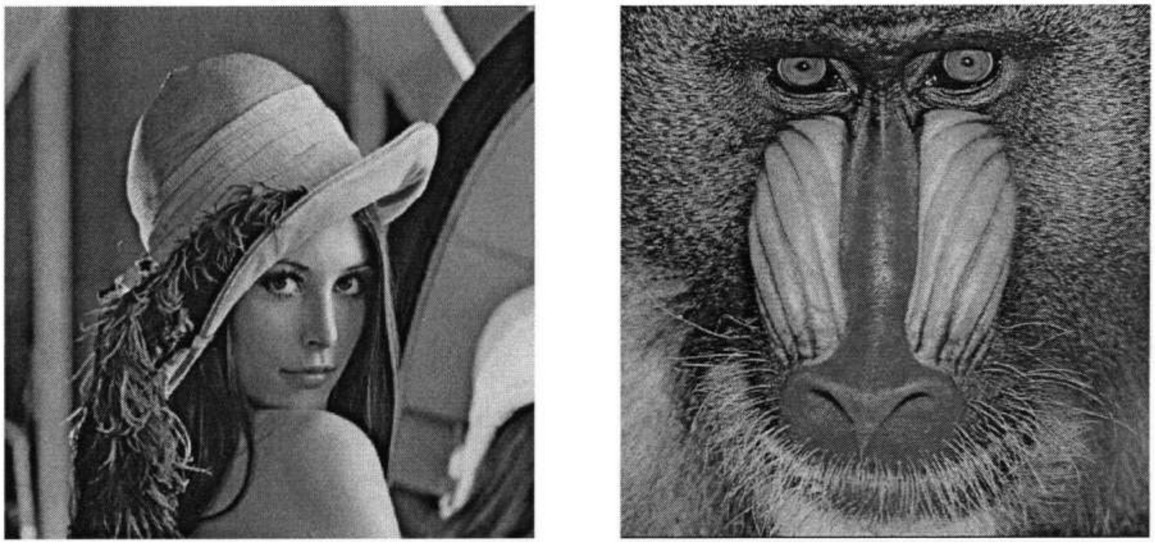
$$D_i = \sum_{j=1}^{N} (b_{i,j} 2^{p_i - 1})^2, \tag{3.3}$$

where $i = 1, 2, ..., 640$, N is the number of 8×8 blocks in the input image, $b_{i,j}$ is the value of the ith bit of the jth parallelepiped and p_i is the position of the ith bit. Encoding of bits at different locations in the parallelepipeds also results in different scan sizes or bit rates. Thus, a rate-distortion optimized importance measure should involve the total distortion and bit rate values, and an appropriate importance measure I_i is then given by

$$I_i = \frac{\Delta D_i}{\Delta R_i}, \tag{3.4}$$

where I_i, ΔD_i and ΔR_i are the priority value, distortion reduction and scan size (or bit rate increase) associated with the ith bit, respectively. In this manner, the bit that contributes the most distortion reduction while requiring the least number of bits (smallest scan size) is assigned the greatest priority value. Table 3.7 shows the priority values of

[2] Since the DCT is a unitary transform, the distortion in the DCT domain is theoretically equal to that in the spatial domain.



(a)

(b)

Figure 3.5: Original images: (a) 512×512 image LENA and (b) 512×512 image MANDRIL.

all the bits of the first 30 DCT coefficients (obtained by tracing the 64 DCT coefficients in the normal JPEG zig-zag way) of the 512×512 image LENA shown in Figure 3.5 (a). Throughout the chapter, the popular standard image LENA will be used whenever subjective quality illustrations are provided. However, in the last section, our algorithm will also be applied to the image MANDRIL[3] shown in Figure 3.5 (b).

As expected, the less significant bits, especially those of coefficients occuring at high frequencies, are assigned small priority values. Such bits contribute only high frequency or noise-like information, which generally reduce only slightly the objective/subjective distortion. Based on the priority value defined in equation (3.4), all the 640 bits in an 8×8 block can be sorted in terms of decreasing order of priority values. Figure 3.6 shows the priority values of the 20 most important bits, for the 512×512 image LENA. Notice that the priority values associated with the first few bits are very large, as compared to

[3] Such an image is known to contain much more statistically random information than the other standard images.

DCT index	MSB									LSB
0	270719	136142	55399	15099	3874	966	246	62	16	4
1	6899	44227	16384	5163	1626	463	127	36	10	3
2	0	21845	13749	4233	1392	417	117	32	9	3
3	0	0	4029	2538	1039	318	92	27	8	2
4	0	4795	9732	3857	1241	384	106	29	8	2
5	0	5958	10377	3941	1272	373	111	31	8	2
6	0	1725	3218	2658	1020	325	92	28	8	2
7	0	0	5532	2983	1093	322	96	27	8	2
8	0	0	2731	2933	1009	309	95	26	8	2
9	0	0	0	1154	724	245	79	24	7	2
10	0	0	0	286	337	186	65	22	7	2
11	0	0	0	1069	706	267	81	24	7	2
12	0	0	2249	2464	1024	300	91	26	7	2
13	0	0	1707	2268	969	297	90	25	7	2
14	0	0	431	1682	848	273	81	24	7	2
15	0	0	0	972	595	222	72	23	7	2
16	0	0	0	793	691	256	78	23	7	2
17	0	0	0	1601	820	268	78	25	7	2
18	0	0	0	991	711	269	79	24	7	2
19	0	0	0	0	256	223	65	22	7	2
20	0	0	0	0	89	118	56	20	6	2
21	0	0	0	0	0	63	44	18	6	2
22	0	0	0	0	91	121	57	20	6	2
23	0	0	0	0	293	209	74	21	7	2
24	0	0	0	512	613	249	76	23	7	2
25	0	0	0	562	603	226	74	22	7	2
26	0	0	0	286	451	217	73	21	7	2
27	0	0	0	108	394	179	64	22	6	2
28	0	0	0	0	120	137	59	21	7	2
29	0	0	0	205	218	164	64	21	7	2

Table 3.7: Priority values of the bits of the coefficients of the 512 × 512 image LENA.

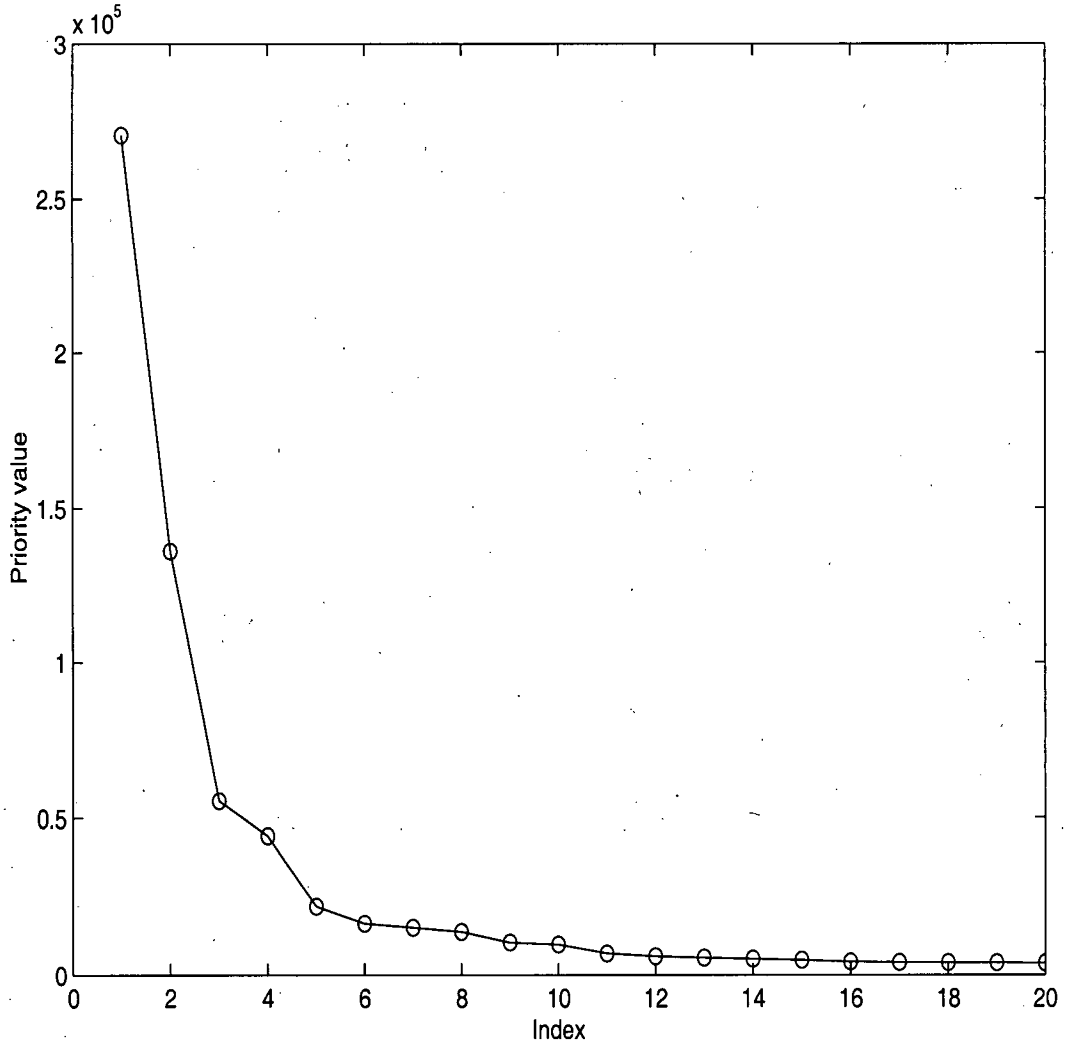


Figure 3.6: First 20 bits in descending order of priority level for the image LENA.

those of the other bits. Such a feature is well-suited for low-bandwidth channels, such as those used in wireless applications. By transmitting the bits according to the proposed prioritization method, acceptable reproduction quality is quickly obtained after receiving the first $1-4$ bits. Unfortunately, however, the order obtained above cannot be used as-is if PJPEG compliance is to be maintained. As stated in Section 2.3.2, for a specific bit of a DCT coefficient to be encoded/decoded, PJPEG requires that all the more significant (in terms of bit position in the binary representation) bits of the subject coefficient be encoded/decoded first. Figure 3.7 shows the first 20 bits to be transmitted satisfying the constraint. As expected, the graph in the figure is not as smooth as the one in Figure 3.6.

3.2 Grouping of Bits

Encoding bits according to their priority values and the above PJPEG constraint is still far from "optimal", as such a method does not take into account the PJPEG header and encoding structures. In PJPEG, a header is assigned to each scan, carrying information such as the number of image components. If each scan contained a single bit plane, the additional header information would be excessive, offsetting most of the compression gain that would have been obtained using such a method. Moreover, encoding/decoding 640 bit planes is not necessarily the most compression efficient method in light of the specific coding methods used in PJPEG. For example, encoding the first DC bit following the PJPEG method requires at least one bit per pixel, leading to expansion instead of compression, i.e., subsequent DC bits are coded as-is, and extra header information is added to each scan. Another example is PJPEG's across-block run-length method, which becomes very inefficient when the high frequency AC bits are encoded. Fortunately, PJPEG allows a group of consecutive most significant bits of a coefficient and/or bits at

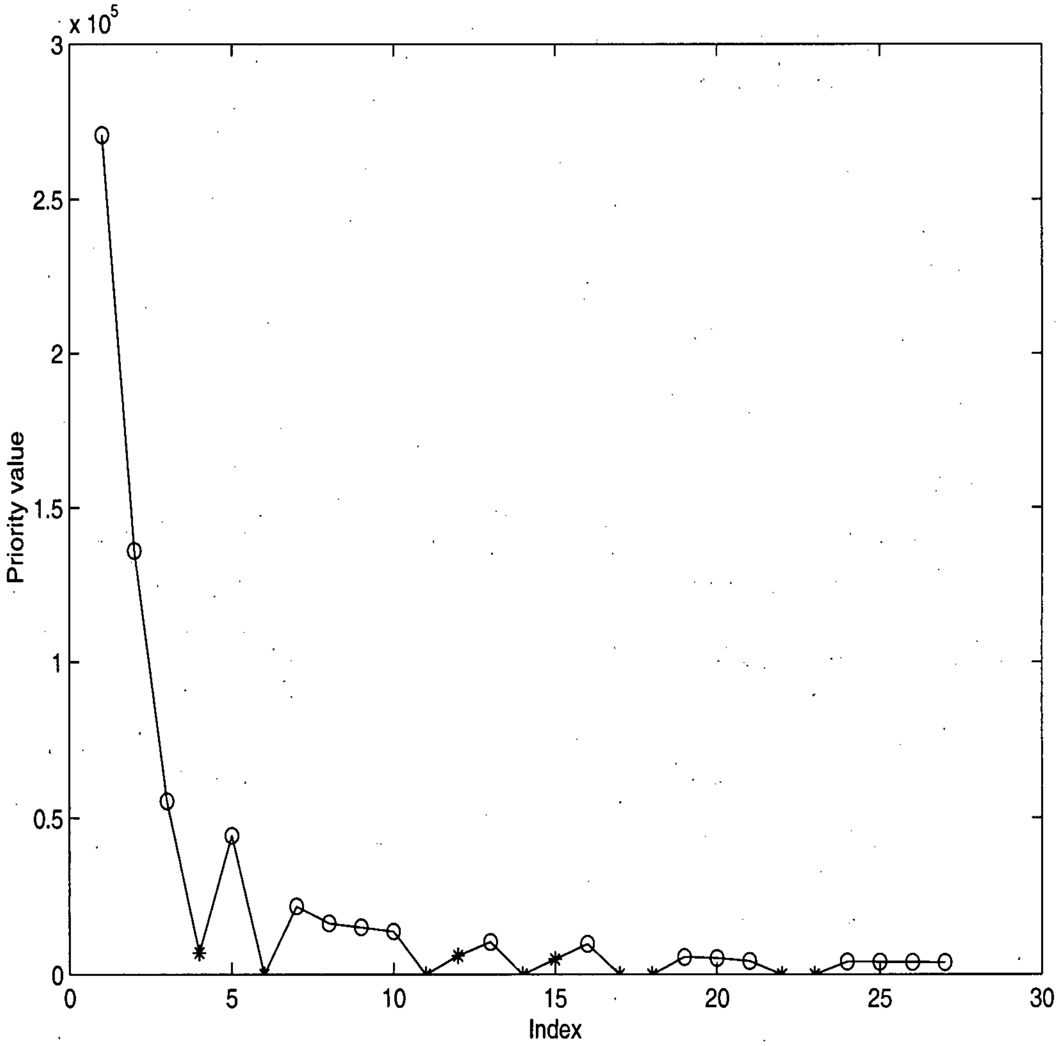


Figure 3.7: First 20 bits in descending order of priority levels for the image LENA where the last PJPEG constraint listed in Section 2.3.2 is satisfied.

the same position of consecutive coefficients to be encoded in a single scan. Considering the fact that the header size is essentially the same regardless of how many bits are represented in a single scan, grouping bits will reduce substantially the number of header bits. As will be shown later, grouping of bits may also result in improved compression efficiency of PJPEG's across-block run-length coder.

To determine which bits should be grouped together, we begin by grouping only consecutive bits in the same DC or AC coefficient. First, it is better (in terms of compression efficiency) to combine all the bits of the DC coefficient into one scan. This is partly because 1) the first group of bits is encoded as in sequential JPEG, and 2) all subsequent scans must represent only one bit, which is sent as-is for each 8×8 block. Another reason is that only a single header is needed when one scan represents all the bits of the DC coefficients. The optimal number of bits to be grouped can be determined experimentally. Figure 3.8 shows the distortion reduction vs. rate for 4 cases: no grouping and groupings of 2, 3, and 4 most significant bits of the DC coefficient for the image LENA. When no bits are grouped, there are 10 scans, each scan representing a bit. When the first 2, 3, and 4 first bits are grouped into one scan, we obtain 9, 8, and 7 scans, respectively. Clearly, the amount of reduction in distortion increases (for the same bit rate) as more and more bits are grouped, but the rate of increase levels off after grouping $2-3$ bits. Figure 3.9 shows the results when the first 2, 3, and 4 first bits of the first AC coefficient are grouped into one scan. In this case also, the rate of increase of distortion reduction levels off after grouping $2-3$ bits. The number of bits that should be grouped in a scan appears to be increasing as a function of the AC position along the zig-zag direction.

Determining the "best" grouping method for the DCT bits, along the direction from the most to the least significant bit, is not difficult since there are only 10 possible different cases for each coefficient. However, by combining successive approximation with spectral selection, grouping AC bits can then be performed in two directions simultaneously, going

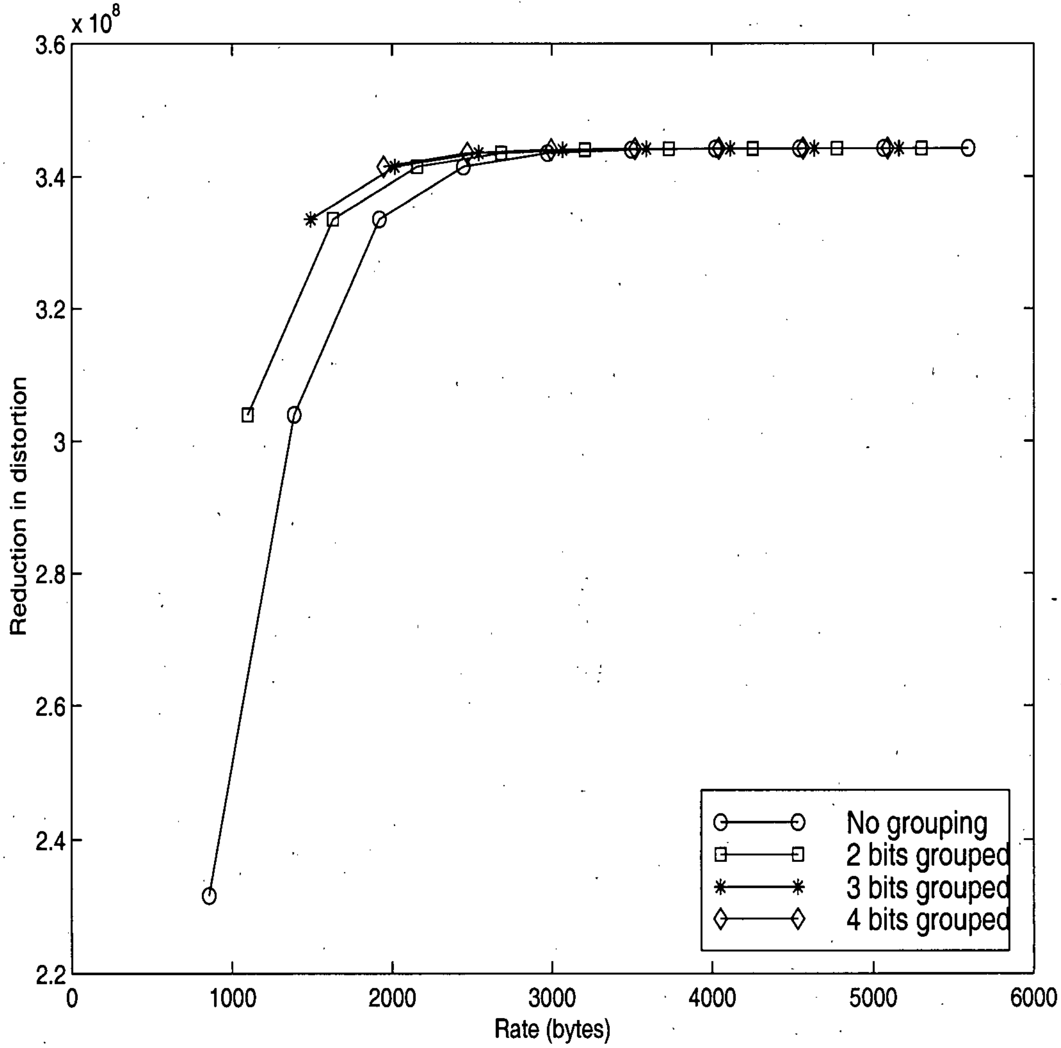


Figure 3.8: Reduction in distortion vs. Rate for the first 4 groupings of DC coefficient using successive approximation for the image LENA.

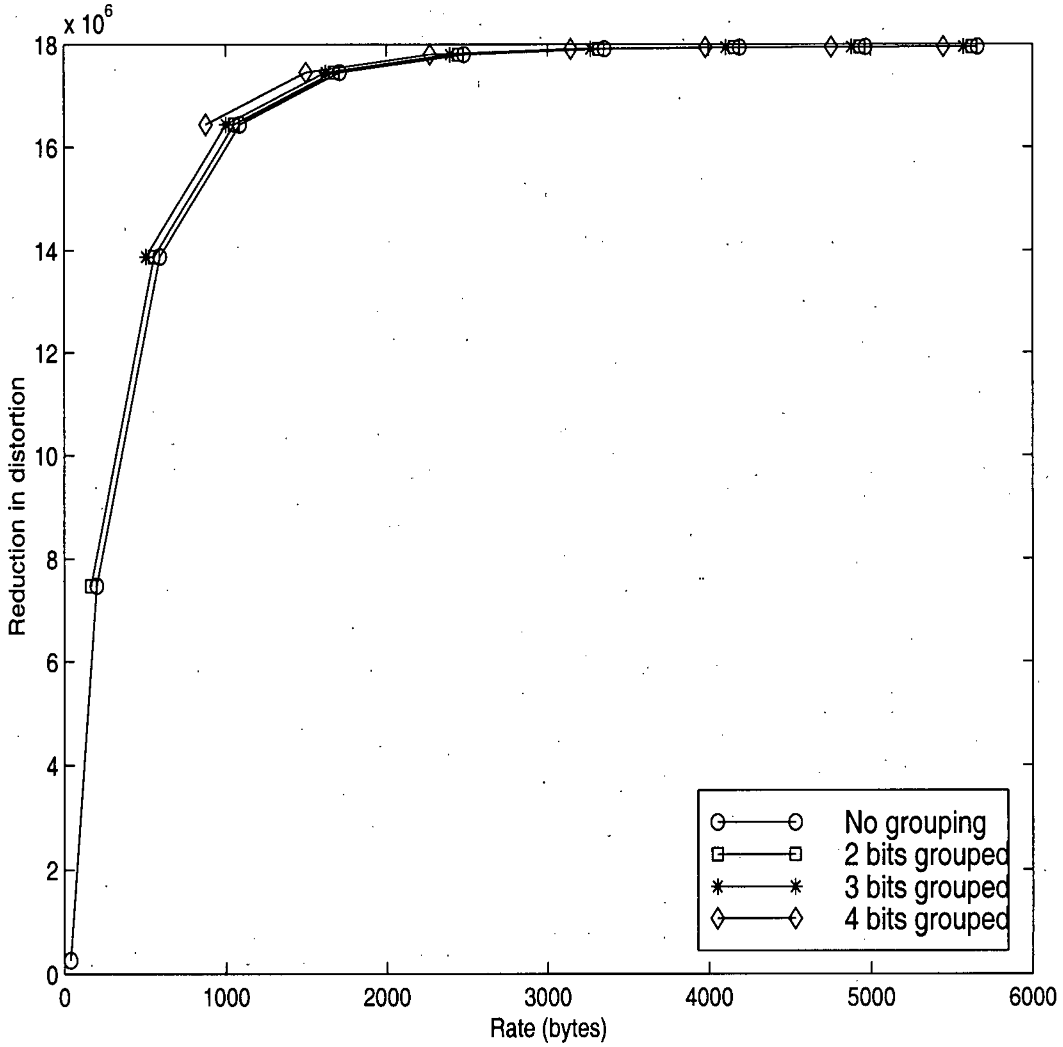


Figure 3.9: Reduction in distortion vs. Rate for groupings of the first AC coefficient for the image LENA.

from the most to the least significant bit and in the zig-zag direction from the first AC coefficient to the highest frequency AC coefficient. Obviously, the number of possible groupings is very large. On the one hand, one can construct two groups, yielding a DC scan and an AC scan. Such a procedure is almost equivalent to that of the sequential JPEG mode. On the other hand, one can construct 640 scans representing 10 DC bits and 630 AC bits. Clearly, a design method exists, and is here developed by imposing some additional constraints, that yields good tradeoffs between compression efficiency and progressiveness resolution.

First, the analysis described above, using the priority values and taking into account the last PJPEG syntax constraint listed in Section 2.3.2, is applied to group the most significant bits of the DC and AC coefficients separately. The results of the first step are shown in Table 3.8 for the 512 × 512 image LENA. Second, grouped bits of neighboring (in the zig-zag direction) AC coefficients having similar grouping structures and priority values can be further combined to form spectral bands, as shown in Table 3.9 for the same image. Finally, subsequent bits of the AC coefficients, located at the same bit position level and having similar priority values, can also be combined to yield larger groups of bits, as shown in Table 3.10. Note the last two steps are expected to improve compression efficiency, since 1) combining groups with similar priority values would not theoretically decrease rate-distortion performance significantly, 2) many of the significant bits of the AC coefficients typically have zero values, improving across-block run-length coding efficiency, and 3) grouping many bits into a single scan would reduce the number of required header bits. Looking at Tables 3.8 and 3.10, it is clear that the priority values increase monotonically as more bits are grouped, indicating improved compression efficiency. In summary, the overall PJPEG grouping algorithm is given below:

1. Compute the priority values of all 640 bits in the parallelepiped using equation 3.4.

DCT index	MSB							LSB
0 (DC)		15099	3874	966	246	62	16	4
1		5163	1626	463	127	36	10	3
2		4233	1392	417	117	32	9	3
3		2538	1039	318	92	27	8	2
4		3857	1241	384	106	29	8	2
5		3941	1272	373	111	31	8	2
6		2658	1020	325	92	28	8	2
7		2983	1093	322	96	27	8	2
8			1009	309	95	26	8	2
9			724	245	79	24	7	2
10			337	186	65	22	7	2
11			706	267	81	24	7	2
12			1024	300	91	26	7	2
13			969	297	90	25	7	2
14			848	273	81	24	7	2
15			595	222	72	23	7	2
16			691	256	78	23	7	2
17			820	268	78	25	7	2
18			711	269	79	24	7	2
19			256	223	65	22	7	2
20			89	118	56	20	6	2
21			0	63	44	18	6	2
22			91	121	57	20	6	2
23				209	74	21	7	2
24				249	76	23	7	2
25				226	74	22	7	2
26				217	73	21	7	2
27				179	64	22	6	2
28				137	59	21	7	2
29				164	64	21	7	2
30				200	63	21	7	2
31				212	69	21	7	2
32				215	70	21	7	2
33					62	20	6	2
34					45	19	6	2
35					35	18	6	2
36					34	17	6	2
37					43	17	6	2
38					59	20	6	2
39					59	19	6	2
40					60	20	7	2
41					58	21	6	2
42					55	19	6	2
43					51	19	6	2
44					51	19	6	2
45					54	19	6	2
46					57	19	6	2
47					44	18	6	2
48						17	6	2
49						16	6	2
.		.				.	.	.
.		.				.	.	.
.		.				.	.	.
62						15	6	2
63						15	2	2

Table 3.8: The first step of the grouping procedure for the image LENA. The most significant bits of DC and AC coefficients are grouped separately.

DCT index	MSB							LSB
0 (DC)	223695	15099	3874	966	246	62	16	4
1	24016	5163	1626	463	127	36	10	3
2		4233	1392	417	117	32	9	3
3		2538	1039	318	92	27	8	2
4		3857	1241	384	106	29	8	2
5		3941	1272	373	111	31	8	2
6		2658	1020	325	92	28	8	2
7		2983	1093	322	96	27	8	2
8	3867		1009	309	95	26	8	2
9			724	245	79	24	7	2
10			337	186	65	22	7	2
11			706	267	81	24	7	2
12			1024	300	91	26	7	2
13			969	297	90	25	7	2
14			848	273	81	24	7	2
15			595	222	72	23	7	2
16			691	256	78	23	7	2
17			820	268	78	25	7	2
18			711	269	79	24	7	2
19			256	223	65	22	7	2
20			89	118	56	20	6	2
21			0	63	44	18	6	2
22			91	121	57	20	6	2
23	940			209	74	21	7	2
24				249	76	23	7	2
25				226	74	22	7	2
26				217	73	21	7	2
27				179	64	22	6	2
28				137	59	21	7	2
29				164	64	21	7	2
30				200	63	21	7	2
31				212	69	21	7	2
32				215	70	21	7	2
33	251				62	20	6	2
34					45	19	6	2
35					35	18	6	2
36					34	17	6	2
37					43	17	6	2
38					59	20	6	2
39					59	19	6	2
40					60	20	7	2
41					58	21	6	2
42					55	19	6	2
43					51	19	6	2
44					51	19	6	2
45					54	19	6	2
46					57	19	6	2
47					44	18	6	2
48	70					17	6	2
49						16	6	2
.						.	.	.
.						.	.	.
.						.	.	.
62						15	6	2
63						15	2	2

Table 3.9: The second step of the grouping procedure for the image LENA. Grouped bits of neighboring AC coefficients with similar grouping structure are combined to form spectral bands.

<table>
<tr><th>DCT index</th><th colspan="7">MSB</th><th>LSB</th></tr>
<tr><td>0</td><td>223695</td><td>15099</td><td>3874</td><td>966</td><td>246</td><td>62</td><td>16</td><td>4</td></tr>
<tr><td>1</td><td rowspan="7">24016</td><td rowspan="7">5236</td><td rowspan="22">1566</td><td rowspan="32">426</td><td rowspan="47">110</td><td rowspan="52">28</td><td rowspan="52">9</td><td rowspan="52">3</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td></tr>
<tr><td>4</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td><td colspan="2" rowspan="15">3867</td></tr>
<tr><td>9</td></tr>
<tr><td>10</td></tr>
<tr><td>11</td></tr>
<tr><td>12</td></tr>
<tr><td>13</td></tr>
<tr><td>14</td></tr>
<tr><td>15</td></tr>
<tr><td>16</td></tr>
<tr><td>17</td></tr>
<tr><td>18</td></tr>
<tr><td>19</td></tr>
<tr><td>20</td></tr>
<tr><td>21</td></tr>
<tr><td>22</td></tr>
<tr><td>23</td><td colspan="3" rowspan="10">940</td></tr>
<tr><td>24</td></tr>
<tr><td>25</td></tr>
<tr><td>26</td></tr>
<tr><td>27</td></tr>
<tr><td>28</td></tr>
<tr><td>29</td></tr>
<tr><td>30</td></tr>
<tr><td>31</td></tr>
<tr><td>32</td></tr>
<tr><td>33</td><td colspan="4" rowspan="15">251</td></tr>
<tr><td>34</td></tr>
<tr><td>35</td></tr>
<tr><td>36</td></tr>
<tr><td>37</td></tr>
<tr><td>38</td></tr>
<tr><td>39</td></tr>
<tr><td>40</td></tr>
<tr><td>41</td></tr>
<tr><td>42</td></tr>
<tr><td>43</td></tr>
<tr><td>44</td></tr>
<tr><td>45</td></tr>
<tr><td>46</td></tr>
<tr><td>47</td></tr>
<tr><td>48</td><td colspan="5" rowspan="5">70</td></tr>
<tr><td>49</td></tr>
<tr><td>⋮</td></tr>
<tr><td>62</td></tr>
<tr><td>63</td></tr>
</table>

Table 3.10: The final step of the grouping procedure for the image LENA. Bits of the AC coefficients, located at the same bit position level and with similar priority values are combined to yield large group of bits.

High priority values are assigned to the bits reducing substantially the reconstruction distortion and costing only a few bits.

2. Obtain the best individual coefficient grouping structure, i.e. the structure yielding the best tradeoff between compression efficiency and progressiveness resolution.

3. Combine groups of bits of consecutive AC coefficients in the zig-zag direction.

4. Compute the new priority values using as distortion reduction the value obtained by computing the sum of the individual distortion reduction values of the constituent bit groups, and as cost the number of code bits required by PJPEG to represent the resulting scan.

5. Generate scans in the order of decreasing priority values.

The performance of our RD-optimized PJPEG encoder is presented in Section 3.4.

3.3 Complexity, Memory and Computation Requirements Vs. Robustness

In this section, we discuss the complexity, memory and computation requirements at both the encoder and decoder. As the proposed encoding algorithm is PJPEG compliant, decoding can be performed using existing hardware/software decoders, whose complexity, memory and computation demands are clearly larger than those of sequential JPEG decoders, but are arguably smaller than those of today's high performance subband/wavelet decoders [13].

Despite the fact that quantization can be avoided in PJPEG, sequential JPEG encoders are still simpler and less memory intensive. However, the computational demands of conventional PJPEG encoders are comparable to those of sequential JPEG encoders, and are substantially smaller than those of state-of-the-art subband/wavelet encoders.

For the benefit of better rate-distortion performance and precise rate/distortion control, the proposed encoder appears to be more complex and much more demanding in terms of computations. Making PJPEG useful in constant bit rate applications is worth the additional complexity, due mainly to the above rate control algorithm. However, the gain in compression performance may not be worth the seemingly two orders of magnitude increase in encoding computations.

The surge in computational demands by the encoder is due mainly to the computations required to estimate the priority values. Such computations are required to encode 640 scans (for 8-bit precision) and produce corresponding distortion and bit rate (scan size) values. Another source is the re-computation of priority values of newly grouped bits.

While all the computations can sometimes be performed off-line, a fast algorithm is still desired that can be implemented on-line. By applying the algorithm obtained in the previous section to a set of training images[4], a general grouping structure can be obtained. When such a general grouping structure is used, complexity, computational demands, and memory requirements of our algorithm reduce to levels comparable to those of conventional PJPEG encoders. As will be discussed in Section 3.4, our proposed algorithm is very robust with respect to changes in input image content and resolution. Thus, the above grouping structure yields only a slight loss in compression performance.

3.4 Experimental Results

In this section, we present experimental results that illustrate the performance of our RD optimized PJPEG compliant encoding algorithm. Figure 3.10 shows the PSNR for the PJPEG encoders based on our algorithm, a baseline JPEG coder, a well optimized

[4] Averaging of the grouping structures obtained separately is very difficult and often not systematic. In our approach, the algorithm is applied only once to a big combined image of a set of training images.

sequential JPEG coder, and the SPHIT coder [13], at medium and high bit rates for the 512×512 image LENA. As it is evident, our algorithm outperforms both the baseline and optimized sequential JPEG coders at high bit rates. At medium bit rates, however, the performance of the algorithms are comparable. Note that the sequential JPEG coders were designed for such medium bit rates. Moreover, notice that the wavelet SPHIT coder outperforms significantly (by 1-3 dB) the JPEG/PJPEG ones at most bit rates. Figure 3.11 shows the PSNR for the same coders at low bit rates. Below 0.2 bpp, the difference in performance between sequential JPEG encoders and our encoder is considerable. Some of the bit rates cannot even be achieved by sequential JPEG. Also illustrated in the figure is the fact that SPHIT is much more compression efficient than the other algorithms.

Figure 3.12 shows the PSNR for our PJPEG coder, sequential and optimized sequential JPEG, and SPHIT at medium and high bit rates for the 512×512 image MANDRIL. Again, the performance of our algorithm is better than that of the sequential JPEG. Figure 3.13 shows the same information at low bit rates. As in the previous case, the difference between sequential JPEG and our algorithm is considerable below 0.2 bpp. As expected, SPHIT-EZW algorithm, again, outperforms the JPEG/PJPEG coders, although by less than 1 dB at most bit rates.

Figures 3.14 and 3.16 show the 512×512 images LENA and MANDRIL coded at 0.075 bpp and 0.055 bpp,[5] respectively using an optimized sequential JPEG algorithm and our PJPEG algorithm. Clearly, the visual quality of the image coded using our algorithm is much better than the other one. Figures 3.15 and 3.17 show the same images coded at 0.1 bpp using the same optimized sequential JPEG and our PJPEG algorithm. The superior performance of our algorithm is clearly illustrated.

To demonstrate the robustness of the our grouping algorithm with respect to varying

[5] The two images were encoded at different bit rates since the sequential JPEG does not provide rate control. However, the images encoded using RD optimized PJPEG were obtained with precise rate control algorithm discussed in Chapter 4.

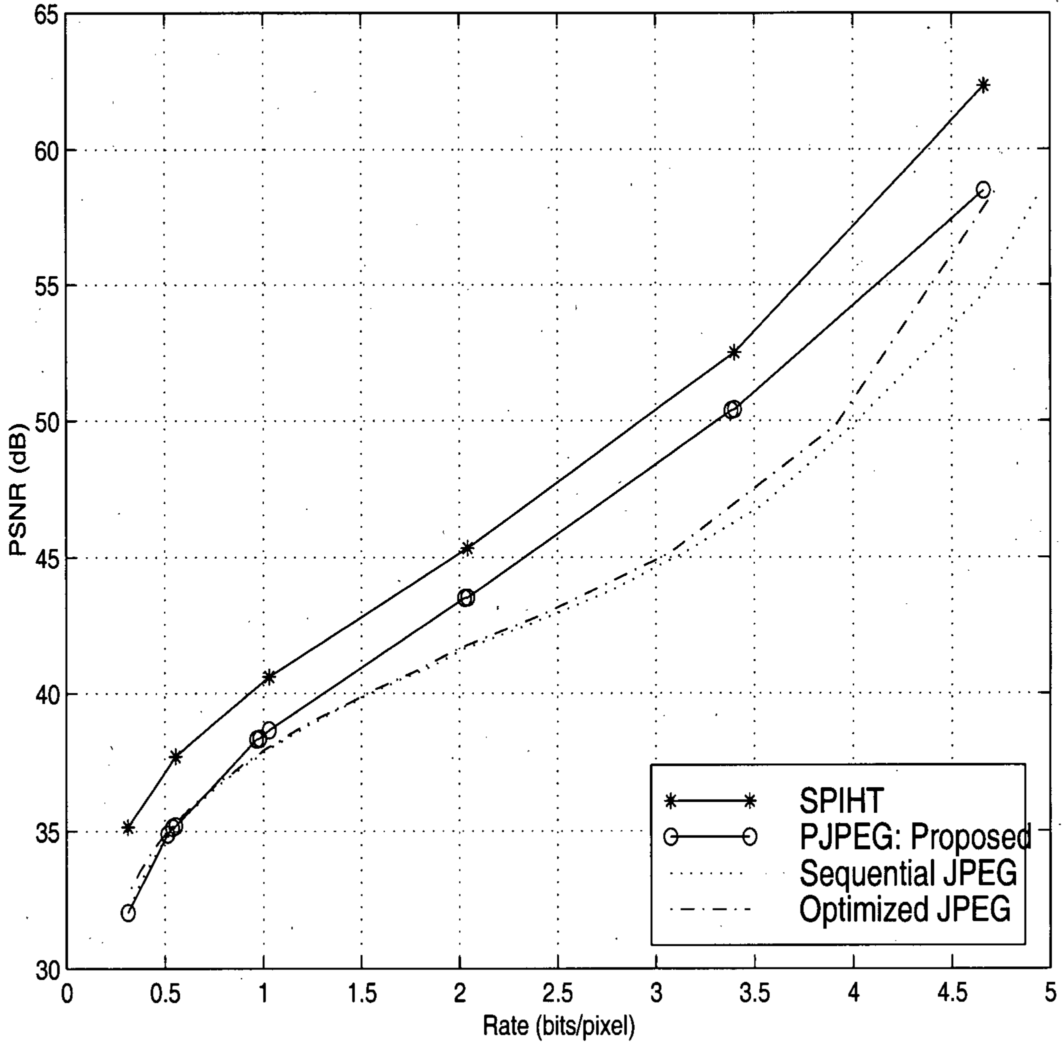


Figure 3.10: Comparison of the proposed algorithm with the sequential JPEG and with EZW at high bit rates for the image LENA.

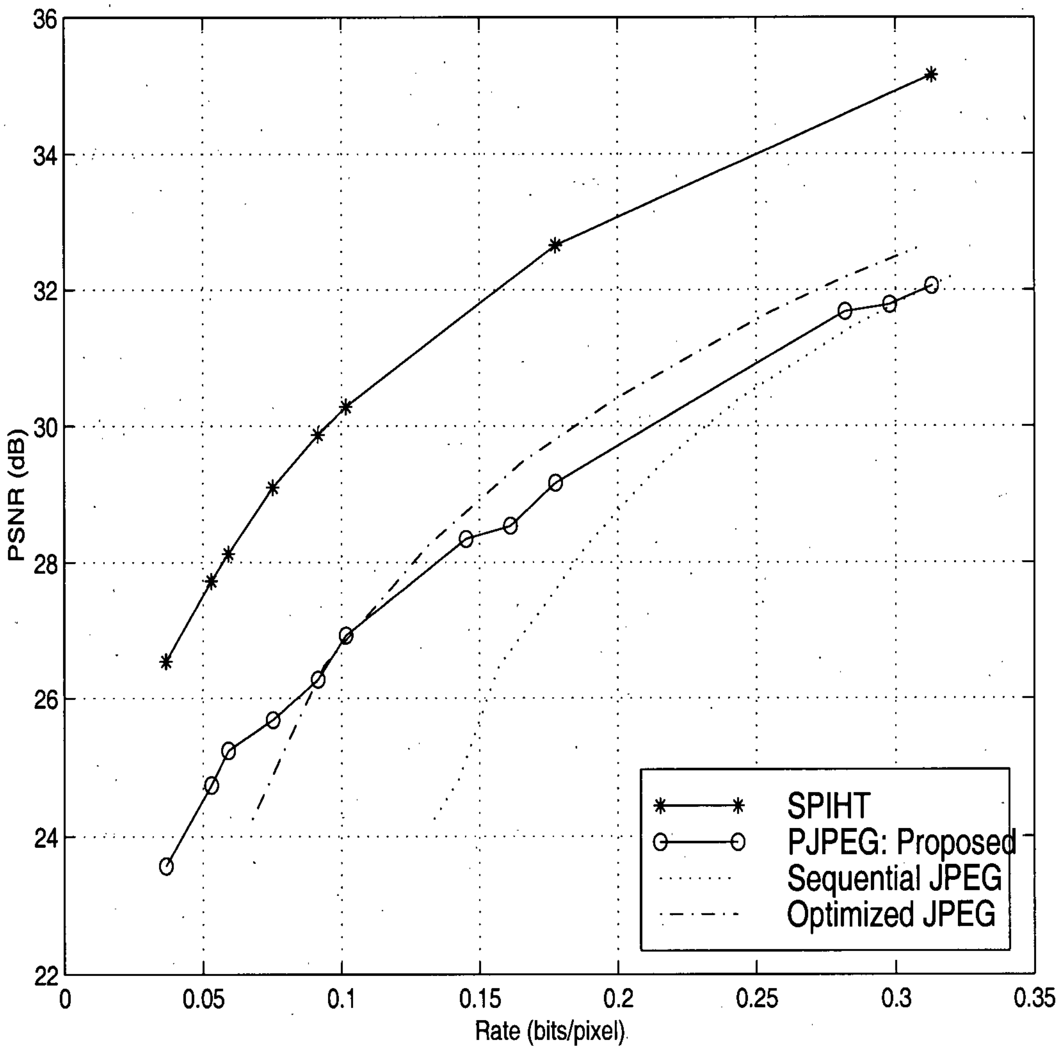


Figure 3.11: Comparison of the proposed algorithm with the sequential JPEG and with EZW at low bit rates for the image LENA.

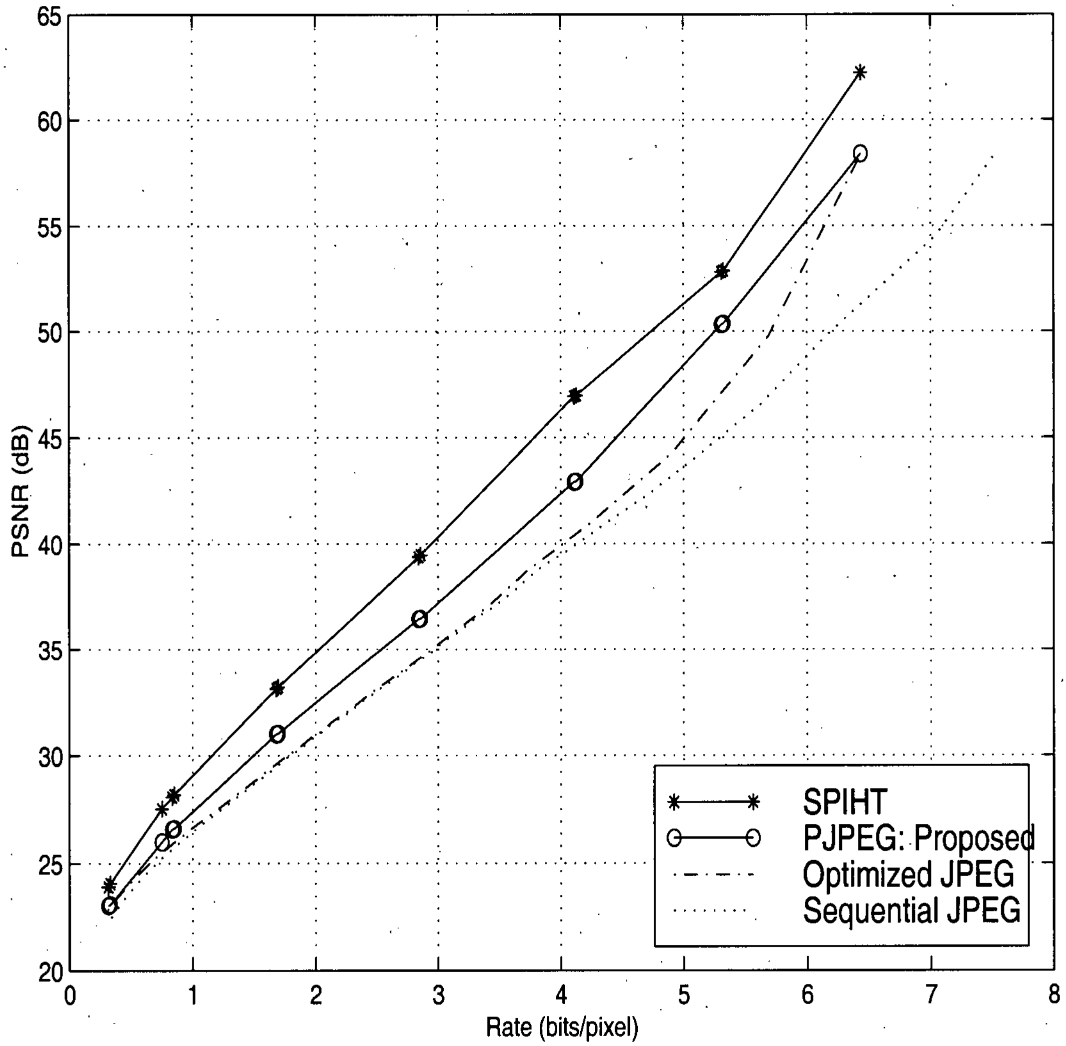


Figure 3.12: Comparison of the proposed algorithm with the sequential JPEG and with EZW at high bit rates for the image MANDRIL.

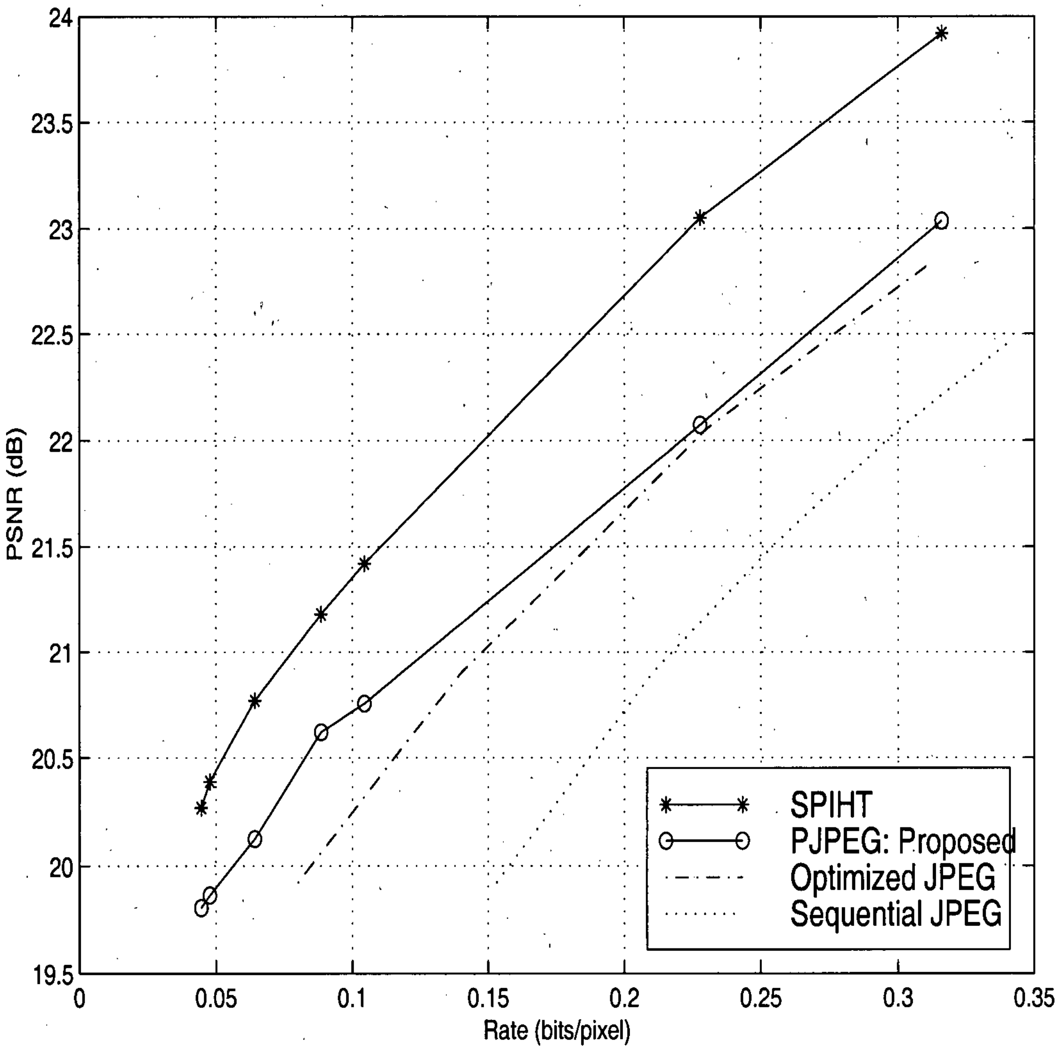
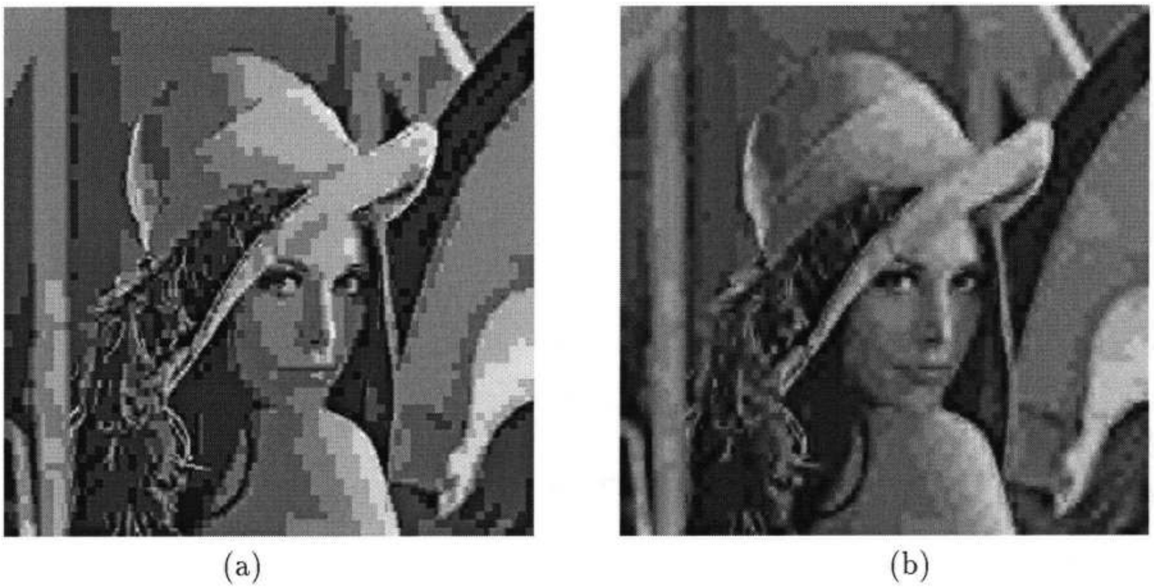
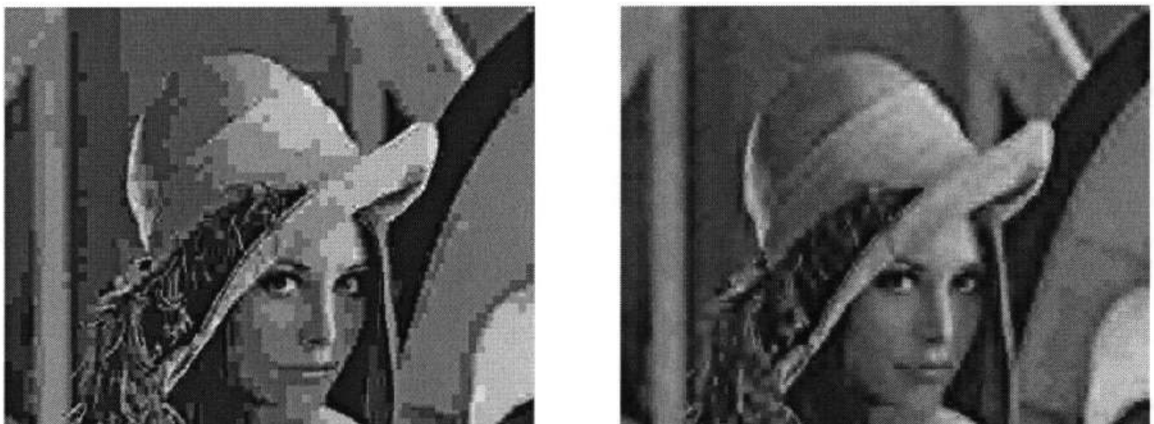


Figure 3.13: Comparison of the proposed algorithm with the sequential JPEG and with EZW at low bit rates for the image MANDRIL.



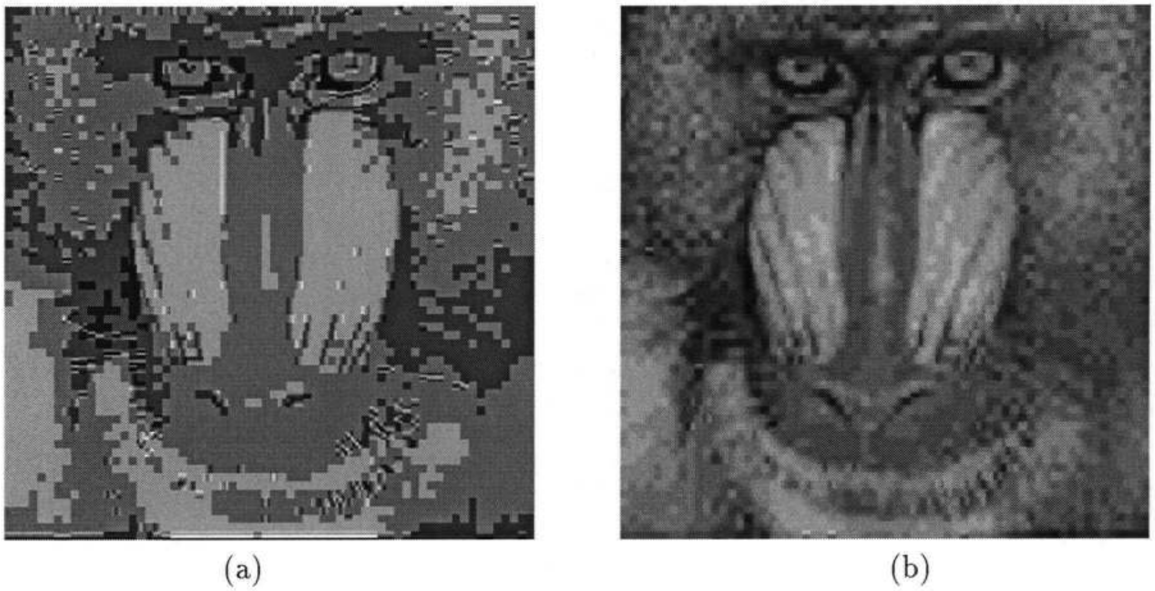
(a) (b)

Figure 3.14: Image LENA coded using (a) optimized sequential JPEG and (b) proposed PJPEG at 0.075 bpp.



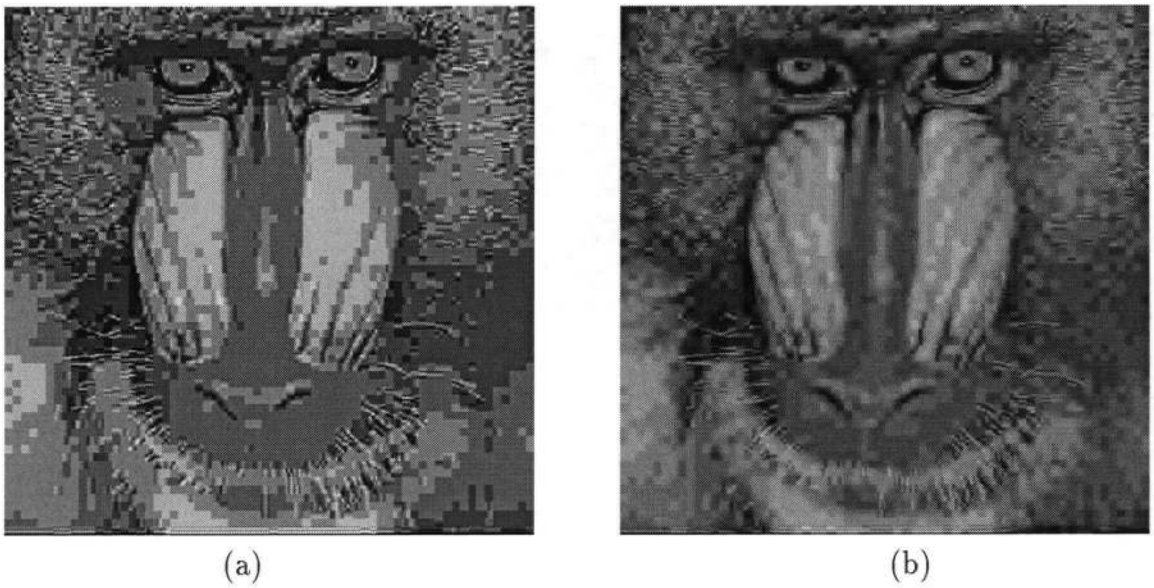
(a) (b)

Figure 3.15: Image LENA coded using (a) optimized sequential JPEG and (b) proposed PJPEG at 0.1 bpp.



(a) (b)

Figure 3.16: Image MANDRIL coded using (a) optimized sequential JPEG and (b) proposed PJPEG at 0.055 bpp.



(a) (b)

Figure 3.17: Image MANDRIL coded using (a) optimized sequential JPEG and (b) proposed PJPEG at 0.1 bpp.

image content and resolution, two grouping patterns, one obtained using a set of four 512×512 training images (BIRD, FIGHTER, PEPPERS, TRUCK) and another using the same images at 256×256 resolution, are applied to the 512×512 images LENA and MANDRIL. The selected training images differ substantially in content, as shown in Figure 3.18. Figures 3.19 and 3.20 show the different PSNR values at medium and high bit rates for the images LENA and MANDRIL, respectively. Figures 3.21 and 3.22 show the same information at low bit rates. Clearly, the performance of the grouping pattern obtained using the training images is very close to that of the grouping pattern designed using both of the test images LENA and MANDRIL.

Figures 3.19 through 3.22 illustrate that our grouping algorithm performs quite well for different images and resolutions. However, note that substantial changes in content (e.g., natural vs medical) or resolution (e.g., 64×64 vs 2048×2048) do impact the compression performance more significantly. Nevertheless, different patterns can still be designed off-line that achieve consistently good compression performance for a very wide range of image types and resolutions.

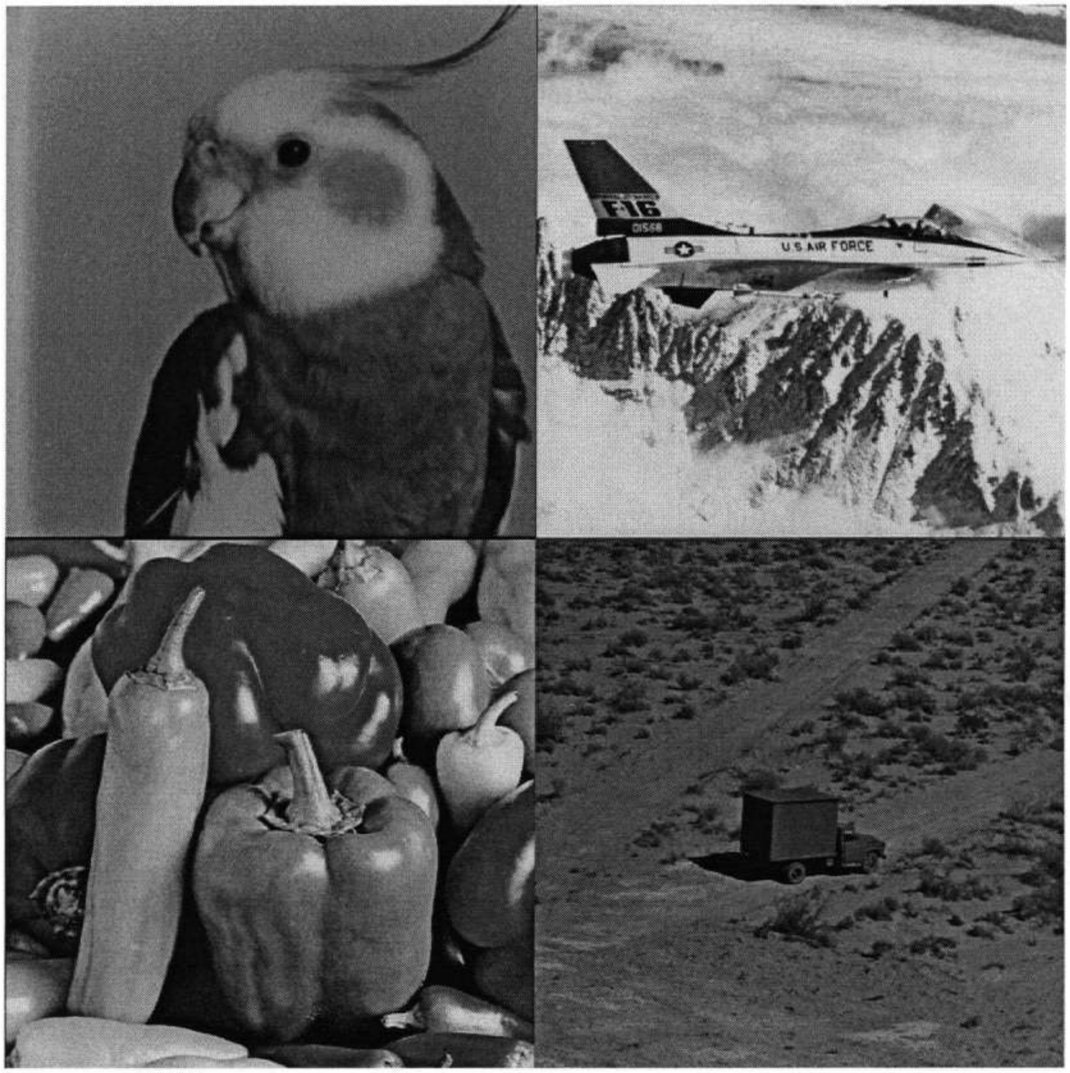


Figure 3.18: Four training images: BIRD, FIGHTER, PEPPERS, and TRUCK.

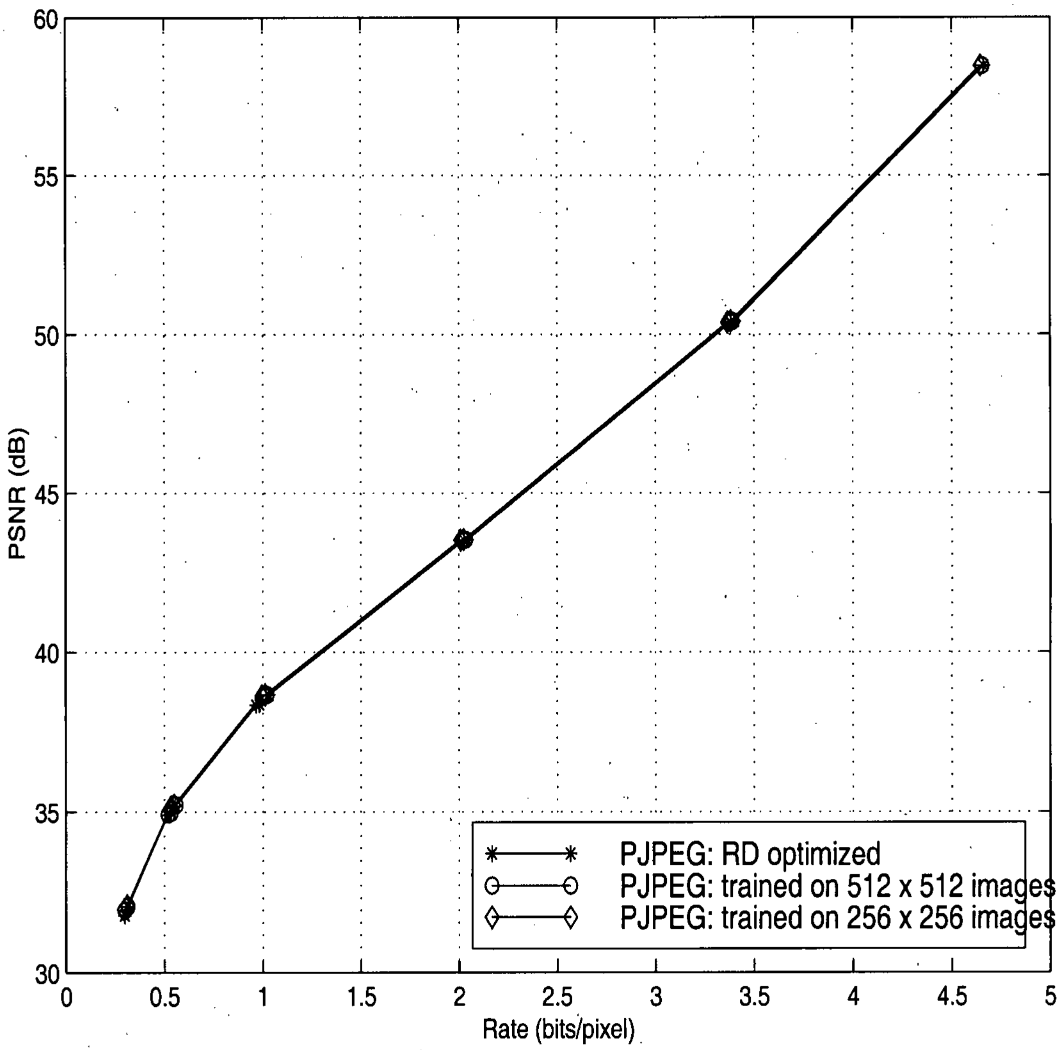


Figure 3.19: Comparison between the performance of image specific and non-specific grouping algorithms at medium and high bit rates for the image LENA.

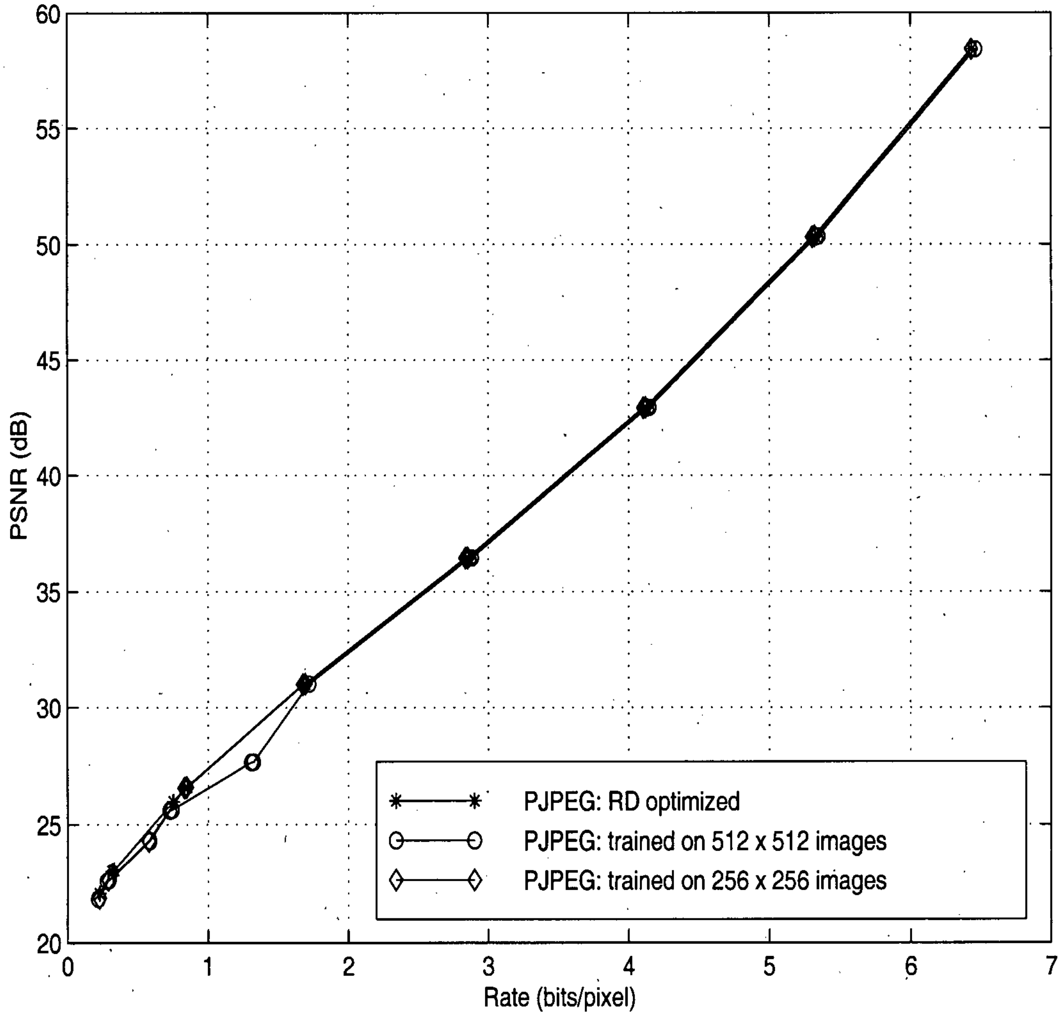


Figure 3.20: Comparison between the performance of image specific and non-specific grouping algorithms at medium and high bit rates for the image MANDRIL.

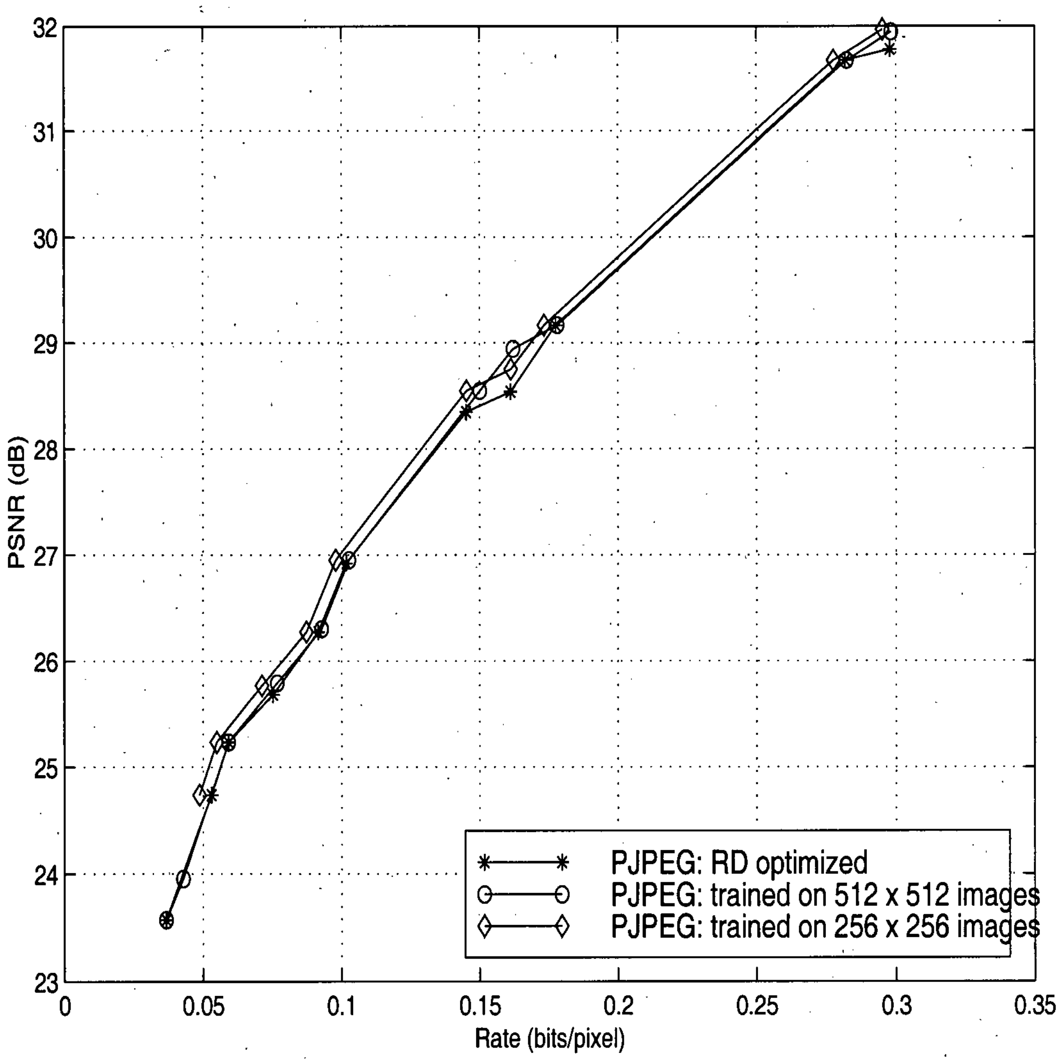


Figure 3.21: Comparison between the performance of image specific and non-specific grouping algorithms at low bit rates for the image LENA.

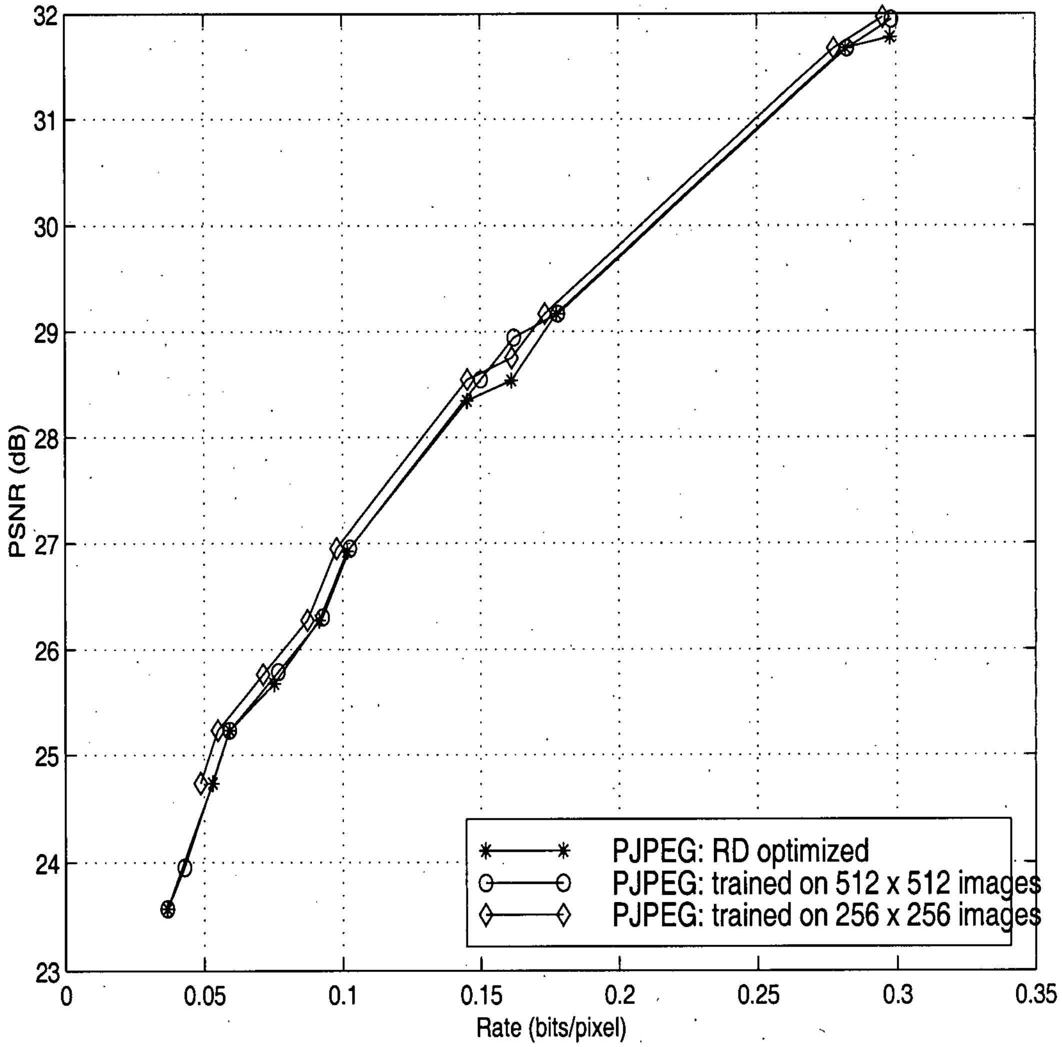


Figure 3.22: Comparison between the performance of image specific and non-specific grouping algorithms at low bit rates for the image MANDRIL.

Chapter 4

Constrained Rate Control in PJPEG

Rate control is the ability of coding an image up to a specified rate [14]. Our goal here is to seek the best possible image reproduction quality given the user-specified bit rate. As is well known, encoders that are sequential JPEG compliant cannot achieve automatic rate control in the sense that a desired bit rate can only be achieved approximately by using a trial-and-error approach, i.e., each time quantization table elements are scaled up/down, DCT coefficients are scaled down/up, resulting in decreased/increased bit rate. The lack of a rate control functionality in sequential JPEG encoders limits their usefulness in many important applications. Rate control has thus become a key desired feature of the emerging JPEG-2000 standard [15].

Despite the various syntax constraints of PJPEG, precise rate control can still be achieved. In fact, shown in the following sections is a method that allows the above proposed PJPEG compliant encoder to maintain high reproduction quality subject to the constraint of a fixed bit rate. Figure 4.23 shows the high level block diagram of the rate control algorithm, which is applied to scans ordered according to the grouping procedure described above. Let L be the maximum number of bits available anytime during the encoding process, H be the size in bits of the PJPEG's frame header, H_s be the size in bits of the current scan's header, and N is the number of 8×8 blocks. First, the value of L, which is initially set to the maximum allowed number of bits (i.e. desired bit rate), is compared to the size H of PJPEG's frame header, and if $L < H$, the encoding procedure is terminated. This case represents the very unlikely event that the

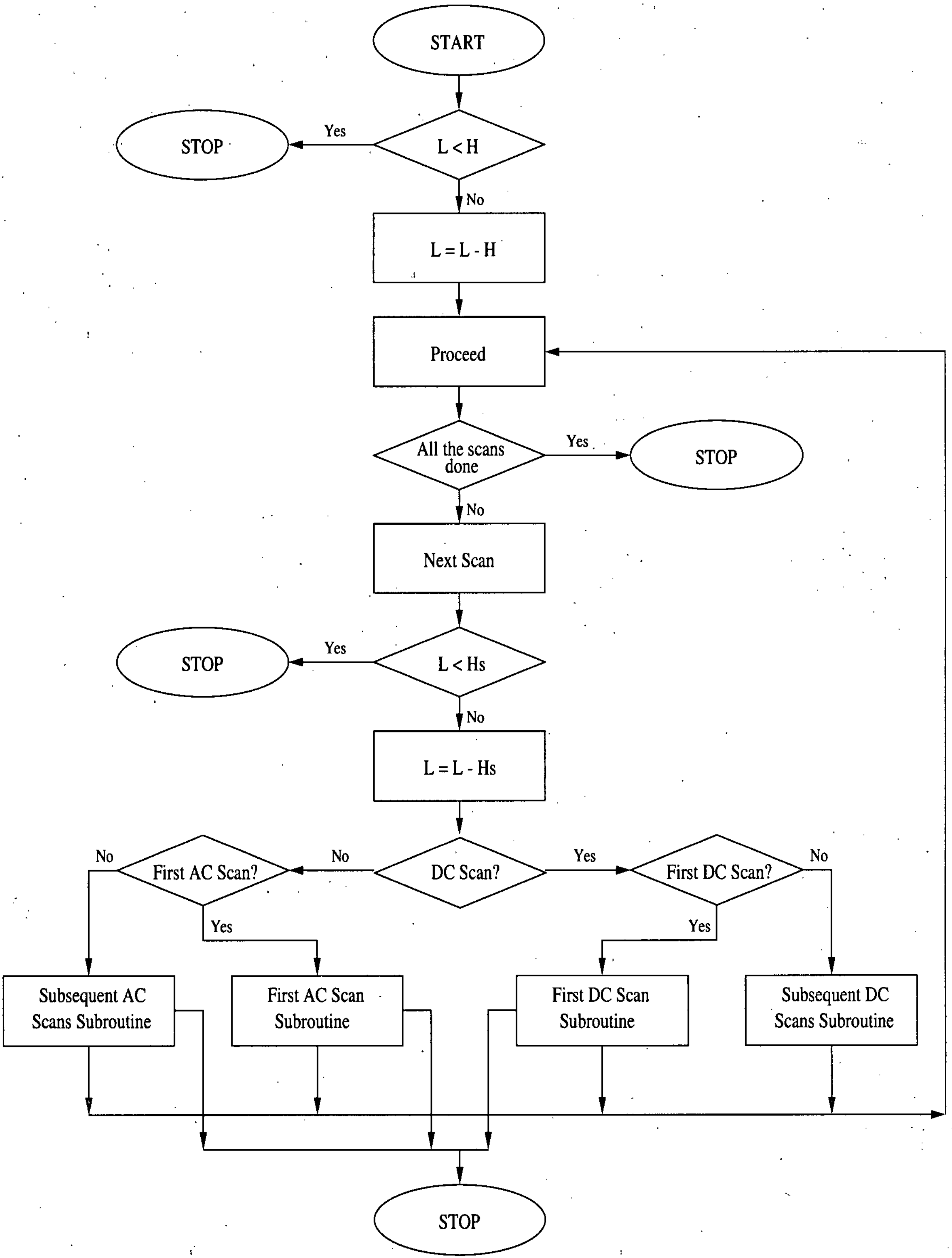


Figure 4.23: High level block diagram of the rate control algorithm. L is the maximum number of bits available anytime, initially set to the maximum allowed number of bits, H is the size of PJPEG's frame header in bits, H_s is the size of the current scan header in bits.

maximum allowed number of bits is too small to accommodate even PJPEG's header. Otherwise, L is decremented by H, and we then proceed by 1) making sure that at least one scan is available and 2) comparing the new value of L to H_s, the size of the current scan header. In the very probable event that $L \geq H_s$, L is further decremented by H_s, and we then proceed encoding the DCT bit values in the subject scan in a way that depends on its type (DC or AC, first or subsequent). Figures 4.24, 4.25, 4.26 and 4.27 illustrate the algorithms corresponding to the first DC scan, subsequent DC scans, first AC scan, and subsequent AC scans, respectively. Each of the algorithms is described in detail in the following sections.

4.1 First DC Scan

JPEG does not allow encoding of only a few DC coefficients of an image. Therefore, the size of the bit stream after encoding all the DC coefficients is the minimum rate achievable by conventional JPEG encoders. However, we can "fool" the decoder by replacing some of the actual DC bits with zeros which require fewer code bits to achieve lower rate.

As illustrated in Figure 4.24, we first verify that we have at least $2N$ code bits available, where N is the number of 8×8 blocks in the image being coded. The number $2N$ is the minimum number of bits required to encode the fist DC scan since the variable length codeword for the DC difference of zero is "00". Satisfying this condition, we encode the group of bits of the first non-coded 8×8 block, decrement L by the corresponding B code bits used to encode the current block, and compare L to $2n$, where $2n$ is the number of code bits needed to represent n "0"s, and n is the number of non-coded blocks. If $L < 2n$, the code bits representing the current block are removed, $2n$ code bits (which happen to be "0"'s) are added to the bit stream, and the encoding procedure is terminated. Otherwise, we encode the following block and so on until we exhaust all

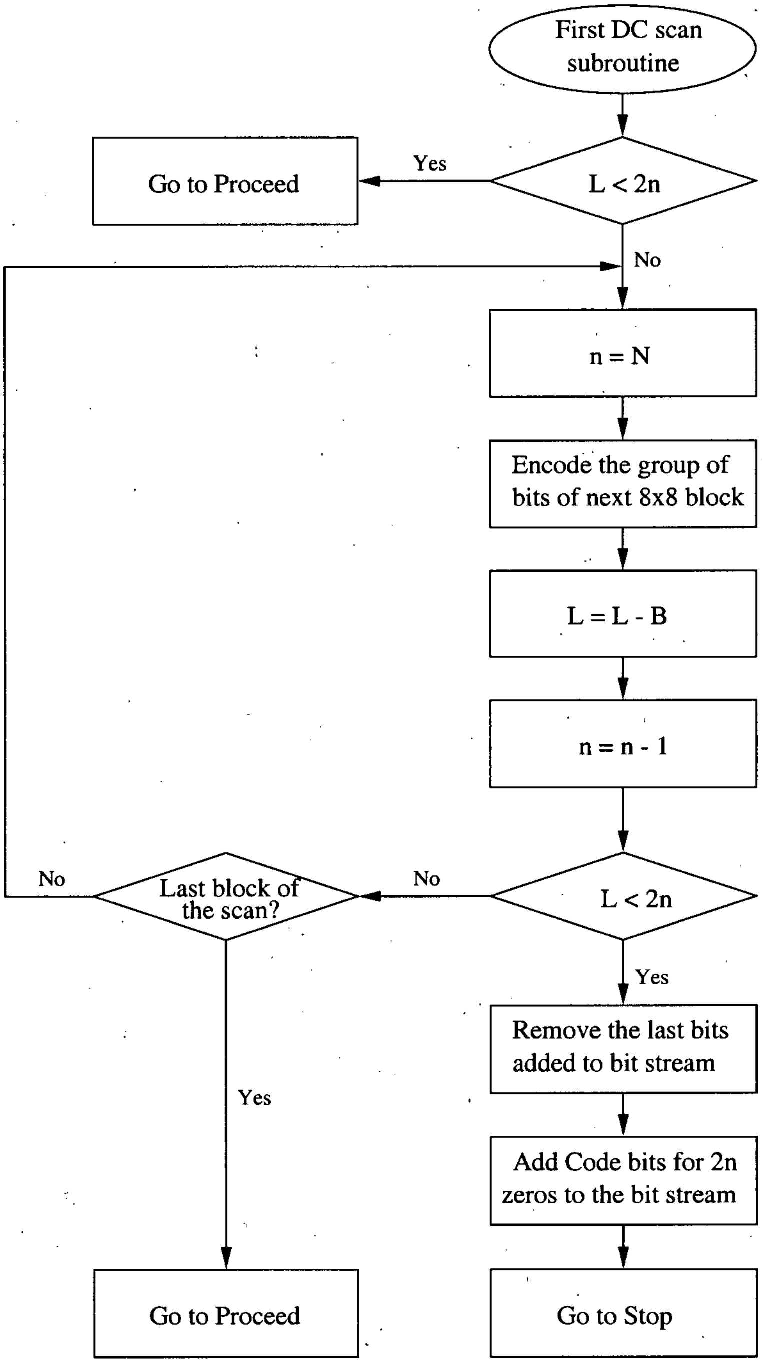


Figure 4.24: Block diagram of the rate control algorithm used for the first DC scan. n is the number of non-coded blocks, N is the number of 8×8 blocks and B is the number of code bits generated by encoding the current 8×8 block.

available bits or the last block in the image has been encoded. In the latter case, we proceed as shown in Figure 4.23 to encode the next available scan.

When the image with some of the DC coefficients replaced with zeros is decoded, its reconstructed image will contain blocks whose gray levels are vividly different from their neighboring blocks. Although such degradation in image quality is not desired, the above algorithm can be useful when only extremely low bandwidths are available.

4.2 Subsequent DC Scan

For all subsequent DC scans, we simply verify if we have N available code bits to represent as-is [4] the DC bits. If there are enough code bits, the DC scan is copied to the bit stream in a sequential order (left to right, top to bottom), and we then proceed to encode the following scan. Otherwise, L is incremented by H_s, the current subsequent DC scan is discarded, and the following scan (which would hopefully be an AC scan) is considered. Figure 4.25 illustrate the rate control algorithm when a subsequent DC scan is being encoded.

4.3 First AC Scan

As shown in Figure 4.26, we here do not encode 8×8 blocks individually, rather, we start and continue processing AC sub-coefficients or groups of bits until the first nonzero sub-coefficient is encountered. At such a point, we encode all zero sub-coefficients and the nonzero sub-coefficient as described in Chapter 2, which produces B bits and processes m blocks. Then, L is decremented by B bits. To determine the number of non-coded blocks, we verify if the nonzero sub-coefficient is the last element of the spectral band of the subject block. If it is not the last element of the band, the number of non-coded blocks should include the current block (there still exist some sub-coefficients to be coded in the

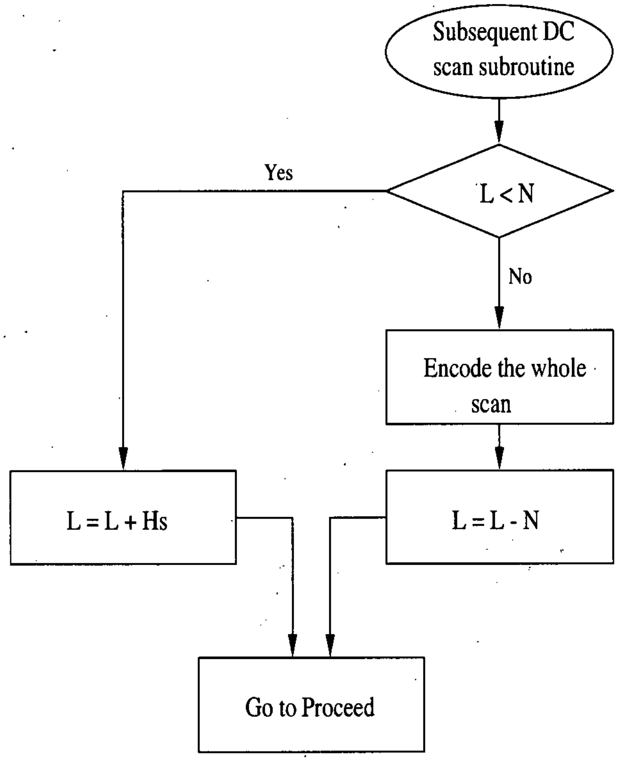


Figure 4.25: Block diagram of the rate control algorithm used for the subsequent DC scans.

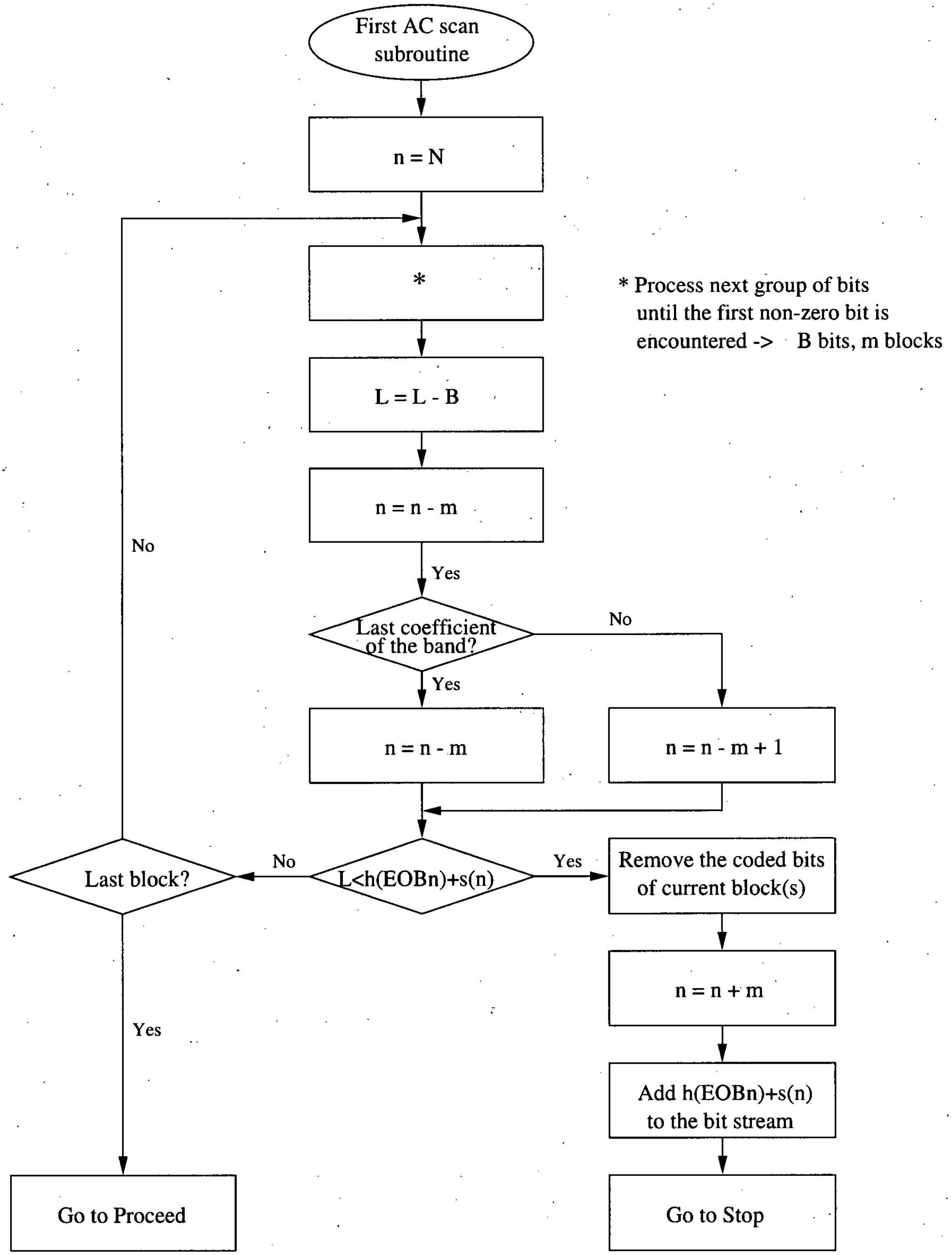
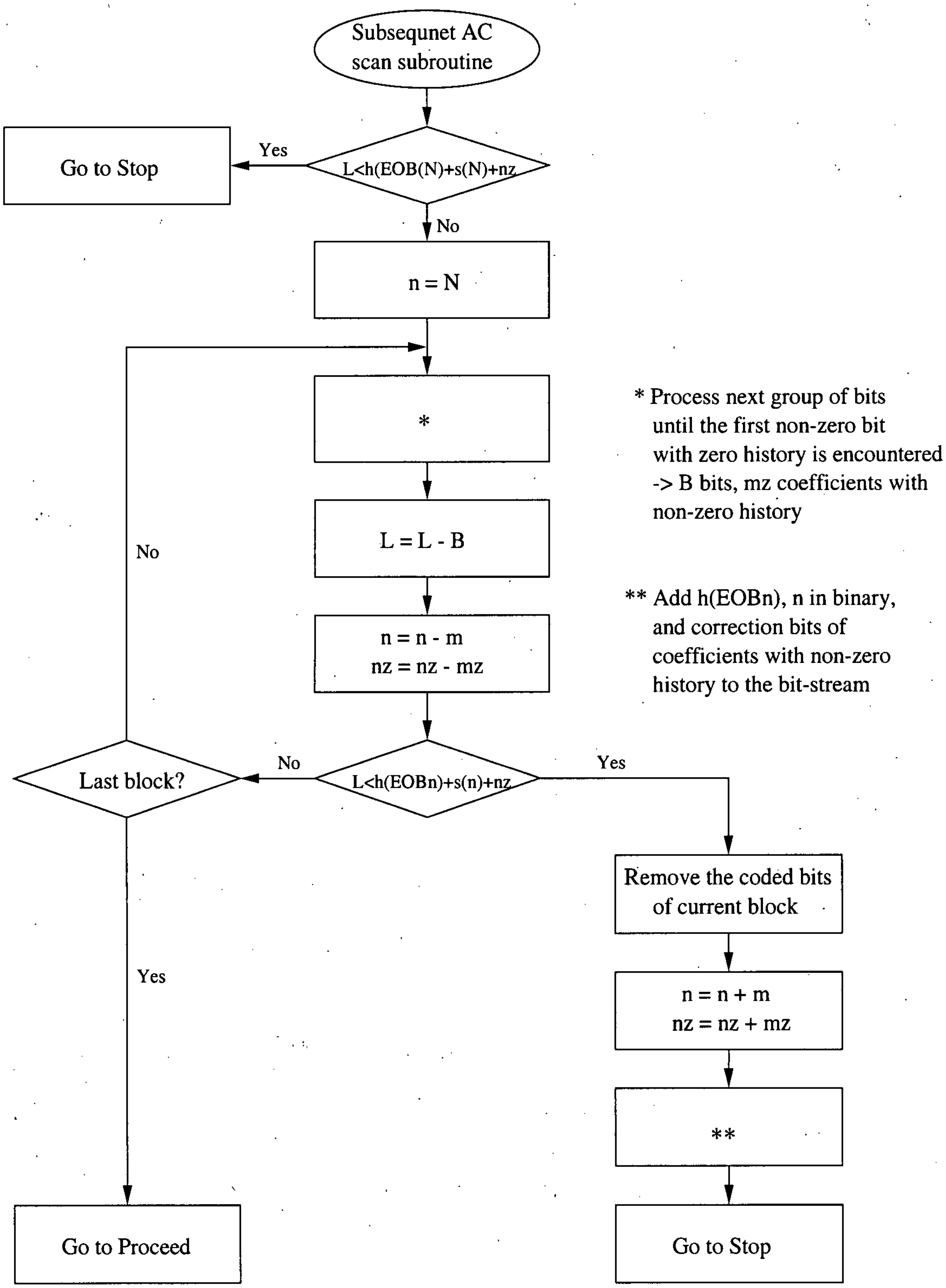

First AC scan subroutine
n = N
*
* Process next group of bits until the first non-zero bit is encountered -> B bits, m blocks
L = L - B
n = n - m
Yes
Last coefficient of the band?
No
Yes
n = n - m
n = n - m + 1
L<h(EOBn)+s(n)
No
Yes
Last block?
No
Yes
Remove the coded bits of current block(s)
n = n + m
Add h(EOBn)+s(n) to the bit stream
Go to Proceed
Go to Stop

Figure 4.26: Block diagram of the rate control algorithm used for the first AC scan. B is the number of code bits generated by encoding the current 8×8 block or blocks, m the number of blocks processed, $h(EOBn)$ the length in bits of the variable length code representing EOBn and $s(n)$ the number of bits required to represent n.

current block). Therefore, the current number n of non-coded blocks is decremented by $m - 1$ if the nonzero sub-coefficient is not the last element of the corresponding band, and by m otherwise. The parameter L is subsequently compared to $h(EOBn) + s(n)$. $h(EOBn)$ is the length in bits of the variable length code representing $EOBn$, the end-of-band when n blocks of all zeros are encountered, and $s(n)$ is the number of bits required to represent n. $h(EOBn) + s(n)$ is the minimum number of code bits required to encode the sub-coefficients in non-coded blocks that were ignored by simply assuming them to be zero. If $L < h(EOBn) + s(n)$, the code bits representing the current block(s) are removed, n is restored to its previous value and code bits for $EOBn$ and n in binary are added to the bit stream, and the encoding procedure is terminated. Otherwise, we encode the following block and so on until we exhaust all available bits or the last block in the image has been encoded.

4.4 Subsequent AC Scan

As shown in Figure 4.27, for all subsequent AC scans, we first verify if we have $h(EOB(N)) + s(N) + nz$ available code bits, where nz is the number of the DCT coefficients that have been partially encoded with a nonzero history. [4]. This is the minimum number of bits required to represent the coefficients with nonzero history and force the other coefficients to zeros. $h(EOB(N))$ is the length in bits of the variable length code representing $EOB(N)$, the end-of-band when N (initial number of blocks in an image) blocks of zeros are encountered, and $s(N)$ is the number of bits required to represent N. Additional nz bits are required since the DCT coefficients with nonzero history are not variable length coded, but appended as-is. If the total number of available bits is too small, we continue processing AC sub-coefficients until the first nonzero sub-coefficient with zero history is encountered. At such a point, we encode all zero sub-coefficients and the nonzero



* Process next group of bits until the first non-zero bit with zero history is encountered -> B bits, mz coefficients with non-zero history

** Add h(EOBn), n in binary, and correction bits of coefficients with non-zero history to the bit-stream

Figure 4.27: Block diagram of the rate control algorithm used for the subsequent AC scans. nz is the number of DCT coefficients with nonzero history and mz the number of DCT coefficients with nonzero history which were processed.

sub-coefficient as described in Section 2.3, producing B bits. Then, L is decremented by B bits. It is assumed that processing of the coefficients up to this point caused mz coefficients with nonzero history and m blocks to be coded. Therefore, the values of n and nz are decremented by m and mz respectively. L is subsequently compared to $h(EOB(n)) + s(n) + nz$. If $L < h(EOB(n)) + s(n) + nz$, the code bits representing the current block(s) are removed, n and nz are restored to their previous values, code bits for $EOBn$ and n in binary representation and correction bits of the coefficients with nonzero history are added to the bit stream, and the encoding procedure is terminated. Otherwise, we encode the following blocks and so on until we exhaust all available bits or the last block in the image has been encoded.

To summarize, the above algorithm can achieve relatively precise rate control through sending both complete and partial scans. The partial scans are obtained by appropriately forcing the encoding of "0"'s once the size of the encoded image is approaching the desired maximum number of bits. The same algorithm can also be used, in a straightforward way, to achieve distortion or quality control. Although PJPEG imposes many constraints on the encoder, more precise rate/distortion control can still be achieved by RD optimizing the control algorithm explicitly. That is, we should not wait until all available bits would be exhausted before forcing the rest of the coefficient bits to be "0"'s. Instead, we can employ a Lagrangian multiplier, whose value is adaptively changed during the encoding process, to decide optimally in the rate-distortion sense whether a "1" should be converted to a "0". However, the gain in rate/distortion control precision does not, so far, appear to justify the additional complexity.

The performance of the proposed constrained rate control algorithm is shown in Table 4.11. The table shows the target bit rates of 0.0625, 0.125, 0.25, 0.5 bpp, as well as achievable bit rates and the corresponding PSNR values for the image LENA. Using our rate control algorithm, the actual bit rates are very close to the desired bit rates.

Desired bit rate (bpp)	Obtained bit rate (bpp)	Obtained PSNR
0.0625	0.0618	24.74
0.125	0.125	26.43
0.25	0.249	29.14
0.5	0.499	32.42

Table 4.11: Rate Control Results for the image LENA.

Chapter 5

Hierarchically Embedded PJPEG

In Chapter 3, we developed a stand-alone RD optimized PJPEG compliant encoding algorithm. In this chapter, the algorithm is employed as a submode of HJPEG to allow quality progressive control for each resolution layer. Figure 5.28 shows the block diagram of the hierarchically embedded PJPEG (HPJPEG) coding procedure with three resolution layers. In HJPEG, an image is encoded as a sequence of frames, while each frame is an independently encoded JPEG image. These frames provide reference reconstructed images which are needed for prediction of subsequent frames. Except for the first frame, difference frames, obtained by computing the difference between the original image and the reference reconstructed image from a frame at a lower resolution, are encoded. Downsampling and upsampling filters are used to provide the reference reconstructed images.

The design of an efficient HPJPEG coding algorithm involves two steps: 1) selection of appropriate filters and 2) selection of proper transition points from lower to higher resolutions. These steps are discussed in detail in following sections.

5.1 Selection of Filters

The selection of downsampling and upsampling filters is not specified in the JPEG standard and is left to specific encoder implementations. In order to evaluate its influence on the performance of our HPJPEG encoder, an experiment is performed where different set of simple filters are compared.

The first set of filters consists of simple averaging (for downsampling) and bi-linear

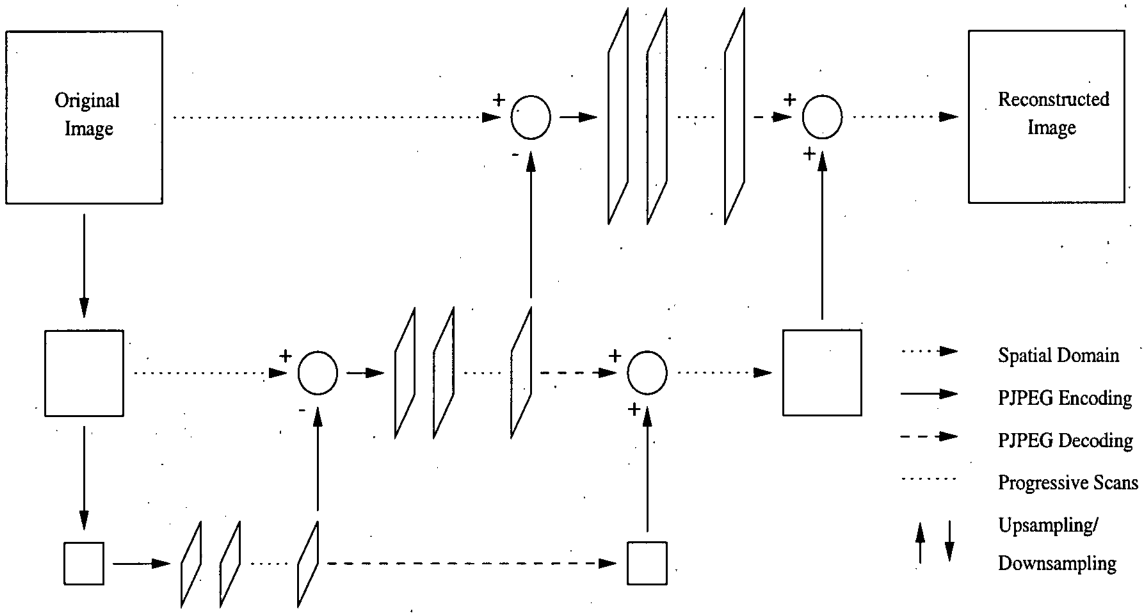


Figure 5.28: Encoding and decoding procedures for hierarchically embedded PJPEG with three layers.

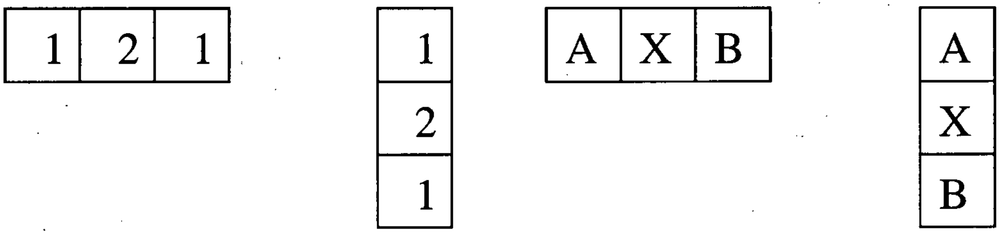
interpolation (for upsampling) as shown in Figure 5.29.

The center sample in Figure 5.29 (a) is aligned with the left-most column and top row of the original image when computing the left-most column and top row of the lower resolution image, respectively. Filter output values are normalized by the sum of the filter weights. Sample values at the boundaries are replicated when filter weights are outside of the image boundaries. Each 1-D filter can be applied sequentially to achieve 2-D downsampling.

An interpolated value using Figure 5.29 (b) can be calculated by applying the simple equation to the adjacent sample values, R_A and R_B;

$$P_X = \frac{R_A + R_B}{2}, \tag{5.5}$$

where P_X is the newly interpolated value. When the filter value is outside of the image, opposite boundary conditions are applied, as in the downsampling case.



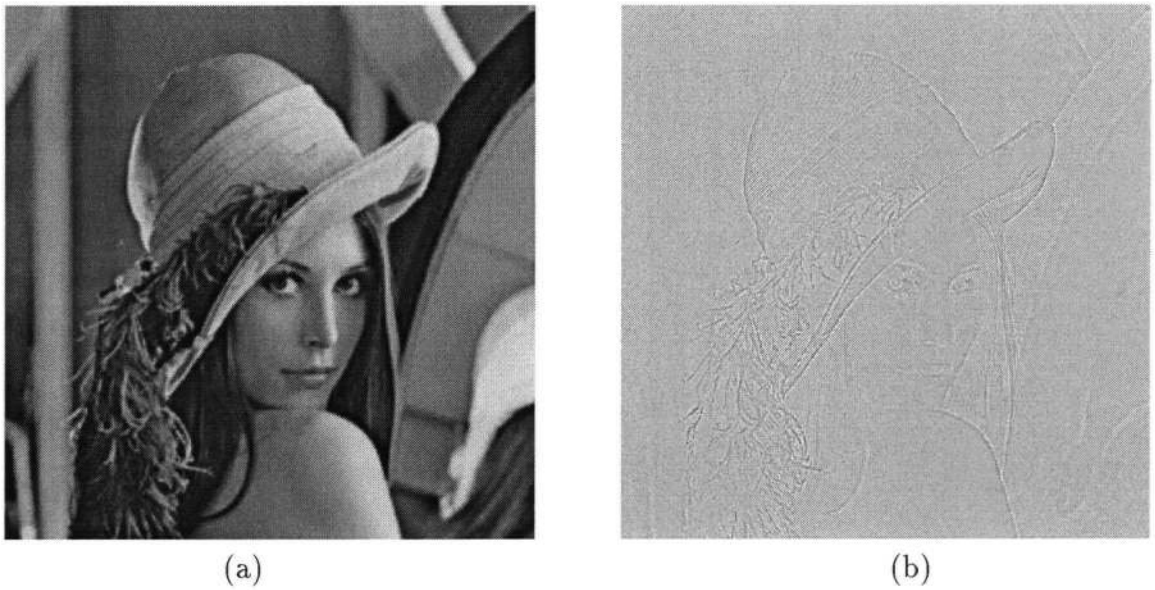
(a) (b)

Figure 5.29: Downsampling and upsampling filters: (a) 1-D averaging downsampling filter and (b) 1-D bi-linear interpolation upsampling filter.

A good alternative is a set of filters that are obtained using spline interpolation and approximation techniques [16]. Splines are polynomial functions that can be used to interpolate discrete sample values. In [16], an upsampling filter using a specific type of spline, called a B-spline, is designed to minimize the least mean squared error for a specific lowpass filter.

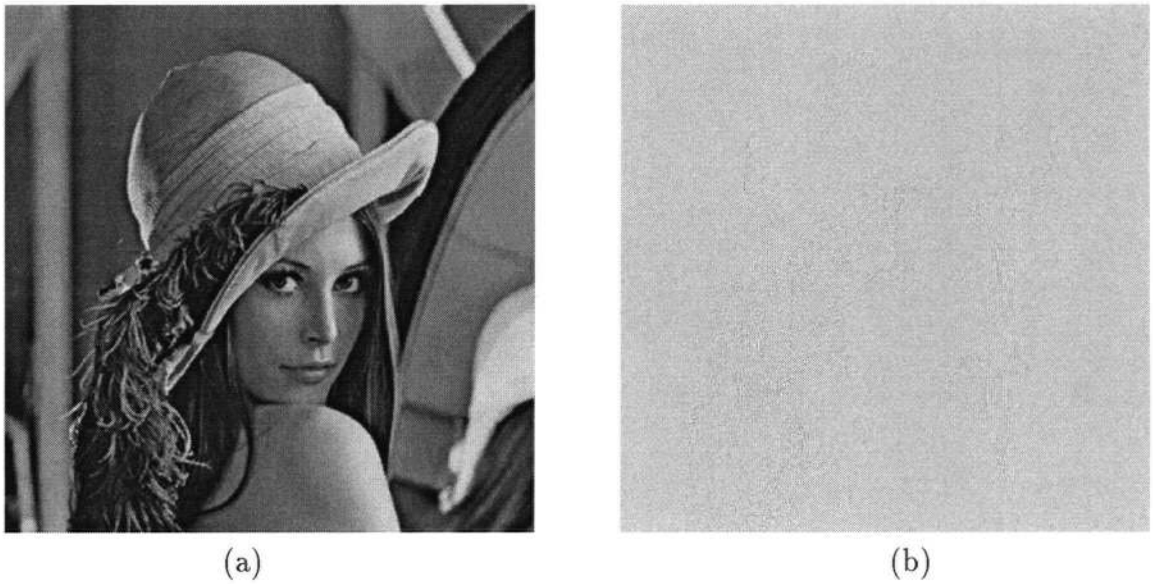
Figures 5.30 and 5.31 show the downsampled/upsampled versions of the 512 × 512 image LENA and the difference images using the two sets of filters, respectively. As can be seen from the difference images, the B-spline filters clearly outperform the averaging and bi-linear interpolation filters. It can be easily predicted that coding of the difference image in Figure (b) requires a fewer number of bits than that in Figure (a), as the difference image in Figure (b) contains more zeros and less high frequency information (e.g., edges) than Figure (a).

As a result, we employ the B-spline filters as part of out HPJPEG encoder. Note that better filters can always replace the existing ones to improve our encoder's complexity-performance tradeoffs.



(a) (b)

Figure 5.30: (a) The 512 × 512 image LENA sampled and reconstructed using the filters shown in Figure 5.29, and (b) the corresponding difference image.



(a) (b)

Figure 5.31: (a) The 512 × 512 image LENA sampled and reconstructed using B-spline filters, and (b) the corresponding difference image.

5.2 Selection of Transition Points

In conventional HJPEG, transition from the current layer to the higher resolution layer occurs only after the frame at the current layer has been fully decoded and upsampled, since the differential frames at the higher layers have been already constructed based on the frame at the current layer. In such a case, even after the desired image quality has been achieved at the current resolution, the decoder has to wait until it receives the rest of the information before proceeding to the higher resolution layer. The HPJPEG algorithm, however, allows transitions from one layer to the higher one by allowing the decoder to decide on the reference image quality, i.e., the encoder transmits the encoded differential frames based on the reconstructed image at the decoder at any time.

Although our algorithm can be used as-is to provide the receiver with full progressiveness control, it would then require many computations because prioritized scans would have to be generated at each layer as discussed in Section 3.3. Also transition from one layer to the higher one, if not done properly, can cause a decrease in overall coding efficiency. Therefore a default coding procedure is desired that maintains high compression performance at all bit rates and resolutions. Figures 5.32 shows the PSNR vs. bit rate for the image LENA at three resolutions, each of which is encoded using the RD optimized PJPEG algorithm obtained in Chapter 3.

Evidently, the performance of the PJPEG encoder encoding the 128 × 128 resolution image is better than the other two resolutions at most bit rates up to the potential transition point marked as T1. Beyond the point T1, it is outperformed by the PJPEG encoder encoding the next higher resolution image. Naturally, T1 would be the first desired transition point for the HPJPEG encoder. Likewise, the second desired transition point would be T2 as shown in the Figure 5.32. Although the first transition can occur exactly at T1, the second transition would not necessarily have to be T2, since T2 is the

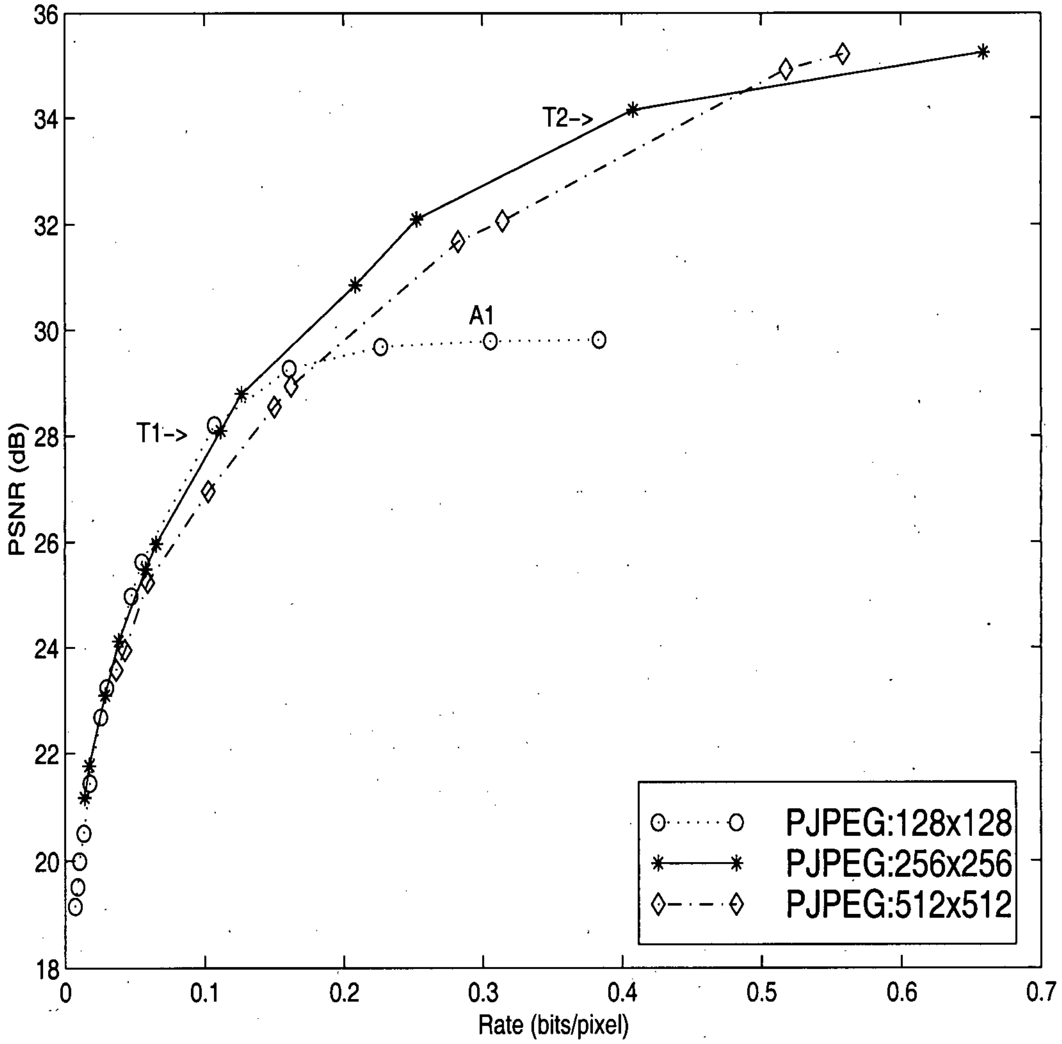


Figure 5.32: PSNR vs. Rate for 128 × 128, 256 × 256, and 512 × 512 LENA images coded using PJPEG at low and medium bit rates.

point generated originally by the PJPEG encoder encoding the 256 × 256 image and is not generally reachable by encoding the differential frame during the hierarchical coding procedure. Nevertheless, the best second transition point obtained by differential frame encoding using the same PJPEG algorithm occurs approximately at the same position as T2. The performance of the HPJPEG encoder is presented in the next section.

5.3 Experimental Results

Figure 5.33 shows the PSNR vs. rate of the HPJPEG encoder with three layers and the RD optimized PJPEG encoder for the 512×512 image LENA at low and medium bit rates. While the HPJPEG coder can generate compressed images at all the bit rates supported by each of the RD optimized PJPEG encoders, its performance is nearly the same as or better than the "best" of the three stand-alone RD optimized PJPEG encoders with more progressiveness stages as shown in the figure. Also shown in the figure are the performance levels of the SPIHT encoder and the optimized JPEG encoder. Likewise in Section 3.4, the HPJPEG encoder outperforms the optimized JPEG encoder at all bit rates, while achieving some of the very low bit rates not supported by optimized JPEG. However, SPIHT algorithm is still more compression efficient than HPJPEG, as shown in the figure. Figure 5.34 also shows the performance levels of the same encoders at high bit rates.

In Figure 5.35, the image LENA was reconstructed using conventional HJPEG and HPJPEG, respectively. Figure 5.35 (a) shows the reconstructed image corresponding to point A1 (0.3 bpp) in Figure 5.32. In conventional HJPEG, the receiver has to wait until all the scans beyond the point A1 are received before it can make a transition to a higher resolution layer. However, receiving further scans does not improve the reconstructed image quality, as shown in Figure 5.32. Such inefficiency is avoided in our HPJPEG

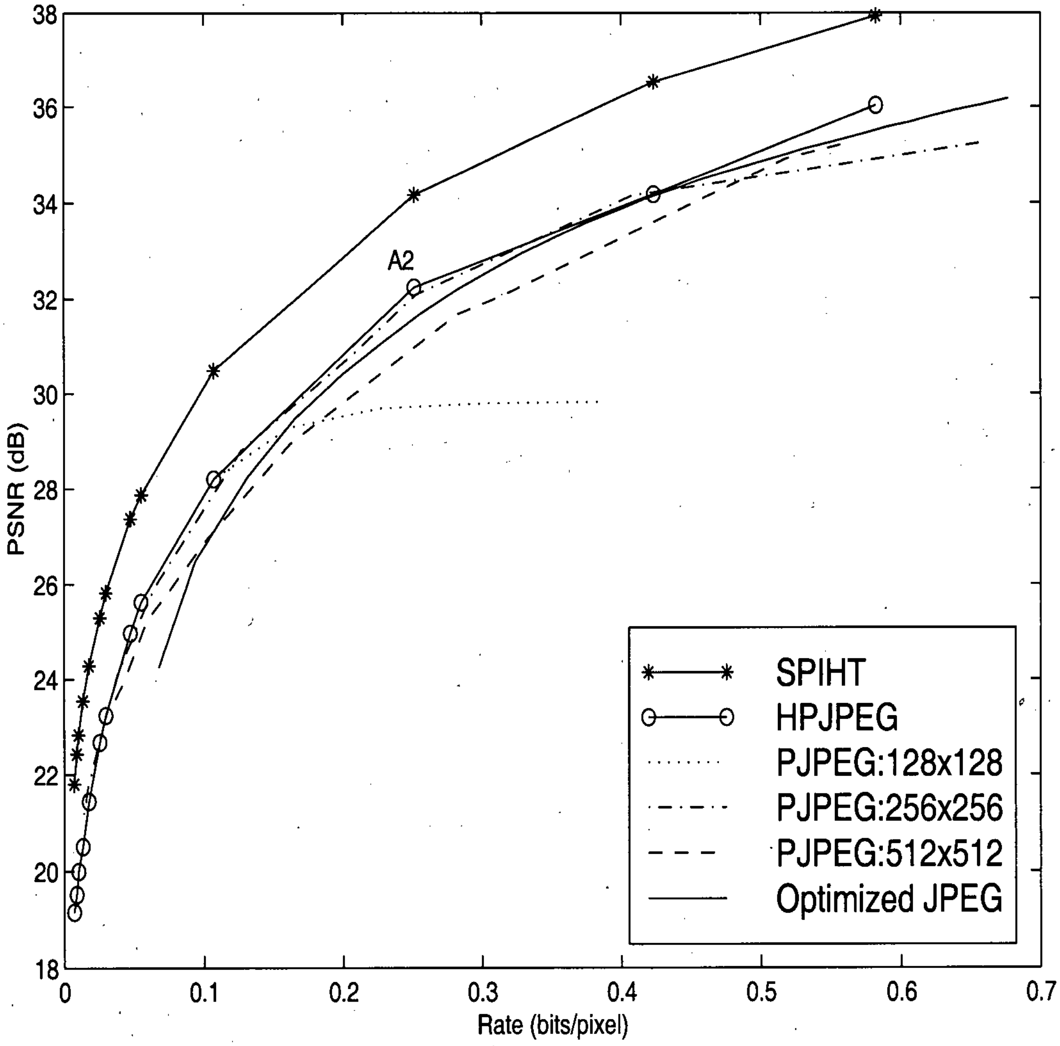


Figure 5.33: PSNR vs. Rate for 128 × 128, 256 × 256, and 512 × 512 LENA images encoded using PJPEG, and the 512 × 512 LENA image coded using HPJPEG, SPIHT and optimized JPEG at low and medium bit rates.

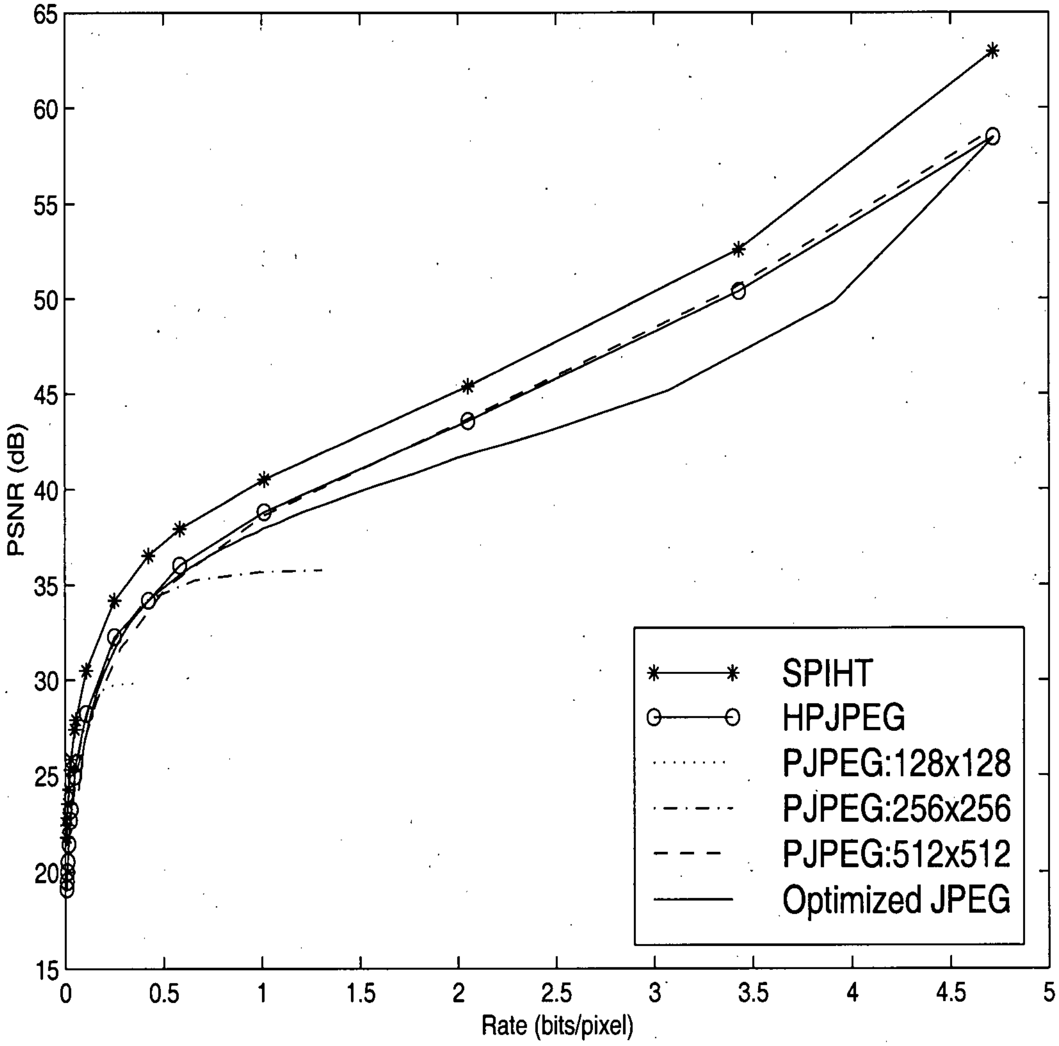
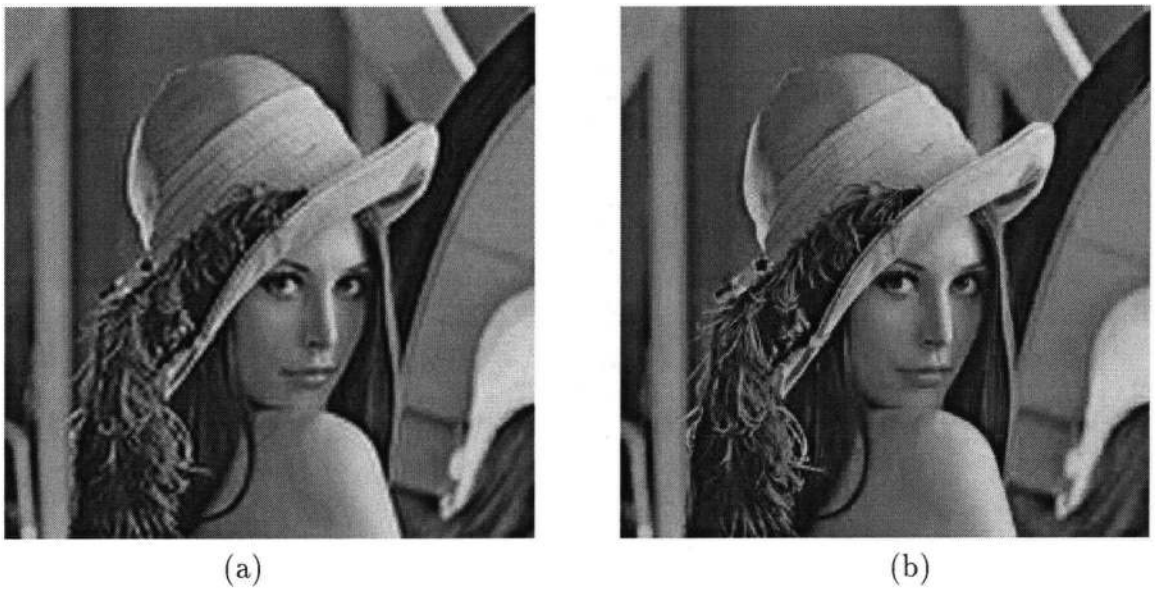


Figure 5.34: PSNR vs. Rate for 128 × 128, 256 × 256, and 512 × 512 LENA images encoded using PJPEG, and the 512 × 512 LENA image coded using HPJPEG, SPIHT and optimized JPEG at all bit rates.



(a)

(b)

Figure 5.35: The 512 × 512 image LENA reconstructed (a) at 0.3 bpp using conventional HJPEG, and (b) at 0.25 bpp using HPJPEG.

algorithm. Using the default transition points obtained in the previous section in our algorithm, a higher PSNR value is achieved at even a lower bit rate for point A2 (0.25 bpp) in Figure 5.33. The corresponding reconstructed image is shown in Figure 5.35 (b). The difference in image quality is vivid in areas of high frequency information such as hair and edges.

Our HPJPEG encoder outperforms conventional HJPEG both objectively and subjectively. The main advantage of our HPJPEG encoder, however, is that reconstructed images at very low bit rates can be achieved using resolution control.

Chapter 6

Conclusions

Progressive JPEG encoders are well suited in many image coding applications. We developed an RD optimized hierarchically embedded progressive JPEG encoder that yields progression in both reproduction image quality and resolution.

First, a stand-alone RD optimized PJPEG encoder was designed that generates a sequence of single-bit scans, ordered based on their priority values. The DCT bits are then grouped using an efficient design algorithm which provides a good balance between compression efficiency and progressiveness resolution. In addition, we have also developed a rate control algorithm by incorporating some simple control routine into the PJPEG encoding process. The PJPEG compliant encoders outperform sequential JPEG encoders substantially at low and high bit rates, and they achieve precise rate/distortion control.

Second, the encoding algorithm obtained in the first step was applied as a submode of hierarchical JPEG to allow spatial resolution control. The resulting hierarchical PJPEG (HPJPEG) encoder allows automatic transition from one layer to the next layer at the receiver's request without fully decoding the lower resolution image. The HPJPEG encoder using the default transition points achieves high rate-distortion performance at all progressive stages, while allowing the decoder to reconstruct images at a wide range of resolutions and bit rates.

Substantially better compression performance and precise rate control, achieved by our JPEG compliant hierarchically embedded progressive encoding algorithms, are two highly desired features currently sought for the emerging JPEG-2000 standard.

Bibliography

[1] K. R. Rao and J. J. Hwang, *Techniques and Standards for Image, Video and Audio Coding.* Englewood Cliffs, NJ: Prentice-Hall, 1996.

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

[3] J. K. Han and G. C. Polyzos, "Networking applications of the hierarchical mode of the JPEG standard," in *IPCCC'96*, (Pheonix, AZ, USA), Mar. 1996.

[4] ISO/IEC, *Digital compression and coding of continuous tone still images: requirements and guidelines.* International Standard Organization, 1994.

[5] B. Chitprasert and K. R. Rao, "Human visual weighted progressive image transmission," *IEEE Transactions on Communications*, vol. 38, pp. 1040–1044, A 1990.

[6] C. Pires, J. Afonso, and F. Pereira, "Progressive still images coding; transform analysis," in *6th Mediterranean Electrotechnical Conference*, pp. 22–24, May 1991.

[7] S. Sridharan, A. Ginige, and D. Lowe, "Progressive image transmission," in *Proceedings of the International Conference on Image Processing and its Applications*, pp. 115–118, Mar. 1992.

[8] Y. Huang, H. M. Dreizen, and N. P. Galatsanos, "Prioritized DCT for compression and progressive transmission of images," *IEEE Transactions on Image Processing*, vol. 1, pp. 477–487, Oct. 1992.

[9] J. Li, J. Li, and C. J. Kuo, "Layerd DCT still image compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 440–443, April 1997.

[10] A. Jain and S. Panchanathan, "Scalable compression for image browsing," in *Proc. of International Conference on Consumer Electronics*, pp. 394–400, August 1994.

[11] C. Loeffler, A. Ligtenberg, and G. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in *ICASSP'89*, 1989.

[12] G. K. Wallace, "The jpeg still picture compression standard," *Communications of the ACM*, vol. 34, pp. 30–44, Apr. 1991.

[13] A. Said and W. A. Pearlman, “A new fast and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June 1996.

[14] T. Chiang and Y. Q. Zhang, “A new rate control scheme using quadratic rate distortion model,” *IEEE Transactions on Circuit and Systems for Video Technology*, vol. 7, pp. 246–250, Feb. 1997.

[15] ISO/IEC JTC1/SC29/WG1, *Call for contributions for JPEG 2000 (JTC 1.29.14, 15444): Image Coding System.* ISO/IEC JTC1/SC29/WG1, 1997.

[16] M. Unser, A. Aldroubi, and M. Eden, “B-spline signal processing: Part II-efficient design and applications,” *IEEE Transactions on Signal Processing*, vol. 41, pp. 834–848, Feb 1993.