

# **BLOCKING ARTIFACTS REMOVAL OF DISCRETE COSINE TRANSFORM BASED DECOMPRESSED IMAGES**

by

**YING LUO**

B.Sc., Electrical Engineering, Huazhong University of Science and Technology, Wuhan,  
China, 1991

**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 ENGINEERING**

**We accept this thesis as conforming  
to the required standard**

**THE UNIVERSITY OF BRITISH COLUMBIA**

**June 1999**

**© Ying Luo, 1999**

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

The University of British Columbia  
Vancouver, Canada

Date July 2 . 1999



# Abstract

Block-based DCT image/video compression methods have been successfully used in image and video compression applications due to their nearly optimal energy compaction property and relative ease of implementation. However, compression distortion becomes significant when these algorithms are used to code images or video under a certain bit rate.

The objective of this thesis is to offer post-processing solutions to reduce the block-based DCT image compression artifacts. The emphasis will be placed on the blocking artifacts since the blocking effect is one of the most noticeable degradations of block transform coding.

We first analyze the causes and properties of the blocking effect. Then we propose a robust method which detects the locations of the image block boundaries where this effect occurs. The detection results using this method are proved to be accurate at different compression bit rates. Several practical blocking artifacts removal techniques are then reviewed and analyzed. The performance of each technique is studied and evaluated.

Finally, a new blocking artifact reduction algorithm which is also effective at very low bit rate is proposed. Our algorithm takes advantage of the fact that the original pixel levels in the same block are of good continuity and we use this property and the correlation between the neighboring blocks to reduce the discontinuity of the pixels across the boundaries where blockiness appear. Our algorithm can highly preserve the high frequency components while smoothing out the boundary discontinuity. Simulation results show that the proposed algorithm significantly reduces the blocking artifacts in both the objective and the subjective measures. It can effectively remove blocking artifacts even at very low bit rates. The amount of computation it requires is also

acceptable compared to the iterative blocking artifact removal techniques.

# TABLE OF CONTENTS

<b>Abstract.....</b>	<b>ii</b>
<b>Table of Contents .....</b>	<b>iv</b>
<b>List of Tables .....</b>	<b>vii</b>
<b>List of Figures.....</b>	<b>viii</b>
<b>Acknowledgements .....</b>	<b>xi</b>
<b>1      Introduction .....</b>	<b>1</b>
1.1    Background .....	1
1.2    Thesis Objectives .....	2
1.3    Thesis outline .....	2
<b>2      Compression Artifacts .....</b>	<b>4</b>
2.1    Introduction .....	4
2.2    Human vision system .....	4
2.2.1    Weber ratio .....	4
2.2.2    Spatial frequency characteristics of the HVS .....	5
2.2.3    Color perception .....	6
2.3    Coding schemes .....	6
2.3.1    DCT transform coding .....	6
2.3.2    Motion representation and video compression .....	9
2.4    Visual artifacts in the compressed images and video streams .....	12
2.4.1    Blocking artifacts .....	12
2.4.2    Resolution loss .....	14
2.4.3    Ringing artifact .....	14
2.4.4    Mosquito artifact .....	14

2.4.5	Color bleeding and chrominance mismatch .....	17
<b>3</b>	<b>Blocking Artifacts Detection .....</b>	<b>18</b>
3.1	Introduction .....	18
3.2	Gradient threshold method .....	18
3.3	Difference of Slope (DOS) .....	19
3.4	Experimental results .....	22
3.5	Summary .....	32
<b>4</b>	<b>Blockiness Removal Techniques .....</b>	<b>33</b>
4.1	Introduction .....	33
4.2	Blockiness reduction schemes used in the encoding process .....	33
4.2.1	Block overlapping method .....	33
4.2.2	Lapped orthogonal transform .....	34
4.3	Blockiness reduction schemes used after the decoding process .....	35
4.3.1	Symmetric, two-dimensional filtering .....	35
4.3.2	Anisotropic filtering .....	42
4.3.3	Adaptive filtering using edge classification .....	49
4.3.4	Adaptive filtering using local statistical estimation .....	51
4.3.5	Iterative blocking artifact removal techniques .....	53
4.3.6	Dithering .....	53
<b>5</b>	<b>A New Blockiness Reduction Scheme .....</b>	<b>56</b>
5.1	Introduction .....	56
5.2	The frequency representation of the spatial block discontinuity in the DC domain .....	57
5.3	Recover the continuity of block C in the DCT domain .....	60
5.4	Examples .....	62
5.4.1	Examples 1 .....	62
5.4.2	Example 2 .....	64

5.5	Computation cost for blockiness reduction procedure in the DCT domain	67
5.5.1	The computation cost for calculating DCT coefficients	67
5.5.2	The computation cost for coefficients modification	70
5.5.3	The computation cost for inverse DCT	71
5.5.3	Summary	71
5.6	Spatial filtering	71
5.7	Results	73
<b>6</b>	<b>Conclusions and Future Work</b>	<b>81</b>
6.1	Compression artifacts classification	81
6.2	The design of blocking artifacts detector	81
6.3	Analysis of blocking artifacts removal techniques	82
6.4	A new blockiness reduction scheme	83
6.5	Future work	84
	<b>Bibliography</b>	<b>85</b>
	<b>Appendix A</b>	<b>87</b>

## List of Tables

4.1	Symmetric filter coefficients.....	36
4.2	PSNR comparison of Indian image before and after post-processing. ....	42
4.3	PSNR comparison of Lena image before and after post-processing. ....	42
4.4	PSNR comparison of Peppers image before and after post-processing.....	42
4.5	Anisotropic filter coefficients used to filter vertical boundary .....	43
4.6	Anisotropic filter coefficients used to filter horizontal boundary.....	43
4.7	PSNR comparison of Lena image before and after post-processing .....	49
4.8	PSNR comparison of Indian image before and after post-processing .....	49
4.9	PSNR comparison of Peppers image before and after post-processing.....	49
5.1	PSNR comparison of Lena image before and after post-processing .....	74
5.2	PSNR comparison of Indian image before and after post-processing .....	74
5.3	PSNR comparison of Peppers image before and after post-processing.....	74

# List of Figures

2.1	The frequency sensitivity of the eye to a sinusoidal grating [5].	5
2.2	Block diagram of DCT-Based Coding Algorithm [4]	8
2.3	Illustration of I-frames, P-frames, and B-frames	10
2.4	Encoder Prediction Loop [4].	11
2.5	Decoder block diagram [4]	11
2.6	Blocking artifacts	15
2.7	Magnified blocking artifacts	16
3.1	Sobel Operators	18
3.2	Difference of Slope	19
3.3	Original image and Blocking Artifacts	23
3.4	Detection result by Sobel Operators	24
3.5	Detection result by DOS operators	25
3.6	Original Image	26
3.7	Decompressed image with blocking artifacts	27
3.8	Detection result for original image by Sobel operators	28
3.9	Detection result for decompressed image by Sobel operators	29
3.10	Detection result for original image by DOS operator	30
3.11	Detection result for decompressed image by DOS operator	31
4.1a	Decompressed image, bitrate=0.298 bpp, PSNR=32.51 dB	38
4.1b	Post-processed image by symmetric filter, bitrate=0.298 bpp, PSNR=32.52 dB	39

4.2a	Decompressed image, bitrate=0.248 bpp, PSNR=31.2 dB .....	40
4.2b	Post-processed image by symmetric filter, bitrate=0.248 bpp, PSNR=31.52 dB..	41
4.3a	Decompressed image, bitrate=0.298 bpp, PSNR=32.51 dB .....	45
4.3b	Post-processed image by the anisotropic filters, bitrate=0.298 bpp, PSNR=33.04 dB.....	46
4.4a	Decompressed image, bitrate=0.248 bpp, PSNR=31.2 dB .....	47
4.4b	Post-processed image by the anisotropic filters, bitrate=0.248 bpp, PSNR=31.83 dB.....	48
5.1	Blockiness Diagram .....	58
5.2	Original block data of example 1 .....	63
5.3	DCT coefficients for blocks A, B and C.....	63
5.4	Modified coefficients of block C .....	64
5.5	Block data after modification.....	64
5.6	Original block data.of example 2.....	65
5.7	DCT coefficients of Blocks A and B .....	66
5.8	DCT coefficients of Block C before and after modification.....	66
5.9	Block data after modification.....	67
5.10	Blockiness Diagram .....	68
5.11	Fast 1-D 8 points DCT diagram.....	70
5.12	Epsilon filter.....	72
5.13a	Decompressed image, bitrate=0.298 bpp, PSNR=32.51 dB .....	75
5.13b	Post-processed image by our DCT domain filter (without spatial filtering), bitrate=0.298 bpp, PSNR=32.56 dB .....	76
5.13c	Post-processed image by our algorithm (with spatial filtering),	



bitrate=0.298 bpp, PSNR=32.56 dB .....	77
5.14a Decompressed image, bitrate=0.248 bpp, PSNR=31.2 dB .....	78
5.14b Post-processed image by our DCT domain filter (without spatial filtering), bitrate=0.248 bpp, PSNR=31.36 dB .....	79
5.14c Post-processed image by our algorithm (with spatial filtering), bitrate=0.248 bpp, PSNR=31.97 dB .....	80
A.1 Decompressed Lena image, bitrate=0.199 bpp, PSNR=30.40 dB .....	87
A.2 Post-processed Lena image, bitrate=0.199 bpp, PSNR=31.05 dB .....	88
A.3 Decompressed Lena image, bitrate=0.271 bpp, PSNR=31.95 dB .....	89
A.4 Post-processed Lena image, bitrate=0.271 bpp, PSNR=32.33 dB .....	90
A.5 Decompressed Peppers image, bitrate=0.15 bpp, PSNR=27.97 dB .....	91
A.6 Post-processed Peppers image, bitrate=0.15 bpp, PSNR=29.11 dB .....	92
A.7 Decompressed Peppers image, bitrate=0.22 bpp, PSNR=30.39 dB .....	93
A.8 Decompressed Peppers image, bitrate=0.22 bpp, PSNR=31.27 dB .....	94
A.9 Decompressed Indian image, bitrate=0.273 bpp, PSNR=28.17 dB .....	95
A.10 Post-processed Indian image, bitrate=0.273 bpp, PSNR=28.50 dB .....	96
A.11 Decompressed Indian image, bitrate=0.382 bpp, PSNR=29.65 dB .....	97
A.12 Post-processed Indian image, bitrate=0.382 bpp, PSNR=29.74 dB .....	98

## **Acknowledgment**

I would sincerely like to thank my supervisor, Dr. Rabab Ward, for her support and guidance throughout the research and the writing of this thesis. Her valuable advice, encouragement and patience are greatly appreciated.

Thanks also to all fellow graduate students who were always ready to share their knowledge and expertise.

# Chapter 1

## Introduction

### 1.1 Background

Over the past few decades, many digital applications have been taking place such as video conferencing, HDTV, DVD, etc. In the mean time, the demand for storing more images and transmitting more data with limited space and bandwidth keeps increasing. The popular solution to this problem is to compress the images by removing the redundancy in images so that they take less memory to store and less time to transmit. Image/video compression techniques offer better solutions to reduce file sizes while preserving the original content.

There are two forms of compression: lossless and lossy. Lossless compression removes the redundancy in images without removing any image detail. For this kind of compression the reconstructed images are identical to the original ones. Unlike lossless compression, lossy compression offers a trade-off between the amount of information lost and the degree of compression. Therefore lossy compression achieves far greater savings, albeit at the cost of some image degradation.

Block transform coding has been recognized for several decades as an attractive lossy compression technique. It takes advantage of the fact that unitary block transformations have the property of packing the signal energy. These transformations are able to concentrate the signal information in its lower spatial frequencies and most of the higher order spatial frequencies have zero or near-zero amplitude. After applying some state-of-art quantization and entropy coding

techniques, high compression rates are achieved. Entropy coding is lossless and has no influence on the spatial noise distribution in the decompressed image/video. It is quantization which provides the most amount of compression. This is the stage at which image/video information is lost. The compression ratio or alternatively the coding bit rate is controlled by the quantization step. By suitably choosing the quantization step value, a high compression ratio can be achieved without affecting the visual image quality. However, compression distortion becomes significant when the images are coded below a certain bit rate.

Block-based DCT coding has been successfully used in image and video compression applications due to its nearly optimal energy compaction property and relative ease of implementation. It provides the basis of two prevalent compression algorithms: JPEG and MPEG. JPEG which stands for Joint Photographic Experts Group is the standard algorithm for still image compression. MPEG which stands for Moving Picture Experts Group addresses the compression of video signals.

## **1.2 Thesis Objectives**

The objective of this thesis is to classify, analyze and offer solutions to the artifacts of DCT based decompressed images. Emphasis will be placed on the blocking artifacts removal since blocking effects cause the most noticeable degradation of block DCT transform coding.

## **1.3 Thesis outline**

In the next chapter, we are going to classify and characterize different types of transform coding artifacts. An effort will be made to reveal the causes of the artifacts.

In Chapter 3, we first examine the properties of the blocking artifacts, and use this

information to develop an objective criteria to detect blocking artifacts. The performance of this criteria will also be evaluated.

Several artifacts removal techniques will be introduced in Chapter 4. The performance of each technique will be studied and evaluated.

Finally, in Chapter 5, a new blocking artifacts reduction algorithm which is also effective at very low bit rate is proposed. This new approach is based on the analysis of the frequency distribution of the blockiness energy. The performance of the proposed method will be evaluated using both objective and subjective measures.

In the last chapter, conclusions about the coding artifacts reduction algorithm will be drawn, and possible future work brought out by this thesis research will be listed and described.

## **Chapter 2**

### **Compression Artifacts**

#### **2.1 Introduction**

In this chapter, we will first introduce some properties of the human visual system since these properties play a very important role in the perception of the artifacts. The sensitivity of the human visual perception to an artifact depends on the type of artifact. These artifacts are visually different from each other. The spatial activities and brightness surrounding an artifact area should be considered since they have masking effects on the artifact. To better understand the cause of compression artifacts, we will then briefly review DCT transform coding and other compression techniques. Finally, we will introduce and classify several compression artifacts. An attempt will be made to reveal the causes of the artifacts.

#### **2.2 Human visual system**

The human visual system has several important properties. These include brightness perception, spatial frequency response and color perception. These properties are considered for the purpose of improving the subjective visual quality in the encoding procedure. They should also be considered in the artifacts removal procedure since they affect the perception of the artifacts.

##### **2.2.1 Weber ratio**

The Weber ratio provides the key factor to the brightness-intensity relationship. It indicates the functional relationship between brightness and intensity. Assuming  $I$  is the intensity of

the background and  $\Delta I$  is the difference value at which the contrast between two different intensities is just noticeable then the ratio of  $\Delta I$  to  $I$  is roughly a constant [6] which is

$$\frac{\Delta I}{I} \approx 0.02. \quad (2.1)$$

Weber ratio indicates that  $\Delta I$  is proportional to  $I$ . This means the sensitivity of the human visual system to image detail is a function of the local average brightness. Loss of detail in dark areas of the image is less visible than in brighter areas.

### 2.2.2 Spatial frequency characteristics of the HVS

The human visual system resembles a lowpass/bandpass filter. The frequency sensitivity of the eye to a sinusoidal grating is shown in Figure 2.1. This curve suggests that the human visual system is most sensitive to mid-frequencies and least sensitive to high frequencies [5]. Therefore, image details presented in areas with high spatial activities such as textured areas are much less visible than those in relatively flat areas.

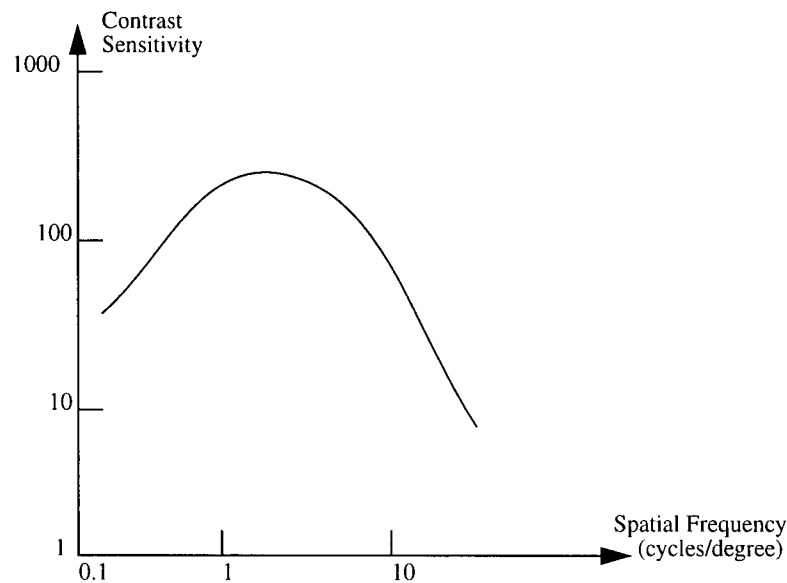


Figure 2.1 The frequency sensitivity of the eye to a sinusoidal grating [5].

## 2.2.3 Color perception

The human visual system responds differently to the luminance and chrominance components of an image. It is less sensitive to high frequencies in the chrominance components than to high frequencies in the luminance component.

## 2.3 Coding schemes

### 2.3.1 DCT transform coding

The general image transform coding strategy can be summarized as follows. A digital image is first divided into subimages of 8x8 pixels known as blocks. Then each block undergoes a two-dimensional discrete transform. The discrete cosine transform (DCT) has become the most widely used discrete transform in today's transform image coding systems because of its near optimal compaction property, and because it can be implemented very efficiently by fast algorithms. The  $8 \times 8$  DCT is defined as following:

$$X[k_1, k_2] = \frac{1}{4} \sum_{n_1=0}^7 \sum_{n_2=0}^7 C_{k_1} C_{k_2} x[n_1, n_2] \cos\left[\frac{(2n_1+1)k_1\pi}{16}\right] \cos\left[\frac{(2n_2+1)k_2\pi}{16}\right], \quad (2.2)$$

and its inverse DCT is given by

$$x[n_1, n_2] = \frac{1}{4} \sum_{k_1=0}^7 \sum_{k_2=0}^7 C_{k_1} C_{k_2} X[k_1, k_2] \cos\left[\frac{(2n_1+1)k_1\pi}{16}\right] \cos\left[\frac{(2n_2+1)k_2\pi}{16}\right], \quad (2.3)$$

where

$$C_{k_1}, C_{k_2} = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } k_1, k_2 = 0 \\ 1 & \text{otherwise} \end{cases} \quad (2.4)$$

Note that the two dimensional DCT is a real and separable transform which is obtained by a 1-D



transformation carried on the rows of the image block followed by a 1-D transformation of the resulting columns [6].

The block transform coefficients reflect the frequency characteristics of the image block. The upper left corner of the transformed block corresponds to the low-frequency coefficients of the DCT, while the bottom right corner corresponds to the highest frequencies. Due to the fact that natural images tend to have more energy at low frequencies, the energy of the transformed block is concentrated in the upper left corner of the block. Most of the higher order spatial frequencies have zero or near-zero amplitudes. To achieve high data compression, only the dominant coefficients are retained. Therefore, a quantization weighting matrix is used to round the coefficients to the nearest integer and most of the high order coefficients are rounded to zero. This weighting matrix is also called "Q-table". The Q-table controls the quantization step-size, which is related to the bit rate. By scaling the Q-table, different number of coefficients are retained and thus different degrees of compression can be realized. High compression ratios can be achieved without visibly affecting the image quality by choosing suitable quantization steps. Since humans are more sensitive to errors in the low frequency components of the image than those in the high frequencies, the Q-table has larger values in its entries that corresponding to the high frequencies than those that correspond to low frequencies. As a result, high frequency components are quantized more coarsely than low frequency components. A quantization matrix used in the TM5 codec is shown in (2.5).

$$\begin{bmatrix}
 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\
 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\
 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\
 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\
 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\
 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\
 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\
 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83
 \end{bmatrix}
 \quad (2.5)$$

After the quantization, the DC coefficients are coded differentially in order to take into account the strong correlation between the DC coefficients of neighboring blocks. The AC coefficients are all entropy coded on a block-by-block basis. The compressed image is either transmitted or stored depending on the application. Decoding constitutes an inverse operation of this procedure. General block diagrams of the transform encoding and decoding are shown in Figure 2.2.

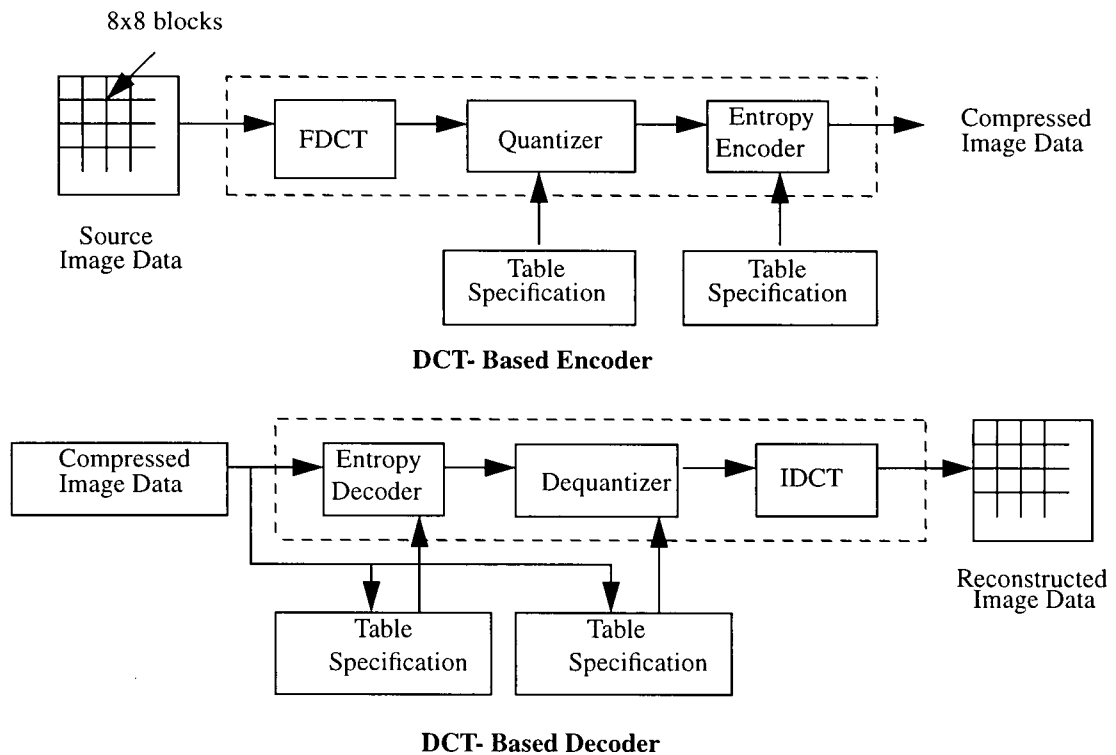


Figure 2.2 Block diagram of DCT-Based Coding Algorithm [4].

The DCT-Based coding forms the basis of the JPEG algorithm. This coding type is also used as the basis of spatial redundancy reduction in the MPEG algorithm.

### **2.3.2 Motion representation and video compression**

A video sequence can be considered as a sequence of still images. If a frame is encoded independently of other frames, then this frame coding is called intraframe coding. Intraframes provide the capability of accessing any random frame in stored video. However, for motion pictures, high compression ratios can not be achieved by encoding every frame as an intraframe since this does not take into account the extensive temporal redundancy present in a video sequence. Thus MPEG's basic scheme is to predict motion from frame to frame in the temporal direction, and then to encode the differences in the frames. This encoding is called interframe encoding which involves coding a frame in reference to a previous or a future frame.

In order to reduce the temporal redundancy, two types of interframe coding techniques are suggested: forward prediction and bidirectional prediction. Thus, three types of pictures are considered in MPEG: Intraframes (I-frames), predicted frames (P-frames) and bidirectional frames (B-frames). Intraframes are typically distributed periodically throughout the consecutive frames of a video sequence. They are used as random access points and they provide the lowest compression ratios. A P-frame is coded using forward prediction which is based on a previous reference frame. P-frames have significantly higher compression ratios than those of I-frames. Bidirectional frames are coded using forward, backward prediction, or interpolation. B-frames provide the highest amount of compression but requires both a past and a future reference frame for prediction. Figure 2.3 illustrates this coding scheme.

Motion prediction is carried on the luminance channel on 16x16 pixels macroblocks basis.

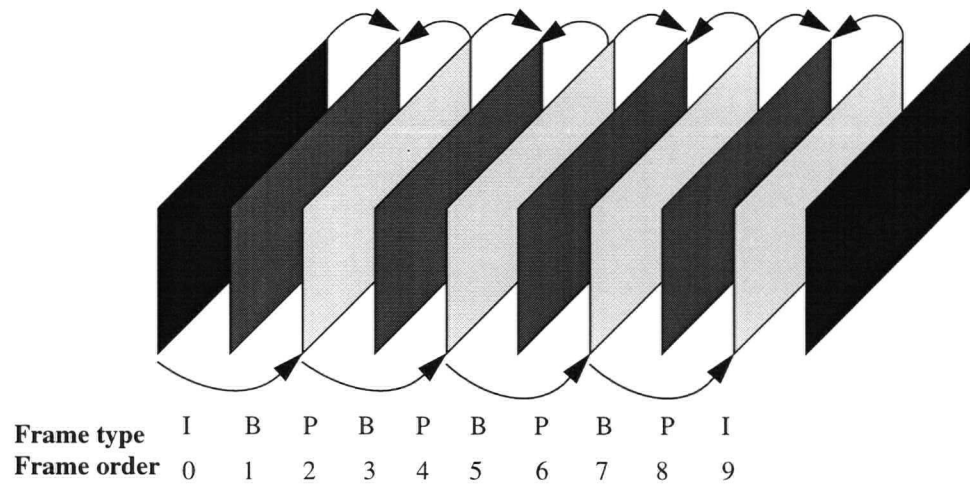


Figure 2.3 Illustration of I-frames, P-frames, and B-frames.

Given a macroblock in a current B frame which is being encoded, a close match to that macroblock is searched in a previous, a future frame or both. For a P-frame, only the previous frame is searched. The corresponding macroblock-sized region in the reference frame can be offset in its position to that of the considered macroblock in the current frame. These macroblock offsets are known as motion vectors. Therefore, given a set of motion vectors, one vector for each macroblock, we can use macroblocks in a previous, a future frame or both to represent the image in the current frame. Since the motion compensated prediction is not able to represent the image accurately, the prediction error, i.e. the difference between the predicted picture values and those of the original pictures needs to be coded. The spatial signal of the prediction error, as well as the intraframe coded picture are then DCT encoded. DCT are done on 8x8 block basis, and the DCT coefficients are quantized by a quantizer. In the quantization process, the precision of each of the transform coefficients is adjusted to its relative importance to the human visual perception. This relative coefficient precision information is represented by a quantizer matrix. Each value in the quantizer matrix represents the coarseness of quantization of the related DCT coefficient. The

degree of coarseness of the entire macroblock is adjusted by the scale factor of the quantizer. The macroblocks which are deemed to be visually less important, such as regions with large spatial activity, can be quantized more coarsely by using large scale factors. After the quantization, the quantized DCT coefficients, the motion vectors, and the quantization parameters are further compressed using a combination of run-length and entropy coding. The encoder and decoder diagrams are shown in Figure 2.4 and Figure 2.5, respectively.

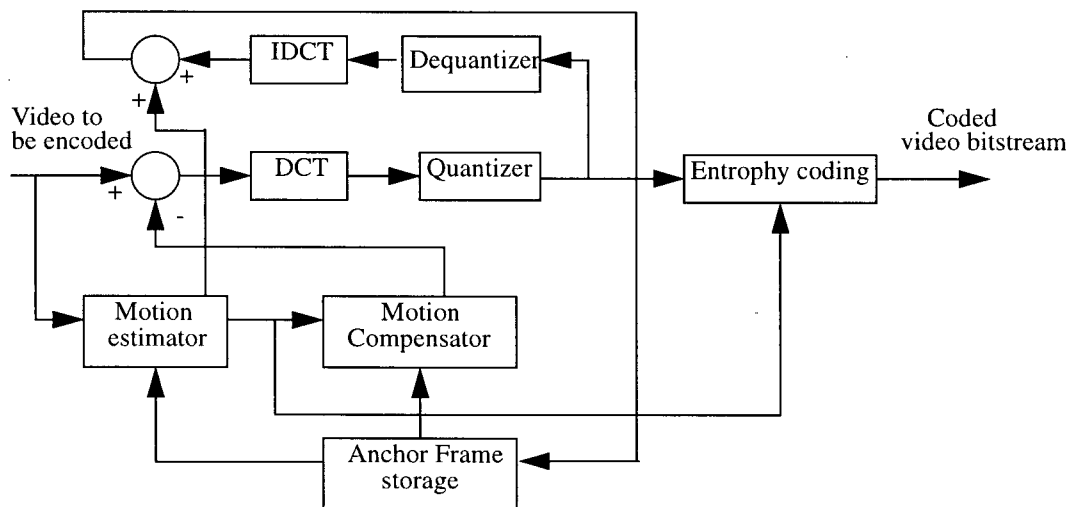


Figure 2.4 Encoder Prediction Loop [4].

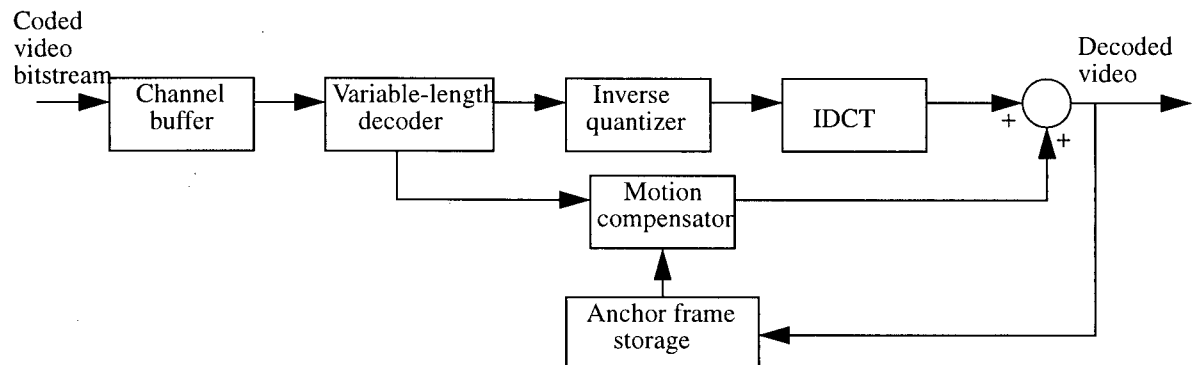


Figure 2.5 Decoder block diagram [4].

## 2.4 Visual artifacts in the compressed images and video streams

To provide users with quality image/video services, causes of these artifacts and their effects on viewing quality need to be understood [8].

### 2.4.1 Blocking artifacts

Due to image block segmentation and the subsequent independent transform coding of the blocks, existing correlations among adjacent block pixels in a frame are not pursued. The signal continuities between blocks are not always guaranteed and the discontinuity between two blocks may become visible at the boundary between those two blocks. This artifact is called the “blocking effect,” which is more prominent at low-bit rate coding. This kind of edge artifact is composed only of horizontal and vertical edges at the boundaries between two blocks. The image reconstructed by transform image coding at 0.19 bit per pixel(bpp) is shown in Figure 2.6, where the blocking effect can be easily noticed and is visually unpleasant.

At low bit rates, the high frequency components of an image cannot be adequately represented in the transform domain since many of the higher order transform coefficients are quantized to zero. For a slowly varying region with no image edges, blocking artifacts appear in the reconstructed image as a regular pattern of visible block boundaries. In a complex image region, besides the sharp boundary intensity change, the original image edges appear irregular. The regular pattern and their fixed grid size make them visually prominent.

Blocking artifacts can also be caused by motion compensation. Starting with an empty present frame, the predicted macroblocks are offset by the motion vectors from the reference frame to the present frame, the resultant image is composed of copied macroblocks and obviously

have blocking effect. This effect can be removed by adding the prediction error image. However, if this prediction error image is inadequately coded, blocking effects remain in the reconstructed images.

Typically, there are three types of blocking artifacts according to human visual perception. They are described briefly below.

#### **2.4.1.1 Staircase noise**

Staircase noise arises in an image region where there is an isolated, clearly distinguishable diagonal edge or feature. The diagonal edges that used to be smooth in the original image are represented by a number of horizontal or vertical steps (Figure 2.7a) in the reconstructed image. This is because edges can not be well reconstructed by the coarsely quantized coefficients and because each image block is processed independently. As a result, the continuity of an edge across a block boundary is not assured.

#### **2.4.1.2 Grid noise and false contouring**

Grid noise (Figure 2.7b) appears as a regular pattern of visible grid in the relative flat areas of the image. False contouring appears as abrupt intensity change in the reconstructed image in areas with gradually changing luminance. In these two kinds of regions, the variations in luminance are small, and so, most of the signal energy is concentrated in the DC coefficient. Due to the coarse quantization, the AC coefficients represent the gradual change in luminance are quantized to zero. Without these AC coefficients, an abrupt change in luminance across the block boundary is produced in the reconstructed image, which is perceived as a patchiness or contouring effect.

### **2.4.1.3 Texture noise**

Texture noise (Figure 2.7c) is a type of quantization distortion which appears in textured image regions, such as a collection of randomly-oriented edges. It can be a combination of staircase and grid noise. Due to the masking property of the human visual system in areas of high detail, the staircase effect in texture regions is not as prominent as that in strong edge regions. Instead, it appears as loss of detail and smearing.

### **2.4.2 Resolution loss**

Resolution loss is caused by lowpass filtering or equivalently by suppression of higher order DCT coefficients through coarse quantization. It appears as a loss of spatial detail in the reconstructed images.

### **2.4.3 Ringing artifact**

The ringing effect is visually prominent along high contrast edges with generally smooth background in the reconstructed image. Strong rippling may result around a high contrast edge. This appears as shimmering or rippling outwards from the edge up to the block's boundary [8]. Ringing artifacts can not be predicted easily. They can result in edge artifacts at object edges with any orientation. They may consist of a series of edges located very close to each other and their visibility can be affected by the masking effects caused by surrounding high frequency contents.

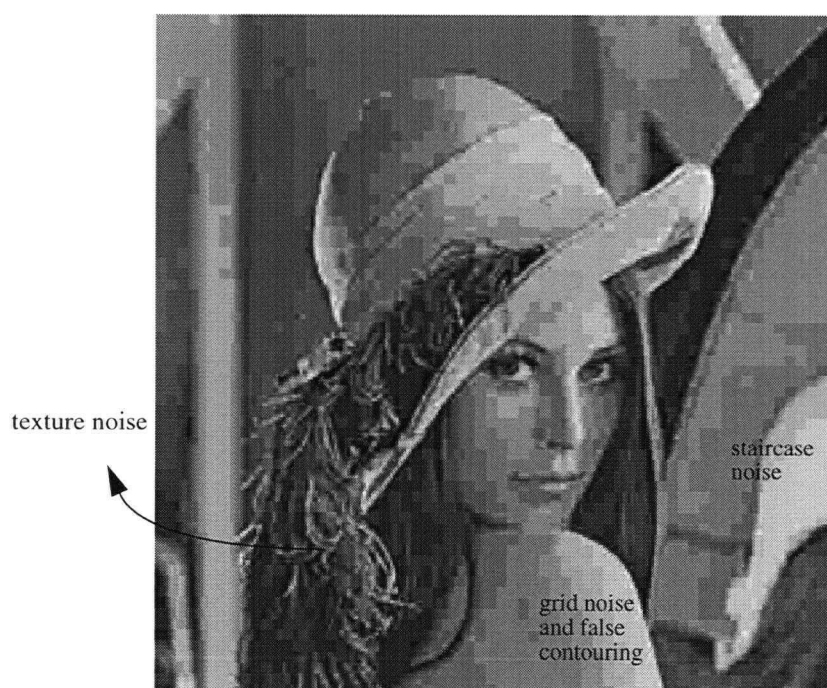
### **2.4.4 Mosquito artifact**

The mosquito artifacts are temporal artifacts related to high frequency distortion and coding inaccuracies. They are seen as fluctuations of luminance chrominance levels in a video sequence. They usually occur around sharp edges with smooth background or in relative



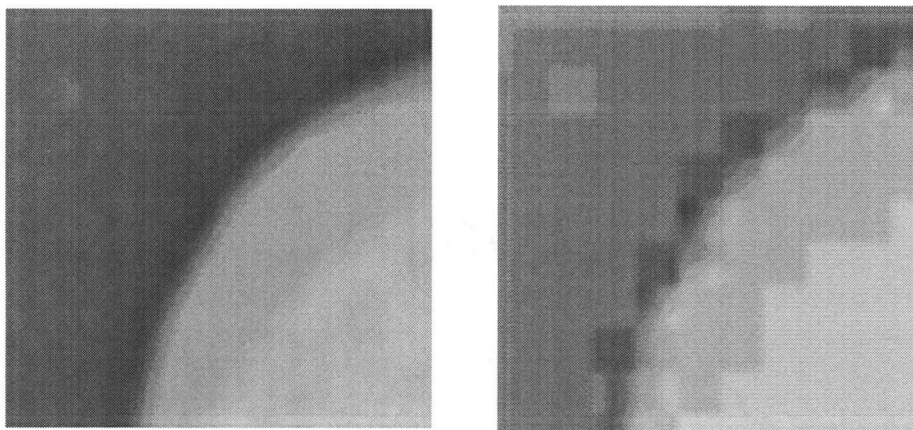


Original image

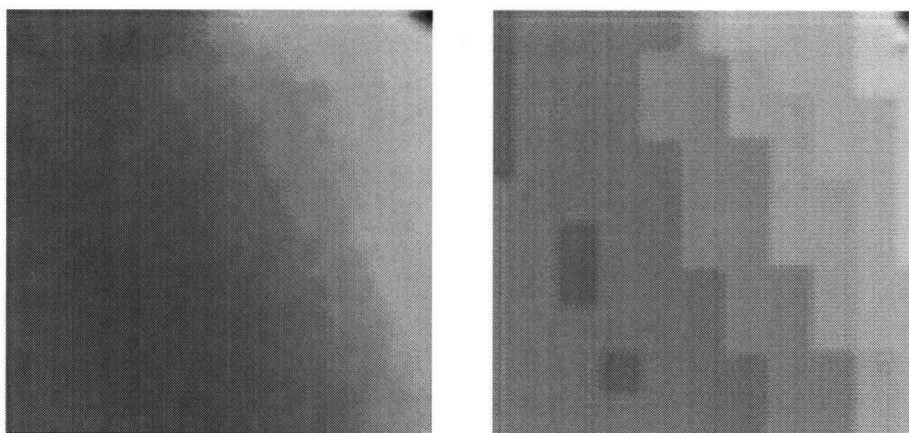


Reconstructed image, bitrate=0.19 bpp

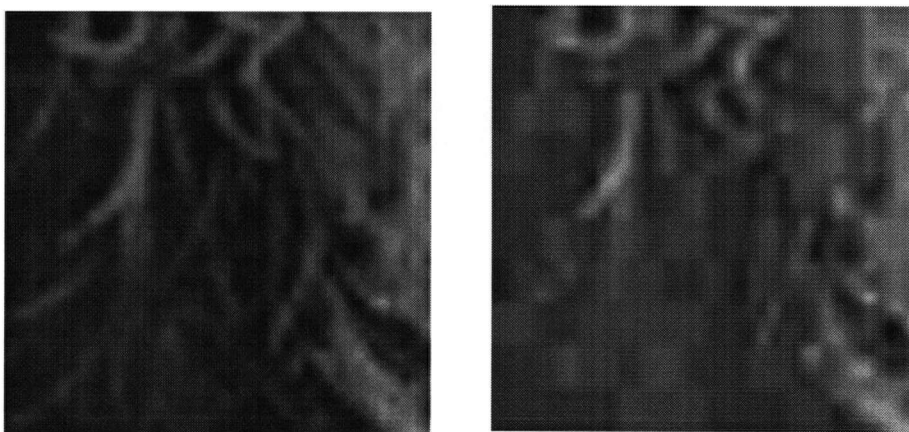
Figure 2.6 Blocking artifacts.



(a) Staircase noise



(b) Grid noise and false contouring



(c) Texture noise

Figure 2.7 Magnified blocking artifacts.

stationary areas containing high spatial activities.

#### **2.4.5 Color bleeding and chrominance mismatch**

Color bleeding is a kind of color artifact due to the chrominance information loss. It results in a smearing of the color between areas with high contrast chrominance values. Chrominance mismatch appears as color misplacement of a macroblock with respect to the color of the surrounding area. This is caused by motion compensation since motion compensation is carried out using luminance information only.

## Chapter 3

### Blocking Artifacts Detection

#### 3.1 Introduction

As most image artifacts occur in certain places in an image, artifacts detection is an important step to be carried out prior to the removal or reduction of the artifact intensity. It can also be used to evaluate the image/video compression quality. In this chapter, we focus on blockiness detection.

#### 3.2 Gradient threshold method

As mentioned earlier, blocking effects appear as discontinuities across block boundaries. Since blocking artifacts have the property of having precise spatial localization, they are easier to detect than other impairments. However, usually not all block boundaries have blocking appearances or effects. Therefore, one common method of detecting the block boundaries where blocking effects occur is to apply gradient operators such as sobel filters [5] (Figure 3.1) along the block boundaries. Operator (a) is used to detect vertical edges and operator (b) is used to detect horizontal edges. The region with high gradient is considered as having edges. Strong edges are detected by thresholding the gradients.

$$\begin{array}{cc} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \\ \text{(a)} & \text{(b)} \end{array}$$

Figure 3.1 Sobel Operators.

With such a method, both the genuine edges along the block boundaries as well as the blocking artifacts will be detected. Therefore we must distinguish real edges from false edges caused by blocking artifacts. To solve this problem, we introduce another metric which is similar to the one defined in [11].

### 3.3 Difference of Slope (DOS)

Although both the genuine edges in an image and the blocking artifacts have large gradients, they are different in appearance. For natural images, the transition of the intensity can be considered to be regular as shown in Figure 3.2 and edges are unlikely to line up with block boundaries exactly. However, blocking artifacts manifest themselves as abrupt intensity changes across block boundaries and have regular patterns.

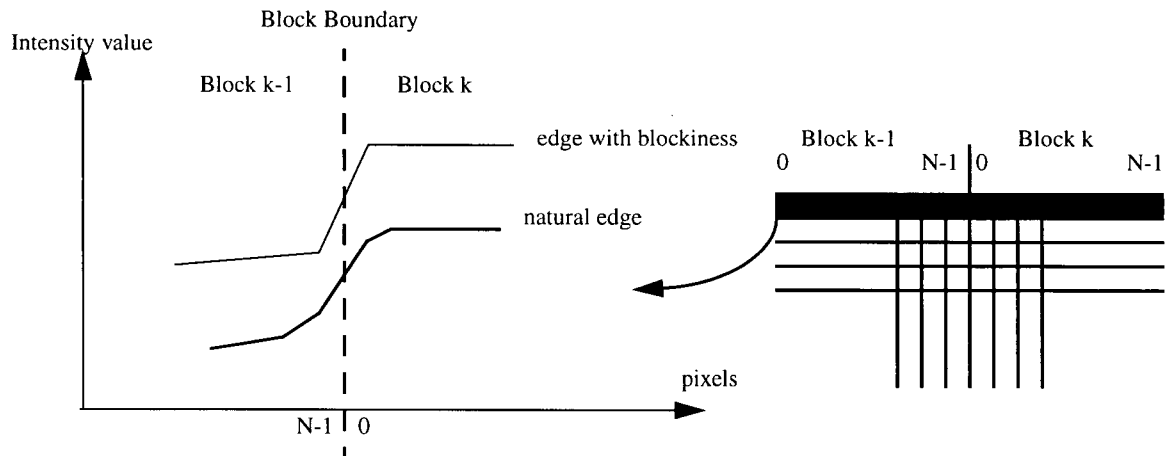


Figure 3.2 Difference of Slope.

The intensity slope between two adjacent pixels reflects the intensity change between them. Thus within each block the degree of intensity continuity close to the block's boundary can be measured by this intensity slope. For two horizontally adjacent blocks  $k-1$  and  $k$ , the slope for

the  $k-1$  block at the  $i$ th row and for the last  $N-1$  and  $N-2$  column positions of the block is defined by:

$$d_{k-1}(i) = x_{k-1}(i, N-1) - x_{k-1}(i, N-2), \quad (3.6)$$

where  $x_{k-1}$  denotes the intensities of block  $k-1$ . For the  $k$ th block, and for the 0 and first column positions of the  $i$ th row, the intensity slope is:

$$d_k(i) = x_k(i, 1) - x_k(i, 0). \quad (3.7)$$

It is also reasonable to assume that the average of these two slopes reflects the degree of continuity within these two adjacent blocks. This average is defined by:

$$\bar{D}(i) = [d_{k-1}(i) + d_k(i)]/2 \quad (3.8)$$

Similarly, the degree of intensity continuity across two horizontally adjacent blocks can be measured by the intensity slope across their boundary. This quantity is given by:

$$\tilde{D}(i) = x_k(i, 0) - x_{k-1}(i, N-1) \quad (3.9)$$

For natural images, the transition of the intensity is regular. Thus, the value of  $\tilde{D}(i)$  and  $\bar{D}(i)$  should be similar. For JPEG or MPEG compressed images, the intensity continuity is preserved within each block but it is not ensured at the boundary between any two blocks.  $\tilde{D}(i)$  increases a lot compared to  $\bar{D}(i)$  due to blocking artifacts. Therefore, the difference between  $\tilde{D}(i)$  and  $\bar{D}(i)$  forms a good estimate of the blocking artifacts. This estimate is defined by:

$$\begin{aligned} \epsilon(i) &= \bar{D}(i) - \tilde{D}(i) \\ &= [x_k(i, 0) - x_{k-1}(i, N-1)] - [d_{k-1}(i) + d_k(i)]/2 \end{aligned}$$

$$= \frac{3}{2}x_k(i, 0) - \frac{1}{2}x_k(i, 1) - \frac{3}{2}x_{k-1}(i, N-1) + \frac{1}{2}x_{k-1}(i, N-2) \quad (3.10)$$

Thus the original image may have a large gradient across the boundary of two blocks, but  $\epsilon$ , the difference of slope, is small. Blocking artifacts manifest themselves as regular patterns,

thus they can be detected by thresholding  $\sum_{i=0}^{N-1} \epsilon(i)$ , the sum of  $\epsilon(i)$  across the boundary. The

threshold should be adaptive to the local brightness according to Weber's law.

As mentioned in the previous chapter, the visibility of blockiness also depends on the local spatial activity. Blocking artifacts are usually prominent at regions with low local spatial activities, while the regions which have genuine edges usually have high local spatial activities. Thus local spatial activity can be used in a criterion to distinguish genuine edges from those due to blocking artifacts. An estimate of the local spatial activity is the local variance. However, this estimate require much computational effort. To avoid having to carry out more computations, we use the variation of  $\epsilon$  to measure this local spatial activities. Let

$$d_{max} = MAX(\epsilon(i)), i = 0, \dots, N, \quad (3.11)$$

and

$$d_{min} = MIN(\epsilon(i)), i = 0, \dots, N, \quad (3.12)$$

Then if the variation

$$d_{max} - d_{min} < T_1, \quad (3.13)$$

the region is considered as having low spatial activity, otherwise, it is considered as having high spatial activity and post filtering such as low pass filtering will not be performed in the region with high spatial activity so to avoid unwanted smoothing.

In summary, the blockiness detection is based on two measurements:

$$1 > |d_{max} - d_{min}| < T_1,$$

$$2 > \left| \sum_{i=0}^{N-1} \varepsilon(i) \right| > T_2.$$

When the blockiness is visible, the variations of the intensity across the boundaries should have similar pattern (either increasing or decreasing). This will make the first measurement very small and the second one very large. And thus this boundary is marked as having a blocking artifact.

### 3.4 Experimental results

According to several tests, the detections using our difference of slope, DOS operators are rather accurate. Sobel operators on the other hand can hardly tell the difference between real edges and blocking artifacts. Figure 3.4a shows the edge map detected from the original image (Figure 3.3a) by Sobel operators. Obviously, these are genuine edges which are faultily identified as blocking artifacts. Figure 3.4b shows the edge map detected from decompressed image (Figure 3.3b) by Sobel operators. The detected edges include genuine edges and those due to blocking artifacts.

If we apply DOS operators to the original image, then no edges (blocking artifacts) are detected (Figure 3.5a). If we apply DOS operators to the image with blocking artifacts (Figure 3.3b), edges due to blocking artifacts are accurately detected (Figure 3.5b). Another set of detection results using the Sobel operators and the DOS operator is shown in Figure 3.6 to Figure 3.11.



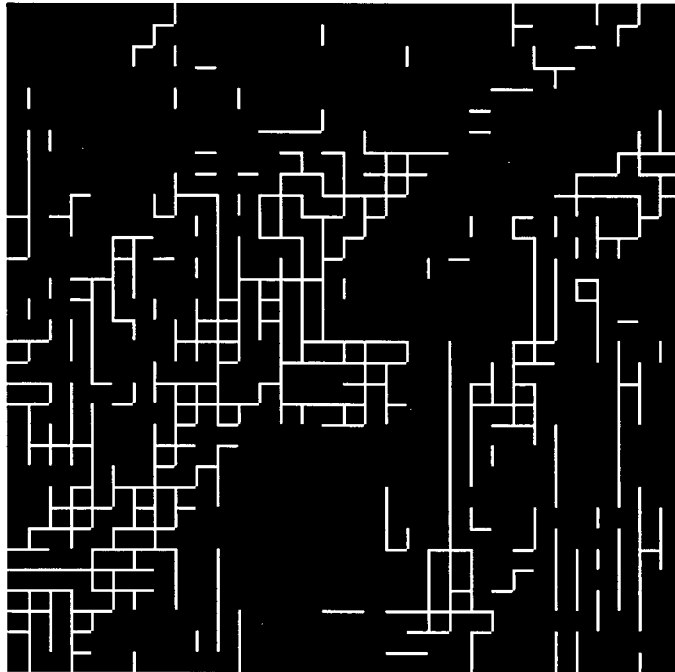


(a) Original Image

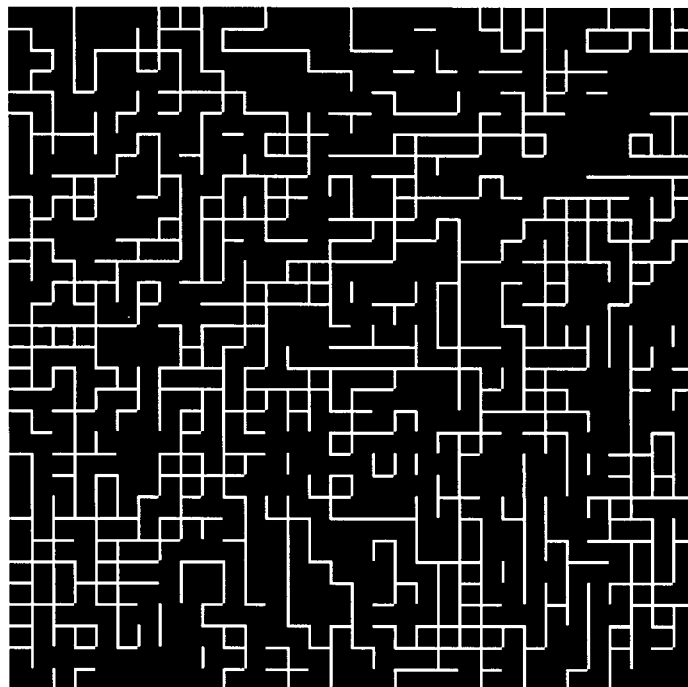


(b) Blocking Artifacts

Figure 3.3 Original image and Blocking Artifacts.

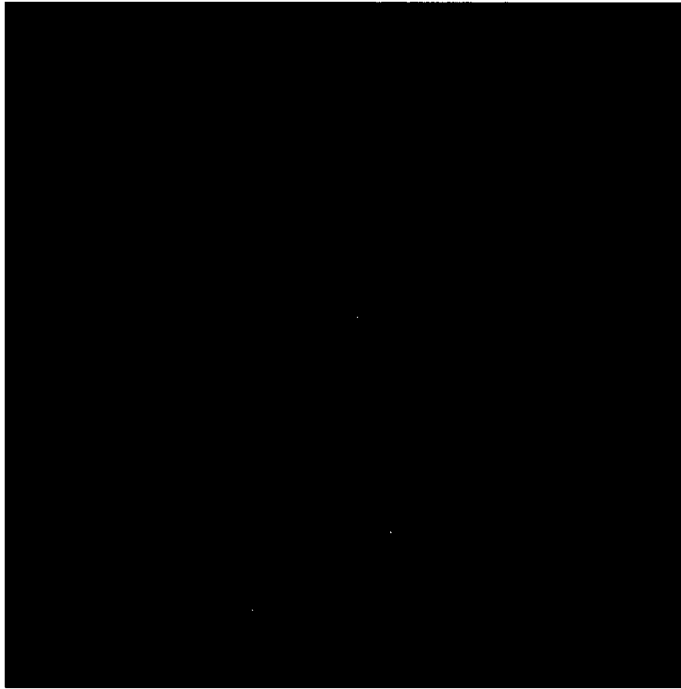


(a) Edges in the original image detected by Sobel operators

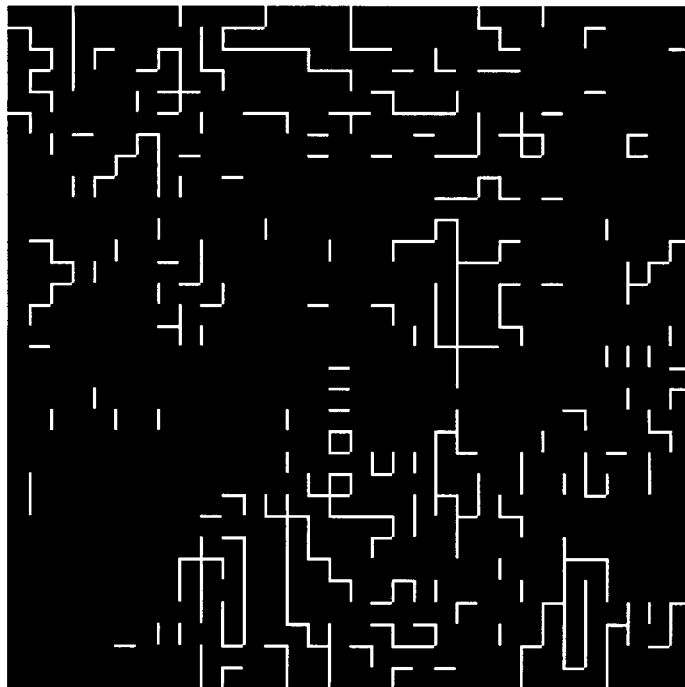


(b) Blocking Artifacts detected by Sobel operators

Figure 3.4 Detection result by Sobel Operators.



(a) No blocking artifacts is detected in the original image by DOS operators



(b) Blocking artifacts detected by DOS operators

Figure 3.5 Detection result by DOS operators.

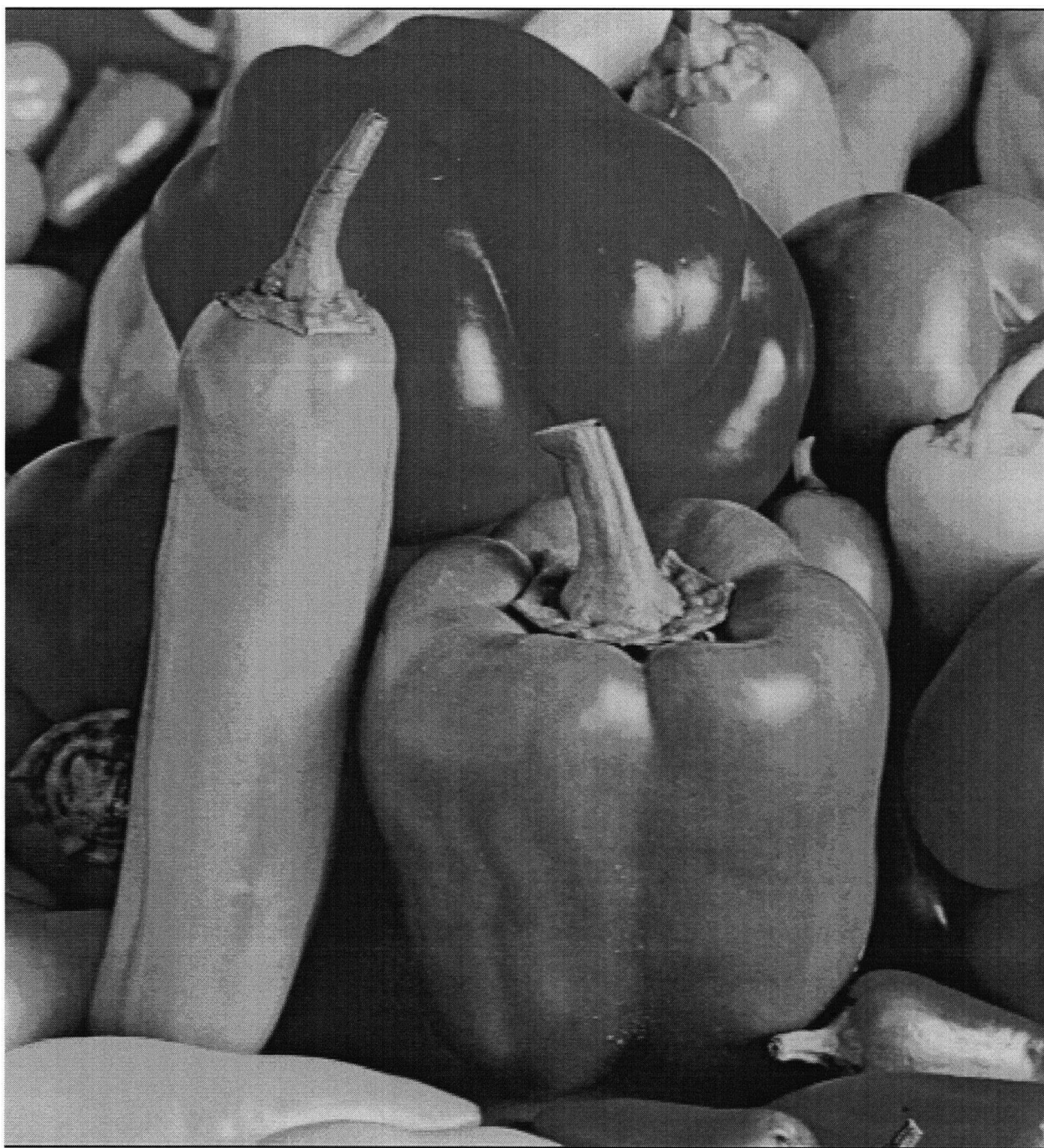


Figure 3.6 Original Image.

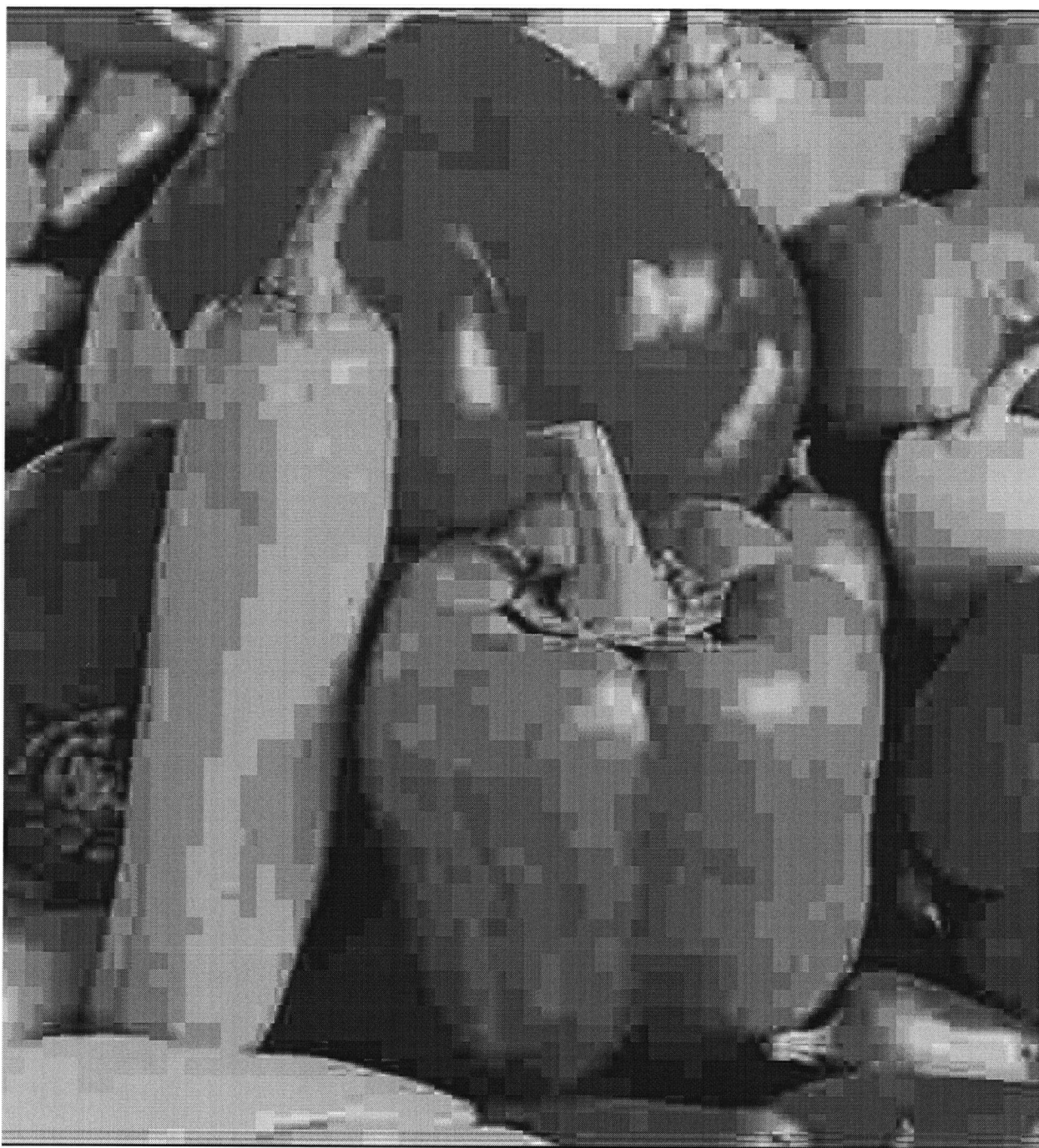


Figure 3.7 Decompressed image with blocking artifacts.

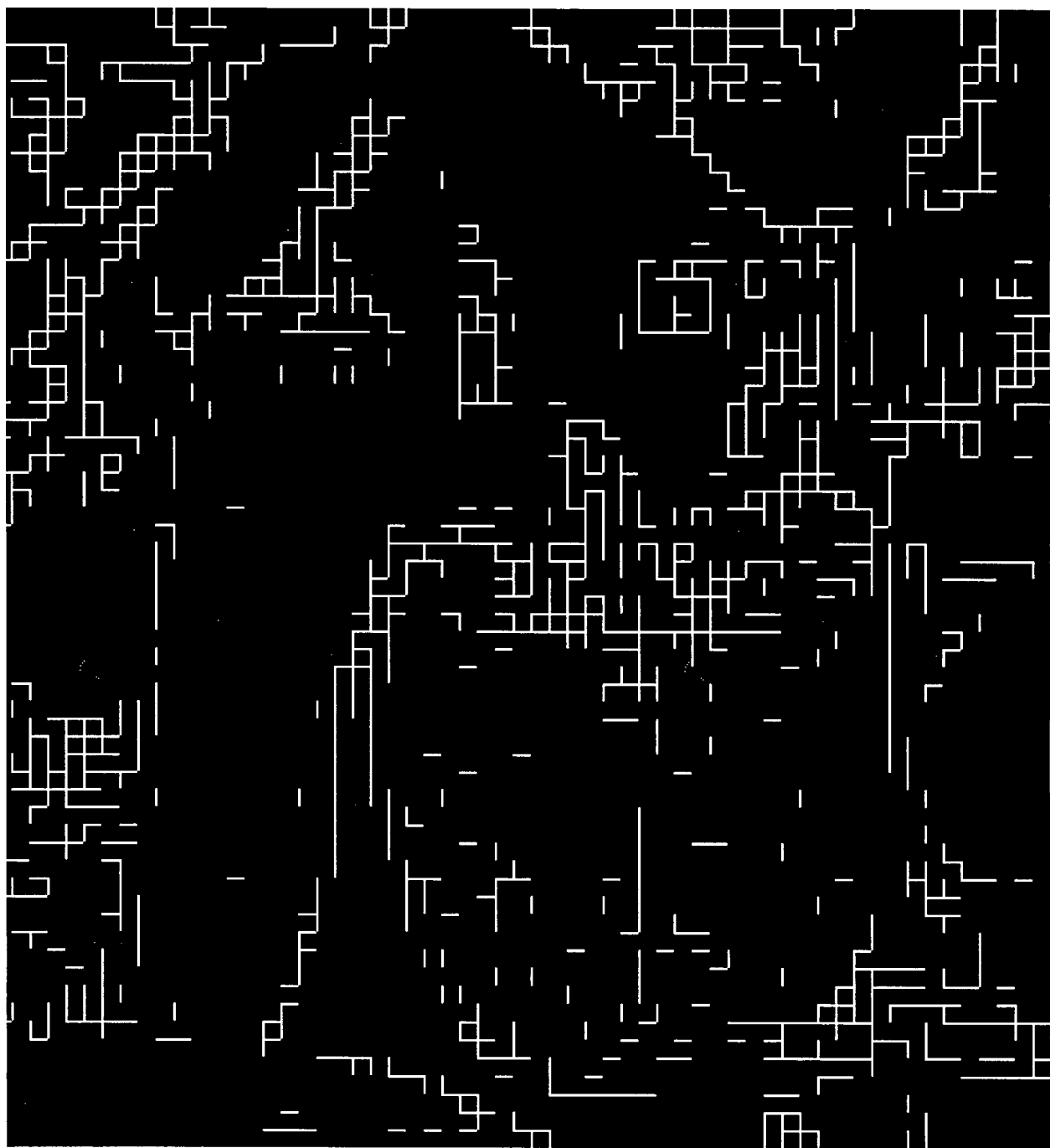


Figure 3.8 Detection result for original image by Sobel operators.

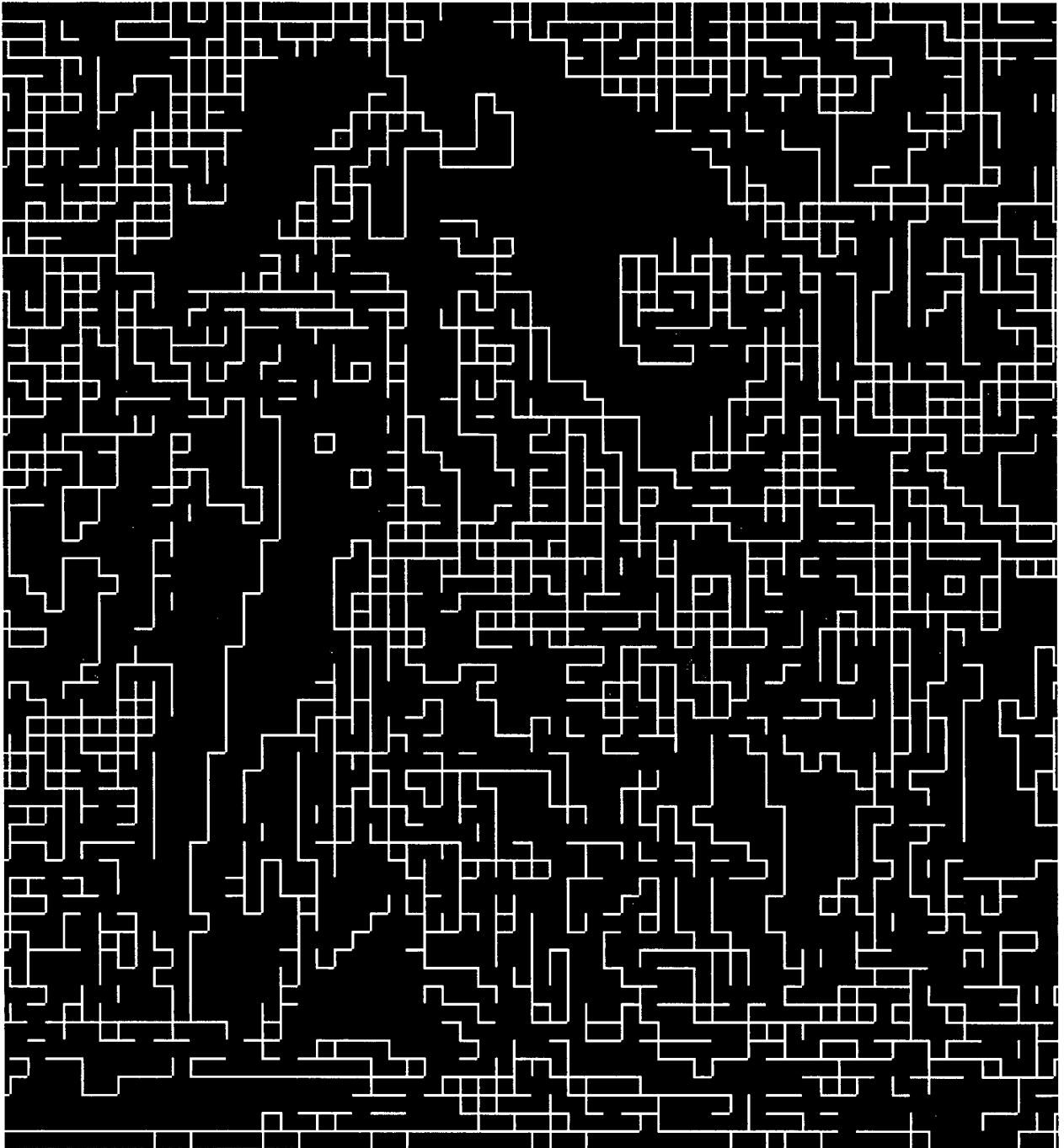


Figure 3.9 Detection result for decompressed image by Sobel operators.

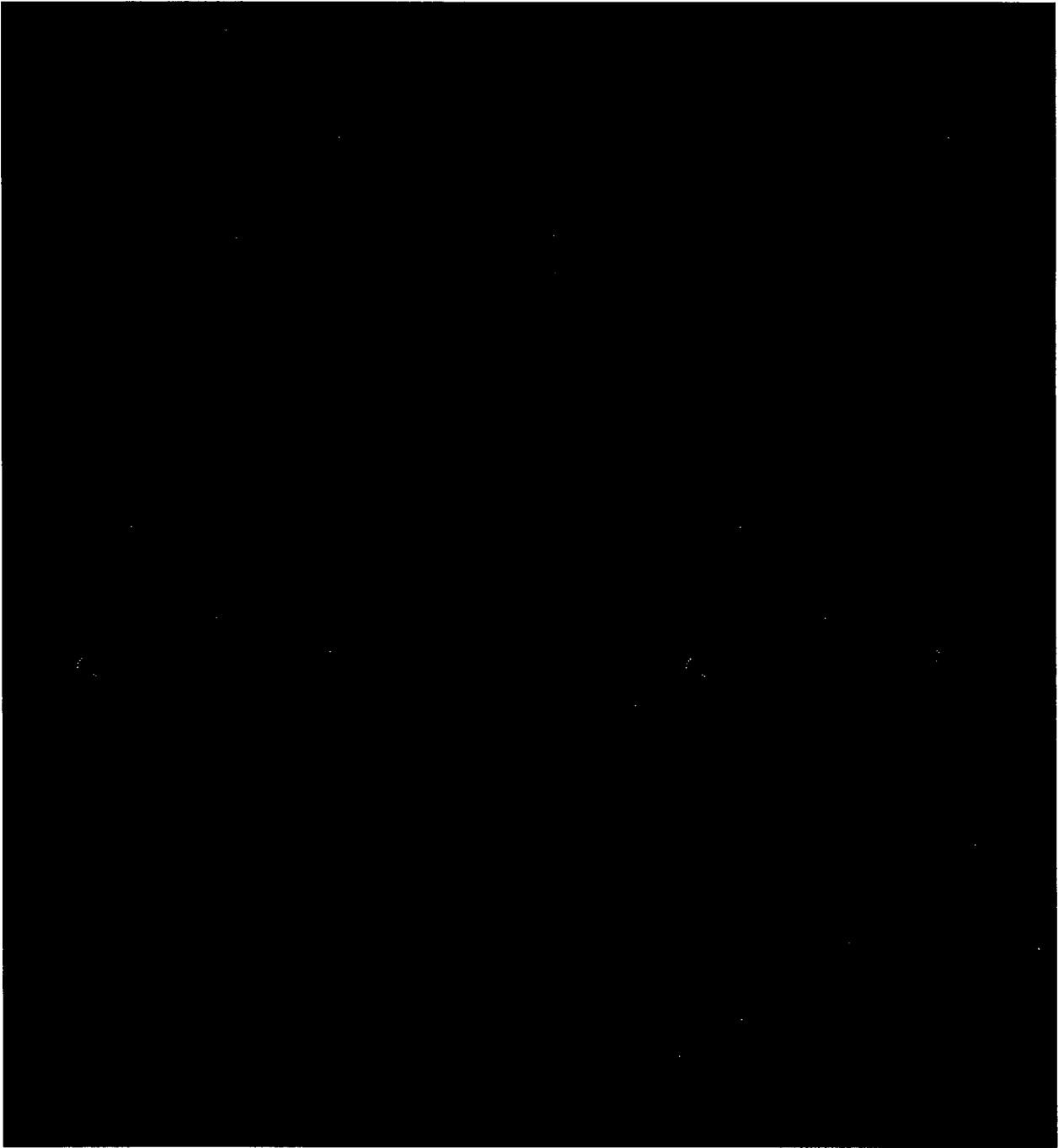


Figure 3.10 Detection result for original image by DOS operator.



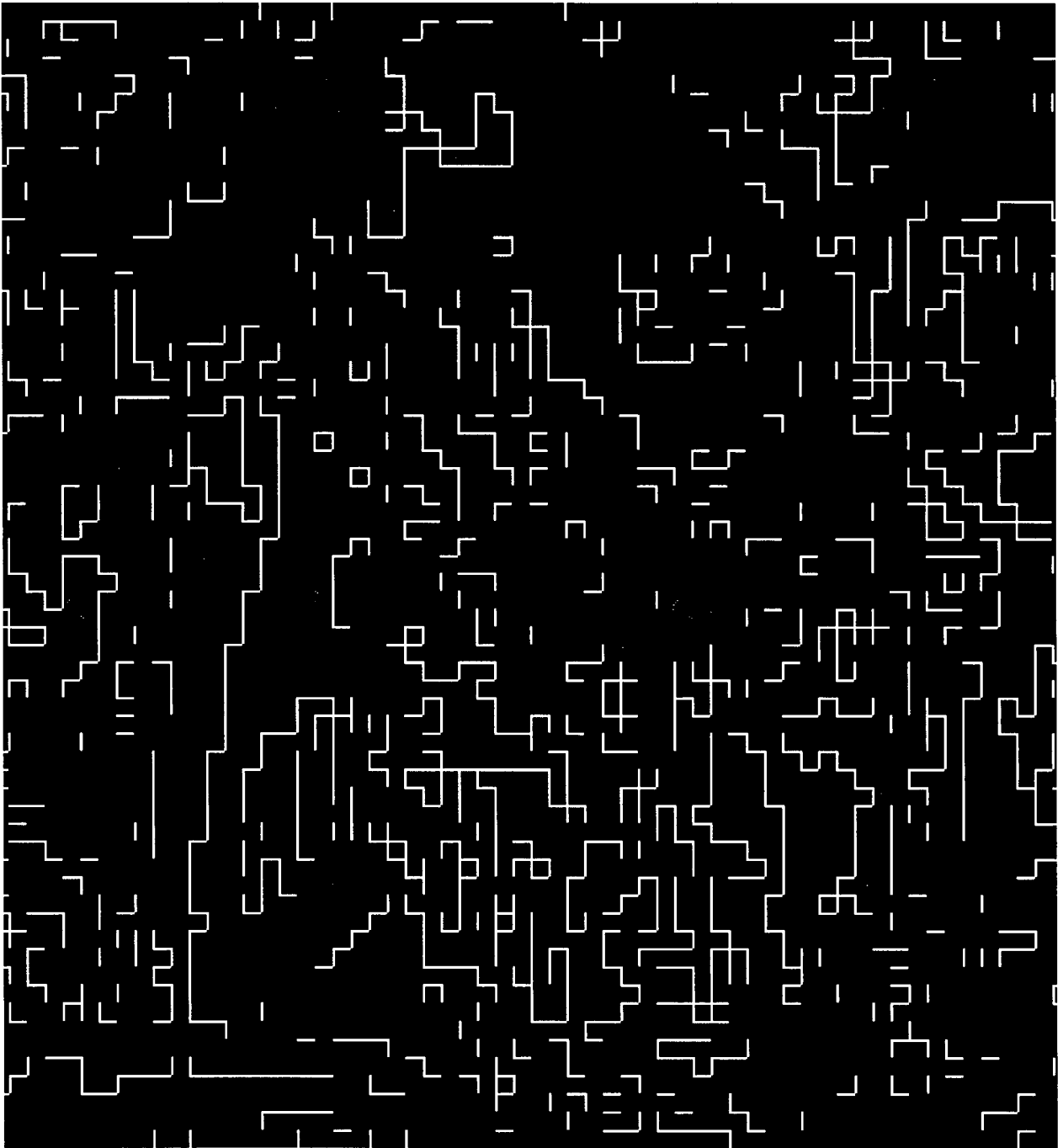


Figure 3.11 Detection result for decompressed image by DOS operator.

### 3.5 Summary

In this chapter we have shown how edges due to blocking artifacts can be detected. We first showed that using Sobel operators, the genuine edges as well as the ones due to blocking artifacts are detected. Then we introduced the difference of slope (DOS) operator and showed that this operator only detects the edges due to the blocking artifacts.

## **Chapter 4**

# **Blockiness Removal Techniques**

### **4.1 Introduction**

The “Blocking artifact” is one of the most noticeable degradation of the block transform coding. During the past few years, several types of techniques have been developed to reduce the blocking effects. They can be applied during the encoding process and they can also be applied during or after the decoding process. In this chapter, we will review and compare several deblocking techniques.

### **4.2 Blockiness reduction schemes used in the encoding process**

#### **4.2.1 Block overlapping method**

In conventional transform image coding, the image is always segmented into non-overlapping, rectangular shape blocks for easy processing and for yielding better compression ratios. The discontinuities at block boundaries appear as a regular pattern and are thus very visually annoying. Instead of dividing an image into non-overlapped blocks, an overlap encoding method is suggested by Reeve and Lim [7]. This technique employed overlapped blocks where there is a small overlap around the perimeter of each block. The overlapped pixels at the boundaries would then be encoded twice. The boundary discontinuity is reduced by averaging the values of overlapped pixels in the decoding process. Block overlapping has been shown to be effective in reducing the blocking effect. However, this approach increases the amount of computations and it increases the bit rate as well since redundant information (overlapped pixels) has to be encoded. There is a 13%

increase in the bit rate (based on a 256x256 pixel image with 16x16 pixel subimages) according to [7].

### 4.2.2 Lapped orthogonal transform

A more efficient technique to reduce blocking effect is called Lapped Orthogonal Transform (LOT) [17]. In a traditional transformation, the number of transform coefficients is the same as the number of the input samples. This new type of transformation is capable of mapping  $L$  samples into  $N$  transform coefficients,  $L > N$ . For a traditional 1-D transformation case, assume that the incoming signal vector  $X$  has the length  $MN$ . This vector is subdivided into  $M$  segments and each segment has  $N$  samples. Let the vector  $Y$  denotes the transform coefficients of  $X$  and that  $Y$  is given by

$$Y = T^T X, \quad (4.1)$$

where  $T^T$  is the transpose of an  $MN \times MN$  block-diagonal matrix, in the form

$$T = \begin{bmatrix} D & & 0 \\ & D & \\ & & \dots \\ 0 & & & D \end{bmatrix}, \quad (4.2)$$

where  $D$  is a  $N \times N$  matrix, whose columns are the transform basis. The LOT can also be defined as (4.1) except that  $T$  is given by

$$T = \begin{bmatrix} P_1 & & & 0 \\ & P_0 & & \\ & & \dots & \\ & & & P_0 \\ 0 & & & & P_2 \end{bmatrix}, \quad (4.3)$$

where  $P_0$  is an  $L \times N$  ( $N < L \leq 2N$ ) matrix and  $P_1$  &  $P_2$  are  $L' \times N$  ( $L' < L$ ) matrices. These three matrices contain the LOT basis. Therefore the neighboring segments overlap by  $L - N$  samples with the number of transform coefficients equal to the number of input samples. The matrices  $P_1$  and  $P_2$  are introduced because the first and last segments have only one neighboring segment, and thus the LOT for the first and last segments must be defined in a slightly different way to keep the transformation within the segment boundaries [17]. Note that the LOT of a single block is not invertible, since  $P_0$  is not a square matrix. However, as long as  $T$  has the orthogonality property, the whole input vector  $X$  would be reconstructed. The LOT has the advantage of reducing blocking effects without increasing the bit rate. The disadvantage is that the fast LOT algorithm requires about 20-30 percent more computations than the fast DCT transform.

### 4.3 Blockiness reduction schemes used after the decoding process

The above section described two methods which can be utilized at the encoder so as to reduce or prevent the occurrence of blocking effects. Blocking effects can also be reduced after the decoding process i.e. as a post-processing step. This approach uses filtering techniques to reduce the high frequency components near the block boundaries.

#### 4.3.1 Symmetric, two-dimensional filtering

In [7], Reeve and Lim proposed a symmetric, two-dimensional Gaussian spatial filtering method. This filtering method is better than the overlap method in the overall performance [7]. The characteristics of this method is given below.

#### 4.3.1.1 Filter definition

The exact filter specification is given by:

$$h(n_1, n_2) = 0.2042 \exp[-0.5(n_1^2 + n_2^2)], \quad (4.4)$$

where  $n_1 = 1, 0, -1$  and  $n_2 = 1, 0, -1$ . The coefficients used in the filter is listed in Table 4.1.

Table 4.1 Symmetric filter coefficients.

0.075	0.124	0.075
0.124	0.204	0.124
0.075	0.124	0.075

#### 4.3.1.2 Computation cost

To avoid unwanted blurring, this filter is only applied to pixels at the block boundaries. For such a boundary pixel, this implementation requires at least 3 multiplications and 9 additions. Thus for the 4 boundaries of an  $8 \times 8$  pixels block, the total number of computation for filtering each block is  $9 \times 8 \times 4 = 288$  additions and  $3 \times 8 \times 4 = 96$  multiplications. This makes a total of 384 operations per block. Thus this method is not computationally demanding.

#### 4.3.1.3 Experimental results

To evaluate the performance of the post-processing methods, we evaluate them both objectively and subjectively. The objective measure is the peak signal-to-noise ratio (PSNR). Assume that the size of the original image  $f$  is  $N \times M$  and the reconstructed image is denoted by  $\hat{f}$ , the PSNR in decibels (dB) is given by:

$$PSNR = 20 \log_{10} \frac{255}{RMSE}, \quad (4.5)$$

where  $RMSE$  is defined as

$$RMSE = \sqrt{\frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M [f(i, j) - \hat{f}(i, j)]^2} \quad (4.6)$$

The JPEG-coded image at a bit rate of 0.298 bpp with PSNR 32.51 dB is shown in Figure 4.1 a. The filtered image is shown in Figure 4.1 b. As we can see, the filtered image shows much less blocking effects. The PSNR of the filtered image is 32.52 dB. The PSNR comparison of several images before and after post-processing at different compression rates obtained by our experiments are shown in Table 4.2 to Table 4.4. Although there is PSNR improvement at very low bit rates (i.e. 1/4 bpp and below), the visual blocking effects reduction is not good enough. The JPEG-coded image at 0.248 bpp and resultant filtered image are shown in Figure 4.2 a and Figure 4.2 b respectively. Strong blocking artifacts remain in the processed image.

Due to its low-pass nature, this spatial filtering approach also has the drawback of loss of high-frequency detail. Since the frequency characteristics of vertical boundaries are very different from those of horizontal boundaries, they require different amounts of high-frequency suppression. However, the symmetric filter provides the same amount high-frequency suppression in both directions. To try to minimize this blurring, various schemes have been suggested.

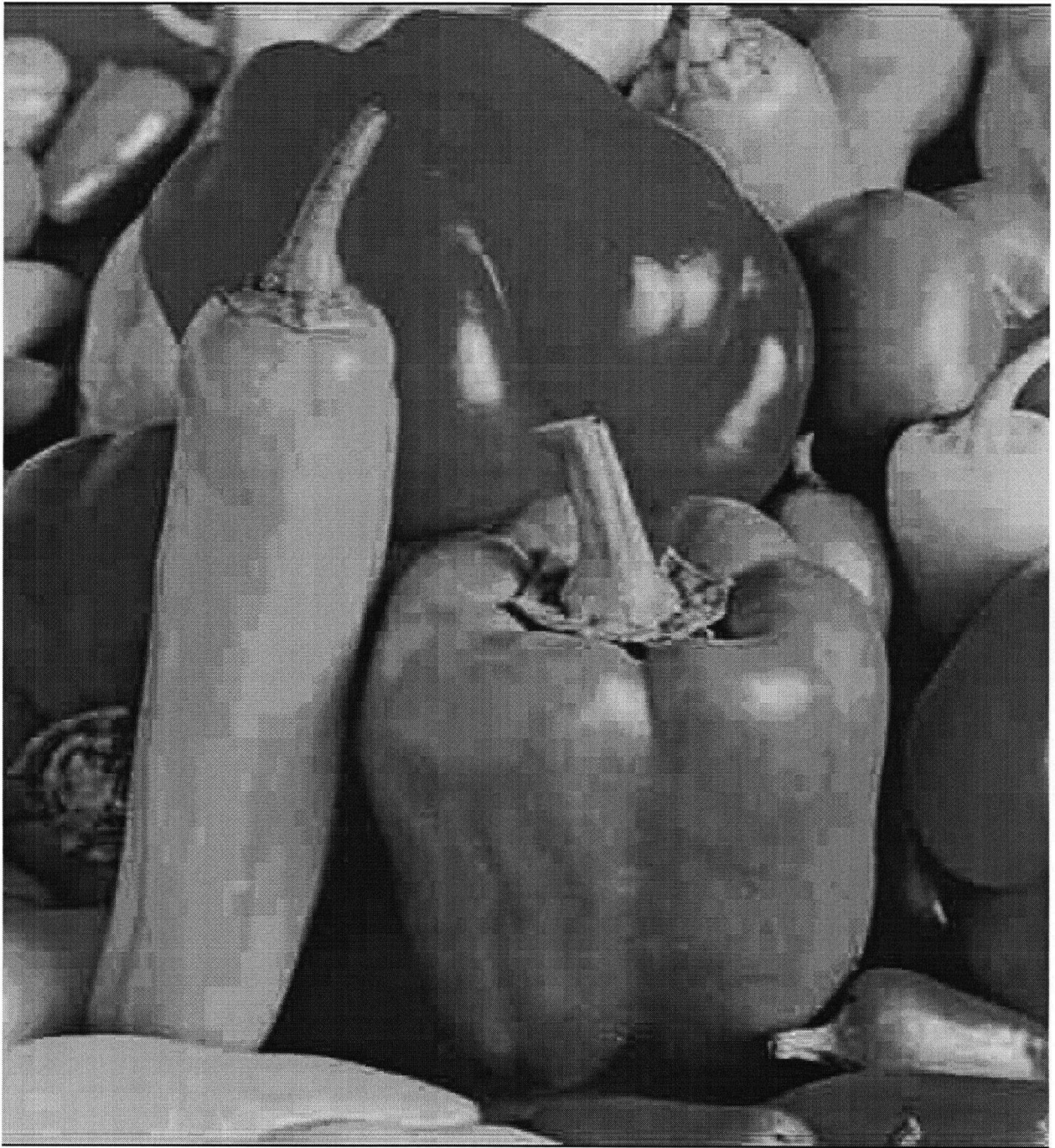


Figure 4.1 a Decompressed image, bitrate=0.298 bpp, PSNR=32.51 dB.



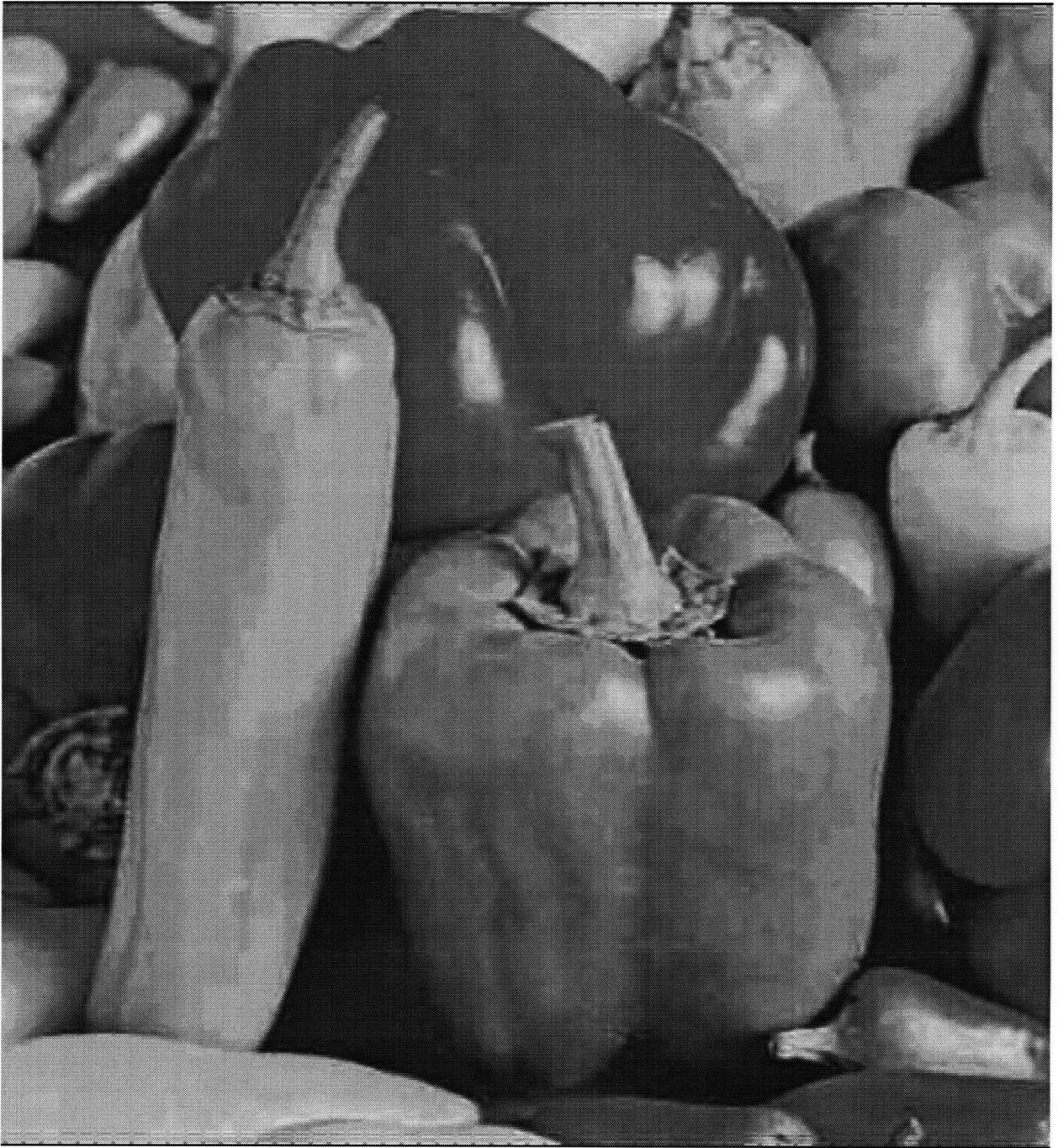


Figure 4.1 b Post-processed image by symmetric filter, bitrate=0.298 bpp, PSNR=32.52 dB.

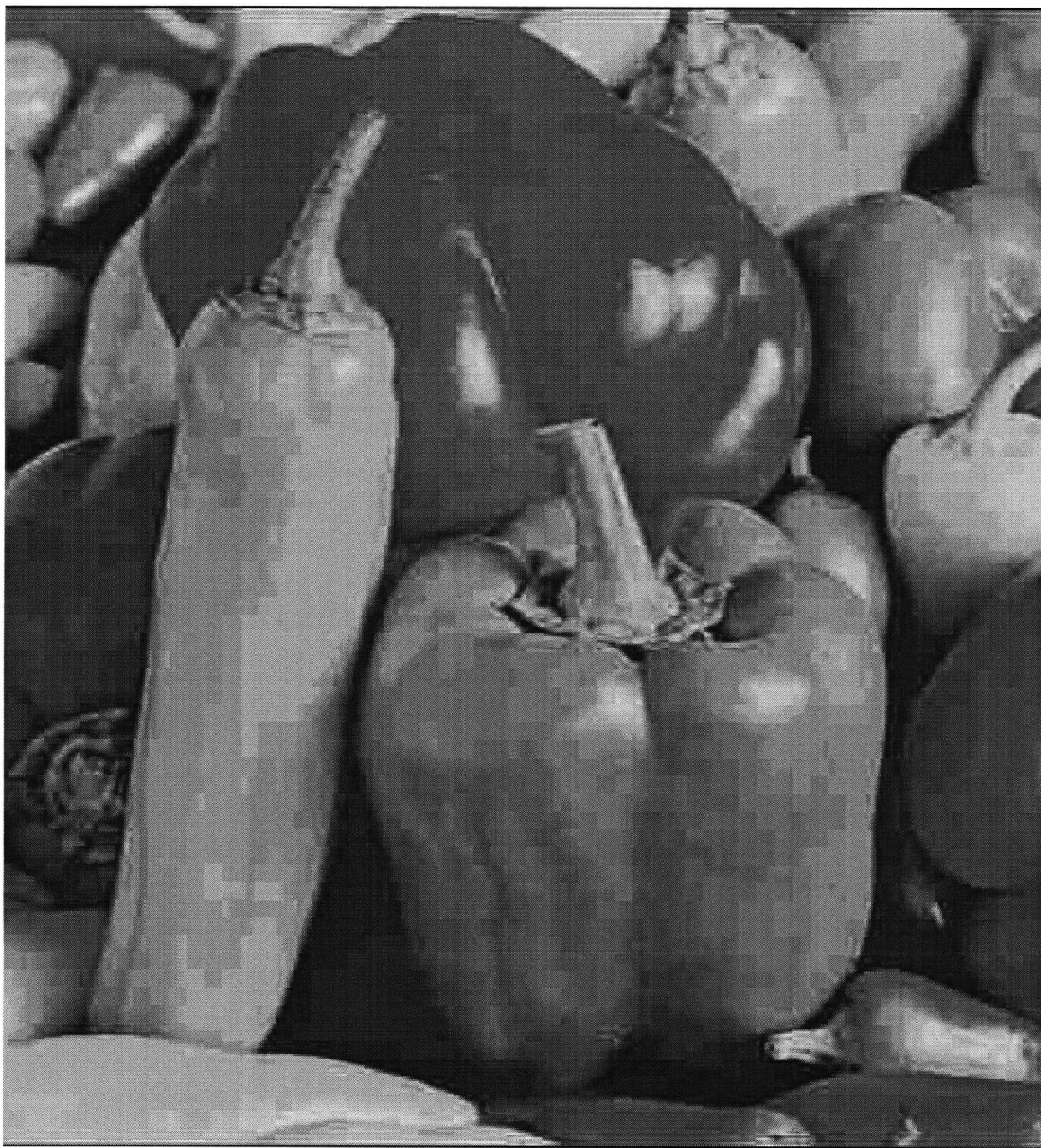


Figure 4.2 a Decompressed image, bitrate=0.248 bpp, PSNR=31.2 dB.



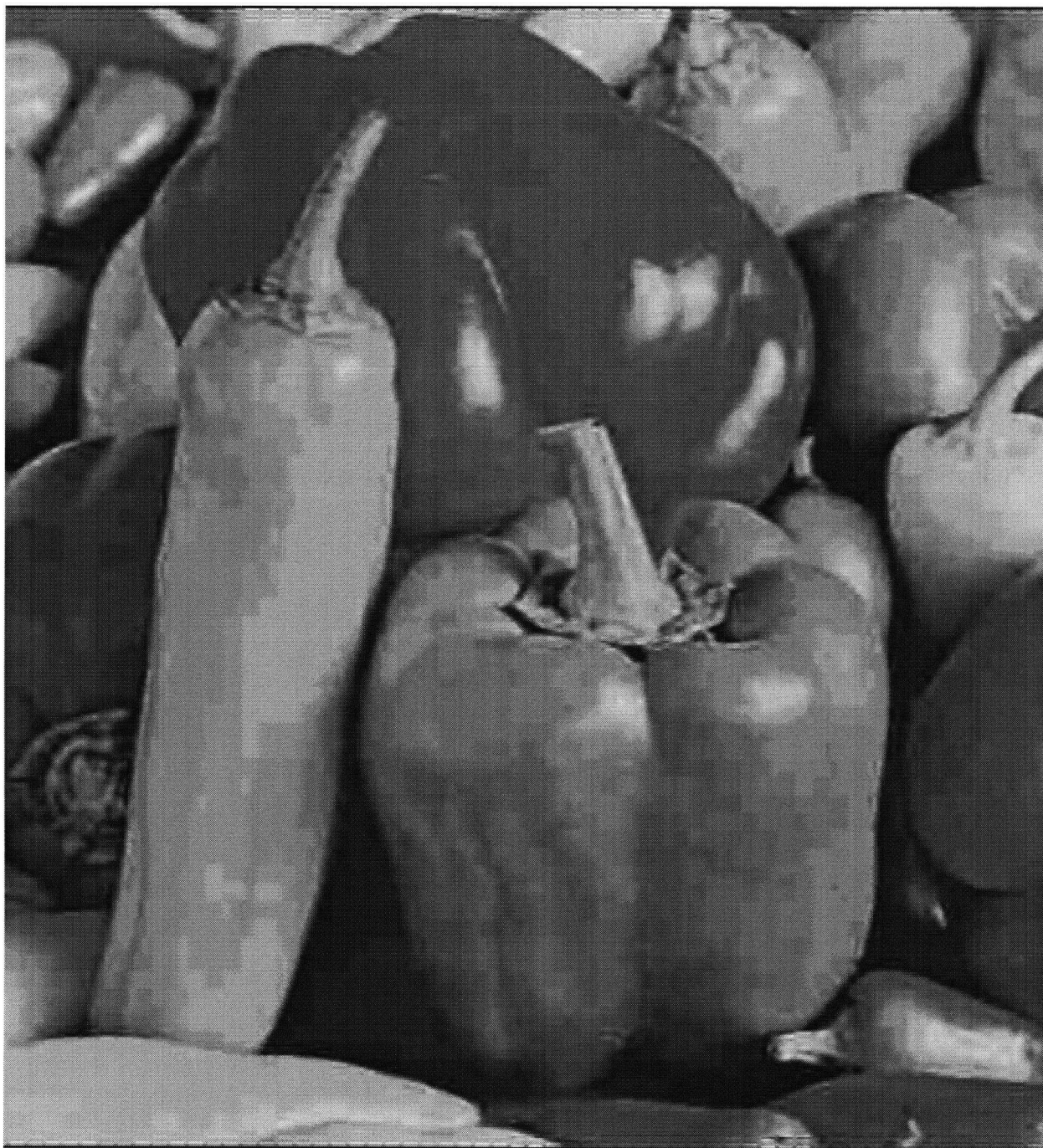


Figure 4.2 b Post-processed image by symmetric filter, bitrate=0.248 bpp, PSNR=31.52 dB.

Table 4.2 PSNR comparison of Indian image before and after post-processing.

Compression Bit-Rate (bpp)	.200	.248	.273	.318	.341	.382	.423
PSNR Before Post-Processing (dB)	26.82	27.78	28.17	28.86	29.15	29.65	30.07
PSNR After Post-Processing (dB)	27.50	28.38	28.71	29.32	29.54	29.93	30.24

Table 4.3 PSNR comparison of Lena image before and after post-processing.

Compression Bit-Rate (bpp)	.198	.251	.273	.320	.339	.379	.400
PSNR Before Post-Processing (dB)	26.52	27.70	28.24	29.02	29.35	29.91	30.15
PSNR After Post-Processing (dB)	27.33	28.50	28.97	29.60	29.85	30.35	30.54

Table 4.4 PSNR comparison of Peppers image before and after post-processing.

Compression Bit-Rate (bpp)	.150	.185	.220	.248	.274	.285	.298
PSNR Before Post-Processing (dB)	27.97	29.27	30.39	31.20	31.20	32.21	32.51
PSNR After Post-Processing (dB)	28.80	29.92	30.86	31.52	31.52	32.31	32.52

### 4.3.2 Anisotropic filtering

In [13], a circular-asymmetric two-dimensional filter is proposed. In an image with blocking effects, the discontinuities between blocks are perpendicular to the block boundaries and the pixels in the direction parallel to block boundaries still preserve smoothness. Therefore, the filter coefficients have to be adjusted to the directions of the boundaries so that the filter has different responses in the vertical and horizontal directions. The desired large high-frequency suppression is applied in the direction perpendicular to the block boundaries and the filter has almost no filtering effects in the other direction since little filtering is required in the direction parallel to block boundaries.

#### 4.3.2.1 Coefficients design

Let  $X(i, j)$  be the pixel value of a transform-coded image and  $Y(i, j)$  be the pixel value after filtering and is defined as:

$$Y(i, j) = \sum_{k=-I}^I \sum_{l=-J}^J C(k, l) X(k-i, l-j), \quad (4.7)$$

where  $C(k, l)$  is a Gaussian type filter defined as:

$$C(k, l) = \frac{1}{2\pi\sigma_1\sigma_2} e^{-\left(\frac{k^2}{2\sigma_1^2} + \frac{l^2}{2\sigma_2^2}\right)}, \quad (4.8)$$

where  $-I \leq k \leq I$  and  $-J \leq l \leq J$ . Obviously, we can control the degree of high-frequency suppression by varying the parameter  $\sigma_1$  and  $\sigma_2$ . A larger  $\sigma$  provides more suppression and a smaller  $\sigma$  provides less suppression. Therefore,  $\sigma$  can be varied to obtain a desired degree of high-frequency suppression. Let  $\sigma_1^2 = 0.72$  and  $\sigma_2^2 = 0.13$ . Thus the filter has larger high-frequency suppression in the horizontal direction and smaller suppression in the vertical direction. We can get an opposite filter response by switching the value of  $\sigma_1$  and  $\sigma_2$ . The normalized coefficients used in the filters which are used to remove horizontal blockiness and vertical blockiness are shown in Table 4.5 and Table 4.6, respectively.

Table 4.5 Anisotropic filter coefficients used to filter vertical boundary.

0.005	0.01	0.005
0.24	0.48	0.24
0.005	0.01	0.005

Table 4.6 Anisotropic filter coefficients used to filter horizontal boundary.

0.005	0.24	0.005
0.01	0.48	0.01
0.005	0.24	0.005

Anisotropic low pass filters are only applied to pixels adjacent to the subimage boundaries. Therefore, they can effectively reduce the blocking effects while preserving the sharpness of

the image. However, when the blockiness is heavy, it is usually not enough to filter only at the block edges. The smoothing of the boundary pixels can lead to a new discontinuity between each modified boundary pixel and the pixel immediately adjacent to it within the block. This simple post filtering process is not good enough to remove the blocking effects in the severe cases, i.e. for bit rates below 1/4 bpp.

#### 4.3.2.2 Computation cost

For each  $Y(i,j)$ , this implementation requires at least 4 multiplications and 9 additions. The total computation cost for filtering each block is:

$$9 \times 8 \times 4 = 288 \text{ additions and}$$

$$4 \times 8 \times 4 = 128 \text{ multiplications.}$$

Thus the computational effort for this method is similar to that of the symmetric filtering.

#### 4.3.2.3 Experimental results

Post-processing results obtained using the anisotropic filter for JPEG-coded image at bit rates of 0.298 bpp and 0.248 bpp are shown in Figure 4.3 b and Figure 4.4 b, respectively. The PSNR comparisons of several images before and after post-processing at different compression rates are shown in Table 4.7 to Table 4.9. Compared to the results using the symmetric filter, the texture regions and the edges are preserved better by using the anisotropic filters.

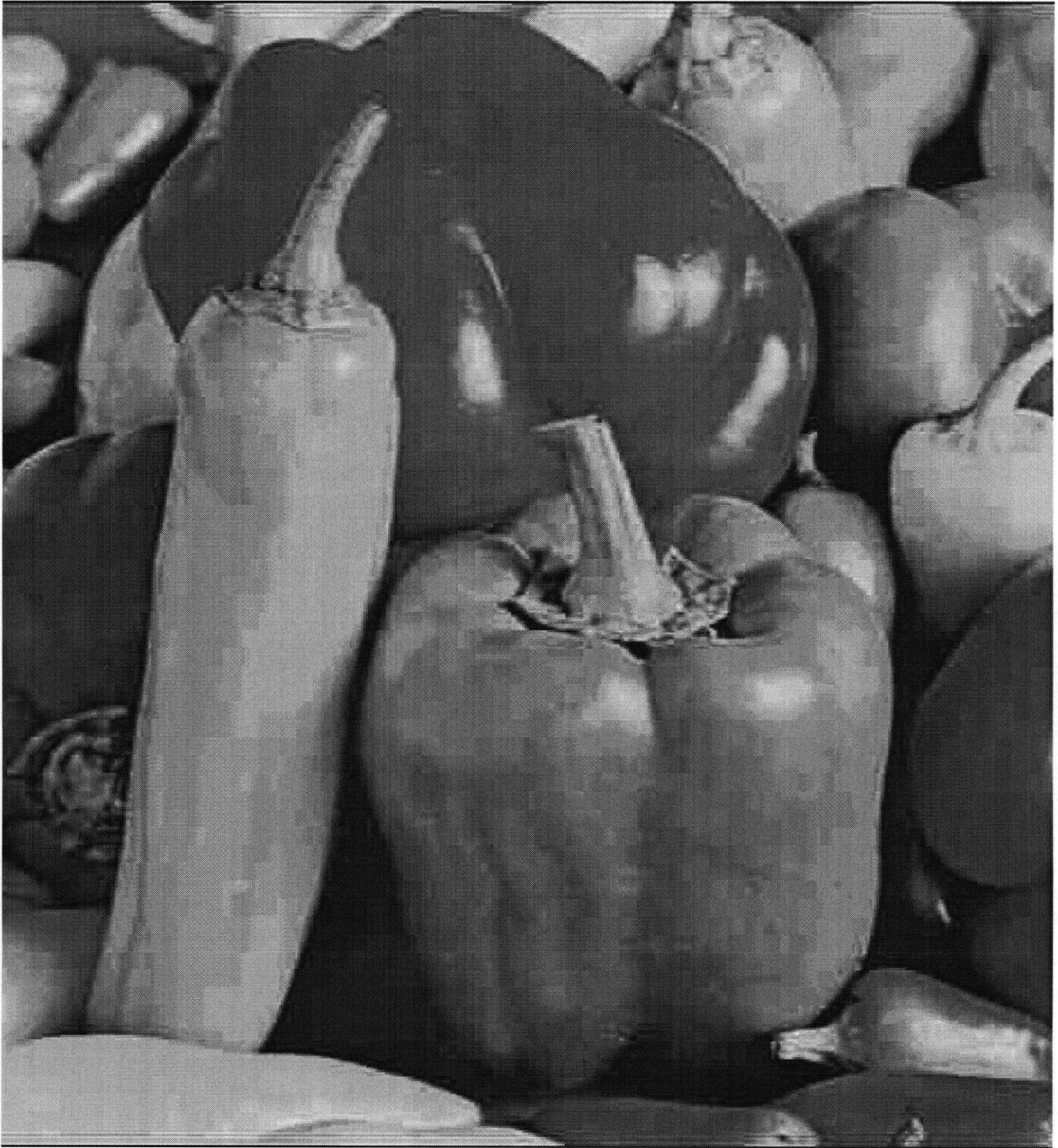


Figure 4.3 a Decompressed image, bitrate=0.298 bpp, PSNR=32.51 dB.



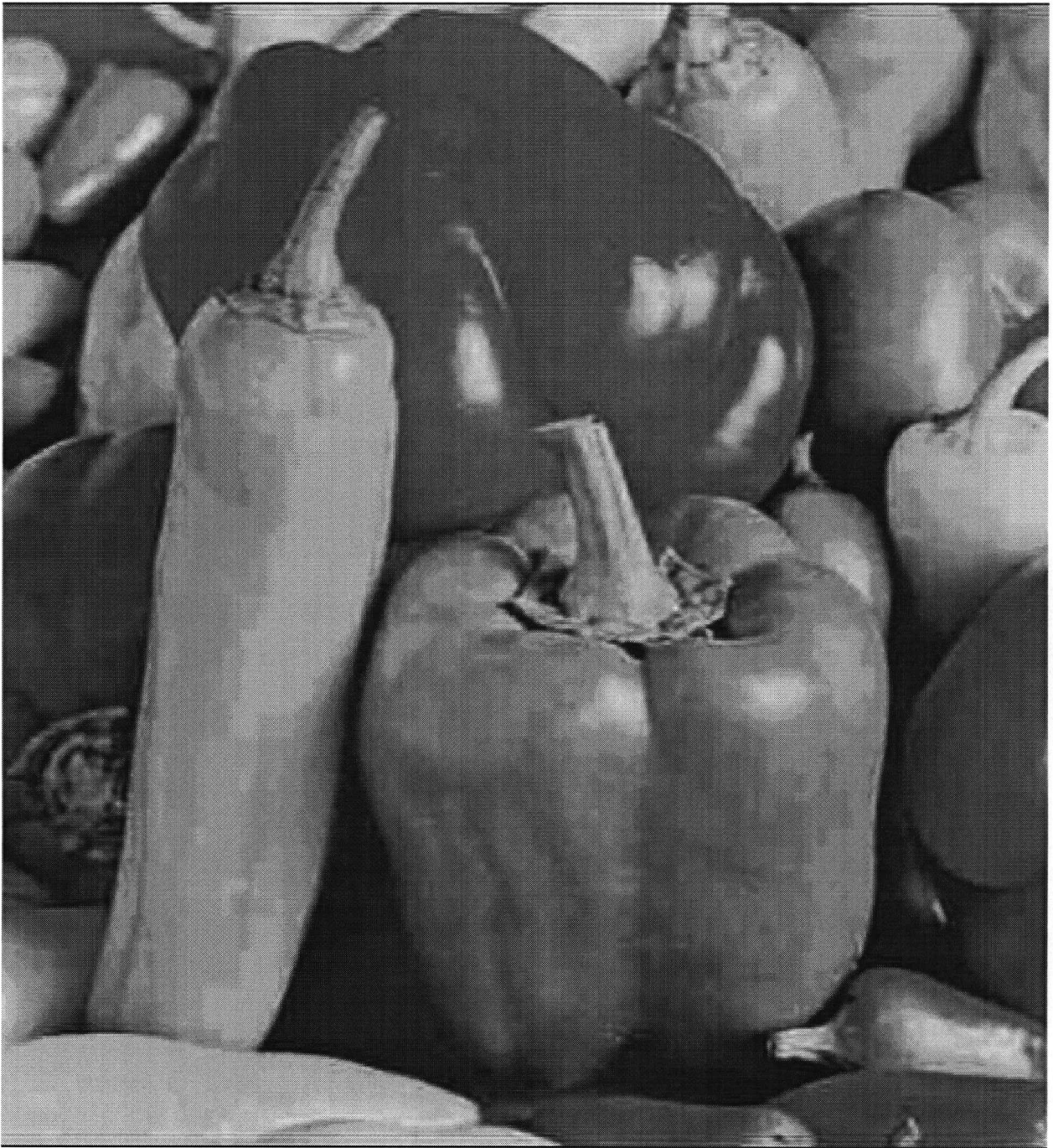


Figure 4.3 b Post-processed image by the anisotropic filters, bitrate=0.298 bpp, PSNR=33.04 dB.



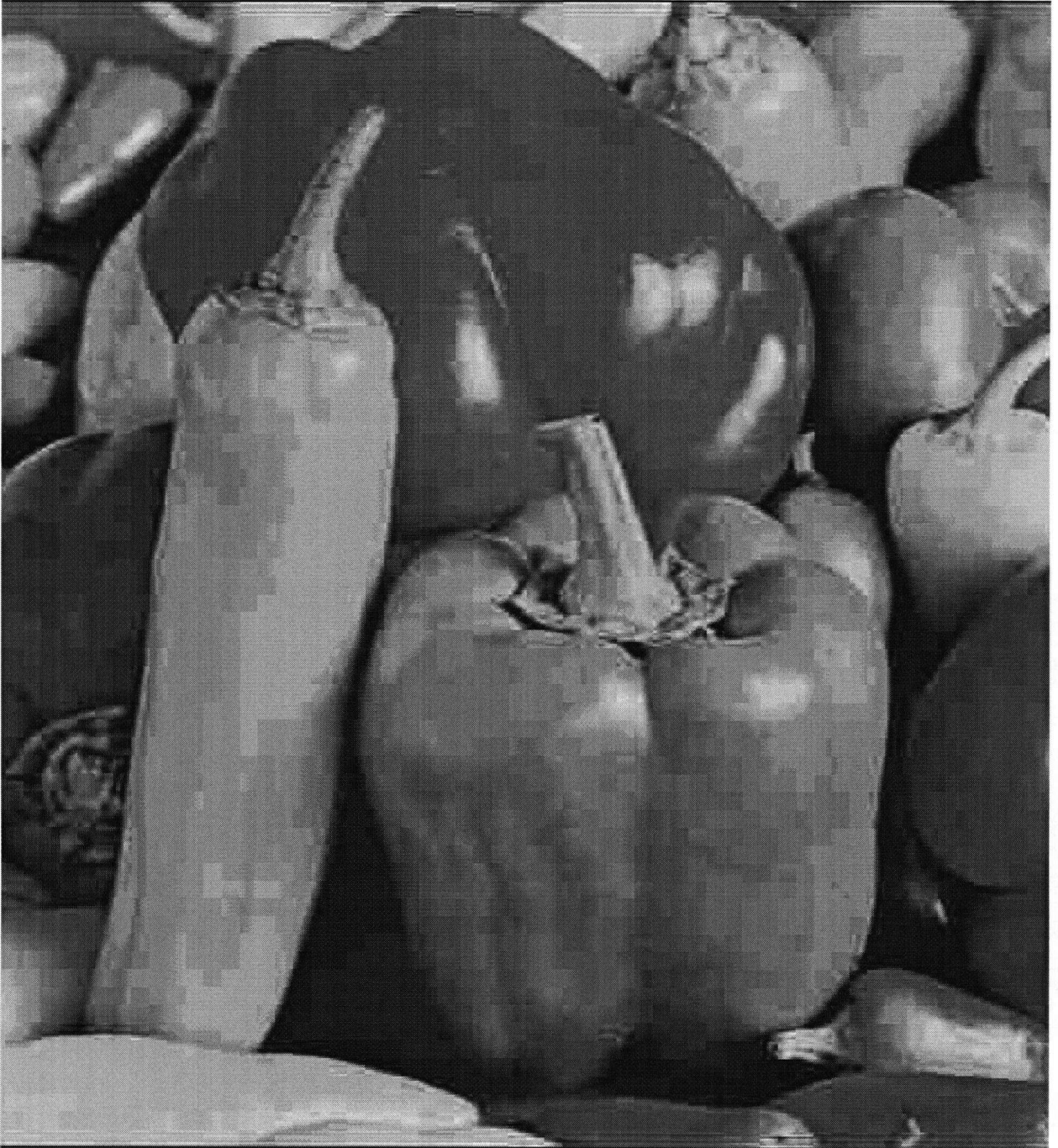


Figure 4.4 a Decompressed image, bitrate=0.248 bpp, PSNR=31.2 dB.

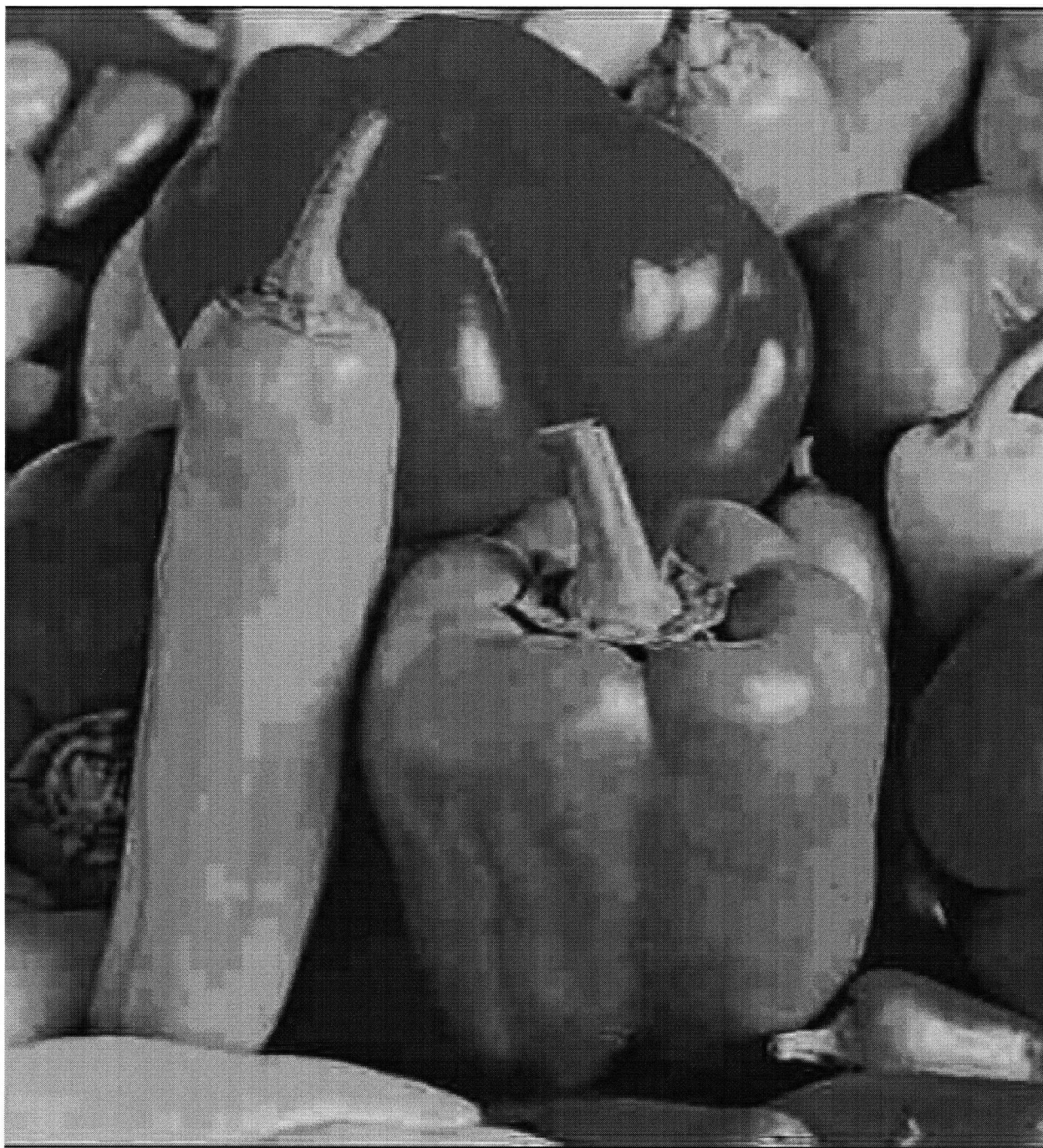


Figure 4.4 b Post-processed image by the anisotropic filters, bitrate=0.248 bpp, PSNR=31.83 dB.

Table 4.7 PSNR comparison of Lena image before and after post-processing.

Compression Bit-Rate (bpp)	.198	.251	.273	.320	.339	.379	.400
PSNR Before Post-Processing (dB)	26.52	27.70	28.24	29.02	29.35	29.91	30.15
PSNR After Post-Processing (dB) by symmetric filter	27.33	28.50	28.97	29.60	29.85	30.35	30.54
PSNR After Post-Processing (dB) by anisotropic filter	27.26	28.47	28.97	29.66	29.94	30.50	30.68

Table 4.8 PSNR comparison of Indian image before and after post-processing.

Compression Bit-Rate (bpp)	.200	.248	.273	.318	.341	.382	.423
PSNR Before Post-Processing (dB)	26.82	27.78	28.17	28.86	29.15	29.65	30.07
PSNR After Post-Processing (dB) by symmetric filter	27.50	28.38	28.71	29.32	29.54	29.93	30.24
PSNR After Post-Processing (dB) by anisotropic filter	27.47	28.39	28.76	29.41	29.66	30.10	30.46

Table 4.9 PSNR comparison of Peppers image before and after post-processing.

Compression Bit-Rate (bpp)	.150	.185	.220	.248	.274	.285	.298
PSNR Before Post-Processing (dB)	27.97	29.27	30.39	31.20	31.20	32.21	32.51
PSNR After Post-Processing (dB) by symmetric filter	28.80	29.92	30.86	31.52	31.52	32.31	32.52
PSNR After Post-Processing (dB) by anisotropic filter	28.75	28.98	31.05	31.83	32.47	32.77	33.04

### 4.3.3 Adaptive filtering using edge classification

The human visual system is not sensitive to blockiness that occur in texture and edge regions. Low-pass filtering at texture and edge regions also results in undesired blurring. Therefore a low-pass filtering that adapts to different regions is able to prevent low-pass blurring of regions with texture and edges. In [10], [14], [15], [19], different edge classification procedures are used to distinguish between monotone, textured and strong edge regions of the image. The spatial filtering schemes are then driven by the output of the edge classifiers. Edge and texture

regions are not filtered since humans are more sensitive to errors in the low frequency components of the image than those in the high frequencies.

There are different kinds of edge classification methods. The classification can be done in the spatial or DCT domains. In [9], [10] and [14], spatial methods such as gradient / threshold detector and histogram methods are used. However, it is difficult to detect edges of images with complex content which are coarsely quantized. The block boundaries may be artificially identified as edges. In [15] and [19], block classification is performed by examining the DCT coefficients. The blocks which have none zero coefficients outside the  $(2 \times 2$  or  $3 \times 3)$  top left corner region of the DCT  $8 \times 8$  transform are declared as edge or texture regions. This method is briefly described as following.

A block is marked as low frequency block if

$$C_{DCT}(i, j) \bullet K_{low} = \vec{0}, \quad (4.9)$$

Similarly, a block is marked as high frequency block if

$$C_{DCT}(i, j) \bullet K_{high} \neq \vec{0}, \quad (4.10)$$

where  $C_{DCT}(i, j)$  is the  $8 \times 8$  block of DCT coefficients of block  $(i, j)$ ,  $\bullet$  is the element-by-element multiplication,  $K_{low}$  and  $K_{high}$  are the test matrices used for detection of low frequency blocks and high frequency blocks and are defined by

$$K_{low} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad (4.11)$$

and

$$K_{high} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad (4.12)$$

respectively.  $\hat{0}$  is the  $8 \times 8$  matrix of zeros.

#### 4.3.4 Adaptive filtering using local statistical estimation

A method with impressive post-processing results is given in [18]. In this algorithm, the filtering strategy relies on the estimation of the quantization error  $T$  in the spatial domain for each  $n \times n$  block. The estimation of  $T$  is given by:

$$T = \frac{1}{n^2} \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} E_{uv}, \quad (4.13)$$

where  $E_{uv}$  is determined by the quantization step and the probability distribution of the original unquantized DCT coefficient at frequency  $(u, v)$ . The definition of  $E_{uv}$  is:

$$E_{uv} = \frac{\int_{l_{uv}}^{r_{uv}} (y - \hat{C}_{uv})^2 p_{uv}(y) dy}{\int_{l_{uv}}^{r_{uv}} p_{uv}(y) dy} \quad u, v = 1, \dots, n, \quad (4.14)$$

where  $p_{uv}$  is the probability distribution of the DCT coefficient at frequency  $(u, v)$  before quantization, and  $l_{uv}$ ,  $r_{uv}$  are the left and right quantization boundaries.  $\hat{C}_{uv}$  is the minimum mean squared error estimate of each DCT coefficient which is calculated by:

$$\begin{aligned}\hat{C}_{uv} &= E[y | l_{uv} \leq y \leq r_{uv}] \\ &= \frac{\int_{l_{uv}}^{r_{uv}} y p_{uv}(y) dy}{\int_{l_{uv}}^{r_{uv}} p_{uv}(y) dy} \quad u, v = 1, \dots, n.\end{aligned}\tag{4.15}$$

Since the original DCT coefficients can not be accessed at the decoder, the  $p_{uv}$  for each block is estimated from the moments of the quantized coefficients. However, the estimation method is not reported in [18] and thus we can not simulate the results of this method.

The steps used to smooth the block boundary pixels are summarized as follows:

1. Set a threshold  $t = 2\sqrt{T}$  for distinguishing artificial edges, where  $T$  is calculated by (4.13).
2. Calculate the pixel differences across horizontal and vertical block boundaries, where are stored in vectors  $d_{col}$  and  $d_{row}$ , respectively.
3. Set  $\alpha = (t - v)/(2t)$ , where  $v$  is proportional to the mean of the entire image.
4. For each entry  $d = x - y$  in  $d_{col}$  and  $d_{row}$  such that  $|d| \leq t$ , smooth the corresponding pixels via  $\hat{x} = x - \alpha d$  and  $\hat{y} = y + \alpha d$ .
5. For each entry  $d > t$ , smooth the corresponding pixels via  $\hat{x} = x - \alpha t$  and  $\hat{y} = y + \alpha t$ , and for each  $d < -t$ , smooth the pixels via  $\hat{x} = x + \alpha t$  and  $\hat{y} = y - \alpha t$ .
6. To alleviate the discontinuity offset problem caused by the smoothing of boundary pixels, the smoothing process is iterated until the center of the block is reached.

Note that since most of the high order DCT coefficients are quantized to zero, thus after this coarse quantization, it is difficult to get a good estimate of the probability distribution of the

DCT coefficient at each frequency from the mean and variance of quantized coefficients. If we can not get an accurate estimate of  $T$  for each block, we may risk over-smoothing or under-smoothing in the filtering process. Another drawback of this method is that the algorithm relies on the values of the DCT coefficients and quantization factors and thus has to access data from the decoder. Thus it has to be built inside the decoder and cannot be applied after the decoder.

### 4.3.5 Iterative blocking artifact removal techniques

The projection onto convex sets (POCS) based recovery algorithm [11]-[12] is another powerful spatially-adaptive approach. Its basic idea is to optimize the value of the quantized transform coefficients subject to some smoothness constraint and some quantization constraint. A low-pass filter is applied along block boundaries to remove the discontinuity of block boundaries. The DCT coefficients of the restored image are re-mapped back to their original quantization levels to prevent oversmoothing of images. Such procedure needs to be iterated several times. The major drawback of this approach is its high computational complexity. It takes at least 3~5 iterations for this algorithm to converge, where both forward and inverse DCT are required in each iteration, this is besides other required computational overhead [12]. It also needs to access data from the decoder, and thus it has to be built inside the decoder too.

### 4.3.6 Dithering

This method [10] adds a carefully-controlled, random dither signal to the transform coefficients during the decoding process. The dither signal conforms to a model of the average energy distribution of the transform coefficients prior to quantization.

At the encoder, each  $N \times N$  coefficients block undergoes a quantization process to round

each coefficient value to the nearest integer that can be described mathematically as:

$$\hat{F}_{k,l} = \text{rnd}(F_{k,l}/q_{k,l}) \quad 0 \leq k, l \leq N-1, \quad (4.16)$$

where

$\hat{F}_{k,l}$  is the quantized value of  $F_{k,l}$ , the  $(k, l)$ th transform coefficient in the transformed block,

$N$  is the sub-block size,

$k, l$  are the matrix row and column indices respectively, and

$q_{k,l}$  is the corresponding quantization coefficient.

At the decoder, a reverse procedure is applied and the reconstructed transform coefficients  $F'_{k,l}$  are generated by:

$$F'_{k,l} = q_{k,l} \cdot \hat{F}_{k,l}. \quad (4.17)$$

In [20], McLaren proposes the following model for the average energy distribution  $P_{i,j}$  of the DCT coefficients:

$$P_{k,l} = 34.10\omega_{k,l}^{-0.94} - 1 \quad 0 \leq k, l \leq N-1, \quad (4.18)$$

where  $\omega_{k,l} = \sqrt{k^2 + l^2}$  is the spatial frequency corresponding to transform coefficient  $F'_{k,l}$ .

The dither signal is then generated by weighting each  $P_{i,j}$  by a Gaussian signal  $G(0, \sigma)$  with zero mean and  $\sigma$  variance. In order to avoid introducing an excessively large random component, the dither signal has to be restricted within a maximum permissible amplitude. This is achieved by thresholding  $G(0, \sigma)$ . Thus, the final form of the proposed dithering scheme is given by:



$$\bar{F}_{k,l} = \begin{cases} q_{k,l} \cdot \hat{F}_{k,l} + P_{k,l} \cdot G(0, \sigma) & \text{if } |G(0, \sigma)| < G_{max} \\ q_{k,l} \cdot \hat{F}_{k,l} & \text{otherwise} \end{cases}, \quad (4.19)$$

where the truncation parameter  $G_{max}$  is bit-rate dependent.

According to our experiments, the spatial dithering is proved to be a good method of masking blocking artifacts and providing the illusion of continuous-tone pictures on displays.

## Chapter 5

### A New Blockiness Reduction Scheme

#### 5.1 Introduction

In this chapter, we will propose a new approach for reducing the blocking effects which is also effective at very low bit rates. Compared to [11] and [12], this method requires a moderate amount of computational effort. Unlike [18], the thresholds it requires are constant throughout the whole process and are bit-rate independent. This method also can be applied after the image or video has been decoded.

This method consists of two stages. The first stage is applied in the DCT domain. In this stage, the correlation between the intensity values of the boundary pixels of two neighboring blocks is explored. The continuity of the pixels across the boundary is recovered from the continuity of the pixels within the two neighboring blocks. After applying this stage, most of the blockiness are removed. The very severe blocking artifacts may not be completely removed by the above stage, however they are significantly reduced. When the blocking effects are very severe and computation time is not critical, stage 2 which consists of spatial filtering is applied to remove these remaining artifacts.

In stage 2, the circular-asymmetric two-dimensional spatial filter [13] discussed in the last chapter, is applied to totally remove the remaining discontinuities in the image. This filter is only applied to the boundaries which were detected to have strong blockiness. The detection method we used is discussed in Chapter 3. The detection of these regions is carried out prior to the appli-

cation of stage 1, this is because strong artifacts appear in the original decoded image as abrupt step changes in intensity across the block boundaries and are more easily detected before applying stage 1. An epsilon-filter [21] may also be applied when the compression noise is heavy. The epsilon-filter helps reduce other coding artifacts such as ringing artifacts. The combination of filtering techniques in the DCT and spatial domain is found to be a powerful blockiness reduction technique. Our algorithm can be summarized as follows:

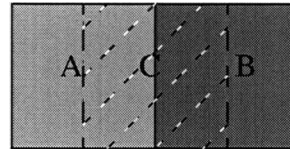
1. Identify all blocks with strong blockiness effects.
2. Reduce the block discontinuity in the DCT domain. This filtering is applied to all the images, i.e. to all blocks.
3. Apply the spatial filtering.

## 5.2 The frequency representation of the spatial block discontinuity in the DCT domain

In this section, we will discuss the frequency representation of the spatial block discontinuity in the DCT domain. If we can locate the positions of the frequencies where noise energy mostly fall, we might be able to reduce the noise without much affecting the genuine image signal.

Let's examine two  $8 \times 8$  pixels horizontally adjacent blocks A and B, and assume all pixels in block A have a constant intensity value  $a$  and block B has a constant intensity value  $b$  ( $a \neq b$ ). Let  $8 \times 4$  pixels of the right half of block A and the  $8 \times 4$  pixels of the left half of block B form a block denoted as block C. Block C is the  $8 \times 8$  block which contains the boundary pixels. Obviously, there is a horizontal step in block C, see Figure 5.1. The intensity values of Block C approximate the discontinuity that results from a blocking artifact.

Let  $F_a(u, v)$ ,  $F_b(u, v)$  and  $F_c(u, v)$  be the DCT coefficients of blocks A, B and C. Since the pixels of block A and B have constant intensity values, each of  $F_a(u, v)$  and  $F_b(u, v)$  has only one non zero value which is at (0,0).



Horizontal Blockiness

Figure 5.1 Blockiness Diagram.

Block C has discontinuity in the middle of the block. Its DCT coefficients  $F_c(u, v)$  are:

$$F_c(u, v) = C_u C_v \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} c(k, l) W_{DCT}(u, k) W_{DCT}(v, l), \quad (5.1)$$

where  $c(k, l)$  is the intensity value of the  $(k, l)$  pixel in C,  $C_u$ ,  $c_v$  and  $W_{DCT}$  are given by

$$C_u, C_v = \begin{cases} \frac{1}{\sqrt{N}} & \text{for } u=0, v=0, \\ \sqrt{\frac{2}{N}} & \text{otherwise,} \end{cases}, \quad (5.2)$$

and

$$W_{DCT}(u, k) = \cos[(2k+1)\pi u/2N]. \quad (5.3)$$

Note that since  $c(k, l)$  is a horizontal step image, (5.1) can be rewritten as

$$F_c(u, v) = C_u C_v \sum_{l=0}^{N-1} c(k, l) W_{DCT}(v, l) \sum_{k=0}^{N-1} W_{DCT}(u, k). \quad (5.4)$$

If  $u \neq 0$ , then  $\sum_{k=0}^{N-1} W_{DCT}(u, k) = 0$ . That is  $F_c(u, v) = 0$  for  $u \neq 0$ .

If  $u = 0$ , then  $\sum_{k=0}^{N-1} W_{DCT}(u, k) = N$ . Thus  $F_c(u, v) = C_u C_v N \sum_{l=0}^{N-1} c(k, l) W_{DCT}(v, l)$  for  $u = 0$ .

$$\begin{aligned}
 \text{If } N=8, \text{ we have } F_c(0, v) &= C_u C_v N \sum_{l=0}^7 c(k, l) W_{DCT}(v, l) \\
 &= C_u C_v N \left( \left( \sum_{l=0}^3 a \cdot W_{DCT}(v, l) \right) + \left( \sum_{l=4}^7 b \cdot W_{DCT}(v, l) \right) \right) \\
 &= a C_u C_v N \sum_{l=0}^7 W_{DCT}(v, l) + (b - a) (C_u C_v N) \sum_{l=4}^7 W_{DCT}(v, l) \\
 &= (b - a) (C_u C_v N) \sum_{l=4}^7 W_{DCT}(v, l) \\
 &= (b - a) (C_u C_v N) \left( \cos \frac{9}{8} v \pi + \cos \frac{11}{8} v \pi + \cos \frac{13}{8} v \pi + \cos \frac{15}{8} v \pi \right)
 \end{aligned}$$

It can be calculated that  $F_c(0, v) = 0$  for  $v=2, 4, 6$ .

Therefore,  $F_c(u, v)$  has none zero values only for  $v = 0, 1, 3, 5, 7$  and  $u = 0$ . This implies that the noise energy of the horizontal boundary discontinuity of block C only affects the first row of the DCT coefficient array. It also can be easily deduced that the heavier the blockiness, the larger the ripples in these non zero values.

### 5.3 Recover the continuity of block C in the DCT domain

The above result suggests that to reduce blockiness,  $F_c(u, v)$  for  $v = 0, 1, 3, 5, 7$  and  $u = 0$  should be modified. However, before such modification is performed, one has to ensure that the edge appearance (blockiness) between blocks A and B is not due to a genuine abrupt horizontal change in the gray levels of the pictures at that position. We also note that the presence of genuine strong edges or textures tend to mask the blockiness appearance of the picture in these areas. Thus for such areas, blockiness appearance reduction will not be performed so as to minimize as much as possible any unnecessary changes in the picture and thus the modified picture appears more natural and sharp.

Thus the conditions that should be met before modifying the blockiness appearance are:

1. Block A has a similar horizontal frequency property as that of block B.
2. There is no genuine strong edge or texture between block A and block B. The presence of a genuine strong edge or texture between block A and block B will mask the blocking effect in block C.

To meet condition 1. above, the first column of the DCT matrix of block A and that of block B should be close in values. As discussed in [15] and [19], the coefficients in the  $2 \times 2$  or  $3 \times 3$  top left corner of the DCT coefficient matrix are good representatives of the block frequency property. This is because most of the image energy is compacted in these low frequencies. Thus to save on the number of computations and to meet the first condition above, we only impose the following two constraints:

$$\text{a. } |F_a(0, 0) - F_b(0, 0)| < T_1; \quad (5.5)$$

$$\text{b. } |F_a(0, 1) - F_b(0, 1)| < T_2, \quad (5.6)$$

To address condition 2. above we note that if there is a genuine strong edge between block A and block B, this edge appears in block C. Thus for condition 2. to be satisfied, we also have to make sure that block C has only low frequency contents. A straightforward method is to examine the high order coefficients of block C. However, by our experiments it was found that to save the number of computations it is enough to meet the following constraint:

$$c. |F_c(3, 3)| < T_3, \quad (5.7)$$

Please note that  $T_1$ ,  $T_2$  and  $T_3$  are all predetermined i.e. fixed thresholds.

If the three above constraints are satisfied, we perform the blockiness reduction in the DCT domain by modifying the five relevant coefficients of  $F_c(u, v)$ . The first row of the DCT coefficient matrix of block C is modified by the weighted average of blocks A, B and C. Let  $F_{Mc}(u, v)$  be the modified DCT coefficient of block C. The modification is formulated as follows:

$$F_{Mc}(0, v) = \alpha_0 F_c(0, v) + \beta_0 [F_a(0, v) + F_b(0, v)] \quad v = 0, 1 \quad (5.8)$$

$$F_{Mc}(0, v) = \alpha_1 F_c(0, v) + \beta_1 [F_a(0, v) + F_b(0, v)], \quad v = 3, 5, 7, \quad (5.9)$$

$$F_{Mc}(u, v) = F_c(u, v), \text{ for all } v \neq 0, 1, 3, 5, 7, \quad (5.10)$$

where  $\alpha_0 + 2\beta_0 = 1$  and  $\alpha_1 + 2\beta_1 = 1$ .

Then the pixels of block  $C(k, l)$ , are replaced by  $C_{Mc}(k, l)$ , the inverse DCT of  $F_{Mc}(u, v)$ . It can be easily shown that the modified gray levels of the pixels of block C are:

$$C_{Mc}(k, l) = c(k, l) + \zeta(l), \quad (5.11)$$

$$\text{where } \zeta(l) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C_u C_v (F_{Mc}(u, v) - F_c(u, v)) W_{DCT}(u, k) W_{DCT}(v, l)$$

$$= C_0 C_0 [F_{Mc}(0, 0) - F_c(0, 0)] + \sum_{i=0}^3 C_0 C_v [F_{Mc}(0, v) - F_c(0, v)] W_{DCT}(v, l), \quad (5.12)$$

where  $v = 2i + 1$ .

The processing procedure described above is used to reduce the discontinuity at the boundary of two horizontally adjacent blocks. A similar procedure can be used to remove the boundary discontinuity between two vertically adjacent blocks. In the later case, the first column of the DCT coefficient array of block C is modified by the weighted average of blocks A, B, and C.

## 5.4 Examples

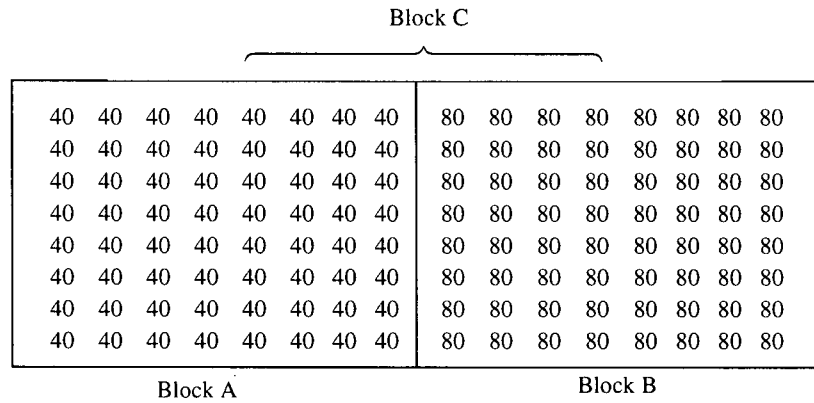
To better understand this method, we discuss two examples.

### 5.4.1 Examples 1

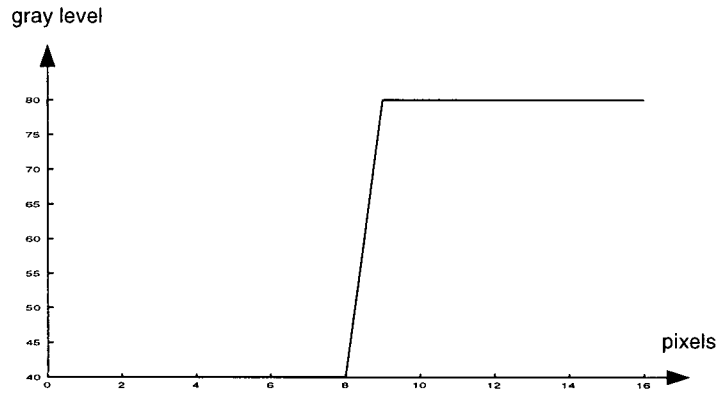
Assume that block A and block B have pixel values as listed in Figure 5.2. We can see that there is a large step intensity change across the block boundary. The DCT coefficient matrices of block A, block B and block C are listed in Figure 5.3.

Now we use equations (5. 8), (5. 9) and (5. 10) to do the modifications. Let  $\alpha_0 = 0.6$ ,  $\beta_0 = 0.2$ ,  $\alpha_1 = 0.5$ , and  $\beta_1 = 0.25$ . The reason why  $\alpha_0 > \alpha_1$  and  $\beta_0 < \beta_1$  is that lower order coefficients are more accurate than higher order coefficients because of the nature of the quantization matrix and thus they need less filtering. The modified coefficients of block C are shown in Figure 5.4. The modified pixel values of block C calculated by (5. 11) and (5. 12) are listed in Figure 5.5a. As we can see from Figure 5.5b that the large intensity step across the block boundary is now broken into several small steps.





(a) Original block data



(b) The intensity plot of a row of block data.

Figure 5.2 Original block data of example 1.

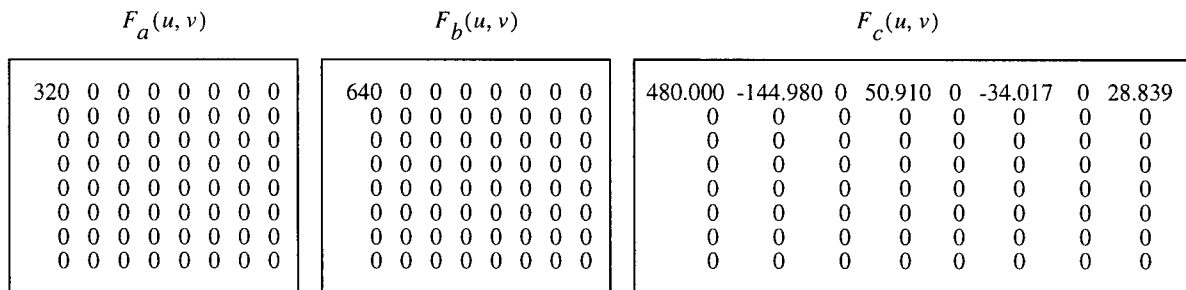


Figure 5.3 DCT coefficients for blocks A, B and C.

480.000	- 86990	0	25.460	0	-17.010	0	14.42
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 5.4 Modified coefficients of block C.

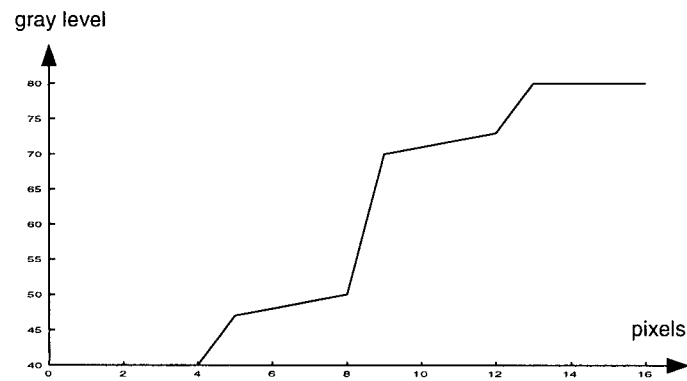
Block C

40	40	40	40	47	48	49	50	70	71	72	73	80	80	80	80
40	40	40	40	47	48	49	50	70	71	72	73	80	80	80	80
40	40	40	40	47	48	49	50	70	71	72	73	80	80	80	80
40	40	40	40	47	48	49	50	70	71	72	73	80	80	80	80
40	40	40	40	47	48	49	50	70	71	72	73	80	80	80	80
40	40	40	40	47	48	49	50	70	71	72	73	80	80	80	80
40	40	40	40	47	48	49	50	70	71	72	73	80	80	80	80
40	40	40	40	47	48	49	50	70	71	72	73	80	80	80	80
40	40	40	40	47	48	49	50	70	71	72	73	80	80	80	80

Block A

Block B

(a) The modified block data



(b) The intensity plot of a row of block data.

Figure 5.5 Block data after modification.

## 5.4.2 Example 2

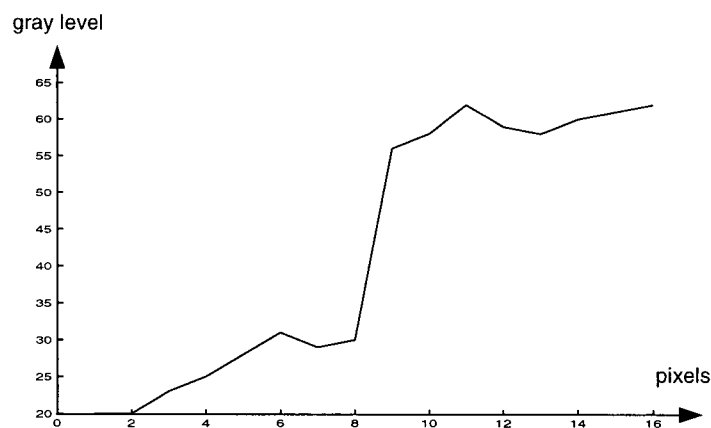
Now let us take a look at another example to see how well does this deblocking method preserve the image details. The original data is listed in Figure 5.6. We can see that there is a big

step at the block boundary and some small intensity variations within blocks. The DCT coefficient matrices of block A, block B are listed in Figure 5.7.

If we apply low-pass spatial filter to the blocks, those small variations will be wiped out. However, after applying our above method, the intensity step at the boundary is reduced a lot while the small variations within the blocks are preserved. The DCT coefficient matrices of block C and the modified coefficients of block C are listed in Figure 5.8. The modified pixel values of block C calculated by (5. 11) and (5. 12) are listed in Figure 5.9.

Block C															
Block A								Block B							
20	20	23	25	28	31	29	30	56	58	62	59	58	60	61	62
20	20	23	25	28	31	29	30	56	58	62	59	58	60	61	62
20	20	23	25	28	31	29	30	56	58	62	59	58	60	61	62
20	20	23	25	28	31	29	30	56	58	62	59	58	60	61	62
20	20	23	25	28	31	29	30	56	58	62	59	58	60	61	62
20	20	23	25	28	31	29	30	56	58	62	59	58	60	61	62
20	20	23	25	28	31	29	30	56	58	62	59	58	60	61	62
20	20	23	25	28	31	29	30	56	58	62	59	58	60	61	62
20	20	23	25	28	31	29	30	56	58	62	59	58	60	61	62

(a) Original block data



(b) The intensity plot of a row of block data

Figure 5.6 Original block data of example 2

$F_a(u, v)$							
206.0	-22.322	-4.656	2.953	0	0.784	3.472	-0.659
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$F_b(u, v)$							
467.0	-7.063	-0.219	-6.906	-4.250	0.844	3.156	1.188
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 5.7 DCT coefficients of Blocks A and B.

$F_c(u, v)$							
353.0	-77.375	3.219	23.156	-4.938	-14.094	-5.156	12.563
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

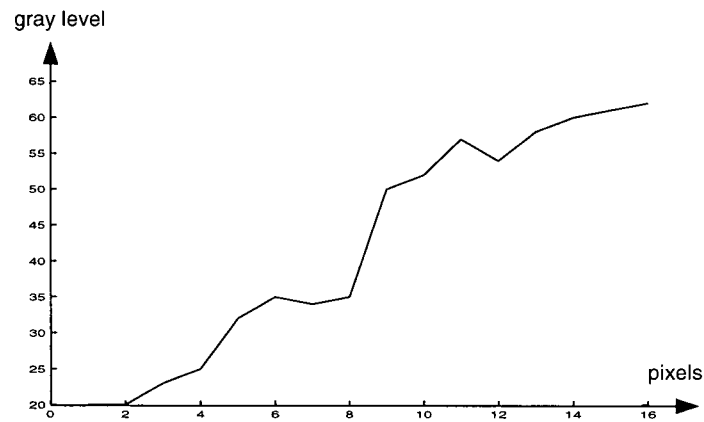
  

$F_{Mc}(u, v)$							
348.188	-52.313	3.219	10.594	-4.938	-7.031	-5.156	6.406
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 5.8 DCT coefficients of Block C before and after modification.

Block C															
Block A								Block B							
20	20	23	25	32	35	34	35	50	52	57	54	58	60	61	62
20	20	23	25	32	35	34	35	50	52	57	54	58	60	61	62
20	20	23	25	32	35	34	35	50	52	57	54	58	60	61	62
20	20	23	25	32	35	34	35	50	52	57	54	58	60	61	62
20	20	23	25	32	35	34	35	50	52	57	54	58	60	61	62
20	20	23	25	32	35	34	35	50	52	57	54	58	60	61	62
20	20	23	25	32	35	34	35	50	52	57	54	58	60	61	62
20	20	23	25	32	35	34	35	50	52	57	54	58	60	61	62

(a) The modified block data



(b) The intensity plot of a row of block data

Figure 5.9 Block data after modification.

## 5.5 Computation cost for blockiness reduction procedure in the DCT domain

The computational cost for blockiness reduction procedure in the DCT domain includes three parts: calculating the DCT coefficients, coefficients modifications and the inverse DCT.

### 5.5.1 The computation cost for calculating DCT coefficients

Most of the computational effort is spent in calculating the DCT coefficients. Let's consider the number of computations needed to remove the appearance of one horizontal and one vertical edge of a block. Let  $8 \times 4$  pixels of the right half of block A and the  $8 \times 4$  pixels of the left

half of block B form a block denoted as block C (see Figure 5.10a). Let  $4 \times 8$  pixels of the bottom half of block A and the  $4 \times 8$  pixels of the upper half of block D form a block denoted as block E (see Figure 5.10b). The DCT coefficients of the original block, block C and block E are required.

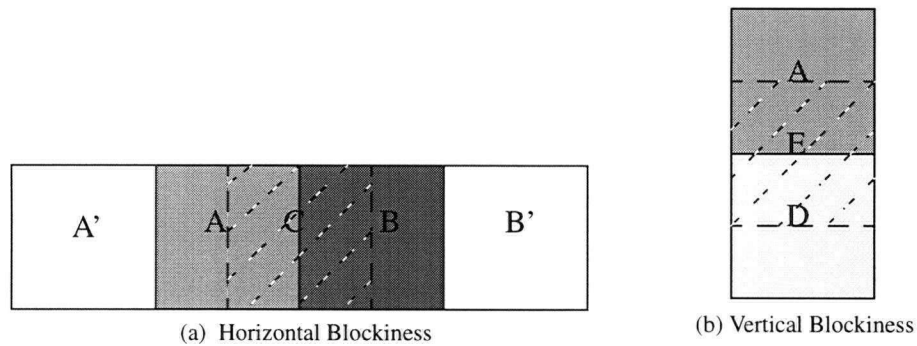


Figure 5.10 Blockiness Diagram.

The horizontal blockiness removal procedure, i.e. removing the vertical edge between block A and B, needs to calculate 5 DCT coefficients for each of the original blocks (A & B) and block C, these are the DC coefficient and the 4 odd-numbered coefficients in the first row of each block. Please note that the results of calculating the 5 DCT coefficients for the first row of blocks A and B will be used in removing the edges between blocks A & A' and B & B'. Similarly, the vertical blockiness removal procedure needs to calculate the DC coefficient and the 4 odd-numbered coefficients in the first column of each original blocks (A & D) and block E. The DCT coefficients for the first column of original blocks can also be used twice. This means to remove the edges between A & B and between A & D four times the cost of calculating the 4 odd-numbered DCT coefficients of one row (or column) and three times the calculation cost of calculating one DC coefficient are required. Let's consider the calculation involved in the first row 2-D DCT coefficients for example.

$$\begin{aligned}
F_c(0, v) &= C_u C_v \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} c(k, l) W_{DCT}(0, k) W_{DCT}(v, l) \\
&= C_u C_v \sum_{l=0}^{N-1} \left( \sum_{k=0}^{N-1} c(k, l) \right) W_{DCT}(v, l).
\end{aligned} \tag{5.13}$$

Thus, this calculation takes two stages. First we calculate the summation of each column. Let

$$\tilde{x}(l) = \sum_{k=0}^{N-1} c(k, l), \quad l = 0 \dots N-1. \tag{5.14}$$

Obviously, this takes 64 additions. Then we take 1-D DCT of  $\tilde{x}(l)$ .

We can use a fast algorithm to calculate the 1-D DCT of  $\tilde{x}(l)$ . Since we only need to calculate the DC and the 4 odd-numbered coefficients, the decimation-in-frequency (DIF) fast DCT may be a good choice. The 1-D Hou algorithm [22] provides a numerically stable, fast algorithm for computing the 1-D DCT coefficients. The logic diagram of calculating the odd-numbered coefficients of the 8-point 1-D DCT is illustrated in Figure 5.11.

We can see 16 additions, 8 multiplications and 4 shifts ( $\times 2$ ) are needed to calculate all the 4 odd-numbered coefficients. Therefore, the total computation cost for calculating the odd-numbered DCT coefficients is  $64+16=80$  additions, 8 multiplications, and 4 shifts. The DC coefficient is the sum of  $\tilde{x}(l)$  ( $l = 0 \dots N-1$ ) and thus this takes 8 additions.

We also need to calculate  $F(3, 3)$  of both block C and block E for constraints checking.  $F(3, 3)$  can be calculated by (5.1). This takes about 64 additions and 64 multiplications.

In summary, the total computation cost needed for calculating the DCT coefficients is:

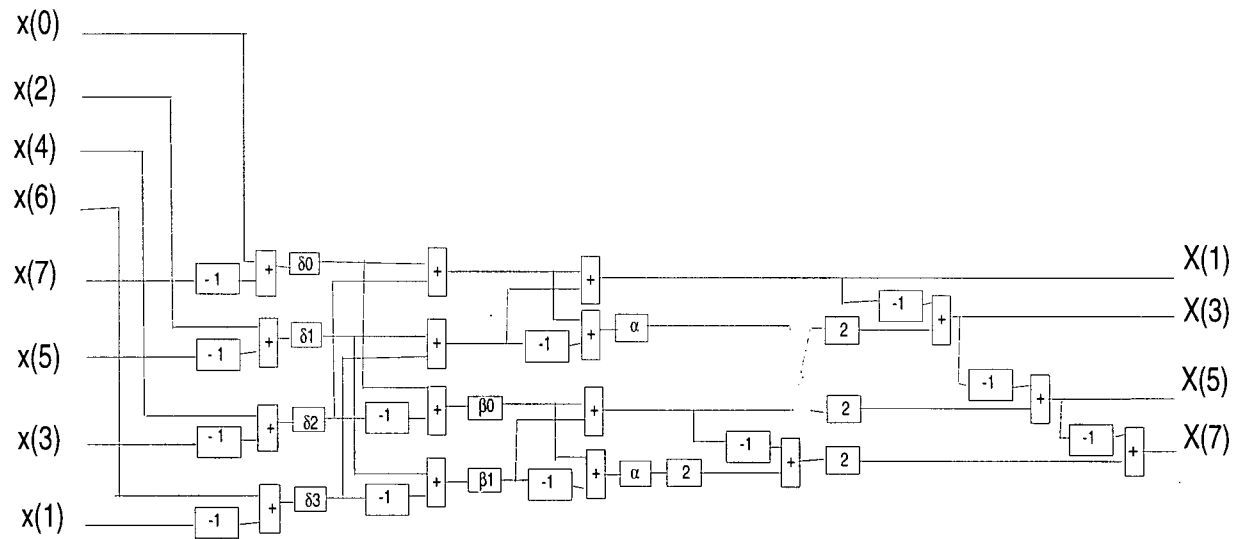


Figure 5.11 Fast 1-D 8 points DCT diagram.

$$80 \times 4 + 8 \times 3 + 64 \times 2 = 472 \text{ additions,}$$

$$8 \times 4 + 64 \times 2 = 160 \text{ multiplications, and}$$

$$4 \times 4 = 16 \text{ shifts.}$$

This makes a total of 648 operations.

This is less than the cost of computing one block 2-D DCT, since calculating one block 2-D DCT is equivalent to calculating 16 1-D DCT transforms and each 1-D DCT transform takes 29 additions, 12 multiplications and 5 shifts according to Hou's fast algorithm. This makes a total of  $(29 + 12 + 5) \times 16 = 736$  operations.

### 5.5.2 The computation cost for coefficients modification

For the horizontal blockiness reduction procedure, the computation cost for constraints checking, equations (5. 5), (5. 6) and (5. 7), will be 2 additions and 3 comparisons. The computa-



tion cost for coefficients modifications, equations (5. 8) and (5. 9), will be  $2 \times 5 = 10$  additions and  $2 \times 5 = 10$  multiplications.

The same number of computations is required in the vertical blockiness reduction procedure. Thus this makes a total of  $(2 + 3 + 10 + 10) \times 2 = 50$  operations.

### 5.5.3 The computation cost for inverse DCT

The rest of computation cost is spent in calculating inverse DCT. The cost of calculating  $\zeta(l)$ ,  $l = 0 \dots 7$  (see equation (5. 12)) is  $8 \times 5 = 40$  additions and  $8 \times 4 = 32$  multiplications. The cost of changing original intensity value by  $\zeta(l)$  will be 64 additions according to equation (5. 11).

Since such procedure is performed twice, one for horizontal blockiness reduction and one for vertical blockiness reduction, the total computation cost for calculating inverse DCT is  $2 \times (40 + 64) = 208$  additions and  $2 \times 32 = 64$  multiplications. This makes a total of 272 operations.

### 5.5.4 Summary

In summary, the total computation cost used for blockiness reduction in DCT domain is  $648 + 50 + 272 = 970$  operations per block. We can carry on the constraints checkings first. If the constraints checkings are failed, then the subsequent computations are not needed.

## 5.6 Spatial filtering

For those boundaries which were detected to have heavy blockiness appearance in the

originally decompressed image, we can further apply a circular-asymmetric two-dimensional filter [13] which is adapted to the orientation of the block boundary. An epsilon-filter [21] may also be applied to remove any blocking artifacts that might have remained. The later filter also helps to reduce other coding artifacts, such as ringing artifacts. The epsilon-filter described in Figure 5.12 is proven to be able to smooth out fairly uniform regions while preserving sharp edges.

Let  $x_{i,j}$  be the pixel value of the image, and  $y_{i,j}$  be its value after applying the epsilon filtering.

$$y_{i,j} = x_{i,j} - \frac{1}{N} \sum_{k=-c}^c \sum_{l=-c}^c f(x_{i,j} - x_{i+k,j+l}), \quad (5.15)$$

where  $f(x)$  is shown in Figure 5.12 and  $N$  is the number of pixels in  $2c \times 2c$  ( $c=1$  or  $2$ ) neighborhood of  $x_{i,j}$

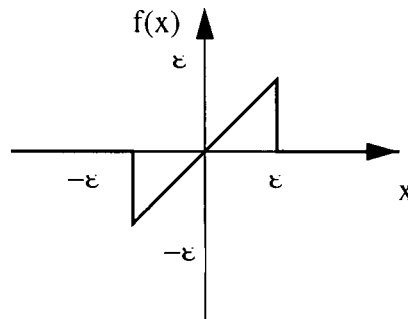


Figure 5.12 Epsilon filter.

The epsilon-filter performs a conditional smoothing operation. The small variations in intensity values of neighboring pixels are taken into account while large variations are ignored, thereby smoothing out fairly uniform regions while preserving sharp edges. The epsilon param-

ter,  $\epsilon$  represents the filtering “strength” of this postfilter. The choice of the value of the parameter epsilon should clearly depend on the strength of the artifacts, i.e. on the quality of the decoded image--lower values of epsilon are desirable when the image quality is high, higher values are desirable when the image quality is poor.

## 5.7 Results

From our experiments, we have found that the proposed algorithm produces better subjective quality and between 0.1 - 1.3 dB increase in PSNR over that of JPEG images and MPEG2 TM5. Figure 5.13 a shows the decoded JPEG image at 0.298 bpp. The post-processing results obtained by our method with and without spatial filtering are shown in Figure 5.13 b and Figure 5.13 c, respectively. As we can see, our method produces better subjective quality than the results obtained by the symmetric or anisotropic filtering. Most of the blocking artifacts in the decoded image after post-processing by our DCT domain filter (without spatial filtering) are removed except for those in the texture regions. However, blocking effects are less noticeable in the texture regions, especially in a video stream. Therefore, when the computation time is critical, spatial filtering may not be applied.

Our method can efficiently eliminate blocking effects even for images coded at very low bit rate. It smooths out the undesired block edges while retaining the sharpness of the image. Figure 5.14 a shows the decoded JPEG image at 0.248 bpp, which is very blocky and the blocking effects are hardly removed by symmetric or anisotropic filtering (see Figure 4.2 b and Figure 4.4 b). Figure 5.14 b shows the image processed by the DCT domain filter only. The blocking effects are almost gone except for several very strong blockiness. After applying spatial filtering, the blocking artifacts are smoothed out completely (Figure 5.14 c). More post-processing results

obtained by our method are shown in Appendix A. The PSNR comparison of several images before and after post-processing at different compression rates are shown in Table 5.1 to Table 5.3.

The main attraction of our algorithm is that we can highly preserve the high frequency components while smoothing out the boundary discontinuities. This is because only the DCT coefficients related to the blocking artifact are modified while other frequency components remain the same. The modification also takes into account the correlation between neighboring blocks. Our algorithm is adaptive to different contents and qualities.

Table 5.1 PSNR comparison of Lena image before and after post-processing.

Compression Bit-Rate (bpp)	.198	.251	.273	.320	.339	.379	.400
PSNR Before Post-Processing (dB)	26.52	27.70	28.24	29.02	29.35	29.91	30.15
PSNR After Post-Processing (dB)	27.42	28.49	28.92	29.56	29.81	30.25	30.43

Table 5.2 PSNR comparison of Indian image before and after post-processing.

Compression Bit-Rate (bpp)	.200	.248	.273	.318	.341	.382	.423
PSNR Before Post-Processing (dB)	26.82	27.78	28.17	28.86	29.15	29.65	30.07
PSNR After Post-Processing (dB)	27.29	28.10	28.49	29.11	29.33	29.73	30.10

Table 5.3 PSNR comparison of Peppers image before and after post-processing.

Compression Bit-Rate (bpp)	.150	.185	.220	.248	.274	.285	.298
PSNR Before Post-Processing (dB)	27.97	29.27	30.39	31.20	31.20	32.21	32.51
PSNR After Post-Processing (dB)	29.11	30.28	31.27	31.97	32.54	32.83	33.09

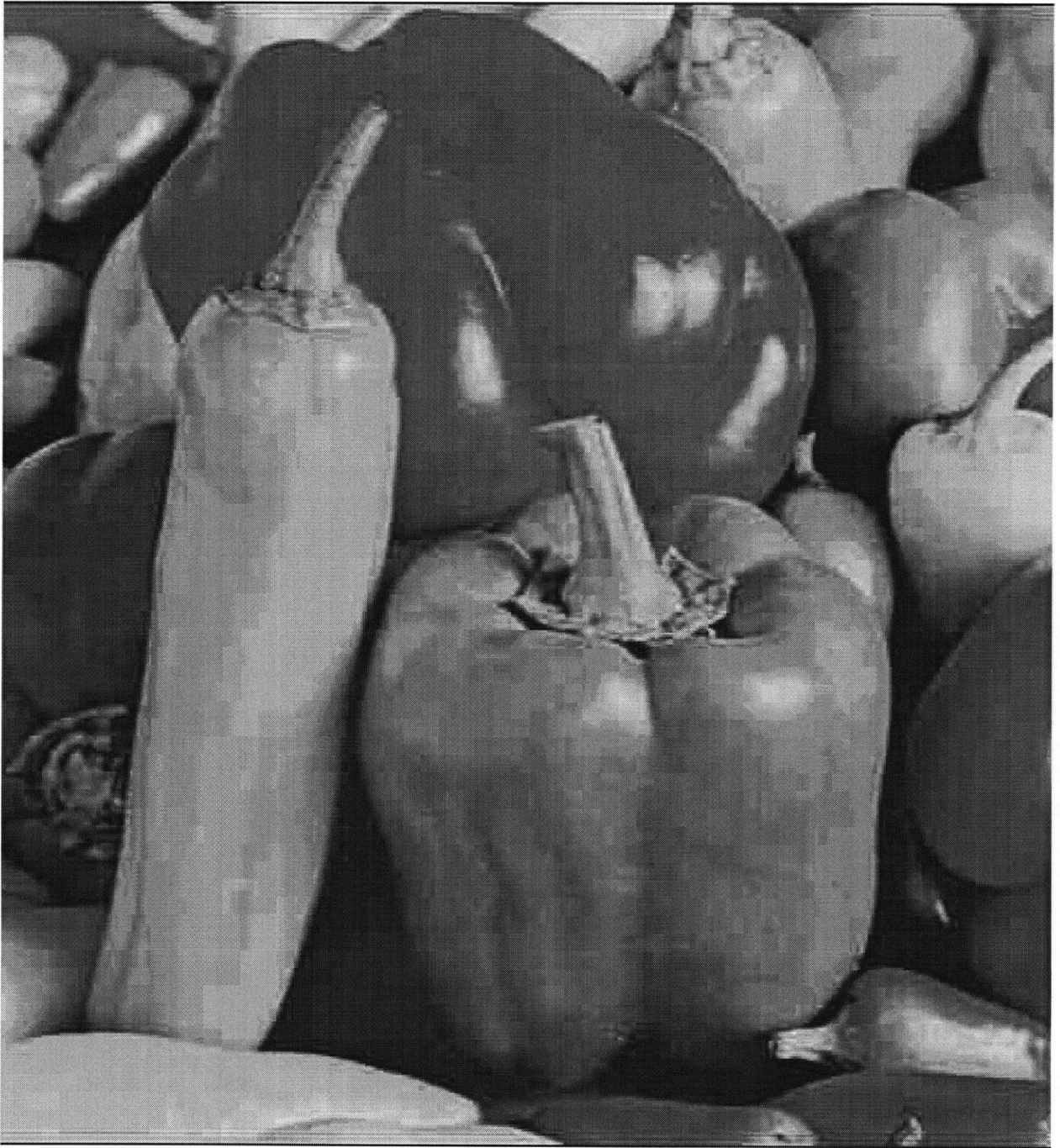


Figure 5.13 a Decompressed image, bitrate=0.298 bpp, PSNR=32.51 dB.

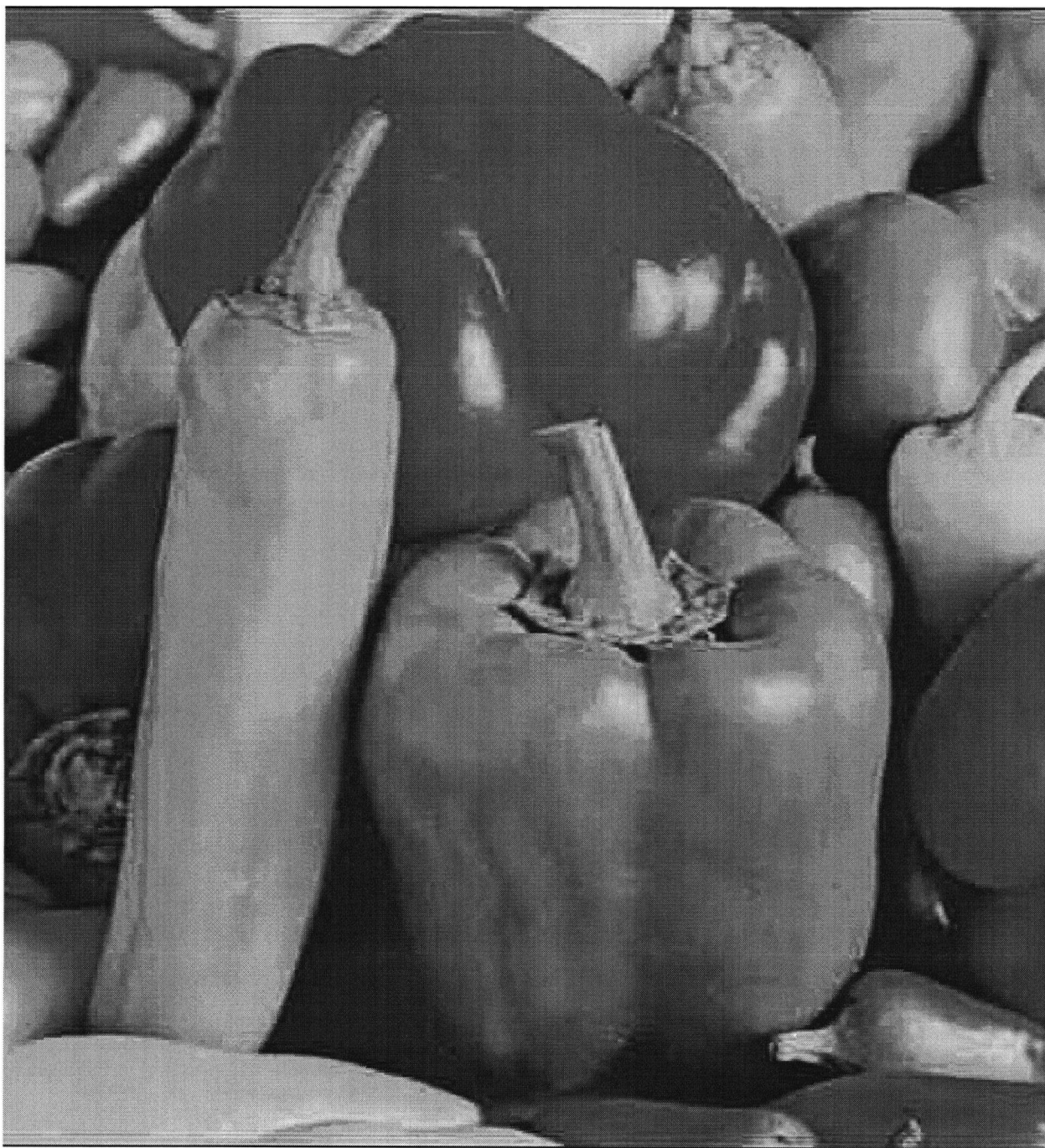


Figure 5.13 b Post-processed image by our DCT domain filter (without spatial filtering),  
bitrate=0.298 bpp, PSNR=32.56 dB.



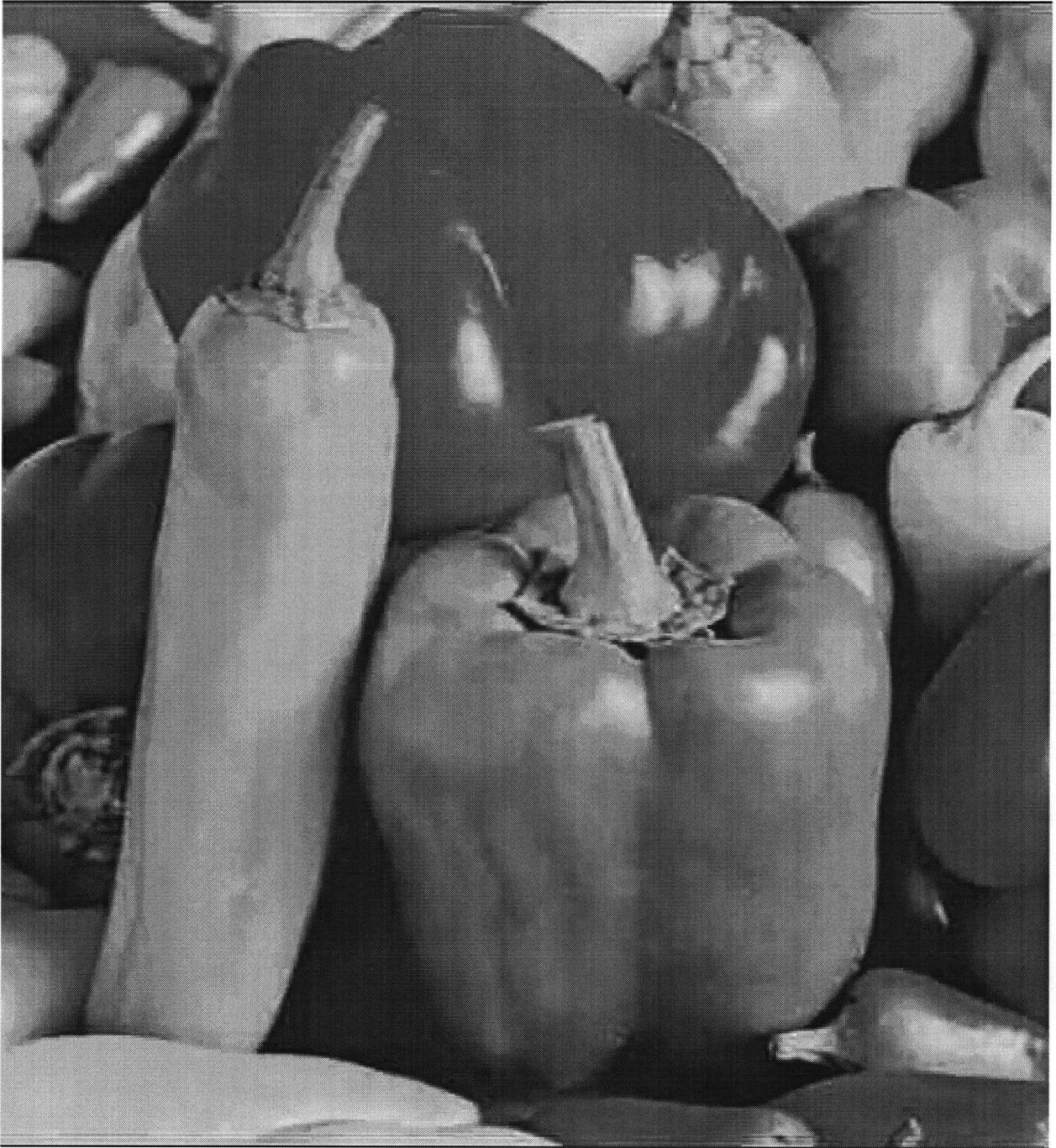


Figure 5.13 c Post-processed image by our algorithm (with spatial filtering),  
bitrate=0.298 bpp, PSNR=33.09 dB.

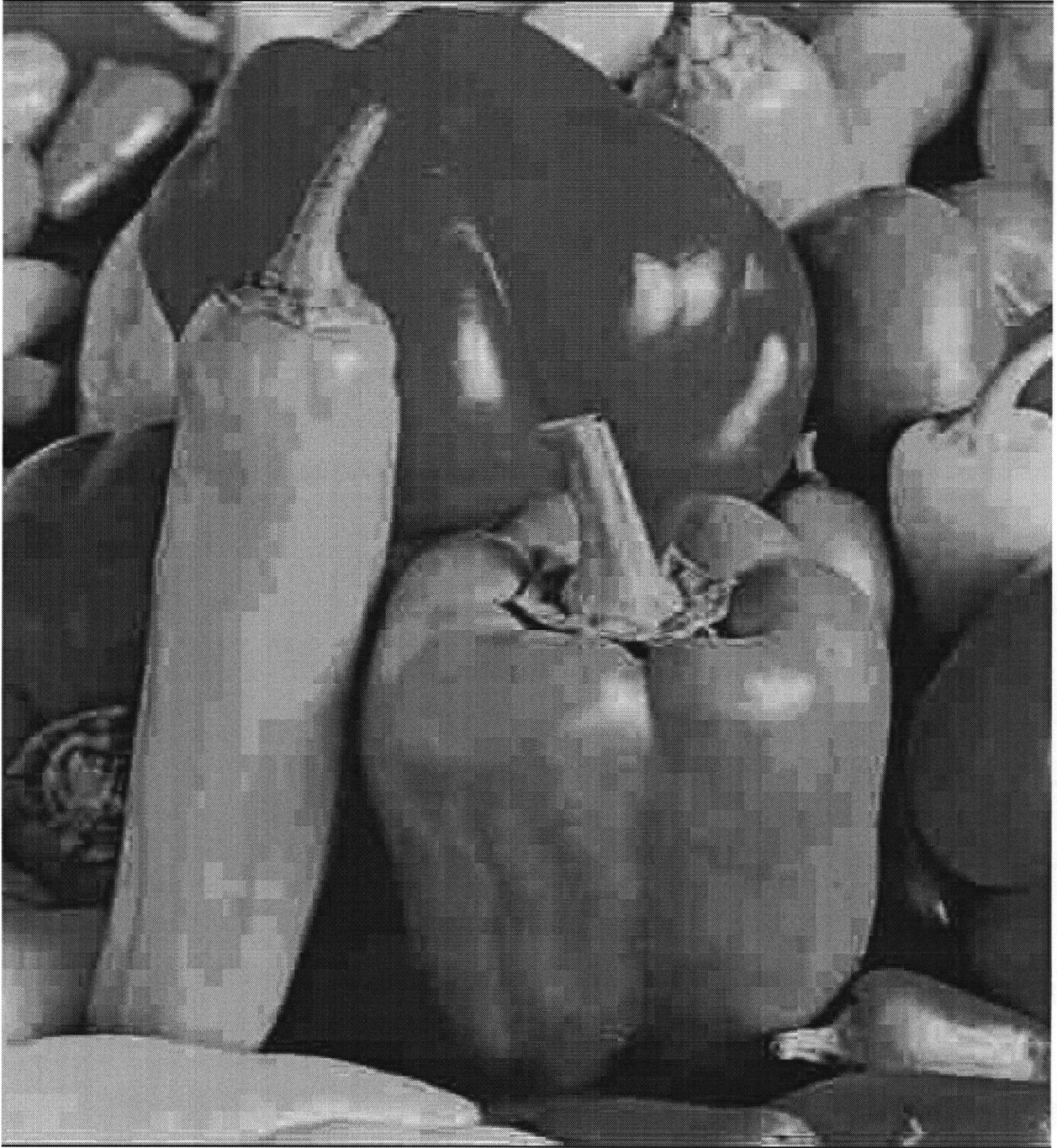


Figure 5.14 a Decompressed image, bitrate=0.248 bpp, PSNR=31.2 dB



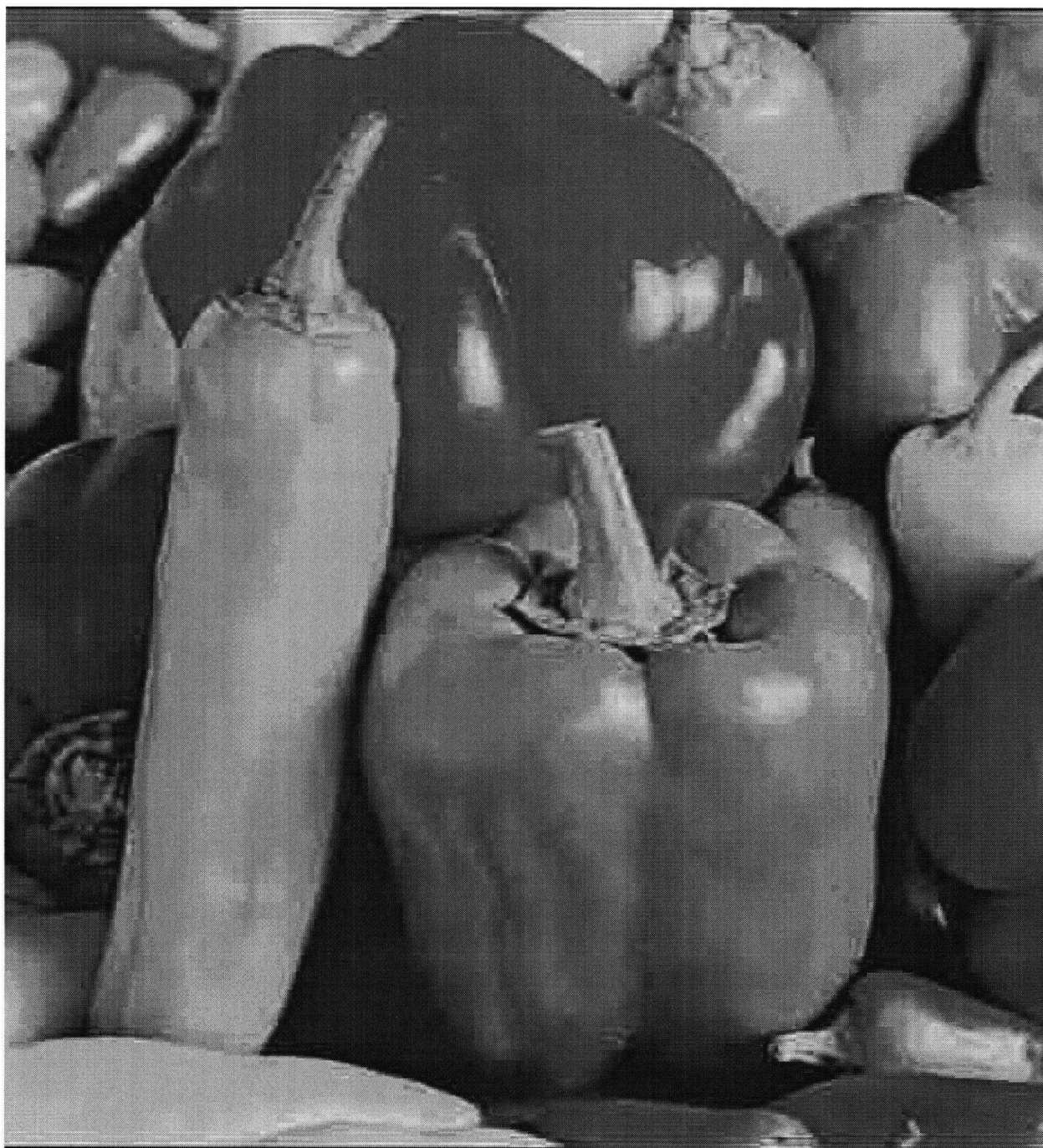


Figure 5.14 b Post-processed image by our DCT domain filter (without spatial filtering),  
bitrate=0.248 bpp, PSNR=31.36 dB.

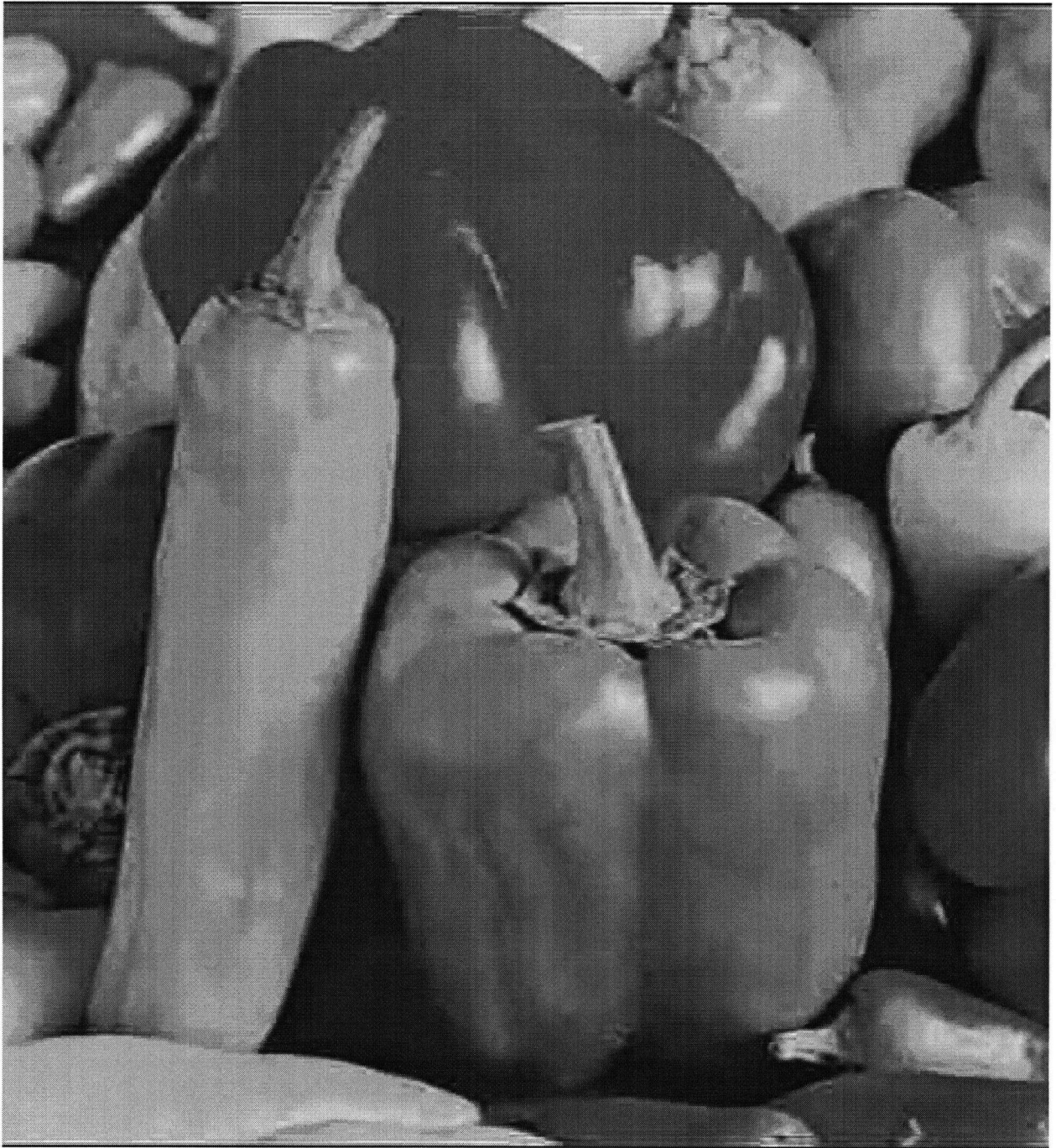


Figure 5.14 c Post-processed image by our algorithm (with spatial filtering),  
bitrate=0.248 bpp, PSNR=31.97 dB.

## **Chapter 6**

### **Conclusions and Future Work**

This thesis addresses artifacts that arise in block based DCT image compression and in specific the blocking effects. The objective of the thesis is to classify, analyze and find solutions to reduce the visual appearance of blocking artifacts. The thesis covers the following topics.

#### **6.1 Compression artifacts classification**

In this part, we classified several most noticeable compression artifacts. The appearance of each type of artifact and their visual perception to human eyes are examined and analyzed. Further more, by analyzing the DCT transform coding and other compression techniques such as motion compression, etc., the cause of each type of artifact is revealed. Understanding the cause and properties of compression artifacts is the basis of artifacts reduction algorithm design.

#### **6.2 The design of blocking artifacts detector**

We first discussed a common method used for blocking artifacts detection. This is the gradient threshold method. We then introduced the difference of slope method. These two methods are edge detectors. However, the difference of slope method is designed according to the properties of the blocking artifacts. As a result, it is able to distinguish genuine edges from those of blocking artifacts. According to several experimental tests, the detection results using our difference of slope method are more accurate than those obtained by the gradient threshold method.

### 6.3 Analysis of blocking artifacts removal techniques

In general, there are two classes of blocking artifacts removal techniques. One is applied during encoding process, the other is applied during or after the decoding process. The blocking artifacts removal techniques applied in the encoding process are proved to be effective in reducing the blocking effects. However, the standard compression codec has to be changed according to each specific implementation. Each method has its own problem too. The bit rate is increased when using block overlapping method. Lapped orthogonal transform requires 20-30 percent more computations.

Blocking artifacts removal techniques applied during or after the decoding process have the advantage of not interfering with the standard encoding processes. This makes the application more flexible.

Several post processing methods have been proposed for blocking artifacts removal. The symmetric, two-dimensional filtering is easy to implement. It needs the least computation efforts too. However, it has the drawback of high-frequency detail loss due to its low-pass filtering nature and it doesn't work at very low bit rate, i.e. less than 1/4 bpp.

Anisotropic filtering is proved to be able to effectively reduce the blocking effects while preserving the sharpness of the image since it is designed to take into account the property of the directional appearance of blocking artifacts. However, this method doesn't work well at very low bit rates either.

To avoid undesired blurring that results from the above filters, adaptive filtering using edge classification is another possible choice. The classification can be done in the spatial or DCT

domains. The spatial filtering scheme is then driven by the output of the edge classifier. Edge and texture regions are usually not filtered since humans are more sensitive to low frequency errors than to high frequency ones. However, it is difficult to detect edges of images with complex content which are coarsely quantized. The block boundaries may be artificially identified as edges.

Adaptive filtering using local statistical estimation is a method which needs to estimate the local quantization errors in the spatial domain. Since most of the high order DCT coefficients are quantized to zero after coarse quantization, it is difficult to get a good estimate of the local spatial quantization error. Another drawback of this method is that the algorithm has to access data from decoder. Thus it has to be built inside the decoder.

Iterative blocking artifact removal technique is another powerful spatial-adaptive approach. The major drawback of this approach is its high computational complexity. It has to be built inside the decoder too.

Dithering is a method of providing masking effects and illusion of continuous-tone pictures by introducing a carefully-controlled, random dither signal which conforms to a model of the average energy distribution of the transform coefficients prior to quantization during the decoding process.

#### **6.4 A new blockiness reduction scheme**

We propose a new approach which performs blockiness reduction in the DCT domain to reduce the block-to-block discontinuities present in the spatial domain. Our algorithm takes advantage of the fact that the original pixel levels in the same block are of good continuity and it

uses this property and the correlation between the neighboring blocks to reduce the discontinuity of the pixels across the boundaries. Our algorithm can highly preserve the high frequency components while smoothing out the boundary discontinuity. Simulation results show that the proposed algorithm significantly reduces the blocking artifacts in both objective and subjective measures. It is able to be adaptive to different image content and qualities. It can effectively remove blocking artifacts even at very low bit rate. The amount of computation it requires is also acceptable compared to iterative blocking artifact removal techniques.

## **6.5 Future work**

Blocking artifact is the most noticeable degradation resulting from block based compression. The new algorithm proposed in this thesis is proved to be a powerful blockiness reduction method. However, since it requires a large number of computations, the design of a real-time version algorithm to be applied in the video signal processing is very important.

Mosquito noise is another prominent noise of motion picture compression. Unlike the blocking artifacts, mosquito effects are difficult to detect and thus are difficult to remove. Further study of mosquito detection and removal forms worthwhile further research.

## Bibliography

- [1] Gregory K. Wallace, "The JPEG Still Picture Compression Standard," Communications of the ACM, April 1991, Vol.34, No. 4, pp:29-44.
- [2] Didier Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications," Communications of The ACM, April 1991, Vol. 34, No. 4, pp:47-58.
- [3] ISO/IEC 13818-2, 1995.
- [4] "Guide to the use of the ATSC Digital Television Standard," Advanced Television Systems Committee.
- [5] Anil K. Jain, "Fundamentals of Digital Image Processing," Prentice Hall, Englewood Cliffs, 1989.
- [6] Mark J.T. Smith, Alen Docef, "A study guide for digital Image Processing," Scientific Publisher, 1997.
- [7] H.Reeve and J. Lim, "Reduction of Blocking Effect in Image Coding," Proc. ICASSP'83, pp.1212-1215, 1983.
- [8] H.R.Wu, M.Yuen and B.Qiu, "Video Coding Distortion Classification and Quantitative Impairment Metrics," Proceedings of ICSP' 96, 962-965.
- [9] H.C. Kim and H.W. Park, "Signal Adaptive Post-processing for Blocking Effects Reduction in JPEG Image," IEEE International Conference on Image Processing . v2 , 1996, 41-44.
- [10] J.D. McDonnell, R.N. Shorten and A.D. Fagan, "An Edge Classification Based Approach To The Post-Processing of Transform Encoded Images," Proc. ICASSP '94, (V): 329-332, 1994.
- [11] Shigenobu Minami and Avidesh Zakhori, "An Optimization Approach for Removing Blocking Effects in Transform Coding," IEEE Transactions On Circuits And Systems For Video Technology, Vol. 5, No. 2, pp:74-82, April, 1995.

- [12] Yongyi Yang, Nikolas P. Galatsanos and Aggelos K. Katsaggelos, "Projection-Based Spatially Adaptive Reconstruction of Block-Transform Compressed Images," *IEEE Transactions On Image Processing*, Vol. 4, No. 7, pp: 896-908 July 1995.
- [13] Kou-Hu Tzou, "Post-Filtering of Transform-Coded Images," *SPIE Vol. 974 Applications of Digital Image Processing XI* (1988), pp:121-126.
- [14] Lee, H.C. Kim, and H.W. Park, "Blocking Effect Reduction of JPEG Images by Signal Adaptive Filtering," *IEEE Transactions on Image Processing*, Vol. 7, No. 2, pp:229-234, Feb. 1998.
- [15] William E. Lynch, Amy R. Reibman and Bede Liu, "Post Processing Transform Coded Images Using Edges," *Proc. ICASSP'95*, pp: 2323-2326, 1995.Y.L.
- [16] Thomas Meier, King N. Ngan and Gregory Crebbin, "A Region-based Algorithm for Enhancement of Images Degraded by Blocking Effects," *IEEE*, 1996, 405:408
- [17] Henrique S. Malvar and David H. Staelin, "The LOT: Transform Coding Without Blocking Effects," *IEEE Transactions On Acoustics, Speech, and Signal Processing*, Vol. 37, No. 4, pp: 553-559, April 1989.
- [18] Jim Chou, Matthew Crouse and Kannan Ramchandran, "A Simple Algorithm for Removing Blocking Artifacts in Block-Transform Coded Images," *IEEE Signal Processing Letters*, Vol. 5, No.2, pp: 33-35, Feb. 1998.
- [19] Jianping Hu, Nadir Sinaceur, Fu Li, Kwok-Wai Tam, Zhigang Fan, "Removal of Blocking and Ringing Artifacts In Transform Coded Images," *Proc. ICASSP'97*, pp: 2565-2568.
- [20] David L. McLaren, "Removal of Subjective Redundancy From DCT-coded Images," *IEE Proceedings-I*, Vol. 138, No. 5, Oct. 1991, pp:345-350.
- [21] Arnaud Jacquin, and Hiroyuki Okada, "Content-Adaptive Postfiltering for Very Low Bit Rate Video," *Data Compression Conference*, 1997. pp: 111-120.
- [22] Hsieh S. Hou, "A Fast Recursive Algorithm for Computing the Discrete Cosine Transform," *IEEE Tran. on ASSP*, Vol. 35, No. 10, Oct. 1987, pp:1455-1461.
- [23] Si Jun Huang, "Adaptive Noise Reduction and Image Sharpening for Digital Video Compression," *Proc. of the IEEE International Conference on Systems Man and Cybernetics*. v4, 1997, pp:3142-3147.



## Appendix A.



Figure A.1 Decompressed Lena image, bitrate=0.199 bpp, PSNR=30.40 dB.



Figure A.2 Post-processed Lena image, bitrate=0.199 bpp, PSNR=31.05 dB.





Figure A.3 Decompressed Lena image, bitrate=0.271 bpp, PSNR=31.95 dB.



Figure A.4 Post-processed Lena image, bitrate=0.271 bpp, PSNR=32.33 dB.



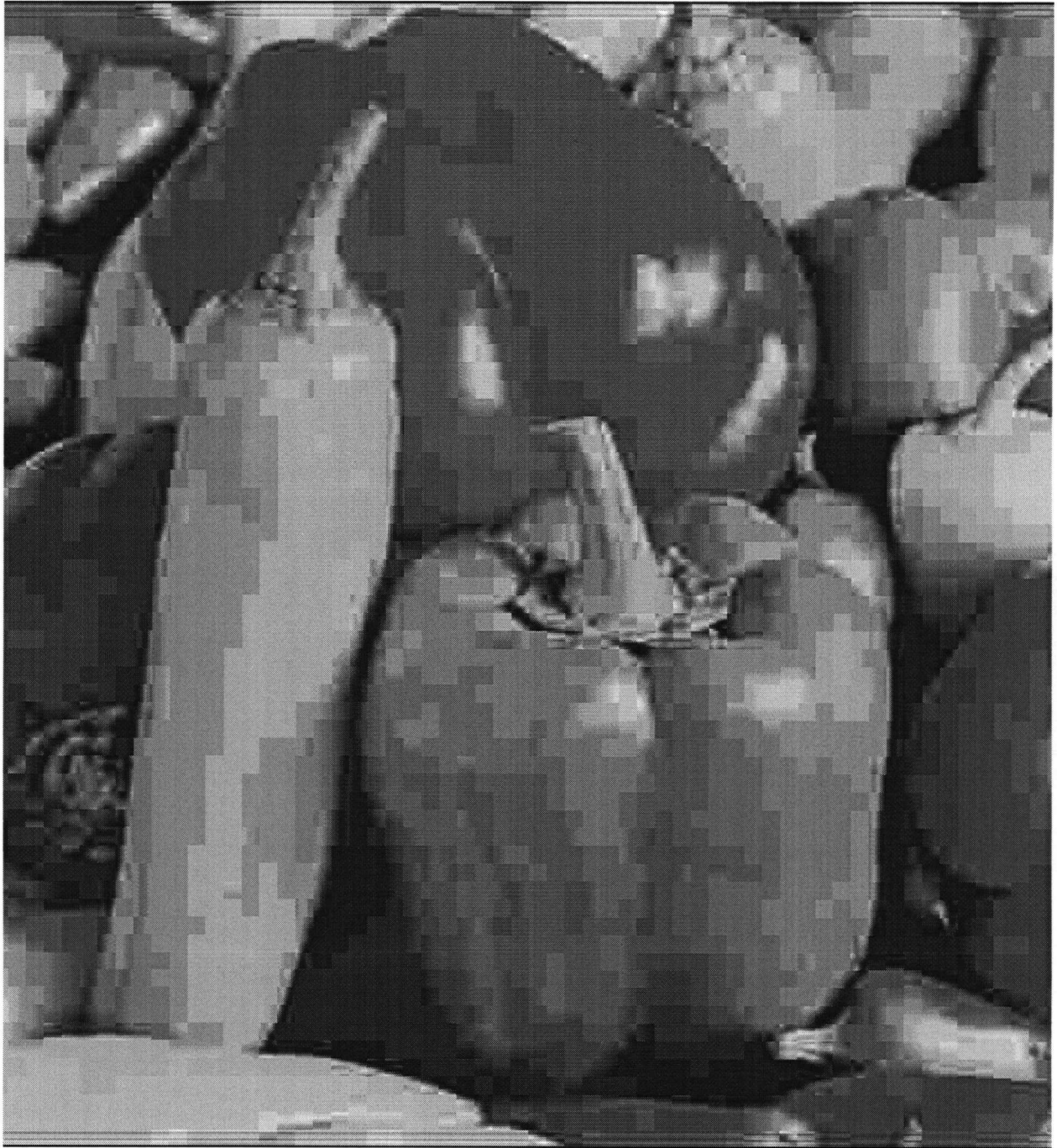


Figure A.5 Decompressed Peppers image, bitrate=0.15 bpp, PSNR=27.97 dB.

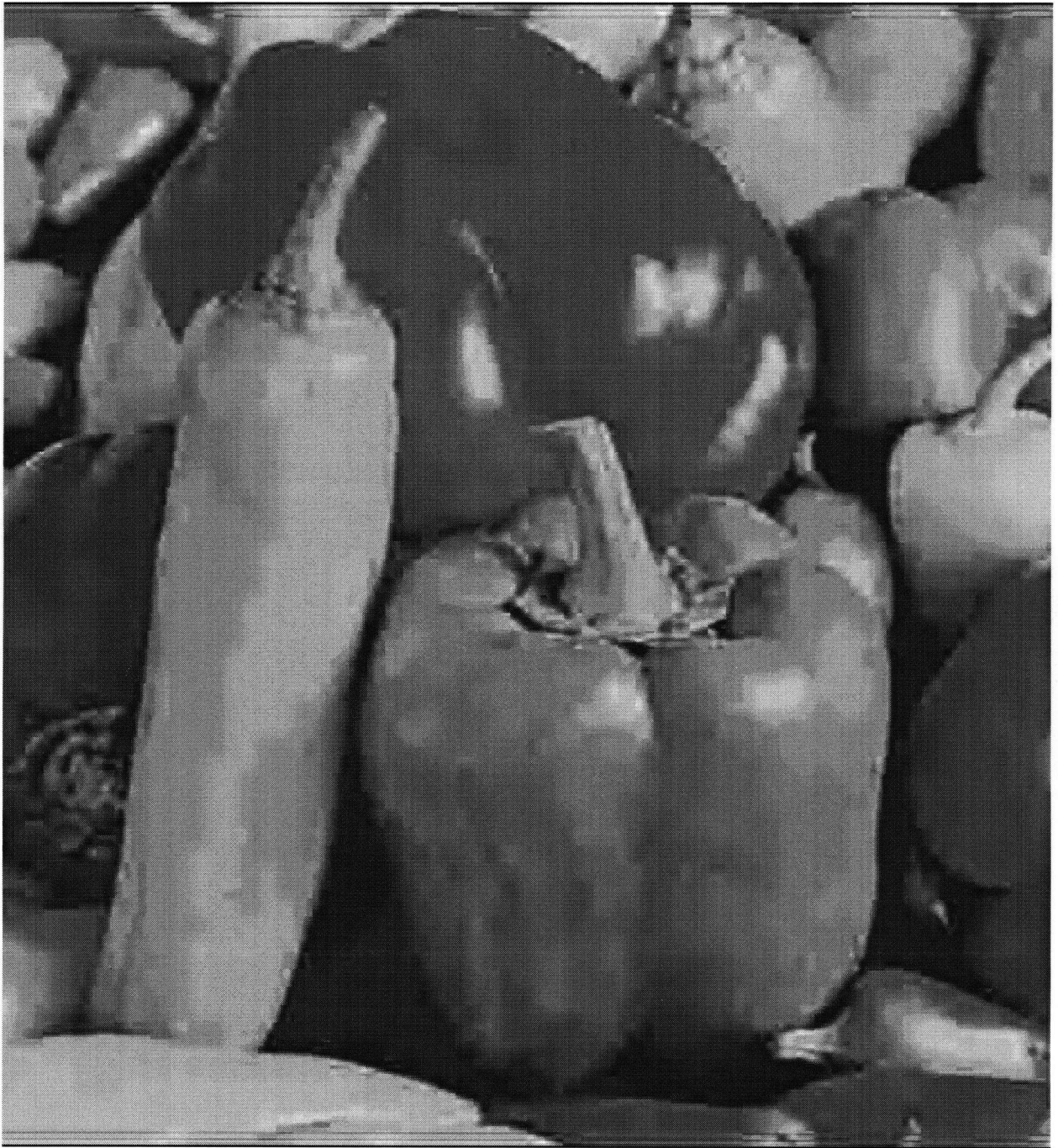


Figure A.6 Post-processed Peppers image, bitrate=0.15 bpp, PSNR=29.11 dB.



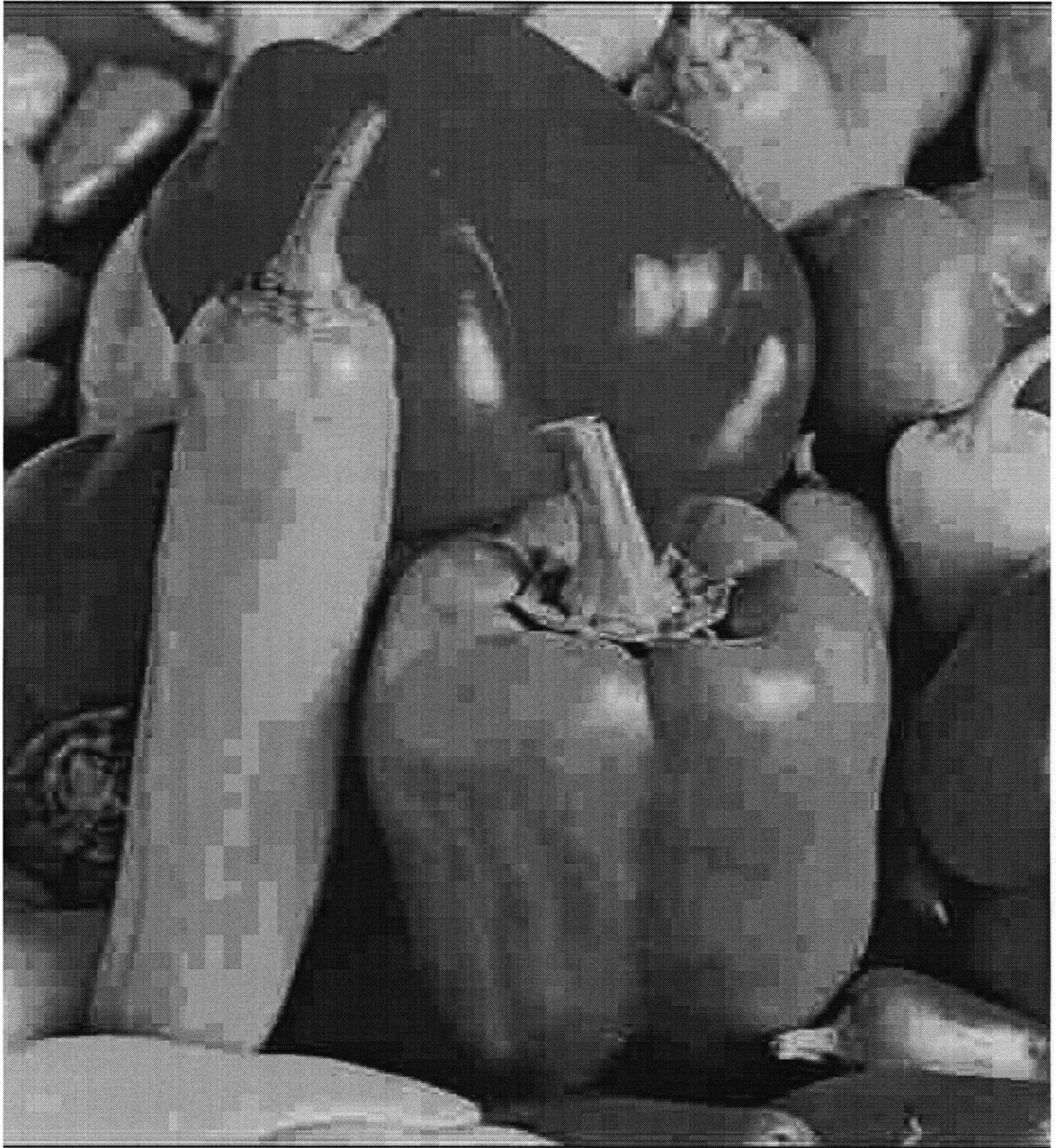


Figure A.7 Decompressed Peppers image, bitrate=0.22 bpp, PSNR=30.39 dB.

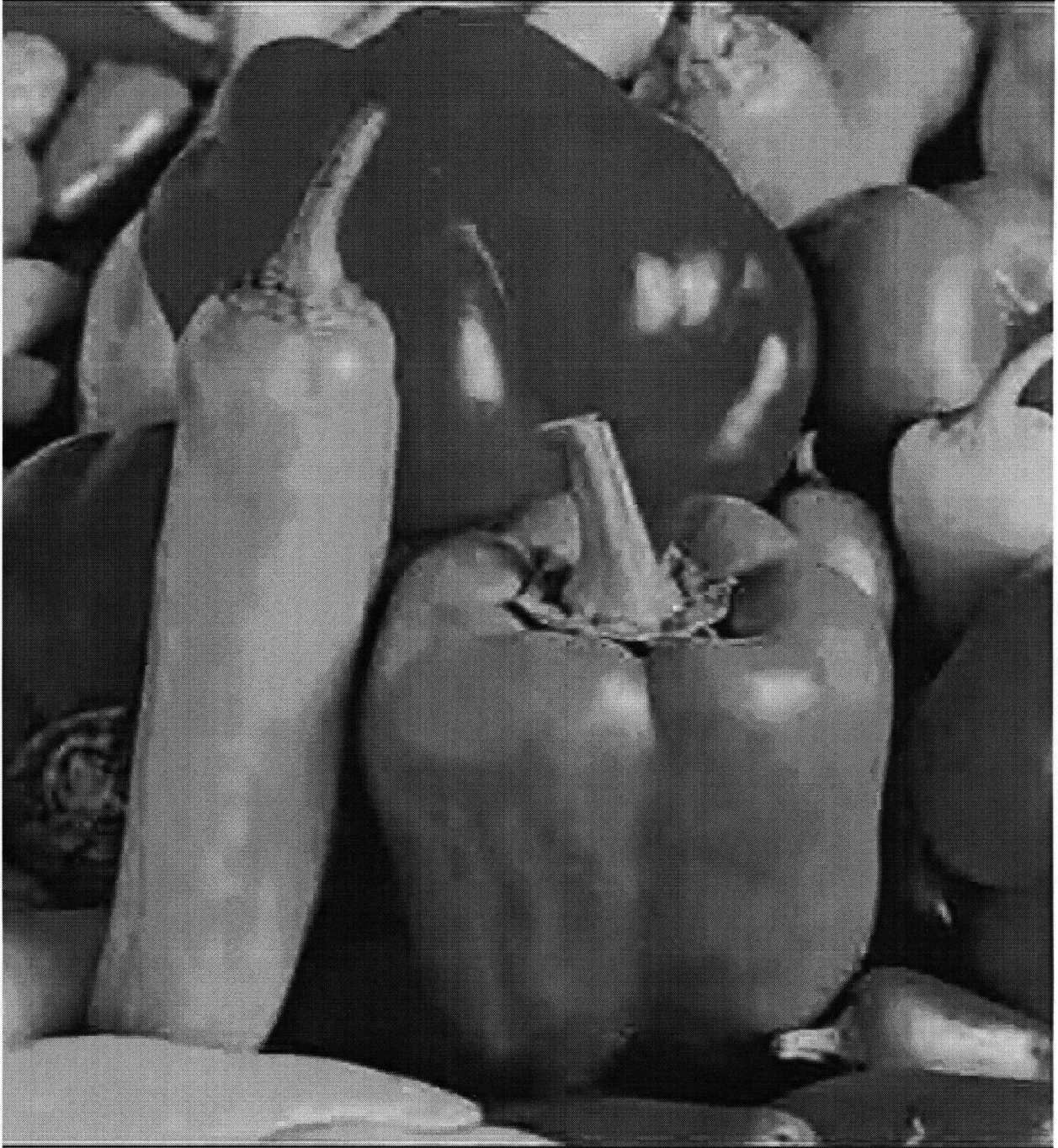


Figure A.8 Decompressed Peppers image, bitrate=0.22 bpp, PSNR=31.27 dB.





Figure A.9 Decompressed Indian image, bitrate=0.273 bpp, PSNR=28.17 dB.



Figure A.10 Post-processed Indian image, bitrate=0.273 bpp, PSNR=28.50 dB.



Figure A.11 Decompressed Indian image, bitrate=0.382 bpp, PSNR=29.65 dB.





Figure A.12 Post-processed Indian image, bitrate=0.382 bpp, PSNR=29.74 dB.