Purdue University

Purdue e-Pubs

Department of Electrical and Computer Engineering Technical Reports Department of Electrical and Computer Engineering

8-1-1991

Blockwise Transform Image Coding Enhancement and Edge Detection

Sabzali Aghagolzadeh Purdue University

Follow this and additional works at: https://docs.lib.purdue.edu/ecetr

Aghagolzadeh, Sabzali, "Blockwise Transform Image Coding Enhancement and Edge Detection" (1991). *Department of Electrical and Computer Engineering Technical Reports.* Paper 747. https://docs.lib.purdue.edu/ecetr/747

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

BLOCKWISE TRANSFORM IMAGE CODING,

ENHANCEMENT AND EDGE DETECTION

A Thesis

Submitted to the Faculty

of

د با رو د د د

Purdue University

by

Sabzali Aghagolzadeh

In Partial Fulfillment of the Requirements for the Degree

of

Doctor of Philosophy

August 1991

This is dedicated to my parents Abbas and Habibeh, my wife, Saeedeh, and my children Mehdi, Maryam and Mohammad. ii

ACKNOWLEDGEMENTS

1948년 · [1948] · [1949] · [1942] · [1942] · [1943] · [1944] · [1945] · [1945] · [1945] · [1945] · [1945] · [194

I would like to thank my advisor, Professor Okan Ersoy, for his generous guidance, encouragement and support during the course of this research. I would also like to thank Professor Jan Allebach, Professor Charles Bouman and Professor Vaughan Weston for their invaluable discussion and serving in my committee. Other people have also played a major part in my success. Most of all, I would like to express my special appreciation to my parents who have patiently supported me in reaching my goals. More important, I sincerely thank my wife and children for their patience and encouragement during my studies in graduate school. Finally, I greatly appreciate many friends for their help in preparing these manuscripts.

TABLE OF CONTENTS

	이 바늘 병원은 전에 있는 것은 것이 있는 것이 많은 것이 같이 많이 많이 많이 많이 없다.
LIST (OF TABLES
LIST (OF FIGURESv
ABST	RACTx
INTR	ODUCTION
CHAP	TER 1 - A REVIEW OF BASIC PRINCIPLES
1.1	Predictive Coding
1.2	Transform Coding
1.3	Adaptive Transform Coding
1.4	Scalar Quantization
1.5	Visual Blockwise Processing
1.6	Image Enhancement
1.7	Edge Detection
1.8	Motivation for this Research
	승규는 가슴에 가슴을 가슴을 가슴을 가슴을 가슴을 가슴을 가슴을 가슴을 가슴을 다.
CHAP	TER 2 - OPTIMAL ADAPTIVE MULTISTAGE IMAGE
	TRANSFORM CODING1
2.1	Introduction1
9.9	The MSE Function for the Proposed Method

2.2	The MSE Function for the Proposed Method	16
2.3	Error Models and Optimal Bit Allocation	18
2.4	Experimental Results	24
2.5	Discussion	26
2.6	Conclusions	27

Page

		Pa
HAP	TER 3 - TRANSFORM IMAGE ENHANCEMENT	5(
3.1	Introduction	50
3.2	Fast Transforms	5
3.3	Generalized Filtering in the Transform Domain	5
3.4	Alpha-Rooting	6
3.5	Modified Unsharp Masking	6
3.6	Filtering Based on the Human Visual System Response	6
3.7	Removal of Block Edge Effects By an Overlap-Save Method	6
3.8	Similarity of Modified Unsharp Masking and HVS Filtering	6
3.9	Conclusions	6
HAP	TER 4 - TRANSFORM EDGE DETECTION	10
4.1	Introduction	10
4.2	Bandpass Masking	11
4.3	Interpolation	11
4.4	Conclusions	11
1.1		
IAP	TER 5 - CONCLUSIONS AND	at a s a tar
1	FUTURE RESEARCH DIRECTIONS	16
2.5		
5.1	Conclusions	16
5.2	Future Research Directions	16
ST (OF REFERENCES	16
TA		17
		1983 1997 - 1997
at d		

LIST OF TABLES

Table	Pag
1.1 Some gray scaling functions for image enhancement [15]	13
2.1 The error model for Gaussian and Laplacian distribution of the form $f_k(b_{kij}) = A2^{-Bb_{kij}}$ given in Ref. [15]	30
2.2 Simulation results for multistage transform coding	35
3.1 The number of additions and multiplications for different transforms.	71
4.1 Simulation results for transform edge detection with noisy checkerboard image and 4 pixels overlapping	133
4.2 Compares the performance of three edge operators using an 11×11 window on the noisy checkerboard image [59]	142
4.3 The number of real additions and multiplications per pixel in transform edge detection method with different transforms, different block sizes and different number of overlapping pixels	162

vi

LIST OF FIGURES

vii

Figure	Page
1.1 Adaptive cosine transform coding system [13]	11
1.2 A quantizer	12
2.1 Block diagram of multistage image transform coding	29
2.2 Approximation of the error model for the Gaussian pdf given in Table 2.1 with $B_k = 1.52$ in the form of $f_k(b_{kij}) = 2^{-B_k b_{kij}}$	31
2.3 Approximation of the error model for the Laplacian pdf given in Table 2.1 with $B_k = 1.23$ in the form of $f_k(b_{kij}) = 2^{-B_k b_{kij}}$	32
2.4 Original images	33
2.5 Reconstructed image	37
2.6 The difference images	41
2.7 Experimental results with two stage coding for all possible values for R_0 and R_1	49
3.1 The energy distribution of different transforms with different threshold levels (see text)	72
3.2 Generalized filtering with a fast transform	73
3.3 Conventional alpha-rooting for image enhancement	74

Figure	Page
3.4 Modified alpha-rooting for image enhancement	74
3.5 Original image "catbrain"	75
3.6 Enhanced image with conventional alpha-rooting with the DFT and $lpha$ equal to 0.85	76
3.7 Enhanced image with conventional alpha-rooting with the DCT and α equal to 0.85	77
3.8 Enhanced image with conventional alpha-rooting with the SRDFT and α equal to 0.85	
3.9 Enhanced image with conventional alpha-rooting with the DC3T and α equal to 0.85	79
3.10 Enhanced image with the modified alpha-rooting with the RDFT and α equal to 0.85	
3.11 Enhanced image with the modified alpha-rooting with the DCT and α equal to 0.85	
3.12 Enhanced image with the modified alpha-rooting with the DCT and α equal to 0.7	
3.13 Enhanced image with the modified alpha-rooting with the DFT and α equal to 0.85	
3.14 Enhanced image with the modified alpha-rooting with the DC3T and α equal to 0.85	84
3.15 Enhanced image with the modified alpha-rooting with the DC3T and α equal to 0.7	
3.16 Enhanced image with the modified alpha-rooting with the SRDFT and α equal to 0.85	

Figure	Page
3.17 Unsharp masking for image enhancement	87
3.18 Modified unsharp masking for image enhancement	88
3.19 Enhanced image with the modified unsharp masking with the RDFT and C equal to 7.0	89
3.20 Enhanced image with the modified unsharp masking with the DFT and C equal to 7.0	90
3.21 Enhanced image with the modified unsharp masking with the DC3T and C equal to 4.0	91
3.22 Enhanced image with the modified unsharp masking with the DC3T and C equal to 7.0	92
3.23 Enhanced image with the modified unsharp masking with the DCT and C equal to 4.0	93
3.24 Enhanced image with the modified unsharp masking with the DCT and C equal to 7.0	94
3.25 The HVS spatial frequency sensitivity models for the DFT, the DCT, and the modified HVS model for the DCT	95
3.26 Image enhancement based on the human visual system	95
3.27 Enhanced image with HVS-filtering with the DCT	96
3.28 Enhanced image with the modified HVS-filtering with the DC3T	97
3.29 Enhanced image with HVS-filtering with the DFT	9 8
3.30 Enhanced image with HVS-filtering with the RDFT	99

ix

figure	Page
3.31 Enhanced image with conventional alpha-rooting by the overlap-save method with the SRDFT, α equal to 0.7 and N equal to 32	e 100
3.32 Enhanced image with the modified unsharp masking by the overlap- method with the DC3T, C equal to 7, and N equal to 32	save 101
3.33 Enhanced image with the modified HVS-filtering by the overlap-save method with the DC3T and N equal to 32	102
3.34 Enhanced image with the modified HVS-filtering by the overlap-save method with the DC3T, and N equal to 16	103
3.35 Comparison of HVS-filtering with modified unsharp masking	104
4.1 Input signal f(x), its first derivative f'(x), and its second derivative f''(x) for a typical 1-D edge	119
4.2 Graph of $-d(x,y) = -\nabla^2 h(x,y)$ and its Fourier transform, $-D_c(f_x, f_y)$	120
4.3 Graph of 1-D cross-section of d(x,y) and its Fourier transform	122
4.4 Frequency distributions of different transforms with a typical image	124
4.5 1-D graph of two types of bandpass masks	125
4.6 2-D graph of bandpass masks of different types and different transform	m đ 27
4.7 Zero-crossing and its slope in 1-D	131
4.8 The perfect and noisy checkerboard images	132
化化物学 医结核性性 化合成合物 化乙酰氨酸 化氟化氨基酸 化分子 医白色的 化化合成 医视觉的复数形式 化合成分子 网络新闻教师 化合成合成合成合成合成合成合成合成合成合成合成合成合成	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

4.9 Zero-crossings with mask type II, the DCT or the DC3T, block size of 16, parameter values from 0.05 to 0.40 with increment of 0.05 and 4 pixels overlapping..... ...134

x

xi

Figure	Page
4.10 Zero-crossings with the DCT or the DC3T; the first two rows: mask type I, block size of 16, parameter values from 0.025 to 0.20 with increment of 0.025 and 4 pixels overlapping, the second two rows: mask type II, block size of 32, parameter values equal to 0.012, 0.015, 0.02, 0.028, 0.04, 0.0625, 0.11, 0.25 and 4 pixel overlapping	135
4.11 Zero-crossings with mask type II, the RDFT, block size of 16, parameter values from 0.05 to 0.40 with increment of 0.05 and 4 pixels overlapping	136
4.12 Zero-crossings with the RDFT; the first two rows: mask type I, block size of 16, parameter values from 0.025 to 0.20 with increment of 0.025 and 4 pixels overlapping, the second two rows: mask type II, block size of 32, parameter values equal to 0.012, 0.015, 0.02, 0.028, 0.04, 0.0625, 0.11, 0.25 and 4 pixel overlapping	137
4.13 Zero-crossings with mask type II, the SRDFT, block size of 16, parameter values from 0.05 to 0.40 with increment of 0.05 and using 4 pixels overlapping	138
4.14 Zero-crossings with the SRDFT; the first two rows: mask type I, block size of 16, parameter values from 0.025 to 0.20 with increment of 0.025 and 4 pixels overlapping, the second two rows: mask type II, block size of 32, parameter values equal to 0.012, 0.015, 0.02, 0.028, 0.04, 0.0625, 0.11, 0.25 and 4 pixel overlapping	139
4.15 Zero-crossings after thresholding: counting from left to right starting at the top, each image corresponds to the entries of the first row to the 8th row of Table 4.1	140
4.16 Zero-crossings after thresholding: counting from left to right starting at the top, each image corresponds to the entries of the 9th row to the 13th row of Table 4.1	141

λ.

	xi	i
	Figure	ıge
	4.17 Illustrates the edges obtained by the 11×11	
	Marr-Hildreth zero crossing of Laplacian operator set for three	
	different zero crossing thresholds and three different standard	
	deviations for the associated Mexican hat filter [59]14	3
	4.18 Illustrates the directional derivatives edge operator for a window	
	size of 11×11 and deciding that the true	
	gradient is nonzero when the estimated gradient is higher than the	
	thresholds of 12, 14, 16, or 18 [59]14	4
	4.19 Compares the directional derivative edge operator with the	
	Marr-Hildreth edge operator and the Prewitt edge operator.	
	The thresholds chosen were the best possible ones [59]14	5
ž	4.20 (a) The zero-crossings obtained from the Marr-Hildreth	
	implementation of the $\nabla^2 G$ operator with $\sigma = 2.5$.	
	(b) Zero-crossings that remain after thresholding so as to	
	equalize the conditional probabilities [60]14	6
	4.21 The results of transform edge detection with original image of "girl256", the DCT or DC3T, bandpass mask of type II, block size of 32, $\sigma^2 = 0.012$ and threshold value of 16: top-left, no overlapping; top-right, 2 pixels overlapping; bottom-left, 4 pixels overlapping	7
	4.22 The results of transform edge detection with original image of "catbrain", the DCT or DC3T, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 without overlapping	8
	4.23 The results of transform edge detection with original image of	
	"catbrain", the DCT or DC3T, bandpass mask of type II, block size	
	of 32, $\sigma^2 = 0.0625$, threshold value of 7 and with	
	1 pixel overlapping14	9
	에는 것이다. 이렇게 가장에 통해 통해 통해 물건을 통해 있는 것을 가지 않는 것은 것은 것을 것을 것을 것을 것을 것이다. 가지 않는 것은 것은 것을 것을 것이다. 가지 않는 것은 것은 것은 것은 같은 것이다. 이렇게 같은 것은	
	이 가지 않는 것 같은 것 같	
	는 방법은 이상 전통 전화적으로 가지 비용하는 것이 가지 않는 것 같은 것 같은 것 같은 것이 있다. 가지 않는 것은 것을 알았는 것이 같은 것을 알았는 것이 있다. 이 사이지 아이들은 것은 것은 것은 것은 것은 것은 것은 것이 같은 것이 있다. 이상에 가지 않는 것은 것은 것은 것이 같은 것이 같은 것이 같은 것이 같은 것이 같이 있다. 것은 것은 것은 것이 있	
	그렇게 있는 것은 것을 통해야 한국을 통해할 수 있는 것을 가 없다. 그는 것은 것은 것은 것은 것은 것은 것을 가지 않는 것을 가지 않는 것을 가지 않는 것을 가지 않는 것을 가 있다. 같은 것은	
	같은 것은	ne e La re
ā.	· 1999년 1988년 1987년 1987년 1988년 1989년 1989년 1997년 1997년 1987년 1987년 1988년 1988년 1988년 1988년 1988년 1997년 1997년 1997년 1997년 - 1997년 1987년 1997년 1987년 1987년 1987년 1997년 1997년 1997년 1987년 1987년 1987년 1987년 1987년 1987년 1987년 1997년 1997년 - 1997년 1987년 1	
	는 것 같은 것 같	** - **

Figure

alle although all all a

. 12.

 $\begin{array}{l} \sum_{i=1}^{n-1} \frac{1}{2} \sum_{i=1}^{n-1}$

4.24 The results of transform edge detection with original image of "catbrain", the DCT or DC3T, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 and with 4 pixels overlapping	150
4.25 The results of transform edge detection with original image of "catbrain", the DCT or DC3T, bandpass mask of type II, block size of 32, $\sigma^2 = 0.04$, threshold value of 7 and with 4 pixels overlapping	151
4.26 The results of transform edge detection with original image of "girl256", the SRDFT, bandpass mask of type II, block size of 32, $\sigma^2 = 0.012$ and threshold value of 16: top-left, no overlapping; top-right, 2 pixels overlapping; bottom-left, 8 pixels overlapping	152
4.27 The results of transform edge detection with original image of "catbrain", the SRDFT, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 without overlapping	153
4.28 The results of transform edge detection with original image of "catbrain", the SRDFT, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 and with 1 pixel overlapping	154
4.29 The results of transform edge detection with original image of "catbrain", the SRDFT, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 and with 4 pixels overlapping	155
4.30 The results of transform edge detection with original image of "catbrain", the SRDFT, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 and with 8 pixels overlapping	156

Page

Figure

Page

.157
.158
.159
.160
.161
.163

ABSTRACT

Aghagolzadeh, Sabzali. Ph.D., Purdue University. August 1991. Blockwise Transform Image Coding, Enhancement and Edge Detection. Major Professor: Okan K. Ersoy.

The goal of this thesis is high quality image coding, enhancement and edge detection. A unified approach using novel fast transforms is developed to achieve all three objectives. Requirements are low bit rate, low complexity of implementation and parallel processing. The last requirement is achieved by processing the image in small blocks such that all blocks can be processed simultaneously. This is similar to biological vision. A major issue is to minimize the resulting block effects. This is done by using proper transforms and possibly an overlap-save technique. The bit rate in image coding is minimized by developing new results in optimal adaptive multistage transform coding. Newly developed fast trigonometric transforms are also utilized and compared for transform coding, image enhancement and edge detection. Both image enhancement and edge detection involve generalized bandpass filtering with fast transforms. The algorithms have been developed with special attention to the properties of biological vision systems.

INTRODUCTION

This thesis is concerned with blockwise multistage transform image coding, transform image enhancement and edge detection techniques. Special emphasis is given to develop algorithms which perform similarly to the human visual system.

Image coding is a process or a sequence of processes in order to reduce . the total number of bits in image representation, and simultaneously to minimize the degradation of the decoded image. Image coding has applications in efficient image communications and storage. Many different methods and techniques have been reported in literature, and some good surveys can be found in Refs. [1,2,3]. In general, all techniques can be grouped in two major categories: predictive coding and transform coding. Predictive coding techniques are carried out in the spatial (image) domain, while transform techniques, in contrast, are applied in the transform (frequency or sequency) domain. In predictive coding, the correlation between adjacent pixels is used to estimate or to predict the incoming pixel value, given values of the past pixels. In transform coding, the image is first transformed; then quantization, to be explained later, is applied; and finally in the decoding process, the quantized image is inverse transformed back to the image domain. It is also possible to combine these two categories in so called hybrid coding, which exploits the advantages of both approaches to achieve better results.

Image enhancement consists of methods to enhance some desired features in a given image, either to make the image more satisfactory to the viewer or to help the machine or the human being to find and classify relevant image features easily and efficiently. A survey of digital image enhancement methods can be found in Ref. [4]. One class of image enhancement methods includes gray scale modification, deblurring and smoothing. Transform techniques form another class and are among the major topics in this research. Since both transform coding and transform enhancement require applying transform to the image, it would be very efficient, in the sense of computational complexity, to perform both together with good match between successive processes when both processing are desired to be applied.

Transform edge detection is also studied to understand its compatibility with blockwise processing and transform image coding and enhancement. A major issue is whether edge detection is compatible with transform image enhancement and whether they can all be done blockwise effectively together with image coding.

The outline of this thesis is as follows: Chapter 1 discusses briefly the coding methods, particularly, transform coding, and some basic concepts for coding, such as quantization and adaptive transform coding, a survey of image enhancement methods in the spatial domain and edge detection.

In Chapter 2, an optimal method is developed for adaptive multistage image transform coding. It is shown that considerable improvement can be achieved with little increase in the complexity of coding processes.

Chapter 3 introduces some fast relevant transforms for image enhancement and involves comparative experimental results with three transform enhancement techniques.

Transform edge detection is discussed in Chapter 4. It is shown that different fast transforms can be used for edge detection with considerably good quality results and low computational complexity as well as parallelism through simultaneous processing of blocks. Experimental results are compared to other edge detection techniques based on bandpass filtering.

Finally, in Chapter 5, the conclusions and the future research topics as an extension of the present research results are discussed.

CHAPTER 1 A REVIEW OF BASIC PRINCIPLES

This chapter discusses coding methods, particularly, transform coding. Some basic concepts for coding such as adaptive transform coding and quantization are reviewed. The chapter also includes a brief discussion of image enhancement in the spatial domain, the properties of the human visual system, and edge detection.

轮 网络白色白白白白白白白

1.1 Predictive Coding

In predictive coding the strong correlation between adjacent pixels, either in the spatial domain or in the temporal domain, is exploited. Predictive coding attempts to estimate or to predict approximately the pixel values to be coded from the available information about the previously coded pixels. Then the difference or the error in prediction is quantized and coded. In the decoding procedure, the parameters of the prediction and the decoded difference signal are used to reconstruct the pixel value. If the difference signal is coded with 2 levels (1 bit per sample) the predictive coding is called delta modulation (DM). Otherwise, with higher number of levels, it is called differential pulse code modulation (DPCM). In general, performance of the DM coding depends on the quantizer step size. Large step size is good for following large transitions in the image such as sharp edges, but high quantization noise in flat areas results. On the other hand, small step size results in less quantization noise in flat areas, but the image is smoothed and sharp edges is smeared off. In order to solve this problem, the step size can be made adaptive with respect to the local signal values in a very small neighborhood based on the slope of the signal or some previously coded bits. For DPCM coding, there is no such problem since large number of bits (or levels) is used for quantizing the difference signal. In general, there are many different kind of predictors for

both DM and DPCM coding. Some of them are linear predictor, intrafield predictor, interframe predictor, motion estimator, block matching, etc. [5].

4

1.2 Transform Coding

Transform coding forms another class of coding. In transform coding the signal is first transformed to another domain, usually called frequency or spectral domain or transform domain, by a linear (not necessarily unitary) transformation. This transformation provides less correlated coefficients in the spectral domain that can be quantized independently. Another characteristic of such a transformation is that it packs most signal energy in a few coefficients in the spectral domain. Two measures of efficiency of a given transform is the degree of decorrelation and the degree of energy packing.

The optimal transform in this regard is the Karhunen-Loeve transform (KLT) [6]. Unfortunately, the basis for this transform are signal dependent and it is difficult to compute them in real time. However there are other transforms which are very close to the KLT and easier to be implemented. Some of these transforms are the discrete Fourier transform (DFT), the Walsh-Hadamard transform (WHT) [7], the discrete cosine transform (DCT) [8], the scrambled real discrete Fourier transform (SRDFT) [9] and the discrete cosine-III transform (DC3T) [10]. These transforms are described in the following chapters.

Transform coding gives better performance in the sense of compression and reconstruction error, however it does involve more computational complexity. For reducing the computational complexity, fast transforms have been developed for the above transforms. In general, transform coding is preferred to predictive coding for bit rates below 2 or 3 bits per pixel. Like predictive coding, some adaptive methods have been developed for transform coding in order to increase the efficiency, and they are discussed in the following section in this chapter.

Almost any natural image has areas with different amount of details, i.e., different distributions for its pixel values. Some have flat areas with little difference in pixel values and easily can be coded by a few bits, and others may have sharp edges in different directions and more bits are required to code them. These regions are called low or high "activity" regions, respectively. For more efficiency, the characteristics of these regions are exploited. This is

one reason why transform image coding is usually carried out in blocks. For this purpose the image is divided into square blocks of size usually equal to an integral power of 2 (4,8,16,...), and then each block is transformed independently. The transform coefficients of each block fall in either low or high "activity" classes. There are some problems to be considered in this procedure. First, If the block size is too small, correlations among pixels are not properly used. Correlations among pixels exist up to a distance around 10-20 pixels, depending on the degree of activity, the kind of image, and the kind of sampling. Secondly, with decreasing block size, the number of blocks for a given image is increased, and this problem increases the number of bits for overhead information which needs to be transmitted. On the other hand, small block size reduces the complexity of implementing. Another problem is that, at low bit rates, block effects become visible. These considerations lead to block sizes such as 8,16, or 32.

1.3 Adaptive Transform Coding

There are a number of adaptive schemes for transform image coding. Adaptive techniques increase the efficiency of coding but also increase the complexity of implementation. Adaptivity may be applied to the selection of transform [11], the number of coefficients to be coded (zonal coding), the value of coefficients to be coded (threshold coding), or the kind of quantizer and quantizer levels. Here, we discuss a well-known adaptive method developed by Chen and Smith [13], which is also used in Chapter 2 for coding. This method is very efficient in both monochrome and color image coding. A block diagram of this method is shown in Fig. 1.1. In this method, the image is divided into blocks of the same size, and the transform is applied to each block. Then a measure of "activity" for each class is calculated by adding squared values of all coefficients except the DC coefficient. After sorting the measured values for their "activity", blocks are classified into several groups, usually called classes, with equal number of blocks in each group. The variance of coefficients is estimated within each class, and based on the variance matrix, the bit allocation map and the normalization coefficients are found for each class. Then, the normalized coefficients are quantized and coded. At the receiver, after receiving the bit map, the class map, and the norm factor as overhead information, the decoding procedure is applied and finally the image pixel

5

values are reconstructed by inverse transforming. This adaptive method increases the overhead information very little, but decreases the reconstruction error, in the sense of mean square error, to be explained later, as much as 25%.

1.4 Scalar Quantization

In scalar quantization, a continuous input random variable e is converted to a discrete output random variable \hat{e} that can take L levels, $r_1, r_2, ..., r_L$. we define L + 1 decision levels, $t_1, t_1, ..., t_{L+1}$, where usually $t_1 = -\infty$ and $t_{L+1} = \infty$. Referring to Fig. (1.2), the output \hat{e} takes the value of r_k if the input e is between the decision levels t_k and t_{k+1} . The mean square error (MSE) for the quantizer is defined as

$$MSE = E\left[\left(e - \hat{e}\right)^{2}\right] = \int_{-\infty}^{\infty} (e - \hat{e})^{2} p_{e}(e) de$$
$$= \sum_{k=1}^{k=L} \int_{t_{k}}^{t_{k+1}} (e - r_{k})^{2} p_{e}(e) de$$
(1.1)

where $p_e(.)$ is the probability density function of e.

In order to minimize the MSE, Eq. (1.1) is differentiated with respect to r_k and t_k . After equating the results to zero, the following relations are found:

$$t_{l} = \frac{r_{l} + r_{l-1}}{2} \text{ for } l = 1, ..., L+1$$
 (1.2)

and

Ĩ

$$h_{l} = \frac{\int_{t_{i}}^{t_{i+1}} ep_{e}(e)de}{\frac{t_{i}}{t_{i+1}}} \quad \text{for } l = 1, ..., L \quad (1.3)$$

$$\int_{t_{i}}^{t_{i}} p_{e}(e)de$$

$= \mathbf{E} \left[\mathbf{e} \mid \mathbf{e} \in [t_l, t_{l+1}) \right]$

Eqs. (1.2) and (1.3) are the necessary conditions that should be satisfied for the optimum mean square quantizer. Eq. (1.2) states that the input threshold level t_i lies halfway between two adjacent output levels r_i and r_{i+1} , and Eq. (1.3) states that the output level r_i lies at the center of mass of the probability density between the input levels t_i and t_{i+1} .

7

Eqs. (1.2) and (1.3) together form a nonlinear system of equations. They can be solved by an iterative procedure that is due to S. P. Lloyd and published by J. Max and called Lloyd-Max quantizer [12]. The procedure is as follows:

- 1. Choose a set of initial values for the output levels r_k , k = 1, ..., L.
- 2. Calculate the input threshold values t_k , k = 1, ..., L+1 by Eq. (1.2).
- 3. Calculate the new values for r_k , k = 1,...,L by Eq. (1.3).
- 4. Goto the step 2.

The above algorithm can be stopped when the change in new values for r_k is small enough to be neglected, and its success depends on the initial chosen set.

When the probability density function, $p_e(.)$, is uniform, the optimum mean square error quantizer is called the uniform optimal quantizer, or the linear quantizer. In this case Eq. (1.3) has the following form

$$r_l = \frac{t_l + t_{l+1}}{2}$$
 for $l = 1, ..., L$. (1.4)

1.5 Visual Blockwise Processing

The main advantages of block processing are reduced complexity of transforms, adaptivity to image details and parallel processing of the blocks. It is interesting to compare this type of processing to the human visual system. Eyes can receive light from a large angle, about 120 degrees, but can not focus at two different points simultaneously. Thus, the angle of detailed vision is limited. It has been reported in Ref. [14] that the resolution of human visual system decreases from the center of fixation, and within a cone of $2^{\circ} \times 2^{\circ}$ there is a "fairly" good detailed vision. It can be concluded that good detailed vision

is limited to 2° for average observers. Let us consider an example. Suppose an observer is watching a fairly detailed image of size 512×512 pixels with a physical size of 4×4 inches in a distance of 12 inches. Then the observer looks at a block of size 28×28 pixels within a cone angle of $1^{\circ}\times1^{\circ}$. If the observer wants to look at another location in the image, he or she has to change the angle of vision, usually continuously. It is also known that the maximum spatial bandwidth of the human visual system is about 64 cycles/degree. All of these observations suggest that blockwise image processing is similar to what the human eyes do. In Chapter 3, more properties of human eyes will be considered.

8

1.6 Image Enhancement

Image enhancement involves processing of an image to make the image more satisfactory to the viewer. There are a number of techniques for digital image enhancement, and a survey of them can be found in Refs. [4,15]. All techniques can be grouped into two categories: spatial and transform domain techniques. A number of transform domain techniques will be discussed in Chapter 3. Here we briefly review some techniques in the spatial domain.

Two kinds of operations may be applied for image enhancement in the spatial domain: gray level scaling and spatial filtering. In gray level scaling, the gray level value of each pixel is mapped into another value according to a function f(.). The form of the function f(.) depends on the kind of desired modification, and examples of this function and their applications are given in Table. 1.1. Histogram equalization is one special case of gray level scaling techniques where the function f(.) depends on the distribution of the gray level values of the input image, and a uniform histogram for the output image is desired. In spatial filtering, the gray level value of each pixel is changed by local operations on neighborhoods of input pixels. Noise smoothing, median filtering, unsharp masking, low-pass filtering, bandpass filtering and high-pass filtering are some examples of this technique. For any of these methods, an array of numbers is used as a mask to decide the value of a pixel from the values of other pixels in the neighborhood. The performance of these techniques depends on the size of array and design of numbers in the mask array, and this issue was discussed in Ref. [4].

19月1日発展の業務営業は19月1日の10月1日

 1216日建筑家

1.7 Edge Detection

Edge detection is very useful in many applications such as image segmentation, registration and object identification. Edge points are pixels at which abrupt gray level changes occur which may reflect the change in surface orientation, depth or physical properties of materials. Edge detection has also been used in image compression in which only edges are coded. There are different kinds of gray level changes in real images, and therefore there is no single definition for edges.

There are many different methods for edge detection. The two most known family of edge detection algorithms are gradient-based and Laplacianbased methods. A survey of edge detection methods can be found in [27,40,66,67]. We will discuss blockwise transform edge detection methods as compared to other methods in Chapter 4.

1.7 Motivation for This Research

There are many applications in image processing in which image transmission or storage is required. Satellite communications, remote sensing, biomedical imaging are some examples. Regarding high demand for fine quality images, a large bandwidth or memory is required to transmit or to store images. Also for many of the above mentioned applications, enhancement of original images is necessary. So it is desirable to improve the efficiency of coding and enhancement of images simultaneously, while reducing the complexity of implementation. Transform coding is considered as a efficient approach to image coding. Therefore, it is interesting to improve the performance of image coding and image enhancement based on transform techniques and to investigate transform edge detection following image enhancement. Blockwise processing also allows real-time implementation through parallel processing and adaptivity. It is also similar to the properties of the human visual system. Thus, it is desirable to improve the performance of block transform coding and to implement image processing in blocks.

In transform image coding, we have developed a technique called optimal multistage image transform coding, which provides considerable improvement in performance of transform image coding. In parallel with this research, we investigated to improve blockwise image enhancement techniques with the same fast transforms. This is followed by blockwise transform edge detection. In this way, we have had the goal of developing a unified approach to most tasks in image processing through blockwise fast transform processing.

신물은 물고 싶었.

있다. (See All Market) (Market)

i ingen i



Figure 1.1. Adaptive cosine transform coding system [13].

3e St





April & and a second second

8 8 Star

de la

1. Contrast stretching	$f(u) = \begin{cases} \alpha u, & 0 \le u < a \\ \beta(u-a) + v_{\bullet}, & a \le u < b \\ \gamma(u-b) + v_{\bullet}, & b \le u < L \end{cases}$	The slopes α , β , γ determine the relative contrast stretch.
2. Noise clipping and thresholding	$f(u) = \begin{cases} 0, & 0 \le u < a \\ \alpha u, & a \le u \le b \\ L, & u \ge b \end{cases}$	Useful for binary or other images that have bimodal distribution of gray levels. The a and b define the valley between the peaks of the histogram. For $a = b = t$, this is called thresholding.
3. Gray scale reversal	f(u)=L-u	Creates digital negative of the image.
4. Gray-level window slicing	$f(u) = \begin{cases} L, & a \le u \le b \\ 0, & \text{otherwise} \end{cases}$	Fully illuminates pixels lying in the interval [a, b] and removes the background.
5. Bit extraction	$f(u) = (i_n - 2i_{n-1})L$ $i_n = Int\left[\frac{u}{2^{B-n}}\right], n = 1, 2, \dots, B$	 B = number of bits used to represent u as an integer. This extracts the nth most-significant bit.
6. Bit removal	$f(u) = 2u \mod (L+1), \qquad 0 \le u \le L$	Most-significant-bit removal.
	$f(u) = 2 \ln \left[\frac{u}{2} \right]$	Least-significant-bit removal.
7. Range compression	$v = c \log_{10} (1 + u), u \ge 0$ $c \stackrel{\Delta}{=} \frac{L}{\log_{10} (1 + L)}$	Intensity to contrast transformation.

Table 1.1. Some gray scaling functions for image enhancement [15].

Ì.

13

έą.,

CHAPTER 2 OPTIMAL ADAPTIVE MULTISTAGE IMAGE TRANSFORM CODING

14

2.1 Introduction

是你们还在这些你们的就是你们是你们的这些就不能就能是这些你的的情况。"

Transform coding is widely used in coding of images since it gives a very high compression ratio. The effectiveness of transform coding has a lot to do with the properties of decorrelating the pixel values and packing the energy of the signal in a few transform coefficients. Based on these two criteria, the Karhunen-Loeve transform (KLT) is the optimal transform for image coding [16]. However, the KLT is signal dependent and difficult to compute in real time. The discrete cosine transform (DCT) is among the best fast transforms to approximate the KLT in image coding [8]. One technique for improving the efficiency of image coding is to apply adaptivity to the coding procedure by classifying image blocks in a number of classes. An efficient adaptive algorithm for image coding was proposed by Chen and Smith [13].

In this chapter, we discuss an optimization technique of transform image coding in the form of a multistage procedure in which the error signal resulting from the quantization of the previous stage is input to the following stage. Multistage transform coding thus involves transform domain quantization in a number of stages such that each stage attempts to correct the errors in the previous stage. The technique to be discussed is different from progressive image coding even though there is some degree of similarity. In progressive image coding, first a low-grade version of the image is sent, and then the image is refined by sending more information in the following stages. In the technique discussed in this chapter, the bits are allocated to the pixels of each stage when the number of stages and the total bit rate are given. Consequently, the stages are coupled, unlike progressive image coding. However, the present technique can also be used in progressive image coding if each image sent is coded in multistages.

A number of different techniques for progressive image coding in both

spatial and transform domains have been discussed by Tzou [17], Wang and Goldberg [18,19]. In these techniques, the coefficients of each stage are quantized by a predetermined average rate and the number of stages are increased until satisfactory image reconstruction is obtained at the receiver. So far, no adaptive method has been reported in order to adjust the number of bits for each stage based on the statistics of the coefficients of different stages, and for a total given bit rate.

The method to be discussed in this chapter involves optimal adaptive multistage transform coding with a fixed total number of bits per pixel and a fixed number of stages. It is optimal in the sense that it minimizes the total final reconstruction error with the given total number of bits and stages. The statistics of the coefficients in different stages are used to optimize the division of the total number of bits among different stages. The adaptivity introduced does not significantly add to the complexity of the coding system since it utilizes the information that is necessary for any kind of multistage transform coding. Simulation results have shown a considerable percentage decrease in reconstruction error in a large number of test images. In addition, the remaining error image is more noise-like than the error image in one stage coding, especially with reduced error around the edges. Smooth areas of the image look smoother with multistage coding than one stage coding as well. These are believed to be the main reasons why multistage transform coding gives subjectively more pleasing results than one stage coding at the same bit rate.

There are a number of subjective and objective error measures to quantify the quality of image reconstruction, but the mean square error (MSE) is the most widely used. The MSE is also the measure in this chapter to be used to compare experimental results. The experimental results will also be discussed in terms of subjective performance.

The chapter consists of 6 sections. In Sec. 2.2, the new proposed method is introduced, and a mathematical expression is derived for the total final reconstruction error which is to be minimized during bit allocation and coding. This expression is based on the mean square error. The optimal bit allocation for different stages to minimize the quantization error is explained in Sec. 2.3 by using the statistics of the coefficients in different stages. In Sec. 2.4, the experimental results with the discrete cosine transform (DCT) are discussed with a number of images and rates, as well as with one class and multiclass adaptive procedures. Sec. 2.5 is discussion concerning

implementation of multistage transform coding. Sec. 2.6 is conclusions.

2.2 The MSE Function for the Proposed Method

The block diagram for adaptive multistage transform coding is shown in Fig. 2.1. The transform coefficients of the first stage are assumed to have little correlation so that they are quantized and coded independently with an optimal bit map for the first stage to be considered. The two dimensional error signal resulting from the first stage quantization is fed to the second stage and subsequently quantized and coded with another optimal bit map. This procedure is continued for the given total number of stages.

Next, we derive a mathematical expression for the total final reconstruction error based on the mean square error (MSE) measure. We assume that unitary transforms are used for transform coding. Then, the variance of the reconstruction error is equal to that introduced during the quantization of coefficients in the transform domain [20].

Referring to Fig. 2.1, the following notations are defined:

- n : The number of stages.
- E_k : The coefficient matrix of size N×N as input to stage k+1, k = 0,1,...,n-1.
- \dot{E}_k : The matrix of size N×N for the quantized coefficients as output of stage k+1, k = 0,1,...,n-1.
- e_{kij} : The ijth coefficient of the matrix E_k .
- $\hat{\mathbf{e}}_{kij}$: The ijth coefficient of the matrix \mathbf{E}_k .
- b_{kij} : The number of bits used to quantize e_{kij} .
- $f_k(b_{kij})$: The mean square distortion function of the b_{kij} -bit quantizer for unity variance input (see Sec. 2.3).
 - σ_{kij}^2 : The variance of e_{kij} .
 - $\hat{\sigma}_{kij}$: The variance of $\hat{\mathbf{e}}_{kij}$.

There are different kinds of quantizers such as optimum mean square (Lloyd-Max) and uniform optimal quantizer [12]. The optimum mean square quantizer is used in this chapter. Suppose e_{kij} and \hat{e}_{kij} are the input and the output of the optimum mean square quantizer. They have the following properties [15]:

$$\mathbf{E}(\mathbf{e}_{kij}) = \mathbf{E}(\hat{\mathbf{e}}_{kij}) \longrightarrow \mathbf{E}(\mathbf{e}_{kij} - \hat{\mathbf{e}}_{kij}) = \mathbf{E}(\mathbf{e}_{(k+1)ij}) = \mathbf{0}$$
(2.1)

17

$$\mathbf{E}(\mathbf{e}_{kij}\hat{\mathbf{e}}_{kij}) = \mathbf{E}(\hat{\mathbf{e}}_{kij}^{2}) \longrightarrow \mathbf{E}\left(\hat{\mathbf{e}}_{kij}(\mathbf{e}_{kij}-\hat{\mathbf{e}}_{kij})\right) = \mathbf{0}$$
(2.2)

$$\hat{\sigma}_{kij}^{2} = \left(1 - f_{k}(b_{kij})\right) \sigma_{kij}^{2}$$
(2.3)

The following equation can be derived from Eqs. (2.1),(2.2) and (2.3):

E
$$(e_{kij}\hat{e}_{kij}) = E(\hat{e}_{kij}^2) = \hat{\sigma}_{kij}^2 = (1-f_k(b_{kij}))\sigma_{kij}^2$$
 for $k = 1, 2, ..., n-1$. (2.4)

Referring to Fig. 2.1, the final reconstructed image is formed by taking the inverse transform of $(\hat{E}_0 + \hat{E}_1 + \cdots + \hat{E}_{n-1})$. Therefore the mean square error (MSE) is given by

$$MSE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} E \left(e_{0ij} - (\hat{e}_{0ij} + \hat{e}_{1ij} + \cdots + \hat{e}_{(n-1)ij}) \right)^2. \quad (2.5)$$

Using Eq. (2.2) and the fact that the average value of the coefficients in each stage, excluding the DC coefficient of the first stage, is zero, the following is obtained at stage k:

$$\mathbf{E}\left(\left(\mathbf{e}_{kij}-\hat{\mathbf{e}}_{kij}\right)^{2}\right)=\sigma_{(k+1)ij}^{2}=\sigma_{kij}^{2}-\hat{\sigma}_{kij}^{2}$$
(2.6)

Applying Eq. (2.6) to stage k+1, we have

$$\mathbb{E}\left(\left(e_{(k+1)ij} - \hat{e}_{(k+1)ij}\right)^{2}\right) = \sigma_{(k+1)ij}^{2} - \hat{\sigma}_{(k+1)ij}^{2} = \sigma_{kij}^{2} - \hat{\sigma}_{kij}^{2} - \hat{\sigma}_{(k+1)ij}^{2}$$
(2.7)

By iterating Eq. (2.7) and using Eq. (2.4), the MSE for n stages becomes

$$MSE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left[\sigma_{0ij}^2 f_0(b_{0ij}) + \sigma_{1ij}^2 \left(f_1(b_{1ij}) - 1 \right) + \cdots + \sigma_{(n-1)ij}^2 \left(f_{n-1}(b_{(n-1)ij}) - 1 \right) \right].$$
(2.8)

Eq. (2.8) is the objective function that is to be minimized in order to achieve the minimum mean square error. The procedure of bit allocation in order to minimize this function is given in the next section.

2.3 Error Models and Optimal Bit Allocation

In general, an analytic expression for the quantizer error is desirable. Usually, such an expression is given in terms of the variance of the input sequence to the quantizer, the number of bits (or the number of levels) used for quantization, and some parameters that depend on the distribution of the input. A closed form expression for the MSE is very difficult to derive, and most reported results have been obtained either by numerical or approximate means. In the case of the optimum mean square (Lloyd-Max) quantizer, the MSE is usually expressed in the form of $\sigma_{kij}^2 f_k(b_{kij})$ where $f_k(b_{kij})$ is a function of b_{kij} and the probability density function (pdf) of the input signal to the quantizer. One such expression for $f_k(b_{kij})$ for Gaussian distribution is given by [12]

$$f_{k}(b_{kij}) = \begin{cases} 1.32 \ (2^{-1.74b_{kij}}) & \text{for } b_{kij} \text{ around } 2\\ 2.21 \ (2^{-1.96b_{kij}}) & \text{for } b_{kij} \text{ around } 5.17 \end{cases}$$
(2.9)

Another approximate model in the case of Gaussian distribution is given by [21]

$$f_{k}(b_{kij}) = \begin{cases} 2^{-1.57b_{kij}} & \text{for } b_{kij} \leq 2.32 \\ \\ \frac{2.698 \ 2^{2b_{kij}}}{(2^{2b^{kij}} + 0.8532)^{3}} & \text{for } b_{kij} \geq 2.32 \end{cases}$$
(2.10)

Some other error functions have been reported for both Gaussian and Laplacian distributions in Ref. [15] and is given in Table 2.1.

All of the above models are either for the Gaussian or the Laplacian distribution. In practice, the input to the quantizer usually has neither Gaussian nor Laplacian distribution exactly but some distribution close to one of them.

Recently it was reported that most of AC coefficients for the first stage of the DCT transform have Laplacian pdf [22]. It was also mentioned that the DC coefficients have a pdf close to Gaussian. We performed the Kolmogorov-Smirnov(K-S) [23] test for the coefficients of the second stage. The results indicate that the 8-bit quantization error for the DC coefficients has a pdf close to the uniform distribution. All the AC coefficients which have been allocated 2 or more bits have a pdf closest to Gaussian whereas most coefficients which have been allocated 1 bit have a pdf closest to Gaussian. Of course, those coefficients which have not been allocated any bit at the first stage have a Laplacian pdf in most cases. Overall, a large number of coefficients which will receive non-zero bits in the second stage have a Gaussian pdf. Therefore in our simulations, we assume Gaussian pdf for the second stage and use the error model given in Table 2.1.

Having the error models for each stage, the total final error given in Eq. (2.8) can be minimized through an optimal bit allocation procedure. Coefficients in each stage usually have different variances, and their variances are also different from stage to stage. Therefore, different number of bits should be assigned to each coefficient. The major constraint that should be satisfied is that the total number of bits is fixed. There are a number of methods for bit allocation, and they are not necessarily optimal in minimizing the MSE. Some methods assume the number of bits to be a continuous variable in order to get an optimal and closed form expression, but the result has to be rounded to the nearest integer and is no longer optimal. The procedure for obtaining optimal non-integer number of bits was discussed in Ref. [24]. In this chapter, we use marginal analysis described in Ref. [25] to develop an optimal method with integer number of bits. The piecewise error models given in Table 2.1 are strictly convex functions and guarantee that the

global minimum is achieved. Here, we give the steps for bit allocation with 2 stages. The generalization for more stages is straightforward. The steps involved in bit allocation according to marginal analysis are as follows:

- 1. Set $b_{kij} = 0$, for k = 0, 1 and i, j = 0, 1, ..., N-1.
- 2. Calculate the marginal return, Δ_{kij} , which is the reduction in the total final error given by Eq. (2.8) if 1 bit is assigned to the coefficient e_{kij} , for k=0,1, and i,j=1,...,N.
- 3. Allocate one bit to the coefficient e_{kij} which has the largest marginal return Δ_{kij} .
- 4. If the total number of assigned bits is equal to or greater than the total number of bits, stop; otherwise go to Step 2 to decide for the next bit.

If ties happen in step 3, the same procedure is repeated among the coefficients which have the same value for Δ_{kij} by assigning another bit to these coefficients and looking for the winner.

The above procedure for bit allocation can be applied to find the bit map that minimize the total final reconstruction error for the multistage transform coding if the multiclass adaptive method is not used for each stage. In deriving the estimated total final error given in Eq. (2.8), we assume that the coefficient e_{kij} is the resulting error of quantizing the coefficient $e_{(k-1)ij}$. On the other hand, if the multiclass adaptive method is used, the class map of each stage is possibly different, so the above assumption does not hold.

For multiclasses, we introduce another method of optimization to minimize the total error. In this method, we first derive a relation between the total average rate R, and the average rate for each stage R_k , k = 0, ..., n-1. When the average rate of each stage is known, the bit allocation procedure for each stage can be done independently. It is also possible to use different number of classes for the following stages since the spectra in those stages are more flat than the first stage.

First we will find the relation between R and R_k , k = 0,...,n-1 for n = 2 (two stages). Then, we will show that for $n \ge 3$ the procedure is straightforward. For n = 2, the problem is
minimize
$$\frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left[\sigma_{0ij}^2 f_0(b_{0ij}) + \sigma_{1ij}^2 \left[f_1(b_{1ij}) - 1 \right] \right]$$
(2.11)
subject to
$$\frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left[b_{0ij} + b_{1ij} \right] = \mathbf{R}.$$

The variances σ_{1ij}^2 depend on the bits b_{0ij} allocated to the first stage, which is not known in advance. In order to get around this problem, we will first assume that the variances of the coefficients of both stages, σ_{0ij}^2 and σ_{1ij}^2 , for i, j = 1, ..., N-1, are available. Based on this, we will derive the optimal bit rates for two stages. Once the rates are known, the new values of σ_{1ij}^2 will be computed. The process is iterated with these new values until the optimum point is reached. In practice, we found that two or three iterations are sufficient.

In the analysis, we will assume that b_{0ij} and b_{1ij} are continuous. Since we are looking for an analytical expression for the rates R_0 and R_1 , the error functions $f_0(.)$ and $f_1(.)$ must be known. The piecewise functions given in Table 2.1 can be used for marginal analysis bit allocation, but it is not easy to use them in the above minimization problem. Instead, we try to approximate these functions with another function in the form of $f_k(b_{kii}) = 2^{-B_k b_{kii}}$, for k = 0, 1. We choose the single parameter B_k such that the proposed function is the closest approximation to the corresponding piecewise model in the least mean square sense. Figs. 2.2 and 2.3 show the approximation for particular B_k . It is observed that the fit is very close. The problem can now be restated in the following form:

minimize
$$\frac{1}{N^{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left[\sigma_{0ij}^{2} \ 2^{-B_{0}b_{0ij}} + \sigma_{1ij}^{2} \left(2^{-B_{1}b_{1ij}} - 1 \right) \right]$$
(2.12)
subject to
$$\frac{1}{N^{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left(b_{0ij} + b_{1ij} \right) = R.$$

Using the Lagrange multiplier method [26] for this optimization problem, it it easy to show that

$$b_{0ij} = \frac{B_1}{B_0} b_{1ij} + \frac{1}{B_0} \left| \log_2 \left(\frac{\sigma_{0ij}^2}{\sigma_{1ij}^2} \right) + \log_2 \left(\frac{B_0}{B_1} \right) \right|.$$
(2.13)

By taking the summation over i and j of both sides of Eq. (2.13), and defining

$$S_{0} = \frac{1}{N^{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \log_{2} \sigma_{0ij}^{2}$$
(2.14)

and

$$S_{1} = \frac{1}{N^{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \log_{2} \sigma_{1ij}^{2}$$
(2.15)

we obtain

$$R_{0} = \frac{B_{1}}{B_{0}} R_{1} + \frac{1}{B_{0}} \left[S_{0} - S_{1} + \log_{2} \left(\frac{B_{0}}{B_{1}} \right) \right]$$
(2.16)

where

$$R_0 = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b_{0ij}$$
(2.17)

and

$$R_1 = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b_{1ij} .$$
 (2.18)

Using the fact that $R_0 + R_1 = R$, the average rate for the first stage becomes

$$R_{0} = \frac{B_{1}}{B_{0} + B_{1}} R + \frac{1}{B_{0} + B_{1}} \left[S_{0} - S_{1} + \log_{2} \left(\frac{B_{0}}{B_{1}} \right) \right]$$
(2.19)

Then, R_1 is found as $R - R_0$.

Extending the above procedure to the case n = 3 is easy. Suppose we can approximate the error function model of the third stage by $f_2(b_{2ij}) = 2^{-B_2b_{2ij}}$, for some B_2 (see Sec. 2.5.) Then, similar to Eq. (2.16), the following equation is derived:

where

$$R_2 = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b_{2ij}$$
(2.21)

and

$$S_{2} = \frac{1}{N^{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \log_{2} \sigma_{2ij}^{2}$$
(2.22)

Solving Eqs. (2.16) and (2.20) with the constraint $R_0 + R_1 + R_2 = R$ results in the following relations for R_0 and R_1 :

$$R_{0} = \frac{B_{1}B_{2}}{B_{0}B_{1} + B_{0}B_{2} + B_{1}B_{2}} R$$

$$+ \frac{B_{1} + B_{2}}{B_{0}B_{1} + B_{0}B_{2} + B_{1}B_{2}} \left(S_{0} + \log_{2}\frac{B_{0}}{B_{1}}\right)$$

$$+ \frac{B_{1}}{B_{0}B_{1} + B_{0}B_{2} + B_{1}B_{2}} \left(\log_{2}\frac{B_{1}}{B_{2}} - S_{2}\right)$$

$$- \frac{B_{2}}{B_{0}B_{1} + B_{0}B_{2} + B_{1}B_{2}} S_{1} \qquad (2.23)$$

and

$$R_{1} = \frac{B_{0}B_{2}}{B_{0}B_{1} + B_{0}B_{2} + B_{1}B_{2}} R - \frac{B_{2}}{B_{0}B_{1} + B_{0}B_{2} + B_{1}B_{2}} S_{0}$$

$$+ \frac{B_{0} + B_{2}}{B_{0}B_{1} + B_{0}B_{2} + B_{1}B_{2}} S_{1} - \frac{B_{0}}{B_{0}B_{1} + B_{0}B_{2} + B_{1}B_{2}} S_{2}$$

$$- \frac{B_{2}}{B_{1}B_{0}B_{1} + B_{0}B_{2} + B_{1}B_{2}} \log_{2}\frac{B_{0}}{B_{2}}$$

$$+ \frac{B_{1}}{B_{0}B_{1} + B_{0}B_{2} + B_{1}B_{2}} \log_{2}\frac{B_{1}}{B_{2}} \qquad (2.24)$$

Again, R_2 is found as $R - R_0 - R_1$. The above procedure can be generalized to any number of stages. Once

the average bit rates for each stage are known, the bit allocation for each stage can be done independently by using marginal analysis or any other techniques for any desired number of classes.

2.4 Experimental Results

The multistage transform coding technique discussed above was applied to a set of different images of sizes 128×128 , 256×256 and 512×512 , shown in Fig. 2.4. All images were quantized with 8 bits (256 levels.) The number of stages used were either 2 or 3. The two-dimensional DCT was used as the unitary transform. Coding was carried out with a block size of 16×16 . We compared multistage transform coding with one stage coding. The adaptive coding technique of Chen and Smith [13] with 4 classes was used for each stage. The total rates used were 1.0, 0.5 and 0.25 bits per pixel (bpp).

For the first stage, the optimum mean square error quantizer was used with the Laplacian distribution for the AC coefficients and the Gaussian distribution for the DC coefficients. The optimum mean square quantizer with Gaussian distribution for all the coefficients was used for the second stage. This choice was based on the statistical tests explained in Sec. 2.3.

For two stages with one class (without using Chen and Smith adaptive method), the total number of bits were allocated according to the marginal analysis method discussed in Sec. 2.3 to minimize the total final error function given by Eq. (2.8). For this case, two schemes are possible. Either the variances of the second stage σ_{1ij}^2 can be estimated by the known variances of the first stage by $\sigma_{1ij}^2 = \sigma_{0ij}^2 f_0(b_{0ij})$, or we can start from initial rates for the first and second stages and then iterate once the variances of the second stage are known. Our experiments showed that the second scheme is not as efficient as the first scheme. In addition, the first scheme is much better in terms of computational cost. Therefore, we chose the first scheme.

For the two stage multiclass adaptive method, we used Eq. (2.19) to allocate the total bits between two stages. In this case, we started with the initial rates $R_0 = R$ and $R_1 = 0$. This choice was based on our observation that, for optimum rate division, R_0 is usually greater than R_1 . In most cases one or two iterations were sufficient to get the optimum rates R_0 and R_1 .

In all simulations, 8 bits were allocated to the DC coefficients of the first stage. Thus, in computing S_0 with Eq. (2.14), the variance of the DC

coefficient was not included in the sum. For the same reason, $8/N^2$, which is the average bit rate of the DC coefficient at the first stage was subtracted from R in Eq. (2.19). For the second stage, the DC coefficients have an average close to zero, which is easily justified by Eq. (2.1).

We calculated the MSE and the Normalized MSE (NMSE) [27] by the following equations:

MSE =
$$\frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left(x(i,j) - \hat{x}(i,j) \right)^2$$
, (2.25)

$$NMSE = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left(x(i,j) - \hat{x}(i,j) \right)^{2}}{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left(x(i,j) \right)^{2}}$$
(2.26)

where x(i,j) and $\hat{x}(i,j)$ are pixel values of the original image and the reconstructed image, respectively. The improvement was also calculated in term of dB defined by

10 log₁₀ One stage Mean Square Error Multistage Mean Square Error

a start and a start of the start

Table 2.2 shows the numerical results. In this table we included the optimum rates for each stage. In the case of two stages, the multistage transform coding resulted in as much as 14.65% improvement for one class and 11.54% improvement for 4 classes. Fig. 2.5 shows some of the reconstructed images using multistage image transform coding with different rates for both one and multiclasses. The corresponding one stage coding results are also shown for comparison. It is observed that the results of multistage coding are more preferable than the results of one stage coding.

To compare the reconstructed images with the original images, the difference images were generated and are shown in Fig. 2.6. For better presentation, we generated the difference images in two different ways, which are referred to as method I and method II. In method I, the absolute values of the differences are normalized by the maximum difference value of each difference image, and the results are integer values from 0 to 255, with the brighter grey value representing the larger difference. In method II, the larger of the maximum difference value of one stage coding and the corresponding

two stage coding is used for normalization. It is clear from Fig. 4.6 that the difference images with multistage coding are more noise-like than the corresponding difference images with one stage coding. In addition, the differences on and around the edges are less dominant with multistage coding. The smooth regions are also smoother, as clearly observed in the case of the "baboon" image, especially around the nose.

이 것 이 가지는 것이 사람들은 것이 많이 많이 많다.

In the two stage experiments, we tested all possible combinations of R_0 and R_1 for rates equal to 0.5 and 0.25 bits per pixel, and some of the results are shown in Figs. 2.7. It is clear that the optimum points are generally close to what we found by either minimizing Eq. (2.8) directly for the one-class case or dividing the total rate by Eq. (2.19) for the multiclass case.

We also tested three stages, at the total rate of 0.5 bpp with using 4 classes, the original image "girl256" shown in Fig. 2.4. The results showed 13.88% improvement over one stage. This is 5.33% more than the improvement with two stages, and the same type of improvement is expected for other cases.

2.5 Discussion

经保持委托 化化学学 网络拉尔 化原料学 化合金 网络小麦

As mentioned in Sec. 2.4, a large number, but not all, of coefficients in the second stage have a pdf close to Gaussian. Since one kind of pdf is usually assumed during quantization, we chose the Gaussian pdf in Secs. 2.3 and 2.4. Even though, more than one choice of pdf is possible, it increases the overhead information that should be known during decoding. Thus, for one kind of pdf assumption, we have the error of mismatch between the assumed pdf and the real pdf for some coefficients. This error was experimentally studied by Mauersberger [28]. The reported results show that the error resulting from using Gaussian quantizer for a random variable with Laplacian pdf is more than the error resulting from using Laplacian quantizer for a random variable with Gaussian pdf (assuming the same variance and number of levels.) In practice, the total error depends on the number of mismatch cases. For the third stage in multistage image transform coding, our statistical tests showed that the coefficients have a mixture of uniform, Gaussian and Laplacian pdf. Again, since more than half of them have Gaussian pdf, we used the Gaussian quantizer. We are investigating further how the mismatch error can be minimized for multistage image transform coding. One possible solution is to

use a bit allocation method that can be adapted to any kind of probability density function for coefficients and where the error function can be calculated iteratively. Specifically, we are studying the implementation of the method given by Shoham and Gersho [29] for this problem.

It must be mentioned that the multistage procedure discussed in this chapter will slightly increase the overhead information. The main part of overhead information is the bit map. For the optimal multistage image transform coding, we assume that the total number of bits (or the corresponding total average rate) is fixed. When the total rate is divided between stages, more number of pixels per stage assume zero bits. Thus, the overhead information is not doubled. Usually, about 0.03 bpp is needed in one stage coding for overhead information, including error protection bits, for the 0.5 bpp case with an image of size 256×256 [13]. For finding the net improvement, we assumed 0.015 bpp for extra overhead in two stage coding, which is an overestimate. When we increased the bit rate by this amount in one stage coding for the original image "girl256", we found that the net improvement was about 1.5% less than what is given in Table 2.2. In another test, we considered 0.01 bpp additional overhead information for the "baboon" image with two stage and 4 classes coding, which is definitely more than the necessary. In this case, the net improvement was just 0.63% less than what is given in Table 2.2.

With sequential video images, it may also be possible to use the same variances in corresponding blocks of successive images to reduce the computation in the iterative procedure of finding the bit rates R_0 , R_1 , \cdots in the multiclass problem.

出版 医周期的 医克尔特加氏征闭注闭的

2.6 Conclusions

法认为法法 化二磷酸合物 医试验 经转

승규님 성실 사람이 물질 것 같아요. 여기가 같은

化合金型 经济税利益

Both theoretical and experimental results indicate that optimal adaptive multistage image transform coding is quite effective in reducing mean square reconstruction error over what is possible with one stage transform coding. Optimality is achieved by the minimization of the total final error using marginal analysis. This minimization determines how to allocate bits to the coefficients in each stage. After the first stage, the pdf of the coefficients appear to be either Gaussian or uniform. The reconstruction of the quantized image is obtained by adding together the quantized transform coefficients

from all the stages and computing a single inverse transform of the results. Further improvements in the techniques described are expected to reduce reconstruction error more.

Maria Maria Maria

요즘 사람이는 것이 같은 것이 같은 것이 같은 것이 없는 것이 없

In this chapter, we considered MSE as the performance criterion. However, the difference images shown in Fig. 2.6 indicate that the reconstruction errors are more noise-like in multistage coding than in one stage coding, with especially reduced errors at the edges. The smooth regions are also smoother, as clearly observed in the case of the "baboon" image, especially around the nose. These are believed to be the reasons why the reconstructed images with the multistage method are subjectively much more preferable than the reconstructed images with the one stage method at the same bit rate.

Although the proposed method was tested for DCT and monochrome images, it can be easily applied to other transforms and color images.

The second s

a set the set of the set of the



Figure 2.1. Block diagram of multistage image transform coding.

1	able	2.1.	The e form i	error moe f _k (b _{kii}) =	del for (= A2 ^{-Bb} l	Gaussian ^{(ij} given	and L in Ref.	aplaciar [15].	ı distribu	ition of	the

Distribution	0≤b _{kij} ≤2.32	2.32 <b<sub>kij≤5.17</b<sub>	5.17 <b<sub>kij≤9</b<sub>		
	A B	A B	AB		
Gaussian	1 1.5047	1.5253 1.8274	2.2573 1.9626		
Laplacian	1 1.1711	2.0851 1.7645	3.6308 1.9572		



Figure 2.2. Approximation of the error model for the Gaussian pdf given in Table 2.1 with $B_k = 1.52$ in the form of $f_k(b_{kij}) = 2^{-B_k b_{kij}}$.



Figure 2.3. Approximation of the error model for the Laplacian pdf given in Table 2.1 with $B_k = 1.23$ in the form of $f_k(b_{kij}) = 2^{-B_k b_{kij}}$.



(a)



(b)



(c)

Figure 2.4. Original images: (a) "girl128", 128 × 128. (b) "girl256", 256 × 256. (c) "lenna", 256 × 256.



(d)

Figure 2.4 (continued) (d) "baboon", 512×512 .

	Rates	1.	00	0.	50	0.	25
Images	# of classes	1	4	1	4	1	4
	MSE for 1 stage	31.83	25.88	67.76	53.35	126.3	98.82
	NMSE(%) for 1 stage	0.450	0.366	0.959	0.755	1.787	1.398
	MSE for 2 stages	30.10	24.23	62.89	47.78	116.9	88.31
"girl128"	NMSE(%) for 2 stages	0.426	0.343	0.890	0.676	1.654	1.250
	Improvement in MSE (%)	5.44	5.99	7.19	10.44	7.44	10.64
	Improvement in dB	0.243	0.286	0.324	0.479	0.336	0.488
	R ₀	0.707	0.827	0.344	0.338	0.176	0.052
	R ₁	0.293	0.173	0.156	0.162	0.074	0.198
	MSE for 1 stage	30.78	18.16	62.78	36.04	116.9	70.29
	NMSE(%) for 1 stage	0.426	0.251	0.868	0.498	1.616	0.972
	MSE for 2 stages	27.66	18.12	55.35	32.96	99.77	63.60
"girl256"	NMSE(%) for 2 stages	0.382	0.251	0.765	0.456	1.379	0.879
	Improvement in MSE (%)	10.14	0.22	11.83	8.55	14.65	9.52
	Improvement in dB	0.464	0.010	0.547	0.388	0.688	0.434
	RO	0.695	0.86	0.352	0.34	0.176	0.148
	R ₁	0.305	0.14	0.148	0.16	0.074	0.102

Table 2.2. Simulation results for multistage transform coding.

Table 2.2. (continued)

	MSE for 1 stage	39.60	28.02	92.29	62.57	174.2	118.2
	NMSE(%) for 1	0.225	0.159	0.524	0.355	0.989	0.671
	stage						
	MSE for 2 stages	35.48	27.14	80.50	55.35	150.7	105.5
"lenna"	NMSE(%) for 2 stages	0.201	0.154	0.457	0.314	0.855	1.250
	Improvement in MSE (%)	10.40	3.14	12.77	11.54	13.49	10.74
	Improvement in dB	0.477	0.139	0.594	0.532	0.629	0.494
성장 관계 관계 관계 관계	Ro	0.691	0.770	0.344	0.332	0.172	0.156
	R 1	0.309	0.230	0.156	0.168	0.078	0.094
	MSE for 1 stage	281.2	205.0	449.0	359,4	593.2	513.2
	NMSE(%) for 1 stage	10.92	7.96	17.43	13.95	23.03	19.92
	MSE for 2 stages	247.3	185.9	410.4	326.8	562.8	483.8
"babon"	NMSE(%) for 2 stages	9.60	7.22	15.93	12.69	21.85	18.78
	Improvement in MSE (%)	12.06	9.32	8.60	9.07	5.12	5.73
	Improvement in dB	0.558	0.425	0.390	0.413	0.228	0.256
	R ₀	0.613	0.729	0.293	0.316	0.145	0.115
	R	0.387	0.271	0.207	0.184	0.105	0.135











Figure 2.5. Reconstructed image of: (a) "girl256" for one stage coding with rate 0.5 bpp and 1 class. (b) "girl256" for two stage coding with rate 0.5 bpp and 1 class. (c) "girl256" for one stage coding with rate 0.5 bpp and 4 classes. (d) "girl256" for two stage coding with rate 0.5 bpp and 4 classes.



Figure 2.5 (continued) (e) "girl256" for one stage coding with rate 0.25 bpp and 4 classes. (f) "girl256" for two stage coding with rate 0.25 bpp and 4 classes. (g) "lenna" for one stage coding with rate 0.5 bpp and 4 classes. (h) "lenna" for two stage coding with rate 0.5 bpp and 4 classes.



(i)

Figure 2.5 (continued) (i) "baboon" for one stage coding with rate 1.0 bpp and 1 class.



的复数使用的复数

و الم



Figure 2.5 (continued) (j) "baboon" for two stage coding with rate 1.0 bpp and 1 classes.



Figure 2.6. The difference images for: (a) Fig. 2.5(a) by method I. (b) Fig.
2.5(b) by method I. (c) Fig. 2.5(a) by method II. (d) Fig. 2.5(b) by method II.



Figure 2.6 (continued) (e) Fig. 2.5(c) by method I. (f) Fig. 2.5(d) by method I. (g) Fig. 2.5(c) by method II. (h) Fig. 2.5(d) by method II.

Ìs i



Figure 2.6

(continued) (i) Fig. 2.5(e) by method I. (j) Fig. 2.5(f) by method I. (k) Fig. 2.5(e) by method II. (l) Fig. 2.5(f) by method II.



Figure 2.6 (continued) (m) Fig. 2.5(g) by method I. (n) Fig. 2.5(h) by method I. (o) Fig. 2.5(g) by method II. (p) Fig. 2.5(h) by method II.



(q)

Figure 2.6 (continued) (q) Fig. 2.5(i) by method I.



(r)

Figure 2.6 (continued) (r) Fig. 2.5(j) by method I.



(s)

Figure 2.6 (continued) (s) Fig. 2.5(i) by method II.



Figure 2.6

(continued) (t) Fig. 2.5(j) by method II.



Experimental results with two stage coding for all possible values Figure 2.7. for R_0 and R_1 ($R = R_0 + R_1$): (a) with "girl256", rate 0.5 bpp and 1 class. (b) with "girl256", rate 0.25 bpp and 1 class. (c) with "girl128", rate 1.0 bpp and 4 classes.

CHAPTER 3 TRANSFORM IMAGE ENHANCEMENT

3.1 Introduction

Image enhancement involves processing of an image to make the image more satisfactory to the viewer. Image enhancement may be followed by image segmentation, which is to partition the image space into meaningful regions. The algorithm used for image enhancement also affects the results of image segmentation.

A survey of digital image enhancement methods can be found in Ref. [4]. One class of image enhancement methods includes gray scale modification, deblurring and smoothing. Transform techniques form another class. Image transforms provide a spectral decomposition of an image into spectral coefficients which can be modified, linearly or nonlinearly, for the purpose of image enhancement.

Images are usually digitized with 8 or 16 bits, and large memory is needed to store them. Hence image coding is necessary for storage and transmission of images. Image transform coding techniques are among the most powerful coding algorithms, [5,15,27,30]. Hence, fast transforms for image coding have been more thoroughly studied than for other purposes, and the best transforms in the sense of performance and computational complexity have been determined. The discrete cosine transform (DCT) has often been preferred for image coding because of its closeness to the optimal Karhunen -Loeve transform for Markov-I type signals, which is a reasonable model for images [8]. However, two other more recently studied transforms have attractive properties for image coding. The scrambled real discrete Fourier transform (SRDFT) has much less multiplicative complexity of implementation with almost the same coding performance as the DCT [9]. As a matter of fact, visually the SRDFT results may be preferable to the DCT results. The discrete cosine-III transform (DC3T) has computational complexity midway between the SRDFT and the DCT, and has the best performance in terms of the mean-square reconstruction error as well as visual criteria [10].

An additional advantage of transform image enhancement techniques is low complexity of computations if they are implemented together with transform image coding. Previously, transform image enhancement has usually been based on the discrete Fourier transform (DFT). There are two major drawbacks with the DFT. First, it has high complexity of implementation involving complex multiplications and additions with intermediate results being complex numbers. Secondly, it creates severe block effects if implemented blockwise as in image coding. In addition, the quality of enhancement is not as good as what is possible with some other transforms as discussed in this chapter.

A major motivation for this chapter is the determination of the best transform for image enhancement with low computational complexity, coupled with the requirement to perform image enhancement blockwise without creating objectionable block effects, with all blocks possibly computed in parallel. Three transform image enhancement techniques are utilized for a comparative analysis of transform image enhancement. These are alpharooting, modified unsharp masking, and filtering based on the properties of the human visual system response (HVS). It will be observed that the best transforms for image coding are also the best in image enhancement.

The chapter consists of 9 sections. In Sec. 3.2, the fast transforms to be compared for blockwise image enhancement are discussed. In Sec. 3.3, we describe the generalized filtering procedure in the transform domain to be used in the enhancement techniques. Secs. 3.4, 3.5 and 3.6 involve a detailed description of the three enhancement techniques and comparative experimental results. In Sec. 3.7, an overlap-save method which completely removes edge-effects is discussed. The similarity between the modified unsharp-masking and HVS-filtering techniques are described in Sec. 3.8. Sec. 3.9 is conclusions.

3.2 Fast Transforms

In this section, we will describe the 2-D fast transforms which are to be used on a comparative basis in the following sections. The following notation will be used: $X(n_1, n_2)$: transformed image of size $N_1 \times N_2$.

The 2-D DFT of an image $x(n_1,n_2)$ of size $N_1 \times N_2$, denoted by $X(n_1,n_2)$, is defined as

$$\mathbf{X}(\mathbf{n}_{1},\mathbf{n}_{2}) = \sum_{\mathbf{k}_{1}=0}^{N_{1}-1N_{2}-1} \sum_{\mathbf{k}_{2}=0}^{-j2\pi \left(\frac{\mathbf{n}_{1}\mathbf{k}_{1}}{N_{1}} + \frac{\mathbf{n}_{2}\mathbf{k}_{2}}{N_{2}}\right)}$$
(3.1)

The inverse DFT is

$$\mathbf{x}(\mathbf{n}_{1},\mathbf{n}_{2}) = \frac{1}{N_{1}N_{2}} \sum_{\mathbf{k}_{1}=0}^{N_{1}-1N_{2}-1} \sum_{\mathbf{k}_{2}=0}^{j2\pi \left(\frac{\mathbf{n}_{1}\mathbf{k}_{1}}{N_{1}} + \frac{\mathbf{n}_{2}\mathbf{k}_{2}}{N_{2}}\right)}$$
(3.2)

The DFT involves complex multiplications and additions. Many fast algorithms for the DFT have been developed, and are known as the fast Fourier transform (FFT) [31].

The 2-D discrete cosine transform (DCT) of the signal $x(n_1,n_2)$ is defined as [8]

$$X(n_{1},n_{2}) = \frac{4c(n_{1})c(n_{2})}{N_{1}N_{2}}\sum_{k_{1}=0}^{N_{1}-1}\sum_{k_{2}=0}^{N_{2}-1}x(k_{1},k_{2})\cos\left[\frac{(2k_{1}+1)n_{1}\pi}{2N_{1}}\right]$$

$$\cos\left[\frac{(2k_{2}+1)n_{2}\pi}{2N_{2}}\right]$$
(3.3)

The 2-D inverse DCT is

$$\begin{aligned} \kappa(n_1, n_2) &= \sum_{k_1=0}^{N_1-1N_2-1} c(k_1) \ c(k_2) \ X(k_1, k_2) \ \cos\left[\frac{(2n_1+1)k_1\pi}{2N_1}\right] \\ &\cos\left[\frac{(2n_2+1)k_2\pi}{2N_2}\right] \end{aligned}$$
(3.4)

where, with N equal to N_1 or N_2 ,

$$c(k) = \frac{1}{\sqrt{2}} \quad \text{for } k = 0$$

= 1 for k = 1,...,N-1. (3.5)

The real discrete Fourier transform (RDFT), denoted by $X(k_1,k_2)$, of the image $x(n_1,n_2)$ is defined as [32]

$$\begin{aligned} \mathbf{X}(\mathbf{n}_{1},\mathbf{n}_{2}) &= \sum_{\mathbf{k}_{1}=0}^{\mathbf{N}_{1}-1\mathbf{N}_{2}-1} \mathbf{x}(\mathbf{k}_{1},\mathbf{k}_{2}) \cos\left[\frac{2\pi\mathbf{n}_{1}\mathbf{k}_{1}}{\mathbf{N}_{1}} + \Theta(\mathbf{n}_{1})\right] \\ &\cos\left[\frac{2\pi\mathbf{n}_{2}\mathbf{k}_{2}}{\mathbf{N}_{2}} + \Theta(\mathbf{n}_{2})\right] \end{aligned}$$
(3.6)

The inverse RDFT is

$$\mathbf{x}(\mathbf{n}_{1},\mathbf{n}_{2}) = \frac{4}{N_{1}N_{2}} \sum_{\mathbf{k}_{1}=0}^{N_{1}-1N_{2}-1} \mathbf{X}(\mathbf{k}_{1},\mathbf{k}_{2})\mathbf{c}(\mathbf{k}_{1})\mathbf{c}(\mathbf{k}_{2})\cos\left[\frac{2\pi\mathbf{k}_{1}\mathbf{n}_{1}}{N_{1}} + \Theta(\mathbf{k}_{1})\right]$$

$$\cos\left[\frac{2\pi\mathbf{k}_{2}\mathbf{n}_{2}}{N_{2}} + \Theta(\mathbf{k}_{2})\right]$$
(3.7)

where, with N equal to N_1 or N_2 ,

승객님 않는 것

$$\Theta(\mathbf{n}) = 0 \quad \mathbf{0} \le \mathbf{n} \le \frac{\mathbf{N}}{2}$$
$$= \frac{\pi}{2} \quad \mathbf{n} > \frac{\mathbf{N}}{2}, \qquad (3.8)$$

and

$$c(n) = 1 \quad n \neq 0, \ \frac{N}{2}$$

$$=\frac{1}{2} \quad n=0, \ \frac{N}{2}.$$
(3.9)

The RDFT is the discretized version of the real Fourier transform. The RDFT involves only real multiplications and additions in contrast to the DFT. The explicit relation between the DFT and the RDFT was discussed in Ref. [33]. The fast algorithms for the RDFT are known as the real fast Fourier transforms (RFFT).

The scrambled real discrete Fourier transform (SRDFT), is similar to the RDFT. The SRDFT, denoted by $X(k_1,k_2)$, of the image $x(n_1,n_2)$ is defined as [9]

$$X(n_{1},n_{2}) = \sum_{k_{1}=0}^{N_{1}-1N_{2}-1} \sum_{k_{2}=0} x(n_{1},n_{2}) \cos\left[\frac{2\pi n_{1}k_{1}'}{N_{1}} + \Theta(n_{1})\right]$$
$$\cos\left[\frac{2\pi n_{2}k_{2}'}{N_{2}} + \Theta(n_{2})\right].$$
(3.10)

The inverse SRDFT is

$$\mathbf{x}(\mathbf{n}_{1},\mathbf{n}_{2}) = \frac{4}{N_{1}N_{2}} \sum_{\mathbf{k}_{1}=0}^{N_{1}-1N_{2}-1} \mathbf{X}(\mathbf{k}_{1},\mathbf{k}_{2})\mathbf{c}(\mathbf{k}_{1})\mathbf{c}(\mathbf{k}_{2})\cos\left[\frac{2\pi\mathbf{k}_{1}\mathbf{n}_{1}'}{N_{1}} + \Theta(\mathbf{k}_{1})\right]$$

$$\cos\left[\frac{2\pi\mathbf{k}_{2}\mathbf{n}_{2}'}{N_{2}} + \Theta(\mathbf{k}_{2})\right]$$
(3.11)

where $\Theta(k)$ and c(k) are defined as in Eqs. (3.8) and (3.9), respectively; k' equal to k_1 or k_2 in Eq. (3.10) and n'_1 or n'_2 in Eq. (3.11) is given by

$$\mathbf{k}' = \frac{\mathbf{k}}{2}, \qquad \mathbf{k} \text{ even}$$
$$= -\frac{(\mathbf{k}+1)}{2}, \qquad \mathbf{k} \text{ odd.} \qquad (3.12)$$

Similar to the RDFT, the SRDFT involves real multiplications and

additions. It consists of permutations of input data followed by the RDFT.

The discrete cosine-III transform (DC3T) [10] consists of preprocessing followed by the discrete symmetric cosine transform (DSCT) [34]. Let x(n), n = 0, 1, ..., N-1 be the input sequence. We define z(n), n = 0, 1, ..., N, as

$$z(0) = \frac{x(0)}{\sqrt{2}}$$

$$z(N) = \frac{x(N-1)}{\sqrt{2}}$$

$$z(n) = \frac{x(n) + x(n-1)}{2}, \quad n=1,2,...,N-1.$$
(3.13)

The DC3T of x(n), denoted by X(k), is defined as

$$X(n) = v(n) \sqrt{\frac{2}{N}} \sum_{k=0}^{N} v(k) z(k) \cos \frac{\pi n k}{N} \quad k=0,1,...,N-1. \quad (3.14)$$

Eq. (3.14) defines the DSCT. The inverse DC3T is

$$u(n) = v(n) \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} v(k) X(k) \cos \frac{\pi n k}{N}$$
 n=0,1,...,N-1 (3.15)

where

$$\mathbf{v}(\mathbf{n}) = \frac{1}{\sqrt{2}} \quad \mathbf{n} = \mathbf{0}$$

= 1 otherwise. (3.16)

and z(n) is converted to x(n) by the inverse of Eq. (3.13).

The two-dimensional DC3T is obtained by applying the onedimensional DC3T to the rows and then the columns of the image. It is noted that the DC3T involves real multiplications and additions, and it can be computed by fast algorithms for the DSCT [35].

Fig. 3.1 shows the energy distribution of different transforms for typical images. In this figure, we applied the transformation $\ln \left(1 + |X(k_1,k_2)|^2\right)$ where |.| indicates the magnitude of coefficients, and then normalized the results to be between 0 and 255. Finally, we thresholded the resulting image by indicated levels.

The computational complexities in terms of number of additions and multiplications of the given transforms are given in Table. 3.1 as a function of block size N. The number of additions and multiplications for the RDFT are the same as those for the SRDFT. For the DFT, there are many fast Fourier transform (FFT) methods with the computational complexity being generally higher than that of RDFT [31].

Another transform which is needed in Sec. 3.3 is the discrete sine transform (DST). The 1-D DST is given by

$$X(n) = \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} x(k) \sin \frac{\pi n k}{N} \quad n=1,...,N-1.$$
(3.17)

The DST is its own inverse.

3.3 Generalized Filtering in the Transform Domain

All the image enhancement techniques to be discussed in the following sections involve "generalized" filtering concepts with fast transforms. By generalized filtering, we mean processing of the image as shown in Fig. 3.2 The image is transformed, then multiplied by a matrix and inverse transformed. What this means with different transforms in terms of convolution is discussed below.

With the DFT, multiplication of the transform of the impulse response by the transform of the signal is equivalent to circular convolution in the signal domain. In order to prevent aliasing, the impulse response of the filter and the signal can be appended by a number of zeros before taking the DFT. The number of zeros is equal to or greater than N-1, where N is the length of the signal or of the filter impulse response, whichever one is larger. Doing so converts circular convolution to linear convolution. The 2-D case is similar with zero-filling in both directions for achieving linear rather than circular convolution.

The RDFT of the signal x(.) is equivalent to separating x(.) into $x_1(.)$, the even part of the signal, and $x_0(.)$, the odd part of the signal, followed by the computation of the DSCT of $x_1(.)$ to give $X_1(.)$, and the DST of $x_0(.)$ to give $X_0(.)$. $x_1(n)$ and $x_0(n)$ are given by
$$x_1(0) = x(0)$$

 $x_1(\frac{N}{2}) = x(\frac{N}{2})$ if N even, (3.18)

and for $0 < n < \frac{N}{2}$,

$$x_1(n) = x(n) + x(N - n)$$

 $x_0(n) = x(n) - x(N - n)$ (3.19)

The multiplication of $X_1(.)$ and $X_0(.)$ by a filter transfer function corresponds to the circular convolution of $x_1(.)$ with an even impulse function $h_1(.)$, and the circular convolution of $x_0(.)$ with a odd impulse function $h_0(.)$. The two circular convolutions are combined in the end since the inverse RDFT is utilized. Let $H_1(.)$ and $H_0(.)$ be the DSCT and the DST of $h_1(.)$ and $h_0(.)$, respectively. The process described above can be written as

$$Y_{1}(0) = X_{1}(0)H_{1}(0)$$

$$Y_{1}(\frac{N}{2}) = X_{1}(\frac{N}{2})H_{1}(\frac{N}{2}) \text{ if N even}$$
(3.20)

and for 0 < n < N/2,

$$\begin{bmatrix} Y_1(n) \\ Y_0(n) \end{bmatrix} = \begin{bmatrix} H_1(n) & 0 \\ 0 & H_0(n) \end{bmatrix} \begin{bmatrix} X_1(n) \\ X_0(n) \end{bmatrix}$$
(3.21)

where $\begin{bmatrix} Y_1(n) & Y_0(n) \end{bmatrix}^T$ is the RDFT of the output signal. This can be compared to the circular convolution of x(.) with an impulse response function h(.) whose even and odd parts are h₁(.) and h₀(.), respectively. Then, Eq. (3.18) is modified to

$$\begin{bmatrix} Y_1(n) \\ Y_0(n) \end{bmatrix} = \begin{bmatrix} H_1(n) & -H_0(n) \\ H_0(n) & H_1(n) \end{bmatrix} \begin{bmatrix} X_1(n) \\ X_0(n) \end{bmatrix}$$
(3.22)

In the 2-D case, the problem is more complicated. There are effectively four signals $x_{11}(n_1,n_2)$, $x_{10}(n_1,n_2)$, $x_{01}(n_1,n_2)$ and $x_{00}(n_1,n_2)$. For n_1 and n_2

not equal to 0 or $N_1/2$ and $N_2/2$, respectively, they are given by

If n_1 or n_2 is equal to 0 or $N_1/2$ and $N_2/2$, respectively, then the 1-D equations are valid with one of the indices fixed.

The signals $x_{11}(n_1, n_2)$, $x_{10}(n_1, n_2)$, $x_{01}(n_1, n_2)$ and $x_{00}(n_1, n_2)$ are transformed by 2-D DSCT, DSCT-DST, DST-DSCT and 2-D DST transforms to yield $X_{11}(n_1, n_2)$, $X_{10}(n_1, n_2)$, $X_{01}(n_1, n_2)$ and $X_{00}(n_1, n_2)$, respectively. The same is done with the 2-D impulse response $h(n_1, n_2)$ to yield $H_{11}(n_1, n_2)$, $H_{10}(n_1, n_2)$, $H_{01}(n_1, n_2)$ and $H_{00}(n_1, n_2)$. Then, $X_{ij}(n_1, n_2)$ is multiplied by the $H_{ij}(n_1, n_2)$ to yield $Y(n_1, n_2)$, which is the 2-D RDFT of the output $y(n_1, n_2)$.

The end result is that $x_{ij}(n_1,n_2)$ is circularly convolved with $h_{ij}(n_1,n_2)$. Then the four circular convolutions are combined together due to the inverse RDFT processing.

A similar circular convolution property was discussed with the DCT [36]. For this purpose, the definition of c(k) in Eq. (3.5) was extended as c(k) equal to zero for $N \le k \le 2N-1$, and the following was shown: If

$$W_{c}(k) = X_{c}(k) H_{c}(k)$$
(3.24)

then

$$\mathbf{w}(n) = \hat{\mathbf{x}}(n) * \hat{\mathbf{h}}(n) * \mathbf{z}(n)$$
 (3.25)

where " * " denotes circular convolution; $\hat{\mathbf{x}}(n)$ and $\hat{\mathbf{h}}(n)$ are symmetric sequences defined as

$$\hat{\mathbf{x}}(n) = \begin{cases} \mathbf{x}(n) & \text{for } n = 0, 1, ..., N-1 \\ \mathbf{x}(2N-1-n) & \text{for } n = N, N+1, ..., 2N-1, \end{cases} (3.26)$$

$$\hat{\mathbf{h}}(n) = \begin{cases} h(n) & \text{for } n = 0, 1, ..., N-1 \\ h(2N-1-n) & \text{for } n = N, N+1, ..., 2N-1, \end{cases} (3.27)$$

where x(n) and h(n) are the inverse DCT of $X_c(k)$ and $H_c(k)$, respectively; z(n)

is given by

If

$$z(n) = 4 \left\{ \left(\frac{1}{2\sqrt{2}} - 1 \right) + \cos\left(\frac{\pi}{4N} (2n-1)(N-1) \right) \frac{\sin\left[\frac{\pi}{4} (2n-1) \right]}{\sin\left[\frac{\pi}{4N} (2n-1) \right]} \right\}, \quad (3.28)$$

Another type of filtering with the DCT was discussed in Ref. [37]. In this reference, c(k) in Eq. (3.5) was extended as c(k) equal to 1 for k = -N, -N+1, ..., -1. By this extension, the DCT of a real signal is even; in other words $X_c(k)$ equal to $X_c(-k)$ for k = 0, 1, ..., N-1, and $X_c(-N)$ equal to zero. By defining the DCT in the range of $-N+1 \le k \le N-1$, it was shown that

$$X_{c}(k) = 2c(k) \exp\left(\frac{-j\pi k}{2N}\right) \hat{X}_{f}(k), \text{ for } k = -N, -N+1, ..., N-1. (3.29)$$

where $X_f(k)$ is the DFT of $\hat{x}(n)$. Then, for real and even h(n), if

 $W_{c}(k) = X_{c}(k) H_{f}(k), \text{ for } k = -N, -N+1, ..., N-2, N-1,$ (3.30) where $H_{f}(k)$ is the DFT of h(n),

$$w(n) = \hat{x}(n) * h(n), \text{ for } n = 0, 1, ..., N-1.$$
 (3.31)

Since h(n) is even, $H_f(k)$ is also the same as $H_1(k)$, the DSCT of h(n).

From Eqs. (3.30), (3.31), it can be concluded that for a filter which has real and even frequency response, multiplication in the transform domain is equivalent to 2N-point circular convolution of the sequences $\hat{x}(n)$ and h(n).

Eqs. (3.30) and (3.31) are easily modified for the two-dimensional case as follows:

$$W_{c}(k_{1},k_{2}) = X_{c}(k_{1},k_{2}) H_{f}(k_{1},k_{2}), \text{ for } k_{1},k_{2} = -N,...,N-1, (3.32)$$

있는 Actual 2007

where $H_f(k_1, k_2) = H_f(k_1, -k_2) = H_f(-k_1, k_2) = H_f(-k_1, -k_2)$ and is real, then

$$w(n_1, n_2) = \hat{x}(n_1, n_2)^{**} h(n_1, n_2), n_1, n_2 = 0, 1, ..., N-1,$$
 (3.33)

where the two-dimensional symmetric sequence, $\hat{x}(n_1, n_2)$ is defined similar to Eq. (3.26), and "** " denotes two-dimensional circular convolution.

The relation between the DCT and the DC3T was given explicitly in Ref. [10]. Denoting the DC3T and the DCT of $x(n_1,n_2)$ by $X_{c3}(k_1,k_2)$ and $X_c(k_1,k_2)$, respectively, this relation is

$$X_{c3}(k_1,k_2) = \cos\left(\frac{\pi k_1}{2N}\right) \left(\frac{\pi k_2}{2N}\right) X_c(k_1,k_2) \text{ for } k_1,k_2 = 0,1,...,N-1. \quad (3.34)$$

요즘 사람들은 것을 하는 것을 하는 것을 하는 것을 사람들을 못 하는 것을 하는 것을 가지 않는 것 같이 있는 것

Therefore, by multiplying both sides of Eq. (3.32) by the same cosine factors as in Eq. (3.34), the same conclusion can be reached for the DC3T. In other words, if

$$W_{c3}(k_1,k_2) = X_{c3}(k_1,k_2) H_f(k_1,k_2)$$
(3.35)

then

$$w(n_1,n_2) = \hat{x}(n_1,n_2) ** h(n_1,n_2)$$
(3.36)

The SRDFT corresponds to input permutations according to Eq. (3.12) followed by the RDFT. Thus, the discussion of generalized filtering with the RDFT also applies to the SRDFT with respect to the permuted signal sequence.

The relation between the DCT and the SRDFT was given in Ref. [9]. Considering this relation, there is some similarity between the DCT and the SRDFT, but the energy distribution in the transform domain shown in Fig. 3.1 reveals a major difference. The SRDFT components are more dominant along the upper right and lower left edges in the transform plane. This property necessitates a more careful design of appropriate window in the transform domain.

3.4 Alpha-Rooting

This technique is also known as coefficient rooting or root filtering [38]. Fig. 3.3 shows the block-diagram of alpha-rooting. In this technique, the magnitude of each transform coefficient is raised to a power α , $0 < \alpha < 1$, and the sign or the phase of the coefficient is unchanged. The modified transform coefficient X'(n₁,n₂) may be written as

$$\begin{aligned} \mathbf{X}'(\mathbf{n}_{1},\mathbf{n}_{2}) &= \frac{\mathbf{X}(\mathbf{n}_{1},\mathbf{n}_{2})}{\left| \mathbf{X}(\mathbf{n}_{1},\mathbf{n}_{2}) \right|} \left| \mathbf{X}(\mathbf{n}_{1},\mathbf{n}_{2}) \right|^{\alpha} \\ &= \mathbf{X}(\mathbf{n}_{1},\mathbf{n}_{2}) \left| \mathbf{X}(\mathbf{n}_{1},\mathbf{n}_{2}) \right|^{\alpha-1} \end{aligned}$$
(3.37)

Thus, $|X(n_1,n_2)|^{\alpha-1}$ corresponds to the signal-dependent filter transfer function. When α equal to zero, only the phase or the sign of the coefficients is retained. With $\alpha < 1$, the amplitude of the large transform coefficients are reduced relative to the amplitude of the small transform coefficients. Since high frequencies are often associated with the small transform coefficients, the end result is enhanced edges and details of the image. In practice, $0.50 < \alpha < 0.99$ is used for image enhancement. The optimum value of α is image dependent and should be adjusted interactively by the user.

Alpha-rooting with the DFT and the RDFT introduces certain undesired artifacts related to sharp edges in the processed image. This problem was discussed using linear filtering theory and a modification of alpha-rooting when using the DFT was suggested in Ref. [39]. The block diagram of the modified method is given in Fig. 3.4 In this method, the equivalent transfer function $|X(n_1,n_2)|^{\alpha-1}$ is inverse transformed to the space domain where windowing is applied in order to smooth filter coefficients before transforming back to the frequency domain. The modified alpha-rooting is effective in reducing artifacts due to sharp edges in the processed image. We observed experimentally that this is also true with the other transforms discussed in Sec. 3.3. Our experiments showed that modified alpha-rooting is also effective to reduce block effects when the technique is implemented blockwise.

One major reason for artifacts in alpha-rooting with the DFT and the RDFT is the underlying property that the image is periodic with discontinuities at the edges between the periodic blocks. This is also the reason for circular convolution instead of linear convolution when the DFT is used. The end result is aliasing effects such as folding of edges. This problem is significantly reduced with transforms such as the DCT, SRDFT, and DC3T since the discontinuities at the edges of the periodic image are very small. The experimental results discussed below also confirm this property. Consequently, conventional alpha-rooting can be used with the DCT, the SRDFT, and the DC3T without significant artifacts and block effects, unlike the DFT and the

RDFT.

We experimented with both methods with different transforms and using blockwise processing with block sizes of 8, 16, and 32. The 512×512 image used, to be referred to as "catbrain", was a slice of a cat's brain shown in Fig. 3.5. The pixels were quantized with 8 bits from 0 to 255. With the modified method, we tried radial Gaussian and Butterworth windows given by the following equations:

Radial Gaussian

$$w_1(n_1, n_2) = \exp\left[-\frac{n_1^2 + n_2^2}{2\sigma^2}\right]$$
 (3.38)

Radial Butterworth

$$w_{2}(n_{1},n_{2}) = \frac{1}{1 + \left\lfloor \frac{n_{1}^{2} + n_{2}^{2}}{C_{0}} \right\rfloor^{n}},$$
(3.39)

where σ^2 in Eq. (3.38) is the variance of the window, and n and C₀ in Eq. (3.39) are the order of the window and a constant which controls the cutoff of the window, respectively. The values of σ^2 and C₀ depend on the block sizes. The Gaussian type window was experimentally found to perform better. The variance of the Gaussian window used was 90.0 for the DCT, the DC3T, and the SRDFT, 30.0 for the DFT and the RDFT with a block size of 32. These values also depend on the energy distribution of different transforms given in Fig. 3.1. The gray value of pixels in each block of the enhanced images was multiplied by a constant in order to keep the energy of each block of the processed image the same as that of the original image. Otherwise the results would be fuzzy.

A part of simulation results for different transforms and some specified values of α with both methods are given in Figs. 3.6-3.16. Figs. 3.6-3.9 are the results of conventional alpha-rooting with the DFT, the DCT, the SRDFT and the DC3T and α equal to 0.85. The block effects are obvious in all but are more severe in the DFT case. The results of the DCT, the SRDFT and the DC3T are almost the same. The RDFT results were similar to the DFT results for conventional alpha-rooting. Figs. 3.10-3.16 show the results for the modified alpha-rooting with different transforms and α equal to 0.85 and 0.7.

For all transforms, the block effects were reduced as a consequence of windowing. In this case, the SRDFT gave the best result.

We also tested these methods for different transforms with high quality images in order to make sure that the image quality is not degraded as a result of the enhancement process. The results were images with high details and sharp edges without any degradation.

3.5 Modified Unsharp Masking

The block-diagram of unsharp masking is shown in Fig. 3.17. The original image $x(n_1,n_2)$ is first divided into a low-pass image $x_L(n_1,n_2)$ and a high-pass image $x_H(n_1,n_2)$. The high-pass image is multiplied by a scalar C > 1 before being recombined with the low-pass image [40]. Since this process is similar to high-pass filtering, the result is enhanced edges and details of the image.

The optimum value of C is image-dependent and should be adjusted interactively by the user. This method is somewhat similar but simpler than what was discussed as modification of local contrast and local luminance in Ref. [40].

Experimentally, we observed that boosting of very high frequencies by unsharp masking leads to salt-and-pepper type of noise. In order to remove such noise, we modified unsharp masking by having two filters with transfer functions $H_L(n_1,n_2)$ and $H_N(n_1,n_2)$ as shown in Fig. 3.18. All processing is done in the transform domain. Then, the output image spectrum $Y(n_1,n_2)$ is given by

$$Y(n_1, n_2) = X(n_1, n_2) \left| H_L(n_1, n_2)(1-C) + C \right| H_N(n_1, n_2)$$
(3.40)

where $X(n_1, n_2)$ is the input image spectrum.

 $= \sum_{i=1}^{N} \frac{d_{i}^{2}}{d_{i}^{2}} e^{i \frac{1}{2}} \frac{d_{i}^{2}}{d_{i}^{2}} \frac{d_{i}^{2}}{d_{i}^{2}} e^{i \frac{1}{2}} \frac{d_{i}^{2}}{d_{i}^{2}} e^{i \frac{1}{2}} \frac{d_{i}^{2}}{d_{i}^{2}} \frac{d_{i}^{2}}{d_{i}^{2}} e^{i \frac{1}{2}} \frac{d_{i}^{2}}{d_{i}^{2}} \frac{d_{i}^{2}}}{d_{i}^{2}} \frac{d_{i}^{2}}{d_{i}^{2}} \frac{d_{i}$

We tried the radial Gaussian and Butterworth type filters whose transfer functions are given by

물건물 전문 방법 전자 이 것이 있는 것 같은 것 같은 것이 있는 것 같은 것이 없다.

$$F_1(f) = \exp\left[-\frac{f^2}{2\sigma^2}\right]$$
(3.41)

for Gaussian type, and

$$F_{2}(f) = \frac{1}{1 + \left[\frac{f^{2}}{f_{0}^{2}}\right]^{n}}$$
(3.42)

2. "这些人的"这个人的"这个事实是是不是

for the Butterworth type, where f is equal to $\sqrt{n_1^2 + n_2^2}$ is the radial frequency. In Eq. (3.41), σ^2 is the variance of the filter, and in Eq. (3.42), n and f_0 are the order of the filter and the cutoff frequency of the filter, respectively.

Experimentally, Gaussian type filters gave better results. So we will describe the results with the Gaussian window. In generating a low-pass image with a low-pass filter $H_L(n_1,n_2)$ of the type given in Eq. (3.41), the variance used depends on the block size and the applied transform. The low-pass image was subtracted from the original image in the transform domain to obtain the high-pass image to be multiplied by C. We used the same type of filter for $H_N(n_1,n_2)$ but with larger variance to separate noise at high frequencies. Subsequently, the energy of each block was adjusted as explained in Sec. 3.5. We tried three different block sizes, namely, 8, 16, and 32.

Simulation results for the block size of 32 with different transforms and with C equal to 7.0 and 4.0 are shown in Figs. 3.19-3.24. For the DFT and the RDFT, we used the variance of 20.0 to generate the low-pass image and the variance of 30.0 to remove high frequency noise. The variance of 50.0 and 90.0 were used for both the DCT and the DC3T. These values depend on the block size and are designed to eliminate high frequency noise sufficiently. The results show little block effects for the DCT and the DC3T, but the block effects in the case of the DFT or the RDFT are objectionable. The DCT and the DC3T gave very similar results, so have the DFT and the RDFT. By comparing Figs. 3.21 and 3.22 to Figs. 3.23 and 3.24, respectively, it can be concluded that the block effects will be increased slightly by more amplifying the high frequency.

3.6 Filtering Based on the Human Visual System Response

Since human beings are most likely the ultimate judges of processed images, it is appropriate to process the images in accordance with the properties of the human visual system response (HVS).

Studies indicate that the HVS contains different channels tuned to different spatial frequencies [14,41,42,43,44,45]. Fig. 3.25 shows the experimental spatial frequency response of the HVS [45] for the DFT. It is observed that human vision is most sensitive to midfrequencies, sensitivity tapering off at higher frequencies. Thus, the HVS is similar to a bandpass filter.

Let $x'(n_1, n_2)$ be the input image. Results of a number of studies indicate that before linear filtering, $x'(n_1, n_2)$ is passed through a zero-memory nonlinear transformation in the form [44]

$$x(n_1, n_2) = g(x'(n_1, n_2))$$
 (3.43)

in which g(.) is a monotonic increasing concave \bigcap function.

The HVS model in the form of pointwise nonlinearities followed by a generalized linear filter implemented by a fast transform is shown in Fig. 3.26.

In previous studies, $H(n_1, n_2)$ was assumed to be isotropic [44], i.e.,

$$H(n_1, n_2) = H(n)$$
 (3.44)

where

$$n = \sqrt{n_1^2 + n_2^2} \tag{3.45}$$

In the continuous case with the complex Fourier transform, $H(f_1, f_2)$ was modeled as [41,44]

$$H(f_1, f_2) = H(f)$$
 (3.46)

where

$$f = \sqrt{f_1^2 + f_2^2},$$
 (3.47)

and

$$H_{ft}(f) = a \left[b + \left(\frac{f}{f_o}\right)^{k_1} \right] e^{-\left(\frac{f}{f_o}\right)^{k_2}}$$
(3.48)

where f_1 , f_2 are the frequencies along the x- and the y-directions in cycles/degree, and a, b, f_0 are constant parameters. A number of experimental

방법이 가슴을 물건물을

studies were carried out for obtaining the "best" values for the parameters in the above model [41,44]. Another study by [46] gives a model that is close to the models discussed above. This model can be written as

$$H_{ft}(f) = (0.2 + 0.45f)e^{-0.18f}$$
 (3.49)

 $H_{ft}(f)$ has a peak value of 1.0 at f equal to 5.1 cycles/degree and a zero-frequency intercept of 0.2. It is shown in Fig. 3.25. $H_{ft}(f)$ was further perfected in the experimental work discussed below.

A number of studies concentrated on the choice of g(x) as x^c or log(d + x), where d is a small number such as 1.0 to avoid very large negative numbers [43]. The choice of $x^{0.33}$ was found to be the best during rating experiments [44]. The H_{ft}(f) model was studied in detail [14] and applied in image coding and image distortion measure [14,44].

Exploiting the relation between the DFT and the RDFT [33], we conclude that the HVS model discussed for the DFT is also appropriate for the RDFT. However this model is not necessarily optimal for other transforms. Refs. [46,47] discussed the extension of this model to the cosine transform. It was concluded in Ref. [47] that the HVS transfer function corresponding to the DFT can be applied to the DCT without much difference. An explicit model for this case was given in Ref. [46], and Ref. [48] gave the following equation that fits the model suggested by [46]:

$$H_{ct}(f) = \begin{cases} \frac{-\frac{(f-7.72)^2}{20.12}}{0.97e^{-\frac{1}{20.12}}} & \text{for } f < 7.0\\ -9\left[\left|\log_{10}\left(\frac{f}{9.0}\right)\right|\right]^{23} & \text{for } f \ge 7.0 \end{cases}$$
(3.50)

where f is given in Eq. (3.47). This model is shown in Fig. 3.25.

Comparison of $H_{ft}(f)$ and $H_{ct}(f)$ shows that they are similar except for the location of the peak and the intercept value.

The $H_{ct}(f)$ model was used successfully with the DCT for transform image coding [48,49] and image quality assessment [46,48,49,50]. In this chapter, it is desired to use the properties of the HVS for image enhancement using different transforms introduced in Sec. 3.2. Having the HVS model for the DCT and exploiting the similarity between the DCT and the DC3T [10], we conclude that the $H_{ct}(f)$ model should be appropriate for both transforms as well. The extension of this model for the SRDFT is under investigation.

We experimented with the HVS models with the DFT, the RDFT, the DCT, and the DC3T. Our simulation results showed that the nonlinearity applied to the original image does not contribute any visible changes in enhancement, so it was ignored. A similar conclusion was reported in Ref. [50].

In our experiments, we assumed that the viewer is watching an image of size 512×512 with a physical size of 4×4 inches at a distance of 12 inches. We processed the image in blocks of sizes 8, 16, and 32. The image in the transform domain was multiplied by the HVS filter transfer function, and then inverse transformed. The processed block was then multiplied by a constant factor in the image domain in order to have the same energy as the original image block.

The initial experimental results with HVS-filtering were not satisfactory. An example is shown in Fig. 3.27. It is observed that there is too much noise due to excessive attenuation of low frequencies. To remedy this problem, we modified the HVS filter transfer function for low frequencies without changing it for high frequencies. Since the value of the transfer function given in Fig. 3.25 is less than 1.0 for all frequencies, the transfer function values for low frequencies were modified by taking the square-root of the values up to the frequency at which the transfer function has the maximum value of 1.0. In this way, the smoothness of the filter was preserved. The modified HVS filter is given by

$$H_{modct}(f) = \begin{cases} \frac{(f-7.72)^2}{40.24} & \text{for } f < 7.0\\ -9 \left[\left| \log_{10}(\frac{f}{9}) \right| \right]^{23} & \text{for } f \ge 7.0 \end{cases}$$
(3.51)

The modified HVS filter is shown in Fig. 3.25. We experimented with the modified HVS filter with the same conditions as described above, and the result with the DC3T is shown in Fig. 3.28. The noise effects were removed, and the enhancement results were satisfactory. The DFT and the RDFT as well as the DCT and the DC3T results were similar. Some other results are shown in Figs. 3.29 and 3.30. The application of the HVS model for the

21 a.S.

SRDFT case is under investigation.

1.1

3.7 Removal of Block Edge Effects by an Overlap-Save Method

In the experimental work discussed above, the enhancement results with the SRDFT, the DC3T, and the DCT were observed to have considerably reduced edge effects as compared to the DFT and the RDFT results. However, the edge-effects were still objectionable. In this section, we will discuss how to remove these effects by using overlapped blocks and saving only the central parts of the processed blocks (the terminology "overlap-save" should not be mixed with the overlap-save method of computing linear convolutions). The procedure which was used in the experiments is as follows:

The blocks are of size $N \times N$ and neighboring blocks overlap by N/2 pixels. After the processing of each block, the $N/2 \times N/2$ central portion of the block is saved and the rest is discarded. Obviously, the amount of overlap and how much of the block is saved can be modified.

The experimental results for the overlap-save method for different enhancement techniques, alpha-rooting, modified unsharp masking and modified HVS-filtering, are shown in Figs. 3.31 thru 3.34 with the SRDFT and the DC3T. It is observed that the edge effects are no longer visible. The results with N equal to 16 were basically the same as the results with N equal to 32. The smaller block size has the advantage of reduced number of operations, small storage requirements and allowing faster processing through parallel implementation.

3.8 Similarity of Modified Unsharp Masking and HVS Filtering

The HVS filter transfer function shown in Fig. 3.25 was obtained as a result of experimental studies of the human visual system. On the other hand, modified unsharp masking corresponds to two linear filtering operations. The first filter is controlled by a parameter C which determines how much the high frequencies in the image are boosted. The second filter has the function of removing high frequency noise to make the image smooth. The combination of the two filters is a single filter whose transfer function is given by

a she waa aha

$$H(n_1, n_2) = |H_L(n_1, n_2)(1-C) + C|H_N(n_1, n_2)$$
(3.52)

This transfer function is compared to the HVS transfer function for the 1-D case in Fig. 3.35 as a function of the parameter C. In this case, the HVS transfer function values were calculated based on Eq. (3.50) for an image of size 512×512 with a physical size of 4×4 inches viewed at a distance of 30 inches, and the horizontal values are the index of coefficients. $H_{L}(n_1, n_2)$ and $H_N(n_1, n_2)$ are the same as used for modified unsharp masking for a block size of 32. It is observed that the two transfer functions are very similar for C greater than 1.0, except for the very low frequency response, and the similarity increasing with increasing values of C. As discussed in Sec. 3.6, the given low frequency response of the HVS filter was experimentally found unacceptable for image enhancement. If this is corrected, the HVS filter and the modified unsharp masking filter become practically the same. In turn, we can conclude that the modified unsharp masking is in good agreement with the human visual system. By this analogy, it is interesting to consider whether the human visual system is adaptive in the sense of a control parameter or parameters as in the interactive control of the parameter C in modified unsharp masking.

3.9 Conclusions

的复数形式的过去式

A number of transforms which have applications in image transform coding have been investigated with respect to image enhancement. We have also discussed three different methods for image enhancement in the transform domain. These techniques have been further developed for blockwise processing of images. The simulation results indicate that those transforms, namely, the DCT, the SRDFT and the DC3T, which are best for transform image coding are also the best for image enhancement. They also provide reduced edge-effects due to blockwise processing even though such effects are still visible. The edge-effects due to blockwise processing can be completely removed by an overlap-save technique. The modified unsharp masking and the HVS-filtering are practically equivalent.

As a final conclusion, transform image enhancement yields highly satisfactory performance, is biologically sound, provides parallel models for

\$P\$ 不可能的这些问题的"可能的"的问题。我们还是我们的是一种问题。

implementation, and can be performed simultaneously with transform image coding.

N	Additions			Multiplicatios		
	SRDFT & RDFT	DCT	DC3T	SRDFT & RDFT	DCT	DC3T
8	20	29	34	2	12	7
16	60	81	89	10	32	19
32	164	209	222	34	80	51
64	420	513	535	98	192	131
128	1028	1217	1256	258	448	323
256	2436	2817	2889	642	1024	721
512	5636	6401	6538	1538	2304	1795

Table 3.1. The number of additions and multiplications for different transforms.

3.16.53





Figure 3.1.

The energy distribution of different transforms with different threshold levels (see text): (a) the DFT (123). (b) the DCT (92). (c) the RDFT (53). (d) the SRDFT (62). (e) the DC3T (64).



Figure 3.2. Generalized filtering with a fast transform.



Figure 3.3. Conventional alpha-rooting for image enhancement.



Sho

Figure 3.4. Modified alpha-rooting for image enhancement.



Figure 3.5. Original image "catbrain".



Figure 3.6. Enhanced image with conventional alpha-rooting with the DFT and α equal to 0.85.



Figure 3.7. Enhanced image with conventional alpha-rooting with the DCT and α equal to 0.85.



Figure 3.8. Enhanced image with conventional alpha-rooting with the SRDFT and α equal to 0.85.



Figure 3.9.

Enhanced image with conventional alpha-rooting with the DC3T and α equal to 0.85.



Figure 3.10. Enhanced image with the modified alpha-rooting with the RDFT and α equal to 0.85.



Figure 3.11. Enhanced image with the modified alpha-rooting with the DCT and α equal to 0.85.



Figure 3.12. Enhanced image with the modified alpha-rooting with the DCT and α equal to 0.7.



Figure 3.13. Enhanced image with the modified alpha-rooting with the DFT and α equal to 0.85.



Figure 3.14. Enhanced image with the modified alpha-rooting with the DC3T and α equal to 0.85.



Figure 3.15. Enhanced image with the modified alpha-rooting with the DC3T and α equal to 0.7.

ALCONTRACTOR



Figure 3.16. Enhanced image with the modified alpha-rooting with the SRDFT and α equal to 0.85.



Figure 3.17. Unsharp masking for image enhancement.

çar eştir.



Figure 3.18. Modified unsharp masking for image enhancement.



Figure 3.19. Enhanced image with the modified unsharp masking with the RDFT and C equal to 7.0.



Figure 3.20. Enhanced image with the modified unsharp masking with the DFT and C equal to 7.0.



Figure 3.21. Enhanced image with the modified unsharp masking with the DC3T and C equal to 4.0.



Figure 3.22. Enhanced image with the modified unsharp masking with the DC3T and C equal to 7.0.


Figure 3.23. Enhanced image with the modified unsharp masking with the DCT and C equal to 4.0.



Figure 3.24. Enhanced image with the modified unsharp masking with the DCT and C equal to 7.0.



Figure 3.25. The HVS spatial frequency sensitivity models for the DFT, the DCT, and the modified HVS model for the DCT.



Figure 3.26. Image enhancement based on the human visual system.



Figure 3.27. Enhanced image with HVS-filtering with the DCT.





99

Figure 3.30. Enhanced image with HVS-filtering with the RDFT.



Figure 3.31. Enhanced image with conventional alpha-rooting by the overlap-save method with the SRDFT, α equal to 0.7 and N equal to 32.



Figure 3.32. Enhanced image with the modified unsharp masking by the overlap-save method with the DC3T, C equal to 7, and N equal to 32.



Figure 3.33. Enhanced image with the modified HVS-filtering by the overlap-save method with the DC3T and N equal to 32.



Figure 3.34. Enhanced image with the modified HVS-filtering by the overlap-save method with the DC3T, and N equal to 16.

9. E. K.



Figure 3.35. Comparison of HVS-filtering with modified unsharp masking.

CHAPTER 4 TRANSFORM EDGE DETECTION

4.1 Introduction

Edge detection is of major importance in computer vision. Edge points are pixels at which abrupt gray level changes occur which may reflect the change in surface orientation, depth or physical properties of materials. Edges may indicate the object boundaries, and can be used for segmentation, registration and object identification.

What is considered an edge depends on the application. For example, in object recognition, it may be only the object boundaries which are the necessary edges to be detected.

There are several criteria which are considered most important in edged detector performance as follows:

- 1) The error rate which can be defined either as the probability of missing a true edge or as the probability of detecting a false edge due to
- existence of noise. 2) The edge points should be "localized" well. This means that the
- distance between points marked by the detector and the center of true edge is minimized.
- 3) The elimination of multiple responses near the true edge.
- 4) Computational complexity. Low computational cost enables the effective use of edge detection in real time applications.

Consider the 1-D function f(x) in Fig. 4.1. The point x_0 can be considered an edge. At this point, the first derivative f'(x) has a local extremum (maximum or minimum), and the second derivative has a zero- crossing. The steepness of the edge is indicated by the size of the extremum in the case of the first derivative and the slope of f''(x) at the zero-crossing point in the case of the second derivative.

The generalizations of f'(x) and f''(x) to the 2-D case with the function f(x,y) are the gradient $\nabla f(x,y)$ and the Laplacian $\nabla^2 f(x,y)$, respectively, given

by

$$\nabla f(\mathbf{x}, \mathbf{y}) = \frac{\partial f(\mathbf{x}, \mathbf{y})}{\partial \mathbf{x}} \mathbf{e}_{\mathbf{x}} + \frac{\partial f(\mathbf{x}, \mathbf{y})}{\partial \mathbf{y}} \mathbf{e}_{\mathbf{y}}$$
(4.1)

and

$$\nabla^2 f(\mathbf{x}, \mathbf{y}) = \frac{\partial^2 f(\mathbf{x}, \mathbf{y})}{\partial \mathbf{x}^2} + \frac{\partial^2 f(\mathbf{x}, \mathbf{y})}{\partial \mathbf{y}^2}$$
(4.2)

where e_x and e_y are the unit vectors in the x and the y-directions, respectively.

 $\nabla f(x, y)$ can be approximated in a number of ways in the discrete case, resulting in particular edge-detector operators. For example, Sobel edge-detector [51] and Roberts operator [52] are such detectors.

Methods based on the gradient are usually sensitive to noise. Since the magnitude of the gradient is compared to a threshold to decide the existence of an edge, the edges obtained are usually thick, and an edge-thinning algorithm may be necessary to improve the results. Gradient-based edge detection algorithms may also cause discontinuities in the detected edge contours.

In Laplacian-based methods, choosing all zero-crossing points as edges tends to generate too many edge points, and many false edge contours may be generated. One advantage of Laplacian-based methods is that edges are thin, and edge-thinning algorithms are not needed.

The disadvantages of the methods discussed above can be reduced by a low-pass filter prior to edge detection operation. For example, a commonly used filter for smoothing is the Gaussian filter given by

$$h(x,y) = e^{\frac{-(x^2 + y^2)}{2\sigma^2}}$$
(4.3)

where σ^2 is inversely related to the cutoff frequency. Different σ 's corresponds to different degrees of smoothing of the image, and can be used to obtain edge maps of different scales [53].

The complex Fourier transform (CFT) of h(x,y) is given by

$$H_{c} (f_{x}, f_{y}) = 2\pi\sigma^{2} e^{-4\pi^{2}\sigma^{2} \frac{(f_{x}^{2} + f_{y}^{2})}{2}}$$
(4.4)

It is observed that both f(x,y) and $H_c(f_x,f_y)$ are Gaussian and thereby smooth and localized. Such a filter is desirable so that the locations of the edges are not altered and false edges are not created. Yuille and Pogio [54] showed that the Gaussian filter is the only filter which does not create false edges with different scales.

The Gaussian filter is commonly followed by the Laplacian. We observe the following:

$$\nabla^{2}\left[f(x,y) ** h(x,y)\right] = \nabla^{2} \int_{-\infty-\infty}^{\infty} \int_{-\infty-\infty}^{\infty} f(x',y')h(x-x',y-y')dx'dy'$$
$$= \int_{-\infty-\infty}^{\infty} \int_{-\infty-\infty}^{\infty} f(x',y')\nabla^{2}h(x-x',y-y')dx'dy'$$
$$= f(x,y) ** \nabla^{2}h(x,y)$$
(4.5)

 $abla^2 h(x,y)$ is given by

$$d(x,y) \equiv \nabla^{2}h(x,y) = \left(\frac{x^{2} + y^{2} - 2\sigma^{2}}{\sigma^{4}}\right) e^{\frac{-(x^{2} + y^{2})}{2\sigma^{2}}}$$
(4.6)

The CFT of d(x, y) is

$$D_{c} (f_{x}, f_{y}) = -2\pi\sigma^{2} \left(f_{x}^{2} + f_{y}^{2} \right) e^{-4\pi^{2}\sigma^{2} \frac{(f_{x}^{2} + f_{y}^{2})}{2}}$$
(4.7)

d(x,y) and $D_c(f_x, f_y)$ are shown in Fig. 4.2 for σ^2 equal to 1. A one-dimensional cross-section of d(x,y) is shown in Fig. 4.3. It is clearly observed that d(x,y) is a bandpass filter.

The edges are detected by finding the zero-crossings of the bandpassfiltered image. Which edges are found depends highly on the value of σ^2 . This fact is made use of to generate edge maps of different scales corresponding to different values of σ^2 in image understanding [53].

d(x,y) can be approximated by d'(x,y) equal to the difference of two Gaussians:

$$d'(x,y) = e^{\frac{-(x^2 + y^2)}{2\sigma_1^2}} - e^{\frac{-(x^2 + y^2)}{2\sigma_2^2}}$$
(4.8)

The CFT of d'(x,y) is

$$D_{c}'(f_{x},f_{y}) = 2\pi \left[\sigma_{1}^{2} e^{-4\pi^{2}\sigma^{2}} \frac{(f_{x}^{2}+f_{y}^{2})}{2} - \sigma_{2}^{2} e^{-4\pi^{2}\sigma_{2}^{2}} \frac{(f_{x}^{2}+f_{y}^{2})}{2} \right]$$
(4.9)

Shanmugam et al. [55] looked for a linear space-invariant bandlimited filter to minimize the error in localization. They further assumed that the filter is rotationally invariant and has a zero response to the slowly varying input. Exploiting the properties of prolate spheroidal wave functions and representing them by a closed form asymptotic approximation, and also incorporating a correction by Lunscher [56], the Fourier transform of the corresponding desired filter for one-dimensional step edge is given by

$$H(w) = \begin{cases} K w^{2} e^{\frac{-cw^{2}}{2\Omega^{2}}} |w| < \Omega \\ 0 & \text{otherwise} \end{cases}$$
(4.10)

which is valid if $|w| < \Omega c^{-\frac{1}{4}}$. In Eq. 4.10, $c = \frac{\Omega I}{2}$ where I is the desired resolution interval for detecting the edges. In this method, edges can be found by marking the zero-crossing points. Clearly, Eq. 4.10 is also a band-pass filter. Lunscher compared the above filtering method to that of Marr and Hildreth and concluded that they are identical [57].

Canny applied optimization theory to edge detection [58]. He considered three main criteria for the edge operator:

a) Low error rate which is inversely proportional to signal to noise ratio.

b) Well-localized edge.

c) Removing multiple responses.

Considering the above constraints in mathematical form, Canny looked for an edge detector for a specific kind of edge function, the step edge, and found

that the optimal operator is close to the first derivative of a Gaussian function. In his method, edges are marked at maxima in gradient magnitude after convolving the given image with the optimal filter. Since the first derivative of a Gaussian function has a Fourier transform which has bandpass behavior, the optimal edge detection can be coarsely approximated as bandpass filtering followed by local maximum detection.

In conclusion, most filters for edge detection discussed in the literature are band-pass filters, with different models and parameters. Such filters have often been derived for a specific kind of edge such as step edge. In practice, there are quite a few kinds of edges in a real image, e.g. line edges, roof edges and ramp edges. Consequently a specific kind of filter can not be optimal for all types of edges. Therefore, any derived filter by any optimal method for a specific type of edge would be sub-optimal in real applications.

The computational complexity for real-time processing can be reduced if the filtering is done in some transform domain, using the advantages of fast algorithms. The processing time may be reduced further using parallel processing if the filtering is done blockwise. In the past, the complex discrete Fourier transform (DFT) has been the leading transform for filtering in the transform domain. The convolution of the filter impulse response with the signal in the signal domain corresponds to the multiplication of the DFT coefficients of the filter and signal in the transform domain if the two sequences of length N and M, respectively, are extended by zero padding to length greater than or equal to N+M-1. The DFT has two major drawbacks. First, it needs complex additions and multiplications which are computationally expensive, and represents data as complex in the transform domain. In addition, it creates severe "block effects" when applied blockwise.

In this chapter, we investigate transform edge detection with Fourierrelated transforms introduced in Chapter 3, in which the image is processed in small blocks. In the sense of generalized bandpass filtering, edge-detection is carried out by multiplying the coefficients of the signal in the transform domain by a mask. We also show that these transforms create less "block effects" than the DFT when applied blockwise. An overlapping method is proposed to eliminate the "block effects" with a little increase in the computational complexity. We also show that interpolation can be implemented with these transforms in order to increase the resolution of edge location and to decrease the effects of inherent noise in the real images. Simulation results are discussed for each transform and method.

4.2 Bandpass Masking

Different transforms map the given signal in the space (or time) domain to another domain, usually called transform domain, in a form that is specific to a particular transform. Fig. 4.4 shows frequency distributions of different transforms with typical images whose energy can be considered to be concentrated at low frequencies.

The DFT and the RDFT have similar frequency distributions in a sense that the low frequency components of the signal can be considered to be in the four corners of the two dimensional transform window. The high frequency components can be considered to be in the middle of the window. One can change the location of different frequency components buy the signal by $(-1)^{n+m}$ and thereby centering the low frequency components in the middle, but we are using them without centering. For the DCT and DC3T, the low frequency components are in one corner and the high frequency components in the opposite corner. The SRDFT, which results from a specific permutation of the input signal followed by the RDFT, has a frequency distribution very similar to the DCT and the DC3T. In fact, the input permutations move most of the low frequency energy from three corners in the RDFT domain to that corner with the DC coefficient. In this chapter, we assume that the SRDFT has the same frequency distribution as the DCT or the DC3T.

As we discussed before, most edge detection methods result in bandpass filtering. Now, having the location of different frequencies for each transform, we want to design bandpass masks which attenuate the low and high frequencies but keep the middle frequencies unchanged. The two major issues are the shape, the low and the high frequency cutoffs for the desired masks. Certainly, the desired masks can not have very sharp cutoffs since this would lead to oscillations similar to Gibbs phenomenon which lead to multiple edges. There are many different possibilities for the shape of masks. In this chapter, we used two types of bandpass masks for edge detection, and they are given as

$$H(i,j) = K u e^{-\sigma^2 u^2}$$
 for $i,j = 0,1,...,N-1$, (4.11)

$$H(i,j) = K u^{2} e^{-\sigma^{2}u^{2}} \text{ for } i,j = 0,1,...,N-1, \qquad (4.12)$$

where $u = \sqrt{u_i^2 + u_j^2}$, and

k for k = 0, 1, ..., N-1, for the DCT, the DC3T and the SRDFT

$$u_{k} = \begin{cases} k & \text{for } k = 0, 1, ..., \frac{N}{2}, \\ N-k & \text{for } k = \frac{N}{2} + 1, ..., N - 1, \end{cases} \text{for the RDFT}$$
(4.13)

K is some constant value parameter that can be adjusted to make the mask coefficient values integer-valued, which makes implementation easier in a fixed-point environment. In Eqs. 4.11 and 4.12, the single parameter σ^2 is used to adjust the peak location and the cutoff frequencies. We will refer to the masks given in Eqs. 4.11 and 4.12 as type I and type II, respectively. Fig. 4.5 shows the 1-D graph of these masks for some arbitrary value of the parameter σ^2 . Mask type I has its peak value at $u = \frac{1}{\sqrt{2\sigma^2}}$, and mask type II has its peak value at $u = \frac{1}{\sqrt{\sigma^2}}$. The 2-D graph of these masks for different transforms are given in Fig. 4.6. The single parameter σ^2 in each case is adjusted for each particular transform to give the best results.

As pointed out earlier, computation complexity of an edge detection algorithm is an important design criterion, specially in real time processing. In general, signal processing with different fast transforms involves less computation, but that may not be enough for real time applications. To remedy this problem, blockwise processing has been introduced and has been applied to image transform coding and image transform enhancement. In this chapter, we introduce blockwise processing for edge detection.

The blockwise transform edge detection method comprises of the following:

1). The given gray scale image is segmented into a number of small blocks.

2) Each block is transformed by the chosen transform.

- 3) The transform coefficients of each block are modified by multiplying with the mask coefficients.
- 4) The modified coefficients of each block are inverse transformed.
- 5) Zero-crossing points in the whole processed image are found, and the "slope" of zero-crossing for each point is calculated by the following equation:

$$S = \sqrt{s_1^2 + s_2^2}, \qquad (4.14)$$

where s_1 and s_2 are the slope of the zero-crossing in the x and y directions, respectively. The slope of zero-crossing for each direction is defined to be the difference value of two points on each side of zero-crossing and is illustrated in Fig. 4.7.

- 6) The slopes are normalized to integer values between 0 and 255.
- 7) Since all zero-crossing points are not associated with edge locations, a thresholding procedure is applied to remove the false edges due to noise.

Removing the false edges results in deleting some true edges which are "weaker" than others, and detecting all true edges results in creating some false edges. It is desirable to have a scheme to determine "optimal" threshold which has been used to compare different edge detection methods when the true edge locations are known. One such scheme is to choose the threshold value such that the number of missed edges points are close to the number of created false edges [59]. In other words, the threshold value is chosen such that the conditional probability P(AE | TE) of the assigned edge given true edge, and the conditional probability P(TE | AE) of the true edge given the assigned edge are as close as possible. P(AE | TE) indicates what percentage of true edge points has been assigned to be edge, and P(TE | AE) indicates what percentage of assigned edge points are true edges. If the number of missed edge points is close to the number of created false edge points, these two conditional probabilities will be close.

The transform edge detection method was simulated with the DCT, the DC3T, the RDFT and the SRDFT transforms. These transforms are described in Chapter 3. The block sizes of 16 and 32 were used for blockwise processing. We tested both type I and type II masks given in Eqs. 4.11 and 4.12. In order to compare the results of edge detection method given in this chapter to the results of some other methods given in [59] and [53], we used an original checkerboard image. Each square has a size of 20×20 . The dark squares and the bright squares have gray values of 75 and 175, respectively. Independent Gaussian noise with zero mean and a standard deviation of 50 was added to the perfect checkerboard image. Defining the signal to noise ratio as 10 times the logarithm of the range of the signal divided by the rms of the noise, the noisy checkerboard image has a signal to noise ratio of about 3 dB. The

perfect and noisy checkerboard images are shown in Fig. 4.8.

In transform edge detection method, we tried coarsely to find the best parameter σ^2 with both types of masks with different transforms and different sizes of blocks. Defining the true edge position the two pixel wide range region in which each pixel has some neighboring pixel with different gray value [59], we counted the number of false and missed edge points to calculate the two conditional probabilities defined above. We also calculated the error distance defined as the average distance to closest true edge pixels of pixels which are assigned nonedge but which are true edge [59]. Simulation results with two type of masks and different transforms with block sizes of 16 and 32 are given in Table 4.1. The parameters are the values of σ^2 in Eqs. 4.11 and 4.12. In these simulations, we used an overlap-save method discussed in Chapter 3 to eliminate block effects due to the use of small blocks. We also discuss the case without overlapping later.

The results show that the DCT and the DC3T give completely similar results for both type of masks and different block sizes. This was expected from the relation between the DCT and the DC3T given in Chapter 3. The SRDFT gives results close to the DCT or DC3T, but overall the DCT or DC3t give a little better results. The RDFT gives the best results for a block size of 32, but for a block size of 16, the results are a little worse than other transforms. The two type of masks give very similar results. The results with a block size of 32 are much better than those for a block size of 16. Of course, this has the disadvantage of more computational cost. Images of edge detection results with transform edge detection method are shown in Figs. 4.9-4.16. Similar conclusions in comparing the results of different transforms can be seen in Figs. 4.15 and 4.16.

For avoiding any difference in the implementation of other edge detection algorithms, we used the results published in [59] and [60] with the same original image, but different sizes. The numerical results for Prewitt [61], zero-crossing of Laplacian [53] and second directional derivatives [59] methods

are given in Table 4.2. Comparing the results given in Table 4.1 to those given in Table 4.2 reveals that the results of transform edge detection method for a block size of 16 are much better in terms of conditional probabilities and very close to the results of the other techniques in terms of error distances. For a block size of 32, the results of transform edge detection method are better in terms of both conditional probabilities and error distances. Some different results was given in [60] for zero-crossing of Laplacian method for the same window size but different parameter. Comparing those results to the results given in Table 4.1 show that the results of transform edge detection are still comparable for block size of 16 and much better for block size of 32. For comparison, images of the results of the above edge operators are given in Figs. 4.17 - 4.20.

We also simulated transform edge detection method for the real images, "girl256" and "catbrain" given in Figs. 2.4 and 3.6, respectively. In this simulation, we used different number of overlapping pixels, including the nonoverlapping case. The results for different number of overlapping pixels and with different transforms, including the results for the noisy checkerboard image, are given in Figs. 4.21-4.35. These results indicate that two pixel overlapping is basically enough for the DCT, the DC3T and the SRDFT for avoiding visible block edge effects. For the RDFT, the block edge effects are visible for 4 or less number of overlapping pixels, but for 8 pixel overlapping, the block edge effects are not visible.

Regarding the computational complexity of the transform edge detection method, we calculated the number of real additions and multiplications per pixel for different transforms and for different number of overlapping pixels. We have used the data given in Table 3.1, and we have included one real multiplication per pixel for applying the bandpass mask. The results are given in Table 4.3. The convolution with a filter of size 11×11 as used by Refs. [53] and [59] can also be done using fast convolution. In this case, a window size of either 6×6 or 22×22 of the input signal is expanded to a

window size of 16×16 or 32×32 , respectively, by zero padding. Using the fast methods given by Ersoy and Hu [64] for 2-D real circular convolution, the number of real additions and multiplications per pixel for the case of 16×16 are 128.06 and 28.39, respectively. The number of real additions and multiplications per pixel for the case of 32×32 are 50.25 and 12.16, respectively. Comparing these numbers with those in Table 4.3 reveals that the number of additions and multiplications per pixel for transform edge detection method are on the average lower for 4 or less number of overlapping pixels than the corresponding numbers in linear filtering.

4.3 Interpolation

Interpolation is the generation of new signal sample values between known signal sample values. There is evidence that the human visual system does interpolation in both time and space [62]. Image interpolation has many applications in image processing such as edge detection, coding and signal representation. There are many different methods for interpolation such as use of splines of different orders. Another type of interpolation involves using transform methods. For example, the discrete Fourier transform (DFT) has been used for interpolation. In this method, first, the DFT of the N-point signal is computed. Then, the signal is padded with zeros in the transform domain to the desired number, say M, points. Finally, one M-point inverse DFT is applied. However, the DFT is a complex transform and involves complex additions and multiplications which have high computational cost. In this section, we propose interpolation with the 2-D real discrete Fourier transform (RDFT) which needs just real additions and multiplications and, therefore, reduces the computational cost. The underlying procedure of interpolation by the RDFT is the same as that by the DFT, except generalized

的编码和编码的编辑

RDFT's are involved after the first transformation of the input signal. Ersoy [63] has shown that the computational cost of interpolation by RDFT can be reduced by applying generalized inverse RDFT's of the same size as the original image instead of taking a large inverse RDFT after zero padding of twice the size.

We simulated interpolation with the RDFT in transform edge detection method. We first subsampled the original "girl256" image given in Fig. 2.4 by a factor of 2. Taking blocks of size 16 with 4 pixels overlapping, we first applied the edge detection mask after forward RDFT. Then the blocks were padded to size 32 with zeros (the zeros are padded in the middle of the transform block). Finally an inverse RDFT of size 32 with 8 pixels overlapping was applied for each block. The zero-crossing detection and calculating the slope at zero-crossing points was the last step which is done for the whole image. The interpolation results are given in Fig. 4.36 for different parameters of the bandpass mask type II. Comparing Fig. 4.36 with Fig. 4.31 shows that the results of transform edge detection with interpolation are very close to the the results of transform edge detection with the original image.

4.4 Conclusions

Blockwise transform edge detection method in the form of generalized bandpass masking with a number of different fast real transforms was proposed. The transform edge detection method consists of modifying the transform coefficients of small blocks of input image by pre-designed bandpass masks, followed by locating the zero-crossing points and calculating their slopes. The final step of edge detection method is thresholding. An overlapsave method was applied for removing the block edge effects. Simulation results show that the proposed transform edge detection method is, on the average, better in terms of simulation results than previous edge detection methods based on bandpass filtering. In addition, it has the advantage of less computational complexity, even with high number of overlapping pixels, which is essential in real time applications. The simulation results also show that overlapping with about 2 pixels (4 pixels with the RDFT) is sufficient for removing visible block edge effects in most cases.

고가 안망 물건을 가지 않는 것이 없다.



Figure 4.1. Input signal f(x), its first derivative f'(x), and its second derivative f''(x) for a typical 1-D edge.



Figure 4.2. Graph of $-d(x,y) = -\nabla^2 h(x,y)$ and its Fourier transform, $-D_c(f_x, f_y)$.





Figure 4.3. Graph of 1-D cross-section of d(x,y) and its Fourier transform.







North Al

124

The RDFT and the DFT



The DCT, the DC3T and the SRDFT

Figure 4.4. Frequency distributions of different transforms with a typical image.









Figure 4.6. 2-D graph of bandpass masks of different types and different transforms.

a hard day










2-D mask type II for the RDFT



Figure 4.7. Zero-crossing and its slope in 1-D.



Figure 4.8.

The perfect and noisy checkerboard images.

Transform	Block size	Type of Mask	Parameter	P(AE/TE)	P(TE/AE)	Error Distance
DCT	16	1	0.075	0.863	0.861	1.326
&		I II I	0.20	0.852	0.855	1.354
DC3T	32	II	0.04	0.924	0.923	1.169
	16	I	0.175	0.835	0.834	1.427
		II	0.30	0.822	0.820	1.421
RDFT		II	0.35	0.817	0.817	1.416
	32	II	0.25	0.948	0.973	1.294
		Π	0.11	0.925	0.927	1.140
		I	0.075	0.846	0.844	1.352
	16	II	0.10	0.833	0.835	1.287
SRDFT		Π	0.15	0.856	0.856	1.539
		II	0.20	0.865	0.864	1.474
	32	II	0.0625	0.912	0.911	1.242

Table 4.1.Simulation results for transform edge detection with noisy
checkerboard image and 4 pixels overlapping.



Figure 4.9. Zero-crossings with mask type II, the DCT or the DC3T, block size of 16, parameter values from 0.05 to 0.40 with increment of 0.05 and 4 pixels overlapping.

	N.C.		ger si,	te jî	: î		s	£ 1		(e				×.,			112			i ng ti				١.
Ø.		174	S.	K.	92	N.F.	2	25	ર્સ	8	ţ،Ç	4	2	1.2	\¥	5	<u>(</u>	12	8	158	ŭ	1	69	
N 3	2		Ż	₽Ĵ2	G	ŝQ	5	Q	2	20	ି. ଚ	0.]	٥ [Ω	2	70	<u>s</u> .	0	<u> </u>	<u>[</u> 2	8	10	Q.	Ī
jež.	02) -	<u> </u>	A\$	еç,	3	يوزي	<u>ç.</u>	<u>ج</u>	30	60	ک	٦	- -	2		٢.	0) ⊽		2	2	P.	0	Ī
300	35	80	<u>نگ</u>	29	A.	1.5	•0	₽ e 0 0	6.)	ß	3	2	٩	•	-8		2	Д	D		. B	ہ]	6	
Č,	20	3	33	88	Ø	13 5	8	3	Ø	D *	683	37	90	0) ŋo	B	3	?:	9	0	60	ß	à	
	Ş.	(%	65	ઝિઝ	බ	12.51	3	(B)	68	30	2	<u>(</u>	3	•	<u>a</u>	8	(9	۲	2	A	Po	Ľ
ン () - イン (80	3.	123	2	ેલ	ې د ^د ۱	5	រា	5	5	1.6		۵.	গ	20	27	10		0	3	5	2) ti	
0) ō	\overline{a}	2	60	2	a	50	2	2	5 Q	2	a	0	2	ر م	.0	α.	a	0	D	P	Q	a.	Ī
20	0	7	5-	P.	0	38	0	D		P. 1	a	٦٣	0	D		P .	0	29	ם	P		P.	a	Ē
	D) 	σ,]		0		D		J,		0	Ĩ	a					Ē	۵	•	2			• • •
7:	97	0	Ca	0	B	7 -	9	0	Ы	0	ы	7	1	ฮ	5	0	Ъ	7	. 90	១	Ы	0) E	
<u>م</u>	Ŋ	6	Śõ	6	2	R	<u>a</u>		5	6	ิล	្រវា	2) B	0	โล	2			۵)	<u>s</u>		
溪	₩,	35	3	K	Ŗ	r s	З°	ટ્રેડ	3	X	Ly Ly	ૻૢૢૡૢૼ	3	్ధక	:মু	K	64	ر مېزې	\$	৾য়	স্থ	Ş	お	0
R	23	(Q)	På	05	ŝ	8,723	25	Ş.	Ps	2	G.	800	2	8.	.0	ob	G.	0	Õ	ি		0	0	-
<u>نې</u>	Ŕ	So	F\$	5	ŝ	Lig	2	5	AS	Co l	N	w?	2	.89	AS	S	$\widehat{\mathbf{v}}$	28	•0	0	~~ {	ŝ	ব	- -
ЭŚ.	S.	32	8	138	ŝ	53	<u>~</u> ~	28	28	8	(Å	રુજી	<u>~</u> 0	28	්රී	8	9	स्टि	d.	00	20	8		Ś
ZZ.	3	R	1II	88	6	S.S.	30	ধি	Q.	228	PA	8,2	40	G	Ŵ	98	20	રી કે	4	0	Cu .	0 °	°	-
<u>s</u> e	R.	R.	Æ	8	\$	AL.	25	ଙ୍କି	A.	35	3	ET?	2	N	Ŕ	8	æ	6.51	5	5	N	.0	2	7
<u>រ</u> ଚ	S.	T	S.	8	5	-23-	e g	3.	ঙ্গি	<u> </u>	n' ^	175 4 5	5		क्ति]^.{		0		6		$\hat{1}$	-
$\widetilde{\circ}$	0	0	\int	0	0	0					•									\bigcap		7		-
201		0	{]	0	0	>>						5			\Box					\bigcirc		7		
کر ا	0)	٥٥		0	٥	\Box						\Box			\Box							\Box	- -
\sum		0	0		0		\Box	0								\Box		\Box				\Box		- -
9	0	•	(õ	[·]	•												\square			\square			\Box	
it was a second se									· · · · ·					_										

Figure 4.10. Zero-crossings with the DCT or the DC3T; the first two rows: mask type I, block size of 16, parameter values from 0.025 to 0.20 with increment of 0.025 and 4 pixels overlapping, the second two rows: mask type II, block size of 32, parameter values equal to 0.012, 0.015, 0.02, 0.028, 0.04, 0.0625, 0.11, 0.25 and 4 pixel overlapping.



Figure 4.11. Zero-crossings with mask type II, the RDFT, block size of 16, parameter values from 0.05 to 0.40 with increment of 0.05 and 4 pixels overlapping.

																		10 H H	- N					
			25					溪		157	诚		13	174			R	同語。	192	} %	1		19%	•
		λŔ				Č.	ΪŔ			43			13	E.	ß	E.			S	(\$.		1.2	12	1
		S.					12		25	ŧ.		iĝ.	92	1 1 1 1 1 1		E	ner ner	1305	0D	<u></u>		le;	ing)	-
								120	÷.		ff;	₩	5	190	13	No.	in the second se	1	à	100	1	32.		5
5									đi.			ize.	1.20	(ĨŇ	83	63	272	-20	8	1.08	189	En.	5.
÷,				36				Č4						(\$					3	È	60		in the	
							1	1 3.				1	150	<u>ب</u>			3	· · · · · ·	1	fra-	(:			<u>م</u> ترا
		49. 		18	12×	میں میں	(¥		200	B.	الأده		(Ľ	جهر سبب	2) <u> </u>	**¥	1 mil	(¥		1-4		12	مبأ
Ö	ର	æ	لي م	15%	S.	i@	12	8	2	10	<u>}.</u>	0	12	E.	2 X	10	<u>.</u>	<u>Q</u>	5	2	22	(00)	8.	ب
jegi Sin di	0	3	2	50	12	jog	<u>وي</u>	} R	ÊĽ	60		^و يز {	(02) (02)	<u>}</u>	Z.	Co	2	مرد مرد	<u>,</u>	27		E.		
<u>i</u>	<u>ي</u> ن ا	00	20		1	्रि ।	<u>.</u>	00		8	3	<u> </u>	<u>[.</u> 0		1.00	1	2	<u>'S</u>	٩	<u> </u>	<u>3</u> -	6	2	_
3	40	ТС [•]	. ÇŞ	13.0	e ge	.35.	30	23	{ <u>(</u>]	po	Ê	35	R 0 •	8) P	E	25	90	ধ		D.	P CA	
6 22	č 6	ά¢.	82	20	8	đ.	4	(* *	6.5	30	2	12	্ৰ		60	20	ঝ	\$ 3'(8	-	éñ	30	ઝ	<u> </u>
	395			(set		¥65	100	Refer		ī.	EE.	i se	295		20		19.3	ĺ¥∰.	int.	Ð.	2.0	her	60	ĺ.
					61		3						Б.				ΚĒ		575	ŝ		e e	Æ	2
								ġ,								ła,				E.		100		202
		Ľ,																	irin Nite					ŝ
		92	Ś											H					PŞ.		1272		08 72	A R
					2														E.				ζË,	e Z
		58			33									ČÉ.										3 8
ġ,	$\mathcal{G}_{\mathbb{Z}}^{*}$	33	Š.	12	Ċ,	8 2%	7	S.	S.	Ŀ	d:	8	6	• ••	30	8	3				}			
Č2	25	Ê		66	15	i di	马	S.	105	50	G	60	0	8			a			\bigcap				
	Ŕ	ŝ	AN C	E.		L.P	2	530	28	S.	ST ST	و در	· •	0	•		0				\int	$\left \right\rangle$		ŀ
<u>کې</u>	<i>e</i> C	3	88	R	đ,	2.3	29	SS SS	20	R	3	00	10	.	00	8	9							مبر ا
ŝ	3		J.	28		Č.S	40	B	20	23	CAN 1	E A		0	0	0.	P	~	<u>ب</u>	\sim	1			
Ŕ	Sec.	୍ଦିଭୁ		33	No.	ê F	25	ବ୍ୟି	A	35	a	e a	5	$\overline{(}$	(~)	0	3		~		$\langle \gamma \rangle$	\sim	\neg	
~	i ai		5	50		50	5	5	i cari	100	0 1		}		5.							لسيا	<u> </u>	-

Figure 4.12. Zero-crossings with the RDFT; the first two rows: mask type I, block size of 16, parameter values from 0.025 to 0.20 with increment of 0.025 and 4 pixels overlapping, the second two rows: mask type II, block size of 32, parameter values equal to 0.012, 0.015, 0.02, 0.028, 0.04, 0.0625, 0.11, 0.25 and 4 pixel overlapping.



Figure 4.13. Zero-crossings with mask type II, the SRDFT, block size of 16, parameter values from 0.05 to 0.40 with increment of 0.05 and using 4 pixels overlapping.

	ТĘ,	Ž .		读 1		[ß	13	ર્સ	17	የማ	(:24)	6	13	5	17	° 95	ن رو ا راهو]	j G	12	13	{V	, ମା	
Q,	G	Ş.		e,		Q.	ð	S.	2	80	}œ	[.a	0	3	8	80	0	0	0	2	2	$\overline{0}$	2	Ī
jeg j	S.	<u>.</u>	ويكون			ંગ્રે	V	<u>}</u>	68	C,	2) 0		<u>ج</u>		P,	0) 7⊽	Q. 1	P	E	P	10	5
	Ę.	80	30	¥,	Ĩ		Ŋ	0	0	10	9	2'	D)。	0		1 se	<u>[.</u>]	D		N			
3	32	ন্থ	12:	[\$??	2	13°1	90	5		100	P	.?~	4	0	(M	<u>]</u> []	P	3~	4	g	ы	6	12	
E.	133	Ľ	<u>Inž</u>	30	3	-51	\$	3	í Ś	34	à	5	4	4	6	57	3	<u>[</u> _9]	9	[]	S	6.	R	L
"ن`ز ب حو	E.	<u>រ</u>	13	15	સ	ں ر [*]	Ľ	<u> </u> 3	2	15	્સ	1		2	ă	15	0	<u> </u>	4	ฏ	2	21	្រា	
a	0	1	D	10) CL	0	0	2	2	PT)	2		0	B	(p	2) a.	<u>a</u>	õ	2	y	$\overline{\mathbf{v}}$	0.	آ سا
20		P	Ĩ	? .	a	27	2	P		<u> </u> 7.		2₫	2	P	<u>[] _</u>	5.	0	20	R.		<u>.</u>	97.		Ĺ
0	D	المرا		Ŀ	(°.	0	D		0	<u>.</u>		<u>a</u>	2		lo	<u>l</u>		D	D			- 1		-
2		đ	6	<u>ل</u> م	1 E	7.	ا ه" سعر	B	<pre>M</pre>	6	10			D .	12	B	L L		1	D m	1	B		
2	a	5	ß	<u>.</u>	R	No.	a	[<u>¬</u>	ß	مرا	R	12	a	[-:	5	Ь	ุ๊ก		D		5	<u> </u>	ন	Ļ
æ	18	* *	È	K	Ŕ	¥.	¥.	25		a.	<u>ل</u> ور ا		8	5.	1	8	θ	1. J. C.	8	હ		B	20	C
Res.	5	N.		100			15	8		60)	26	En:	\$	8	<u>]</u> <u>8</u>	40	20	0. 	n	9	2	0	0	
2	ĽŻ.	æ	Æ	2	X	36		25-	Æ	3	X).)\$ [22	39.	<u>eo</u>	ŝ		<u>مدر</u>	-0	2	201	-0	0	ļ
	Y	50	180		F	F.E	R	280	(R)	8	BI	<u>-</u> 25	32	200	<u>[</u>]	B	1º2	26	R		.0	8		
S.	<u>.</u>	প্র	3		Ē	<u>B</u>		ß		28	Ê	3.2	2	ន		28	Ć	3.		D		•	2	0
	100	19.34 	R	88	<u>&</u>	55	2	2	<u>S</u>	8	3	<u>દ્ધ</u> રુદ્ધ	2	[S]	5	. 99	8	50			19		3	
ン ア イ	l C	<u>S</u>	12			~	5	F					\bigcup						Ů				\Box	
0	Ó	0	0	0	0		•	ŀ	-	4														í
201		2			0	2						ļ										1		/- -
ٰک	ھ	2	0		اما	<u> </u>	ٺنا)	٥		<u> </u>				6		۱ <u> </u>							
<u>}</u>		0	, 6d								المرا		ليعوا						لي	<u>ل</u>		ĻJ	أسرا	
2	•	.	٩										Ļ		النسياً.								Ц	÷
	1 - S	0	V P	V AV		F - 1	6 T T		1 1	L 1		144 144			1.154 k.	L 1	19-1 - 1	1.11		- 2.5	4 J	1 1	n an an 📕	

Figure 4.14. Zero-crossings with the SRDFT; the first two rows: mask type I, block size of 16, parameter values from 0.025 to 0.20 with increment of 0.025 and 4 pixels overlapping, the second two rows: mask type II, block size of 32, parameter values equal to 0.012, 0.015, 0.02, 0.028, 0.04, 0.0625, 0.11, 0.25 and 4 pixel overlapping.



Figure 4.15. Zero-crossings after thresholding: counting from left to right starting at the top, each image corresponds to the entries of the first row to the 8th row of Table 4.1.

				y in is T		
1						
3	_	<u>)</u>	÷ (j.
			<u> </u>		ب ر : ا	
	Ĩ		[]	Ĩ		
			ار ک		[•••]	
				لترك		

ا د	, :		4		Lii	
ہے۔ ن <u>ب</u> ہ			ر ا، ا ا			
2	- - 1) 				}_
<u> </u>						
:			ار . اد _ ا	ار . ا ـ ـ ـ ـ		 -

				•	i ii	
•		E]	$\left[\overbrace{}\right]$		}
5	7			$\left[\overline{\mathbb{R}} \right]$		5
).]	<u>ر</u>]			ĺ
	Ī			<u>,</u>	 ا	1 1 1
	[]	$\left\{ \begin{array}{c} \cdot \\ \cdot \end{array} \right\}$	ر			
			1	<u> </u>		



Figure 4.16. Zero-crossings after thresholding: counting from left to right starting at the top, each image corresponds to the entries of the 9th row to the 13th row of Table 4.1.

Table 4.2. Compares the performance of three edge operators using an 11×11 window on the noisy checkerboard image [59].(Copyright IEEE, with permission)

	Prewitt	Marr-Hildreth	Directional Derivative
Parameters	Gradient Threshold = 18.5	Zero Crossing Strength = 4.0 $\sigma = 5.0$	Gradient Threshold = 14.0 $\rho = 0.5$
$\overline{P(AE TE)}$	0.6738	0.3977	0.7207
P(TE AE)	0.6872	0.4159	0.7197
Error Distance	1.79	1.76	1.16



Figure 4.17. Illustrates the edges obtained by the 11×11 Marr-Hildreth zero crossing of Laplacian operator set for three different zero crossing thresholds and three different standard deviations for the associated Mexican hat filter [59]. (Copyright IEEE, with permission)



Figure 4.18. Illustrates the directional derivatives edge operator for a window size of 11×11 and deciding that the true gradient is nonzero when the estimated gradient is higher than the thresholds of 12, 14, 16, or 18 [59].(Copyright IEEE, with permission)



Figure 4.19. Compares the directional derivative edge operator with the Marr-Hildreth edge operator and the Prewitt edge operator. The thresholds chosen were the best possible ones [59].(Copyright IEEE, with permission)

1	~		بشرا		اتر		-	بت	
2	0	D	8	8	0	0	0		4
•	0		0		0	0	•	•	4
•	0	9	8	30	0	P	80		1
•]	8	4	υ	0	0	•	9	•	6
7	0	0		0	0	0		•	1
5	0	2	9	0	•	0.	•	•••	Ì
٦	-	••	8	0	0	•	0	0	٢.
,	0			2	•	0	•	0	•
									Ι.
				(2	I)				Ċ.
-	<u> </u>)		μ_	<u>} </u>	 	ŀ
_		2	5		L,	$\lfloor 2 \rfloor$	<u>.</u>	Ĺ	Ŀ
		•	{``						{;
		k	۱۰.		- 1		•••		ſ
	7	7-	<u>[</u>	Ţ.~	1		1		Γ
. •••) -	1.	Ĵ	í T			T
-	Π	5	٢	(•••	İ –	(F)		7	Ì
		6		1	5 -		h) }	Ì
	.	\mathbf{h}	h	† ~	┢╴	ζ		┢╴	ſ
-	<u> </u>	<u>ن</u> -را	┢	┣	[<u>}</u>	<u> </u>	┢

Figure 4.20. (a) The zero-crossings obtained from the Marr-Hildreth implementation of the $\nabla^2 G$ operator with $\sigma = 2.5$. (b) Zerocrossings that remain after thresholding so as to equalize the conditional probabilities [60].(Copyright IEEE, with permission)



Figure 4.21. The results of transform edge detection with original image of "girl256", the DCT or DC3T, bandpass mask of type II, block size of 32, $\sigma^2 = 0.012$ and threshold value of 16: top-left, no overlapping; top-right, 2 pixels overlapping; bottom-left, 4 pixels overlapping.



Figure 4.22. The results of transform edge detection with original image of "catbrain", the DCT or DC3T, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 without overlapping.



Figure 4.23. The results of transform edge detection with original image of "catbrain", the DCT or DC3T, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 and with 1 pixel overlapping.



Figure 4.24. The results of transform edge detection with original image of "catbrain", the DCT or DC3T, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 and with 4 pixels overlapping.



Figure 4.25. The results of transform edge detection with original image of "catbrain", the DCT or DC3T, bandpass mask of type II, block size of 32, $\sigma^2 = 0.04$, threshold value of 7 and with 4 pixels overlapping.



Figure 4.26. The results of transform edge detection with original image of "girl256", the SRDFT, bandpass mask of type II, block size of 32, $\sigma^2 = 0.012$ and threshold value of 16: top-left, no overlapping; top-right, 2 pixels overlapping; bottom-left, 8 pixels overlapping.



Figure 4.27. The results of transform edge detection with original image of "catbrain", the SRDFT, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 without overlapping.



Figure 4.28. The results of transform edge detection with original image of "catbrain", the SRDFT, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 and with 1 pixel overlapping.



Figure 4.29. The results of transform edge detection with original image of "catbrain", the SRDFT, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 and with 4 pixels overlapping.



Figure 4.30. The results of transform edge detection with original image of "catbrain", the SRDFT, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 and with 8 pixels overlapping.



Figure 4.31. The results of transform edge detection with original image of "girl256", the RDFT, bandpass mask of type II, block size of 32, $\sigma^2 = 0.012$ and threshold value of 16: top-left, no overlapping; top-right, 2 pixels overlapping; bottom-left, 8 pixels overlapping.



Figure 4.32. The results of transform edge detection with original image of "catbrain", the RDFT, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 without overlapping.



Figure 4.33. The results of transform edge detection with original image of "catbrain", the RDFT, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 and with 1 pixel overlapping.



Figure 4.34. The results of transform edge detection with original image of "catbrain", the RDFT, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 and with 4 pixels overlapping.



Figure 4.35. The results of transform edge detection with original image of "catbrain", the RDFT, bandpass mask of type II, block size of 32, $\sigma^2 = 0.0625$, threshold value of 7 and with 8 pixels overlapping.

2月19日第日以

			Additions		Multiplications					
Block size	# of verla- pping pixels	SRDFT & RDFT	DCT	DC3T	SRDFT & RDFT	DCT	DC3T			
	0	15	20.25	22.25	3.5	9	5.75			
16	1	19.59	26.45	29.06	4.57	11.75	7.51			
	2	26.81	36.19	39.76	6.25	16.08	10.28			
	4	60	81	89	14	36	23			
	0	20.5	26.125	27.75	5.25	11	7.375			
	1	23.33	29.73	31.58	5.97	12.52	8.39			
32	2	26.77	34.12	36.24	6.86	14.37	9.63			
	4	36.45	46.67	49.34	9.33	19.56	13.11			
	8	82	104.5	111	21	44	29.5			

Table 4.3. The number of real additions and multiplications per pixel in transform edge detection method with different transforms, different block sizes and different number of overlapping pixels.





Figure 4.36. The results of interpolation with transform edge detection method, bandpass mask type II and threshold value of 16 for different parameters: top-left, $\sigma^2 = 0.04$; top-right, $\sigma^2 = 0.0625$; bottom-left, $\sigma^2 = 0.11$.

CHAPTER 5 CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

5.1 Conclusions

In this thesis, we have developed a unified approach to use novel transforms for transform image coding, enhancement and edge detection. Requirements for transform image coding are low bit rate and low complexity of implementation while keeping the quality of decoded image high. For these purposes, the optimal adaptive multistage transform coding has been introduced in Chapter 2. In this method, the optimality is achieved by minimization of the total final reconstruction error using marginal analysis. Both theoretical and experimental results indicate that optimal adaptive multistage image transform coding is certainly effective in reducing mean square reconstruction error over what is possible with one stage transform coding. This has been shown by generating the difference images in two different ways. The reconstructed images with multistage coding method are also subjectively much more preferable that the reconstructed images with the one stage coding method at the same bit rate.

In Chapter 3, transform image enhancement with a number of newly developed fast transforms has been investigated. We have discussed three different methods for image enhancement in the transform domain. In order to reduce the computational complexity of enhancement, all these techniques are implemented blockwise. The simulation results indicate that the DCT, the SRDFT and the DC3T are the best for image enhancement. For reducing the block edge effects due to blockwise processing, an overlap-save technique has been introduced. This technique is completely effective with a little increase in computational complexity.

In Chapter 4, we have discussed the transform edge detection. An investigation of many edge detection methods indicate that they are very similar to bandpass filtering. Exploiting the frequency characteristics of a number of real fast transforms, generalized bandpass filtering with two type of

a she and a she was to be a she to be

는 그는 것이 같은 것이 같은 것을 물질한 것은 것을 것을 했다.

bandpass masks has been proposed for edge detection. The transform edge detection method consists of modifying the transform coefficients of small blocks of input image by pre-designed bandpass masks, followed by locating the zero-crossing points and calculating their slopes. The final step of edge detection method is thresholding. Transform edge detection has been done blockwise and the overlap-save method discussed in Chapter 3 has been used for reducing the block edge effects. Simulation results show that transform edge detection is quite comparable with and generally better than other bandpass filtering methods despite its lower computational complexity.

5.2 Future Research Directions

The following issues will be considered as future research directions in the extension of previous chapter results:

- We have discussed the problem involving the implementation of multistage transform coding for 3 or more number of stages in Sec.
 2.5. The most obvious problem is the error due to mismatch between the assumed pdf and the real pdf for some coefficients in the course of bit allocation and quantization. It is suggested to study the implementation of the bit allocation method given by Shoham and Gersho [29] which may get closer to the optimum case by an iterative procedure.
- 2. The optimum mean square error quantization has been used for multistage coding in Chapter 2. It would be interesting to study the multistage coding method with uniform quantization which is the quantization base for newly adopted image coding standard [65].
- 3. It is suggested to study the implementation of the complete multistage transform coding procedure in the frame of the above mentioned standard for image coding with possible different number of stages for luminance and color components, Y, I and Q.
- 4. It is possible to investigate how to make optimal multistage transform coding totally adaptive in the sense of the system deciding itself how many stages it needs.
- 5. It is interesting to investigate optimal multistage transform coding with the inclusion of source coding in overall design. We observed that the number of bits in the second stage are mainly 0 or 1. It is possible that source coding such as Huffman coding may lead to further gains due to reduced overhead information in the second stage.
- 6. For image enhancement, the results of some applied transforms are promising. Therefore it is appropriate to improve these techniques with these transforms further. For instance, in modified alpharooting, it is interesting to investigate whether we can replace the windowing in the image domain by some proper filtering in the frequency domain to prevent more complexity with respect to conventional alpha-rooting.
- 7. One possible research direction could be using different alpha values with alpha-rooting enhancement method inside a processing block or different alpha values for different blocks based on the amplitude of coefficients or the energy of each block, respectively.
- 8. It has been observed in simulation results of transform edge detection that real pictures with different signal to noise ratio need different threshold values in the last step of edge detection. It is suggested to study the possibility of developing some adaptive technique to determine the best threshold for each picture based on some energy measure in the transform domain. It would not increase the complexity of implementation much since the transformed coefficients are already calculated during the course of applying bandpass masking.
- 9. Another direction in transform edge detection can be the study of designing some optimal masks for different transform which

probably give better results than the masks used in Chapter 4.

10. Another quantitative measure called Pratt figure of merit [27] can be used for comparing the performance of transform edge detection method with other methods. LIST OF REFERENCES

LIST OF REFERENCES

- [1] A. N. Netravali and J. O. Limb, "Picture Coding: A Review," Proceeding of the IEEE, Vol. 68, pp. 366-406, March 1980.
- [2] A. Habibi, "Survey of Adaptive Image Coding Techniques," IEEE Transactions on Communications, Vol. 25, pp. 1275-1284, November 1977.
- [3] A. K. Jain, "Image Data Compression: A Review," *Proceeding of IEEE*, Vol. 69, pp. 349-389, March 1981.
- [4] D. C. C. Wang, A. H. Vagnucci and C. C. Li, "Digital Image Enhancement," Computer Vision, Graphics, Image Processing Vol. 24, pp. 363-381, 1983.
- [5] A. N. Netravali and B. G. Haskell, Digital Pictures Representation and Compression, Plenum Press, New York and London, 1988.
- [6] P. A. Wintz, "Transform Picture Coding," *Proceeding of IEEE*, Vol. 60, pp. 809-820, July 1972.
- [7] J. L. Walsh, "A Closed Set of the Orthogonal Functions," American Journal of Mathematics, Vol. 45, pp. 5-24, 1923.
- [8] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete Cosine Transform," IEEE Transactions on Computers Vol. 23, No. 1, pp. 90-93, 1974.
- [9] O. K. Ersoy and C. H. Chen, "Transform-Coding of Images with Reduced Complexity, " Computer Vision, Graphics, and Image Processing, Vol. 42, pp. 19-31, 1988.
- [10] O. K. Ersoy and A. Nouira, "Image Coding with the Discrete Cosine-III Transform," Proceeding of Information Sciences and Systems Conference, John Hopkins University, March 1989, and to be published in IEEE Journal on Selected Areas in Communications, J-SAC, June 1992.

- [11] K. Rose, A. Heiman and I. Dinstein, "DCT/DST Alternate-Transform Image Coding," *IEEE Transactions on Communications*, Vol. 38, No. 1, pp. 94-101, January 1990.
- [12] J. Max, "Quantizing for Minimum Distortion," IRE Transactions on Information Theory, Vol. 6, pp. 7-12, March 1960.
- [13] W. H. Chen and H. Smith, "Adaptive Coding of Monochrome and Color Images," *IEEE Transactions on Communications*, Vol. 25, No. 11, pp. 1285-1292, November 1977.
- [14] D. J. Sakrison, "On the Role of the Observer and a Distortion Measure in Image Transmission," *IEEE Transactions on Communications*, Vol. COM-25, No. 11, November 1977.
- [15] A. K. Jain, Fundamentals of Digital Image processing, Prentice-Hall, New Jersey, 1989.
- [16] H. Hotelling, "Analysis of a Complex of Statistical Variances into Principal Components," Journal of Educational Psychology, Vol. 24, pp. 417-441, 1933.
- [17] K. H. Tzou, "Progressive Image Transmission: A Review and Comparison of Techniques," Optical Engineering, Vol. 26, No. 7, pp. 581-589, July 1987.
- [18] L. Wang and M. Goldberg, "Progressive Image Transmission by Transform Coefficient Residual Error Quantization," *IEEE Transactions on Communications*, Vol. 36, No. 1, pp. 75-87, January 1988.
- [19] L. Wang and M. Goldberg, "Progressive Image Transmission by Multistage Transform Coefficient Quantization," Proceeding of IEEE International Conference on Communication, Toronto, Ont., Canada, June 1986, pp. 419-423.
- [20] N. S. Jayant and P. Noll, "Digital Coding of Waveforms," Prentice Hall, New Jersey, 1984.
- [21] R. C. Wood, "On Optimum Quantization," *IEEE Transactions on Information Theory*, Vol. 15, No. 2, pp. 248-252, March 1969.
- [22] R. C. Reininger and J. D. Gibson, 'Distributions of the Two-Dimensional DCT Coefficients for Images," *IEEE Transactions on Communications*, Vol. 31, No. 6, pp. 835-839, June 1983.

- [23] L. D. Marascuilo and M. McSweeny, "Nonparametric and Distribution-Free Methods for the Social Science," Monterey, CA: Brooks-Cole, 1977.
- [24] A. Segall, "Bit Allocation and Encoding for Vector Sources," IEEE Transactions on Information Theory, Vol. 22, No. 2, pp. 162-169, March 1976.
- [25] B. Fox, "Discrete Optimization Via Marginal Analysis," Management Science, Vol. 13, No. 3, pp. 210-216, November 1966.
- [26] D. Luenberger, "Linear and Non-linear Optimization," Academic Press, 1986.
- [27] W. K. Pratt, "Digital Image Processing," Academic Press, 1978.
- [28] W. Mauersberger, "Experimental Results on the Performance of Mismatched Quantizers," *IEEE Transactions on Information Theory*, Vol. 25, No. 4, July 1979.
- [29] Y. Shoham and A. Gersho, "Efficient Bit Allocation for an Arbitrary Set of Quantizers," *IEEE Transactions on Acoustics, Speech, and* Signal Processing, Vol. 36, No. 9, September 1988.
- [30] R. J. Clarke, Transform Coding of Image, Academic Press, 1985.
- [31] J. G. Proakis and D.G. Manolakis, Introduction to Digital Signal Processing, Macmillan, 1988.
- [32] O. K. Ersoy, "Real Discrete Fourier Transform," IEEE Transactions on Acoustics, Speech, Signal Processing, Vol. ASSP-33, No. 4, pp. 880-882, 1985.
- [33] O. K. Ersoy and M. Zeng, "Nonlinear Matched Filtering and Neural Networks," Annual Optical Society of America Conference, Rochester, NY 198?.
- [34] H. Kitajima, "A Symmetric Cosine Transform," *IEEE Transactions on Computers*, Vol. C-29, pp. 317-323, April 1980.
- [35] M. J. Narasimha and A. M. Peterson, "On the Computation of the Discrete Cosine Transform," *IEEE Transactions on Communications*, Vol. COM-26, No. 6, June 1978.

- [36] W. H. Chen and S. C. Fralick, "Image Enhancement Using Cosine Transform Filtering," Proceeding of the Symposium of Current Mathematical Problem in Image Science, pp. 186-192, Monterey, California, 1976.
- [37] B. Chitprasert and K. R. Rao, 'Discrete Cosine Transform Filtering, " Signal Processing, Vol. 19, pp. 233-245, 1990.
- [38] H. C. Andrews, A. G. Tescher and R.P. Kruger, "Image processing by Digital Computer, "*IEEE Spectrum*, Vol. 9, No. 7, pp. 20-32, July 1972.
- [39] J. H. McClellan, "Artifacts in Alpha-Rooting of Images," Proceeding of the 1980 IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 449-52, April 1980.
- [40] J. S. Lim, Two-Dimensional Signal and Image Processing, Prentice Hall, 1990.
- [41] J. J. Depalma and E. M. Lowry, "Sine-Wave Response of the Visual System. II. Sine-Wave and Square-Wave Contrast Sensitivity," Journal of the Optical Society of America, Vol. 52, No. 3, March 1962.
- [42] O. D. Faugeras, "Digital Color Image Processing Within the Framework of a Human Visual Model," *IEEE Transactions on Acoustics, Speech,* and Signal Processing, Vol. ASSP-27, No. 4, August 1979.
- [43] D. J. Granrath, "The Role of Human Visual Models in Image Processing," Proceeding of the IEEE, Vol. 69, No. 5, May 1981.
- [44] J. L. Mannos and D. J. Sakrison, "The Effects of Visual Fidelity Criterion on the Encoding of Images," *IEEE Transactions on Information Theory*, Vol. IT-20, No. 4, July 1974.
- [45] T. G. Stockham, JR., "Image Processing in the Context of a Visual Model," *Proceeding of the IEEE*, Vol. 60, No. 7, July 1972.
- [46] N. B. Nill, "A Visual Model Weighted Cosine Transform for Image Compression and Quality Assessment," *IEEE Transactions on Communications*, Vol. COM-33, No. 6, June 1985.
- [47] R. J. Clarke, "Spectral Response of the Discrete Cosine and Walsh-Hadamard Transforms," Proceeding of the IEE, Vol. 130, Part F, No. 4, June 1983.

- [48] L. Carrioli and M. G. Albanesi, "Multigray Image Compression Based on the Human Eye Transfer Function," Alta Frequenza, Vol. LVII, No. 5, June 1988.
- [49] K. N. Ngan, K. S. Leong and H. Singh, "Cosine Transform Coding Incorporating Human Visual System Model," Proceeding of SPIE -Visual Communications and Image Processing, Vol. 707, 1986.
- J. A. Saghri, P. S. Cheatham and A. Habibi, "Image Quality Measure Based on a Human Visual System Model," Optical Engineering, Vol. 28, No. 7, July 1989. Table 3.1. The number of additions and multiplications for
- [51] R. O. Duda and P. E. Hart, "Pattern Classification and Scene Analysis, Wiley, New York, 1973.
- [52] L. G. Roberts, "Machine Preception of Three-Dimensional Solids," in Optical and Electro-Optical Information Processing, J. T. Tippett et al., Ed., Massachusetts Institute of Technology Press, Cambridge, Mass., pp. 159-197, 1965.
- [53] D. Marr and E. Hildreth, "Theory of Edge Detection," Proceeding of Royal Society of London, B 207, pp. 187-217, 1980.
- [54] A. L. Yuille and T. A. Poggio, "Scaling Theorems for Zero Crossings," *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol. 8, No. 1, pp. 15-25, 1986.
- [55] K. S. Shanmugam, F. M. Dickey and J. A. Green, "An Optimal Frequency Domain Filter for Edge Detection in Digital Pictures," *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol. 1, No. 1, 1979.
- [56] W. H. H. Lunscher, "The Asymptotic Optimal Frequency Domain Filter for Edge Detection," *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol. 5, No. 6, pp. 678-680, 1983.
- [57] W. H. H. Lunscher and M. P. Beddoes, "Optimal Edge Detector Design I: Parameter Selection and Noise effects," *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol. 8, No. 2, pp. 164-177, 1986.
- [58] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol. 8, No. 6, pp. 679-698, 1986.

172

- [59] R. M. Haralick, "Digital Step Edges from Zero Crossing of Second Directional Derivatives," IEEE Transactions on Pattern Analysis Machine Intelligence, Vol. 6, No. 1, 1984.
- [60] W. E. L. Grimson and E. C. Hildreth, "Comments on 'Digital Step Edge from Zero Crossing of Second Directional Derivatives'," *IEEE Transactions on Pattern Analysis Machine Intelligence*, Vol. 7, No. 1, 1985.
- [61] J. Prewitt, "Object enhancement and extraction," in *Picture Processing* and *Psychopictoric*, B. Lipkin and A. Rosenfeld, Eds. New York: Academic, 1970, pp. 75-149.
- [62] H. B. Barlow, "Reconstructing the Visual Image in Space and Time," Nature, Vol. 279, May 1979.
- [63] O. K. Ersoy, "Real-Time Interpolation of Images Obtained by Image Detector Arrays," Proceeding of the International Meeting on Image Detection and Quality, Paris, July 1986, pp. 371-374.
- [64] O. K. Ersoy and N. C. Hu, "Fast Computation of Real Circular, Skew-Circular, Linear Convolutions and Toeplitz Forms," presented in *PEEII* workshop on Digital Communication, Purdue University, April 1986.
- [65] "JPEG Technical Specification, Revision 5," JPEG-8-R5, January 1990.
- [66] T. Peli and D. Malah, "A Study of Edge Detection Algorithms," Computer Graphics and Image Processing 20, 1-21(1982).
- [67] A. Rosenfeld and A. C. Kak, Digital Picture Processing, Vol. II, Academic Press, 1982.



VITA

174

化正常分析 化合同分析 化合金合金 网络海绵

Sabzali Aghagolzadeh was born in Babol, Iran. He received the Bachelor of Science degree in electrical and electronic engineering in 1985 from Tabriz University, Tabriz, Iran, and the Master of Science degree in electrical engineering in 1988 from Illinois Institute of Technology, Chicago, Illinois. He was employed as a teaching assistant from 1985 to 1988. He attended the school of Electrical Engineering at Purdue University in August 1988. He has also been employed as a part time research assistant at Purdue University. His research interests include digital signal and image processing, image coding and communication systems, image enhancement and image analysis.