

Compression and Restoration of Noisy Images

A THESIS

Presented to

The Academic Faculty

By

Osama K. Al-Shaykh

In Partial Fulfillment

of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Georgia Institute of Technology

October, 1996

Copyright © 1996 by Osama K. Al-Shaykh

Compression and Restoration of Noisy Images

Approved:



Russell M. Mersereau, Chairman



Monson H. Hayes III



Mark J. (T.) Smith

Date approved by Chairman 2 Dec. 1996

To my parents, brothers, and sister

Acknowledgments

Praise be to God, Most Gracious, Most Merciful.

I thank my advisor, Prof. Russell M. Mersereau, for guiding me and at the same time encouraging me explore different ideas. His continuous support, encouragement, and advice were invaluable. I also thank Prof. Monson H. Hayes and Prof. Mark J. T. Smith for their time and suggestions, as well as, Prof. Mark A. Clements, Prof. Ronald W. Schafer, Prof. Douglas B. Williams, and the rest of the DSP group at Georgia Tech. Kay Gilstrap and Stacy Schultz were so helpful organizing every step efficiently. Mysara Younis and Ragnar Jonsson's help during my first year at Georgia Tech was great and will always be appreciated.

I also thank Edward Dibella for his help using the bull's-eye programs, Andy Barclay and Robert Eisner for providing the SPECT cardiac sequences and for their suggestions. I would also like to thank Dr. Issam Zabalawi for introducing me to the field of image processing, Dr. John F. Doherty for advising me during my stay at Iowa State University, Dr. Hisham Hamdan, and Muhamad K. Abdelazzez for their advice.

On a personal level, my stay in Atlanta could not have been the same without Molham and Manal Aref. They are the best of friends. I would like to thank many more people than I can list here but I'll try: Abdulrahman, Naser Adas, Slim Alouini, Joe Arrowood, Haluk Aydinoglu, Ramzi Barghouthi, Mohamad Ben Romdhane (Mammo), Muhammed Coban (Shobanitsky), Mazhar Elhaj, Hazem and Samar Elsabbagh, Salvador Garcia (Sal), Antonio and Evana Gentile, Khalid and Nida

Hamied, Rahmi Hezar, Timothy John (TJ) Klausutis, Stephen Kogon, Wassef Masri, Hisham Mharmeh, Ali and Aisha Mokkdad, Richard Rao, Balasubramaniam (Balu) Santhanam, Jonathan Su, Hatem Sellami, Radwan and Ahlam Shaaban, Mysara Younis, and Mahdi Zeidan. Sundays and Wednesdays would not have been the same without the soccer games, so thank you guys for these nice games.

I wish to express my gratitude to my family: Khalid, Laila, Obayda, Moa'ath, and Eyas for their love, patience and continuous support. They have always been there for me and I shall always be grateful.

Contents

Acknowledgments	iii
Contents	v
List Of Tables	xii
List Of Figures	xiii
Summary	xvii
1 INTRODUCTION	1
1.1 Outline	5
2 IMAGE COMPRESSION	6
2.1 Introduction	6
2.2 Source Coding Theory	7
2.3 Entropy coding	8
2.4 Quantization	9
2.4.1 Memoryless vector quantization	10
2.4.2 Memory vector quantization	12
2.5 Image Decomposition	13
2.5.1 Linear predictive coding (LPC)	13
2.5.2 Transform coding	14
2.5.3 Subband coding	15

3	NOISE AND ITS EFFECT ON COMPRESSION	18
3.1	Introduction	18
3.2	Gaussian Noise	19
3.3	Poisson Noise	22
3.4	Film Grain Noise	23
3.5	Coding Noisy Images	27
4	IMAGE RESTORATION	31
4.1	Introduction	31
4.2	MMSE Restoration of Noisy Images	31
4.3	Markov Random Fields	32
4.4	Computing the MMSE Estimate	34
4.5	Poisson Noise Case	35
4.6	Film Grain Noise Case	38
4.7	Parameter Estimation and its Effect on Restoration	43
5	COMPRESSION OF NOISY IMAGES	51
5.1	Introduction	51
5.2	Optimal Noisy Coders	52
5.2.1	Poisson noise case	52
5.2.2	Film grain noise case	53
5.3	The Effect of Preprocessing on Conventional Coders	53
5.3.1	Poisson noise case	56
5.3.2	Film grain noise case	56
6	POSTPROCESSING OF LOSSY COMPRESSED IMAGES	60
6.1	Introduction	60
6.2	Postprocessing of Lossy Compressed Images	61
6.3	Modeling Blocking Artifacts	62
6.4	Noise Free Case	64

6.5	Poisson Noise Case	69
6.6	Film Grain Noise Case	77
6.7	Estimation of Restoration Parameters	80
7	APPLICATION: COMPRESSION OF CARDIAC SEQUENCES	82
7.1	Introduction	82
7.2	Medical Image Compression	83
7.3	Evaluating the Quality of Lossy Compressed Medical Images	85
7.3.1	Distortion measures	86
7.3.2	Subjective ratings	87
7.3.3	Diagnostic accuracy	87
7.4	Segmented Coding of SPECT Images	91
7.4.1	Region-based scalar quantization of cardiac sequences	94
7.4.2	Region-based vector quantization of cardiac sequences	94
7.5	Preprocessing of SPECT Cardiac Sequences	96
7.6	Example	96
7.6.1	Effect of preprocessing	102
8	SUMMARY AND SUGGESTIONS FOR FURTHER RESEARCH	112
8.1	Summary	112
8.2	Novel Accomplishments of the Thesis	114
8.3	Suggestions for Further Research	114
Appendix A DEVELOPMENT OF THE RESTORATION		
ALGORITHMS		116
Appendix B GENERALIZED GAUSSIAN DISTRIBUTION		119
Appendix C EXAMPLES OF CODING CARDIAC SEQUENCES		121
C.1	Rest Example	122

BIBLIOGRAPHY

132

VITA

148

List of Tables

7.1	Performance of various lossless coding methods	86
7.2	Region-based scalar quantization of the <i>original</i> cardiac sequence . .	97
7.3	Region-based vector quantization of the <i>original</i> cardiac sequence . .	97
7.4	The averages of the 10 regions of the <i>original</i> sequence when using the SCALAR coder	100
7.5	The averages of the 10 regions of the <i>original</i> sequence when using the VECTOR coder	100
7.6	The standard deviations of the 10 regions of the <i>original</i> sequence when using the SCALAR coder	101
7.7	The standard deviations of the 10 regions of the <i>original</i> sequence when using the VECTOR coder	101
7.8	The averages of the septal, lateral, anterior, and inferior regions and the corresponding ratios of the <i>original</i> sequence when using the SCALAR coder	104
7.9	The averages of the septal, lateral, anterior, and inferior regions and the corresponding ratios of the <i>original</i> sequence when using the VEC- TOR coder	104
7.10	Region-based SCALAR quantization of the <i>filtered</i> cardiac sequence . .	105
7.11	Region-based VECTOR quantization of the <i>filtered</i> cardiac sequence . .	106
7.12	The averages of the 10 regions of the <i>filtered</i> sequence when using the SCALAR coder	109

7.13	The averages of the 10 regions of the <i>filtered</i> sequence when using the VECTOR coder	109
7.14	The standard deviations of the 10 regions of the <i>filtered</i> sequence when using the SCALAR coder	110
7.15	The standard deviations of the 10 regions of the <i>filtered</i> sequence when using the VECTOR coder	110
7.16	The averages of the septal, lateral, anterior, and inferior regions and the corresponding ratios of the <i>filtered</i> sequence when using the SCALAR coder	111
7.17	The averages of the septal, lateral, anterior, and inferior regions and the corresponding ratios of the <i>filtered</i> sequence when using the VECTOR coder	111
C.1	REST EXAMPLE: The averages of the 10 regions of the <i>original</i> sequence when using the SCALAR coder	126
C.2	REST EXAMPLE: The averages of the 10 regions of the <i>original</i> sequence when using the VECTOR coder	126
C.3	REST EXAMPLE: The averages of the 10 regions of the <i>filtered</i> sequence when using the SCALAR coder	127
C.4	REST EXAMPLE: The averages of the 10 regions of the <i>filtered</i> sequence when using the VECTOR coder	127
C.5	REST EXAMPLE: The standard deviations of the 10 regions of the <i>original</i> sequence when using the SCALAR coder	128
C.6	REST EXAMPLE: The standard deviations of the 10 regions of the <i>original</i> sequence when using the VECTOR coder	128
C.7	REST EXAMPLE: The standard deviations of the 10 regions of the <i>filtered</i> sequence when using the SCALAR coder	129
C.8	REST EXAMPLE: The standard deviations of the 10 regions of the <i>filtered</i> sequence when using the VECTOR coder	129

C.9 REST EXAMPLE: The averages of the septal, lateral, anterior, and inferior regions and the corresponding ratios of the <i>original</i> sequence when using the SCALAR coder	130
C.10 REST EXAMPLE: The averages of the septal, lateral, anterior, and inferior regions and the corresponding ratios of the <i>original</i> sequence when using the VECTOR coder	130
C.11 REST EXAMPLE: The averages of the septal, lateral, anterior, and inferior regions and the corresponding ratios of the <i>filtered</i> sequence when using the SCALAR coder	131
C.12 REST EXAMPLE: The averages of the septal, lateral, anterior, and inferior regions and the corresponding ratios of the <i>filtered</i> sequence when using the VECTOR coder	131
C.13 STRESS EXAMPLE: The averages of the 10 regions of the <i>original</i> sequence when using the SCALAR coder	136
C.14 STRESS EXAMPLE: The averages of the 10 regions of the <i>original</i> sequence when using the VECTOR coder	136
C.15 STRESS EXAMPLE: The averages of the 10 regions of the <i>filtered</i> sequence when using the SCALAR coder	137
C.16 STRESS EXAMPLE: The averages of the 10 regions of the <i>filtered</i> sequence when using the VECTOR coder	137
C.17 STRESS EXAMPLE: The standard deviations of the 10 regions of the <i>original</i> sequence when using the SCALAR coder	138
C.18 STRESS EXAMPLE: The standard deviations of the 10 regions of the <i>original</i> sequence when using the VECTOR coder	138
C.19 STRESS EXAMPLE: The standard deviations of the 10 regions of the <i>filtered</i> sequence when using the SCALAR coder	139
C.20 STRESS EXAMPLE: The standard deviations of the 10 regions of the <i>filtered</i> sequence when using the VECTOR coder	139

C.21 STRESS EXAMPLE: The averages of the septal, lateral, anterior, and inferior regions and the corresponding ratios of the <i>original</i> sequence when using the SCALAR coder	140
C.22 STRESS EXAMPLE: The averages of the septal, lateral, anterior, and inferior regions and the corresponding ratios of the <i>original</i> sequence when using the VECTOR coder	140
C.23 STRESS EXAMPLE: The averages of the septal, lateral, anterior, and inferior regions and the corresponding ratios of the <i>filtered</i> sequence when using the SCALAR coder	141
C.24 STRESS EXAMPLE: The averages of the septal, lateral, anterior, and inferior regions and the corresponding ratios of the <i>filtered</i> sequence when using the VECTOR coder	141

List of Figures

1.1	A $32 \times 64 \times 64$ SPECT cardiac image sequence	3
2.1	Image compression system	7
2.2	The block diagram of a communications system	8
2.3	LBG vector quantizer design algorithm	11
2.4	Linear prediction vector quantizer	12
2.5	Linear predictive coding	13
2.6	Subband decomposition	16
2.7	Pyramid structure	17
3.1	Images used to study the effect of Gaussian noise	19
3.2	The effect of Gaussian noise on Lloyd-Max quantizers and JPEG	21
3.3	The effect of the Gaussian noise level on compression	22
3.4	Images used to study the effect of Poisson noise	23
3.5	The effect of Poisson noise on Lloyd-Max quantizers and JPEG	24
3.6	Images used to study the effect of film-grain noise	25
3.7	The effect of film-grain noise on Lloyd-Max quantizers and JPEG	26
3.8	The effect of the film-grain noise level on compression	26
3.9	The effect of γ on compression	27
3.10	The block diagram of a communication system	28
4.1	A first-order neighborhood and first-order clique types	32
4.2	The effect of choosing a Clique potential	34

4.3	Restoration of the modified Shepp-Logan phantom corrupted by Poisson noise	39
4.4	Restoration of the original Shepp-Logan phantom corrupted by Poisson noise	40
4.5	Restoration of LENNA image corrupted by Poisson noise	41
4.6	Restoration of LENNA image corrupted by film grain noise with $\alpha = 1$ and $\gamma = 0.5$	44
4.7	Restoration of LENNA image corrupted by film grain noise with $\alpha = \sqrt{20}$ and $\gamma = 1/3$	45
4.8	The effect of δ on the energy function	46
4.9	The effect of δ on image restoration	46
4.10	LENNA image restored using different δ values	47
4.11	The effect of γ on the level of film-grain noise	48
4.12	LENNA image restored using different γ values	49
4.13	Restoration of images corrupted by Poisson noise using film-grain noise assumptions	49
5.1	A block diagram describing the configuration for optimally compressing noisy images	51
5.2	The performance of the optimal noisy scalar quantizer for the Poisson noise case	54
5.3	The performance of the optimal noisy quantizer for the Poisson noise case	54
5.4	The performance of the optimal noisy quantizer for the film-grain noise case ($\gamma = 0.5$)	55
5.5	The performance of the optimal noisy vector quantizer for the film-grain noise case ($\gamma = 1/3$)	55
5.6	The effect of preprocessing images corrupted by Poisson noise on JPEG	57
5.7	The effect of preprocessing images corrupted by Poisson noise on EPIC	57

5.8	The effect of preprocessing images corrupted by Poisson noise on JPEG and EPIC	58
5.9	The effect of preprocessing images corrupted by film-grain noise on JPEG and EPIC	58
5.10	The effect of preprocessing images corrupted by film-grain noise on JPEG and EPIC	59
6.1	MMSE postprocessing of lossy compressed noisy images	62
6.2	Postprocessing of Lenna image	67
6.3	Generalized Gaussian fit for quantization noise pdf	68
6.4	PSNR versus shape factor	69
6.5	Postprocessing of JPEG compressed modified phantom	70
6.6	Postprocessing of JPEG compressed phantom	71
6.7	Postprocessing of JPEG compressed modified phantom corrupted by Poisson noise	74
6.8	Postprocessing of JPEG compressed phantom corrupted by Poisson noise	75
6.9	Postprocessing of JPEG compressed Lenna image corrupted by Poisson noise	76
6.10	Postprocessing of JPEG compressed SPECT cardiac projection slice	77
6.11	Postprocessing of JPEG compressed Lenna image corrupted by film grain noise	79
6.12	The effect of σ_q on postprocessing	80
7.1	A $32 \times 64 \times 64$ SPECT cardiac image sequence	84
7.2	ROC curve	88
7.3	Transaxial slices and the corresponding bull's-eye	90
7.4	Content-based compression of SPECT cardiac sequences	92
7.5	The segmentation algorithm	92
7.6	Region-based vector quantization of cardiac sequences	95

7.7	Region-based quantization of cardiac sequences	99
7.8	Comparing region-based SCALAR and VECTOR quantization approaches	102
7.9	The bull's-eyes for the <i>original</i> and <i>filtered</i> sequences when using SCALAR quantization	103
7.10	The bull's-eyes for the <i>original</i> sequence when using the VECTOR coder	103
7.11	The bull's-eyes for the <i>filtered</i> sequence when using the VECTOR coder	107
C.1	REST EXAMPLE	123
C.2	REST EXAMPLE: Comparing Region-based SCALAR and VECTOR quan- tization approaches	124
C.3	REST EXAMPLE: The bull's-eyes for the <i>original</i> and <i>filtered</i> sequences when using SCALAR quantization	124
C.4	REST EXAMPLE: The bull's-eyes for the <i>filtered</i> sequence when using the VECTOR coder	125
C.5	REST EXAMPLE: The bull's-eyes for the <i>original</i> sequence when using the VECTOR coder	125
C.6	STRESS EXAMPLE	133
C.7	STRESS EXAMPLE: Comparing Region-based SCALAR and VECTOR quantization approaches	134
C.8	STRESS EXAMPLE: The bull's-eyes for the <i>original</i> and <i>filtered</i> se- quences when using SCALAR quantization	134
C.9	STRESS EXAMPLE: The bull's-eyes for the <i>filtered</i> sequence when using the VECTOR coder	135
C.10	STRESS EXAMPLE: The bull's-eyes for the <i>original</i> sequence when us- ing the VECTOR coder	135

Summary

Noise degrades the performance of any image compression algorithm. The effect of different noise types, e.g., Gaussian, Poisson, and film grain, on lossy image compression is discussed.

To reduce the effect of noise on compression, the distortion of the coder should be measured with respect to the original noise-free image, not to the noisy image input to the coder. The results of noisy source coding theory are used to design the optimal coder. In the minimum mean square error (MMSE) sense, the noisy coder consists of an MMSE estimator followed by an MMSE coder. The image is modeled as a Markov random field. Mean field theory and a saddle point approximation are used to derive the MMSE estimates. The coders for the Poisson noise and film grain noise cases are derived and their performance is studied. The effect of this preprocessing step is also studied using standard coders, e.g., JPEG. As will be demonstrated, the preprocessing step achieves higher quality at lower bit rates.

This approach is applied to noisy SPECT cardiac images and a segmented coding algorithm is proposed. The proposed coder is a region-based coder that quantizes the different regions according to their importance for diagnosis. The performance of the proposed coder is studied with and without the preprocessing step.

Finally, a postprocessing approach that combines the restoration of the decompressed image from both the artifacts introduced by the coder and the degradation noise is presented. The postprocessing filters for the noise-free, Poisson noise, and film grain noise cases are derived.

CHAPTER 1

INTRODUCTION

Image compression is concerned with minimizing the number of bits required to represent an image at a given level of fidelity [28]. Any image compression algorithm attempts to answer two fundamental questions: what information should be stored or transmitted, and how should the chosen information be compressed [8]. The answers to both questions are related to the properties of the image and the requirements of the application. Nevertheless, the answer to both questions usually involves eliminating the redundancies in the image.

In image compression, three basic data redundancies can be identified: interpixel redundancy, psychovisual or data-relevance redundancy, and coding redundancy [23]. Interpixel redundancy is directly related to the correlation between neighboring pixels. To reduce interpixel redundancy, the image is transformed to another format or domain such that the energy of the new image is concentrated in a few components and these components are independent of each other. Examples of these decompositions include discrete Fourier transform (DFT), discrete cosine transform (DCT), and multiresolution representations. Psychovisual or data-relevance redundancy is reduced by quantizing the image values according to their visual importance or their importance to the application. Reducing psychovisual or data-relevance redundancy involves loss of information. Thus, it is nonlinear and noninvertible. Coding redundancy is eliminated by minimizing the average number of bits per symbol. This is achieved by giving fewer bits to more frequent symbols and more bits to less frequent

symbols. Huffman or arithmetic coding schemes are usually used for this purpose.

Images are, in many cases, degraded even before they are encoded. For example, emission and transmission tomography images are usually corrupted by data-dependent noise that can be modeled as quantum noise or Poisson noise [41]. Generally, images formed at low light levels are also corrupted by this kind of noise [49]. Another example of noisy images occurs when scanning images recorded on photographic films for storage and transmission. The resulting image is proportional to the film density [61] and is corrupted by data-dependent noise widely known as film-grain noise.

Cosman *et al.* [15] notice in their evaluation of the quality of compressed medical images that slightly vector-quantized images are often superior to the originals because noise is suppressed by a clustering algorithm. The superiority is evident in both visual clarity and diagnostic accuracy. This can also be seen in the reported results of Aberle *et al.* [1], where compressed images gave better diagnostic accuracy than noisy originals in terms of receiver-operating-characteristic (ROC) analysis for 11 out of 15 reported experiments. Lo *et al.* [39] discuss the effect of noise on loss-less medical image compression. They conclude that noise decreases the compression ratio. This is because the noise reduces interpixel correlation. Melnychuk *et al.* [44] discuss the effect of film-grain noise on DPCM coders. They conclude that the entropy of the noisy images is almost twice that of noise-free images. As a remedy, they suggest ideal filtering of the noise; they did not base this suggestion on noisy source coding theory. Moreover, filtering of noise is not the optimal solution for any quantization.

Another example of noisy image coding is coding of single photon computerized emission tomography (SPECT) cardiac image sequences, shown in Figure 1.1. Although, when first looking at these projection sequences, the frames look similar and thus should be efficiently predicted from each other, the presence of noise, which is effectively Poisson, degrades the performance of such predictors and coders [3].

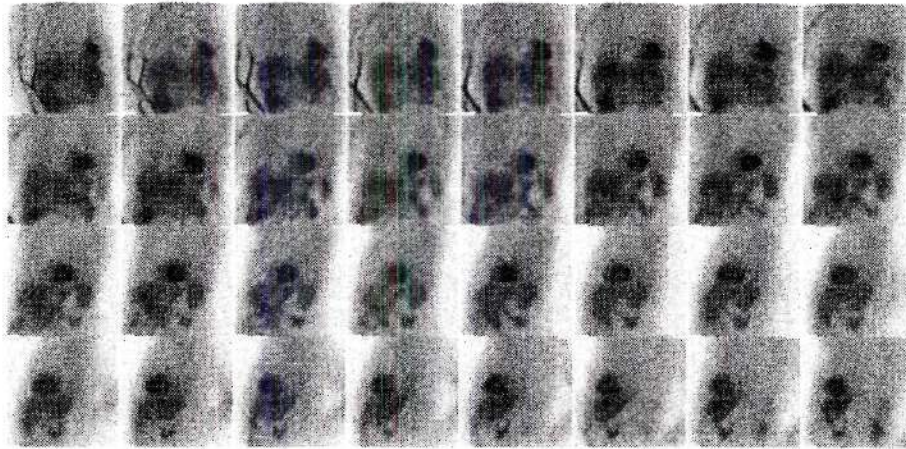


Figure 1.1: A $32 \times 64 \times 64$ SPECT cardiac image sequence.

To illustrate the effect of noise further, let us consider an image sequence that consists of replicas of the same image. Coding this sequence is done trivially by coding the first sequence and sending the number of frames in the sequence. However, if this sequence is corrupted by zero-mean-independent noise and if each frame is predicted from the previous one, the residual error will not be zero and its energy will be twice that of the noise. Moreover, as in most sequence coders, if the residual error is quantized and coded, the coder will be spending bits coding noise, which does not carry any information.

This thesis will discuss the effect of noise on compression. The effect of noise will be explained in terms of its effect on the basic redundancies discussed in this chapter. To understand and reduce the effect of noise on compression, the distortion of the coder is measured with respect to the original noise-free image, not with respect to the noisy image input to the coder. The results of noisy source coding theory are used to design the optimal coder.

Two approaches to enhance the quality of decompressed noisy images are discussed. The first approach is a preprocessing approach where a minimum-mean-square-error (MMSE) restoration algorithm is developed to restore images corrupted by data-dependent Poisson noise and data-dependent film-grain noise. If the MMSE

estimator is followed by an MMSE coder, the overall system will be an MMSE noisy image coder. As will be demonstrated in Chapter 5, preprocessing noisy images will enhance and improve the performance of conventional coders, e.g., JPEG. It is important to note that the objective is to enhance the quality of the image so that it can be coded more efficiently when compressed using any coder.

The second approach is a postprocessing one. The noisy image is compressed using conventional coders, then at the decoder end the decompressed noisy image is restored from degradation noise, quantization noise, and coding artifacts. Mathematically, both approaches are similar, however, the noise model of the preprocessing approach includes only the degradation noise, while the noise model of the postprocessing approach is a more complicated model that includes degradation noise, quantization noise, and coding artifacts. The first approach is optimal when an MMSE coder is used and it gives better results because it prepares the image for compression by smoothing the noise. The second approach can be used in cases where there is not control over the encoder and a better image quality is desired.

As an application, both approaches are applied to cardiac SPECT sequences, which are usually corrupted by data dependent Poisson noise, to improve the performance of compression algorithms. Since it is desired to preserve the diagnostic information in medical images, a content-based coder is developed. This coder preserves the diagnostic information in the image by segmenting the image into different regions, filtering these regions with different image models according to the detail in each region, and coding these regions with different coders. The regions that contain the diagnostic information are compressed without coding distortion, while the other regions are compressed lossily.

1.1 Outline

This thesis is organized as follows: Chapter 2 briefly reviews the basic blocks of a generic image compression system. Chapter 3 discusses the effect of three kinds of noise that frequently corrupt images. Chapter 3 also states the problem of coding noisy images and describes the optimum coder that minimizes the distortion between the original, noise-free image and the coded noisy image. If the distortion measure is the MSE, the optimum coder can be decomposed into an MMSE estimator followed by an MMSE coder.

Chapter 4 develops MMSE restoration algorithms to estimate images corrupted by Poisson noise and film-grain noise by modeling the image as a Markov random field and approximating the estimate using mean field theory and a saddle point approximation. Chapter 5 builds the optimum noisy-image coder and studies its performance. Chapter 5 also studies the effect of the preprocessing step on different conventional coders. Chapter 6 develops MMSE restoration algorithms to restore noise-free and noisy images that are lossy compressed using a block-transform-based coder.

Chapter 7 applies the results of the thesis to compress cardiac image sequences, which are corrupted by Poisson noise. Chapter 7 develops region-based coders that preserve the diagnostic information in cardiac sequences, and evaluates their performance using objective measures and diagnostic accuracy measures. Finally, Chapter 8 concludes the thesis.

CHAPTER 2

IMAGE COMPRESSION

2.1 Introduction

The goal of image compression algorithms is to achieve the best fidelity for a given storage bit rate [15]. This is achieved by eliminating the redundancies in the image. In image compression, three basic data redundancies can be identified: interpixel redundancy, psychovisual or data-relevance redundancy, and coding redundancy [23].

To eliminate these redundancies, an image compression system usually consists of an image decomposition block, a quantizer, and a coder, as shown in Figure 2.1. The image decomposition block transforms the image to another format or domain. The energy of the transformed image is concentrated in a few components and these components are independent of each other, thus eliminating the interpixel redundancy. The quantizer reduces the psychovisual or data-relevance redundancy by throwing away information that is not relevant psychovisually or to the application. This step is nonlinear and noninvertible. The entropy coder minimizes the average number of bits per symbol by giving fewer bits to more frequent symbols and more bits to less frequent symbols.

Image compression can be either lossless (reversible) or lossy (irreversible). An image compression algorithm is lossless if the image can be exactly reconstructed from the compressed information, otherwise it is lossy. Most lossless compression algorithms do not include the quantization step.

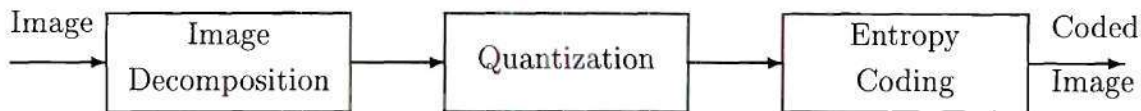


Figure 2.1: Image compression system.

The performance of lossy compression algorithms is usually described by the distortion introduced at a given bit rate. The optimal performance of ideal digital communication systems [25] is studied and described by source coding or rate-distortion theory.

This chapter is organized as follows: Section 2.2 briefly discusses source coding theory. Sections 2.3, 2.4, and 2.5 describe the blocks of the generic image compression system shown in Figure 2.1.

2.2 Source Coding Theory

Source coding theory, or rate-distortion theory, addresses the problem of determining the lowest rate at which information about the source is conveyed to the user in order to achieve a prescribed fidelity [8]. In the communications system depicted in Figure 2.2, let the information source produce a random signal that can be collected in the form of a vector $\mathbf{f} \in \mathcal{F}$. The encoder, $g(\cdot)$, maps \mathbf{f} into \mathbf{u} for transmission over the channel. Afterwards, the received signal \mathbf{v} is mapped by the decoder, $h(\cdot)$, to \mathbf{y} , which is delivered to the information receiver. Let $d(\mathbf{f}, \mathbf{y})$ be a distortion measure defined to tell how well \mathbf{y} approximates \mathbf{f} . The encoder, $g(\cdot)$, and the decoder, $h(\cdot)$, seek to minimize the distortion measure, $d(\mathbf{f}, \mathbf{y})$, for a given bit rate. The average distortion associated with the codec (coder-decoder pair), Q , is

$$d(Q) = \sum_{\mathbf{f}, \mathbf{y}} p(\mathbf{f}, \mathbf{y}) d(\mathbf{f}, \mathbf{y}). \quad (2.1)$$

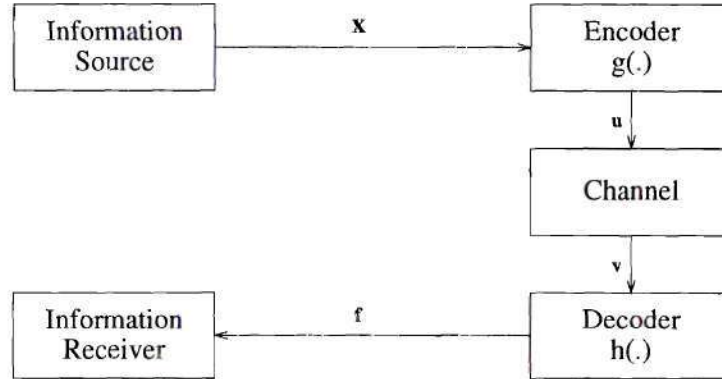


Figure 2.2: The block diagram of a communications system.

A codec is D -admissible if $d(Q) \leq D$. The rate-distortion function $R(D)$ is the infimum of the rates such that the codec is D -admissible [8]. The rate-distortion function is non-negative, continuous, convex, and strictly decreasing [8]. Moreover, $0 \leq R(D) \leq \log M$, where M is the size of the alphabet.

The source Coding Theorem states that for a given average distortion D , there exists a coder and a decoder pair, g and h , such that $R(D)$ bits per symbol are sufficient to reconstruct the source with an average distortion that is arbitrarily close to D [8]. The converse to the source coding theorem states that no D -admissible code has a rate less than $R(D)$ [8].

2.3 Entropy coding

The marginal entropy of a source is defined as the average information generated by it:

$$H = \sum_i -p(a_i) \log p(a_i), \quad (2.2)$$

where $\{a_i\}$ is the set of symbols generated by the source and $\{p(a_i)\}$ are the probabilities of their occurrence. The entropy of a source of independent symbols is the minimum number of information units required to encode the source [28].

Huffman coding [26] and arithmetic coding [34, 53] are one of the most common

coding methods used. Huffman coding achieves optimum coding if the probabilities of the source symbols are in the form 2^{-j} , where j is an integer. Arithmetic coding can theoretically achieve the optimal code, without any restrictions on the probabilities. When the source outputs (pixel values) are dependent, run-length coding [28] and Lempel-Ziv coding [74] sometimes replace Huffman and arithmetic coding.

2.4 Quantization

Quantization, in a general sense, is the mapping of vectors (or scalars) of an information source into a finite collection of codewords for storage or transmission [24]. This involves two processes: encoding and decoding. The encoder blocks the source $\{x_i\}$ into vectors of length n , and maps each vector $X^n \in \mathcal{X}^n$ into a codeword c taken from a finite set of codewords \mathcal{C} . The decoder maps the codeword c into a reproduction vector $Y^n \in \mathcal{Y}^n$ where \mathcal{Y} is a reproduction alphabet. If $n = 1$, it is called *scalar quantization*. Otherwise, it is called *vector quantization*.

The problem of optimum mean squared scalar quantization for a given reproduction alphabet size was independently solved by Lloyd [38] and Max [43]. They found that if x is a real scalar random variable with continuous probability density function $p_x(x)$, then the quantization thresholds are

$$t_k = \frac{r_k + r_{k-1}}{2}, \quad (2.3)$$

which is the geometric mean of the interval $(r_{k-1}, r_k]$, where

$$r_k = \frac{\int_{t_k}^{t_{k+1}} x p_x(x) dx}{\int_{t_k}^{t_{k+1}} p_x(x) dx} \quad (2.4)$$

are the reconstruction levels. Iterative numerical methods are required to solve for the reconstruction and the quantization levels. Memory can be incorporated into scalar quantization by predicting the input from the previous samples and quantizing the residual error, e.g., linear predictive coding.

A fundamental result of Shannon's rate distortion theory is that better performance can be achieved by coding vectors instead of scalars, even if the source is memoryless [24, 45]. Linde, Buzo, and Gray [37] generalized the Lloyd-Max algorithm to vector quantization. If the vector quantizer (VQ) quantizes each vector independent of all other vectors, it is called a memoryless VQ. Otherwise, it is a memory VQ. Memoryless and memory VQ design is briefly discussed next.

2.4.1 Memoryless vector quantization

Linde, Buzo, and Gray found that given a training sequence of the information source and a fixed number of reproduction vectors, there are two necessary conditions to achieve optimal vector quantization [37]. First, given the goal to minimize a distortion measure and given a decoder, no memoryless encoder can do better than to select the codeword that will yield the minimum distortion at the output [24]. Thus, a full search is usually conducted on all codewords to find the one that has minimum distortion. Second, given that goal and the encoder, the best codeword is the centroid based on a distance measure (For the MSE it is the geometric mean) of the source vectors encoded as that codeword. Following that, they developed a procedure for designing the VQ known as the generalized Lloyd algorithm. This is also popularly called the LBG algorithm, and is depicted in Figure 2.3.

The design of the encoder is very expensive. To solve this, less expensive algorithms have been developed, e.g., tree-searched vector quantization (TSVQ), pruned tree-structured vector quantization (PSTVQ), multistage vector quantizer, product codes, and others [24].

The length of the codewords designed by these algorithms (except for the PSTVQ) is constant. Thus, if the resulting probability distribution of the quantized vectors is nonuniform, additional compression can be achieved by entropy coding. However, if the VQ is designed with an entropy constraint instead of the fixed number of quantization levels constraint, then the performance of the VQ can be

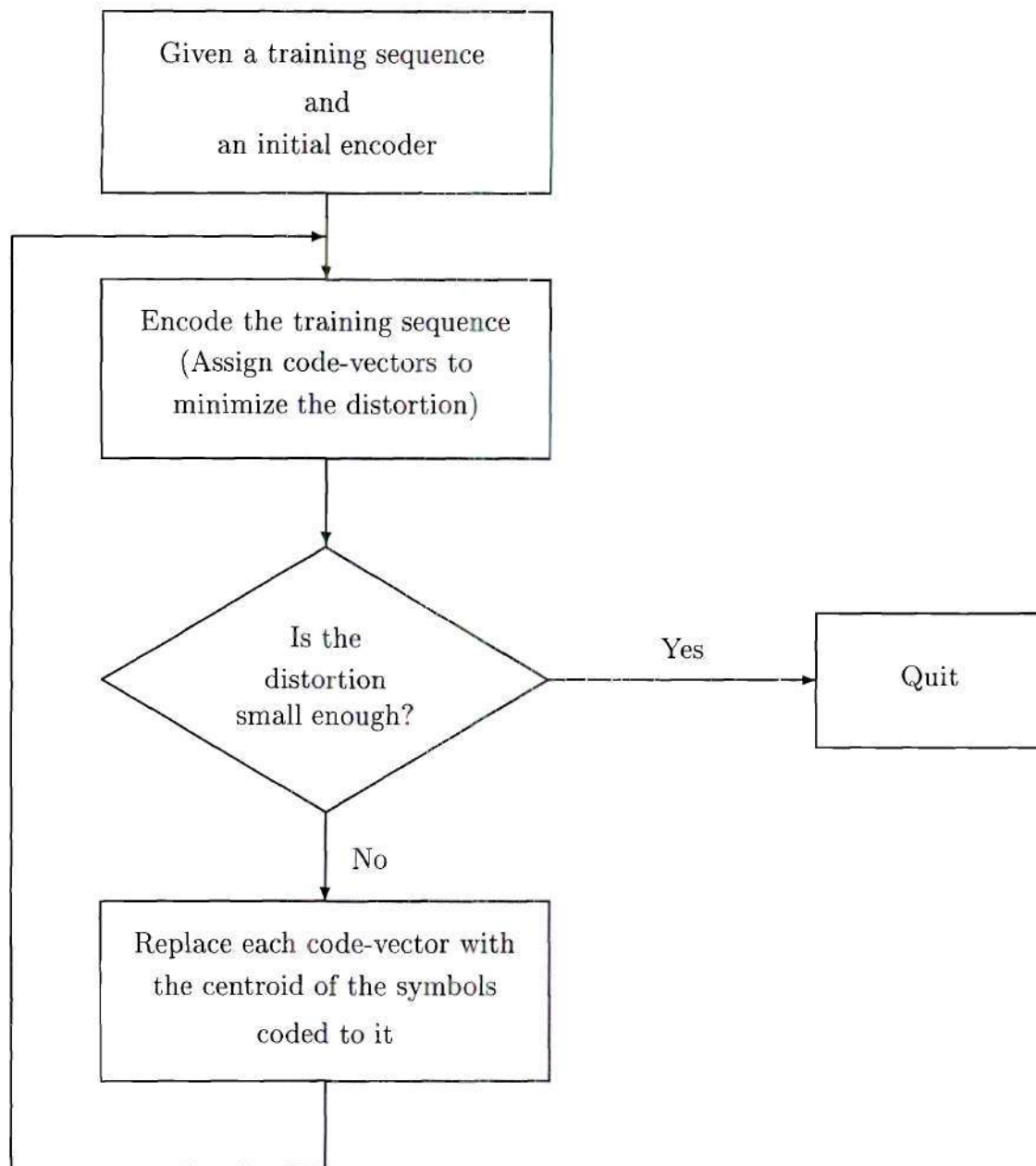


Figure 2.3: LBG vector quantizer design algorithm.

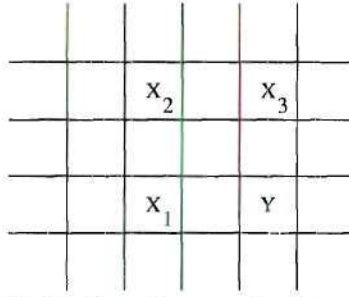


Figure 2.4: Linear prediction with 2×2 vector quantization. Pixel Y is linearly predicted from pixels X_1 , X_2 , and X_3 . The residual error defined as the difference between Y and its predicted value is then vector quantized.

expected to improve. Different algorithms have been developed with this constraint, e.g., entropy-pruned tree-structured VQ (EPTSVQ) [40] and entropy constrained VQ (ECVQ)[12].

2.4.2 Memory vector quantization

The performance of entropy-constrained VQ or other VQs followed by an entropy coder depends on the probability distribution of the quantized vectors. If their probability distribution is highly skewed, low bit rates can be achieved. Image decomposition algorithms decorrelate the input vectors and concentrate the energy of the image in a few components. Memory can be incorporated into vector quantization by predicting the input from the previous samples and quantizing the residual error, e.g., linear prediction and unitary transforms. Different compression algorithms that combined vector quantization with transform coding [45], subband coding [45, 56], and pyramid coding [65] have been developed. Other algorithms use linear prediction, which is discussed in Section 2.5.1, of vectors from neighboring previous vectors [12, 24, 40, 52], as shown in Figure 2.4. Afterwards, the residual error is vector quantized.

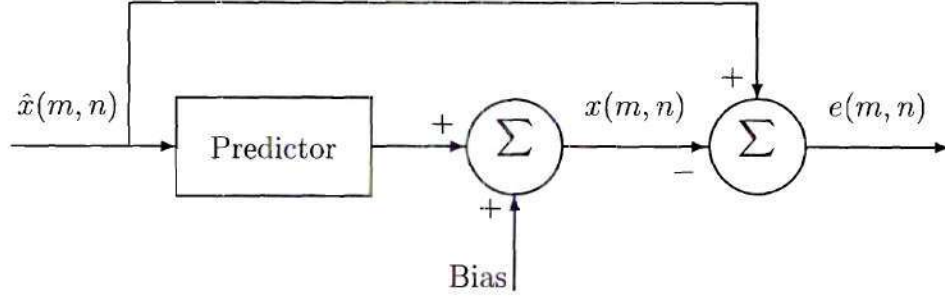


Figure 2.5: Linear prediction and residual error calculation in a linear predictive coder.

2.5 Image Decomposition

The function of the image decomposition block shown in Figure 2.1 is to transform the image to another format or domain such that the energy of the new image is concentrated in a few components and these components are uncorrelated, thus eliminating the interpixel redundancy. This section discusses linear prediction, unitary transforms, and subband decomposition.

2.5.1 Linear predictive coding (LPC)

Linear predictive coding is based on eliminating the interpixel redundancy by predicting the pixel from its neighbors and then encoding the residual error, as shown in Figure 2.5. This corresponds to modeling the image as a two-dimensional autoregressive process driven by zero-mean white noise [42]:

$$x(m, n) = \sum_{\substack{k \\ (k, l) \in \Psi}} \sum_l a(k, l) x(m - k, n - l) + a_0 + u(m, n), \quad (2.5)$$

where $x(m, n)$ is the pixel value, $a(k, l)$ are the prediction coefficients, a_0 is the locally constant bias added because of the assumed zero-mean model, $u(m, n)$ is zero-mean white noise, and Ψ is the region of support of the predictor (the prediction mask).

The bias and the prediction coefficients are chosen to minimize the mean-

squared prediction error (MSE) defined as

$$E = \sum_m \sum_n e^2(m, n), \quad (2.6)$$

where

$$e(m, n) = x(m, n) - \sum_{\substack{k \\ (k, l) \in \Psi}} \sum_l a(k, l) x(m - k, n - l) - a_0 \quad (2.7)$$

is the residual error.

Differential pulse code modulation (DPCM) is a special case of LPC where the region of support of the predictor is simply the previous pixel and the prediction coefficient is unity [28]. Another special case assumes a separable first-order Markov model for which the autocorrelation function is $\phi(k, l; i, j) = \sigma^2 \rho^{|k-l|+|i-j|}$, where $0 \leq \rho \leq 1$ is the correlation coefficient and σ^2 is the image variance [54]. For this model, the optimal predictor is

$$x(m, n) = \rho x(m, n - 1) + \rho x(m - 1, n) - \rho^2 x(m - 1, n - 1). \quad (2.8)$$

2.5.2 Transform coding

Transform-based coding reduces the interpixel redundancy by performing a unitary transformation on the image. There are two properties desirable in a unitary transform for image compression: the energy should be packed into a few transform coefficients, and the coefficients should be as uncorrelated as possible. The optimum transform under these two constraints is the Karhunen-Loève transform (KLT) where the eigenvectors of the covariance matrix of the image are the vectors of the transform [28]. Although the KLT is optimal under these two constraints, it is data-dependent and it is very expensive to compute.

Other transforms have also been used for compression, e.g., the discrete Fourier transform (DFT), the discrete cosine transform (DCT), the discrete sine transform (DST), and others [28]. In the one-dimensional case and for a first-order Markov sequence with a high correlation coefficient, the DCT performs very close to the KLT.

It also performs very close to the KLT in the two-dimensional case for a separable Markov sequence. Often, the image is modeled as a separable Markov process with high correlation between the pixels, e.g., the DPCM model [28]. Thus, it is reasonable to use the DCT for image compression. This is also confirmed by comparing the performance of different transforms [23, 28].

Calculating the transform of an image involves an expensive matrix multiplication operation. To reduce the number of computations in order to make the DCT practical for image compression, small blocks of the image are transformed and then the transformation coefficients are quantized. Quantizing the transform coefficients spreads the quantization error over the whole block. So, it affects groups of pixels, not just a single pixel. However, block-transform-based compression usually introduces blocking distortion at low bit rates.

The DCT proved to be very effective for natural images. Thus, JPEG opted to use it in the joint ISO and CCITT still image lossy compression standard [64].

2.5.3 Subband coding

The basic idea of subband coding is to split the frequency spectrum of the image into (disjoint) subbands and then to code the subband signals. This can be efficient when the image spectrum is not flat. The image can be decomposed into subbands either by means of a filter bank structure or a pyramid structure. Each of these structures can be described in terms of the other. However, since they appear separately in the literature, they will be discussed separately.

Filter bank structure

The image is decomposed into subbands and reconstructed from the subbands using a filter bank (FB) system [70]. The analysis section decomposes the signal into subbands using low-pass and high-pass filters and downsamplers. The synthesis part reconstructs the signal from the subbands using low-pass and high-pass filters and

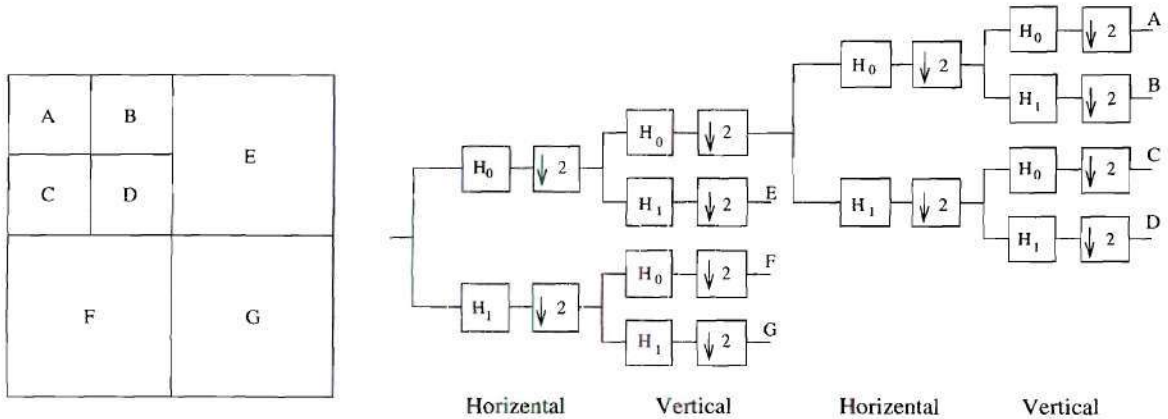


Figure 2.6: A frequency tiling and its corresponding separable filter bank structure. H_0 and H_1 are 1-D low-pass and high-pass filters, respectively. The stages labeled horizontal act on the rows of the images; those labeled vertical operate on the columns.

upsamplers. If the signal can be reconstructed with no error, the FB is a “perfect reconstruction” FB. Quadrature mirror filter banks (QMFB) are very popular. For 1-D signals, the QMFB decomposes the signal into low-pass and high-pass subbands that are of equal frequency bandwidth. The resulting two signals are then downsampled by two so that the total sampling rate is equal to the original one (critical downsampling) [62]. Multidimensional FBs decompose the signal into bands usually using separable filters. For perfect reconstruction and critical downsampling, the FB passbands should form a tiling in the frequency domain. Figure 2.6 shows a tiling and its corresponding FB.

Pyramid structure

The image is decomposed into a sequence of reduced size images by predicting every other pixel from some neighborhood [50]. That is, in terms of filter banks, the image is filtered using an FIR filter and then downsampled by two in each direction. This is continued until a single point remains as shown in Figure 2.7.

Different pyramids were developed by changing the predictor and the way these images are stored. For example, in the Gaussian pyramid, pixels are predicted using a Gaussian-like filter and then the resulting images are stored [9]. The Laplacian

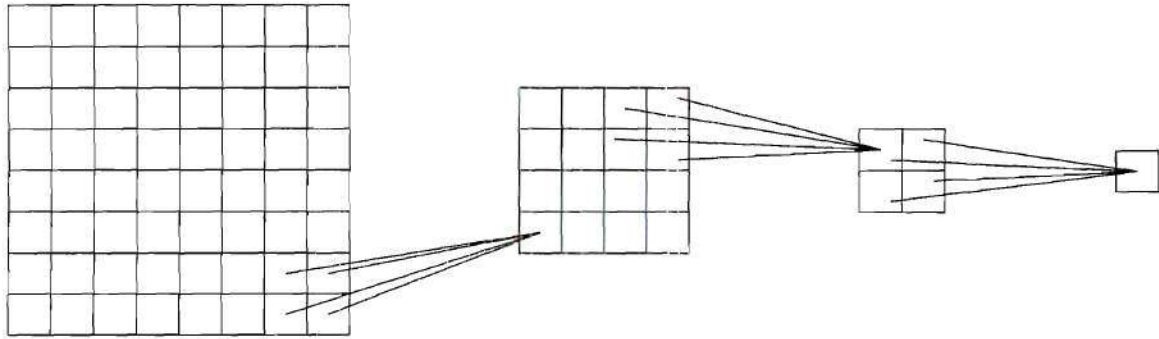


Figure 2.7: A Pyramid structure.

pyramid stores the difference of two consecutive images of the Gaussian pyramid [9]. Other examples include mean, reduced sum, difference, reduced difference, and (Haar) S-transform pyramids [66].

CHAPTER 3

NOISE AND ITS EFFECT ON COMPRESSION

3.1 Introduction

The effect on compression of three noise types, which are Gaussian noise, Poisson noise and film-grain noise, is studied. These noise types are chosen because they are common models for different kinds of noise that frequently corrupt images, as will be discussed in the following sections. The effect of noise will be studied using Lloyd-Max scalar and Generalized Lloyd vector quantizers, and the JPEG compression standard. When using scalar quantizers, noise affects the psychovisual and data-relevance redundancy and the coding redundancy in the image, thus affecting the performance of such quantizers. However, when using vector quantizers and transform-based coders such as JPEG, the noise also reduces the interpixel redundancy in the noisy image, thus limiting the capabilities of such coders.

The design of the coder that minimizes the distortion measured between the original noise-free image and the decompressed noisy image is then discussed. The results of noisy source coding states that the MMSE noisy coder is an MMSE estimator followed by an MMSE coder. These results are used in Chapter 5 to design the MMSE coder for the Poisson noise and the film-grain noise cases. Moreover, Chapter 5 will demonstrate the effectiveness of this preprocessing step in improving the performance of conventional coders.



Figure 3.1: The 256×256 (a) original LENA image and (b) LENA image corrupted by iid Gaussian noise with $\text{PSNR} = 28.157$ dB.

This Chapter is organized as follows: Sections 3.2, 3.3, and 3.4 discuss the effect on compression of Gaussian noise, Poisson noise, and film-grain noise, respectively. Section 3.5 formulates the problem in terms of noisy source coding and reviews some of the noisy source coding results.

3.2 Gaussian Noise

The Gaussian noise model is the most commonly used noise model. Quantization noise is usually modeled as zero-mean white, Gaussian noise that is independent of the image [47, 57]. Other examples of Gaussian noise models include measurement noise and thermal noise. Due to the analytical tractability of Gaussian noise models, other noise models are often modeled as data-dependent Gaussian noise. For example, Poisson noise is often modeled as data-dependent Gaussian noise with variance and mean equal to the original signal [6]. Film grain noise is also modeled as data-dependent Gaussian noise [33].

For images corrupted by independent, identically distributed (iid) Gaussian noise, the conditional probability of the noisy image, \mathbf{x} , given the original image, \mathbf{f} is

given by

$$p(\mathbf{x}|\mathbf{f}) = \prod_{i,j} \frac{\exp \frac{-\{x(i,j)-f(i,j)\}^2}{2\sigma_n^2}}{\sqrt{2\pi}\sigma_n}, \quad (3.1)$$

where $f(i, j)$ and $x(i, j)$ are the values of \mathbf{f} and \mathbf{x} at pixel (i, j) , respectively, and σ_n^2 is the variance of the noise.

To illustrate the effect of noise on compression, the 256×256 Lenna image, shown in Figure 3.1(a), is corrupted by data-independent, additive, Gaussian noise with PSNR = 28.157 dB and is shown in Figure 3.1(b). The curves of peak-signal-to-noise-ratio (PSNR) versus bit rate for the original and the noisy images when quantized using Lloyd-Max scalar and Generalized Lloyd vector quantizers, and when compressed using the lossy JPEG standard, are studied and shown in Figures 3.2(a), (b), and (c), respectively.

The PSNR curves of the noisy images when the distortion is measured with respect to the original image are particularly interesting. At high bit rates, the quality of the images as measured by the PSNR does not significantly improve because the error is dominated by the additive noise. Thus, we could code the image using a lower bit rate while maintaining the same PSNR quality. More interestingly, with the JPEG coder at medium and low bit rates the quality of the image degrades as the bit rate is increased. This is because the coder removes more noise than it introduces as coding artifacts. We also notice that at low bit rates, the lossy compressed image is closer to the original image than to the noisy input to the coder. This has been used as the basis of a nonlinear filtering method by Natarajan [46], who called the resulting filters Occam filters, which filter the noise by compressing the noisy image.

When the PSNR is computed with respect to the input of the coder not the noise-free image, the performance of the scalar quantizer is not affected. The performance of the vector quantizer and JPEG, however, degrades. This is because scalar quantization does not exploit the spatial redundancy in the image and because the dynamic range of the gray levels of the image and the histogram of the image do not change significantly after adding the noise. The performance of vector quantization

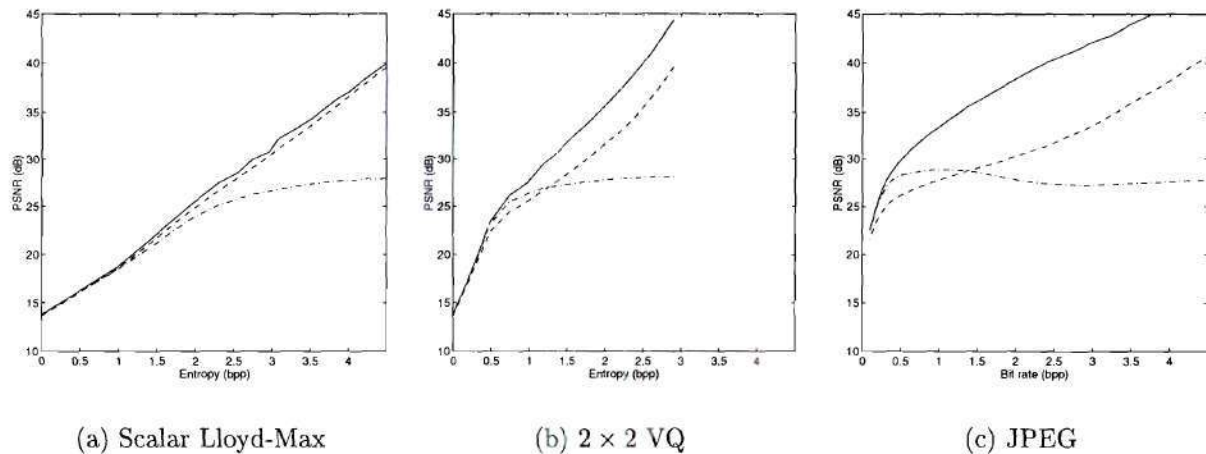


Figure 3.2: The effect of Gaussian noise, where the PSNR of the noisy image is 28.157 dB, on Lloyd-Max quantizers and JPEG: (a) PSNR versus entropy when using Lloyd-Max scalar quantization, (b) PSNR versus entropy when using 2×2 Generalized Lloyd vector quantization, and (c) PSNR versus bit rate when using JPEG. (The solid lines give the rate-distortion tradeoff for the original image, the dashed lines give the rate-distortion tradeoff for the noisy image when the distortion is computed with respect to the noisy image, and the dash-dotted lines give the rate-distortion tradeoff for the noisy image when the distortion is computed with respect to the original image.)

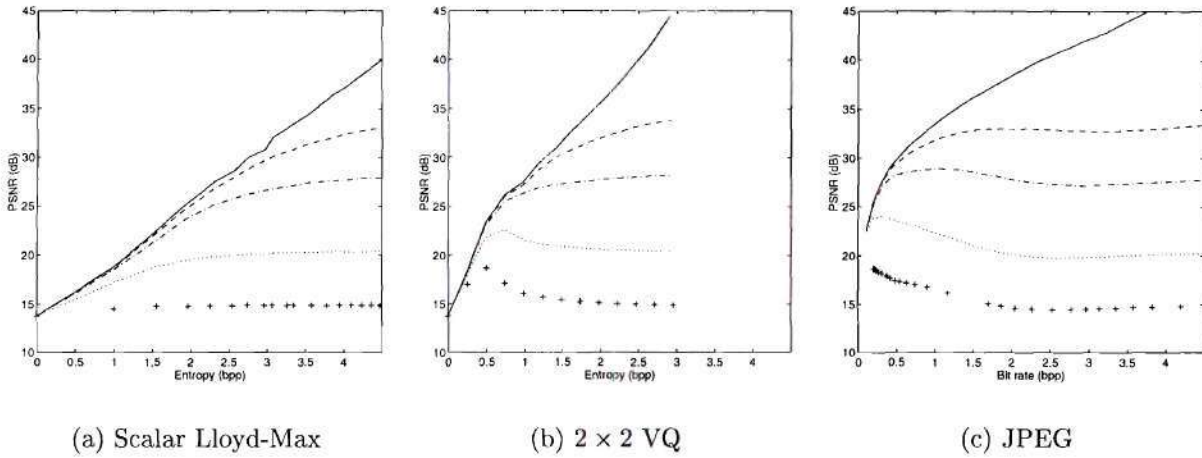


Figure 3.3: The effect of Gaussian noise power level when using (a) Lloyd-Max scalar quantizers, (b) Generalized Lloyd vector quantizers, and (c) JPEG. (The solid lines are for the noise-free case. The PSNRs are: 34.099 dB for the dashed lines, 28.157 dB for the dash-dotted lines, 20.421 dB for the dotted lines, and 14.860 dB for the “+” lines.)

and JPEG degrades because they try to exploit the spatial correlation in the image, which is reduced by the presence of the noise.

Figure 3.3 plots the PSNR versus the bit rate as the noise power is varied. Changing the noise level does not change the general characteristics of the curves; they all display an almost constant PSNR for high and medium bit rates.

3.3 Poisson Noise

Emission and transmission tomography images are usually corrupted by data-dependent noise that can be modeled as Poisson noise [7]. Generally, images formed at low light levels are corrupted by this kind of noise [49]. Examples of such images include computer tomography (CT) scans, SPECT scans, positron emission tomography (PET) scans, and nondestructive testing images.

The conditional probability of the noisy image, \mathbf{x} , given the original image, \mathbf{f}



Figure 3.4: The 256×256 (a) original Lenna image and (b) Lenna image corrupted by data-dependent Poisson noise with PSNR = 28.159 dB.

is given by

$$p(\mathbf{x}|\mathbf{f}) = \prod_{i,j} \frac{e^{-\lambda f(i,j)} [\lambda f(i,j)]^{x(i,j)}}{x(i,j)!}, \quad (3.2)$$

where $f(i,j)$ and $x(i,j)$ are the values of \mathbf{f} and \mathbf{x} at pixel (i,j) , respectively, and λ is the proportionality factor relating the gray level value to the number of counts [32].

To illustrate the effect of Poisson noise on compression, the 256×256 Lenna image, shown in Figure 3.4(a), is corrupted by data-dependent Poisson noise with PSNR = 28.159 dB and shown in Figure 3.4(b). The curves of PSNR versus bit rate for the original and the noisy images when quantized using Lloyd-Max scalar and Generalized Lloyd vector quantizers, and when compressed using the lossy JPEG standard, are shown in Figures 3.5(a), (b), and (c), respectively. Although Poisson noise is data-dependent noise, its effect is similar to that of Gaussian noise.

3.4 Film Grain Noise

If an image recorded on photographic film is scanned for electronic storage or transmission, the scanned image will be proportional to the film density. The scanned

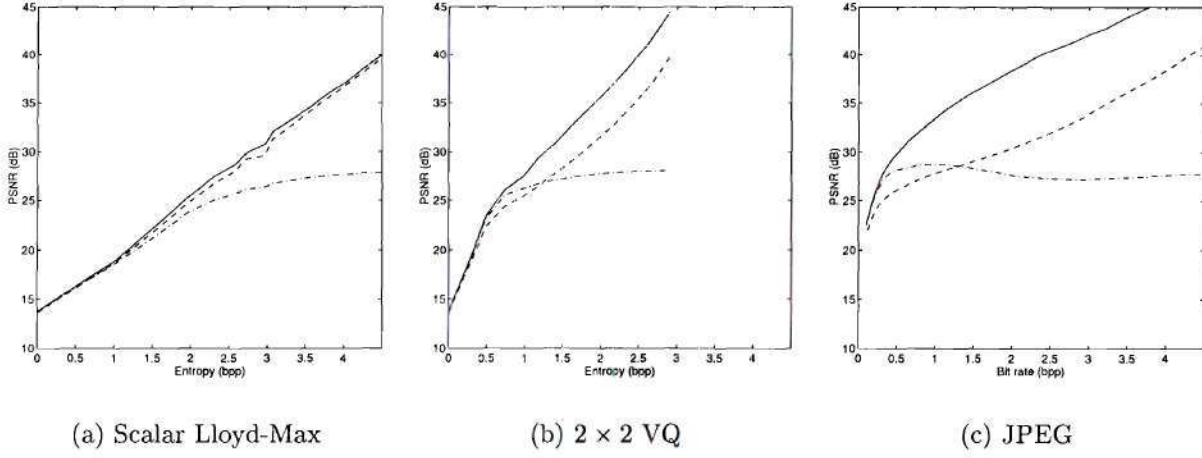


Figure 3.5: The effect of Poisson noise, where the PSNR of the noisy image is 28.159 dB, on Lloyd-Max quantizers and JPEG: (a) PSNR versus entropy when using Lloyd-Max scalar quantization, (b) PSNR versus entropy when using 2×2 Generalized Lloyd vector quantization, and (c) PSNR versus bit rate when using JPEG. (The line types are the same as those of Figure 3.2.)

image, $x(i, j)$, which is corrupted by film-grain noise, can be modeled by [33]

$$x(i, j) = f(i, j) + \alpha[f(i, j)]^\gamma n(i, j), \quad (3.3)$$

where $f(i, j)$ is the density of the original image, α is a proportionality constant, γ is a constant with a value between $1/3$ and $1/2$, and $n(i, j)$ is white Gaussian noise with zero-mean and unit-variance. Thus, the conditional probability of the noisy image, \mathbf{x} , given the original image, \mathbf{f} , is

$$p(\mathbf{x}|\mathbf{f}) = \prod_{i,j} \frac{\exp \frac{-\{x(i,j)-f(i,j)\}^2}{2\alpha^2[f(i,j)]^{2\gamma}}}{\sqrt{2\pi}\alpha[f(i,j)]^\gamma}, \quad (3.4)$$

where $f(i, j)$ and $x(i, j)$ are the values of \mathbf{f} and \mathbf{x} at pixel (i, j) , respectively.

In order to study the effect of noise on compression, the 256×256 LENA image, shown in Figure 3.6(a), is corrupted by film-grain noise with $\gamma = 0.5$ and a peak-signal-to-noise-ratio (PSNR) of 28.174 dB, as shown in Figure 3.6(b). Figure 3.7 shows curves of PSNR versus bit rate for the noise-free and the noisy images when using Lloyd-Max scalar and Generalized Lloyd vector quantization and JPEG,



Figure 3.6: The 256×256 (a) original LENA image and (b) LENA image corrupted by data-dependent film-grain noise with $\text{PSNR} = 28.174$ dB.

respectively. The PSNR versus bit rate curves for the noisy images when the PSNR is computed with respect to the original image, i.e., when the distortion of the coder is computed with respect to the original (free of noise) image, are again very interesting. At high bit rates, the quality of the images when measured in terms of the PSNR does not change. More interestingly, it sometimes even degrades when using JPEG. Figure 3.8 shows the effect of increasing the noise level on compression. The same trend exists at all noise levels. Figure 3.9 shows the effect of γ on compression. The same trend also exists for all different values of γ .

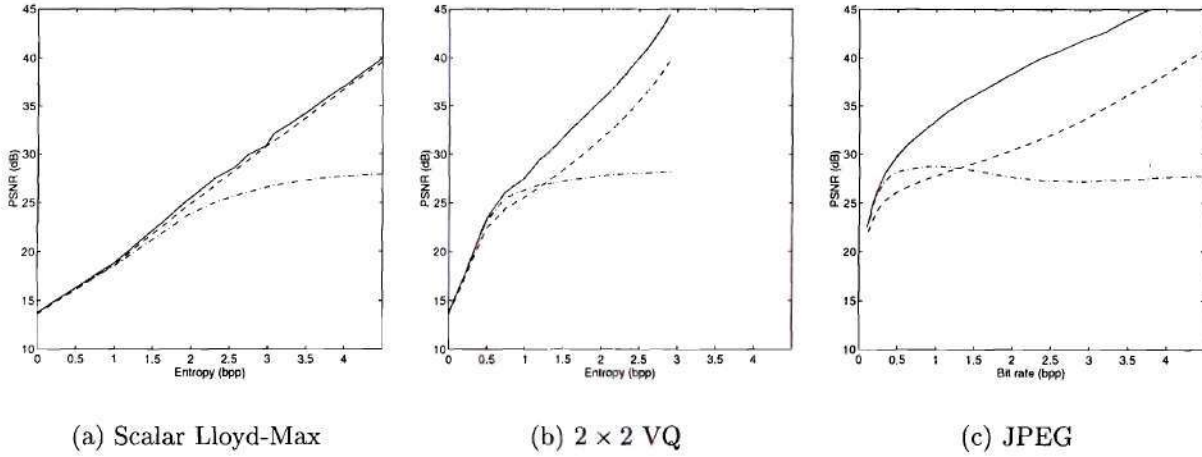


Figure 3.7: The effect of film-grain noise, where the PSNR of the noisy image is 28.174 dB and $\gamma = 0.5$, on Lloyd-Max quantizers and JPEG: (a) PSNR versus entropy when using Lloyd-Max scalar quantization, (b) PSNR versus entropy when using 2×2 Generalized Lloyd vector quantization, and (c) PSNR versus bit rate when using JPEG. (The line types are the same as those of Figure 3.2.)

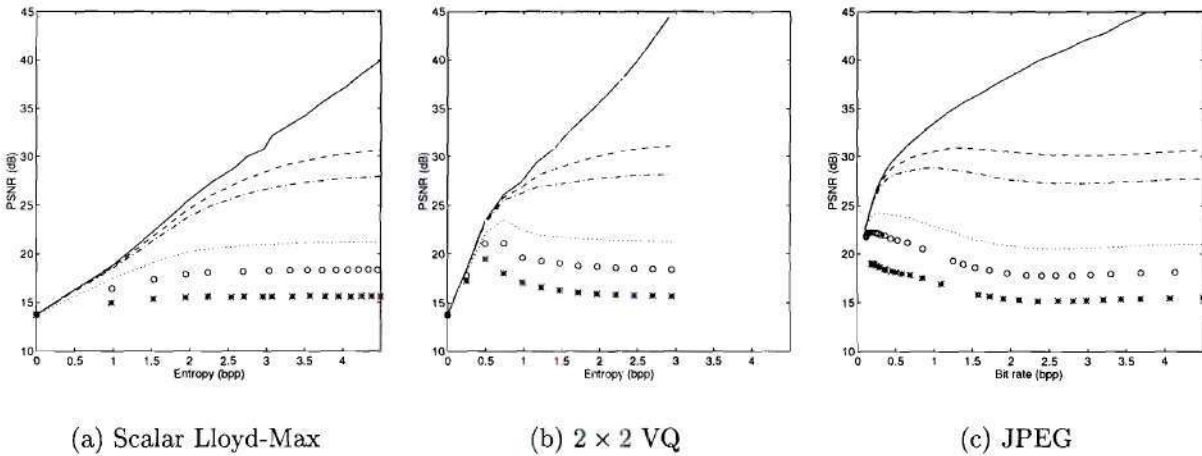


Figure 3.8: PSNR versus bit rate for the data-dependent film-grain noise when using (a) Lloyd-Max scalar quantizers, (b) Generalized Lloyd vector quantizers, and (b) JPEG. (The solid lines are for the noise-free case. The PSNRs are: 31.169 dB for the dashed lines, 28.174 dB for the dash-dotted lines, 21.244 dB for the dotted lines, 18.354 dB for the "o" lines, and 15.599 dB for the "*" lines.)

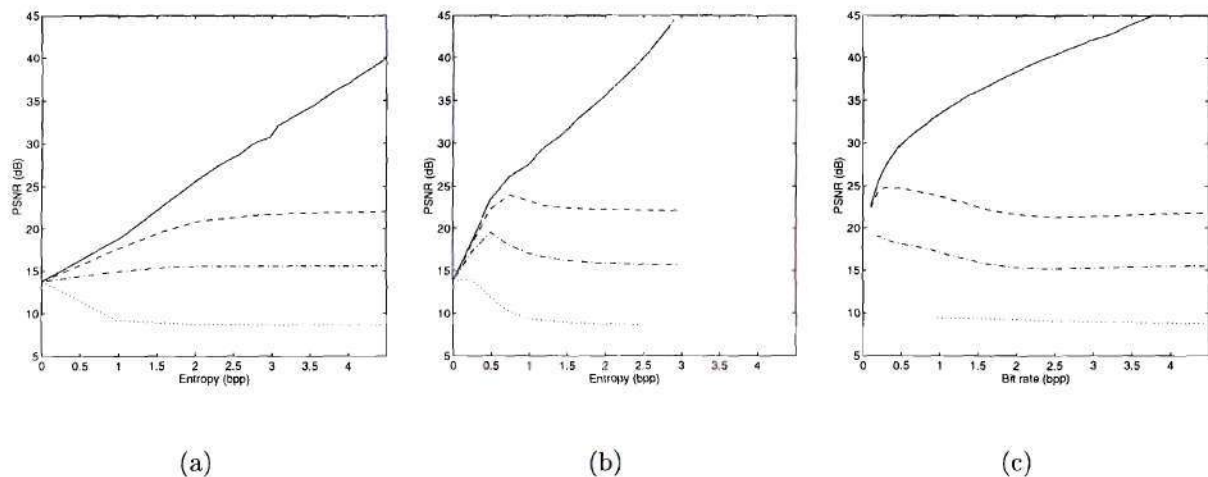


Figure 3.9: The effect of γ , where $\alpha = \sqrt{20}$, when using: (a) Lloyd-Max scalar quantization, (b) Generalized Lloyd vector quantization, and (c) JPEG (The solid lines are for the noise-free case. $\gamma = 1/3$ for the dashed lines, 0.5 for the dash-dotted lines, and 0.8 for the dotted lines.)

3.5 Coding Noisy Images

This chapter studied the effect of noise on compression of images. As demonstrated, noise severely affects the performance of scalar quantizers when the distortion is measured with respect to the original noise-free image. Moreover, if the coder exploits the inherent spatial redundancy in noise free images, e.g., VQ and JPEG, the distortion with respect to the input of the coder will also degrade when the images are noisy. Moreover, at medium and low bit rates the quality of the image sometimes improves if it is coded at a lower bit rate.

At very low bit rates, the quality of the compressed noisy image is close to that of the compressed noise-free image; however, both are of bad quality. Usually, such images are postprocessed in order to enhance their quality and remove the coding artifacts. Joint removal of both degradation noise and coding artifact will improve these images even more. Chapter 6 discusses postprocessing of compressed images.

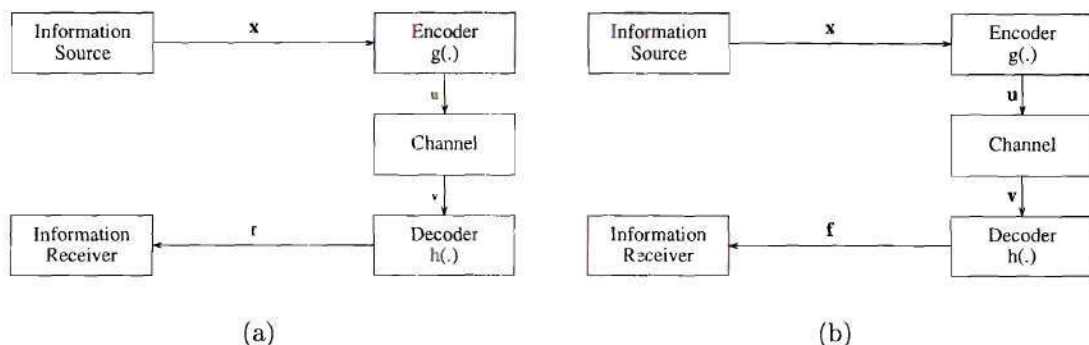


Figure 3.10: The block diagram of a communication system assuming: (a) no degradation, and (b) degradation.

In order to code noisy images more efficiently, the codec should minimize the distortion between the original noise-free image and the output of the codec. This is different than the classical coding problem where the distortion is measured between the input of the codec and the output of the codec. The remainder of this section will discuss this problem.

The conventional problem of source coding or optimal quantization of a random signal assumes that the signal will not be corrupted before encoding or after decoding as implied by the communication channel given in Figure 3.10(a). Thus, the coder-decoder pair simply tries to minimize the distortion between its input and its output. In contrast, the problem of noisy source coding assumes that the signal may be corrupted before encoding or after decoding, as shown in Figure 3.10(b). Thus, the codec should minimize the distortion between the output of the information source, f , and the input of the information receiver, y . This is because minimizing the distortion of the input of the codec involves preserving the noise introduced by the degradation process, which degrades the overall performance of the codec.

The problem of lossy compression of noisy images is a special case of the problem of noisy source coding, as described in Figure 3.10(b) [18, 69], in that there is no receiver degradation block. Thus, the transformation $R(y, w)$ is the identity operator,

i.e., $\mathbf{y} = \mathbf{w}$.

Let the information source produce a random signal that can be collected in the form of a vector $\mathbf{f} \in \mathcal{F}$. This signal is corrupted by a degradation process that can be characterized by $Q(\mathbf{x}, \mathbf{f})$, which is a mapping that can be viewed probabilistically as the conditional probability of the corrupted signal, $\mathbf{x} \in \mathcal{X}$, given the original one, $\mathbf{f} \in \mathcal{F}$, i.e., $p(\mathbf{x}|\mathbf{f})$. The encoder, $g(\cdot)$, maps \mathbf{x} into \mathbf{u} for transmission over the channel. Afterwards, the received signal \mathbf{v} is mapped by the decoder, $h(\cdot)$, to \mathbf{w} , which equals to \mathbf{y} because $R(\mathbf{y}, \mathbf{w})$ is the identity operator. Afterwards \mathbf{y} is delivered to the information receiver. Let $d(\mathbf{f}, \mathbf{y})$ be a distortion measure defined to tell how well \mathbf{y} approximates \mathbf{f} . The encoder, $g(\cdot)$, and the decoder, $h(\cdot)$, seek to minimize the distortion measure, $d(\mathbf{f}, \mathbf{y})$, for a given bit rate.

Dobrushin and Tsybakov have shown that this problem is equivalent to the classical source coding problem. However, the distortion measure is modified to depend only on the input of the encoder and the output of the decoder [18]. Berger [8] noticed that the modified distortion measure, $\hat{d}(\mathbf{x}, \mathbf{y})$, which is the conditional average of the original distortion measure

$$\hat{d}(\mathbf{x}, \mathbf{y}) = \frac{1}{p(\mathbf{x})} \sum_{\mathbf{f}} p(\mathbf{f}) p(\mathbf{x}|\mathbf{f}) d(\mathbf{f}, \mathbf{y}) = \sum_{\mathbf{f}} p(\mathbf{f}|\mathbf{x}) d(\mathbf{f}, \mathbf{y}), \quad (3.5)$$

where

$$p(\mathbf{x}) = \sum_{\mathbf{f}} p(\mathbf{f}) p(\mathbf{x}|\mathbf{f}). \quad (3.6)$$

When the distortion is measured in terms of the mean-squared-error (MSE), $d(\mathbf{f}, \mathbf{y}) = E\{\|\mathbf{f} - \mathbf{y}\|^2\}$, the problem can be decomposed into an optimum estimator followed by an optimum coder [69]. Thus, the minimum distortion d^* is given by

$$\begin{aligned} d^* &= \min_{g,h} E\{\|\mathbf{f} - \mathbf{y}\|^2\} \\ &= E\{\|E\{\mathbf{f}|\mathbf{x}\} - \mathbf{x}\|^2\} + \min_{g,h} E\{\|E\{\mathbf{f}|\mathbf{x}\} - \mathbf{y}\|^2\}. \end{aligned} \quad (3.7)$$

This means that the distortion can never be less than $E\{\|E\{\mathbf{f}|\mathbf{x}\} - \mathbf{x}\|^2\}$. Although this result was derived for the MSE, it holds for any Hilbert space.

Fischer *et al.* extended the previous results to the alphabet-constrained formulation [19]. At time step n , the encoder selects a transmission vector $\mathbf{u}(n)$ based on its information set $\Psi_e(n)$ and the decoder produces an output $\mathbf{y}(n)$ based on its information set $\Psi_d(n)$ where

$$\begin{aligned}\Psi_e(n) = & \{\mathbf{x}(n+k), \mathbf{x}(n+k-1), \dots, \mathbf{x}(n), \\ & \mathbf{x}(n-1), \dots; \\ & \mathbf{u}(n-1), \mathbf{u}(n-2), \dots\},\end{aligned}\tag{3.8}$$

where \mathbf{u} is the output of the encoder as shown in Figure 3.10, and

$$\Psi_d(n) = \{\mathbf{y}(n+m), \mathbf{y}(n+m-1), \dots\}.\tag{3.9}$$

Thus, the minimum distortion is given by

$$d^* = E\left\{\|E\{\mathbf{f}|\Psi_e\} - \mathbf{x}\|^2\right\} + \min_{g,h} E\left\{\|E\{\mathbf{f}|\Psi_e\} - \mathbf{y}\|^2\right\}.\tag{3.10}$$

The previous results state that if the distortion measure is MSE (or equivalently SNR or PSNR) the optimal coder can be decomposed into an optimal MSE estimator followed by a conventional coder. Thus, the noisy image coding problem is reduced to an image restoration problem.

Chapter 4 develops MMSE image restoration algorithms to restore images corrupted by Poisson noise and film-grain noise. Chapter 5 uses these MMSE restoration algorithms to design MMSE noisy image coders for the Poisson noise case and for the film-grain noise case. Chapter 5 also studies the effect of this preprocessing step when using coders that are designed to compress noise-free images, e.g., JPEG and subband-based coders, to code noisy images.

CHAPTER 4

IMAGE RESTORATION

4.1 Introduction

Noisy source coding theory as discussed in Section 3.5 states that if the distortion measure is MSE (or equivalently SNR or PSNR) the optimal coder can be decomposed into an optimal MSE estimator followed by a conventional coder. Thus, the optimal MSE coder problem requires solving the optimal MSE restoration problem. This chapter presents a Markov random field (MRF) based approach to develop optimal MSE algorithms to restore images corrupted by data-dependent Poisson noise and data-dependent film-grain noise.

4.2 MMSE Restoration of Noisy Images

The objective is to use the noisy source coding results, which state that if the distortion is measured in terms of the MSE, $d(\mathbf{f}, \mathbf{y}) = E\{\|\mathbf{f} - \mathbf{y}\|^2\}$, where \mathbf{f} is the original image and \mathbf{y} is the quantized image, the problem will be equivalent to quantizing $E\{\mathbf{f}|\mathbf{x}\} = \sum_{\mathbf{f}} \mathbf{f}p(\mathbf{f}|\mathbf{x})$ where \mathbf{x} is the noisy image. The a posteriori probability, $p(\mathbf{f}|\mathbf{x})$, can be found from Bayes' formula

$$p(\mathbf{f}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{f})p(\mathbf{f})}{p(\mathbf{x})}, \quad (4.1)$$

where $p(\mathbf{x})$ is constant for one realization of the random field \mathbf{x} . The a priori probability distribution, $p(\mathbf{x}|\mathbf{f})$, is known from the assumptions made about the noise.

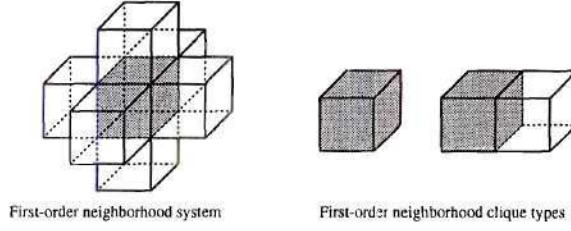


Figure 4.1: A first-order neighborhood and first-order clique types for a 3-D lattice.

Hence the only part left is the probability distribution of the image.

Markov random fields (MRFs) have been successfully used to model images. Their generality for modeling and capacity to incorporate different constraints and characteristics of images, e.g., smooth regions, edges, and local dependence between neighboring pixels, have made them popular [36, 68]. They have been used to develop algorithms to deconvolve blur and to filter white noise [21]. The next section will briefly discuss MRFs.

4.3 Markov Random Fields

Let $S_M = \{(i, j) : 1 \leq i, j \leq M\}$ be an integer lattice with a neighborhood system defined on it. Let c be a subset of S_M . Then c will be called a *clique* if it is a single site (pixel) or a set of sites (group of pixels) such that each site is a neighbor of all the other sites in the set. Figure 4.1 shows some examples of cliques. Let $\mathbf{f} = \{f(i, j) : (i, j) \in S_M\}$ be a collection of random variables defined on S_M . Then, \mathbf{x} will be a Markov random field if all of its realizations have non-zero probabilities and its conditional probability distribution satisfies the Markov property

$$p(f(i, j) | f(k, l), (k, l) \in S_M / (i, j)) = p(f(i, j) | f(k, l), (k, l) \in N_{(i, j)}), \quad (4.2)$$

where $S_M / (i, j)$ is the set S_M excluding the point (i, j) and $N_{(i, j)}$ is the set of neighbors of (i, j) [21, 73]. The joint probability distribution of \mathbf{f} is given by the Gibbs distribution [21]

$$p(\mathbf{f}) = Z^{-1} \exp[-\beta U(\mathbf{f})], \quad (4.3)$$

where Z is a normalization constant, also called the *partition function*

$$Z = \sum_{\mathbf{f}} \exp [-\beta U(\mathbf{f})], \quad (4.4)$$

and $U(\mathbf{f})$ is the energy function defined as

$$U(\mathbf{f}) = \sum_c V_c(\mathbf{f}), \quad (4.5)$$

where $V_c(\cdot)$ are the *clique potentials*, which assign penalties for amplitude separations. The choice of the clique potentials and the neighborhood system determines the joint probability distribution, $p(\mathbf{f})$.

The choice of the energy function affects the quality of the restoration, since it will govern the probabilistic interaction between neighboring pixels. The energy function can be changed by changing the neighborhood system and the clique potentials. Larger connected neighborhoods will usually result in smoother regions. However, they require more computations. Different clique potentials will result in either smoother or sharper images. A popular clique potential is the compound Gauss-Markov (CGM) MRF [73], which is defined as

$$V_c(x(m, n)) = \frac{x(m, n)^2}{2\sigma^2(m, n)} \quad (4.6)$$

$$V_c(x(m, n), x(k, l)) = \frac{-a_{i-k, j-l}x(m, n)x(k, l)}{\sigma^2(m, n)} \quad \text{for } (k, l) \in N_{(m, n)}. \quad (4.7)$$

Geman and McClure [22] used the following energy function

$$V_c(x(m, n), x(k, l)) = \frac{(x(m, n) - x(k, l))^2}{\delta^2 + (x(m, n) - x(k, l))^2}. \quad (4.8)$$

The CGM clique potential increasingly penalizes the amplitude separation between neighboring pixels by increasing their energy and lowering their probabilities. However, the clique potential proposed by Geman and McClure [22] penalizes large separations equally, which results in sharper edges at the expense of less smoothness. The following clique potential is a compromise between the characteristics of these

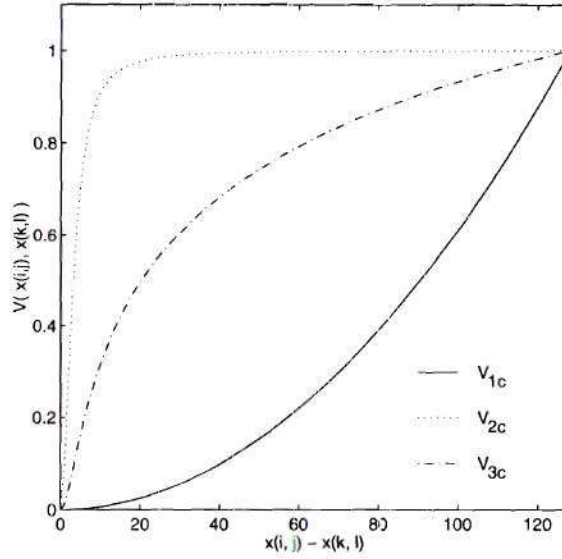


Figure 4.2: Clique potentials normalized to one as a function of pixel separation ($\sigma^2 = \delta^2 = 10$). CGM clique potential increasingly penalizes pixel separation. Geman's clique potential thresholds the penalty while the third clique potential lies in between.

two clique potentials:

$$V_c(x(m, n), x(k, l)) = \log [\delta^2 + (x(m, n) - x(k, l))^2] - \log \delta^2 + \frac{1}{\delta^2 + (x(m, n) - x(k, l))^2} - \frac{1}{\delta^2}. \quad (4.9)$$

Figure 4.2 plots these three clique potentials normalized to one for comparison purposes.

4.4 Computing the MMSE Estimate

Computing the conditional mean of the MRF $\mathbf{f}|\mathbf{x}$ (the random field \mathbf{f} given \mathbf{x}), as required for the optimal filter, requires evaluating a summation over all possible realizations of the MRF, i.e.,

$$E\{f(i, j)|\mathbf{x}\} = \sum_{\mathbf{f}} f(i, j)p(\mathbf{f}|\mathbf{x}). \quad (4.10)$$

As a result, computing $E\{\mathbf{f}|\mathbf{x}\}$ is computationally complex and nonfeasible [72]. Zhang [72] used mean field theory to approximate the summation. Mean-field the-

ory approximates this summation by assuming that the influence of $f(k, l)$, where $(k, l) \neq (i, j)$, can be approximated by $E\{f(k, l)|\mathbf{x}\}$. For a Markov random field this gives

$$E\{f(i, j)|\mathbf{x}\} \approx Z_{(i,j)}^{mf-1} \sum_{f(i,j)} f(i, j) \exp[-\beta U_{(i,j)}^{mf}(f(i, j)|\mathbf{x})], \quad (4.11)$$

where $U_{(i,j)}^{mf}(f(i, j)|\mathbf{x})$, the mean-field local energy, is the part of $U(\mathbf{f}|\mathbf{x})$ that depends on $f(i, j)$ while the influence of other pixels is approximated by their means.

Although the complexity of the problem is reduced, it still involves evaluating a summation over all possible realizations of $f(i, j)$. Geiger and Girosi [20] proposed using the saddle point approximation. This method is based on neglecting the statistical fluctuations of the field and considering only the contribution of the maximum term of the partition function. Then, the problem reduces to solving

$$\left. \frac{\partial}{\partial f(i, j)} U_{(i,j)}^{mf}(f(i, j)|\mathbf{x}) \right|_{f(i,j)=E\{f(i,j)|\mathbf{x}\}} = 0. \quad (4.12)$$

If the energy function, $U_{(i,j)}^{mf}(f(i, j)|\mathbf{x})$, is symmetric and unimodal, the saddle point approximation is the exact solution. This is because the MMSE estimate and the maximum a posteriori (MAP) estimate of a unimodal and symmetric conditional probability function are equal [63]. Although the clique potentials used are symmetric and unimodal (V_c is unimodal for $\delta^2 \leq 1$), the effect of the Poisson noise, film-grain noise, and the variations in the neighboring pixels makes $U_{(i,j)}^{mf}(f(i, j)|\mathbf{x})$ slightly non-symmetric. However, as will be demonstrated, the approximation error is very small.

Sections 4.5 and 4.6 approximate the MMSE estimate for the Poisson noise and film-grain noise cases, respectively.

4.5 Poisson Noise Case

For images limited by counting statistics, the number of photons counted for each pixel is statistically independent of the counts for other pixels [10]. Thus, the conditional

probability distribution of the observed image given the original image is

$$p(\mathbf{x}|\mathbf{f}) = \prod_{i,j} \frac{e^{-\lambda f(i,j)} [\lambda f(i,j)]^{x(i,j)}}{x(i,j)!}, \quad (4.13)$$

where $f(i,j)$ and $x(i,j)$ are the values of \mathbf{f} and \mathbf{x} at pixel (i,j) , respectively, and λ is the proportionality factor relating the gray level value to the number of counts [32]. We have assumed that $\lambda = 1$.

Substituting back into Bayes' formula results in

$$\begin{aligned} p(\mathbf{f}|\mathbf{x}) &= \frac{\exp[-\beta U(\mathbf{f})] \prod_{i,j} \left\{ \frac{\exp[-f(i,j)] [f(i,j)]^{x(i,j)}}{x(i,j)!} \right\}}{Z p(\mathbf{x})} \\ &= \hat{Z}^{-1} \exp[-\beta U(\mathbf{f}|\mathbf{x})], \end{aligned} \quad (4.14)$$

where

$$U(\mathbf{f}|\mathbf{x}) = U(\mathbf{f}) + \sum_{i,j} \frac{1}{\beta} [f(i,j) - x(i,j) \ln f(i,j)]. \quad (4.15)$$

Applying the mean-field theory approximation results in

$$U_{(i,j)}^{mf}(f(i,j)|\mathbf{x}) = U_{(i,j)}^{mf}(f(i,j)) + \frac{1}{\beta} [f(i,j) - x(i,j) \ln f(i,j)] \quad (4.16)$$

$$f^{mf}(i,j) = \frac{\sum_{f(i,j)} f(i,j) \exp[-U_{(i,j)}^{mf}(f(i,j)|\mathbf{x})]}{\sum_{f(i,j)} \exp[-U_{(i,j)}^{mf}(f(i,j)|\mathbf{x})]}, \quad (4.17)$$

$U_{(i,j)}^{mf}(\cdot)$, the mean-field local energy, is the part of $U(\cdot)$ that depends on $f(i,j)$ while the influence of other pixels is approximated by their means, $f^{mf}(i,j)$ is the mean-field theory approximation of $E\{f(i,j)|\mathbf{x}\}$. These equations are iterated until convergence or almost convergence.

Applying the saddle point approximation and using the gradient descent algorithm to solve the resulting set of equations results in the following set of iterative equations

$$\begin{aligned} \bar{f}^{t+1}(i,j) &= \bar{f}^t(i,j) - \omega \left\{ \frac{1}{\beta} \left[1 - \frac{x(i,j)}{\bar{f}^t(i,j)} \right] \right. \\ &\quad \left. + \sum_{(k,l) \in N(i,j)} \left[\frac{\partial V_t(\bar{f}^t(i,j), \bar{f}^t(k,l))}{\partial f(i,j)} \right] \right\}, \end{aligned} \quad (4.18)$$

where $\bar{f}^t(m, n)$ is the estimate of the mean-value of $f(m, n)$ at iteration t , $\beta = 1$ for all examples discussed, and ω is the step size, which is fixed to 0.1 for all examples discussed. From (4.9) the partial derivative that appears in (4.18) is seen to be

$$\frac{\partial V_{\mathbf{f}}(f(i, j), f(k, l))}{\partial f(i, j)} = \frac{(2\delta^2 - 2)(f(i, j) - f(k, l)) + 2(f(i, j) - f(k, l))^3}{[\delta^2 + (f(i, j) - f(k, l))^2]^2}. \quad (4.19)$$

A detailed derivation of these equation is in Appendix A.

This set of iterative equations is the same as the one derived by Hebert and Leahy [27], but the routes taken were different. Their objective was to develop the maximum a posteriori (MAP) image estimate using Gibbs priors. They used the expectation-maximization (EM) method to develop a deterministic MAP estimate. They defined their M-step as a coordinate gradient ascent. Our objective was to find the MMSE image estimate, which is the preprocessing part of the MMSE coder. Mean field theory was used to approximate the MMSE estimate.

The proposed approach is studied on the two Shepp-Logan phantoms shown in Figures 4.3(a) and 4.4(a). The ellipses are as described in [58]. The intensities of the small ellipses of the phantom shown in Figure 4.3(a) are in 25% and 75% increments instead of the 1% and 2% as described in [58]. This phantom will be referred to as “the modified phantom” in the rest of the thesis. The phantom shown in Figure 4.4(a) is the same as that described in [58] (with the original 1% and 2% increments) and will be referred to as “the original phantom” in the rest of the thesis. The restoration algorithm was also studied on the Lenna image, shown in Figure 4.5.

These three images are then corrupted by data-dependent Poisson noise, as shown in Figures 4.3(b), 4.4(b), and 4.5(b). The PSNR values for the noisy modified phantom, noisy original phantom, and the noisy Lenna image are 30.544 dB, 30.055 dB, and 28.159 dB, respectively. Afterwards, these three images are restored using only MFT. The restored images are shown in Figures 4.3(c), 4.4(c), and 4.5(c). The PSNR of the restored modified and original phantoms when using only MFT are 42.376 dB and 43.583 dB, respectively, corresponding to gains of 11.832 dB and

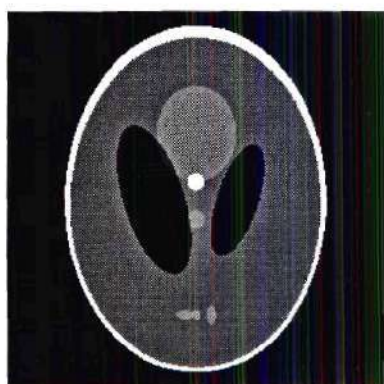
13.528 dB, respectively. Both phantoms were restored with $\delta^2 = 10$ and the pixels were updated after each iteration, i.e., the latest estimates of the neighboring pixels are used in each iteration. The number of iterations used was 7 and 8, respectively. If the pixels are updated after each pass over the image, i.e., all the pixels in the image are updated together, the PSNRs are less by almost 0.4 dB. The PSNR of the filtered noisy LENA image is 32.764 dB corresponding to a gain of 4.605 dB. The LENA image was restored with $\delta^2 = 250$. The value of δ reflects the detail in the image. The smaller δ is the smoother the image. The Shepp-Logan phantoms consist of ellipses with constant gray levels while the LENA image consists of more detailed areas, i.e., the hair, the hat, etc.. Section 4.7 discusses the effect of δ on restoration.

The restored images using both MFT and the saddle point approximation are shown in Figures 4.3(d), 4.4(d), and 4.5(d). The PSNR values of the filtered noisy modified phantom, the filtered noisy original phantom, and the filtered noisy LENA image are 41.482 dB, 42.513 dB, and 32.671 dB, respectively, corresponding to restoration gains of 10.938 dB, 12.458 dB, and 4.512 dB. Using only the MFT approximation involves computing the energy function and the exponential of the energy function at all possible gray levels. Using both approximations involves only computing the derivative of the energy function which is less expensive.

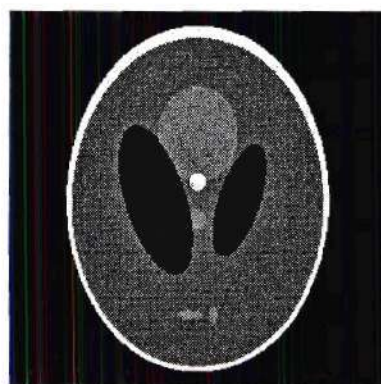
Figure 4.5(e) shows the LENA image using adaptive Wiener filtering. The PSNR of the restored image is 31.737 dB, which is almost 1 dB less than that of the proposed algorithm.

4.6 Film Grain Noise Case

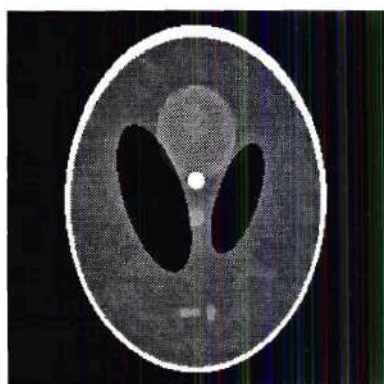
Another example of signal-dependent noise occurs when scanning images recorded on photographic films for storage and transmission. The resulting image is proportional to the film density [61]. If the film is processed in the linear region of the D-logE curve and the blurring effect of the model is ignored, the observed image, $x(i, j)$, can



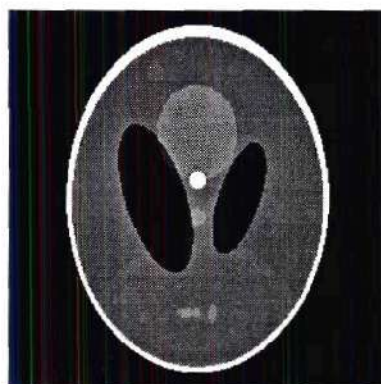
(a)



(b)



(c)



(d)

Figure 4.3: (a) The modified Shepp-Logan phantom, (b) a noisy version of the phantom (the noise is data-dependent Poisson noise with $\text{PSNR} = 30.544$ dB), (c) the restored phantom using only MFT ($\text{PSNR} = 42.376$ dB), and (d) the restored phantom using MFT and the saddle point approximation ($\text{PSNR} = 41.482$ dB). (The phantoms are histogram equalized for display by mapping $[31, 223]$ to $[0, 255]$.)

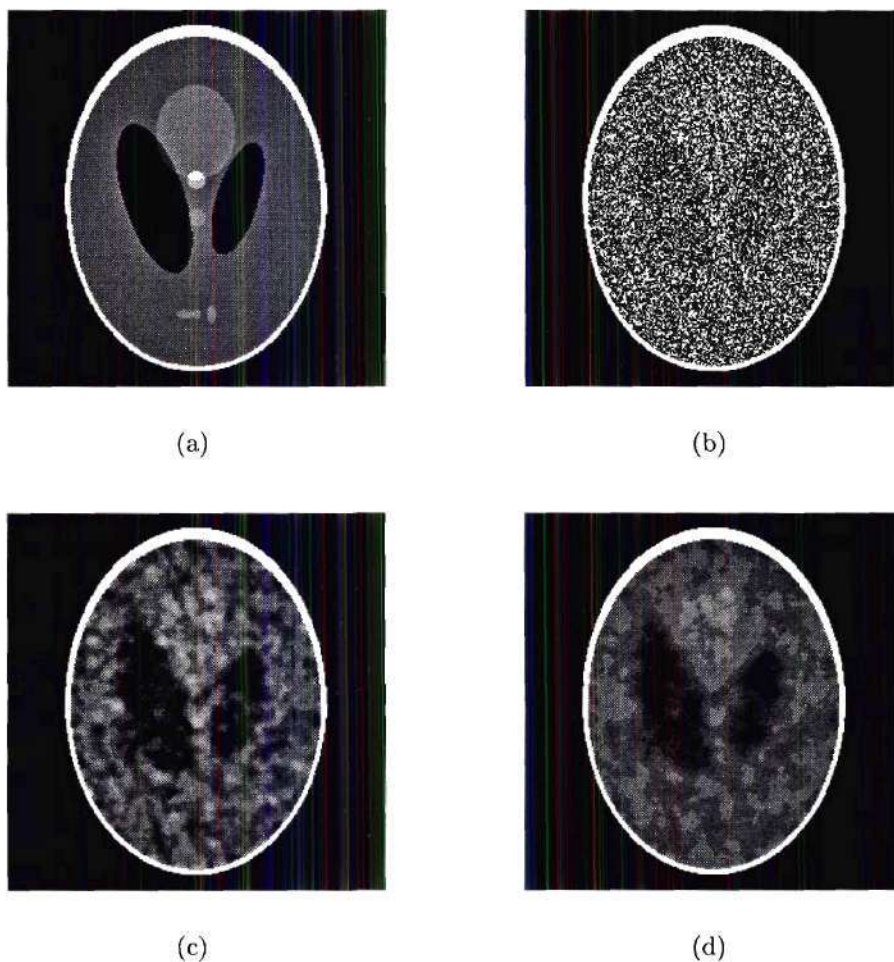


Figure 4.4: (a) The original Shepp-Logan phantom, (b) a noisy version of the phantom (the noise is data-dependent Poisson noise with $\text{PSNR} = 30.055$ dB), (c) the restored phantom using only MFT ($\text{PSNR} = 43.583$ dB), and (d) the restored phantom using MFT and the saddle point approximation ($\text{PSNR} = 42.513$ dB). (The phantoms are histogram equalized for display by mapping $[126, 136]$ to $[0, 255]$. This exaggerates the appearance of the noise.)



(a)



(b)



(c)



(d)



(e)

Figure 4.5: (a) The original 256×256 Lenna, (b) a noisy version of Lenna (The noise is data-dependent Poisson noise with $\text{PSNR} = 28.159$ dB), (c) restored Lenna image using only MFT ($\text{PSNR} = 32.764$ dB), (d) the restored Lenna image using MFT and the saddle point approximation ($\text{PSNR} = 32.671$ dB), and (e) the restored Lenna image using Wiener filtering ($\text{PSNR} = 31.737$ dB).

be modeled by [33]

$$x(i, j) = f(i, j) + \alpha[f(i, j)]^\gamma n(i, j), \quad (4.20)$$

where $f(i, j)$ is the original image density, α is a proportionality constant, γ is a constant with a value between 1/3 and 1/2, and $n(i, j)$ is white Gaussian noise with zero-mean and unit-variance. This model captures the data-dependent nature of film-grain noise.

Using this model, the conditional probability distribution of the observed image given the original image is

$$p(\mathbf{x}|\mathbf{f}) = \prod_{i,j} \frac{\exp \frac{-\{x(i,j)-f(i,j)\}^2}{2\alpha^2[f(i,j)]^{2\gamma}}}{\sqrt{2\pi}\alpha[f(i,j)]^\gamma}. \quad (4.21)$$

Substituting back into Bayes' formula results in

$$\begin{aligned} p(\mathbf{f}|\mathbf{x}) &= \frac{\exp[-\beta U(\mathbf{f})] \prod_{i,j} \left\{ \frac{\exp \frac{-\{x(i,j)-f(i,j)\}^2}{2\alpha^2[f(i,j)]^{2\gamma}}}{\sqrt{2\pi}\alpha[f(i,j)]^\gamma} \right\}}{Zp(\mathbf{x})} \\ &= \hat{Z}^{-1} \exp[-\beta U(\mathbf{f}|\mathbf{x})], \end{aligned} \quad (4.22)$$

where

$$U(\mathbf{f}|\mathbf{x}) = U(\mathbf{f}) + \sum_{i,j} \left[\frac{x(i,j) - f(i,j)}{\sqrt{2\gamma}\alpha[f(i,j)]^\gamma} \right]^2 + \frac{\gamma \ln f(i,j)}{\beta}. \quad (4.23)$$

Applying the mean-field theory approximation results in

$$\begin{aligned} U_{(i,j)}^{mf}(f(i,j)|\mathbf{x}) &= U_{(i,j)}^{mf}(f(i,j)) + \left[\frac{x(i,j) - f(i,j)}{\sqrt{2\gamma}\alpha[f(i,j)]^\gamma} \right]^2 + \frac{\gamma \ln f(i,j)}{\beta} \\ f^{mf}(i,j) &= \frac{\sum_{f(i,j)} f(i,j) \exp[-U_{(i,j)}^{mf}(f(i,j)|\mathbf{x})]}{\sum_{f(i,j)} \exp[-U_{(i,j)}^{mf}(f(i,j)|\mathbf{x})]}, \end{aligned} \quad (4.24)$$

$U_{(i,j)}^{mf}(\cdot)$, the mean-field local energy, is the part of $U(\cdot)$ that depends on $f(i, j)$ while the influence of other pixels is approximated by their means, $f^{mf}(i, j)$ is the mean-field theory approximation of $E\{f(i, j)|\mathbf{x}\}$. These equations are iterated until convergence or almost convergence.

Applying the saddle point approximation and using the gradient descent algorithm to solve the resulting set of equations results in the following set of iterative equations

$$\begin{aligned} \bar{f}^{t+1}(i, j) = & \bar{f}^t(i, j) - \omega \left\{ \frac{\gamma}{\beta \bar{f}^t(i, j)} - \frac{\left[1 - \gamma + \frac{\gamma x(i, j)}{\bar{f}^t(i, j)} \right] [x(i, j) - \bar{f}^t(i, j)]}{\beta \gamma \alpha^2 [\bar{f}^t(i, j)]^{2\gamma}} \right. \\ & \left. + \sum_{(k, l) \in N(i, j)} \frac{\partial V_{\mathbf{f}}(\bar{f}^t(i, j), \bar{f}^t(k, l))}{\partial f(i, j)} \right\}, \end{aligned} \quad (4.25)$$

where $\bar{f}^t(m, n)$ is the estimate of the mean value of $f(m, n)$ at iteration t , $\beta = 1$ for all examples discussed, and ω is the step size, which is fixed to 0.1 for all examples discussed. The required partial derivative is given in (4.19). A detailed derivation of these equation is in Appendix A.

The LENA image is used to study the proposed approach. First, the image is corrupted by film-grain noise with $\gamma = 0.5$, as shown in Figure 4.6(b). The PSNR of the noisy image is 28.174 dB. Afterwards, it is restored using the proposed MMSE restoration algorithm, resulting in the image shown in Figure 4.6(c). The PSNR of the filtered image is 32.518 dB. When restoring the noisy image using adaptive Wiener filtering, the PSNR of the restored image is 31.707 dB.

Figure 4.7(a) shows the LENA image corrupted by film-grain noise with $\gamma = 1/3$ and $\alpha = \sqrt{20}$. The PSNR of the noisy image is 22.007 dB. Figure 4.7(b) shows the filtered image. The PSNR of the filtered image is 27.768 dB, i.e., a gain of 5.761 dB.

4.7 Parameter Estimation and its Effect on Restoration

There is one free parameter, δ , in the proposed energy function. The value of δ determines the energy function and the model of the image. It controls the smoothness

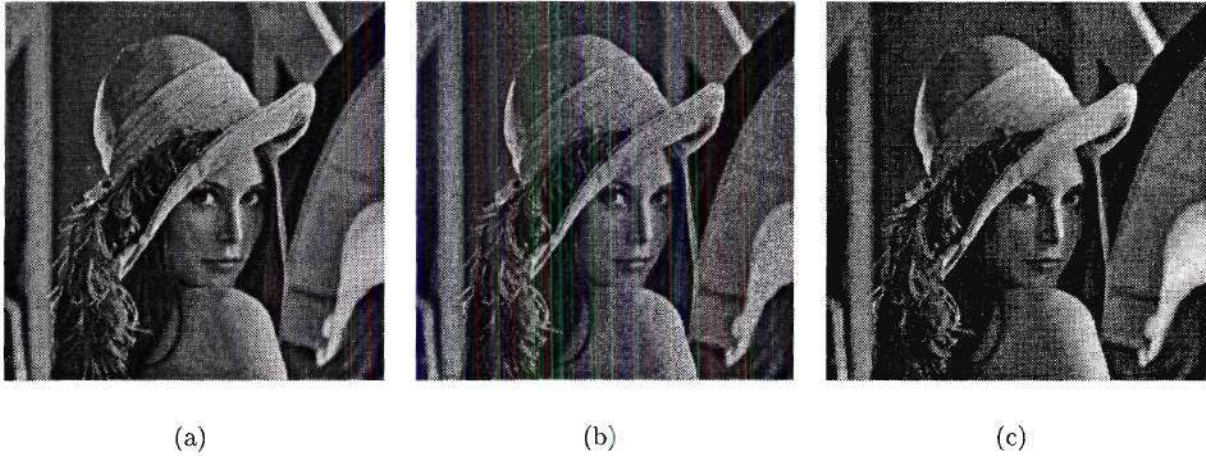


Figure 4.6: (a) The original 256×256 LENA, (b) a noisy version of LENA (The noise is film-grain with $\alpha = 1$, $\gamma = 0.5$, and $\text{PSNR} = 28.174$ dB), and (c) filtered LENA image using both approximations ($\text{PSNR} = 32.518$ dB).

and the detail in the image. The small value for δ results in more penalties for larger separations, thus a smoother image, as shown in Figure 4.8. So, for smooth images, δ should be small while for more detailed images, δ should have a larger value. For example, we used $\delta^2 = 250$, for the LENA image and $\delta^2 = 10$ for the Shepp-Logan phantoms. The values of δ for these images were estimated experimentally by trial and error.

In order to study the effect of δ on the restoration of noisy images, the LENA image example used in the Poisson noise case is restored with different values for δ while fixing the number of iterations. Figure 4.9 shows the PSNR versus δ for 2000 iterations. The value of δ^2 varied between 10 and 400. The best PSNR was 32.844 dB corresponding to $\delta^2 = 150$. Figure 4.10 shows the restored LENA for $\delta^2 = 10, 150$, and 400. The restored LENA image using $\delta^2 = 10$ is very smooth; however, it has sharp edges. This is because when δ is small, small and medium separations will be penalized, resulting in smooth areas. However, the penalty for large separations corresponding to edges between high contrast neighbors saturates resulting in edges



(a)



(b)

Figure 4.7: (a) A noisy version of LENA (The noise is film-grain with $\alpha = \sqrt{20}$, $\gamma = 1/3$, and PSNR = 22.007 dB), and (b) filtered LENA image using both approximations (PSNR = 27.768 dB).

that are not blurred.

Assuming a constant δ for the whole image implicitly means a stationary model for the image, which is not a correct model for images. However, such an assumption reduces the computations and the complexity of the restoration algorithm. Moreover, as demonstrated in this chapter, the noise was smoothed in the restored images without blurring the edges. Nevertheless, especially for images that have different regions that significantly vary in smoothness and detail, e.g., SPECT cardiac images, different values for δ for the different regions improves the performance of the restoration algorithm. Chapter 7 implements a restoration algorithm similar to the one developed in this chapter, with different δ values for the different regions in the SPECT cardiac sequences.

In the film-grain noise case, there are two parameters, α and γ , needed to model the noise. The value of α determines the power of the added noise while the value of γ reflects the data-dependent nature of the noise. As the value of γ increases, the noise depends more heavily on the signal and the greater the power of the noise. However,

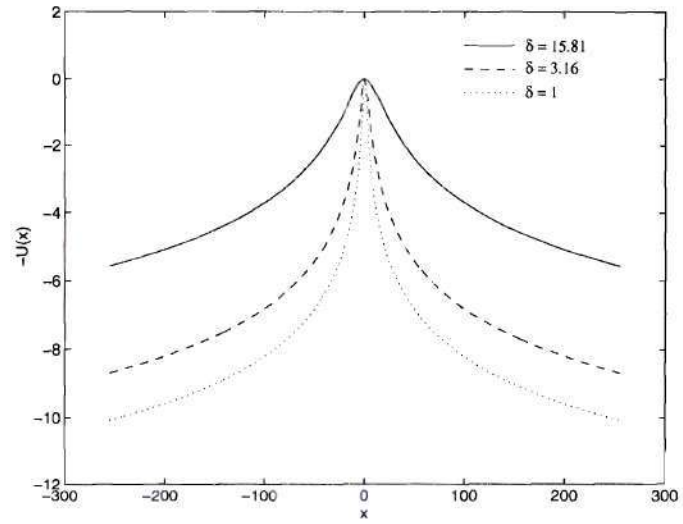


Figure 4.8: The effect of δ on the energy function.

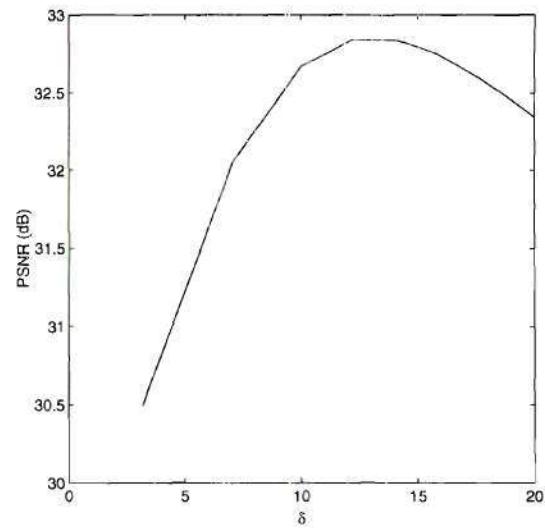


Figure 4.9: The effect of δ on restoring the LENA image shown in Figure 4.5(b).



(a) $\delta^2 = 10$



(b) $\delta^2 = 150$



(c) $\delta^2 = 400$

Figure 4.10: The Lenna image shown in Figure 4.5(b) restored using: (a) $\delta^2 = 10$ (PSNR = 30.494 dB), (b) $\delta^2 = 150$ (PSNR = 32.844 dB), and (c) $\delta^2 = 400$ (PSNR = 32.346 dB).

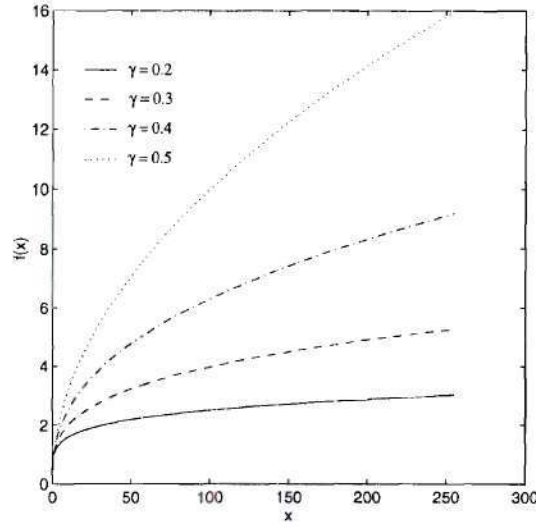


Figure 4.11: The effect of varying the value of γ on the level of the noise ($f(x) = x^\gamma$).

for very small γ , the effect of the noise is very small. The effect of γ is depicted in Figure 4.11, which plots x versus x^γ for different values for γ . For a constant γ , the value of α linearly increases the power of the added noise. The value of γ is estimated for the device used to record the images. An error in the value of γ results in removing more noise (oversmoothing) when it's underestimated and removing less noise when it's overestimated. Figure 4.12 shows the LENA image shown in Figure 4.6(b) restored using $\gamma = 0.3, 0.5$, and 0.7 . The correct value of γ is 0.5 . The image restored using $\gamma = 0.7$ is oversmoothed, while the image restored using $\gamma = 0.3$ is more noisy. Both have lower PSNRs than the one restored using the correct γ value. However, all have a higher PSNR than the noisy image.

If the film-grain noise is identified as Poisson noise or vice versa, the restoration algorithm will still effectively remove part of the noise. This is because Poisson noise can be approximated as film-grain noise with $\alpha = 1$ and $\gamma = 0.5$ [41]. Thus, if the correct noise model is Poisson noise and the image is restored using a film-grain noise restoration algorithm, the restoration algorithm will be successful in removing the noise within the limitations discussed in this section. For example if the modified Shepp-Logan phantom shown in Figure 4.3(b), the original Shepp-Logan phantom

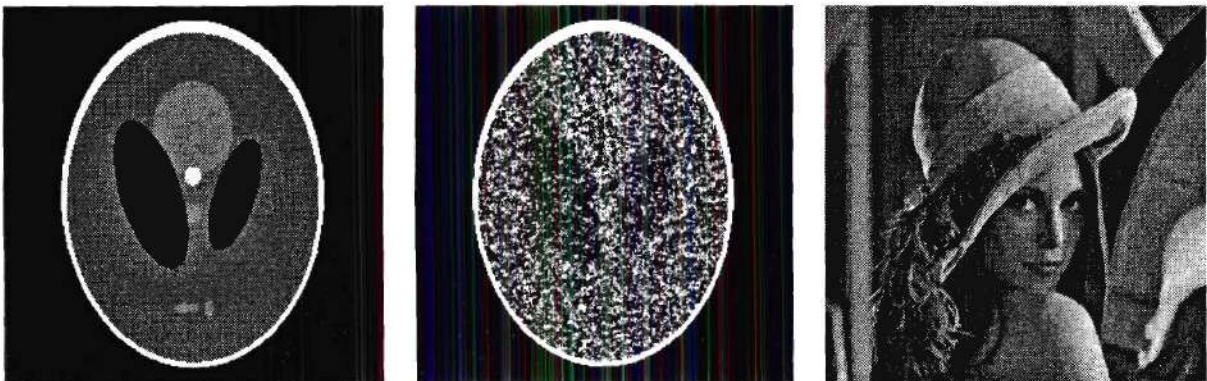


(a) $\gamma = 0.3$

(b) $\gamma = 0.5$

(c) $\gamma = 0.7$

Figure 4.12: The Lenna image shown in Figure 4.6(b) restored using: (a) $\gamma = 0.3$ (PSNR = 28.562 dB), (b) $\gamma = 0.5$ (PSNR = 32.517 dB), and (c) $\gamma = 0.7$ (PSNR = 30.599 dB).



(a)

(b)

(c)

Figure 4.13: (a) Modified Shepp-Logan phantom shown in Figure 4.3(b) (PSNR = 35.731 dB), (b) original Shepp-Logan phantom shown in Figure 4.4(b) (PSNR = 35.665 dB), and (c) Lenna image shown in Figure 4.5(b) (PSNR = 31.942 dB) restored using the film-grain noise restoration algorithm with $\alpha = 1$ and $\gamma = 0.5$.

shown in Figure 4.4(b), and the Lenna image shown in Figure 4.5(b) are restored using the film-grain restoration algorithm with $\alpha = 1$ and $\gamma = 0.5$, the PSNR values are 35.731 dB, 35.665 dB, and 31.942 dB, respectively. The restored images are shown in Figure 4.13. They all have higher PSNRs than the noisy images, however less PSNRs than the restored images using the correct model.

CHAPTER 5

COMPRESSION OF NOISY IMAGES

5.1 Introduction

Based on the performance of JPEG and Lloyd-Max quantization on noisy images, this chapter applies the results of noisy source coding theory to code noisy images at high and medium bit rates. If the quality is measured by the PSNR, the noisy image should be optimally filtered in the mean square sense and then encoded as depicted in Figure 5.1.

This chapter discusses the performance of the MMSE noisy source coder, i.e., an MMSE estimator followed by a scalar or a vector Lloyd-Max quantizer, for images corrupted by data-dependent Poisson, and film-grain noise. It also discusses the effect of the preprocessing step on the performance of JPEG and EPIC [2], which is a pyramid coder. As will be demonstrated, although the estimator-coder configuration is not optimal in these cases, the preprocessing step improves the performance of

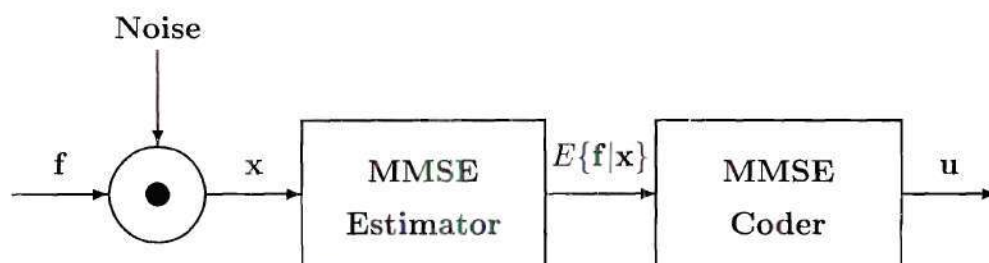


Figure 5.1: A block diagram describing the configuration for optimally compressing noisy images.

these coders.

5.2 Optimal Noisy Coders

If the quality is measured by the PSNR, the noisy image should be optimally filtered in the mean square sense and then encoded as depicted in Figure 5.1. This section discusses the performance of the optimal coder for images corrupted by Poisson noise and film-grain noise.

5.2.1 Poisson noise case

The proposed approach is studied on the two Shepp-Logan phantoms shown in Figures 4.3(a) and 4.4(a) and on the LENNA image shown in Figure 4.5.

The three original, three noisy, and three filtered images are scalar quantized using a Lloyd-Max quantizer, which is an MMSE scalar quantizer [38, 43]. The probability distribution of the quantized values is then estimated and the entropy of that distribution computed. The rate distortion curves of the noise-free, noisy, and restored modified phantoms are shown in Figure 5.2(a); the corresponding curves for the original phantom and the LENNA image are shown in Figures 5.2(b) and 5.3(a), respectively. As seen in these figures, at high and medium bit rates, the rate distortion curves of the restored phantoms are about 10 dB higher than for the noisy phantoms. However, they are limited by the PSNR of the restored phantoms. The same observation can be seen for the LENNA image. However, here the gain is approximately 5 dB. The performance of the Lloyd-Max quantizer, when quantizing the three versions of the modified phantom into one, two, or three levels and when quantizing the three versions of the original phantoms into one or two levels, is the same. This is because the quantizer successfully filters the noise out. When the original phantom is quantized to three levels, the PSNR of the quantized noisy phantom is slightly better than that of the restored phantom. This is because the quantizer

got stuck in a local minimum in which gray level 255 was quantized to 246 for the restored phantom and to 248 for the noisy one.

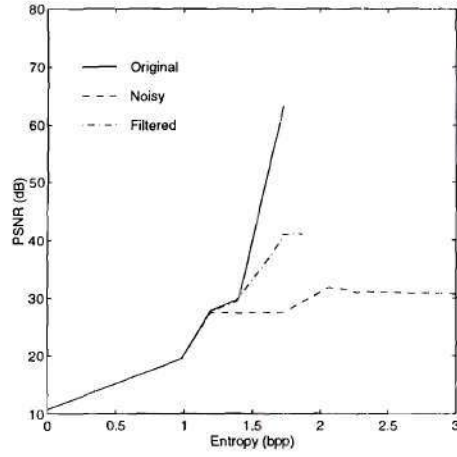
The optimal noisy vector-quantizer was also studied. Figure 5.3(b) shows the rate distortion curves of the noise-free, noisy, and restored LENA image. The results are similar to those of the scalar quantization case.

5.2.2 Film grain noise case

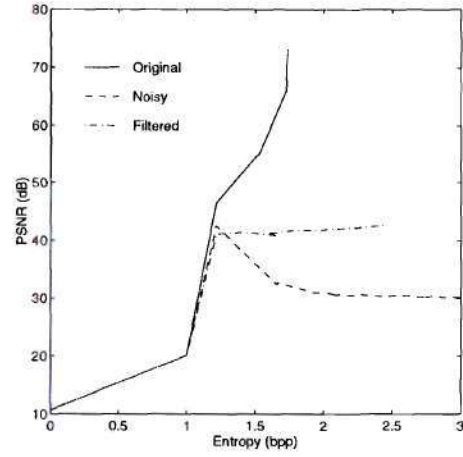
The original, noisy, and filtered images are scalar-quantized using a Lloyd-Max quantizer. The rate distortion curves of the noise-free, noisy, and restored images, when $\alpha = 1.0$, $\gamma = 0.5$, and PSNR = 28.174 dB, are shown in Figure 5.4(a). These images are also vector quantized and the corresponding rate distortion curves are shown in Figure 5.4(b). Figure 5.5 shows these curves for $\alpha = \sqrt{20}$, $\gamma = 1/3$ and PSNR = 22.007 dB. The same observations as in the Poisson case can be made about the optimal solution.

5.3 The Effect of Preprocessing on Conventional Coders

The previous section studied the performance of the optimal MMSE coder, which consists of an optimal MMSE estimator followed by an MMSE coder. This section studies the effect of replacing the MMSE coder by a conventional coder. Although, this is not the optimal coder, some gains are expected when preprocessing the image before coding it since removing the noise will increase the correlation between neighboring pixels. Two coders will be studied. The first coder is JPEG [64], which is a transform-based coder. The second coder is the EPIC [2], which is a pyramid coder. The effect of prefiltering Poisson noise and film-grain noise will be discussed.

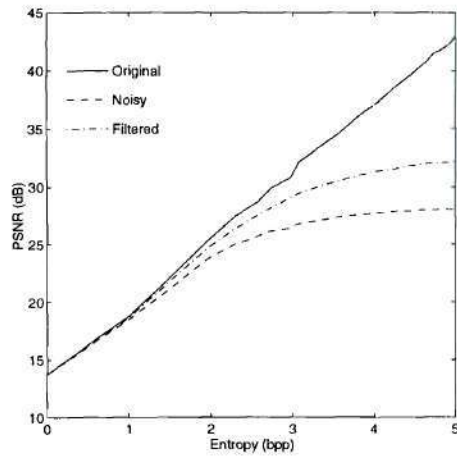


(a) Modified phantom

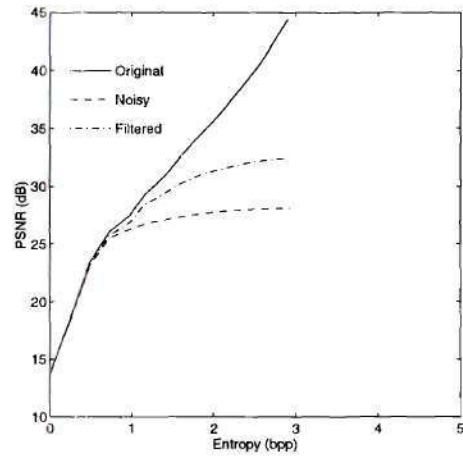


(b) Original phantom

Figure 5.2: The PSNR versus entropy curves when using the scalar Lloyd-Max quantizer for: (a) the modified Shepp-Logan phantom shown in Figure 4.3, and (b) the original Shepp-Logan phantom shown in Figure 4.4.

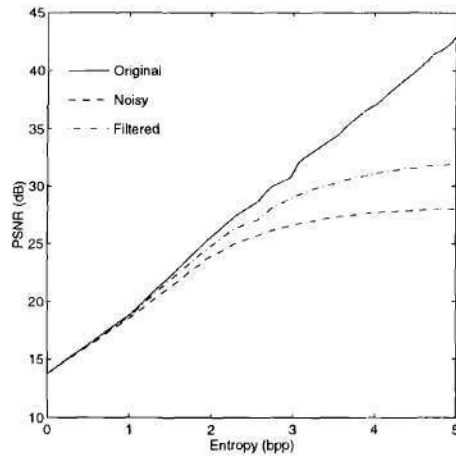


(a) Scalar Lloyd-Max

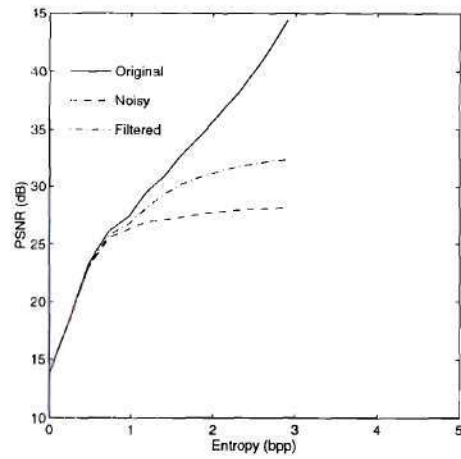


(b) 2×2 VQ

Figure 5.3: The PSNR versus entropy curves for the LENA image shown in Figure 4.5 when using: (a) the scalar Lloyd-Max quantizer and (b) the vector Lloyd-Max quantizer.

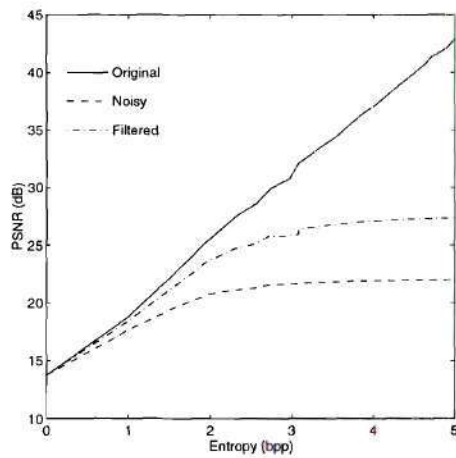


(a) Scalar Lloyd-Max

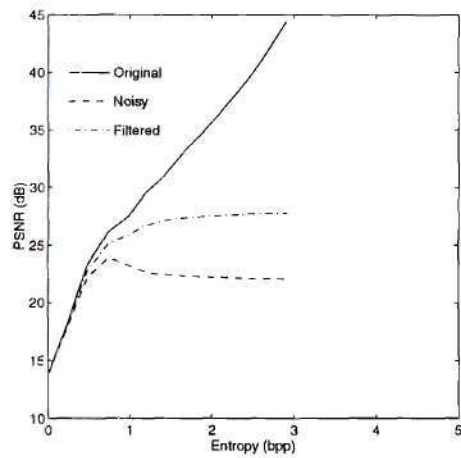


(b) 2×2 VQ

Figure 5.4: The PSNR versus entropy curves for the LENA image shown in Figure 4.6 when using: (a) the scalar Lloyd-Max quantizer and (b) the vector Lloyd-Max quantizer.



(a) Scalar Lloyd-Max



(b) 2×2 VQ

Figure 5.5: The PSNR versus entropy curves for the LENA image shown in Figure 4.7 when using: (a) the scalar Lloyd-Max quantizer and (b) the vector Lloyd-Max quantizer.

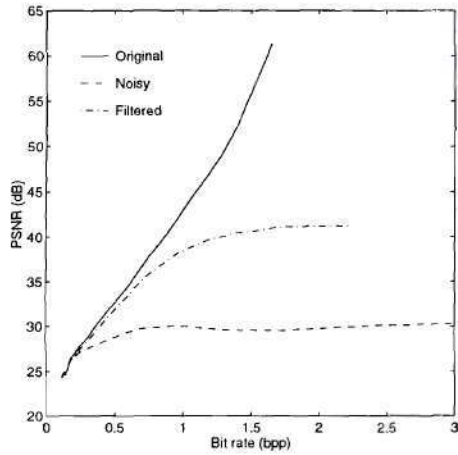
5.3.1 Poisson noise case

The effect of restoration on compression using the JPEG compression standard is studied. Figure 5.6 shows the PSNR versus bit rate curves for the three versions of the modified phantom shown in Figure 4.3 and the original phantom shown in Figure 4.4. Figure 5.8(a) shows the curves for the LENA image shown in Figure 4.5. At bit rates higher than 0.7 bpp the curves for the restored phantoms are around 12 dB higher than those of the noisy phantoms. The gain is about 5 dB for the LENA image. Although, the gain is the same as that for the Lloyd-Max quantizer case, this procedure is not optimal for the JPEG coder. This is because JPEG uses a uniform quantizer. Figure 5.7 shows the PSNR versus bit rate curves when using EPIC for the phantoms and Figure 5.8(b) shows them for the LENA image. The same observations as in the JPEG case are made.

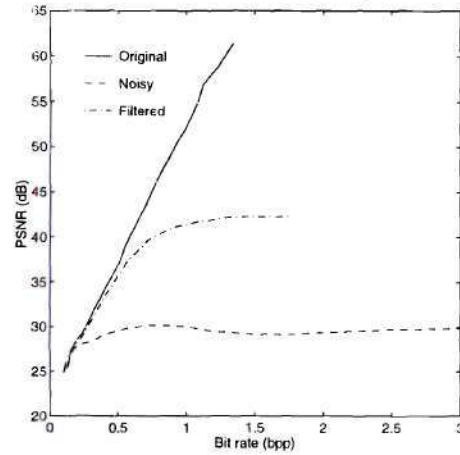
An interesting observation can be made about the curves in the medium bit rate region. As has already been noted, in this region, the quality of the compressed noisy image improves in the PSNR sense by decreasing the bit rate. This is because the coder is removing more noise than is introduced as quantization artifacts. However, when the image is filtered, the PSNR decreases by decreasing the bit rate, which is what we would expect.

5.3.2 Film grain noise case

Figure 5.9 shows the effect of restoration on the compression of the LENA image shown in Figure 4.6 using the JPEG compression standard and using the EPIC coder. Figure 5.10 shows this effect when coding the LENA image shown in Figure 4.7. The same observations as in the Poisson case are made.

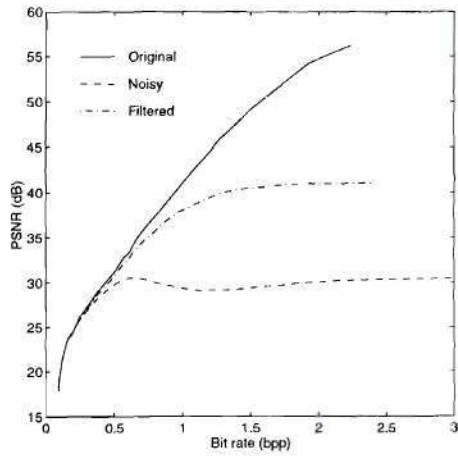


(a) Modified phantom

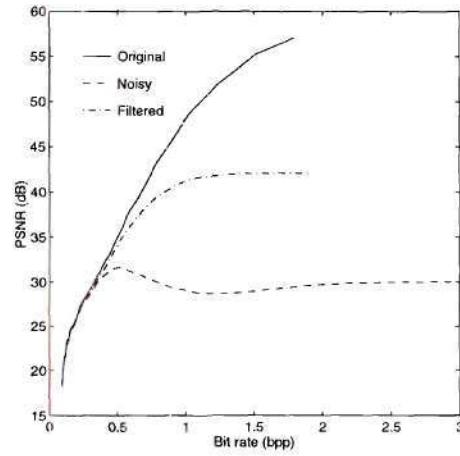


(b) Original phantom

Figure 5.6: The PSNR versus bit rate curves when using JPEG for: (a) the modified Shepp-Logan phantom shown in Figure 4.3, and (b) the original Shepp-Logan phantom shown in Figure 4.4.

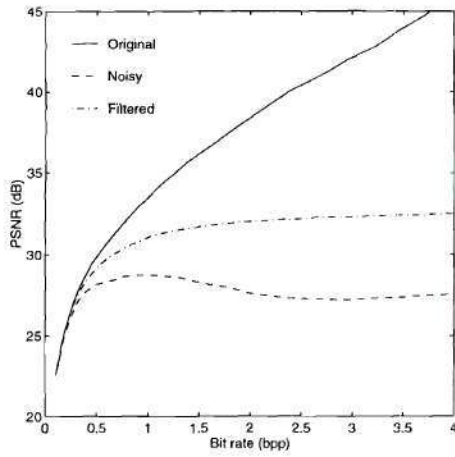


(a) Modified phantom

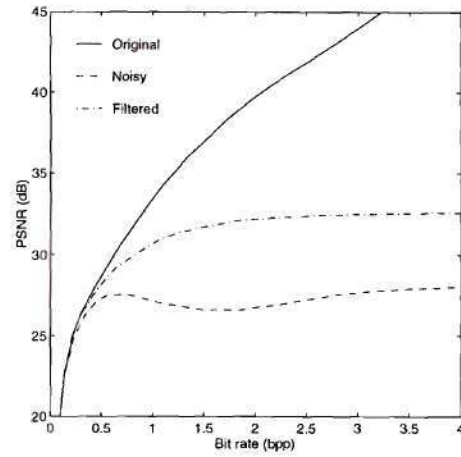


(b) Original phantom

Figure 5.7: The PSNR versus bit rate curves when using EPIC for: (a) the modified Shepp-Logan phantom shown in Figure 4.3, and (b) the original Shepp-Logan phantom shown in Figure 4.4.

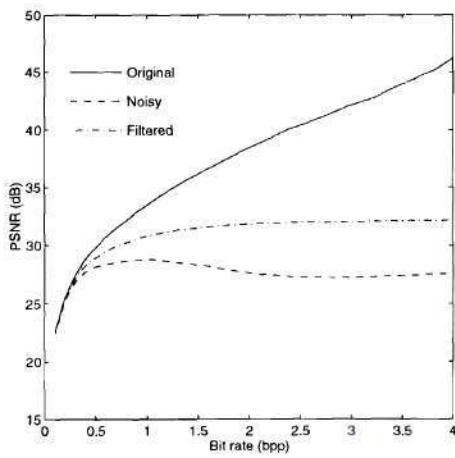


(a) JPEG

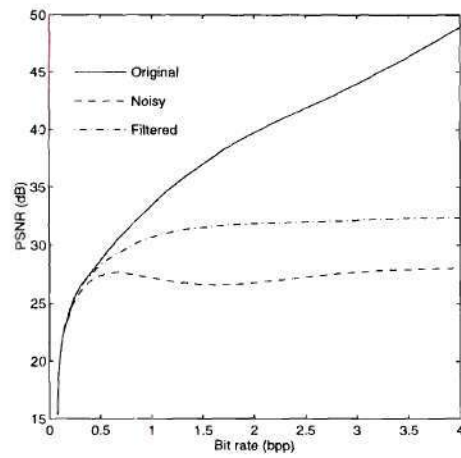


(b) EPIC

Figure 5.8: The PSNR versus bit rate curves for the LENA image shown in Figure 4.5 when using: (a) JPEG and (b) EPIC.

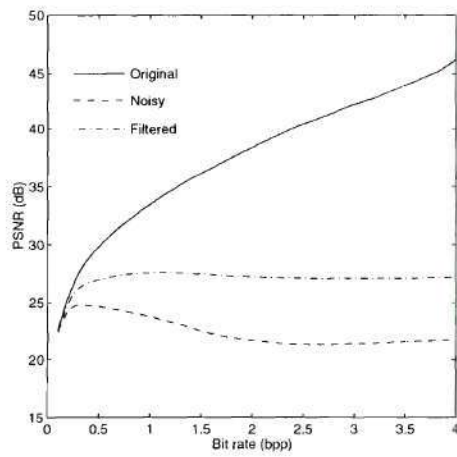


(a) JPEG

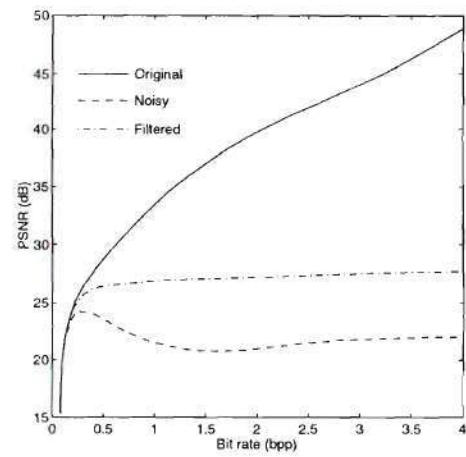


(b) EPIC

Figure 5.9: The PSNR versus bit rate curves for the LENA image shown in Figure 4.6 when using: (a) JPEG and (b) EPIC.



(a) JPEG



(b) EPIC

Figure 5.10: The PSNR versus bit rate curves for the Lenna image shown in Figure 4.7 when using (a) JPEG and (b) EPIC.

CHAPTER 6

POSTPROCESSING OF LOSSY COMPRESSED IMAGES

6.1 Introduction

Chapter 3 studied the effect of noise on image compression. Chapters 3 and 5 developed and studied the performance of the optimal coder that minimizes the distortion measured between the output of the codec and the noise-free image. Chapter 3 observed that at low bit rates, the noisy image when compressed using JPEG or VQ is closer to the original image than the noisy one is. Moreover, the lossy compressed noisy image is close to the lossy compressed noise-free image. This is because the quantization for these coders has a low-pass filtering effect on the image, hence it removes noise, which has higher frequency components. This was not observed with scalar quantization, which can not filter the noise because the quantized values are independent of the value of neighboring pixels.

At very low bit rates, the quality of the compressed image is limited and very distinctive artifacts occur. For example, using a block-transform-based coder, blocking artifacts are a major problem. These artifacts depend on the size of the block used relative to the size of the image, the frequency content of the image, and the quantizer. One way to enhance the quality of the decompressed image is to use the available information about the image, the quantizer, and the artifacts to develop a restoration algorithm [47].

This chapter describes a minimum mean squared error (MMSE) restoration algorithm for estimating noise-free images, images corrupted by data-dependent Poisson noise or by film-grain noise, that have been compressed using a block-transform coder.

This chapter is organized as follows: Section 6.2 describes the general problem of postprocessing images compressed using a block-transform coder. Section 6.3 models the blocking artifacts. Section 6.4 develops the restoration algorithm for the noise-free case. Sections 6.5 and 6.6 develop and apply the restoration algorithm for the Poisson noise case and for the film-grain noise case, respectively.

6.2 Postprocessing of Lossy Compressed Images

The objective is to minimize the mean-squared-error (MSE) between the original image, \mathbf{f} , and the filtered image, $\hat{k}(\mathbf{y})$, as depicted in Figure 6.1. Thus, the restored image is

$$\hat{k}(\mathbf{y}) = \arg \min_{k(\mathbf{y})} E \|\mathbf{f} - k(\mathbf{y})\|^2, \quad (6.1)$$

where \arg is $k(\mathbf{y})$ that minimizes the expected value. For this problem, as for the optimum preprocessing problem, the solution to this minimization problem is the Wiener (MMSE) solution [48] given by

$$\hat{k}(\mathbf{y}) = E(\mathbf{f}|\mathbf{y}) = \sum_{\mathbf{f}} \mathbf{f} p(\mathbf{f}|\mathbf{y}). \quad (6.2)$$

Using Bayes' formula

$$p(\mathbf{f}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{p(\mathbf{y})}. \quad (6.3)$$

Moreover, the quantization noise introduced by the codec and the noise at the input to the codec are independent [29]. Thus, the conditional probability $p(\mathbf{y}|\mathbf{f})$ can be computed by

$$p(\mathbf{y}|\mathbf{f}) = \int_{\mathbf{x}} p(\mathbf{y}, \mathbf{x}|\mathbf{f}) d\mathbf{x} = \int_{\mathbf{x}} p(\mathbf{y}|\mathbf{x}) p(\mathbf{x}|\mathbf{f}) d\mathbf{x}. \quad (6.4)$$

For the special case of coding noise-free images using a transform-based coder, $p(\mathbf{y}|\mathbf{f})$ is related to the pdf of the quantization noise by $p(\mathbf{y}|\mathbf{f}) = p_n(\mathbf{y} - \mathbf{f})$.

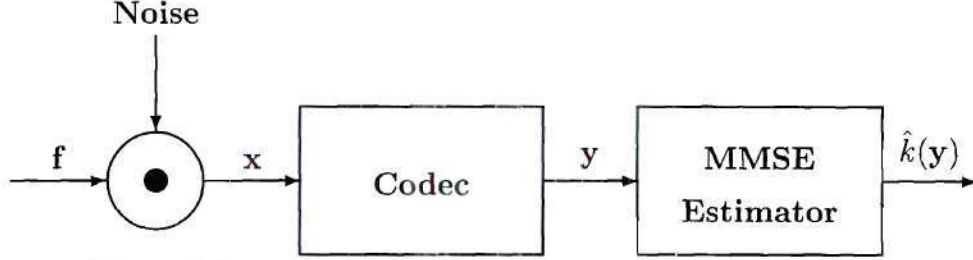


Figure 6.1: MMSE postprocessing of lossy compressed noisy images.

The observed image \mathbf{y} is a noisy version of the input of the coder, \mathbf{x} . The noise added is quantization noise which is modeled as an iid generalized Gaussian noise with zero mean. This is justified because the quantization noise is usually symmetric, unimodal, and has zero mean. Later, the quantization noise will be modeled as an iid Gaussian noise with zero mean and variance σ_q^2 . This will be experimentally justified by optimizing the shape factor of the generalized Gaussian model to achieve the best estimated image in terms of PSNR. This can also be justified by the central limit theorem applied to uniformly quantized transform coefficients.

6.3 Modeling Blocking Artifacts

The next step is to penalize for the blocking artifacts produced by block-transform coders. Two approaches are proposed. In the first approach, a penalty is introduced by multiplying the clique potential with a constant whenever a block boundary occurs. That is,

$$V_{\mathbf{f}}^{block}(f(m, n), f(k, l)) = V_{\mathbf{f}}(f(m, n), f(k, l))(1 + \mu b(m, n)), \quad (6.5)$$

where μ indicates the penalty associated with the blocking artifact and $b(m, n)$ is a binary field that indicates the presence of a horizontal or a vertical block boundary.

In the second approach, the image is modeled by two MRFs: one that represents the smooth regions in the image, and the other that represents the edge information

[21]. That is,

$$p(\mathbf{f}, \mathbf{l}) = Z^{-1} \exp [-\beta \{U_{\mathbf{f}}(\mathbf{f}) + U_{\mathbf{l}}(\mathbf{l}, \mathbf{f})\}], \quad (6.6)$$

where $p(\mathbf{f}, \mathbf{l})$ is the joint probability distribution of the intensity field, \mathbf{f} , and the line field, \mathbf{l} , $U_{\mathbf{f}}(\mathbf{f})$ is the energy function of the intensity field, and $U_{\mathbf{l}}(\mathbf{l}, \mathbf{f})$ is the energy function describing the line field. The following line field energy function, which is similar to the one in [47], is used

$$\begin{aligned} U_{\mathbf{l}}(\mathbf{l}, \mathbf{f}) = & \sum_{m,n} [V_{\mathbf{f}}(f(m, n), f(m, n-1))(1-h(m, n)) \\ & + \alpha_h h(m, n)(1-\epsilon D_h(m, n))(1+\mu b_h(m, n)) \\ & + V_{\mathbf{f}}(f(m, n), f(m-1, n))(1-v(m, n)) \\ & + \alpha_v v(m, n)(1-\epsilon D_v(m, n))(1+\mu b_v(m, n))], \end{aligned} \quad (6.7)$$

where $h(m, n)$ is a binary variable that indicates the presence of a horizontal edge between pixels (m, n) and $(m, n-1)$, $v(m, n)$ indicates the presence of a vertical edge between pixels (m, n) and $(m-1, n)$, $D_h = 0.5[h(m, n-1) + h(m, n+1)]$, $D_v = 0.5[v(m-1, n) + v(m+1, n)]$, $b_h(m, n)$ and $b_v(m, n)$ are binary parameters that indicate the presence of a block boundary, α_h and α_v represent the cost of introducing a horizontal edge and a vertical edge, and μ indicates the penalty associated with the blocking artifact. If there are no edges in the neighborhood of the pixel, only the smoothness penalty represented by $V_{\mathbf{f}}$ is added. However, if an edge occurs, the penalty introduced by $V_{\mathbf{f}}$ will be eliminated and a smaller cost is added to prevent the introduction of false edges. Moreover, if the edge is thick, i.e., D is larger than zero, the cost of introducing a new edge will be reduced by scaling it by $(1-\epsilon D) \leq 1$. However, if the edge occurs at the block boundary, the cost will be increased by scaling it by $(1+\mu b) \geq 1$.

The next three sections derive the equations for the noise-free case, which is similar to the Gaussian noise case, for the Poisson noise case, and for the film-grain noise case.

6.4 Noise Free Case

For the noise-free case, the probability of the observed image, \mathbf{y} , given the original image, \mathbf{f} , is given by the probability distribution of the quantization noise. The quantization noise is modeled as an iid generalized Gaussian noise.

If z is a generalized-Gaussian distributed random variable with zero mean and variance σ^2 , then the probability density function of z is [60, 31]

$$p(z) = a \exp[-|bz|^c], \quad (6.8)$$

where

$$a = \frac{bc}{2\Gamma(\frac{1}{c})}, \quad (6.9)$$

$$b = \frac{1}{\sigma} \sqrt{\frac{\Gamma(\frac{3}{c})}{\Gamma(\frac{1}{c})}}, \quad (6.10)$$

c is the shape factor, and $\Gamma(a)$ is the Gamma function defined as

$$\Gamma(a) = \int_0^\infty z^{a-1} \exp(-z) dz. \quad (6.11)$$

Assuming a generalized Gaussian distribution for the quantization noise defined as the difference between the input of the coder and the output of the decoder, the conditional probability of the observed image, \mathbf{y} , given the original image, \mathbf{f} , is

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i,j} a \exp[-|b(y(i,j) - f(i,j))|^c]. \quad (6.12)$$

There are different methods to estimate the parameters of the quantization noise pdf. These methods include the Kolmogorov-Smirnov test [13], the Chi-square test [13], the maximum likelihood fit [59], and the ratio of moments method [60]. The ratio of moments is the most popular method since it is the simplest and results in fits comparable to the others [31]. Thus, it will be used to estimate the shape factor, c . A zero mean assumption is made while the variance is estimated by

$$\sigma_q^2 = \frac{1}{N} \sum_{m,n=0}^{N-1} (y(m,n) - f(m,n))^2, \quad (6.13)$$

where $y(m, n)$ and $f(m, n)$ are the values of the observed and noise-free images at pixel (m, n) , respectively. We shall also look for the shape factor that results in the best image quality measured in PSNR.

After estimating $p(\mathbf{y}|\mathbf{f})$, the problem is equivalent to the optimum preprocessing problem described and solved in Chapter 4, which approximated the optimum estimate using mean field theory and a saddle point approximation. Applying these approximations and using the gradient descent algorithm to solve the resulting set of equations results in the following set of iterative equations for both penalty approaches:

- First approach (image modeled only by an intensity field):

$$\begin{aligned} \bar{f}^{t+1}(m, n) = & \bar{f}^t(m, n) - \omega \left\{ \sum_{(k, l) \in N(m, n)} \left[\frac{\partial V_{\mathbf{f}}(\bar{f}^t(m, n), \bar{f}^t(k, l))}{\partial f(m, n)} (1 + \mu b(m, n)) \right] \right. \\ & \left. + bc \frac{\text{sgn}(y(m, n) - \bar{f}^t(m, n))}{|y(m, n) - \bar{f}^t(m, n)|^{(1-c)}} \right\}, \end{aligned} \quad (6.14)$$

where $\bar{f}^t(m, n)$ is the estimate of the mean value of $f(m, n)$ at iteration t , and ω is the step size, which is fixed to 0.1 for all examples discussed.

- Second approach (image modeled by an intensity field and a line field):

$$\begin{aligned} \bar{f}^{t+1}(m, n) = & \bar{f}^t(m, n) - \omega \left\{ \sum_{(k, l) \in N(m, n)} \left[\frac{\partial V_{\mathbf{f}}(\bar{f}^t(m, n), \bar{f}^t(k, l))}{\partial f(m, n)} \right] \right. \\ & + bc |y(m, n) - \bar{f}^t(m, n)|^{(c-1)} \text{sgn}(y(m, n) - \bar{f}^t(m, n)) \\ & + \sum_{i=0}^1 (-1)^i \left[(1 - \bar{h}(m, n + i)) \frac{\partial V_{\mathbf{f}}(\bar{f}^t(m, n + i), \bar{f}^t(m, n + i - 1))}{\partial f(m, n)} \right. \\ & \left. \left. + (1 - \bar{v}(m + i, n)) \frac{\partial V_{\mathbf{f}}(\bar{f}^t(m + i, n), \bar{f}^t(m + i - 1, n))}{\partial f(m, n)} \right] \right\}, \end{aligned} \quad (6.15)$$

where $\bar{f}^t(m, n)$ is the estimate of the mean value of $f(m, n)$ at iteration t , ω is the step size, which is fixed to 0.1 for all examples discussed,

$$\bar{h}(m, n) = \frac{1}{1 + \exp(L_h(m, n))}, \quad (6.16)$$

$$\bar{v}(m, n) = \frac{1}{1 + \exp(L_v(m, n))}, \quad (6.17)$$

$$L_h(m, n) = \alpha_h(-\epsilon D_h(m, n))(1 + \mu b_h(m, n)) - V_f(f(m, n), f(m, n-1)), \quad \text{and} \quad (6.18)$$

$$L_v(m, n) = \alpha_v(-\epsilon D_v(m, n))(1 + \mu b_v(m, n)) - V_f(f(m-1, n), f(m, n)). \quad (6.19)$$

The averages of the line fields, \bar{h} and \bar{v} , are derived by Geiger and Girosi in [20] and by Özcelik *et al.* in [47].

To study the effect of the quantization noise model, the original 256×256 LENA image, shown in Figure 6.2(a), is compressed using JPEG. The PSNR of the compressed LENA image is 31.528 dB at a bit rate of 0.705 bpp, shown in Figure 6.2(b). The histogram of the quantization noise is shown in Figure 6.3. Using the ratio of moments method described in Appendix B, the shape factor $c = 0.8086$ and $\sigma_q^2 = 45.738$. Figure 6.3 shows the best fit for the histogram of the quantization noise using the ratio of moments method ($c = 0.8086$) and also the result obtained by using a Gaussian fit, i.e., $c = 2$. Visually, the ratio of moments fit is better than just using a Gaussian distribution fit. However, the shape factor that maximizes the PSNR of the restored image when using the first restoration approach is $c = 2$, i.e., a Gaussian fit. The PSNR of the restored image versus the shape factor is shown in Figure 6.4.

JPEG quantizes the transform coefficients uniformly in the coding stage. In the decoding stage, JPEG first performs an inverse discrete cosine transform, which is basically a weighted summation of the transform coefficients. Thus, the quantization errors, which were initially independent, are linearly combined. This, according to the central limit theory, justifies modeling the quantization noise as an iid Gaussian noise [48]. For the remainder of this chapter, we shall assume an iid Gaussian model for the quantization noise. Figures 6.2(c) and (d) show the results of restoring the LENA image using the no-line-field approach and the line-field approach, respectively. The PSNR of the restored image using the no-line-field approach is 32.134 dB, i.e., a gain



(a)



(b)



(c)



(d)

Figure 6.2: (a) The original 256×256 Lenna image. (b) The Lenna image compressed using JPEG (PSNR = 31.528 dB at bit rate = 0.705 bpp). (c) The decompressed Lenna image restored using the no-line-field approach (PSNR = 32.134 dB). (d) The decompressed Lenna image restored using the line-field approach (PSNR = 31.717 dB).

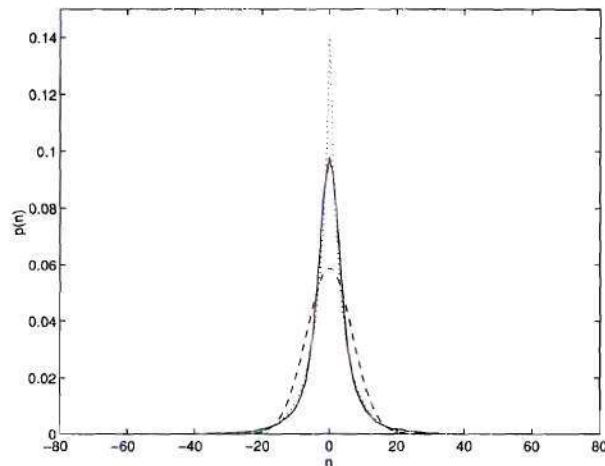


Figure 6.3: The histogram of the quantization noise that resulted from coding LENA image at a bit rate of 0.705 bpp (solid line), generalized Gaussian fit (dotted line), and Gaussian fit (dashed line).

of 0.606 dB. The PSNR of the restored image using the line-field approach is 31.717 dB, i.e., a gain of 0.189 dB.

The noise-free case is also studied on the modified and original Shepp-Logan phantoms, shown in Figures 6.5(a) and 6.6(a). The original phantoms are compressed using JPEG, as shown in Figures 6.5(b) and 6.6(b). The PSNRs of the compressed modified and original phantoms are 28.070 dB at a bit rate of 0.260 bpp and 37.152 dB at a bit rate of 0.508 bpp, respectively. Figures 6.5(c) and 6.6(c) show the result of restoring the modified and original phantoms using the no-line-field approach. The resulting PSNRs are 29.287 dB and 40.071 dB, respectively, corresponding to restoration gains of 1.217 dB and 2.919 dB. Figures 6.5(d) and 6.6(d) show the result of restoring the modified and original phantoms using the line-field approach. The resulting PSNRs are 28.626 dB and 39.701 dB, respectively, corresponding to restoration gains of 0.556 dB and 2.549 dB.

Both approaches resulted in higher PSNRs and better image quality. Both eliminated the blocking artifacts while preserving the edges in the images, however, some areas of the image were smoothed and some texture is lost. The PSNR of the images restored using the no-line field approach were higher than the restored images

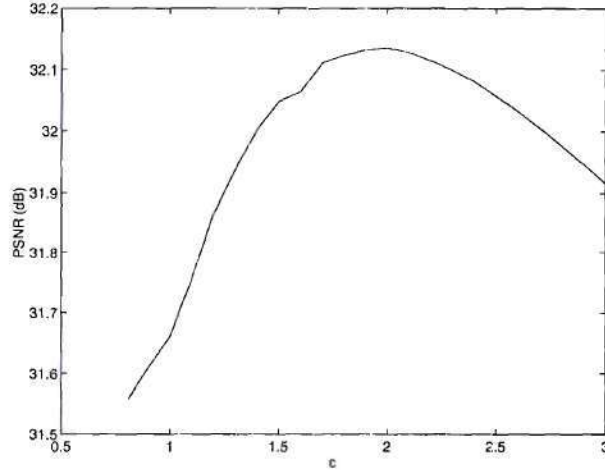


Figure 6.4: The effect of the shape factor on the PSNR of the restored image.

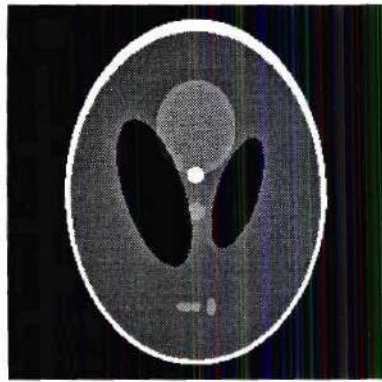
using the line-field approach. However, the perceptual quality is almost the same. Moreover, the no-line field approach is simpler and faster. This is mainly due to the energy function used. It balances the smoothing penalty at the edges and for noisy pixels.

If the image is corrupted by additive iid Gaussian noise before coding, the decoded image can be restored using the same procedure presented in this section. The variance, however, is the sum of the variance of the degradation noise and the variance of the quantization noise.

6.5 Poisson Noise Case

As we have already seen, for images limited by counting statistics, the number of photons counted for each pixel is statistically independent of the number of counts for other pixels [10]. Thus, the conditional probability distribution of the observed image given the original image is

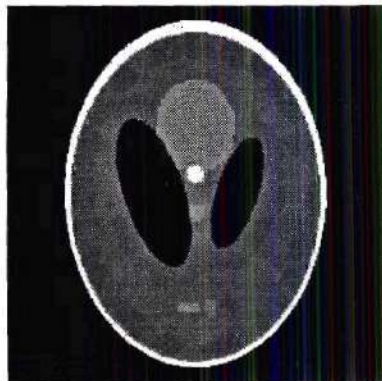
$$p(\mathbf{x}|\mathbf{f}) = \prod_{i,j} \frac{e^{-f(i,j)} [f(i,j)]^{x(i,j)}}{x(i,j)!}, \quad (6.20)$$



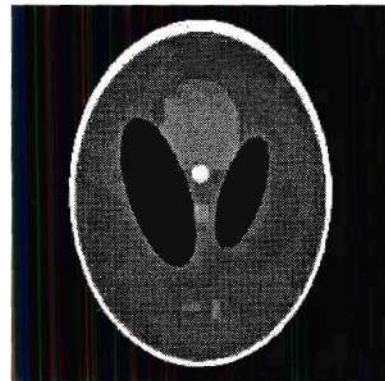
(a)



(b)

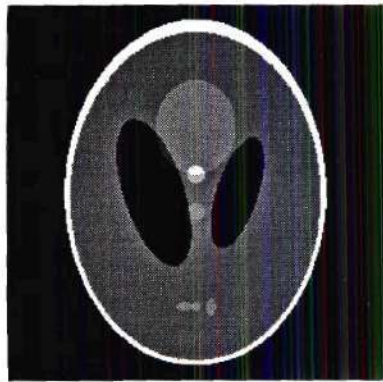


(c)



(d)

Figure 6.5: (a) The modified phantom. (b) The modified phantom compressed using JPEG (PSNR = 28.070 dB at bit rate = 0.260 bpp). (c) The decompressed modified phantom restored using the no-line-field approach (PSNR = 29.287 dB). (d) The decompressed modified phantom restored using the line-field approach (PSNR = 28.626 dB). (The phantoms are histogram equalized by mapping [31, 223] to [0, 255].)



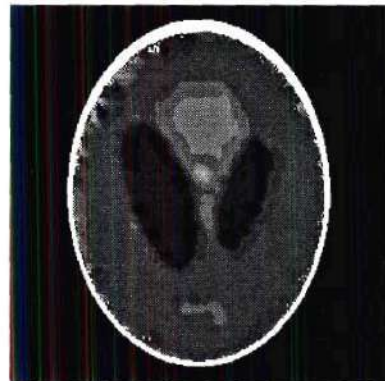
(a)



(b)



(c)



(d)

Figure 6.6: (a) The original phantom. (b) The original phantom compressed using JPEG (PSNR = 37.152 dB at bit rate = 0.508 bpp). (c) The decompressed phantom restored using the no-line-field approach (PSNR = 40.071 dB). (d) The decompressed phantom restored using the line-field approach (PSNR = 39.701 dB). (The phantoms are histogram equalized by mapping [126, 136] to [0, 255].)

where $f(i, j)$ and $x(i, j)$ are the values of \mathbf{f} and \mathbf{x} at pixel (i, j) , respectively. To get a simple closed form for the conditional probability $p(\mathbf{x}|\mathbf{f})$, the Poisson noise is approximated by additive Gaussian noise [41], i.e.,

$$p(x(m, n)|f(m, n)) = \frac{\exp \left\{ \frac{-[x(m, n) - f(m, n)]^2}{2f(m, n)} \right\}}{\sqrt{2\pi f(m, n)}}, \quad (6.21)$$

where $x(m, n)$ is the value of the random vector X at pixel (m, n) .

Substituting into Equation (6.4) results in

$$p(y(m, n)|f(m, n)) = \frac{\exp \left\{ \frac{-[y(m, n) - f(m, n)]^2}{2(\sigma_q^2 + f(m, n))} \right\}}{\sqrt{2\pi(\sigma_q^2 + f(m, n))}}. \quad (6.22)$$

Finally, applying mean field theory and the saddle point approximation and using the gradient descent algorithm to solve the resulting set of equations results in the following set of iterative equations for both penalty approaches:

- First approach (image modeled only by an intensity field):

$$\begin{aligned} \bar{f}^{t+1}(m, n) = & \bar{f}^t(m, n) - \omega \left\{ \frac{2(\bar{f}^t(m, n) - y(m, n)) + 1}{2(\sigma_q^2 + \bar{f}^t(m, n))} \right. \\ & + \sum_{(k, l) \in N(m, n)} \left[\frac{\partial V_f(\bar{f}^t(m, n), \bar{f}^t(k, l))}{\partial f(m, n)} (1 + \mu b(m, n)) \right] \\ & \left. - \frac{(\bar{f}^t(m, n) - y(m, n))^2}{2(\sigma_q^2 + \bar{f}^t(m, n))^2} \right\}, \end{aligned} \quad (6.23)$$

where $\bar{f}^t(m, n)$ is the estimate of the mean value of $f(m, n)$ at iteration t , and ω is the step size, which is fixed to 0.1 for all examples discussed.

- Second approach (image modeled by an intensity field and a line field):

$$\begin{aligned} \bar{f}^{t+1}(m, n) = & \bar{f}^t(m, n) - \omega \left\{ \sum_{(k, l) \in N(m, n)} \left[\frac{\partial V_f(\bar{f}^t(m, n), \bar{f}^t(k, l))}{\partial f(m, n)} \right] \right. \\ & + \frac{2(\bar{f}^t(m, n) - y(m, n)) + 1}{2(\sigma_q^2 + \bar{f}^t(m, n))} - \frac{(\bar{f}^t(m, n) - y(m, n))^2}{2(\sigma_q^2 + \bar{f}^t(m, n))^2} \end{aligned}$$

$$\begin{aligned}
& + \sum_{i=0}^1 (-1)^i \left[(1 - \bar{h}(m, n + i)) \frac{\partial V_f(\bar{f}^t(m, n + i), \bar{f}^t(m, n + i - 1))}{\partial f(m, n)} \right. \\
& \left. + (1 - \bar{v}(m + i, n)) \frac{\partial V_f(\bar{f}^t(m + i, n), \bar{f}^t(m + i - 1, n))}{\partial f(m, n)} \right] \Bigg\}, \quad (6.24)
\end{aligned}$$

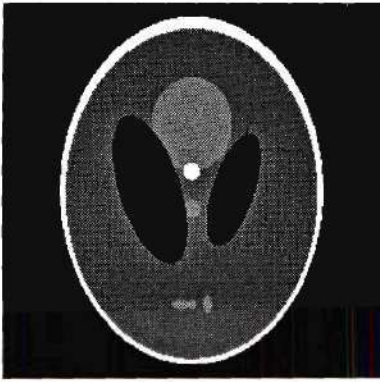
where $\bar{f}^t(m, n)$ is the estimate of the mean value of $f(m, n)$ at iteration t , ω is the step size, which is fixed to 0.1 for all examples discussed, and \bar{h} and \bar{v} are given in 6.17 and 6.18, respectively.

The three images used to study the noisy coding approach in the Poisson noise case are shown in Figures 6.7(a), 6.8(a), and 6.9(a). These three images are corrupted by data-dependent Poisson noise, as shown in Figures 6.7(b), 6.8(b), and 6.9(b). The PSNR values of the noisy modified phantom, noisy original phantom, and the noisy LENA image are 30.544 dB, 30.055 dB, and 28.159 dB, respectively.

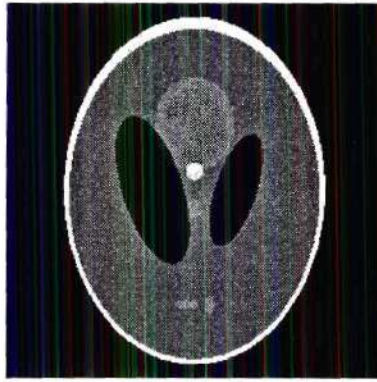
The noisy images are then compressed using JPEG, as shown in Figures 6.7(c), 6.8(c), and 6.9(c). The PSNRs of the compressed noisy modified phantom, noisy original phantom, and noisy LENA image are 27.374 dB at a bit rate of 0.267 bpp, 29.537 dB at a bit rate of 0.503 bpp, and 27.745 dB at a bit rate of 0.395 bpp, respectively.

Figures 6.7(d), 6.8(d), and 6.9(d) show the result of restoring the decompressed noisy modified phantom, original phantom, and the LENA image using the no-line-field approach. The resulting PSNRs are 28.786 dB, 32.814 dB, and 28.714 dB, respectively. Figures 6.7(e), 6.8(e), and 6.9(e) show the result of restoring the decompressed noisy modified phantom, original phantom, and the LENA image using the line-field approach. The resulting PSNRs are 28.093 dB, 32.377 dB, and 28.169 dB, respectively.

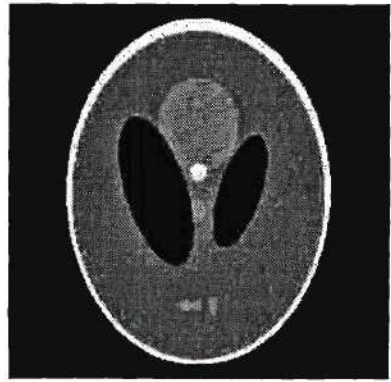
As shown, the visual quality of the restored images is better than the compressed ones. The restoration process removed the blocking artifacts while preserving the edges in the image, however, some areas of the restored images were smoothed and some texture is lost. Moreover, the PSNR of the restored images is higher than



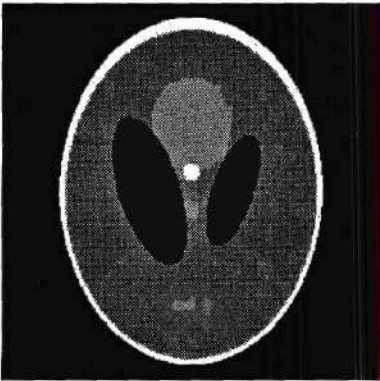
(a)



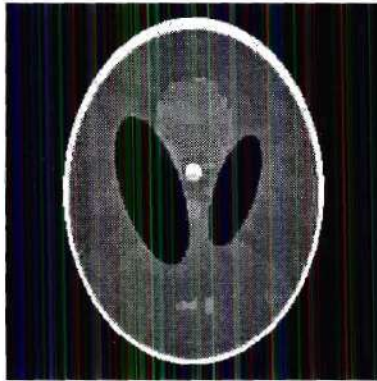
(b)



(c)



(d)



(e)

Figure 6.7: (a) The modified phantom. (b) The noisy modified phantom (The noise is data-dependent Poisson noise with $\text{PSNR} = 30.544$ dB). (c) The noisy modified phantom compressed using JPEG ($\text{PSNR} = 27.374$ dB at bit rate = 0.267 bpp). (d) The compressed noisy modified phantom restored using the no-line-field approach ($\text{PSNR} = 28.786$ dB). (e) The noisy compressed modified phantom restored using the line-field approach ($\text{PSNR} = 28.093$ dB). (The phantoms are histogram equalized by mapping $[31, 223]$ to $[0, 255]$.)

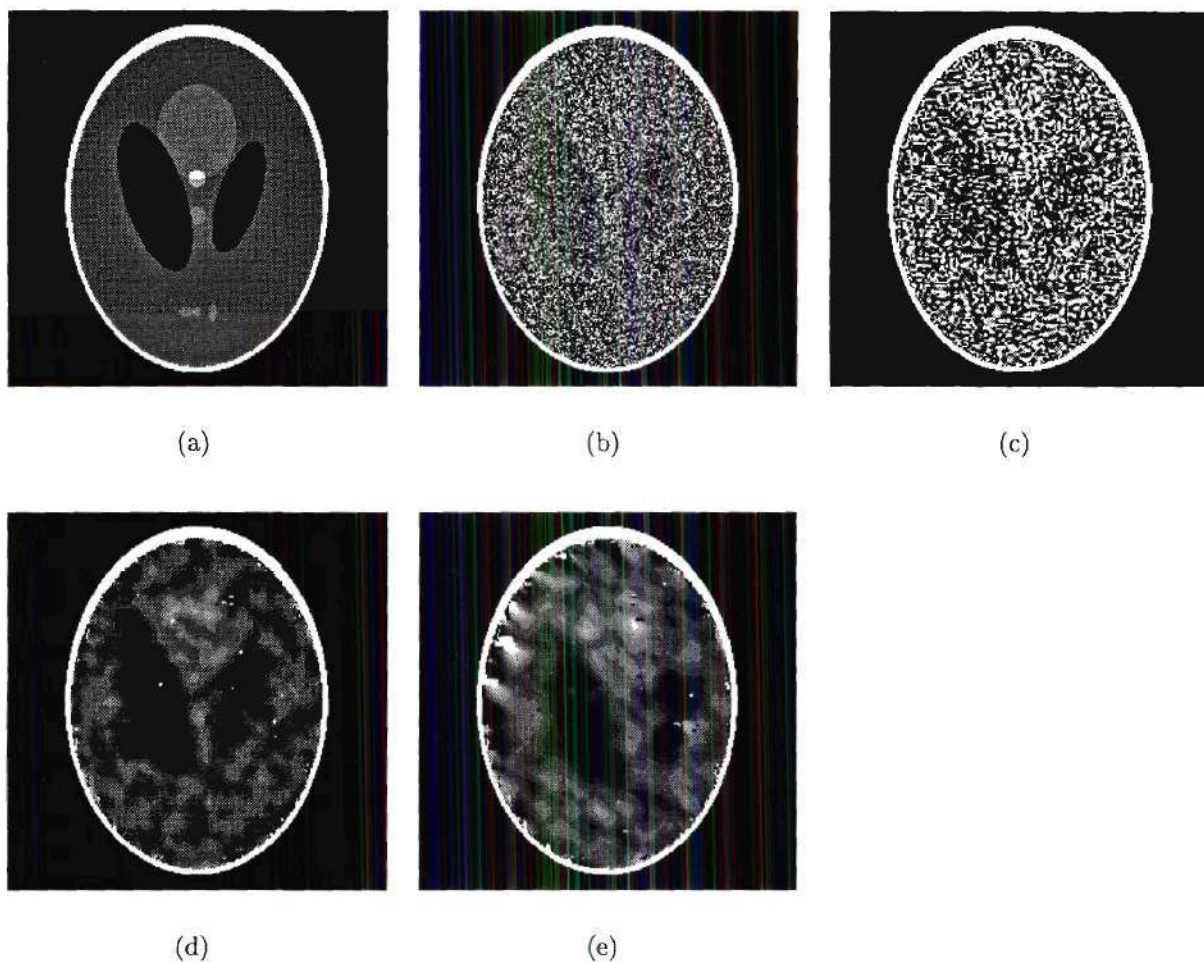


Figure 6.8: (a) The original phantom. (b) The noisy phantom (The noise is data-dependent Poisson noise with $\text{PSNR} = 30.055$ dB). (c) The noisy phantom compressed using JPEG ($\text{PSNR} = 29.537$ dB at bit rate = 0.503 bpp). (d) The compressed noisy phantom restored using the no-line-field approach ($\text{PSNR} = 32.814$ dB). (e) The noisy compressed phantom restored using the line-field approach ($\text{PSNR} = 32.377$ dB). (The phantoms are histogram equalized by mapping $[126, 136]$ to $[0, 255]$. This exaggerates the appearance of the noise.)



(a)



(b)



(c)



(d)



(e)

Figure 6.9: (a) The original LENA image. (b) The noisy LENA image (The noise is data-dependent Poisson noise with $\text{PSNR} = 28.159$ dB). (c) The noisy LENA image compressed using JPEG ($\text{PSNR} = 27.745$ dB at bit rate = 0.395 bpp). (d) The compressed noisy image restored using the no-line-field approach ($\text{PSNR} = 28.714$ dB). (e) The noisy compressed image restored using the line-field approach ($\text{PSNR} = 28.169$ dB).

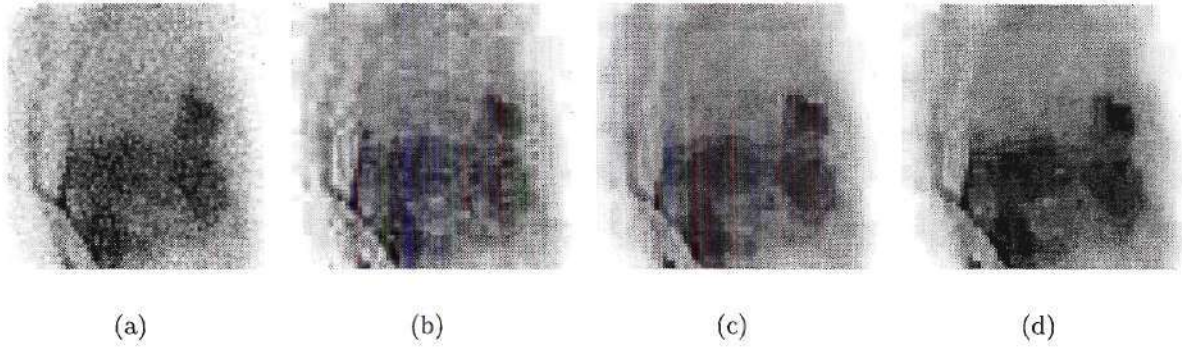


Figure 6.10: (a) The original noisy image. (b) The output of the codec, with compression ratio=8.982. (c) The restored image using the no-line-field approach. (d) The restored image using the line-field approach. (Gray levels are reversed for display.)

that of the compressed ones by 1.0 – 3.3 dB using the no-line-field approach and by 0.4 – 2.8 dB using the line-field approach. The quality and the PSNR of the images restored using the no-line-field approach are better than those of the images restored using the line-field approach.

The proposed approach is used to restore a 64×64 cardiac SPECT projection image compressed using JPEG, as shown in Figure 6.10. The compression ratio is 8.982:1.

6.6 Film Grain Noise Case

Using the model of 3.3, the conditional probability distribution of the observed image given the original image is

$$p(\mathbf{x}|\mathbf{f}) = \prod_{i,j} \frac{\exp \frac{-\{x(i,j)-f(i,j)\}^2}{2\alpha^2[f(i,j)]^{2\gamma}}}{\sqrt{2\pi}\alpha[f(i,j)]^\gamma}, \quad (6.25)$$

where $f(i, j)$ and $x(i, j)$ are the values of \mathbf{f} and \mathbf{x} at pixel (i, j) , respectively.

Substituting in Equation (6.4) results in

$$p(y(m, n)|f(m, n)) = \frac{\exp \left\{ \frac{-[y(m, n)-f(m, n)]^2}{2(\sigma_q^2 + \alpha^2[f(m, n)]^{2\gamma})} \right\}}{\sqrt{2\pi}(\sigma_q^2 + \alpha^2[f(m, n)]^{2\gamma})}. \quad (6.26)$$

Finally, applying mean field theory and the saddle point approximation and using the gradient descent algorithm to solve the resulting set of equations results in the following set of iterative equations for both penalty approaches:

- First approach (image modeled only by an intensity field):

$$\begin{aligned} \bar{f}^{t+1}(m, n) = \bar{f}^t(m, n) - \omega \left\{ \frac{\gamma \alpha^2 [\bar{f}^t(m, n)]^{(2\gamma-1)} + (\bar{f}^t(m, n) - y(m, n))}{(\sigma_q^2 + \alpha^2 [\bar{f}^t(m, n)]^{2\gamma})} \right. \\ \left. + \sum_{(k, l) \in N(m, n)} \left[\frac{\partial V_{\mathbf{f}}(\bar{f}^t(m, n), \bar{f}^t(k, l))}{\partial f(m, n)} (1 + \mu b(m, n)) \right] \right. \\ \left. - \frac{\alpha^2 \gamma (\bar{f}^t(m, n) - y(m, n))^2 [\bar{f}^t(m, n)]^{(2\gamma-1)}}{(\sigma_q^2 + \alpha^2 [\bar{f}^t(m, n)]^{2\gamma})^2} \right\}, \quad (6.27) \end{aligned}$$

where $\bar{f}^t(m, n)$ is the estimate of the mean value of $f(m, n)$ at iteration t , and ω is the step size, which is fixed to 0.1 for all examples discussed.

- Second approach (image modeled by an intensity field and a line field):

$$\begin{aligned} \bar{f}^{t+1}(m, n) = \bar{f}^t(m, n) - \omega \left\{ \sum_{(k, l) \in N(m, n)} \left[\frac{\partial V_{\mathbf{f}}(\bar{f}^t(m, n), \bar{f}^t(k, l))}{\partial f(m, n)} \right] \right. \\ \left. + \frac{\gamma \alpha^2 [\bar{f}^t(m, n)]^{(2\gamma-1)} + (\bar{f}^t(m, n) - y(m, n))}{(\sigma_q^2 + \alpha^2 [\bar{f}^t(m, n)]^{2\gamma})} \right. \\ \left. - \frac{\alpha^2 \gamma (\bar{f}^t(m, n) - y(m, n))^2 [\bar{f}^t(m, n)]^{(2\gamma-1)}}{(\sigma_q^2 + \alpha^2 [\bar{f}^t(m, n)]^{2\gamma})^2} \right. \\ \left. + \sum_{i=0}^1 (-1)^i \left[(1 - \bar{h}(m, n + i)) \frac{\partial V_{\mathbf{f}}(\bar{f}^t(m, n + i), \bar{f}^t(m, n + i - 1))}{\partial f(m, n)} \right. \right. \\ \left. \left. + (1 - \bar{v}(m + i, n)) \frac{\partial V_{\mathbf{f}}(\bar{f}^t(m + i, n), \bar{f}^t(m + i - 1, n))}{\partial f(m, n)} \right] \right\}, \quad (6.28) \end{aligned}$$

where $\bar{f}^t(m, n)$ is the estimate of the mean value of $f(m, n)$ at iteration t , ω is the step size, which is fixed to 0.1 for all examples discussed, and \bar{h} and \bar{v} are given in 6.17 and 6.18, respectively.

A point worth mentioning is that if $\alpha = 0$, the equations derived in this section will be equivalent to the noise-free case.



(a)



(b)



(c)



(d)



(e)

Figure 6.11: (a) The original Lenna image. (b) The noisy Lenna image (The noise is data-dependent with $\text{PSNR} = 28.174$ dB). (c) The noisy Lenna image compressed using JPEG ($\text{PSNR} = 28.527$ dB at bit rate = 0.701 bpp). (d) The compressed noisy image restored using the no-line-field approach ($\text{PSNR} = 29.982$ dB). (e) The noisy compressed image restored using the line-field approach ($\text{PSNR} = 29.495$ dB).

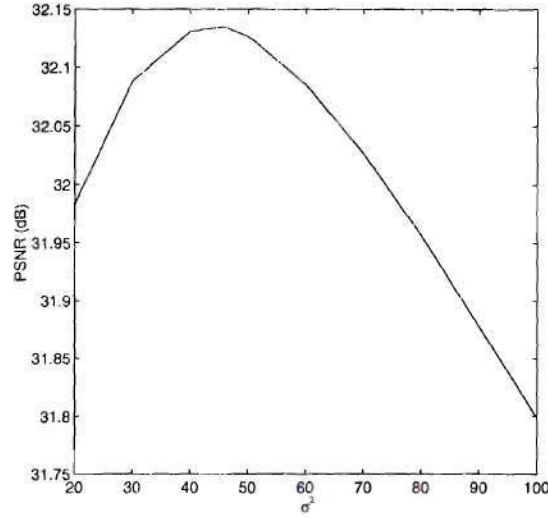


Figure 6.12: PSNR versus σ_q^2 when restoring the LENA image shown in Figure 6.2(b) using the no-line-field approach.

The LENA image, as shown in Figure 6.11(a) is used to study the performance of the restoration algorithm in the film-grain noise case. First, the LENA image is corrupted by film-grain noise with $\gamma = 0.5$ and $\alpha = 1.0$, as shown in Figure 6.11(b). The PSNR of the noisy image is 28.174 dB. Afterwards, the noisy image is compressed using JPEG. The bit rate of the compressed noisy image is 0.701 bpp. The PSNR of the decompressed image is 28.527 dB, as shown in Figure 6.11(c). The resulting image is then restored using the no-line-field and the line-field approaches, as shown in Figures 6.11(d) and (e). The PSNR of the restored images are 29.982 dB and 29.495 dB, respectively, corresponding to restoration gains of 1.455 dB and 0.968 dB.

6.7 Estimation of Restoration Parameters

The parameter estimation of the image and the degradation noise models and its effect on restoration was discussed in Section 4.7. The problem of restoring lossy compressed images requires estimating the variance of the quantization noise. The

variance of the quantization noise was estimated using

$$\sigma_q^2 = \frac{1}{M^2} \sum_{i,j=0}^M (x(i,j) - y(i,j))^2, \quad (6.29)$$

where $M \times M$ is the size of the image.

The variance of the input noise and the quantization noise can be sent by the coder or estimated from the rate distortion curve of the coder [46]. The variance of the quantization noise can be estimated locally by assuming that gray levels are constant for small regions, thus the local variance in small regions is an estimate of the variance of the quantization noise [23, 57].

To study the effect of not estimating the variance of the quantization noise correctly, the Lenna image shown in Figure 6.2 will be restored using different values for σ_q^2 . The PSNR of the compressed Lenna image is 31.528 dB at 0.705 bpp. The estimated quantization noise variance is 45.738. Figure 6.12 shows the PSNR versus σ_q^2 curve. As expected, the best PSNR is when $\sigma_q^2 = 45.738$ and it decays almost symmetrically.

CHAPTER 7

APPLICATION: COMPRESSION OF CARDIAC SEQUENCES

7.1 Introduction

As an application, the results of Chapters 3 and 5 are applied to medical images and in particular to cardiac SPECT sequences. Data-dependent Poisson noise corrupts cardiac SPECT sequences. Thus, degrading the performance of compression algorithms.

Since medical images are different than other classes of images, Section 7.2 briefly discusses medical image compression. Section 7.3 discusses the evaluation of lossy compressed medical images.

Since it is desired to preserve the diagnostic information in medical images, a content-based coder is developed in Section 7.4. This coder preserves the diagnostic information in the image by segmenting the image into different regions, filtering these regions with different image models according to the detail in each region, and coding these regions with different coders. The regions that contain the diagnostic information are compressed without coding distortion, while the other regions are compressed lossily.

Section 7.5 applies the results of Chapters 4 and 5 to this problem. Section 7.6 studies the performance of the proposed algorithms on a SPECT cardiac sequence. Appendix C discusses two more examples.

7.2 Medical Image Compression

Digitization of medical images has forced interest in picture archiving and communication systems (PACS). Digitization allows easy retrieval, efficient storage, and rapid transmission of medical images [15]. Moreover, it makes digital signal processing algorithms, e.g., filtering, enhancement, and classification, accessible to the field of medical imaging.

When applying existing image compression algorithms to medical images, one should take into account the differences between medical images and the images for which these algorithms were developed. Medical images are different from other classes of images. They are different in the way they are taken, the variations of gray levels and colors, the dynamic range, the size, and the type of noise that corrupts them. Moreover, the way medical image compression algorithms are evaluated differs from the evaluation of compression algorithms for other classes of images. By way of explanation, when compressing images of natural scenes, compression algorithms are concerned with the visual perception of the resulting image. However, medical image compression algorithms should be concerned with preserving the diagnostic information in the image. Although lossless image compression preserves all the information in the image, including the diagnostic information, it fails to reduce significantly the number of bits representing the image. Thus, to achieve high compression ratios, we must look at lossy image compression algorithms that preserve the diagnostic information.

Many medical imaging systems are based on three techniques: transmission tomography, reflection tomography, and emission tomography [41]. Transmission tomography is based on measuring the attenuation of X-rays when transmitted into the body [28]. Reflection tomography is based on measuring the reflection of ultrasound waves transmitted through the body [41]. Emission tomography is based on measuring gamma rays emitted from pharmaceuticals that have been deposited in the body [41]. Three types of imaging systems are regularly used in emission tomography.

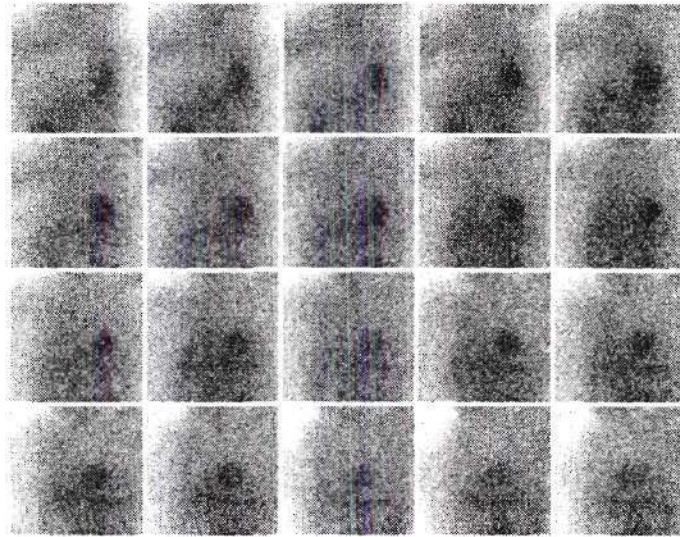


Figure 7.1: The first 20 slices of a $32 \times 64 \times 64$ SPECT cardiac image sequence.

These are SPECT (Single Photon Emission Computerized Tomography) and PET (Positron Emission Tomography) [28]. Medical images can be still images, video images, projection sequences (two-dimensional projections taken around the body), or three-dimensional video. This chapter applies the results of the previous chapters to compress SPECT cardiac images, which are slices of projection sequences and whole projection sequences.

The size of medical images can be as small as 64×64 for SPECT projections or as large as 4096×4096 for mammograms and for whole body scans. Medical images are most commonly represented with 8, 12, 16, 32, or 64 bits per pixel. Although a few projection images are stored as floating point numbers, most are stored as integer values.

The general structure of many medical images consists of an approximately circular or flat region containing almost all the energy of the image. A typical SPECT cardiac projection image consists of a dark background with a light body region and bright small spots corresponding to organs as shown in Figures 1.1 and 7.1 (in which the intensities have been reversed).

Transmission and emission tomography images are usually corrupted by mea-

surement and quantum noises [7, 41]. The measurement noise is effectively modeled as additive Gaussian noise [41]. The quantum noise is caused by the quantization of energy into photons, which are individually counted. It is Poisson distributed and independent of the measurement noise. In practice, the Poisson noise dominates the measurement noise [41], which is often ignored.

7.3 Evaluating the Quality of Lossy Compressed Medical Images

In many countries, it is required to store medical images without distortion in order to preserve all diagnostic information. However, lossless image compression algorithms fail to reduce the storage bit-rate significantly. For example, Table 7.3 compares the marginal entropy of the sequence, Lempel-Ziv coding, LPC coding, coding of projection differences, and JPEG lossless mode 2 coding. These compression algorithms were applied to 16-bit $32 \times 64 \times 64$ cardiac SPECT projection sequences. The linear prediction mask used in the LPC coding is the same as in Equation 2.8, but the prediction parameters are estimated optimally by minimizing the mean squared error. The projection difference coding is done by re-addressing the image pixels to display all the projections of one slice in each image, resulting in a set of 64 32×64 images. Afterwards, the difference between successive images is encoded. As seen in Table 7.3, all algorithms did not perform particularly well. To achieve higher compression ratios, we must look at lossy compression algorithms that preserve diagnostic information.

Evaluating the quality of a compressed image is a major difference between medical images and other classes of images. Medical image quality is usually defined in terms of how well an expert can perform some task of diagnostic interest [5], whereas the quality of images of natural scenes is defined in terms of their visual perception. Evaluating the quality of medical images is very important for both scientific and legal reasons.

Table 7.1: Comparison between various lossless coding methods. The comparison was performed on $32 \times 64 \times 64$ cardiac projection sequences of patients from Crawford Long Hospital. The comparison is in terms of the average number of bits per pixel (bpp). the original sequences were represented with 16 bits.

Patient	Marginal entropy	LPC	Projection difference	Lempel-Ziv	JPEG mode 2
547	5.66	4.32	4.56	5.98	4.56
552	5.52	4.29	4.54	5.74	4.53
590	5.76	4.48	4.73	6.10	4.70
611	5.40	4.28	4.52	5.73	4.50
616	5.30	4.21	4.46	5.57	4.39
640	5.07	4.16	4.42	5.42	4.38
644	4.92	4.15	4.41	5.32	4.39
712	6.28	4.74	4.95	6.64	4.95
720	6.05	4.64	4.85	6.38	4.85
814	6.12	4.76	5.02	6.82	4.94

The quality of lossy compressed medical images is usually evaluated using distortion measures, subjective ratings, diagnostic accuracy measures, or a combination of these measures [15, 16].

7.3.1 Distortion measures

Distortion measures such as mean-squared error (MSE), signal-to-noise ratio (SNR), and peak-signal-to-noise ratio (PSNR) are easy to compute and are tractable in analysis. However, they rarely reflect the degradation in perceptual quality or loss in diagnostic accuracy of the image. Unfortunately, a distortion measure that combines all three of these characteristics together has yet to be developed. Thus, it is common to design compression algorithms using distortion measures such as MSE, and then to use other subjective ratings and/or diagnostic accuracy measures to evaluate these algorithms [15]. This is the approach that we have used.

7.3.2 Subjective ratings

A randomized set of original and processed images is presented to a group of experts and typical users to be evaluated subjectively. The presentation to the evaluators can be done in different ways. One way is to show one image at a time and to ask the evaluator to rate the perceptual quality of the image, the quality of the abnormality that he/she usually looks for, or the quality of the image in terms of detecting these abnormalities. The evaluation is usually done either by giving a rating on a continuous scale or by choosing a distinctive descriptive phrase from a limited number of phrases provided by the test [16].

Another approach is to show two or more images in each presentation and to ask the evaluator to compare their quality. The comparison could be in terms of the perceptual quality of the image [35], the quality of the image for clinical purposes [67], or the preservation of a diagnostic feature, e.g., preserving abnormalities [11].

Researchers have attempted to link these subjective measures to objective distortion measures like SNR in order to incorporate them into the design of compression algorithms [15].

7.3.3 Diagnostic accuracy

Diagnostic accuracy is the most important measure discussed here because it is directly related to medical features in the image that are significant for clinical analysis. Measuring diagnostic accuracy depends on the diagnostic task. If the diagnostic task involves binary detection, e.g., detection of tumors or lung nodules, the most common method used is the receiver operating characteristic (ROC) analysis [1, 14, 55].

The ROC originated in signal detection theory [30]. The ROC curve plots the probability of detection (probability of detecting the signal when it is there) versus the probability of a false alarm (erroneously detecting a signal that is not there) as a threshold is varied. The degree by which the ROC curve departs from the equality line measures the distinctiveness between the two hypothesized models for generating the

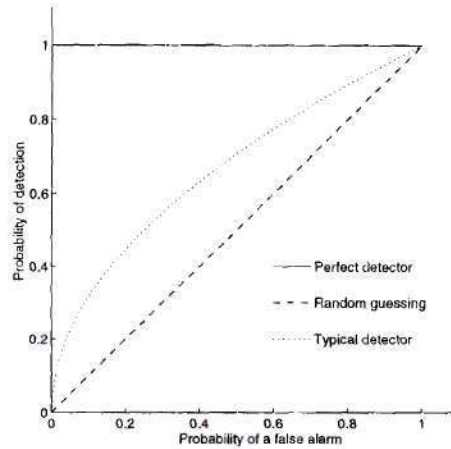


Figure 7.2: Receiver operating characteristic of a perfect detector, a detector based on random guessing, and a typical detector.

observations. A perfect detection technique is one that has a probability of detection of 1 for any false alarm rate including zero. The area under the ROC curve provides a scalar measure of the quality of the detection algorithm. A typical medical image quality test involves plotting the probability of detecting an abnormality versus the probability of false alarm while varying a threshold determined by the confidence of the reader in the reading according to a specified manner. Figure 7.2 shows the probability of detection versus the probability of false alarm for a perfect detector, typical detector, and a detector based on random guessing.

Cosman *et al.* [15, 16] describe an approach based on ROC analysis that applies to non-binary decisions. They defined sensitivity as the probability that something is detected given that it is present in a “gold standard” (defined by either the consensus of all evaluators, the evaluator’s evaluation of the original image, the agreement of an independent panel, or the actual result of the diagnosis). To penalize judges who label abnormalities everywhere, they defined the predictive value positive (PVP) as the chance that an abnormality is present given it is marked. They used the Behrens-Fisher t-statistic to evaluate the performance of different algorithms.

If the diagnostic task involves measuring a specific feature or attribute, such as the left ventricle ejection fraction for a gated blood study [51] or the size of blood

vessels [16], the diagnostic accuracy measure must reflect the difference between the measurement based on the original image and the measurement based on the compressed image.

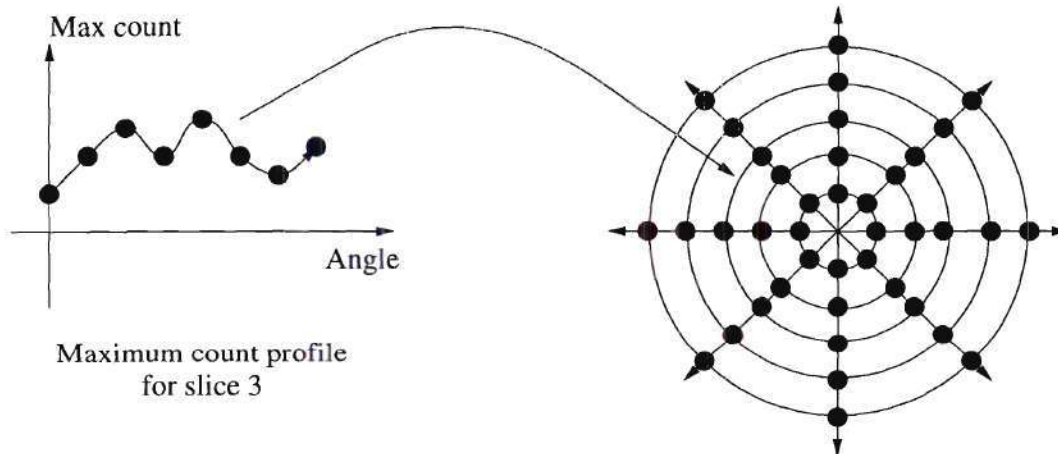
SPECT cardiac scans are usually compared against normal patients' scans when read by physicians. SPECT cardiac sequences are used to reconstruct an image of the heart. The reconstructed heart slices are used to generate a bull's-eye display, which is then used for making the comparisons [17]. The bull's-eye is a two-dimensional polar representation of the three-dimensional volume data from the reconstructed heart slices. It is generated by recording the maximum value encountered by rays originating from the center of the heart in each slice as a function of angle. Afterwards, the maximum count circumferential profiles for each slice are mapped onto a two-dimensional polar map. The distance from the center corresponds to the slice number, and the angular position to the angular position in the profile. The center of the bull's-eye corresponds to the apical slice and the periphery to the basal slice. Figure 7.3 shows the reconstructed slices of the sequence shown in Figure 7.1 and the corresponding bull's-eye. For example Figure 7.3 shows a maximum count profile for slice 3. The points of the profile are then mapped to the third circle of the bull's-eye display. Afterwards and for all slices, the points are interpolated to get the bull's-eye image shown in figure 7.3(d).

The different regions of the heart; apex, anterior, septal, inferior, and lateral, are then analyzed. This is done by dividing the bull's-eye into different regions, as shown in Figure 7.3(d). The apex corresponds to area 1, the anterior to areas 2 and 6, the septal to areas 3 and 7, the inferior to areas 4 and 8, and the lateral to areas 5 and 9. The average and the standard deviation of each region is computed. The ratios of the averages of the different regions are also computed.

Section 7.4 discusses segmented coding of SPECT cardiac sequences and presents two compression algorithms. The quality of the compressed sequences will be discussed using distortion measures, SNR, and diagnostic accuracy measures using the

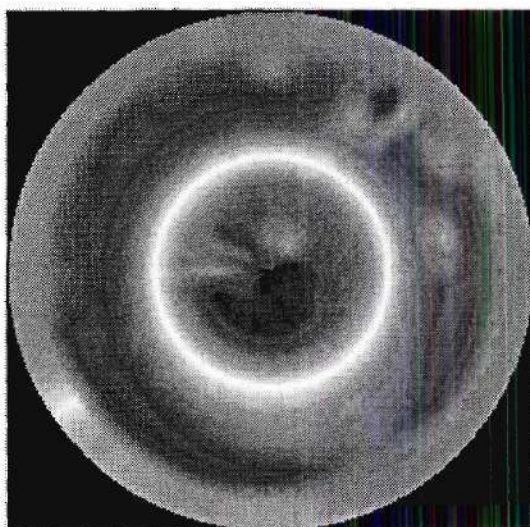


(a)

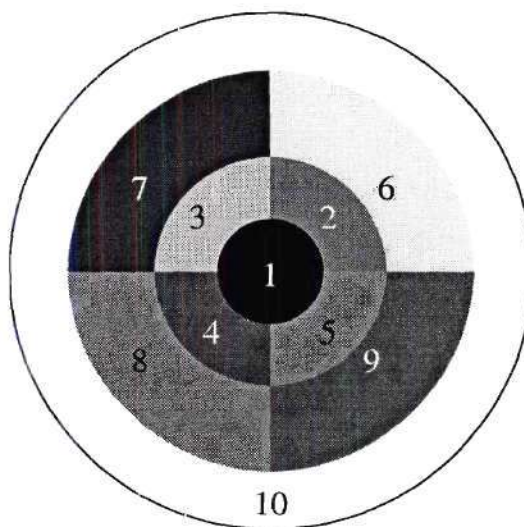


Maximum count profile
for slice 3

(b)



(c)



(d)

Figure 7.3: (a) The transaxial slices reconstructed for the sequence shown in Figure 7.1 and the corresponding bull's-eye. (b) The bull's-eye polar map. (c) The center of the bull's-eye is from the left-most (apical) slice. The periphery of the bull's-eye is from the right-most (basal) slice [17]. (d) The different regions in the bull's-eye. The apex corresponds to area 1, the anterior to areas 2 and 6, the septal to areas 3 and 7, the inferior to areas 4 and 8, and the lateral to areas 5 and 9.

bull's-eye approach.

7.4 Segmented Coding of SPECT Images

SPECT cardiac image sequences are projection images of the heart region taken at different angles, as shown in Figures 1.1 and 7.1. Since it is desired to preserve the diagnostic information in medical images and while compressing them to low bit rates, it is important to control the effect of quantization on the different regions of the image. For example, in cardiac SPECT image sequences, the heart, the blood vessels, and the other organs are very important for diagnosis. However, the body region is of much less importance while the background is of little importance. Taking this into account, we propose segmenting the cardiac images into these three different areas. Then, these areas will be quantized differently according to their importance. Since the region containing the organs is very important for medical diagnosis, it is not quantized at all, i.e., it is compressed without distortion. The body and the background regions are not important for diagnosis. Thus, they are quantized. The proposed approach is depicted in Figure 7.4.

An adaptive histogram-based algorithm is proposed to segment the cardiac SPECT image sequences. Morphological closing and opening is performed on the segmented regions to insure connectivity of the areas and to remove isolated small regions which usually correspond to noise. The segmentation procedure is summarized as follows:

1. The histogram of the sequence is computed and smoothed in order to compute initial gray-level thresholds (a , b , and max) for the three different areas, as shown in Figure 7.5(a). Initially, $b = 2 * max - a$.
2. The threshold of the background region is fixed and is set equal to a . The threshold for the heart region of each frame is initialized with the value b , i.e., $T_k = b$.

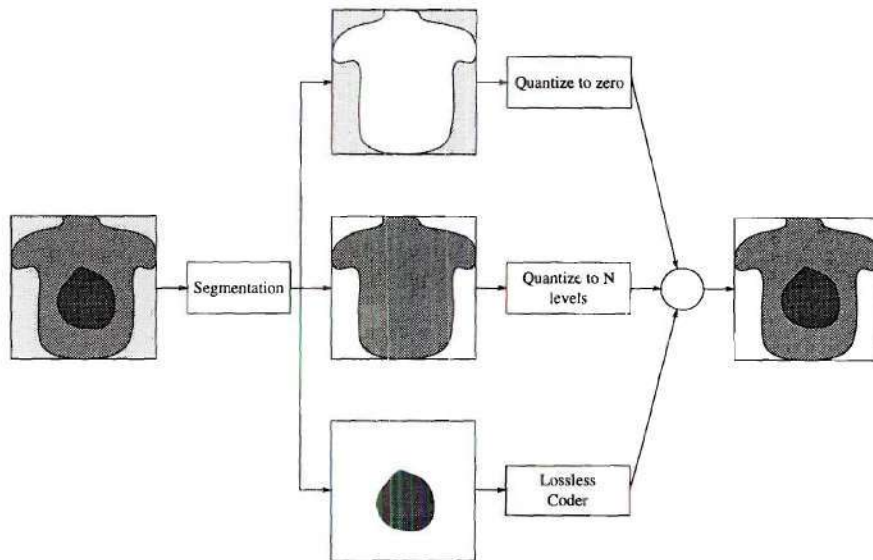
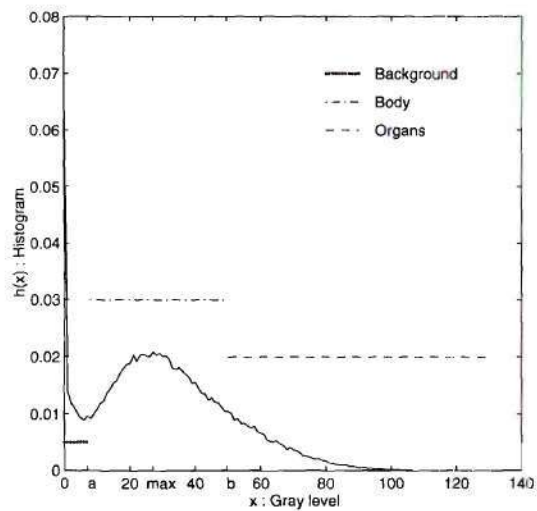
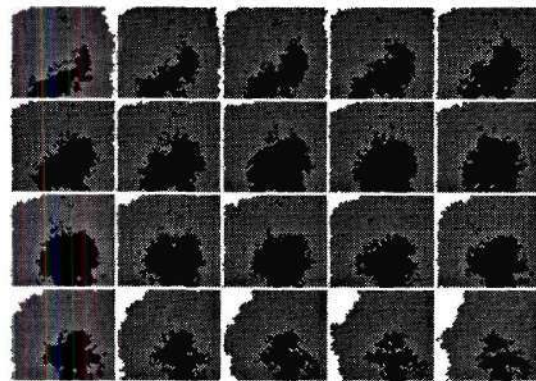


Figure 7.4: Content-based compression of SPECT cardiac sequences.



(a) Histogram



(b) Segmentation regions

Figure 7.5: (a) The histogram of the sequence and the gray level regions corresponding to the different regions in the sequence of Figure 7.1. (b) The segmented regions of the sequence.

3. The heart region is segmented in each frame and its center of mass is found. The horizontal component of the center of mass is modeled by

$$m_x(i) = \alpha + \beta \cos(\theta) + \gamma \sin(\theta), \quad (7.1)$$

where $m_x(i)$ is the center of mass of the heart of frame i in the x direction and θ is the projection angle. The center of mass of the heart in the y direction is assumed to be constant, since the patient is not moving. These assumptions are consistent with the data gathering process. The centers are then changed to the values generated by this model.

4. For $i = 1 \cdots N$, where N is number of frames in the sequence, the following operations are done:
 - (a) The average, $T_{i,k}$, of the pixel values in frame k contained in a small window centered about the center of mass of the heart of frame i is computed. This is done for all $i < k \leq N$.
 - (b) If $T_{i,k} < T_k$, then $T_k = \max(T_{i,k}, \max)$. Otherwise T_k is not changed.
5. Each frame is segmented according to the adjusted thresholds, T_k .
6. Morphological opening and closing is performed on the segmented regions to ensure connectivity and to remove isolated pixels. A 3×3 square structuring element is used.

The inner loop involving the computing of averages in step 4, is added to ensure that the heart is segmented properly in the frames where the images are severely attenuated. This occurs in the frames taken from the sides of the body where the attenuation is high and the liver counts are very high. These frames are the last frames in the sequence in our protocol.

Figure 7.1 shows the first 20 images of a $32 \times 64 \times 64$ 16-bit SPECT projection sequence. Figure 7.5(b) shows the result of segmenting this sequence into the different

regions. This simple segmentation algorithm was tested on thirty different SPECT sequences with different gray level contrasts and it performed sufficiently well on all these sequences.

The proposed coder is an information-progressive coder, i.e., the information can be sent progressively according to its importance for diagnosis. Moreover, each part of the sequence or piece of information can be coded using a resolution progressive coder. Thus, this approach can be progressive in both content and detail. The next section will use two different quantization schemes to code the background and body regions. Different compression algorithms can be developed by replacing the quantization step by a different coder.

7.4.1 Region-based scalar quantization of cardiac sequences

The first coder segments the sequence into the three different regions as shown in Figure 7.4. Afterwards, the background is replaced by zero gray level and the body region is quantized using a Lloyd-Max quantizer. The region containing the organs is not quantized. Afterwards, the resulting sequence is entropy coded.

7.4.2 Region-based vector quantization of cardiac sequences

Scalar quantization does not exploit the correlation between neighboring pixels. In order to incorporate this, the scalar quantizer is replaced by a vector quantizer. However, this involves rearranging the body region into $n \times n$ blocks, where n is the size of the vector. In order to reconstruct the image correctly at the decoder, we need to send the segmentation map along with the codewords. This is an extra burden but it is not that significant at medium bit rates, as will be demonstrated. The modified coder is depicted in Figure 7.6.

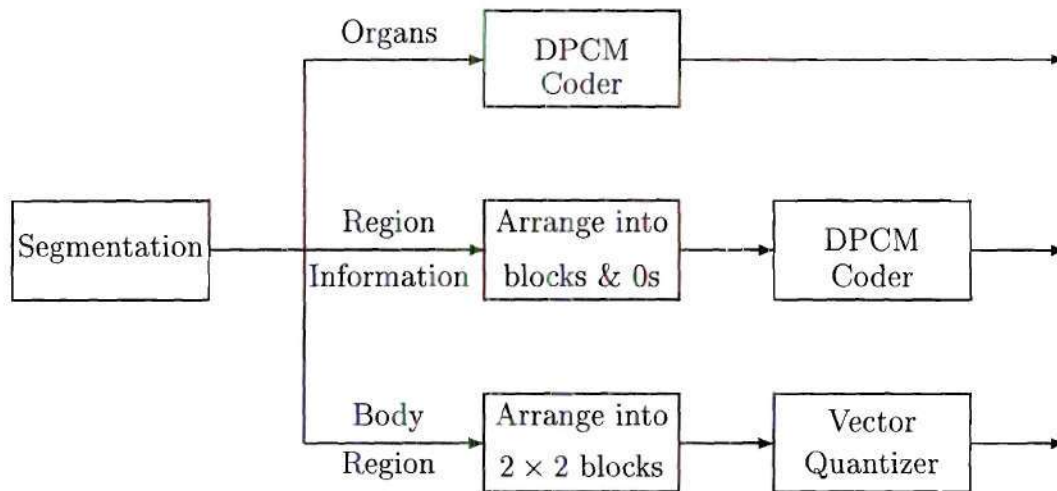


Figure 7.6: Region-based vector quantization of cardiac sequences.

7.5 Preprocessing of SPECT Cardiac Sequences

Since cardiac SPECT sequences are usually corrupted by data dependent Poisson noise, the results of Chapters 4 and 5 are used to improve the performance of the proposed coders. A three-dimensional neighborhood system along with the proposed clique potentials is used in the Markov random field model. These cardiac sequences consist of three regions that vary in detail; the organ region is very detailed, the body region contains less detail, and the background is constant. In order to incorporate this in the filtering, the image is segmented first. Afterwards, the background is set to zero gray level while the other two regions are filtered with two different δ 's. As shown in Chapter 4, a smaller δ results in more penalties for larger separations, thus a smoother image. Thus, for the organ region, which is more detailed, $\delta^2 = 250$, while for the body region, which is a smooth region, $\delta = 1.0$.

7.6 Example

Figure 7.7(c) shows the sequence shown in Figure 7.1 compressed using the region-based scalar quantization algorithm. The body region is quantized to eight gray levels and the entropy of the entire quantized sequence is 4.922 bpp. The percentages of the number of pixels of the different areas and the (zeroth-order) entropies of these areas are shown in Table 7.2 for the original sequence and four quantized versions of it. The four versions correspond to quantizing the gray levels of the body region into two, four, eight, and sixteen levels while quantizing the gray levels of the background region to zero, and not quantizing the region containing the organs at all.

Figure 7.7(e) shows the original sequence compressed using the region-based vector quantization algorithm, i.e., by replacing the background by zero gray level, vector quantizing 2×2 blocks of the body region with a codebook size of 8 vectors, and lossless DPCM coding of the organ region. Table 7.3 shows the bit rates of each region, the overall bit rate, and the SNR of the quantized sequence.

Table 7.2: The percentages of the pixels in the background, body, and organ regions of the *original* sequence and the quantized *original* sequence shown in Figure 7.7(a), respectively, and their corresponding entropies when using the SCALAR coder (The number of levels in the table is the number of levels quantizing the body region. The background is quantized to zero gray level).

Region	Area (%)	Zeroth-order entropy (bpp)				
		Original	16 levels	8 levels	4 levels	2 levels
Background	13.5%	2.479	0.000	0.000	0.000	0.000
Body	67.0%	5.574	3.947	2.975	1.943	0.878
Organs	19.5%	5.722	5.722	5.722	5.722	5.722
Overall	100.0%	6.124	4.922	4.285	3.610	2.896
		SNR (dB)				
		∞	20.177	19.111	15.387	9.387

Table 7.3: The bit rates of the background, body, and organ regions of the *original* sequence and the quantized *original* sequence shown in Figure 7.7(a) and their corresponding SNRs when using the VECTOR coder (The codebook size in the table is the number of vectors used to quantize the body region. The background is quantized to zero gray level).

Codebook Size	Bit Rate (bpp)				SNR (dB)
	Region Information	Organs	Body Region	Overall	
256	0.288	1.747	1.472	3.507	16.923
128	0.288	1.747	1.298	3.333	15.541
64	0.288	1.747	1.103	3.138	14.208
32	0.288	1.747	0.915	2.950	12.893
16	0.288	1.747	0.729	2.764	11.483
8	0.288	1.747	0.534	2.569	9.960

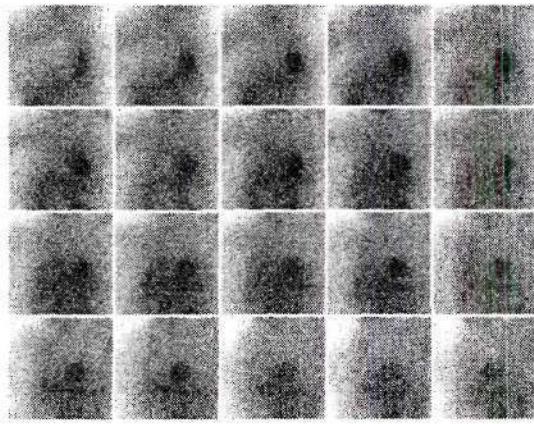
When comparing the vector coder with the scalar one, at around 2.9 bpp, the vector quantized sequence is 3.5 dB better than the scalar quantized sequence. Figure 7.8 shows the SNR versus bit rate curves when using the scalar and vector coders. The vector coder produces sequences with higher SNR than the ones produced by the scalar coder at the same bit rate. However, diagnostic accuracy measures should be used to measure how well the coders preserve the diagnostic information.

Studying the performance of these compression algorithms using the bull's-eye of the compressed sequences reflects the difference in diagnostic information between the compressed sequences and the original one. Figure 7.9 shows the bull's-eyes of the original sequence and three compressed versions of it using the scalar coder. Figure 7.10 shows the bull's-eyes of the original sequence and seven compressed versions of it using the vector coder.

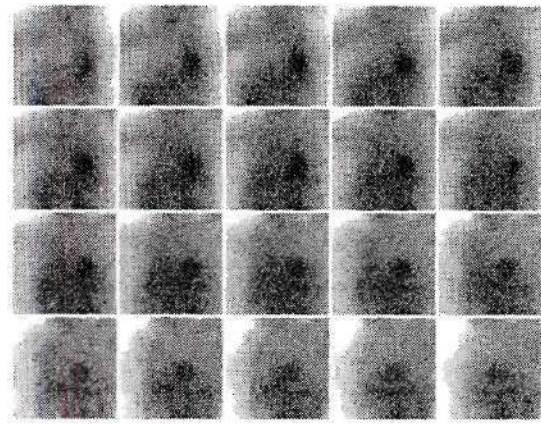
Tables 7.4 and 7.5 show the average count of the 10 different regions of the bull's-eye shown in Figure 7.3(d) when using scalar and vector quantization, respectively. The mean percentage difference ratios, $\frac{\text{original} - \text{compressed}}{\text{original}} * 100\%$, between the averages of the original sequence and the compressed sequence are 0.60%, 1.03%, 1.52%, and 1.97% when quantizing the body region to 16, 8, 4, and 2 levels, respectively. The mean percentage difference ratios are 0.55%, 0.64%, 1.10%, 0.87%, 2.35%, and 2.59% when vector quantizing the body region using 256, 128, 64, 32, 16, and 8 codebook sizes.

Tables 7.6 and 7.7 show the standard deviation of the counts of the 10 different regions of the bull's-eye when using the scalar and vector coders, respectively. The mean percentage difference ratios between the standard deviations of the original sequence and the compressed sequence are 8.25%, 7.09%, 9.32%, and 25.94% when quantizing the body region to 16, 8, 4, and 2 levels, respectively. The mean percentage difference ratios are 5.70%, 3.81%, 25.82%, 8.26%, 15.78%, and 15.78% when vector quantizing the body region using 256, 128, 64, 32, 16, and 8 codebook sizes.

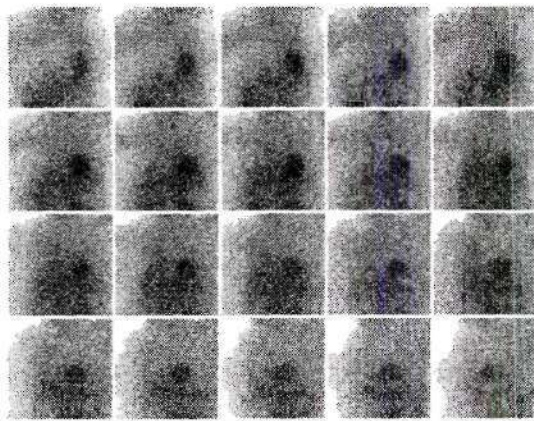
Tables 7.8 and 7.9 show the averages of the septal, lateral, anterior, and inferior



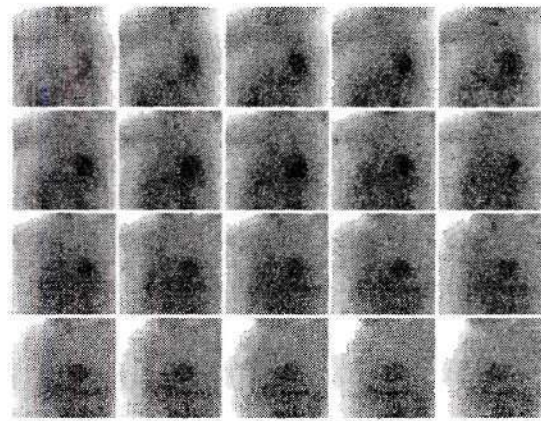
(a) *Original*



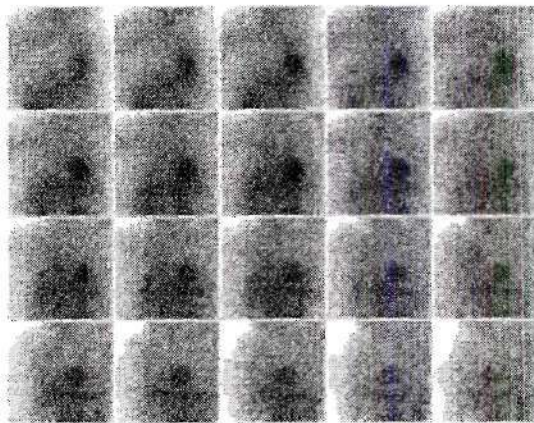
(b) *Filtered*



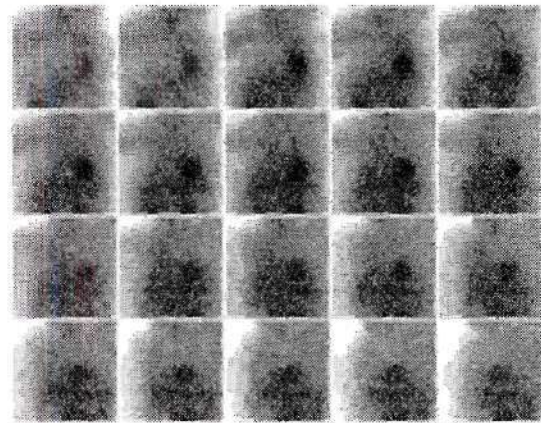
(c) SCALAR quantized *original*



(d) SCALAR quantized *filtered*



(e) VECTOR quantized *original*



(f) VECTOR quantized *filtered*

Figure 7.7: (a) The *original* first $20 \times 64 \times 64$ projection SPECT slices of a 32 slice sequence of a heart patient. (b) The *filtered* sequence using mean field theory and Markov models. (c) 8-level SCALAR quantized *original* sequence. (d) 8-level SCALAR quantized *filtered* sequence. (e) 8-vector codebook VECTOR quantized *original* sequence. (f) 8-vector codebook VECTOR quantized *filtered* sequence. (Gray levels are reversed for display.)

Table 7.4: The averages of the 10 regions of the *original* sequence when using the SCALAR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Original	83	103	99	115	112	125	125	118	123	112
16 levels	83	104	99	114	113	125	123	117	123	111
8 levels	83	104	100	114	113	124	122	116	122	111
4 levels	80	105	101	114	114	124	123	116	123	111
2 levels	84	98	98	115	107	124	120	115	122	112

Table 7.5: The averages of the 10 regions of the *original* sequence when using the VECTOR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Original	83	103	99	115	112	125	125	118	123	112
256	84	102	99	114	112	125	124	117	122	112
128	82	103	98	114	112	125	124	117	124	113
64	81	102	100	115	111	124	123	116	121	112
32	85	104	100	115	114	125	124	117	123	113
16	89	106	102	114	117	124	124	118	121	110
8	82	108	104	112	120	124	124	117	122	110

Table 7.6: The standard deviations of the 10 regions of the *original* sequence when using the SCALAR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Original	22.01	13.69	17.22	6.93	3.01	16.87	18.26	21.04	18.68	14.22
16 levels	21.75	14.70	18.06	6.60	3.90	17.77	20.47	23.45	19.38	13.97
8 levels	21.05	14.53	17.63	6.03	3.43	17.47	20.40	23.43	19.20	13.97
4 levels	22.82	13.65	16.22	6.55	2.57	18.94	21.52	24.86	20.80	13.68
2 levels	22.68	13.07	13.86	9.27	6.69	12.54	21.27	21.96	13.47	13.98

Table 7.7: The standard deviations of the 10 regions of the *original* sequence when using the VECTOR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Original	22.01	13.69	17.22	6.93	3.01	16.87	18.26	21.04	18.69	14.22
256	20.37	14.32	16.43	6.61	3.68	17.14	19.02	21.83	19.14	14.02
128	22.45	13.92	17.34	7.05	3.22	16.96	20.03	23.47	18.11	14.23
64	24.51	17.26	18.85	8.22	7.42	14.40	16.31	19.29	16.87	13.82
32	22.47	14.57	17.44	9.23	3.80	17.13	18.47	21.47	18.50	13.15
16	21.14	15.03	17.08	6.89	1.38	13.88	14.53	17.14	15.65	12.01
8	25.23	9.62	10.73	9.49	5.96	14.64	17.13	19.45	16.80	11.65

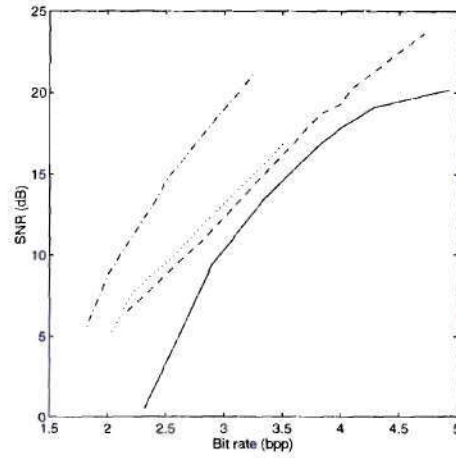


Figure 7.8: Comparing region-based SCALAR and VECTOR quantization approaches. The solid and dotted lines are for the *original* sequence when using the SCALAR and VECTOR coders, respectively. The dashed and dash-dotted lines are for the *filtered* sequence when using the SCALAR and VECTOR coders, respectively.

regions. It also shows the ratios between these different regions. The percentage of the difference ratios of the averages is less than 4% while it is less than 5% for the region ratios.

7.6.1 Effect of preprocessing

Preprocessing is expected to improve the performance of the proposed approaches. Figure 7.7(b) shows the result of the restoration. Figure 7.7(d) shows the filtered sequence compressed using the scalar coder. The body region is quantized to eight gray levels and the entropy of the entire quantized sequence is 4.104 bpp. Table 7.10 shows the entropies of the filtered sequences and four quantized versions of it. It is important to note that the SNR is measured with respect to the filtered sequence. At around 2.9 bpp, the quantized filtered sequence was 1.8 dB better than the quantized original sequence.

Figure 7.7(f) shows the filtered sequence compressed using the region-based vector quantization algorithm, i.e., by replacing the background by zero gray level and vector quantizing the body region with a codebook size of 8 vectors. Table 7.11

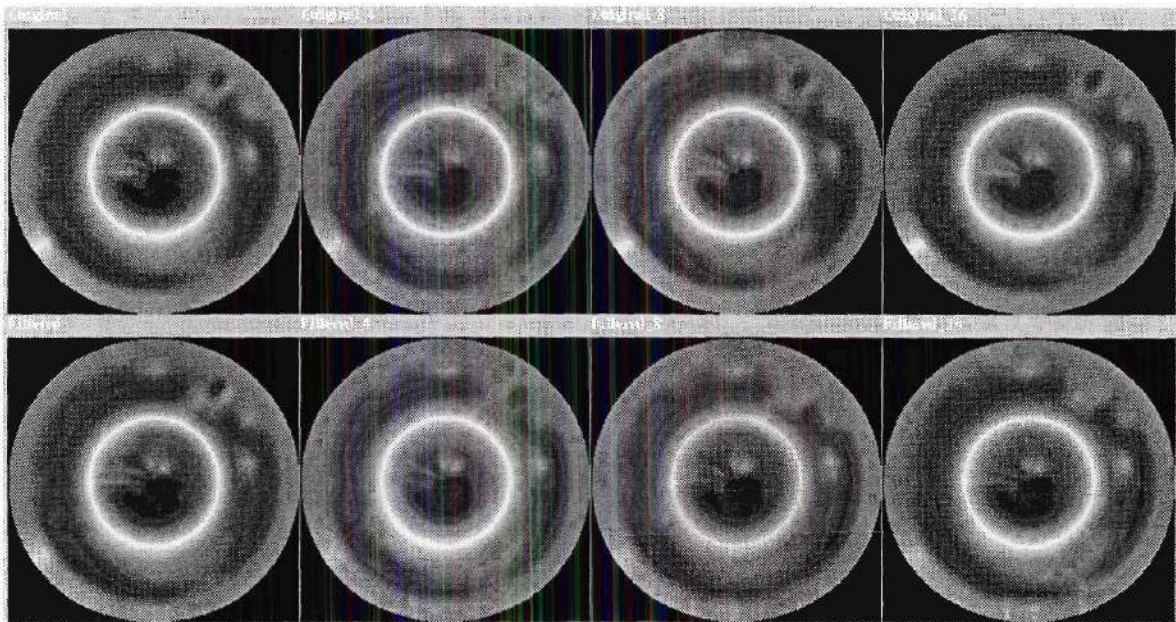


Figure 7.9: The bull's-eyes when using SCALAR quantization. The top row shows from left to right the bull's-eyes for the *original* sequence, the *original* sequence with the body region quantized to 4, 8, 16 levels, respectively. The bottom row shows from left to right the bull's-eyes for the *filtered* sequence, the *filtered* sequence with the body region quantized to 4, 8, 16 levels, respectively.

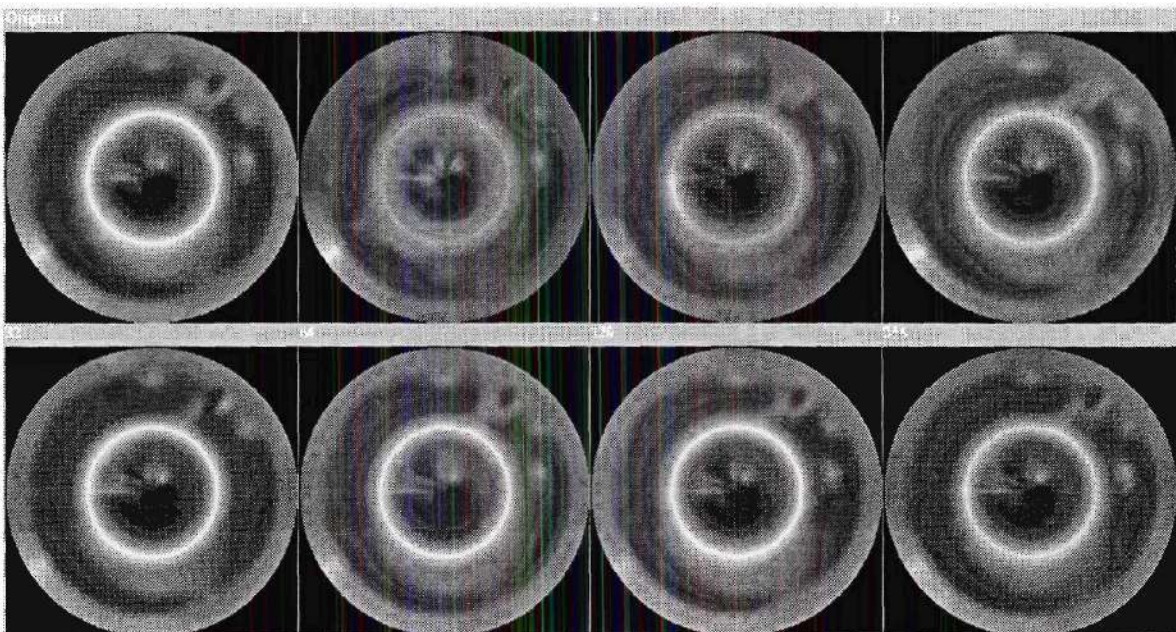


Figure 7.10: The bull's-eyes when using the VECTOR coder. The bull's-eyes are from top to bottom left to right for the *original* sequence and the *original* sequence with the body region quantized using 1, 4, 16, 32, 64, 128, and 256 codebook sizes.

Table 7.8: The averages of the septal (S), lateral (L), anterior (A), and inferior (I) regions and the corresponding ratios of the *original* sequence when using the SCALAR coder.

Sequence	S	L	A	I	S/L	A/L	I/L	A/I
Original	224	235	228	233	0.953	0.970	0.991	0.979
16 levels	222	236	229	231	0.941	0.970	0.979	0.991
8 levels	222	235	228	230	0.945	0.970	0.979	0.991
4 levels	224	237	229	230	0.945	0.966	0.970	0.996
2 levels	218	229	222	230	0.952	0.969	1.004	0.965

Table 7.9: The averages of the septal (S), lateral (L), anterior (A), and inferior (I) regions and the corresponding ratios of the *original* sequence when using the VECTOR coder.

Sequence	S	L	A	I	S/L	A/L	I/L	A/I
Original	224	235	228	233	0.953	0.970	0.991	0.979
256	223	234	227	231	0.953	0.970	0.987	0.983
128	222	236	228	231	0.941	0.966	0.979	0.987
64	223	232	226	231	0.961	0.974	0.996	0.978
32	224	237	229	232	0.945	0.966	0.979	0.987
16	226	238	230	232	0.950	0.966	0.975	0.991
8	228	242	232	229	0.942	0.959	0.946	1.013

Table 7.10: The percentages of the pixels in the background, body, and organ regions of the *filtered* sequence and the quantized *filtered* sequence shown in Figure 7.7(c), respectively, and their corresponding entropies when using the SCALAR coder (The number of levels in the table is the number of levels quantizing the body region. The background is quantized to zero gray level).

Region	Area (%)	Entropy (bpp)				
		Original	16 levels	8 levels	4 levels	2 levels
Background	13.7%	0.905	0.000	0.000	0.000	0.000
Body	68.9%	5.404	3.775	2.847	1.931	0.994
Organs	17.4%	5.559	5.559	5.559	5.559	5.559
Overall	100.0%	5.843	4.708	4.104	3.487	2.852
		SNR (dB)				
		∞	23.585	20.320	16.094	11.198

Table 7.11: The bit rates of the background, body, and organ regions of the *filtered* sequence and the quantized filtered sequence shown in Figure 7.7(b) and their corresponding SNRs when using the VECTOR coder (The codebook size in the table is the number of vectors used to quantize the body region. The background is quantized to zero gray level).

Codebook Size	Bit Rate (bpp)				SNR (dB)
	Region Information	Organs	Body Region	Overall	
256	0.284	1.539	1.441	2.942	21.241
128	0.284	1.539	1.238	2.732	19.511
64	0.284	1.539	1.052	2.557	17.943
32	0.284	1.539	0.878	2.381	16.400
16	0.284	1.539	0.699	2.191	14.806
8	0.284	1.539	0.540	2.059	12.830

shows the bit rates of each region, the overall bit rate, and the SNR of the quantized sequence.

When comparing the compression of the original sequence and the filtered sequence, at around 2.9 bpp the quantized filtered sequence is 8.3 dB better than the quantized original sequence. Moreover, when comparing the vector coder with the scalar one, at around 2.9 bpp, the vector quantized filtered sequence is 10.1 dB better than the scalar quantized filtered sequence and 11.9 dB better than the scalar quantized original sequence. Figure 7.8 shows the SNR versus bit rate curves when using the scalar and vector coders for both the scalar and vector coders. The vector coder outperforms the scalar one and the filtered sequence has better SNR versus bit rate curves than the original one.

Figure 7.9 shows the bull's-eyes of the filtered sequence and four compressed versions of it using the scalar coder. Figure 7.11 shows the bull's-eyes of the filtered sequence and seven compressed versions of it using the vector coder.

Tables 7.12 and 7.13 show the average count of the 10 different regions of the bull's-eye when using scalar and vector quantization, respectively. The mean percentage difference ratio between the averages of the original and those of the

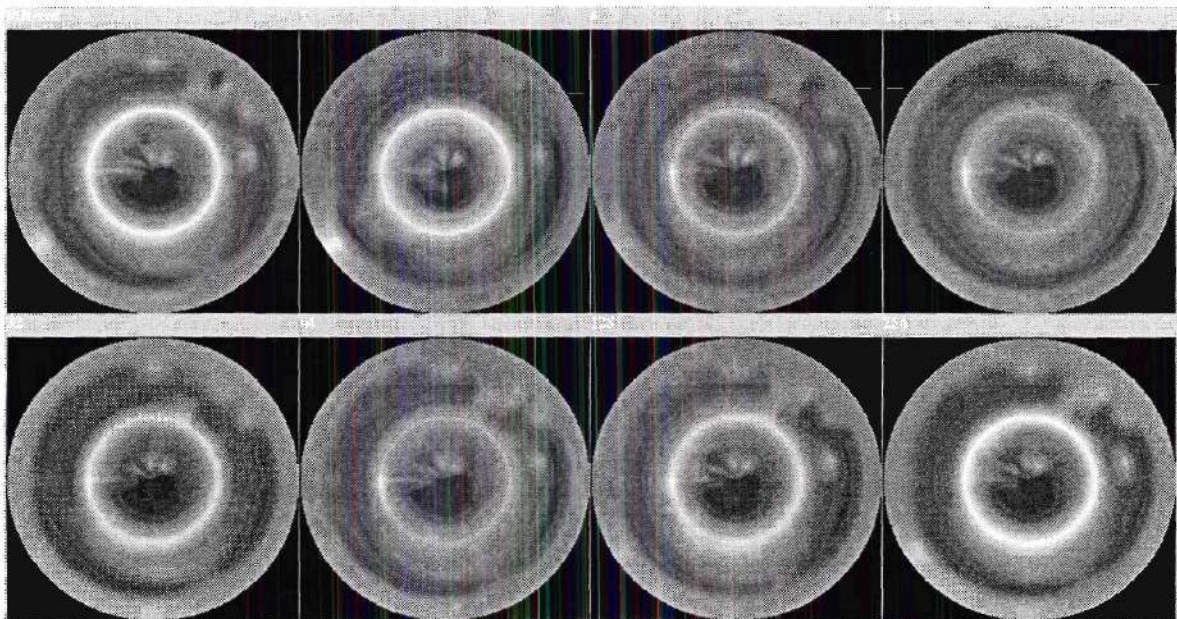


Figure 7.11: The bull's-eyes when using the VECTOR coder. The bull's-eyes are from top to bottom left to right for the *filtered* sequence and the *filtered* sequence with the body region quantized using 1, 4, 16, 32, 64, 128, and 256 codebook sizes.

filtered sequence is 0.56%, which is very small. Tables 7.14 and 7.15 show the standard deviations of the 10 different regions for the filtered sequence. The mean percentage difference ratio between the standard deviations of the original and the filtered is 9.45%. Table 7.16 and 7.17 show the averages of the four main areas and the ratios of their averages for the filtered sequence. The difference between the original and filtered sequence is small. The percentage difference ratio is less than 0.5% for the averages and the ratios.

The mean percentage difference ratios between the averages of the filtered sequence and the compressed filtered sequence are 1.94%, 1.51%, 2.06%, and 1.99% when quantizing the body region to 16, 8, 4, and 2 levels, respectively. The mean percentage difference ratios are 2.64%, 2.15%, 2.64%, 2.24%, 2.83%, 1.27%, and 2.10% when vector quantizing the body region using 256, 128, 64, 32, 16, and 8 codebook sizes. The mean percentage difference ratios between the standard deviations of the filtered sequence and the compressed filtered sequence are 6.48%, 8.83%, 26.19%, and 31.52% when quantizing the body region to 16, 8, 4, and 2 levels, respectively. The mean percentage difference ratios are 7.01%, 22.98%, 18.77%, 13.57%, 11.75%, 11.84%, and 12.37% when vector quantizing the body region using 256, 128, 64, 32, 16, and 8 codebook sizes. The percentage of the difference ratios of the area ratios is less than 3%.

The preservation of the diagnostic information was measured using the bull's-eye display approach. Although, the image has been segmented, filtered with different δ 's for the different regions, and compressed, the averages of the different regions and the ratios between the different regions remained within almost 5% of the averages of the original image. The standard deviations of the decompressed sequences remained close to those of the original sequence. This suggests that the system did not lose the diagnostic information in the sequence. A better test is a complete ROC analysis, which involves a large number of doctors who will subjectively evaluate the performance of the system.

Table 7.12: The averages of the 10 regions of the *filtered* sequence when using the SCALAR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Filtered	84	102	99	116	112	126	125	118	124	111
16 levels	80	101	97	113	111	123	123	117	121	110
8 levels	82	101	98	113	111	123	123	117	121	111
4 levels	77	104	100	114	113	124	124	117	122	109
2 levels	86	105	105	117	115	125	127	119	123	112

Table 7.13: The averages of the 10 regions of the *filtered* sequence when using the VECTOR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Filtered	84	102	99	116	112	126	125	118	124	111
256	80	101	98	114	110	123	122	116	121	109
128	81	102	99	114	111	124	124	117	121	110
64	79	98	95	113	107	123	124	117	121	110
32	80	104	102	113	115	123	125	118	120	109
16	79	103	101	109	114	121	123	117	120	111
8	78	102	100	113	112	122	124	117	120	108

Table 7.14: The standard deviations of the 10 regions of the *filtered* sequence when using the SCALAR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Filtered	21.50	14.34	16.67	6.93	3.37	17.13	18.87	21.74	18.50	13.16
16 levels	21.31	15.55	18.06	6.69	4.26	16.40	18.07	20.40	18.08	12.88
8 levels	21.04	15.20	17.22	6.28	4.98	16.60	17.45	20.13	18.23	12.83
4 levels	21.90	18.93	21.57	9.10	8.64	17.40	19.26	23.08	18.72	12.17
2 levels	20.42	20.38	19.79	8.52	10.43	17.82	17.77	21.74	19.52	11.73

Table 7.15: The standard deviations of the 10 regions of the *filtered* sequence when using the VECTOR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Filtered	21.50	14.34	16.67	6.93	3.37	17.13	18.87	21.74	18.50	13.16
256	23.08	15.26	17.05	7.78	4.99	15.06	16.04	19.28	16.87	11.23
128	21.97	14.88	16.63	7.52	5.26	15.34	15.84	18.86	17.05	12.73
64	22.82	13.81	15.44	5.24	3.51	14.51	14.53	17.55	15.88	12.21
32	21.57	16.38	18.22	7.52	6.13	16.80	17.22	19.98	19.00	11.92
16	22.68	15.55	17.28	11.40	4.32	13.70	16.04	17.22	14.50	12.19
8	23.10	17.26	18.56	9.04	7.16	14.61	17.20	18.75	16.60	10.96

Table 7.16: The averages of the septal (S), lateral (L), anterior (A), and inferior (I) regions and the corresponding ratios of the *filtered* sequence when using the SCALAR coder.

Sequence	S	L	A	I	S/L	A/L	I/L	A/I
Filtered	224	236	228	234	0.949	0.966	0.992	0.974
16 levels	220	232	224	230	0.948	0.966	0.991	0.974
8 levels	221	232	224	230	0.953	0.966	0.991	0.974
4 levels	224	235	228	231	0.953	0.970	0.983	0.987
2 levels	232	238	230	236	0.975	0.966	0.992	0.975

Table 7.17: The averages of the septal (S), lateral (L), anterior (A), and inferior (I) regions and the corresponding ratios of the *filtered* sequence when using the VECTOR coder.

Sequence	S	L	A	I	S/L	A/L	I/L	A/I
Filtered	224	236	228	234	0.949	0.966	0.992	0.974
256	220	231	224	230	0.952	0.970	0.996	0.974
128	223	232	226	231	0.961	0.974	0.996	0.978
64	219	228	221	230	0.961	0.969	1.009	0.961
32	227	235	227	231	0.966	0.966	0.983	0.983
16	224	234	224	226	0.957	0.957	0.966	0.991
8	224	232	224	230	0.966	0.966	0.991	0.974

CHAPTER 8

SUMMARY AND SUGGESTIONS FOR FURTHER RESEARCH

8.1 Summary

The effect of noise on image compression was studied using the JPEG lossy image compression standard and Lloyd-Max quantization. Noise severely degrades the performance of these compression algorithms. It was noticed that at high and medium-high bit rates the quality of the compressed image did not change by increasing the bit rate. Moreover, at medium to low bit rates, the quality of the compressed image, in most examples used, improved by increasing the degree of quantization. At low bit rates, the coders filtered much of the noise and the quality of the compressed noisy images was closer to that of the noise-free images, than to the noisy inputs to the coder.

The results of noisy source coding theory were used to compress images that were corrupted by data-dependent Poisson noise and by film-grain noise. The distortion was measured in terms of the MSE. The restoration-then-compression approach, which is optimal in the MSE sense, resulted in large gains at high and high-medium bit rates. This approach was also used with the lossy JPEG image compression standard and with EPIC, which is a pyramid coder. Although the restoration-then-compression approach is not optimal for the case of JPEG and EPIC, the results were similar to those of the Lloyd-Max quantizer. Moreover, at medium bit rates,

the quality of the compressed noisy images using JPEG, EPIC, or VQ improves in the PSNR sense by decreasing the bit rate. This is because the coder is removing more noise than is introduced as quantization artifacts. When the image is filtered, the PSNR is monotonic in the bit rate, which is what we would expect.

A restoration algorithm for images that are corrupted by noise and compressed using a block-transform coder was also developed. The postprocessing algorithm results in an approximation to the MMSE solution. Two types of noise images were considered: images that are corrupted by data-dependent Poisson noise and images that are corrupted by film-grain noise. The noise-free case was also considered. Two approaches, which are based on modeling the image as a Markov random field, have been developed. The first models the image by an intensity field while the second models it as an intensity field and a line field. Both approaches improved the quality of the compressed noisy images in terms of both perceptual quality and PSNR. However, the first approach was faster and it gave better results. Using a line field increases the number of computations of each iteration. In addition, the clique potential used in the thesis balances the penalty for sharp edges and the reward for smoothness at the edges; thus the line field is an extra burden. Moreover, for the noise-free case, the gain in the PSNR of the first approach is slightly higher than the gains presented in [47] at almost the same bit rate. In [47], at a compression ratio of 30:1, the reduction in the MSE is 0.65 while using the first approach at a compression ratio of 30.05:1, the reduction in the MSE is 0.98.

The restoration-then-compression approach was then applied to code SPECT cardiac sequences. This involved developing a content-based image compression algorithm in order to preserve the diagnostic information in the sequence. The performance of the proposed algorithms was measured using objective measures, MSE, and a diagnostic accuracy measure based on the bull's-eye display. The proposed algorithms preserved the diagnostic information in the sequences while compressing them significantly.

8.2 Novel Accomplishments of the Thesis

- The effect of noise on compression was studied. Noisy source coding theory was extended to data-dependent Poisson noise and film-grain noise. This involved developing a MMSE restoration algorithm as a preprocessing stage in the coder.
- A preprocessing algorithm to be used for encoding images corrupted by data dependent Poisson noise was developed and was applied to synthetic and real images.
- A preprocessing algorithm for estimating images corrupted by data dependent film-grain noise was developed and was applied to synthetic and real images.
- A postprocessing algorithm for estimating noise-free and noisy images compressed using block-transform coders was developed. The postprocessing approach was implemented for the noise-free case, the Poisson noise case, and the film-grain noise case.
- Since medical images are usually corrupted by Poisson noise, the preprocessing approach was applied to cardiac SPECT sequences to improve the performance of different coders.
- A content-based coder was developed to compress cardiac SPECT sequences. The quality of the compressed sequences was measured using MSE and a diagnostic accuracy measure based on the bull's-eye display.

8.3 Suggestions for Further Research

Images are often blurred and corrupted by noise. The effect of degradations upon the formation of images on compression should be studied. If the effect of such degradations severely degrades the performance of image compression algorithms, the

results of the thesis can be applied to enhance the performance of the compression algorithms.

Postprocessing of lossy compressed images was effective in improving both the PSNR and the visual quality of the image. The same approach can be extended to postprocess images compressed using subband coders, which at low bit rates add ringing artifacts to the image.

The proposed algorithms to compress SPECT cardiac sequences are content progressive, i.e., the information can be sent according to its content. It can also be modified to be resolution progressive as well. This can be done by changing the quantization to be a resolution progressive coder.

APPENDIX A

DEVELOPMENT OF THE RESTORATION ALGORITHMS

The conditional probability of the observed image given the original image is an independent identically distributed pdf, i.e.,

$$p(\mathbf{x}|\mathbf{f}) = \prod_{i,j} p(x(i,j)|\mathbf{f}), \quad (\text{A.1})$$

where $x(i,j)$ is the value of \mathbf{x} at pixel (i,j) . The conditional probability of the original image given the observed image, $p(\mathbf{f}|\mathbf{x})$, can be derived using Bayes' formula

$$\begin{aligned} p(\mathbf{f}|\mathbf{x}) &= \frac{\exp[-\beta U(\mathbf{f})] \prod_{i,j} p(x(i,j)|\mathbf{f})}{Z p(\mathbf{x})} \\ &= \hat{Z}^{-1} \exp[-\beta U(\mathbf{f}|\mathbf{x})], \end{aligned} \quad (\text{A.2})$$

where

$$U(\mathbf{f}|\mathbf{x}) = U(\mathbf{f}) + \sum_{i,j} \frac{1}{\beta} \ln p(x(i,j)|\mathbf{f}). \quad (\text{A.3})$$

Applying the mean-field theory approximation results in

$$\hat{U}(f(i,j)|\mathbf{x}) = \hat{U}(f(i,j)) + \frac{1}{\beta} \ln p(x(i,j)|\mathbf{f}), \quad (\text{A.4})$$

where $\hat{U}(f(i,j)) = U(\dots, E\{f(m,n)\}, \dots, f(i,j), \dots, E\{f(k,l)\}, \dots)$ such that $(k,l) \neq (i,j) \neq (m,n)$. The MMSE estimate becomes

$$f^{mf}(i,j) = E\{f(i,j)|\mathbf{x}\} = \frac{\sum_{f(i,j)} f(i,j) \exp[-\beta \hat{U}(f(i,j)|\mathbf{x})]}{\sum_{f(i,j)} \exp[-\beta \hat{U}(f(i,j)|\mathbf{x})]}. \quad (\text{A.5})$$

Since both summations are over all possible values of $f(i, j)$, only the part of $\hat{U}(\cdot)$ that includes $f(i, j)$ remains inside both summations and the other parts cancel each other resulting in

$$f^{mf}(i, j) = \frac{\sum_{f(i, j)} f(i, j) \exp [-U_{(i, j)}^{mf}(f(i, j)|\mathbf{x})]}{\sum_{f(i, j)} \exp [-U_{(i, j)}^{mf}(f(i, j)|\mathbf{x})]}, \quad (\text{A.6})$$

where $U_{(i, j)}^{mf}(\cdot)$, the mean-field local energy, is the part of $U(\cdot)$ that depends on $f(i, j)$ while the influence of other pixels is approximated by their means. For a first-order 2-dimensional neighborhood system

$$U_{(i, j)}^{mf}(f(i, j)) = 2 \{V_c(f(i, j), E\{f(i-1, j)\}) + V_c(f(i, j), E\{f(i+1, j)\}) + V_c(f(i, j), E\{f(i, j-1)\}) + V_c(f(i, j), E\{f(i, j+1)\})\}. \quad (\text{A.7})$$

The saddle point approximation implies solving for the minimum of the energy function. Using the gradient descent algorithm to solve for the minimum of the energy function results in the following set of iterative equations

$$\bar{f}^{t+1}(i, j) = \bar{f}^t(i, j) - \omega \frac{\partial U_{(i, j)}^{mf}(f(i, j)|\mathbf{x})}{\partial f(i, j)}, \quad (\text{A.8})$$

where $\bar{f}^t(m, n)$ is the estimate of the mean-value of $f(m, n)$ at iteration t , and ω is the step size. For the Poisson noise case, these equations are

$$\begin{aligned} \bar{f}^{t+1}(i, j) = \bar{f}^t(i, j) - \omega \left\{ \frac{1}{\beta} \left[1 - \frac{x(i, j)}{\bar{f}^t(i, j)} \right] \right. \\ \left. + \sum_{(k, l) \in N(i, j)} \left[\frac{\partial V_f(\bar{f}^t(i, j), \bar{f}^t(k, l))}{\partial f(i, j)} \right] \right\}. \end{aligned} \quad (\text{A.9})$$

For the film-grain noise case, the iterative equations are

$$\begin{aligned} \bar{f}^{t+1}(i, j) = \bar{f}^t(i, j) - \omega \left\{ \frac{\gamma}{\beta \bar{f}^t(i, j)} - \frac{\left[1 - \gamma + \frac{\gamma x(i, j)}{\bar{f}^t(i, j)} \right] [x(i, j) - \bar{f}^t(i, j)]}{\beta \gamma \alpha^2 [\bar{f}^t(i, j)]^{2\gamma}} \right. \\ \left. + \sum_{(k, l) \in N(i, j)} \frac{\partial V_f(\bar{f}^t(i, j), \bar{f}^t(k, l))}{\partial f(i, j)} \right\}, \end{aligned} \quad (\text{A.10})$$

From (4.9) the partial derivative is seen to be

$$\frac{\partial V_{\mathbf{f}}(f(i, j), f(k, l))}{\partial f(i, j)} = \frac{(2\delta^2 - 2)(f(i, j) - f(k, l)) + 2(f(i, j) - f(k, l))^3}{[\delta^2 + (f(i, j) - f(k, l))^2]^2}. \quad (\text{A.11})$$

APPENDIX B

GENERALIZED GAUSSIAN DISTRIBUTION

The probability density function of a generalized-Gaussian distributed random variable x with mean m and variance σ^2 is [60, 31]

$$p(x) = a \exp[-|b(x - m)|^c], \quad (\text{B.1})$$

where

$$a = \frac{bc}{2\Gamma(\frac{1}{c})}, \quad (\text{B.2})$$

$$b = \frac{1}{\sigma} \sqrt{\frac{\Gamma(\frac{3}{c})}{\Gamma(\frac{1}{c})}}, \quad (\text{B.3})$$

c is the shape factor, and $\Gamma(a)$ is the Gamma function defined as

$$\Gamma(a) = \int_0^\infty x^{a-1} \exp(-x) dx. \quad (\text{B.4})$$

In the rest of the appendix, we'll assume, without loss of generality, that $m = 0$.

The r^{th} order moment in the interval $[t_1, t_2]$ is

$$\mu_r = \int_{t_1}^{t_2} x^r p(x) dx = \mu(t_2, r, c, \sigma) - \mu(t_1, r, c, \sigma), \quad (\text{B.5})$$

where

$$\mu(t, r, c, \sigma) = \int_0^{|t|} x^r p(x) dx. \quad (\text{B.6})$$

The r^{th} order absolute moment in the interval $[t_1, t_2]$ is

$$\mu'_r = \int_{t_1}^{t_2} |x|^r p(x) dx = |\mu(t_2, r, c, \sigma)| - |\mu(t_1, r, c, \sigma)|. \quad (\text{B.7})$$

The “complete” absolute r^{th} moment of the generalized Gaussian is [31]

$$m_r = \int_{-\infty}^{\infty} |x|^r p(x) dx = \mu(\infty, r, c, \sigma) - \mu(-\infty, r, c, \sigma) \quad (\text{B.8})$$

$$= \frac{\Gamma(\frac{r+1}{c})}{\Gamma(\frac{1}{c})} b^{-r}. \quad (\text{B.9})$$

If we define Ψ as

$$\Psi = \frac{m_2}{(m_1)^2} \quad (\text{B.10})$$

$$= \frac{\Gamma(\frac{3}{c})\Gamma(\frac{1}{c})}{(\Gamma(\frac{2}{c}))^2} \quad (\text{B.11})$$

and noting that Ψ is independent of the data, hence it can be computed and stored, Ψ can be used to estimate the shape factor c after estimating the second and second absolute moments. The first absolute moment for an N point data vector can be estimated by

$$\hat{m}_1 = \frac{1}{N} = \sum_{n=0}^{N-1} |x_n|, \quad (\text{B.12})$$

and the second absolute moment can be estimated by

$$\hat{m}_2 = \frac{1}{N} = \sum_{n=0}^{N-1} (x_n)^2 = \hat{\sigma}^2, \quad (\text{B.13})$$

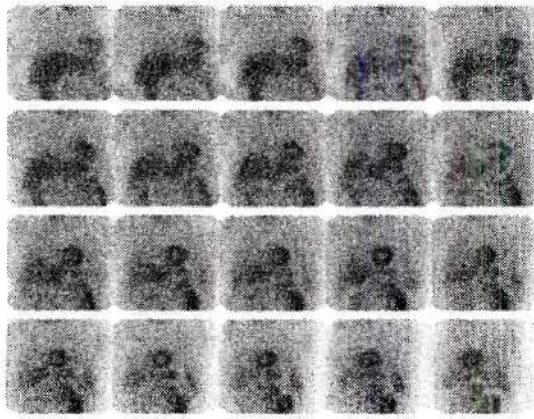
where $\hat{\sigma}^2$ is an estimate of the variance. Equation B.3 is then used to estimate b , afterwards, Equation B.3 is used to estimate a .

APPENDIX C

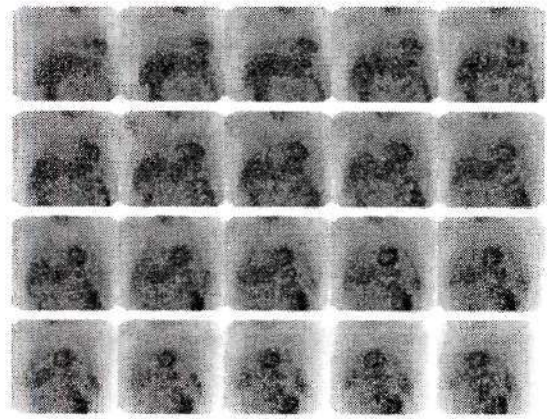
EXAMPLES OF CODING CARDIAC SEQUENCES

Myocardial perfusion imaging is usually performed in conjunction with physical exercise [71]. The images obtained shortly after exercise are compared with the images taken at rest. Such comparison shows the change in blood flow due to coronary stenosis. This appendix studies the performance of the cardiac SPECT compression algorithms proposed in Chapter 7 on a complete rest and stress study. Section C.1 discusses the rest case and Section C.2 studies the stress case. These studies are organized the same as the study discussed in Section 7.6.

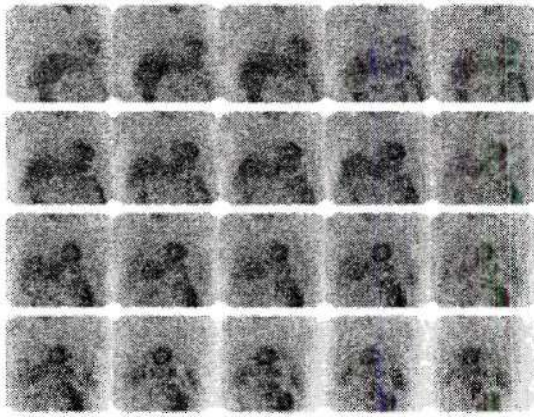
C.1 Rest Example



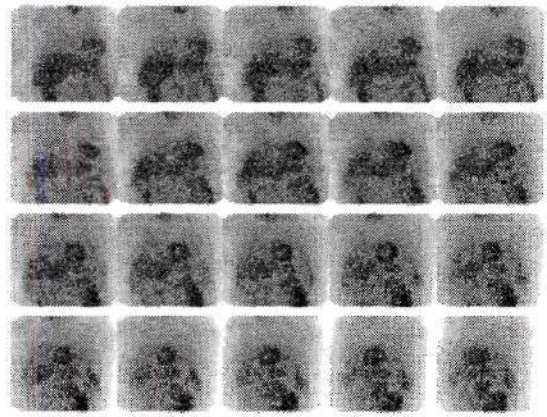
(a) *Original*



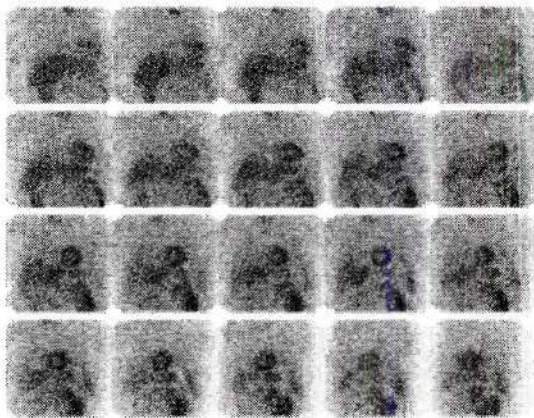
(b) *Filtered*



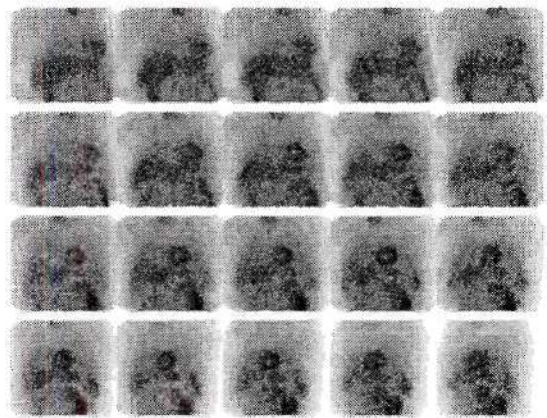
(c) SCALAR quantized *original*



(d) SCALAR quantized *filtered*



(e) VECTOR quantized *original*



(f) VECTOR quantized *filtered*

Figure C.1: REST EXAMPLE: (a) The *original* sequence. (b) The *filtered* sequence. (c) 8-level SCALAR quantized *original* sequence. (d) 8-level SCALAR quantized *filtered* sequence. (e) 8-vector VECTOR quantized *original* sequence. (f) 8-vector VECTOR quantized *filtered* sequence.

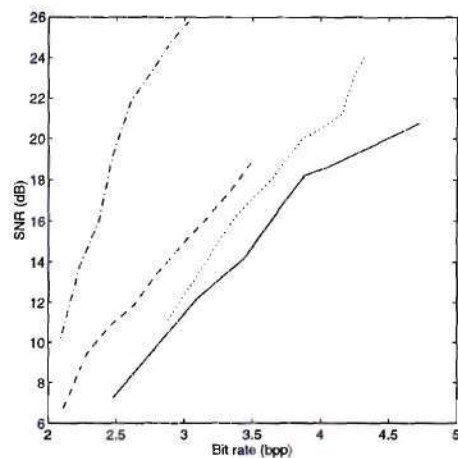


Figure C.2: REST EXAMPLE: Comparing Region-based SCALAR and VECTOR quantization approaches. The solid and dotted lines are for the *original* sequence when using the SCALAR and VECTOR coders, respectively. The dashed and dash-dotted lines are for the *filtered* sequence when using the SCALAR and VECTOR coders, respectively.

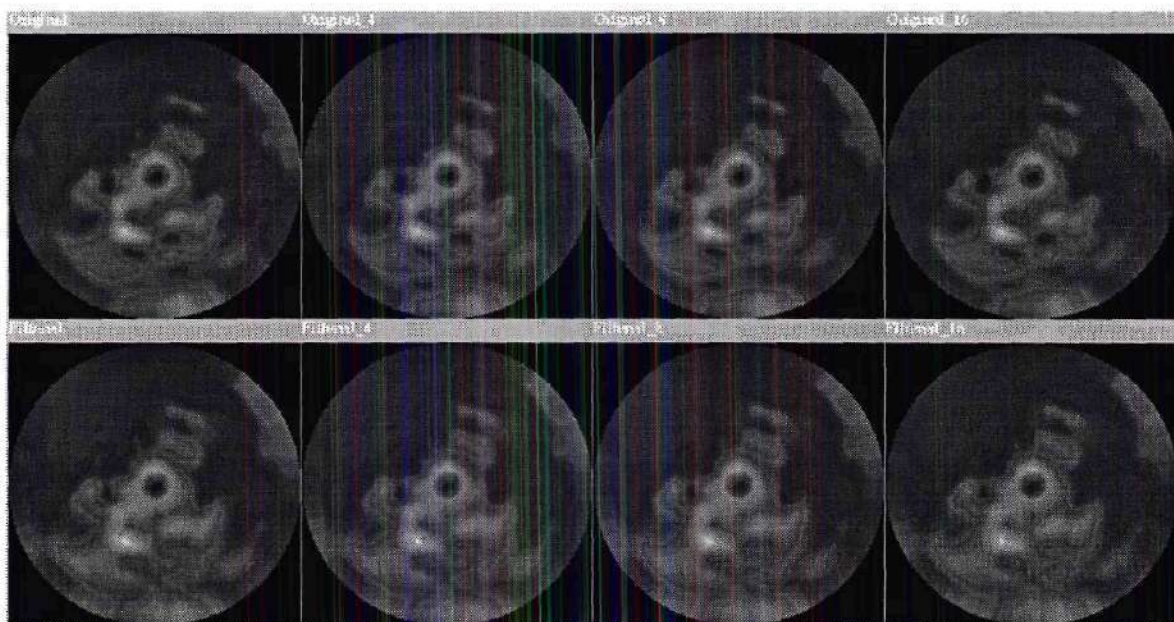


Figure C.3: REST EXAMPLE: The bull's-eyes when using SCALAR quantization. The top row shows from left to right the bull's-eyes for the *original* sequence, the *original* sequence with the body region quantized to 4, 8, 16 levels, respectively. The bottom row shows from left to right the bull's-eyes for the *filtered* sequence, the *filtered* sequence with the body region quantized to 4, 8, 16 levels, respectively.

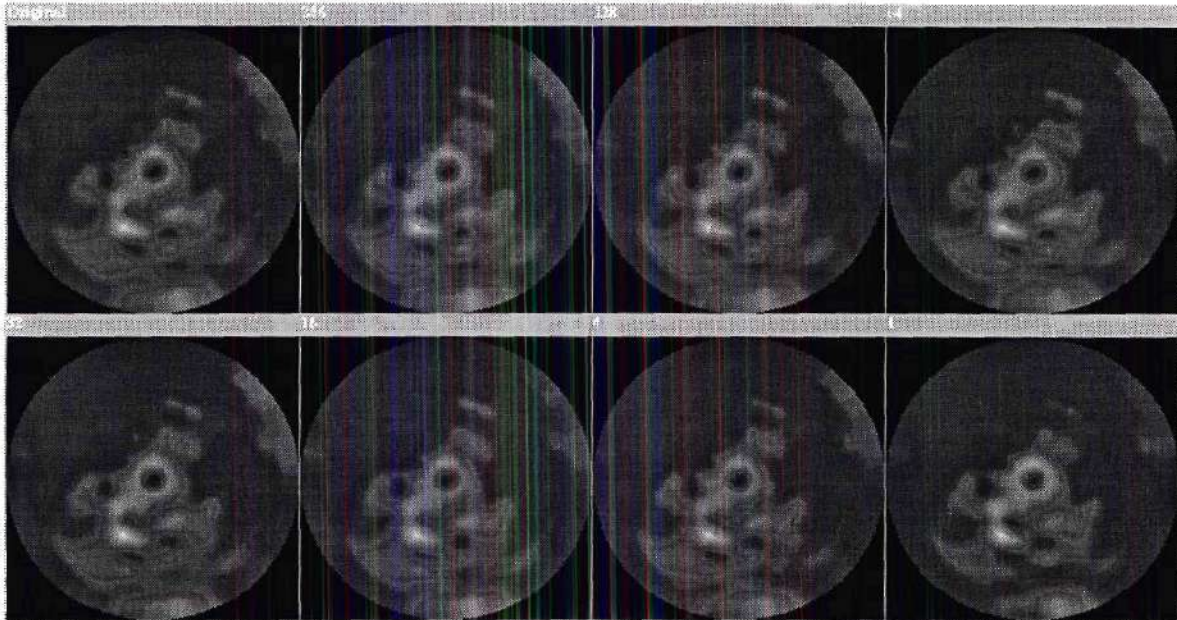


Figure C.4: REST EXAMPLE: The bull's-eyes when using the VECTOR coder. The bull's-eyes are from top to bottom left to right for the *filtered* sequence and the *filtered* sequence with the body region quantized using 1, 4, 16, 32, 64, 128, and 256 codebook sizes.

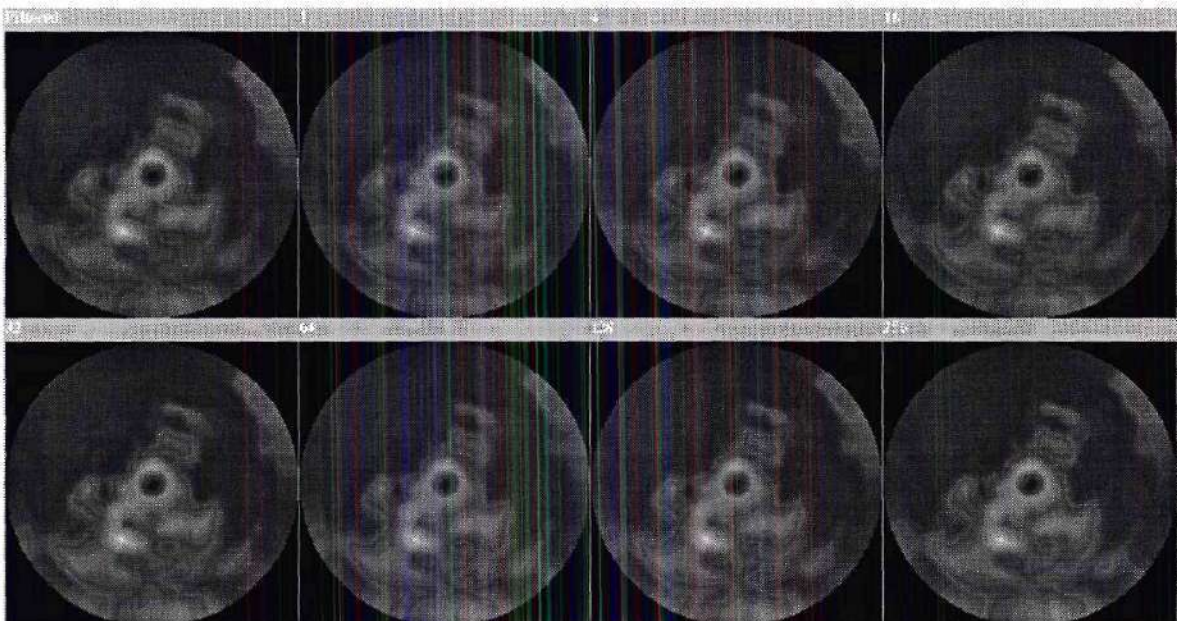


Figure C.5: REST EXAMPLE: The bull's-eyes when using the VECTOR coder. The bull's-eyes are from top to bottom left to right for the *original* sequence and the *original* sequence with the body region quantized using 1, 4, 16, 32, 64, 128, and 256 codebook sizes.

Table C.1: REST EXAMPLE: The averages of the 10 regions of the *original* sequence when using the SCALAR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Original	58	59	65	51	56	54	61	35	46	47
16 levels	60	60	66	51	57	54	62	35	47	47
8 levels	60	61	67	52	57	54	63	36	47	48
4 levels	62	61	68	53	58	55	63	36	48	48
2 levels	64	62	69	53	59	57	65	37	49	49

Table C.2: REST EXAMPLE: The averages of the 10 regions of the *original* sequence when using the VECTOR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Original	58	59	65	51	56	54	61	35	46	47
256	60	58	66	51	56	54	61	35	46	47
128	59	58	65	51	56	54	61	35	46	47
64	58	58	66	51	56	54	61	35	46	47
32	57	59	66	51	56	53	61	35	47	47
16	59	60	66	51	56	54	62	35	47	47
8	59	59	65	50	56	54	62	36	47	48

Table C.3: REST EXAMPLE: The averages of the 10 regions of the *filtered* sequence when using the SCALAR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Filtered	56	59	64	49	55	52	60	33	45	45
16 levels	54	58	63	48	54	51	60	33	45	45
8 levels	55	59	63	48	54	52	59	33	45	45
4 levels	55	59	63	48	53	51	59	33	45	45
2 levels	54	59	63	50	53	51	60	34	45	46

Table C.4: REST EXAMPLE: The averages of the 10 regions of the *filtered* sequence when using the VECTOR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Filtered	56	59	64	49	55	52	60	33	45	45
256	53	58	63	48	54	51	59	33	45	45
128	55	59	63	49	54	51	59	33	45	45
64	54	59	63	49	53	51	59	33	45	45
32	55	59	63	49	54	51	59	33	45	45
16	55	59	63	49	54	51	59	33	45	45
8	54	59	63	49	54	51	59	33	45	45

Table C.5: REST EXAMPLE: The standard deviations of the 10 regions of the *original* sequence when using the SCALAR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Original	16.71	9.75	8.91	6.88	5.40	9.58	10.42	9.90	6.87	11.72
16 levels	16.91	9.84	9.21	7.05	5.59	9.98	10.48	10.31	7.11	11.66
8 levels	16.52	10.08	9.34	7.17	5.31	10.09	10.44	10.51	7.34	11.91
4 levels	16.57	10.34	9.64	7.36	5.18	10.30	10.54	10.80	7.57	12.00
2 levels	18.51	10.58	10.33	8.10	5.55	10.91	10.70	10.98	7.29	12.12

Table C.6: REST EXAMPLE: The standard deviations of the 10 regions of the *original* sequence when using the VECTOR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Original	16.71	9.75	8.91	6.88	5.40	9.58	10.42	9.90	6.87	11.72
256	16.67	9.35	9.48	7.00	5.75	9.60	10.16	10.13	6.91	11.94
128	15.15	10.08	9.53	7.11	5.45	9.56	10.38	10.19	7.01	11.68
64	14.48	9.84	9.22	6.80	5.64	9.48	10.24	9.25	7.02	11.60
32	17.33	9.57	8.58	6.67	5.76	9.60	10.14	9.79	6.93	11.49
16	15.57	9.49	8.82	7.18	5.37	9.60	10.33	10.14	7.25	12.06
8	18.08	9.18	8.89	7.60	5.42	9.92	10.47	10.76	7.11	11.84

Table C.7: REST EXAMPLE: The standard deviations of the 10 regions of the *filtered* sequence when using the SCALAR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Filtered	16.11	7.69	6.40	6.28	5.29	8.70	9.27	9.93	6.16	11.05
16 levels	16.54	7.54	6.49	6.06	5.23	8.48	9.17	9.82	6.14	11.02
8 levels	15.72	7.74	6.36	6.06	5.18	8.71	9.21	9.88	6.24	11.02
4 levels	15.12	8.26	6.21	6.09	5.28	8.72	9.09	9.87	6.38	10.95
2 levels	16.35	7.98	6.35	6.27	5.90	8.82	9.13	9.98	6.61	11.08

Table C.8: REST EXAMPLE: The standard deviations of the 10 regions of the *filtered* sequence when using the VECTOR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Filtered	16.11	7.69	6.40	6.28	5.29	8.70	9.27	9.93	6.16	11.05
256	18.17	7.60	6.34	6.24	5.22	8.38	9.10	9.68	6.07	10.99
128	15.96	7.64	6.32	6.34	5.20	8.77	9.29	9.93	6.17	10.99
64	15.66	7.65	6.41	6.19	5.30	8.74	9.20	9.88	6.26	11.01
32	15.99	7.73	6.57	6.35	5.29	8.67	9.21	9.92	6.22	10.97
16	16.04	7.37	6.22	6.25	5.09	8.74	9.28	9.82	6.30	11.00
8	16.20	7.93	6.35	6.46	4.72	8.60	9.30	9.76	6.07	10.97

Table C.9: REST EXAMPLE: The averages of the septal (S), lateral (L), anterior (A), and inferior (I) regions and the corresponding ratios of the *original* sequence when using the SCALAR coder.

Sequence	S	L	A	I	S/L	A/L	I/L	A/I
Original	126	102	113	86	1.235	1.108	0.843	1.314
16 levels	128	104	114	86	1.231	1.096	0.827	1.326
8 levels	130	104	115	88	1.250	1.106	0.846	1.307
4 levels	131	106	116	89	1.236	1.094	0.840	1.303
2 levels	134	108	119	90	1.241	1.102	0.833	1.322

Table C.10: REST EXAMPLE: The averages of the septal (S), lateral (L), anterior (A), and inferior (I) regions and the corresponding ratios of the *original* sequence when using the VECTOR coder.

Sequence	S	L	A	I	S/L	A/L	I/L	A/I
Original	126	102	113	86	1.235	1.108	0.843	1.314
256	127	102	112	86	1.245	1.098	0.843	1.302
128	126	102	112	86	1.235	1.098	0.843	1.302
64	127	102	112	86	1.245	1.098	0.843	1.302
32	127	103	112	86	1.233	1.087	0.835	1.302
16	128	103	114	86	1.243	1.107	0.835	1.326
8	127	103	113	86	1.233	1.097	0.835	1.314

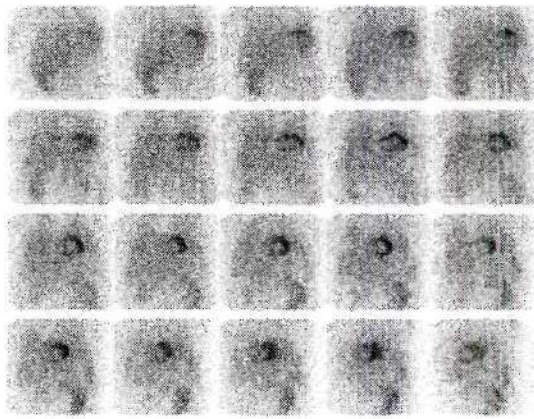
Table C.11: REST EXAMPLE: The averages of the septal (S), lateral (L), anterior (A), and inferior (I) regions and the corresponding ratios of the *filtered* sequence when using the SCALAR coder.

Sequence	S	L	A	I	S/L	A/L	I/L	A/I
Filtered	124	100	111	82	1.240	1.110	0.820	1.354
16 levels	123	99	109	81	1.242	1.101	0.818	1.346
8 levels	122	99	111	81	1.232	1.121	0.818	1.370
4 levels	122	98	110	81	1.245	1.122	0.827	1.358
2 levels	123	98	110	84	1.255	1.122	0.857	1.310

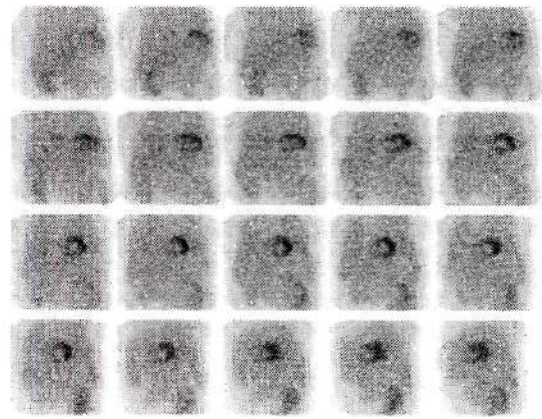
Table C.12: REST EXAMPLE: The averages of the septal (S), lateral (L), anterior (A), and inferior (I) regions and the corresponding ratios of the *filtered* sequence when using the VECTOR coder.

Sequence	S	L	A	I	S/L	A/L	I/L	A/I
Filtered	124	100	111	82	1.240	1.110	0.820	1.354
256	122	99	109	81	1.232	1.101	0.818	1.346
128	122	99	110	82	1.232	1.111	0.828	1.341
64	122	98	110	82	1.245	1.122	0.837	1.341
32	122	99	110	82	1.232	1.111	0.828	1.341
16	122	99	110	82	1.232	1.111	0.828	1.341
8	122	99	110	82	1.232	1.111	0.828	1.341

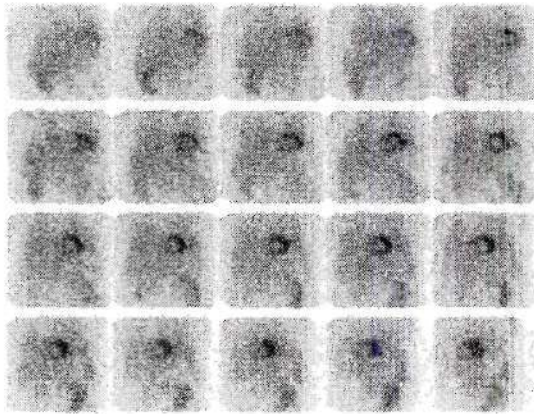
C.2 Stress Example



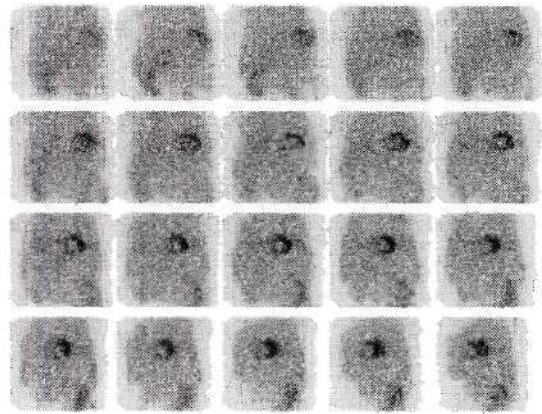
(a) *Original*



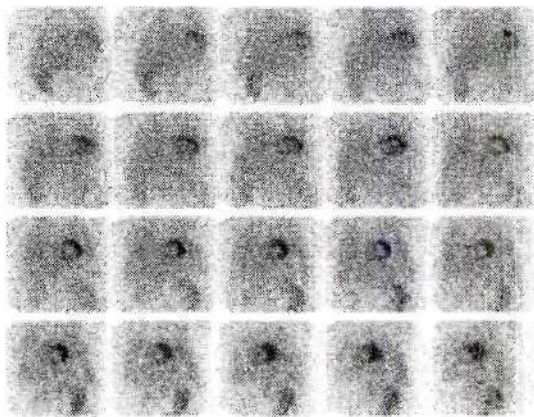
(b) *Filtered*



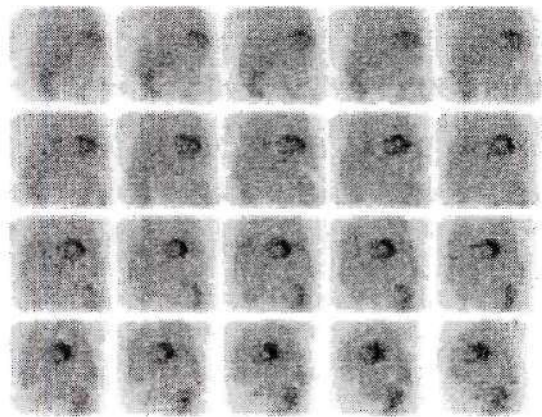
(c) SCALAR quantized *original*



(d) SCALAR quantized *filtered*



(e) VECTOR quantized *original*



(f) VECTOR quantized *filtered*

Figure C.6: STRESS EXAMPLE: (a) The *original* sequence. (b) The *filtered* sequence. (c) 8-level SCALAR quantized *original* sequence. (d) 8-level SCALAR quantized *filtered* sequence. (e) 8-vector VECTOR quantized *original* sequence. (f) 8-vector VECTOR quantized *filtered* sequence.

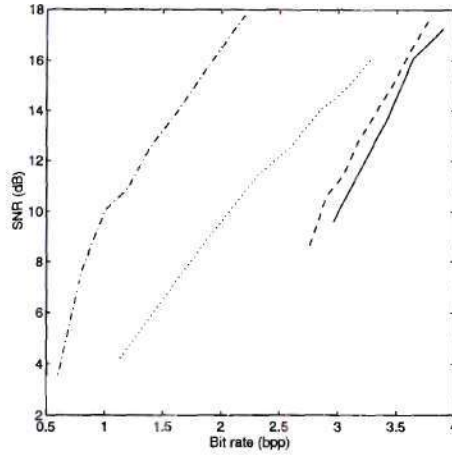


Figure C.7: STRESS EXAMPLE: Comparing Region-based SCALAR and VECTOR quantization approaches. The solid and dotted lines are for the *original* sequence when using the SCALAR and VECTOR coders, respectively. The dashed and dash-dotted lines are for the *filtered* sequence when using the SCALAR and VECTOR coders, respectively.

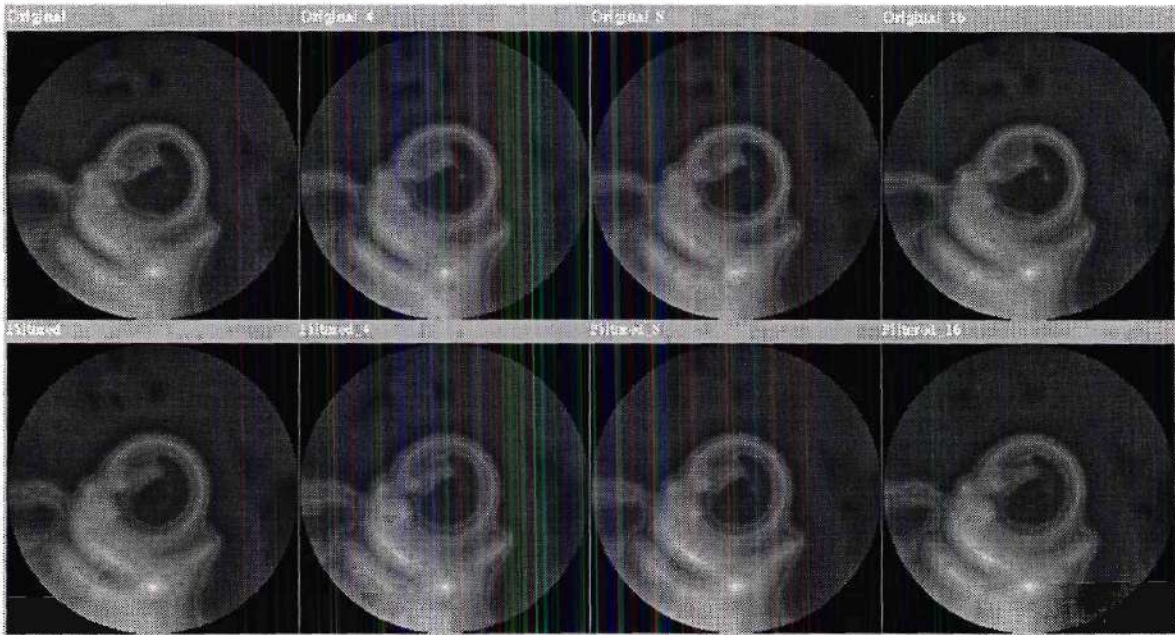


Figure C.8: STRESS EXAMPLE: The bull's-eyes when using SCALAR quantization. The top row shows from left to right the bull's-eyes for the *original* sequence, the *original* sequence with the body region quantized to 4, 8, 16 levels, respectively. The bottom row shows from left to right the bull's-eyes for the *filtered* sequence, the *filtered* sequence with the body region quantized to 4, 8, 16 levels, respectively.

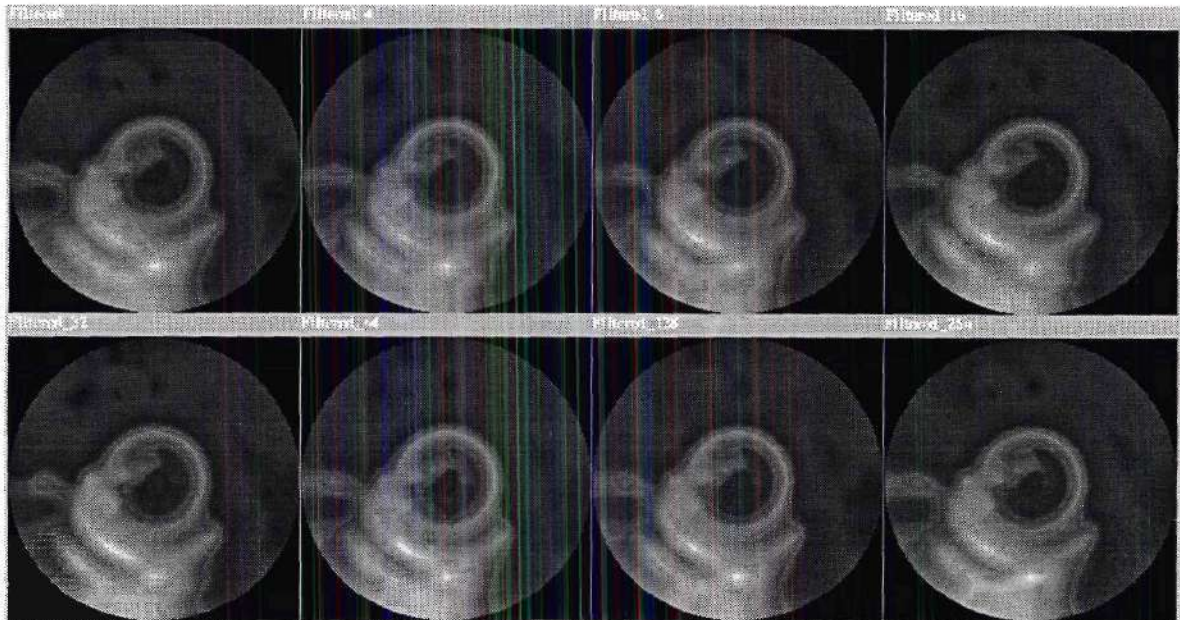


Figure C.9: STRESS EXAMPLE: The bull's-eyes when using the VECTOR coder. The bull's-eyes are from top to bottom left to right for the *filtered* sequence and the *filtered* sequence with the body region quantized using 1, 4, 16, 32, 64, 128, and 256 codebook sizes.

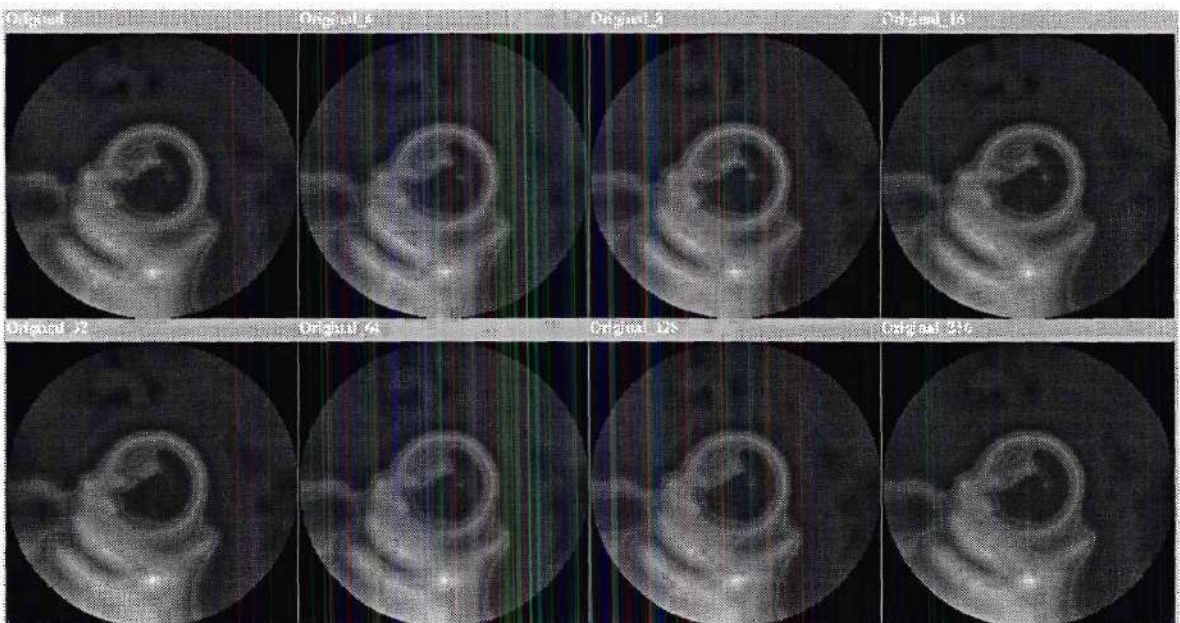


Figure C.10: STRESS EXAMPLE: The bull's-eyes when using the VECTOR coder. The bull's-eyes are from top to bottom left to right for the *original* sequence and the *original* sequence with the body region quantized using 1, 4, 16, 32, 64, 128, and 256 codebook sizes.

Table C.13: STRESS EXAMPLE: The averages of the 10 regions of the *original* sequence when using the SCALAR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Original	41	52	57	60	53	47	67	29	25	37
16 levels	42	53	58	61	54	47	67	29	25	37
8 levels	42	54	59	62	55	47	68	30	25	38
4 levels	42	54	58	62	55	47	68	29	25	38
2 levels	43	54	59	62	55	47	68	30	25	38

Table C.14: STRESS EXAMPLE: The averages of the 10 regions of the *original* sequence when using the VECTOR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Original	41	52	57	60	53	47	67	29	25	37
256	41	52	57	60	53	46	66	29	25	37
128	41	52	57	60	53	46	67	29	25	37
64	41	52	57	60	53	46	67	29	25	37
32	40	52	57	60	53	46	66	29	25	37
16	41	52	57	61	53	46	67	29	25	37
8	42	52	57	61	53	46	66	29	25	37

Table C.15: STRESS EXAMPLE: The averages of the 10 regions of the *filtered* sequence when using the SCALAR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Filtered	39	51	54	56	51	44	63	29	25	34
16 levels	38	51	54	56	51	44	63	28	24	34
8 levels	39	51	53	56	51	44	63	28	24	34
4 levels	39	51	55	57	51	45	65	29	24	34
2 levels	39	53	57	60	53	47	68	30	24	35

Table C.16: STRESS EXAMPLE: The averages of the 10 regions of the *filtered* sequence when using the VECTOR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Filtered	39	51	54	56	51	44	63	29	25	34
256	38	50	53	56	50	43	63	29	24	34
128	35	51	54	56	51	44	63	29	24	34
64	37	52	54	56	52	44	63	28	24	34
32	36	50	53	55	50	44	63	28	23	33
16	35	50	53	57	50	43	63	28	23	34
8	38	51	54	56	51	44	64	28	23	34

Table C.17: STRESS EXAMPLE: The standard deviations of the 10 regions of the *original* sequence when using the SCALAR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Original	10.04	11.75	8.19	4.74	11.26	17.49	8.01	13.18	7.77	21.28
16 levels	10.58	11.77	8.14	4.76	11.32	17.66	8.02	13.26	7.71	21.49
8 levels	10.49	11.77	8.28	4.76	11.32	17.77	8.18	13.30	7.63	21.91
4 levels	10.46	11.77	8.35	4.78	11.37	17.86	8.26	13.39	7.68	21.80
2 levels	10.55	11.77	8.17	4.71	11.32	17.78	8.25	13.28	7.57	21.82

Table C.18: STRESS EXAMPLE: The standard deviations of the 10 regions of the *original* sequence when using the VECTOR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Original	10.04	11.75	8.19	4.74	11.26	17.49	8.01	13.18	7.77	21.28
256	10.08	11.70	8.09	4.74	11.29	17.39	7.95	12.97	7.38	21.26
128	9.75	11.72	8.24	4.68	11.29	17.33	7.96	12.99	7.37	21.14
64	9.76	11.36	8.10	4.74	10.89	17.56	8.00	13.17	7.39	21.31
32	10.27	11.78	8.04	4.65	11.29	17.37	7.89	12.96	7.37	21.20
16	9.77	12.24	8.67	5.07	11.77	17.49	7.99	13.08	7.37	21.14
8	10.53	12.26	8.66	5.07	11.82	17.46	8.04	13.02	7.38	21.24

Table C.19: STRESS EXAMPLE: The standard deviations of the 10 regions of the *filtered* sequence when using the SCALAR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Filtered	8.55	8.03	6.92	3.56	7.77	15.85	6.92	11.54	7.83	19.86
16 levels	8.80	8.00	7.31	3.75	7.80	16.27	7.17	11.29	7.19	20.09
8 levels	8.54	8.00	7.09	3.89	7.82	16.30	7.36	11.26	7.05	19.71
4 levels	9.66	7.53	7.43	3.63	7.30	16.72	7.37	12.15	7.54	21.02
2 levels	11.08	9.31	8.03	3.89	9.26	17.30	7.17	12.62	6.70	22.52

Table C.20: STRESS EXAMPLE: The standard deviations of the 10 regions of the *filtered* sequence when using the VECTOR coder.

Sequence	REGION									
	1	2	3	4	5	6	7	8	9	10
Filtered	8.55	8.03	6.92	3.56	7.77	15.85	6.92	11.54	7.83	19.86
256	9.56	8.50	7.68	4.02	8.27	15.77	7.05	11.35	7.18	19.50
128	11.48	8.50	7.44	4.46	8.29	16.23	7.36	11.06	7.15	19.60
64	9.96	8.00	6.93	4.64	7.83	15.71	7.00	11.35	7.30	19.63
32	9.51	7.57	7.41	4.24	7.34	15.42	7.29	11.65	7.05	19.71
16	10.17	9.47	7.93	4.20	9.25	16.04	7.18	11.49	7.36	20.03
8	9.29	8.50	7.84	5.10	8.37	16.14	7.75	11.37	7.09	19.91

Table C.21: STRESS EXAMPLE: The averages of the septal (S), lateral (L), anterior (A), and inferior (I) regions and the corresponding ratios of the *original* sequence when using the SCALAR coder.

Sequence	S	L	A	I	S/L	A/L	I/L	A/I
Original	124	78	99	89	1.590	1.269	1.141	1.112
16 levels	125	79	100	90	1.582	1.266	1.139	1.111
8 levels	127	80	101	92	1.587	1.262	1.150	1.098
4 levels	126	80	101	91	1.575	1.262	1.137	1.110
2 levels	127	80	101	92	1.587	1.262	1.150	1.098

Table C.22: STRESS EXAMPLE: The averages of the septal (S), lateral (L), anterior (A), and inferior (I) regions and the corresponding ratios of the *original* sequence when using the VECTOR coder.

Sequence	S	L	A	I	S/L	A/L	I/L	A/I
Original	124	78	99	89	1.590	1.269	1.141	1.112
256	123	78	98	89	1.577	1.256	1.141	1.101
128	124	78	98	89	1.590	1.256	1.141	1.101
64	124	78	98	89	1.590	1.256	1.141	1.101
32	123	78	98	89	1.577	1.256	1.141	1.101
16	124	78	98	90	1.590	1.256	1.154	1.089
8	123	78	98	90	1.577	1.256	1.154	1.089

Table C.23: STRESS EXAMPLE: The averages of the septal (S), lateral (L), anterior (A), and inferior (I) regions and the corresponding ratios of the *filtered* sequence when using the SCALAR coder.

Sequence	S	L	A	I	S/L	A/L	I/L	A/I
Filtered	117	76	95	85	1.539	1.250	1.118	1.118
16 levels	117	75	95	84	1.560	1.267	1.120	1.131
8 levels	116	75	95	84	1.547	1.267	1.120	1.131
4 levels	120	75	96	86	1.600	1.280	1.147	1.116
2 levels	125	77	100	90	1.623	1.299	1.169	1.111

Table C.24: STRESS EXAMPLE: The averages of the septal (S), lateral (L), anterior (A), and inferior (I) regions and the corresponding ratios of the *filtered* sequence when using the VECTOR coder.

Sequence	S	L	A	I	S/L	A/L	I/L	A/I
Filtered	117	76	95	85	1.539	1.250	1.118	1.118
256	116	74	93	85	1.568	1.257	1.149	1.094
128	117	75	95	85	1.560	1.267	1.133	1.118
64	117	76	96	84	1.539	1.263	1.105	1.143
32	116	73	94	83	1.589	1.288	1.137	1.133
16	116	73	93	85	1.589	1.274	1.164	1.094
8	118	74	95	84	1.595	1.284	1.135	1.131

Bibliography

- [1] D. R. Aberle, F. Gleeson, J. W. Sayre, K. Brown, P. Batra, D. A. Young, B. K. Stewart, B. K. T. Ho, and H. K. Huang. The effect of irreversible image compression on diagnostic accuracy in thoracic imaging. *Optical Engineering*, 28(5):398 – 403, May 1993.
- [2] E. H. Adelson and E. Simoncelli. Orthogonal pyramid transforms for image coding. In *Visual Communications and Image Processing II*, volume 845, pages 50 – 58. SPIE, 1987.
- [3] O. K. Al-Shaykh and R. M. Mersereau. Lossy compression of noisy cardiac sequences. In *Proc. Data Compression Conference*, pages 43–52, 1996.
- [4] R. B. Arps and T. K. Truong. Comparison of international standards for lossless still image compression. *Proc. IEEE*, 82(6):889 – 899, June 1994.
- [5] H. H. Barrett, T. Gooley, K. Girodias, J. Rolland, T. White, and J. Yao. Linear discriminants and image quality. In *Proc. 1991 Int. Conf. on Information Processing in Medical Imaging (IPMI' 91)*, pages 458–473. Springer-Verlag, July 1991.
- [6] H. H. Barrett and W. Swindell. *Radiological Imaging*, volume I. Academic Press, New York, N.Y., 1981.
- [7] H. H. Barrett and W. Swindell. *Radiological Imaging*, volume II. Academic Press, New York, N.Y., 1981.
- [8] T. Berger. *Rate Distortion Theory*. Prentice-Hall, Englewood Cliffs, N.J., 1971.
- [9] P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. Communications*, COM-31(4):532 – 540, April 1983.
- [10] C. L. Chan, A. K. Katsaggelos, and A. V. Sahakian. Image sequence filtering in quantum-limited noise with application to low-dose fluoroscopy. *IEEE Trans. Medical Imaging*, 12(3):610 – 621, September 1993.
- [11] J. Chen, J. Flynn, B. Gross, and D. Spizarny. Observer detection of image degradation caused by irreversible data compression processes. In *Image Capture, Formatting, and Display*, volume 1444, pages 256–264. SPIE, 1991.

- [12] P. A. Chou, T. Lookabaugh, and R. M. Gray. Entropy constrained vector quantization. *IEEE Trans. Acoust., Speech, and Signal Processing*, 37(1):31 – 42, January 1989.
- [13] W. J. Conover. *Practical Nonparametric Statistics*. John Wiley & Sons, New York, N.Y., 1971.
- [14] L. T. Cook. Loss of information due to compression of medical images. In *Enabling Technologies for High-Bandwidth Applications*, volume 1785, pages 103 – 108. SPIE, 1992.
- [15] P. C. Cosman, R. M. Gray, and R. A. Olshen. Evaluation quality of compressed medical images: SNR, subjective rating, and diagnostic accuracy. *Proc. IEEE*, 82(6):919 – 932, June 1994.
- [16] P. C. Cosman, R. M. Gray, R. A. Olshen, L. E. Moses, H. C. Davidson, C. J. Bergin, and E. A. Riskin. Tree-structured vector quantization of CT chest scans: subjective rating and diagnostic accuracy. *IEEE Trans. Medical Imaging*, 12(4):727 – 739, December 1993.
- [17] E. V. R. Di Bella. *Effects of Attenuation and Blurring in Cardiac SPECT and Compensations Using Parallel Computers*. Georgia Institute of Technology, Atlanta, GA, Ph.D. Thesis, 1995.
- [18] R. L. Dobrushin and B. S. Tsybakov. Information transmission with additional noise. *IRE Trans. Information Theory*, IT-8:293–304, 1962.
- [19] T. R. Fischer, J. D. Gibson, and B. Koo. Estimation and noisy source coding. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 38(1):23 – 34, January 1990.
- [20] D. Geiger and F. Girosi. Parallel and deterministic algorithms from MRF's: surface reconstruction. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(5):401 – 412, May 1991.
- [21] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6:721 – 741, November 1984.
- [22] S. Geman and D. McClure. Bayesian image analysis: An application to single photon emission tomography. In *Proc. Statist. Comput. Sect.*, pages 12 – 18, 1985.
- [23] R. C. Gonzalez and R. F. Woods. *Digital Image Processing*. Addison-Wesley, Reading, MA., 1989.
- [24] R. Gray. Vector quantization. *IEEE ASSP Mag.*, pages 4 – 29, April 1984.
- [25] R. M. Gray. *Source Coding Theory*. Kluwer Academic Publishers, Boston, Ma, 1990.

- [26] R. W. Hamming. *Coding and Information Theory*. Prentice-Hall, Englewood Cliffs, N.J., 1980.
- [27] T. J. Hebert and R. Leahy. Statistic-based MAP image reconstruction from Poisson data using Gibbs priors. *IEEE Trans. Signal Processing*, 40(9):2290–2303, September 1992.
- [28] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, N.J., 1989.
- [29] N. S. Jayant and P. Noll. *Digital Coding of Waveforms: Principles and Applications*. Prentice-Hall, Englewood Cliffs, N.J., 1984.
- [30] D. H. Johnson and D. E. Dudgeon. *Array Signal Processing Concepts and Techniques*. Prentice-Hall, Englewood Cliffs, N.J., 1993.
- [31] R. H. Jonsson. *Adaptive subband coding of video using probability distribution models*. Georgia Institute of Technology, Atlanta, GA, Ph.D. Thesis, 1994.
- [32] R. P. Kleihorst. *Noise Filtering of Image Sequences*. Delft University of Technology, Delft, The Netherlands, Ph.D. Thesis, 1994.
- [33] D. T. Kuan, A. A. Sawchuk, T. C. Strand, and P. Chavel. Adaptive noise smoothing filter for images with signal-dependent noise. *IEEE Trans. Pattern Anal. Machine Intell.*, 7(2):165–177, March 1985.
- [34] G. G. Langdon Jr. An introduction to arithmetic coding. *IBM J. Res. Develop.*, 28(2):135 – 149, March 1984.
- [35] H. Lee, A. H. Rowberg, M. S. Frank, H. S. Choi, and Y. Kim. Subjective evaluation of compressed image quality. In *Image Capture, Formatting, and Display*, volume 1653, pages 241 – 248. SPIE, 1992.
- [36] S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, Berlin; New York, 1995.
- [37] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Trans. Communications*, COM-28(1):84 – 95, January 1980.
- [38] S. P. Lloyd. Least squares optimization in PCM. *IEEE Trans. Information Theory (reproduction of a paper presented at the Institute of Mathematical Statistics meeting in Atlantic City, N. J., September 10-13, 1957)*, IT-28(2):129 – 137, March 1982.
- [39] S. C. B. Lo, B. Krasner, and S. K. Mun. Noise impact on error-free image compression. *IEEE Trans. Medical Imaging*, 9(2):202 – 206, June 1990.

- [40] T. Lookabaugh, E. A. Riskin, P. A. Chou, and R. M. Gray. Variable rate vector quantization for speech, image, and video compression. *IEEE Trans. Communications*, 41(1):186 – 199, January 1993.
- [41] A. Macovski. *Medical Imaging*. Prentice-Hall, Englewood Cliffs, N.J., 1983.
- [42] P. A. Maragos, R. W. Schafer, and R. M. Mersereau. Two-dimensional linear prediction and its application to adaptive predictive coding of images. *IEEE Trans. Acoust., Speech, and Signal Processing*, ASSP-32(6):1213 – 1229, December 1984.
- [43] J. Max. Quantization for minimum distortion. *IRE Trans. Information Theory*, IT-16(2):7 – 12, March 1960.
- [44] P. W. Melnychuk, M. J. Barry, and M. S. Mathieu. The effect of noise and MTF on the compressibility of high resolution color images. In *Image Processing Algorithms and Techniques*, volume 1244, pages 255 – 262. SPIE, 1990.
- [45] N. M. Nasrabadi and R. A. King. Image coding using vector quantization: a review. *IEEE Trans. Communications*, 36(8):957 – 971, August 1988.
- [46] B. K. Natarajan. Filtering random noise from deterministic signals via data compression. *IEEE Trans. Signal Processing*, 43(11):2595 – 2605, November 1995.
- [47] T. Özcelik, J. C. Brailean, and A. K. Katsaggelos. Image and video compression algorithms based on recovery techniques using mean field annealing. *Proceedings of the IEEE*, 83(2):304 – 316, February 1995.
- [48] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, N.Y., third edition, 1991.
- [49] M. Rabbani. Bayesian filtering of Poisson noise using local statistics. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 36(6):933 – 937, June 1988.
- [50] R. P. Rao and W. A. Pearlman. On entropy of pyramid structures. *IEEE Trans. Information Theory*, 37(2):407 – 413, March 1991.
- [51] M. S. Rebelo, S. S. Furuie, A. C. L. Munhoz, L. Moura, and C. P. Melo. Assessment of lossy image compression technique applied to cardiac images in nuclear medicine. In *Computers in cardiology*, pages 827 – 830. IEEE, 1993.
- [52] E. A. Riskin, T. Lookabaugh, P. A. Chou, and R. M. Gray. Variable rate vector quantization for medical image compression. *IEEE Trans. Medical Imaging*, 9(3):290 – 298, September 1990.
- [53] J. Rissanen and G. G. Langdon Jr. Arithmetic coding. *IBM J. Res. Develop.*, 23(2):149 – 162, March 1979.

- [54] P. Roos, M. A. Viergever, , M. C. A. Van Dijke, and J. H. Peters. Reversible intraframe compression of medical images. *IEEE Trans. Medical Imaging*, 7(4):328 – 336, December 1988.
- [55] J. Sayre, D. R. Aberle, M. I. Boechat, T. R. Hall, H. K. Huang, B. K. Ho, P. Kashfian, and G. Rahbar. Effect of data compression on diagnostic accuracy in digital hand and chest radiography. In *Image Capture, Formatting, and Display*, volume 1653, pages 232–240. SPIE, 1992.
- [56] T. Senoo and B. Girod. Vector quantization for entropy coding of image subbands. *IEEE Trans. Image Processing*, 1(4):526 – 533, October 1992.
- [57] M. I. Sezan and A. M. Teaklp. Survey of recent developments in digital image restoration. *Optical Engineering*, 29(5):393 – 404, May 1990.
- [58] L. A. Shepp and B. F. Logan. The Fourier reconstruction of a head section. *IEEE Trans. Nuclear Science*, 21(3):21–43, 1974.
- [59] H. Stark and J. W. Woods. *Probability, Random Processes, and Estimation Theory for Engineers*. Prentice-Hall, Englewood Cliffs, N.J., 1994.
- [60] M. Subbotin. On the low of frequency error. *Mathematicheskii Sbornik*, 31:296 – 301, 1923.
- [61] A. M. Tekalp and G. Pavlović. Image restoration with multiplicative noise: incorporating the sensor nonlinearity. *IEEE Trans. Signal Processing*, 39(9):2132–2136, September 1991.
- [62] P. P. Vaidyanathan. Quadrature mirror filter banks, perfect-reconstruction techniques. *IEEE ASSP Mag.*, pages 4 – 20, July 1987.
- [63] H. L. Van Trees. *Detection, Estimation, and Modulation Theory*. John Wiley & Sons, New York, N.Y., 1968.
- [64] G. K. Wallace. The JPEG still picture compression standard. *IEEE Trans. Consumer Electronics*, 38(1):xviii – xxxiv, February 1992.
- [65] L. Wang and M. Goldberg. Progressive image transmission using vector quantization on images in pyramid form. *IEEE Trans. Communications*, 37(12):1339 – 1349, December 1989.
- [66] L. Wang and M. Goldberg. Comparative performance of pyramid data structures for progressive transmission of medical imagery. In *Image Capture and Display*, volume 1232, pages 403 – 413. SPIE, 1990.
- [67] P. Wilhelm and D. R. Haynor. Lossy image compression for digital medical imaging systems. *Optical Engineering*, 30(10):1479 – 1485, October 1991.

- [68] G. Winkler. *Image Analysis: Markov Fields and Dynamic Monte Carlo Methods*. Springer-Verlag, Berlin; New York, 1995.
- [69] J. K. Wolf and J. Ziv. Transmission of noisy information to a noisy receiver with minimum distortion. *IEEE Trans. Information Theory*, IT-16(4):406 – 411, July 1970.
- [70] J. W. Woods and S. D. O’Neil. Subband coding of images. *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-34(5):550 – 560, October 1986.
- [71] B. L. Zaret and F. J. Wackers. Nuclear cardiology. *The New England Journal of Medicine*, 329(11):775 – 783, September 1993.
- [72] J. Zhang. The mean field theory in EM procedures for Markov random fields. *IEEE Trans. Signal Processing*, 40(10):2570 – 2583, October 1992.
- [73] J. Zhang. The mean field theory in EM procedures for blind Markov random field image restoration. *IEEE Trans. Image Processing*, 2(1):27 – 40, January 1993.
- [74] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Trans. Information Theory*, IT-23(3):337 – 343, May 1977.

Vita

Osama K. Al-Shaykh was born on April, 8th 1969. He received the B.Sc. degree in Electrical Engineering from the University of Jordan in 1990, and the M.Sc. degree in Electrical Engineering from Iowa State University in 1992. During his studies at Iowa State University, he was awarded the Fulbright scholarship. His research interests include image and video restoration and compression, medical imaging, pattern recognition, and neural networks.