Copyright
by
Niranjan Damera-Venkata
2000

ANALYSIS AND DESIGN OF VECTOR ERROR DIFFUSION SYSTEMS FOR IMAGE HALFTONING

by

NIRANJAN DAMERA-VENKATA, B.E., M.S.E.E.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN
December 2000

ANALYSIS AND DESIGN OF VECTOR ERROR DIFFUSION SYSTEMS FOR IMAGE HALFTONING

APPROVED BY
DISSERTATION COMMITTEE:

Brian L. Evans, supervisor

Ross Baldick

Alan C. Bovik

Gustavo de Veciana

Wilson S. Geisler

Joydeep Ghosh

This thesis is dedicated to my family,
and my dear friends Maha and Wade.

Acknowledgments

First, I would like to thank my parents and family for supporting me, and encouraging my graduate studies. While many felt that I did not need a graduate education in the United States, my parents were open minded enough to let me convince them to allow me to pursue my goals with their complete support. I could never cope with life without the strong discipline and values my family instilled in me.

I would like to thank my advisor, Prof. Brian Evans for his support, encouragement and good wishes. Brian has had a great influence not only on my graduate studies as an advisor, but on my whole life. He taught me by example, the meaning of hard work, dedication, integrity and perfection. I have benefited a great deal from my interactions with him, many of them in the most informal of circumstances. I respect him for the patience he had in dealing with my absent-mindedness. I am forever indebted to him.

A thesis could not describe the amount I owe my two best friends, Maha and Wade. I would not be alive today if not for their diligent “babysitting”. The time spent in their company in Austin ranks without doubt as among the best moments in my life.

I would like to thank the folks at the Embedded Signal Processing Laboratory (ESPL) Wade, Güner, Biao, Magesh, Amey, Gummadi, David,

Milos, Clint, Greg and Serene for their wonderful and stimulating company. I am especially indebted to Bla Bla Biao for “looking out for me” and all of her help with Latex (without her help this dissertation would be in Word), and to my dear friend Güner who has helped me become a more practical person.

Among the folks in Prof. Al Bovik’s lab, I have learned a lot from Tom and Dong. Tom is and always will be a “jolly good” buddy of mine. This dissertation was directly influenced by Tom’s work and my early collaboration with him.

I would also like to thank Berna and Dave, who were visiting students from the University of British Columbia, for their friendship.

I would like to thank my committee members, Prof. Baldick, Prof. Bovik, Prof. de Veciana, Prof. Geisler and Prof. Ghosh (in alphabetical order). It is a truly a great privilege to have them on my Ph.D committee. Prof. Baldick teaches a great optimization theory course that has greatly influenced my research. Prof. Bovik’s undergraduate digital image processing course is one of the reasons that this dissertation is on digital halftoning. Al has inspired and encouraged me a great deal all throughout my graduate studies. Prof. de Veciana completely changed the way I used to think, by stressing intuition over blind rigor. My only regret, looking back at my graduate studies at UT Austin, is that I could not take all of the courses he had to offer. I am grateful to Dr. Geisler for his encouragement and for numerous discussions on image quality assessment. A chapter of this dissertation was born out of discussions with him. Although I did not have the opportunity to take one of

Dr. Ghosh's classes, he has encouraged and inspired me.

I gained valuable knowledge and inspiration from Prof. Irwin Sandberg, Prof. John Gilbert and Prof. Ward Cheney at the University of Texas and Prof. K. V. Parthasarathy of Vivekananda College, in Madras, India. These gentlemen introduced me to the power and beauty of mathematics.

My undergraduate advisor, Dr. P.S. Reddy of the University of Madras in India helped me make the transition to research and encouraged me to pursue a doctoral degree. Even as an undergraduate, he motivated me to wrestle with hard research problems, and made me believe in myself.

Finally, I would like to thank the great folks at Hewlett-Packard Laboratories Brian Atkins, Qian Lin, Neerja Raman, and Dan Tretter for their patience in letting me complete my dissertation remotely. I am especially thankful to Brian Atkins for several discussions I had with him on halftoning.

Niranjan Damera-Venkata
September, 2000

ANALYSIS AND DESIGN OF VECTOR ERROR DIFFUSION SYSTEMS FOR IMAGE HALFTONING

Publication No. ____________________

Niranjan Damera-Venkata, Ph. D.
The University of Texas at Austin, December 2000

Supervisor: Brian L. Evans

Digital halftoning is the process by which a grayscale or color continuous-tone image is quantized to a limited number of discrete graylevels for printing or display. In halftoning by error diffusion, the quantization error at each image pixel is diffused to the unprocessed pixels in a neighborhood around the current quantized pixel via an error filter. This process aims at shaping the quantization noise power into the high frequency regions where the human eye is least sensitive. Such noise shaping results in a high-quality halftone reproduction of the continuous-tone image.

This dissertation extends error diffusion halftoning to operate on vector valued images. Vector valued images arise naturally as color images (e.g. RGB images) or synthetically as grayscale images. Vector-valued grayscale images have been subjected to a "blocking" operation, which divides the image into blocks and stores each block as a vector. In each case, the quantization error at a given component of the vector at the current location is not only diffused to the corresponding components of the neighboring vectors but rather to *all* of their components. This requires that the error filter is no longer a conventional filter with scalar-valued coefficients but rather a "multifilter" whose coefficients are matrices.

This dissertation develops a linear matrix "gain" model for the quantizer for the analysis and design of vector error diffusion halftoning systems. Design strategies for the matrix-valued error filter are presented for both the vector color halftoning and the block error diffusion halftoning cases to achieve specific goals, such as minimizing visual quantization error in color halftoning, and producing dot-clusters with user controllable properties in block error diffusion. Efficient parallel implementations of vector error diffusion halftoning are also described.

Table of Contents

Acknowledgments **v**

Abstract **viii**

List of Tables **xii**

List of Figures **xiii**

Chapter 1. Introduction **1**

1.1 Current halftoning methods 1

1.2 Error diffusion 6

    1.2.1 Analysis of grayscale error diffusion systems 8

    1.2.2 Design of the error filter in grayscale image halftoning . . . 10

    1.2.3 Vector color error diffusion 10

    1.2.4 AM-FM halftoning 11

1.3 Contributions 14

Chapter 2. Vector Signal Processing Theory **17**

2.1 Introduction 17

2.2 Notation 18

2.3 Parallel Block Filtering 20

2.4 Efficiency Analysis 22

2.5 Extension to 2-D Vector-Valued Signals 24

2.6 Conclusion 25

Chapter 3. The Matrix Gain Model for Color Halftoning **26**

3.1 Introduction 26

3.2 Linearizing Vector Color Error Diffusion 27

3.3 Validating the Matrix Gain Model 30

3.3.1 Validation by constructing a linearly distorted original . . . 32
3.3.2 Validation by constructing an undistorted halftone 33
3.3.3 Validation of the noise response 39
3.4 Invertibility of matrices used in the model 41
3.5 Conclusion 42

Chapter 4. Optimum Error Filter Design for Color Error Diffusion Halftoning 43
4.1 Introduction 43
4.2 Linear color model for the human visual system 44
4.3 Designing the Error Filter 46
4.4 Conclusion 56

Chapter 5. Clustered Dot Error Diffusion Halftoning 59
5.1 Introduction 59
5.2 Block Error Diffusion 61
5.3 FM halftoning via block error diffusion 63
5.3.1 Error filter design 63
5.3.2 FM halftoning 65
5.4 Enhancements to FM-halftoning via block error diffusion 69
5.4.1 FM halftoning with user-defined sharpness control 69
5.4.2 Embedded multiresolution FM halftoning 72
5.5 AM-FM halftoning via block error diffusion 77
5.5.1 AM-FM halftoning with variable dot shape 77
5.5.2 AM-FM halftoning with user defined dot size 80
5.6 Efficient parallel implementation of block error diffusion 81
5.7 Conclusion 83

Chapter 6. Conclusion 85

Bibliography 89

Vita 96

List of Tables

4.1 Noise gain of the optimal filter on standard test images. 55

List of Figures

1.1 Continuous-tone image 4
1.2 Halftones generated with current halftoning methods. 5
1.3 Standard grayscale error diffusion 6
1.4 Common error filters for grayscale error diffusion halftoning . 7
1.5 Linear gain model of the quantizer 9

2.1 Parallel implementation of a multifilter. 22
2.2 Block filtering as a multifilter operation. 23

3.1 System block diagrams for vector color error diffusion 27
3.2 System block diagrams for vector color error diffusion model . 29
3.3 Validation of matrix gain model by linearly distorting the original image 31
3.4 System block diagrams for vector color error diffusion halftoning with a fixed pre-filter $\mathcal{G}$ having matrix valued coefficients. . . . 32
3.5 System block diagram for modified vector color error diffusion halftoning 34
3.6 Validation of matrix gain model by creating an undistorted halftone. 35
3.7 Predicted and actual spectra for residual noise image 40

4.1 Performance of the separable Floyd-Steinberg filter. 57
4.2 Performance of the optimal filter. 58

5.1 System block diagrams for block error diffusion. 61
5.2 A block error filter operating on pixel-blocks of 2×2 pixels. . 62
5.3 FM halftoning via pixel replication. 65
5.4 Block error diffusion with scalar error filter prototypes. 68
5.5 Block error diffused halftones with user controlled dot shapes. 69

5.6 FM halftone dot shapes. The shaded pixels indicate the pixels in the pixel-blocks that are part of the halftone dot shape. . . 70
5.7 System block diagram for modified block error diffusion halftoning 71
5.8 System block diagram for block error diffusion halftoning with a pre-filter 71
5.9 Block error diffused halftones with user controlled sharpness. . 73
5.10 Halftone dot shape embedding. 75
5.11 FM halftones embedded in halftones with a larger dot size. . . 76
5.12 Block error diffused AM-FM halftones with user defined AM modulation. 78
5.13 Block green noise error diffusion halftoning. 80
5.14 Block error diffused AM-FM halftones with “super-pixel” clustering. 81
5.15 Parallel block filtering. 84

Chapter 1

Introduction

Digital halftoning is the process of representing continuous-tone (a.k.a gray-scale/color) images with a finite number of levels, for the purpose of display on devices with finite reproduction palettes. The resulting images are called halftones. In most printers, the printer is able to either place a dot on the paper, or not place a dot. This means that the printer is a binary device capable of reproducing only two levels where the presence of a dot on the paper may be indicated by the level -1, and the absence of a dot may be indicated by the level $+1$. In other applications, such as display on monochrome or color monitors, the levels available are usually more than two, but finite. In all cases, the goal of digital halftoning is to produce, via an ingenious distribution of dots, the illusion of continuous tone.

1.1 Current halftoning methods

Current halftoning methods fall into the following six categories.

1. **Clustered dot dither:** The continuous tone image is quantized by a periodic array of thresholds. The thresholds are organized to promote the formation of dot-clusters, which are robust to imperfections of the

printed dot. Newspaper printing and many laser printers use clustered dot dither. Clustered dot dither is an example of AM (amplitude modulation halftoning) where the dot size/shape is modulated according to the underlying graylevel, but the dot centers are distributed with constant frequency due to the ordered screen replication. AM halftones are robust to printed dot imperfections due to the inability of most printers to place fine dots on the paper without overlap and dot distortion. Clustered dot halftones suffer from periodic artifacts due to quantization by a periodic threshold array.

2. **Dispersed dot dither:** The continuous tone image is quantized by a periodic array of thresholds, but with the goal of keeping the dots from clustering. This method leads to better halftones when dots are precisely rendered, e.g. on monochrome monitors. An analysis of clustered and dispersed dot dither halftoning is presented in [1]. Dispersed dot dither is an example of FM (frequency modulation) halftoning since the dots are dispersed according to the underlying graylevel. Dispersed dot dither halftones, however, suffer from periodic artifacts due to the periodic replication of an array of thresholds.

3. **Error diffusion:** This major 1976 advance in digital halftoning by Floyd and Steinberg [2] diffuses the quantization error over the neighboring continuous-tone pixels. The image would be scanned, and the unprocessed pixels in a local neighborhood, around the current pixel, would be modified. Anastassiou [3] and Bernard [4] show that error diffusion halftoning is two-dimensional delta-sigma modulation. Delta-sigma

modulation is a popular method for A/D conversion in digital audio that also employs feedback. Error diffusion feeds back a filtered version of the quantization error to the input. The design of the error filter is the key to high quality error diffusion halftoning methods. Error diffusion halftones have significantly better quality over clustered and dispersed dot dither halftones, because they are free from periodic artifacts and shape the quantization noise into the high frequencies where the human eye is least sensitive. High-frequency noise is referred to as “blue-noise”; hence, error diffusion is an example of blue noise halftoning [5]. Since the halftone dots are of single pixel size, the illusion of continuous-tone is created by varying the dot frequency with graylevel. Thus error diffusion is an example of FM (frequency modulation) halftoning.

4. **Blue-noise mask:** In the early 1990s, Mitsa and Parker [6] and Ulichney [7] produced blue-noise halftones by using a large array of thresholds (a.k.a. blue noise mask). These methods have lower complexity than error diffusion and produce FM halftones of quality close to error diffusion. This work represented a major advance in halftoning research.

5. **Direct Binary Search :** Also in the early 1990s, Analoui and Allebach [8] produced blue noise FM halftones by iteratively searching for the best binary pattern to match a given grayscale image by minimizing a distortion criterion. The distortion criterion incorporates a model of the frequency response of the human visual system as a weighting function. This method produces the halftones with the highest visual quality to date. Due to its complexity, it is impractical for use as a halftoning

method in raster image halftoning systems such as desktop printers.

6. **Green-noise halftoning:** Levien [9] clusters dots in error diffused halftones by feeding a filtered version of the output into the quantizer input in error-diffusion. This produces halftones with mid-frequency quantization noise (a.k.a green-noise) [10]. Green-noise halftones are also called AM-FM halftones since the dot frequency and dot shape vary with the underlying graylevel of the continuous-tone image. Such halftones do not suffer from periodic artifacts, and are also robust to imperfections in the rendering process.

Fig. 1.2 shows examples of halftones generated from the continuous-tone image of Fig. 1.1 using current halftoning methods. [1]

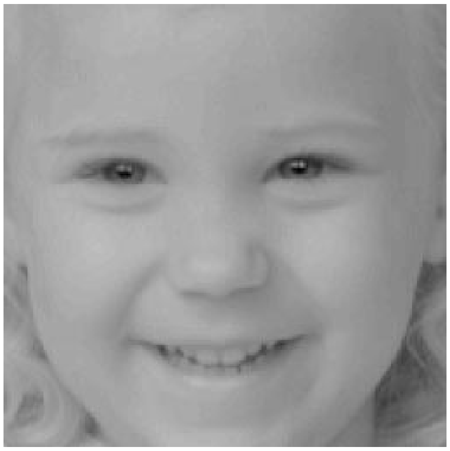
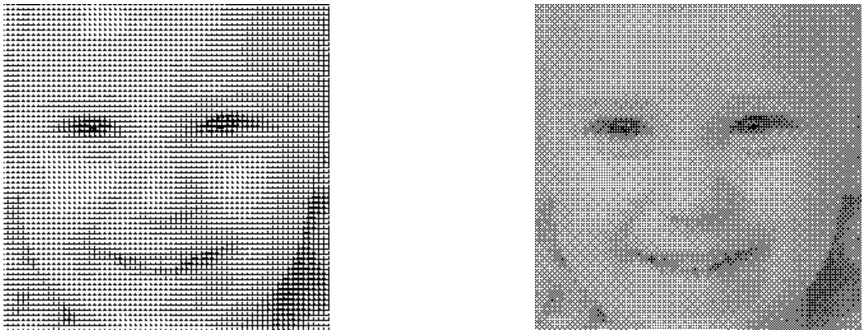


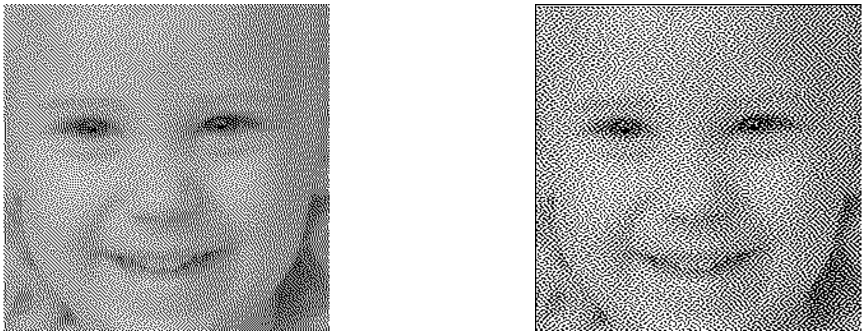
Figure 1.1: Continuous-tone image

---

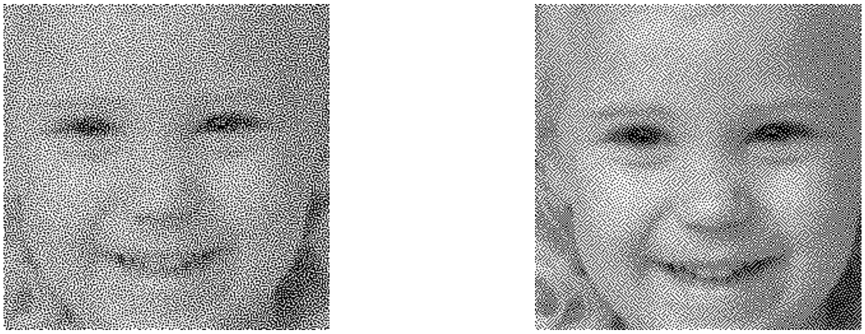
[1] The original image and the DBS halftone were provided by Qian Lin of Hewlett-Packard research labs. The blue noise mask was provided by C. B. Atkins of Hewlett-Packard research labs. Their help is gratefully acknowledged.



(a) Clustered-dot dither. (b) Dispersed-dot dither.



(c) Error diffusion. (d) Green noise.



(e) Blue noise mask. (f) Direct Binary Search.

Figure 1.2: Halftones generated with current halftoning methods.

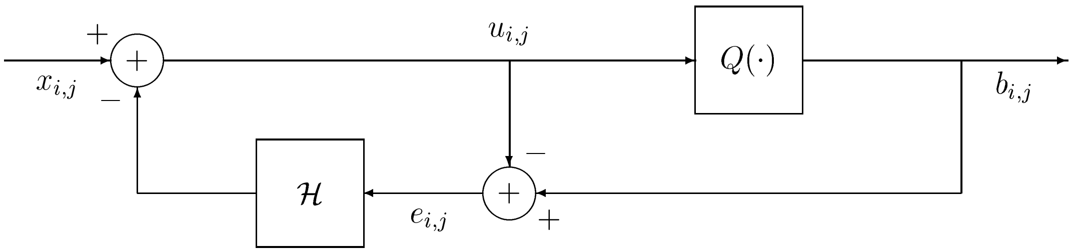


Figure 1.3: System block diagram for grayscale error diffusion halftoning where $\mathcal{H}$ represents a fixed 2-D nonseparable FIR error filter having scalar-valued coefficients.

1.2 Error diffusion

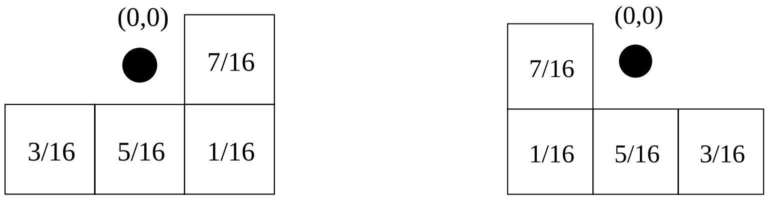
Fig. 1.3 shows conventional error diffusion. I use $x_{i,j}$ to denote the graylevel of the input image at pixel (i, j), such that $x_{i,j} \in [-1, 1]$. I use $b_{i,j}$ to represent the output halftone pixel, such that $b_{i,j} \in \{-1, 1\}$. Here, 1 is interpreted as the absence of a printer dot and −1 is interpreted as the presence of a printer dot. $Q(\cdot)$ denotes the standard quantizer function given by

$$Q(x) = \begin{cases} +1 & x \geq 0 \\ -1 & x < 0 \end{cases} \tag{1.1}$$

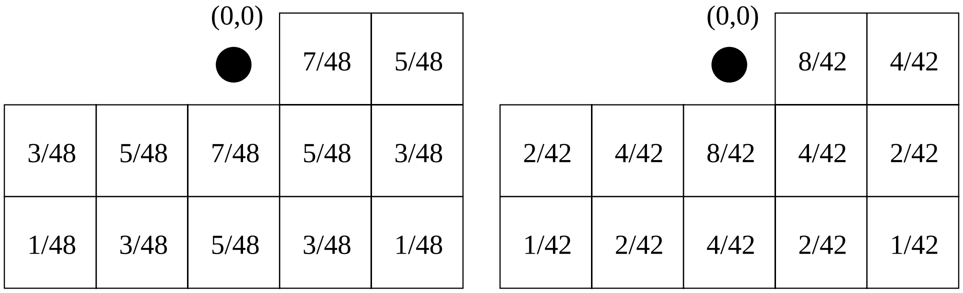
The linear map $\mathcal{H}$, a.k.a. the error filter, filters the previous quantization errors $e_{i,j} \in [-1, 1]$:

$$\mathcal{H}e_{i,j} = \sum_{(k,l)\in S} h_{k,l}\, e_{i-k,j-l} \tag{1.2}$$

$\mathcal{H}e_{i,j}$ is fed back to the input, and the set S defines the extent of the error filter coefficient mask. Note that $(0, 0) \notin S$. The mask is causal with respect to the image scan. Typical raster scan masks for the Floyd-Steinberg filter [2] and Jarvis filter [11] are shown in Figs. 1.4(a) and 1.4(c), respectively. For serpentine scans using Floyd-Steinberg filters, the mask is shown Fig. 1.4(a) for odd rows and Fig. 1.4(b) for even rows. To ensure that all of the quantization



(a) Floyd-Steinberg (raster) (b) Floyd-Steinberg (serpentine)



(c) Jarvis (raster) (d) Stucki (raster)

Figure 1.4: Common error filters for error diffusion halftoning for two different image scans. In a serpentine scan for Floyd-Steinberg error diffusion, the error filter in (a) would be used for odd rows and (b) for even rows. The black dot represents the current pixel being halftoned.

error is diffused, $\mathcal{H}$ must satisfy the constraint

$$\sum_{(k,l)\in S} h_{k,l} = 1 \tag{1.3}$$

This ensures that the error filter eliminates quantization noise at DC where the human visual system is most sensitive [12]. The quantizer input $u_{i,j}$ and output $b_{i,j}$ are given by

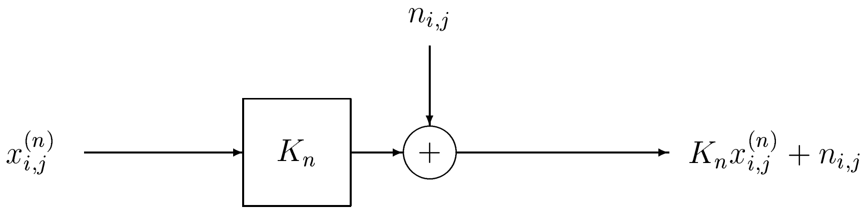
$$u_{i,j} = x_{i,j} - \mathcal{H}e_{i,j} \tag{1.4}$$

$$b_{i,j} = Q(u_{i,j}) \tag{1.5}$$

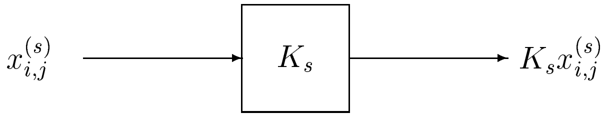
1.2.1 Analysis of grayscale error diffusion systems

Error diffused halftones suffer from several types of degradation. Knox [13] made the first major contribution to the analysis of error diffusion by mathematically showing that halftone quality can be improved by non-standard scanning techniques. For example, using a serpentine scan leads to a more symmetric error distribution. Moreover, he analyzed threshold modulation, which is the process of modulating the input of the quantizer, so as to break up objectionable artifacts in error diffusion [14]. Objectionable artifacts in error diffusion include "worm" artifacts in the very low and high graylevels and limit cycle artifacts in the mid-tones. The mid-tone artifacts are the analogs of “limit cycles” in sigma-delta modulation [15]. Fan and Eschbach [16] analyzed constant input limit cycles in error diffusion and were able to predict which patterns which were most likely to occur. They also demonstrated how the error filter coefficients may be manipulated in order to reduce limit cycle behavior. Fan [17] showed that a sufficient condition for the stability of error diffusion is satisfied if the error filter weights are positive and sum to one. Fan also made the weaker conjecture that the error diffusion halftoning system is stable if the map $1 - \mathcal{H}$ has only one zero at DC on the unit bicircle.

Knox [18] showed that error diffusion halftoning typically sharpens the original image. He noted that the image sharpening was proportional to the correlation of the input image with the error image. Recently, Kite, Evans and Bovik [19] applied the one-dimensional work of Ardalan and Paulos [20] in modeling quantizers in sigma-delta modulation to error diffusion. Kite, Evans and Bovik model the quantizer as a linear gain for the signal component of



(a) Noise path.



(b) Signal path.

Figure 1.5: Linear gain model of the quantizer. The input to the quantizer has been split into signal and noise. The paths are assumed to be independent.

the input to the quantizer and a linear gain plus additive uncorrelated white noise for the quantization noise component. This modeling is shown in Fig. 1.5. The gains are chosen to minimize the error incurred by using the model. Kite, Evans and Bovik showed that $K_n = 1$, which is independent of the error filter. They also showed that the value of K_s is proportional to the amount of image sharpening.

In Fig. 1.5, since the quantizer is modeled by a linear system, the error diffusion system becomes linear and may be analyzed using z-transforms. Using this approach, the authors of [19] derive the necessary and sufficient conditions for the elimination of image sharpening in error diffusion. They also derive two transfer functions for signal shaping and noise shaping, respectively, and show that these transfer functions accurately predict the image sharpening and noise shaping effects in grayscale error diffusion halftoning.

1.2.2 Design of the error filter in grayscale image halftoning

Floyd and Steinberg [2] designed their error filter coefficients by trial and error. However these coefficients have proved to be yield high-quality halftones, and are used even today as the coefficients of choice in implementing grayscale error diffusion. Later designs of four-tap error filters for grayscale error diffusion by Kolpatzik and Bouman [21] use an image-independent formulation that assumes the quantization error image is a white noise process. It turns out that the quantization error image is highly correlated with the input image [18, 19, 22]. Another design method by Wong and Allebach [23] uses an iterative training method which does not make use of the white noise assumption for the error image. The methods in [21, 23] attempt to minimize the weighted mean squared error between the grayscale image and the halftone image, where the weighting function is a model of the frequency response of the human visual system known as a contrast sensitivity function [12]. In this dissertation, I suggest a different approach based on our proposed halftoning model for designing the error filter coefficients for color halftoning.

1.2.3 Vector color error diffusion

Recently, Akarun, Yardimci and Cetin [24] extend grayscale error diffusion to color images, which I refer to as vector color error diffusion. They explicitly account for the correlation between color planes by using an adaptive error filter with matrix-valued coefficients. The adaptation updates all of the elements of the matrix-valued error filter coefficients by adding the previous estimate and normalizing the resulting coefficients, which is computationally intensive. Also, this approach does not incorporate any model for the human visual system.

It is thus desirable to design error filters with matrix-valued coefficients for vector color error diffusion halftoning, that are able to provide optimum visual performance. Also, a theoretical analysis of vector color error diffusion has not yet been reported in literature.

1.2.4 AM-FM halftoning

Digital halftoning may be classified into three categories — AM (amplitude modulation), FM (frequency modulation) and AM-FM hybrid halftoning. In AM halftoning, the dot size is varied depending on the graylevel value of the underlying grayscale image while the dot frequency is held constant. A typical example of this type of halftoning is clustered-dot ordered dither. FM halftones have a fixed dot size and shape, but the frequency of the dots varies with the graylevel of the underlying grayscale image. Conventional digital FM halftones have a fixed dot size of one pixel. Typical examples of FM halftones include halftones produced with dispersed-dot ordered dither and error diffusion halftoning [2]. AM-FM halftones have variable dot shape/size and variable frequency of dots depending on the graylevel value to be reproduced. Typical examples of halftones that are in this category are the green-noise halftones of Levien [9, 25], halftones generated by Velho and Gomez [26] using halftoning on space filling curves and digital halftoning with texture control by Scheermesser and Bryngdahl [27]. AM halftones suffer from systematic periodic artifacts due to the periodicity of the rendered dots. AM screens are limited by the fundamental tradeoff between spatial resolution and rendered graylevels. We require the dither array cell to be as small as possible for increased spatial resolution but as large as possible to reproduce more graylevels. Typical AM clustered-dot screens are designed to achieve a suitable compromise among the above constraints.

FM halftones typically have increased spatial resolution due to the aperiodic nature of the dots although dispersed-dot ordered dither FM screens also suffer from periodic artifacts due to the screen periodicity. Halftones produced via error diffusion [2] and stochastic screens [6, 28] are also FM halftones. Error diffusion diffuses quantization error to the neighboring pixels, which enables it to represent a wide range of graylevels, even though the dot size is only one pixel. Conventional FM halftones (single pixel halftone dots), however, suffer from physical printer imperfections such as dot gain [29]. Dot gain is caused from the increase in printed dot size from the intended dot size. This results in dot overlap, which results in a reduction in tonal range that causes loss of definition in the reproduced image [25]. Lau, Arce and Gallagher [25] further note that if the size/shape variation from printed-dot to printed-dot is small, then the effects of dot gain can be mitigated by dot gain compensation techniques. If, however, there is a large variation in the size/shape of the printed dots, clustering of the digital halftone dots adds robustness to the halftoning process, and in many cases is a necessity [30].

AM-FM hybrid halftones due not suffer from periodic artifacts since they are not realizations of periodic stochastic processes. At the same time, the dots themselves are clustered, and hence these halftones exhibit a more robust printed-dot. Lau, Arce and Gallagher [25] have extensively analyzed the spatial and spectral characteristics of these methods.

Velho and Gomez [26] produce AM-FM halftones by first dividing the image into cells along a space filling curve scan. Then, the average intensity within a cell is computed and a suitable clustered dot pattern is generated to approximate the average intensity. The difference between the intensity of the dot pattern and the average intensity within the cell is propagated to neighboring regions along the scan. The dot patterns are repositioned within the cell to introduce randomness and counteract periodic artifacts. Dot size may be controlled by controlling the number of pixels that make up a cell.

Scheermesser and Bryngdahl [27] use an optimization method based on a cost function to determine how the dot clustering should occur. The cost function represents a tradeoff between perceived image quality and the relative orientation of minority pixels. Adjustable dot clustering is produced by varying the tradeoff between the image quality and minority pixel orientation.

While the above two methods [26, 27], are significantly more computationally intensive than error diffusion, Levien [9] proposed a simple modification to error diffusion in order to promote tunable dot clustering in the reproduced halftone. He shows that by feeding a scaled, filtered version of the output back into the input to the quantizer in conventional error diffusion halftoning, halftones with clustered dots resulted. The scaling factor controls the gross size of the dot clusters, although the size of the actual clusters is dependent on the graylevel to some extent. However, one must use a serpentine scan to generate AM-FM halftones with this method due to the strong diagonal artifacts that appear when using conventional raster scanning. User control of the dot shape and amplitude modulation is not possible with this method.

1.3 Contributions

The following are contributions to the theory, algorithms, design, and implementation of vector error diffusion included in this dissertation, which are described in [31], [32] and [33].

1. I develop a theory for the analysis of vector error diffusion halftoning systems. I validate the theory by using it to predict the linear distortion and noise shaping properties of error diffusion. I generalize the linear system model of grayscale error diffusion in [19] to vector color error diffusion by replacing the "linear gain model" with a new "matrix gain model" and using properties of filters with matrix-valued coefficients. The new model includes the earlier models [19, 34] as special cases.
2. I use the new theory of vector error diffusion halftoning to find the necessary and sufficient condition to eliminate linear shift-invariant frequency distortion in vector color error diffusion halftoning via a pre-filter.
3. I use the theory along with sophisticated color appearance modeling to design fixed matrix-valued error filter coefficients for vector color error diffusion. The new design method outperforms current design methods since it not only incorporates the correlations between the image planes but also minimizes the visual color quantization error.
4. I show that a filter with matrix-valued coefficients can be implemented efficiently via a parallel polyphase structure. Polyphase forms are used

for efficient parallel implementations of filter banks in digital audio [35]. The implementation cost of the error filter may be reduced by as much as a factor of three (in the case of RGB images) because each of the three color planes being input can be buffered and filtered independently of the other color planes. Such an implementation makes vector color error diffusion attractive when using embedded processing, esp. because of the low buffer requirements and the support for raster processing.

5. I introduce a new type of error diffusion for grayscale images, by "blocking" the original image and performing vector error diffusion on the resulting vectorized image. I use this type of generalized error diffusion to produce FM halftones with user-controlled dot size and shape using block quantization and a block filter in the feedback loop. I call this modified quantization and feedback process block error diffusion. Using block error diffusion, FM halftones may be hierarchically embedded into other FM halftones seamlessly, and can be generated with user controlled sharpness. I also show how AM-FM halftones may be produced within the block error diffusion framework. Further, I show that block error diffusion may be implemented efficiently in parallel as a polyphase filterbank.

Chapter 2 introduces the mathematical notation used throughout the dissertation, reviews the basic linear theory of vector signal processing, and shows the duality between vector error diffusion and block error diffusion. Parallel implementations of vector and block error diffusion are presented within a unified framework.

Chapter 3 introduces the matrix gain model for vector color error diffusion and validates the model by predicting linear frequency distortion and noise shaping effects of vector color error diffusion. The necessary and sufficient condition to eliminate linear frequency distortion in vector color error diffusion using a pre-filter is derived.

Chapter 4 designs fixed error filters design for vector color error diffusion using the matrix gain model and a model for color appearance. The model for color appearance [36] incorporates human visual sensitivity to color patterns of different spatial frequencies. The color of a pattern is defined according to the model by the excitation of the fundamental cone photoreceptors in the visual system. Thus it is possible to obtain optimal error filters for calibrated imaging devices such as color monitors and printers.

Chapter 5 introduces a block error diffusion framework based on the vector error diffusion concept for grayscale image halftoning. Error filters to produce user-controlled dot-clustering are designed starting with a conventional scalar filter prototype. Enhancements to the basic block error diffusion framework in the form of dot cluster embedding, user controlled sharpening and user controlled AM-FM halftoning are presented.

Chapter 6 concludes the dissertation by summarizing the contributions and suggesting applications to related literature.

Chapter 2

Vector Signal Processing Theory

2.1 Introduction

This chapter introduces the mathematical notation used throughout the dissertation. It also provides the basic theory relating the processing of vector-valued signals and scalar-valued processing of spatially blocked data. Based on the theory, a parallel implementation of multifilters (filters that operate on vector-valued signals) and corresponding block filters (filters that operate on scalar-valued blocked data) is presented. For simplicity, the results of this chapter are derived for one-dimensional (1-D) signals and then extended to two-dimensional (2-D) signals.

Section 2.2 introduces the notation used throughout the dissertation. Section 2.3 shows the relationship between multifilters and block filters using one-dimensional signals. Section 2.4 shows that multifilters and block filters may take advantage of the features of conventional parallel digital signal processors. It is shown that implementation of multifilters on parallel digital signal processors speeds up the multifiltering operation by the vector dimension of the vector-valued signal being processed. Section 2.5 extends the analysis to vector-valued signals of dimension greater than two. Finally, Section 2.6

concludes the chapter by summarizing the essential concepts discussed in the chapter.

2.2 Notation

In this chapter, $\mathbf{x}(m)$ represents a 1-D input N-vector valued signal to be filtered. $\mathbf{X}(z)$ represents the z-transform of the vector-valued signal sequence

$$\mathbf{X}(z) = \sum_{m} \mathbf{x}(m) z^{-m} \tag{2.1}$$

The multifilter (which is assumed to be causal) will be denoted by the $N \times N$ matrix-valued sequence $\tilde{\mathbf{h}}(\cdot)$. $\tilde{\mathbf{H}}(\cdot)$ represents the z-transform of the matrix-valued multifilter sequence defined by

$$\tilde{\mathbf{H}}(z) = \sum_{m} \tilde{\mathbf{h}}(m) z^{-m} \tag{2.2}$$

A multifilter with M $N \times N$ matrix coefficients can be expressed compactly as the $N \times NM$ matrix

$$\tilde{\mathbf{\Gamma}} = \left[\tilde{\mathbf{h}}(0) \mid \tilde{\mathbf{h}}(1) \mid \ldots \mid \tilde{\mathbf{h}}(M-1)\right] \tag{2.3}$$

where $\tilde{\mathbf{h}}(0) \cdots \tilde{\mathbf{h}}(M-1)$ are the multifilter coefficients. The filtering operation of a multifilter is defined by matrix-vector convolution is given by

$$\mathbf{y}(m) = \sum_{k=0}^{M-1} \tilde{\mathbf{h}}(k) \mathbf{x}(m-k) \tag{2.4}$$

where $\mathbf{y}(\cdot)$ represents the N-vector valued output sequence. In the z-domain, the matrix-vector convolution becomes a linear transformation by an $N \times N$ transformation matrix given by

$$\mathbf{Y}(z) = \tilde{\mathbf{H}}(z) \mathbf{x}(z) \tag{2.5}$$

For a scalar signal $x(m)$, I denote the z-transform by $X(z)$.

Let $\mathbf{x}(m_1, m_2)$ represent an input N-vector valued image to be filtered. $\mathbf{X}(z_1, z_2)$ represents the z-transform of the vector-valued image sequence

$$\mathbf{X}(z_1, z_2) = \sum_{m_1, m_2} \mathbf{x}(m_1, m_2) z^{-m_1} z^{-m_2} \tag{2.6}$$

A 2-D multifilter may filter a vector-valued image (e.g. a color image). The multifilter will be denoted by the $N \times N$ matrix-valued sequence $\tilde{\mathbf{h}}(\cdot)$. $\tilde{\mathbf{H}}(\cdot)$ represents the z-transform of the matrix-valued multifilter sequence defined by

$$\tilde{\mathbf{H}}(z_1, z_2) = \sum_{m_1, m_2} \tilde{\mathbf{h}}(m_1, m_2) z^{-m_1} z^{-m_2} \tag{2.7}$$

A multifilter with an $M \times M$ support with $N \times N$ matrix-valued coefficients can be expressed compactly as the $N \times NM^2$ matrix

$$\tilde{\mathbf{\Gamma}} = \left[\tilde{\mathbf{h}}(0) \mid \tilde{\mathbf{h}}(1) \mid \ldots \mid \tilde{\mathbf{h}}(M^2 - 1)\right] \tag{2.8}$$

where $\tilde{\mathbf{h}}(0) \cdots \tilde{\mathbf{h}}(M^2 - 1)$ are the coefficients of the matrix valued filter ordered by rows. The filtering operation of a 2-D multifilter is defined by matrix-vector convolution is given by

$$\mathbf{y}(m_1, m_2) = \sum_{k_1, k_2} \tilde{\mathbf{h}}(k_1, k_2) \mathbf{x}(m_1 - k_1, m_2 - k_2) \tag{2.9}$$

where $\mathbf{y}(\cdot)$ represents the N-vector valued output sequence. In the z-domain the matrix-vector convolution becomes a linear transformation by an $N \times N$ transformation matrix given by

$$\mathbf{Y}(z_1, z_2) = \tilde{\mathbf{H}}(z_1, z_2) \mathbf{X}(z_1, z_2) \tag{2.10}$$

I use an $\mathbf{m}$ to denote a 2-D index (m_1, m_2).

2.3 Parallel Block Filtering

In the multifilter operation in (2.4), the output vector at each vector sample location m is computed by taking N linear combinations of *all* of the signal vector components in the set

$$\mathcal{S}_m = \{x_i(m-k), i = 0 \cdots N, k = 0 \cdots M\} \tag{2.11}$$

The overall filtering may be expressed in terms of the matrix $\tilde{\mathbf{\Gamma}}$ as

$$\mathbf{y}(m) = \tilde{\mathbf{\Gamma}}\mathbf{x}'(m) \tag{2.12}$$

where

$$\mathbf{x}'(m) = [x_0(m),\ x_1(m) \cdots,\ x_N(m-M+1)]^T \tag{2.13}$$

One can reorder the vector-valued components of $\mathbf{x}'(m)$ in (2.12) to create a scalar-valued sequence $x''(\cdot)$ by

$$\begin{aligned} x''(0) &= x_0(0) \\ x''(1) &= x_1(0) \\ &\vdots \\ x''(r) &= x_{r\ mod\ N}(\left\lfloor \tfrac{r}{N} \right\rfloor) \end{aligned}$$

In many situations, the vector-valued signals are produced in “blocks” of an input scalar-valued signal via the process described above. The samples of the nth component of the output signal vector sequence $\mathbf{y}(\cdot)$ may be recovered exactly by filtering the scalar sequence $x''(\cdot)$ with the filter $h^{(n)}(\cdot)$, with the additional constraint that only every Nth output be retained. The values of $h^{(n)}(\cdot)$ are formed by reversing the nth row of the matrix $\tilde{\mathbf{\Gamma}}$. These operations may be represented by

$$y_n(m) = \sum_{k=0}^{MN-1} h^{(n)}(k)x''(Nm+N-1-k) \tag{2.14}$$

In the frequency domain, this operation may be expressed as

$$Y_n(z) = (\downarrow N)\left(H^{(n)}(z)X''(z)z^{(N-1)}\right) \tag{2.15}$$

Filtering followed by subsampling may be implemented efficiently using a polyphase decomposition [35, 37]. Consider each of the N components of the input signal vector. The type-II polyphase decomposition of the filter is

$$H^{(n)}(z) = \sum_{i=0}^{N-1} H_i^{(n)}(z^N)z^{-(N-1-i)} \tag{2.16}$$

By substituting (2.16) into (2.15), and using the Noble Identity [35, 37] to commute the filtering with the downsampling,

$$Y_n(z) = \sum_{i=0}^{N-1} H_i^{(n)}(z)\,(\downarrow N)\,X''(z)z^{-(N-1-i)}z^{(N-1)} \tag{2.17}$$

For $i = 0, 1, \cdots, N-1$

$$X_i(z) = (\downarrow N)\,X''(z)z^i \tag{2.18}$$

where $X_i(z)$ denotes the z-transform of the ith component of the input signal vector.

Thus, each component of the output signal vector sequence $y_n(\cdot)$ can be decomposed in terms of conventional scalar filtering operations on each of the components of the input signal vector sequence:

$$Y_n(z) = \sum_{i=0}^{N-1} H_i^{(n)}(z)X_i(z) \tag{2.19}$$

The filtering can be expressed in the time domain as

$$y_n(m) = \sum_{i=0}^{N-1}\sum_{k=0}^{M-1} h_i^{(n)}(k)x_i(m-k) \tag{2.20}$$

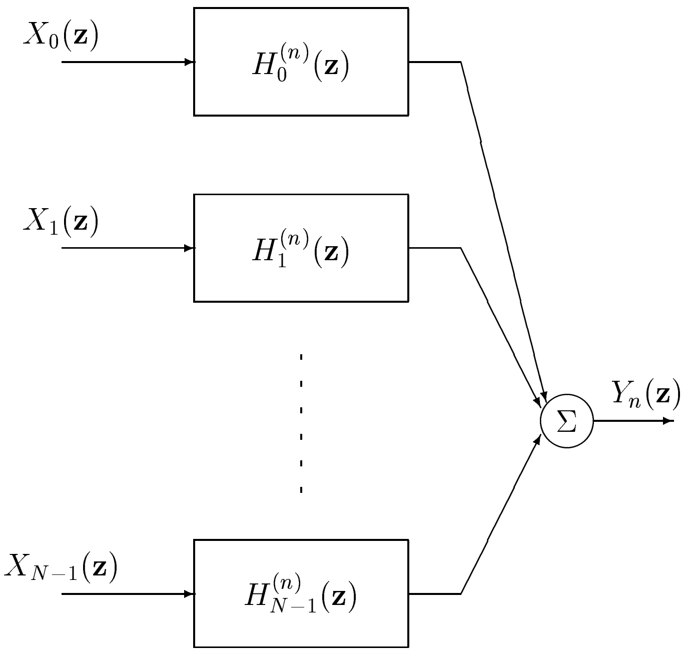


Figure 2.1: A parallel implementation of a multifilter which computes the nth component of the output signal N-vector sequence. N^2 filters are required for the computation of all of the components of the output vector sequence.

This filtering is illustrated in Fig. 2.1 where the filters $h_i^{(n)}$, $n = 0, 1, \cdots, N-1$ and $i = 0, 1, \cdots, N-1$ are the polyphase components of the nth row of the matrix $\tilde{\mathbf{\Gamma}}$ corresponding to the elements that multiply the ith component of the input signal vector in (2.4).

2.4 Efficiency Analysis

Since $\tilde{\mathbf{\Gamma}}$ is fixed, the polyphase components of its rows may be precomputed. The result is a set of conventional filters with scalar coefficients, which enables the components of the input signal vector sequence to be buffered and filtered independently of the other dimensions. Since the size of a row of $\tilde{\mathbf{\Gamma}}$ is N, the throughput is increased by a factor of N when the filtering is im-

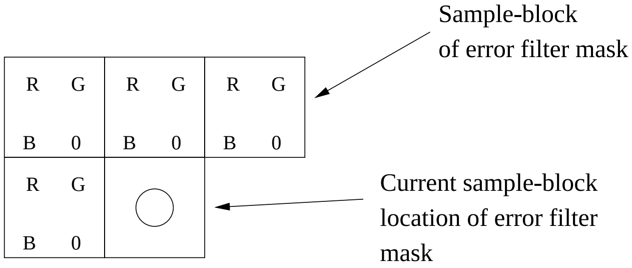


Figure 2.2: Example multifilter operating on an RGB image expressed as sample blocks. Three linear combinations of all samples in the mask are computed at the current location (denoted by the circle) to compute the red, green and blue samples of the multifilter output. The filter moves in a raster scan of the blocks.

plemented in parallel. Because each filter in the parallel filterbank of Fig. 2.1 has M scalar coefficients, the rate at which these filters operate to deliver the same throughput is divided by a factor of N over the single processor implementation of (2.4). Equation (2.4) performs MN^2 multiply accumulates in a sequential fashion to compute all of the components of the output signal vector. When filtering an RGB image [24], the parallel implementation would speed up the processing by a factor of three, although the total number of multiply-accumulates remains the same. We may utilize three low-bandwidth, low-cost processors instead of one high-bandwidth processor to obtain the same performance at a lower cost [35], or exploit the instruction-level parallelism of a SIMD or VLIW processor such as the Intel Pentium MMX or Texas Instruments TMS320C6000, respectively. The efficient implementation computes the output N times as fast if all of the operations are performed at the same speed. The parallel implementation of Fig. 2.1 does not require shared circular buffers. Each component of the input vector sequence is put into a separate circular buffer on one of the N parallel processors. This allows for fast, low-overhead loop code.

2.5 Extension to 2-D Vector-Valued Signals

The parallel implementation may easily be extended to the filtering of multidimensional signals such as images. The multifilter operation on a 2-D N-vector valued signal $\mathbf{x}(\mathbf{m})$ is given by

$$\mathbf{y}(\mathbf{m}) = \sum_{\mathbf{k}\in\mathcal{R}} \tilde{\mathbf{h}}(\mathbf{k})\mathbf{x}(\mathbf{m}-\mathbf{k}) \tag{2.21}$$

where $\mathbf{m}$ and $\mathbf{k}$ are 2-D vector indices and $\mathcal{R}$ is the region of support of the multifilter.

The multifilter is equivalent to a block filter consisting of N samples per block. The filter mask is specified in blocks, and the filter moves from block to block instead of from sample to sample. At each block, the filtering given by (2.21) computes N linear combinations of all of the samples within the block mask. This filtering is illustrated in Fig. 2.2 for the case $N = 3$ (RGB image) and the four matrix-coefficient multifilter ("error-filter") mask in [24]. Each component of the output N-vector is then recovered by filtering and downsampling the vector sequence rearranged into blocks (see Fig. 2.2) by using the scalar 2-D filter corresponding to the rows of the matrix $\tilde{\mathbf{\Gamma}}$ by

$$\tilde{\mathbf{\Gamma}} = \left[\tilde{\mathbf{h}}_0 \mid \tilde{\mathbf{h}}_1 \mid \ldots \mid \tilde{\mathbf{h}}_{|\mathcal{R}|-1}\right] \tag{2.22}$$

where $\mid \mathcal{R} \mid$ is the cardinality of $\mathcal{R}$ and the $\tilde{\mathbf{h}}_i$ matrices represent the coefficients of the multifilter within the filter's support given in row order. The downsampling matrix for the equivalent blocked image is given by $\Lambda = N\tilde{\mathbf{I}}_{N\times N}$ (the blocks may need to be zero padded before using this downsampling matrix).

With this interpretation of block filtering in two dimensions and with $(\downarrow \Lambda)$ replacing $(\downarrow N)$, essentially the same arguments as in Section 2.3 can be used to obtain the parallel form for the multifilter given by

$$y_n(\mathbf{m}) = \sum_{i=0}^{N-1} \sum_{\mathbf{k}\in\mathcal{R}} h_i^{(n)}(\mathbf{k}) x_i(\mathbf{m}-\mathbf{k}) \tag{2.23}$$

The speedup due to parallel implementation is N which is identical to the one-dimensional case.

2.6 Conclusion

This chapter introduced the basic notation that will be used throughout the dissertation. The general linear system theory of filtering vector-valued signals with matrix-valued error filters was reviewed in the time and frequency domains. The close relationship between the processing of vector-valued sequences using a multifilter and the processing of scalar-valued blocked signals with a block filter was established. It was shown that multifilters and associated block filters can be implemented efficiently using parallel digital signal processors. In the next chapter, I will apply some of the theory discussed in this chapter to model vector color error diffusion in the frequency domain. The error filter in this case is a multifilter operating on a vector-valued image of quantization errors. The objective is to predict the signal distortion and noise shaping behavior of vector color error diffusion based on the modeling.

Chapter 3

The Matrix Gain Model for Color Halftoning

3.1 Introduction

This chapter generalizes the linear system model of grayscale error diffusion in [19] to vector color error diffusion by replacing the "linear gain model" with a new "matrix gain model" and using properties of filters with matrix-valued coefficients discussed in Chapter 2. The new model includes the earlier model [19, 34] as a special case. The new model describes vector color diffusion in the frequency domain, and predicts noise shaping and linear frequency distortion produced by error diffusion halftoning. For vector color halftoning, I also derive the necessary and sufficient condition for linear distortion elimination via prefiltering.

Section 3.2 describes how vector error diffusion may be linearized via the proposed matrix gain model. Signal and noise shaping transfer functions are derived based on an analysis of the linearized system. Section 3.3 validates the predictions of the matrix gain model by halftoning test images. Section 3.3.2 shows that signal frequency distortion may be eliminated via pre-filtering. Further, I show that using such a pre-filter is equivalent to a simple modification of error diffusion. Section 3.4 comments on the invertibility of certain

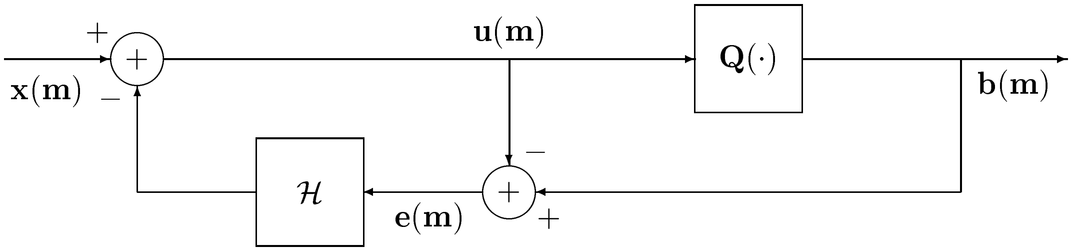


Figure 3.1: System block diagrams for vector color error diffusion halftoning where $\mathcal{H}$ represents a fixed 2-D nonseparable FIR error filter with matrix-valued coefficients.

matrices used in the matrix gain model. Finally Section 3.5 summarizes the contributions of this chapter.

3.2 Linearizing Vector Color Error Diffusion

Fig. 3.1 shows a block diagram of vector color error diffusion halftoning. When halftoning red-green-blue (RGB) images, the quantizer output for each color channel at any pixel is exactly one element from the discrete set $\mathcal{O} = \{-1, 1\}$. Here, -1 represents a red, green or blue dot, depending on the color channel, and 1 represents the absence of a dot for that color channel. Since the outputs of the quantizer are constrained to lie on the corners of a cube in three-space, nothing is gained by using a vector quantizer (in the mean squared error sense). Therefore, we quantize each color channel using a scalar quantizer. The quantizer $\mathbf{Q}(\cdot)$ is defined by

$$\mathbf{Q}(\mathbf{u}) = \begin{pmatrix} Q(u_1) \\ Q(u_2) \\ Q(u_3) \end{pmatrix} \tag{3.1}$$

$$Q(u_i) = \begin{cases} 1 & u_i \geq 0 \\ -1 & u_i < 0 \end{cases} \tag{3.2}$$

where $\mathbf{u}$ is a column vector and u_i, $i = 1, 2, 3$, represents the red, green and blue components of the color vector to be quantized.

The filter in the feedback loop has matrix-valued coefficients. The filter operates on the quantization error sequence $\mathbf{e}(\mathbf{m})$ to produce the feedback signal sequence according to

$$\mathcal{H}\mathbf{e}(\mathbf{m}) = \sum_{\mathbf{k} \in \mathcal{S}} \tilde{\mathbf{h}}(\mathbf{k})\mathbf{e}(\mathbf{m} - \mathbf{k}) \tag{3.3}$$

where $\mathbf{m}$ and $\mathbf{k}$ are two-dimensional vectors, $\tilde{\mathbf{h}}(\cdot)$ is a 3×3 matrix-valued sequence, and $\mathcal{S}$ is the filter support. This dissertation assumes that the four-tap filter support defined by (horizontal, vertical) offsets to the current pixel being processed is $\mathcal{S} = \{(0, 1), (1, 0), (1, 1), (1, -1)\}$.

I model the quantizer by a constant linear transformation denoted by a matrix $\tilde{\mathbf{K}}$ plus spatially-varying additive noise $\mathbf{n}(\mathbf{m})$, as shown in Fig. 3.2. This is a generalization of modeling the quantizers in sigma-delta modulators [20] and grayscale error diffusion [19, 34]. Correlation among the color planes is represented by the off-diagonal terms in the matrix $\tilde{\mathbf{K}}$. I choose the matrix $\tilde{\mathbf{K}}$ to minimize the error in approximating the quantizer with a linear transformation, in the linear minimum mean squared error (LMMSE) sense,

$$\tilde{\mathbf{K}} = \arg\min_{\tilde{\mathbf{A}}} E[\| \mathbf{b}(\mathbf{m}) - \tilde{\mathbf{A}}\ \mathbf{u}(\mathbf{m}) \|^2] \tag{3.4}$$

where $\mathbf{b}(\cdot)$ represents the quantizer output process, and $\mathbf{u}(\cdot)$ represents the quantizer input process. The solution to (3.4) when $\mathbf{b}(\cdot)$ and $\mathbf{u}(\cdot)$ are wide sense stationary processes is [38]

$$\tilde{\mathbf{K}} = \tilde{\mathbf{C}}_{\mathbf{bu}}\tilde{\mathbf{C}}_{\mathbf{uu}}^{-1} \tag{3.5}$$

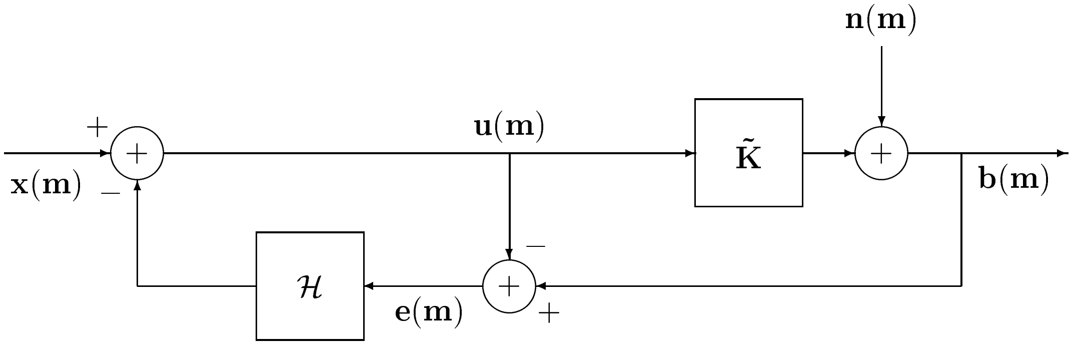


Figure 3.2: System block diagrams for vector color error diffusion model, where $\tilde{\mathbf{K}}$ represents a linear transformation and $\mathbf{n}(\mathbf{m})$ is a noise process uncorrelated with $\mathbf{u}(\mathbf{m})$.

where $\tilde{\mathbf{C}}_{\mathbf{bu}}$ and $\tilde{\mathbf{C}}_{\mathbf{uu}}$ are covariance matrices. As a direct consequence of this modeling [38], the noise process $\mathbf{n}(\cdot)$ due to the approximation error is uncorrelated with the input to the quantizer $\mathbf{u}(\cdot)$. I will analyze error diffusion, by assuming a matrix gain of $\tilde{\mathbf{K}}$ for the signal path and a matrix gain of $\tilde{\mathbf{I}}$ (identity matrix) for the noise path. This corresponds to using the estimator to estimate signal components in the output of the quantizer from signal components at its input, and assuming an uncorrelated noise injection to model the noise. In this way, one may treat the signal shaping and noise shaping independently. This is similar to the analysis for grayscale error diffusion in [19, 34].

Analyzing the linearized vector color error diffusion model of Fig. 3.2 in the frequency domain using z-transforms yields

$$Z\{\mathcal{H}\mathbf{e}(\mathbf{m})\} = \tilde{\mathbf{H}}(\mathbf{z})\mathbf{E}(\mathbf{z}) \tag{3.6}$$

By analyzing the signal path and ignoring the noise path by setting $\mathbf{n}(\mathbf{m}) = \mathbf{0}$

$$\mathbf{X}(\mathbf{z}) = \mathbf{U}(\mathbf{z}) + \tilde{\mathbf{H}}(\mathbf{z})\mathbf{E}(\mathbf{z}) \tag{3.7}$$

$$\mathbf{E}(\mathbf{z}) = (\tilde{\mathbf{K}} - \tilde{\mathbf{I}})\mathbf{U}(\mathbf{z}) \tag{3.8}$$

$$\mathbf{B}_s(\mathbf{z}) = \tilde{\mathbf{K}}\mathbf{U}(\mathbf{z}) \tag{3.9}$$

By manipulating (3.7), (3.8), and (3.9), the response to the signal component becomes

$$\mathbf{B}_s(\mathbf{z}) = \tilde{\mathbf{K}}[\tilde{\mathbf{I}} + \tilde{\mathbf{H}}(\mathbf{z})(\tilde{\mathbf{K}} - \tilde{\mathbf{I}})]^{-1}\mathbf{X}(\mathbf{z}) \tag{3.10}$$

By considering the contribution of the noise component $\mathbf{B}(\mathbf{z})$ to the output $\mathbf{B}_n(\mathbf{z})$,

$$\mathbf{B}_n(\mathbf{z}) = \mathbf{N}(\mathbf{z}) + \mathbf{U}(\mathbf{z}) \tag{3.11}$$

$$\mathbf{U}(\mathbf{z}) = -\tilde{\mathbf{H}}(\mathbf{z})\mathbf{E}(\mathbf{z}) \tag{3.12}$$

$$\mathbf{E}(\mathbf{z}) = \mathbf{N}(\mathbf{z}) \tag{3.13}$$

By rearranging (3.11), (3.12) and (3.13),

$$\mathbf{B}_n(\mathbf{z}) = [\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z})]\mathbf{N}(\mathbf{z}) \tag{3.14}$$

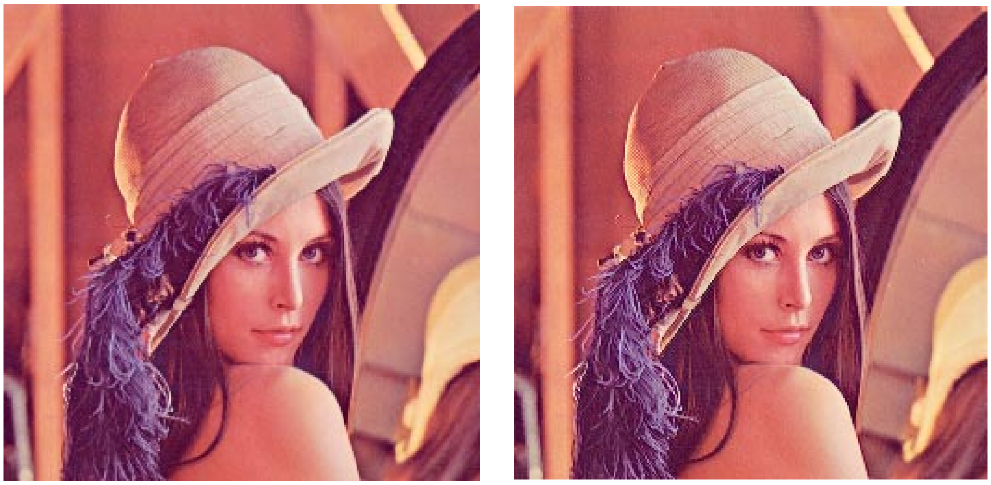
The overall system response is given by

$$\mathbf{B}(\mathbf{z}) = \mathbf{B}_s(\mathbf{z}) + \mathbf{B}_n(\mathbf{z}) \tag{3.15}$$

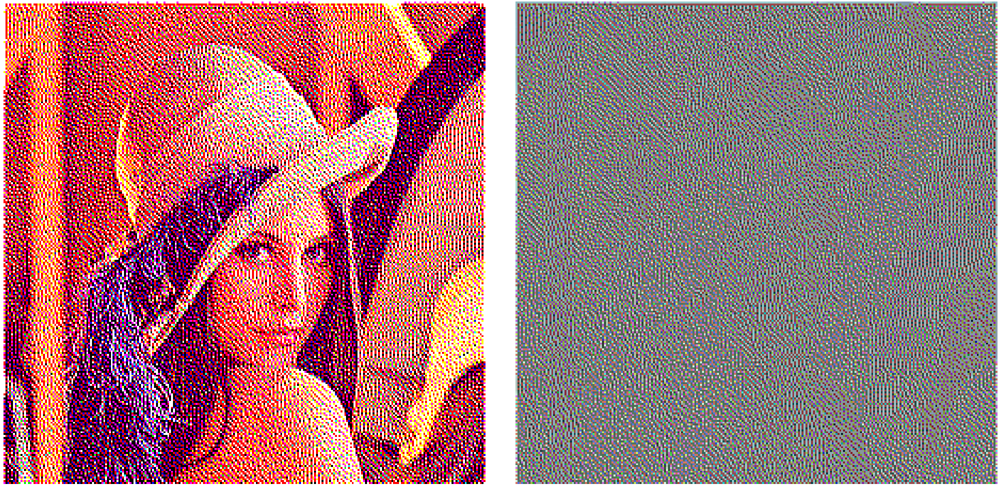
Equations (3.10) and (3.14) reduce to the analogous ones for grayscale error diffusion [19], in which the error filter coefficients and signal gain are scalar valued. The next section validates the analysis given in this section, and shows that it accurately models the linear distortion and noise shaping of vector color error diffusion.

3.3 Validating the Matrix Gain Model

This section validates the matrix gain model by using it to predict the linear distortion and noise shaping effects of vector color error diffusion. Section



(a) 256 × 256 *lenna* (b) *lena* generated using model $\tilde{\mathbf{K}}$



(c) Halftoned *lenna* (d) Residual noise image

Figure 3.3: Validation of matrix gain model by linearly distorting the original image. Here, the residual image has been scaled using a full-scale contrast stretch for display purposes.

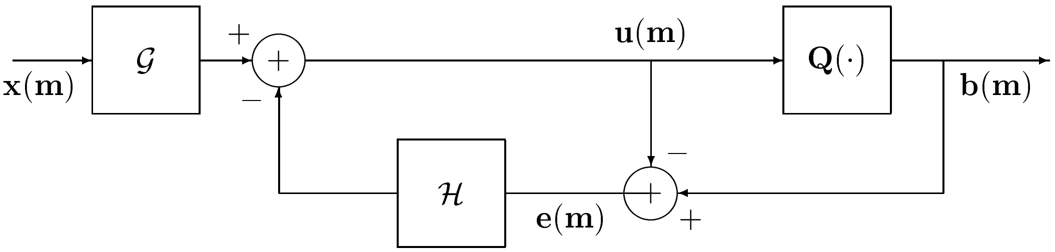


Figure 3.4: System block diagrams for vector color error diffusion halftoning with a fixed pre-filter $\mathcal{G}$ having matrix valued coefficients.

3.3.1 shows that the signal path distortion given by (3.10) accurately models the linear distortion to which the original color image is subjected in vector color error diffusion. Section 3.3.2 shows that by adding a specified linear transformation of the input image to the quantizer input, the linear distortion may be eliminated. Thus, the modeling predicts that a flat frequency response can be achieved. This will be validated through simulation.

Section 3.3.3 validates that the model accurately predicts the noise shaping behavior of vector color error diffusion. In the validation process, I use a fixed matrix-valued error filter whose coefficients were obtained by terminating the adaptive algorithm of [24] after a fixed number of iterations. The results hold for an arbitrary fixed set of matrix valued filter coefficients, and hence, there is no loss of generality.

3.3.1 Validation by constructing a linearly distorted original

I linearly distort the original image without introducing quantization noise by processing the original image of Fig. 3.3(a) by using (3.10). This is equivalent to processing the original image according to Fig. 3.2, with the additive noise ignored. Fig. 3.3(b) shows the resulting image. Fig. 3.3(c) shows the result of halftoning with the fixed error filter. Figs. 3.3(b) and 3.3(c) have comparable linear distortion. To see this, I simply form the residual image by subtracting Fig. 3.3(b) from Fig. 3.3(c). The result is shown in Fig. 3.3(d). The residual in Fig. 3.3(d) is uncorrelated with the original and represents quantization noise. This is consistent with the modeling of Section 3.2. To quantify the degree of correlation of the residual image with the original image, I introduce a correlation matrix defined by

$$\tilde{\mathbf{C}}_{\mathbf{rx}} = \begin{pmatrix} \rho_{r_{red}x_{red}} & \rho_{r_{red}x_{green}} & \rho_{r_{red}x_{blue}} \\ \rho_{r_{green}x_{red}} & \rho_{r_{green}x_{green}} & \rho_{r_{green}x_{blue}} \\ \rho_{r_{blue}x_{red}} & \rho_{r_{blue}x_{green}} & \rho_{r_{blue}x_{blue}} \end{pmatrix} \tag{3.16}$$

where $\rho_{r_i x_j}$ represents the correlation coefficient [38] between the color plane i in the residual and the color plane j in the original image. The correlation matrix for the residual shown in Fig. 3.3(d), with respect to the original image shown in Fig. 3.3(a), is

$$\tilde{\mathbf{C}}_{\mathbf{rx}} = \begin{pmatrix} 0.0067 & 0.0007 & 0.0051 \\ 0.0065 & 0.0039 & 0.0049 \\ 0.0082 & 0.0040 & 0.0062 \end{pmatrix}$$

3.3.2 Validation by constructing an undistorted halftone

The model predicts that the linear distortion suffered by the color input image is given by (3.10). This means that if one prefilters the input color image by using the filter

$$\tilde{\mathbf{G}}(\mathbf{z}) = [\tilde{\mathbf{I}} + \tilde{\mathbf{H}}(\mathbf{z})(\tilde{\mathbf{K}} - \tilde{\mathbf{I}})]\tilde{\mathbf{K}}^{-1} \tag{3.17}$$

then the resulting halftone should exhibit a flat low-frequency response with respect to the original color image. Fig. 3.4 shows error diffusion modified to include the prefilter. I now prove the following proposition.

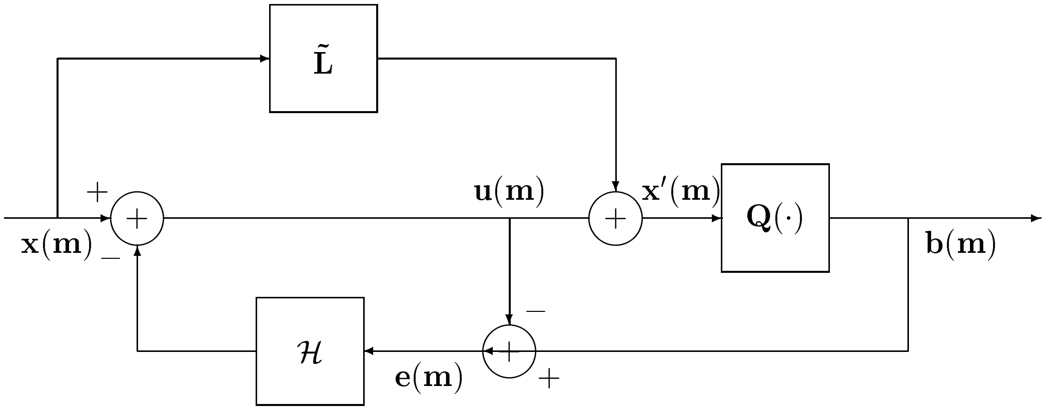


Figure 3.5: System block diagrams for modified vector color error diffusion halftoning. $\tilde{\mathbf{L}}$ represents a constant linear transformation.

Proposition 1: *Fig. 3.4 is exactly equivalent to Fig. 3.5 when* $\tilde{\mathbf{L}} = \tilde{\mathbf{K}}^{-1} - \tilde{\mathbf{I}}$, *whenever* $[\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z})]$ *is invertible.*

Proof: By analyzing Fig. 3.4, the input to the quantizer $\mathbf{u(m)}$ in the z-domain is

$$\mathbf{U(z)} = \tilde{\mathbf{G}}(\mathbf{z})\mathbf{X(z)} - \tilde{\mathbf{H}}(\mathbf{z})\mathbf{E(z)} \tag{3.18}$$

$$\mathbf{E(z)} = \mathbf{B(z)} - \mathbf{U(z)} \tag{3.19}$$

From (3.18) and (3.19),

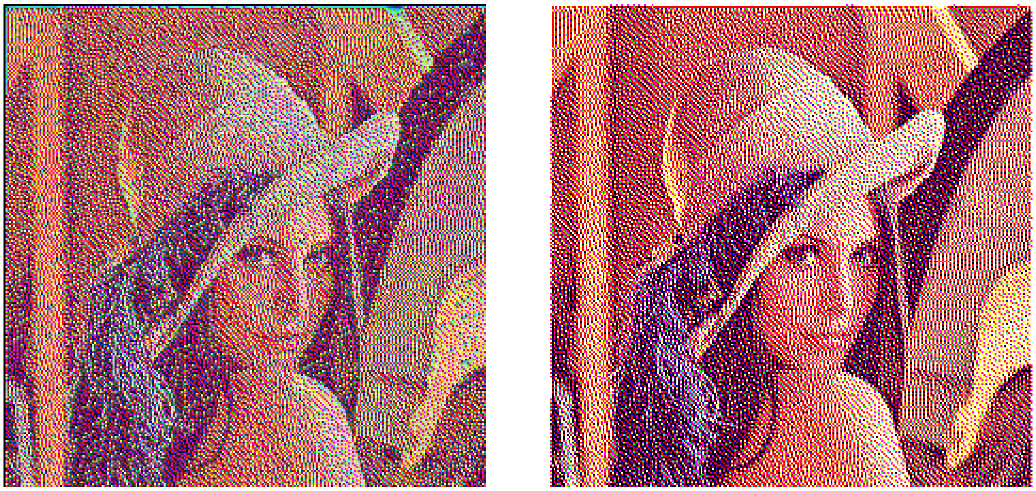
$$\mathbf{E(z)} = [\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z})]^{-1}[\mathbf{B(z)} - \tilde{\mathbf{G}}(\mathbf{z})\mathbf{X(z)}] \tag{3.20}$$

Substituting for $\mathbf{E(z)}$ given by (3.20) in (3.18) yields

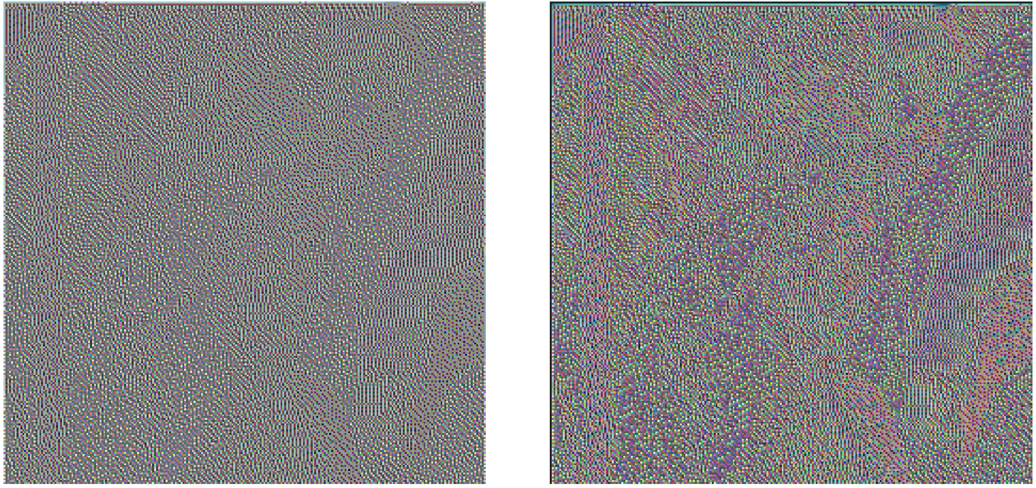
$$\mathbf{U(z)} = [\tilde{\mathbf{I}} + \tilde{\mathbf{H}}(\mathbf{z})(\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z}))^{-1}]\tilde{\mathbf{G}}(\mathbf{z})\mathbf{X(z)} - \tilde{\mathbf{H}}(\mathbf{z})(\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z}))^{-1}\mathbf{B(z)} \tag{3.21}$$

Now, by analyzing Fig. 3.5,

$$\mathbf{U(z)} = \mathbf{X(z)} - \tilde{\mathbf{H}}(\mathbf{z})\mathbf{E(z)} \tag{3.22}$$



(a) Residual image when $\tilde{\mathbf{L}} = \tilde{\mathbf{0}}$ (b) Halftone generated using optimal $\tilde{\mathbf{L}}$



(c) Residual image using optimal $\tilde{\mathbf{L}}$ (d) Input to error filter using optimal $\tilde{\mathbf{L}}$

Figure 3.6: Validation of matrix gain model by creating an undistorted halftone. Here, the residual image and the input to the error filter have been scaled using a full-scale contrast stretch for display purposes.

$$\mathbf{E}(\mathbf{z}) = \mathbf{B}(\mathbf{z}) - \mathbf{U}(\mathbf{z}) \tag{3.23}$$

From (3.22) and (3.23),

$$\mathbf{E}(\mathbf{z}) = [\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z})]^{-1}[\mathbf{B}(\mathbf{z}) - \mathbf{X}(\mathbf{z})] \tag{3.24}$$

Also, since

$$\mathbf{X}'(\mathbf{z}) = [\tilde{\mathbf{I}} + \tilde{\mathbf{L}}]\mathbf{X}(\mathbf{z}) - \tilde{\mathbf{H}}(\mathbf{z})\mathbf{E}(\mathbf{z}) \tag{3.25}$$

we substitute (3.24) into (3.25)

$$\mathbf{X}'(\mathbf{z}) = [(\tilde{\mathbf{I}} + \tilde{\mathbf{L}}) + \tilde{\mathbf{H}}(\mathbf{z})[\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z})]^{-1}]\mathbf{X}(\mathbf{z}) - \tilde{\mathbf{H}}(\mathbf{z})(\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z}))^{-1}\mathbf{B}(\mathbf{z}) \tag{3.26}$$

Comparing (3.21) and (3.26), it follows that Fig. 3.4 and Fig. 3.5 are equivalent in the sense that they have the same quantizer input and hence output if

$$[(\tilde{\mathbf{I}} + \tilde{\mathbf{L}}) + \tilde{\mathbf{H}}(\mathbf{z})[\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z})]^{-1}] = [\tilde{\mathbf{I}} + \tilde{\mathbf{H}}(\mathbf{z})(\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z}))^{-1}]\tilde{\mathbf{G}}(\mathbf{z}) \tag{3.27}$$

By using

$$\tilde{\mathbf{P}}(\mathbf{z}) = \tilde{\mathbf{I}} + \tilde{\mathbf{H}}(\mathbf{z})(\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z}))^{-1} \tag{3.28}$$

equation (3.27) becomes

$$\tilde{\mathbf{L}} = \tilde{\mathbf{P}}(\mathbf{z})[\tilde{\mathbf{G}}(\mathbf{z}) - \tilde{\mathbf{I}}] \tag{3.29}$$

Substituting for $\tilde{\mathbf{G}}(\mathbf{z})$ given by (3.17),

$$\begin{aligned}
\tilde{\mathbf{L}} &= \tilde{\mathbf{P}}(\mathbf{z})[[\tilde{\mathbf{I}} + \tilde{\mathbf{H}}(\mathbf{z})(\tilde{\mathbf{K}} - \tilde{\mathbf{I}})]\tilde{\mathbf{K}}^{-1} - \tilde{\mathbf{I}}] \\
&= \tilde{\mathbf{P}}(\mathbf{z})[\tilde{\mathbf{K}}^{-1} + \tilde{\mathbf{H}}(\mathbf{z}) - \tilde{\mathbf{H}}(\mathbf{z})\tilde{\mathbf{K}}^{-1} - \tilde{\mathbf{I}}] \\
&= \tilde{\mathbf{P}}(\mathbf{z})[[\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z})][\tilde{\mathbf{K}}^{-1} - \tilde{\mathbf{I}}]] \\
&= \tilde{\mathbf{K}}^{-1} - \tilde{\mathbf{I}} \qquad (3.30)
\end{aligned}$$

This completes the proof.

For grayscale error diffusion, this result reduces to the result derived in [19] in which the gain is scalar-valued and the error filter has scalar coefficients. Fig. 3.5 feeds a linear transformation $\tilde{\mathbf{L}}$ of the input image into the quantizer input. The matrix gain model predicts that the linear distortion in the halftoning process must be eliminated. To check this result, I first compute the residual of an unmodified halftone (i.e. halftoned using $\tilde{\mathbf{L}} = \tilde{\mathbf{0}}$) with respect to the original. Fig. 3.3(a) shows the original image to be halftoned. Fig. 3.3(c) shows the halftone image, which was halftoned with $\tilde{\mathbf{L}} = \tilde{\mathbf{0}}$ (usual vector color error diffusion). Fig. 3.6(a) shows the residual with respect to the original by subtracting Fig. 3.3(c) from Fig. 3.3(a). The correlation matrix for the residual is

$$C_{\mathbf{rx}} = \begin{pmatrix} 0.3204 & 0.2989 & 0.0999 \\ 0.2787 & 0.3295 & 0.1605 \\ 0.2063 & 0.2952 & 0.1836 \end{pmatrix}$$

Fig. 3.6(b) shows the halftone image, which was halftoned with $\tilde{\mathbf{L}} = \tilde{\mathbf{K}}^{-1} - \tilde{\mathbf{I}}$ (modified vector color error diffusion). Fig. 3.6(c) shows the residual with respect to the original by subtracting Fig. 3.6(b) from Fig. 3.3(a). The correlation matrix for the residual is

$$\tilde{C}_{\mathbf{rx}} = \begin{pmatrix} 0.0052 & 0.0009 & 0.0040 \\ 0.0054 & 0.0023 & 0.0020 \\ 0.0058 & 0.0011 & 0.0027 \end{pmatrix}$$

This shows that the linear distortion has been removed by modified vector color error diffusion, since the residual with respect to the original is uncorrelated noise (signal components in the residual have been eliminated).

Knox [18] shows that the error image for grayscale error diffusion $\mathbf{e}(\mathbf{m})$ is correlated with the input image. Knox also shows that the sharpness of halftones increases as the correlation of the error image with the input increases.

Kite, Evans and Bovik [19] show that by adding dither, the quantization error may be decorrelated with respect to the input, and the sharpening (linear distortion) effects of error diffusion vanish. They also conclude [19] that image sharpening is due to the fact that the input to the error filter contains signal components, which are fed back and shaped. Since the system has a highpass response, this results in the halftone being sharper than the original image.

I will show by using the matrix gain model that in the case of modified error diffusion (Fig. 3.5), halftoning with the value of $\tilde{\mathbf{L}}$ which cancels linear distortion is a sufficient condition for the error image (input to the error filter) to be free of signal components from the input image.

By replacing the quantizer in Fig. 3.5 with a gain matrix $\tilde{\mathbf{K}}$ and analyzing the signal path,

$$\begin{aligned} \mathbf{E}_s(\mathbf{z}) &= \tilde{\mathbf{K}} \left[\tilde{\mathbf{L}} \mathbf{X}(\mathbf{z}) + \mathbf{U}(\mathbf{z}) \right] - \mathbf{U}(\mathbf{z}) \\ &= \tilde{\mathbf{K}}\ \tilde{\mathbf{L}}\ \mathbf{X}(\mathbf{z}) + \left[\tilde{\mathbf{K}} - \tilde{\mathbf{I}} \right] \mathbf{U}(\mathbf{z}) \end{aligned} \tag{3.31}$$

Since

$$\mathbf{U}(\mathbf{z}) = \mathbf{X}(\mathbf{z}) - \tilde{\mathbf{H}}(\mathbf{z})\ \mathbf{E}_s(\mathbf{z}) \tag{3.32}$$

we obtain

$$\left[\tilde{\mathbf{I}} + \left(\tilde{\mathbf{K}} - \tilde{\mathbf{I}} \right) \tilde{\mathbf{H}}(\mathbf{z}) \right] \mathbf{E}_s(\mathbf{z}) = \left[\tilde{\mathbf{K}}\ \tilde{\mathbf{L}} + \tilde{\mathbf{K}} - \tilde{\mathbf{I}} \right] \mathbf{X}(\mathbf{z}) \tag{3.33}$$

By substituting $\tilde{\mathbf{L}} = \tilde{\mathbf{K}}^{-1} - \tilde{\mathbf{L}}$ into (3.33), $\mathbf{E}_s(\mathbf{z}) = 0$. Hence, there are no signal components in the error image. To check this prediction, and hence validate our modeling, I halftone test images with $\tilde{\mathbf{L}}$ set to cancel linear distortion. Fig. 3.3(a) shows the original image to be halftoned. Fig. 3.6(b) shows the halftone image by halftoning with $\tilde{\mathbf{L}} = \tilde{\mathbf{K}}^{-1} - \tilde{\mathbf{I}}$ (modified vector color error diffusion).

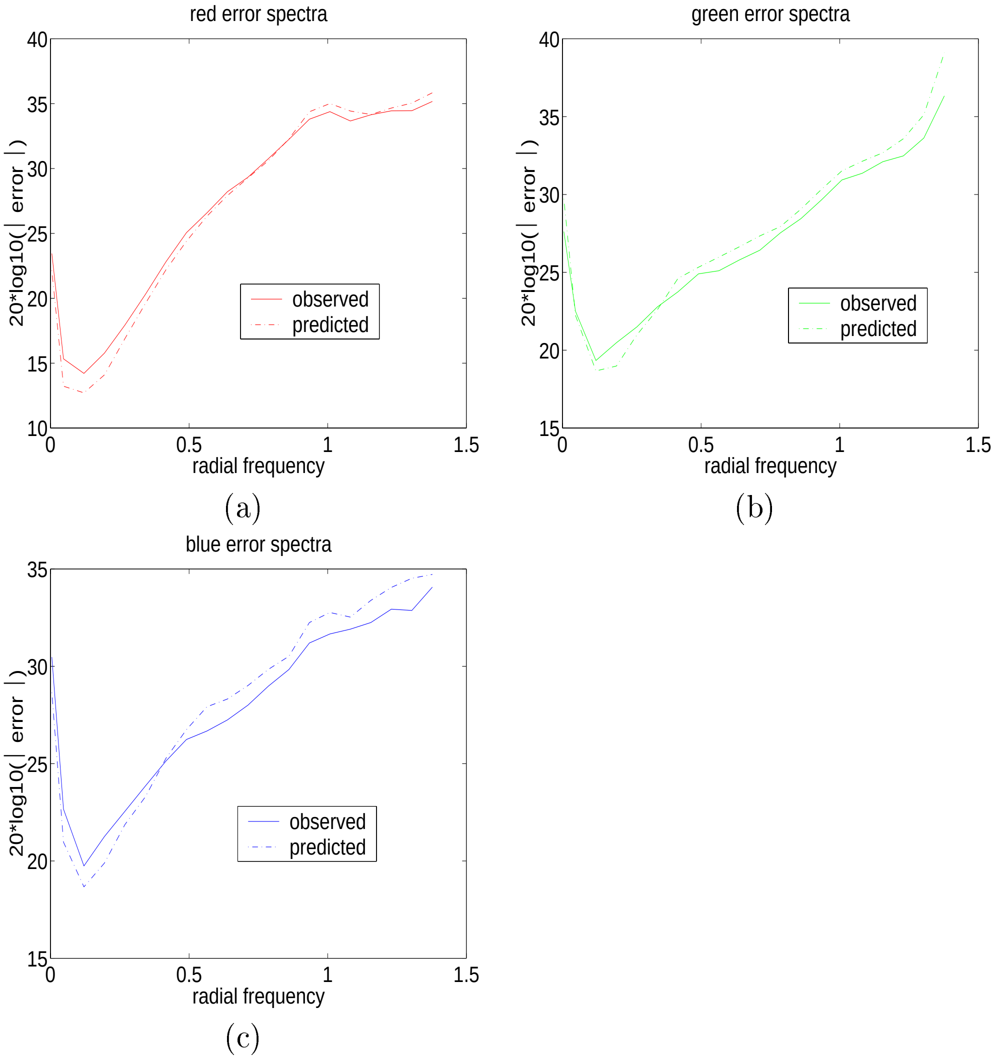
Fig. 3.6(d) shows the error image. The correlation matrix for the error image with respect to the original is

$$\tilde{C}_{\mathbf{ex}} = \begin{pmatrix} 0.0455 & 0.0235 & 0.0122 \\ 0.0493 & 0.0144 & 0.0164 \\ 0.0428 & 0.0142 & 0.0150 \end{pmatrix}$$

The low correlation of the error image was predicted by the theory and therefore strongly corroborates it.

3.3.3 Validation of the noise response

According to our model, the noise shaping is predicted by (3.14). To verify the prediction, I first compute a residual as described in Section 3.3.1. This residual is shaped noise. I need to verify that the noise shaping is in fact given by (3.14). I halftone test images using the optimal linear distortion cancelling method described in Section 3.3.2. This corresponds to halftoning with the value of $\tilde{\mathbf{L}} = \tilde{\mathbf{K}}^{-1} - \tilde{\mathbf{L}}$. The matrix gain model predicts that the input to the error filter has no signal components. The input to the error filter in this case is $\mathbf{N}(\mathbf{z})$. I then filter this noise image (i.e. input to the error filter) according to (3.14) to form a predicted residual. If the noise shaping equation is correct, then this residual must be spectrally close to the actual residual image. This was indeed found to be the case. Fig. 3.7 shows radially averaged spectra of the three color planes of the actual residual noise image and the residual computed using the noise shaping predicted from the model. The close agreement of the spectra confirms the predictions of the matrix gain model. The next section analyzes the valid use of the matrix gain model by considering the existence of matrix inverses assumed by the model.



(a)

(b)

(c)

Figure 3.7: Predicted and actual radially averaged spectra for residual noise image: (a) red, (b) green and (c) blue planes. Solid lines indicate actual spectra while the dashed lines represent predicted spectra

3.4 Invertibility of matrices used in the model

Typically, the matrix $\tilde{\mathbf{K}}$ is diagonally dominant with its diagonal elements greater than 1, so it is invertible. For the same reason, the matrix $[\tilde{\mathbf{I}}+\tilde{\mathbf{H}}(\mathbf{z})(\tilde{\mathbf{K}}-\tilde{\mathbf{I}})]$ in (3.10) is invertible. The proof of *Proposition 1* requires that the matrix $[\tilde{\mathbf{I}}-\tilde{\mathbf{H}}(\mathbf{z})]$ be invertible. This is typically not satisfied at DC for filters like the Floyd-Steinberg filter because $[\tilde{\mathbf{I}}-\tilde{\mathbf{H}}(\mathbf{1})]=\tilde{\mathbf{0}}$. However, empirical results indicate that $\lim_{\mathbf{z}\to\mathbf{1}}\mathbf{E}(\mathbf{z})$ does not blow up [39] because $\mathbf{B}(\mathbf{1})\approx\mathbf{X}(\mathbf{1})$ and $\tilde{\mathbf{G}}(\mathbf{1})=\tilde{\mathbf{I}}$. In fact, the zero at DC cancels the pole at DC, and $\lim_{\mathbf{z}\to\mathbf{1}}\mathbf{E}(\mathbf{z})=\mathbf{0}$ for the block-diagram of Fig. 3.5 and equal to $-\tilde{\mathbf{L}}\mathbf{X}(\mathbf{1})$ for the block-diagram of Fig. 3.4. This means that the two block diagrams are equivalent at DC because they have the same input to the quantizer. This is predicted by the matrix gain model. Consider Fig. 3.5 by noting that

$$\begin{aligned}[\tilde{\mathbf{I}}-\tilde{\mathbf{H}}(\mathbf{z})]\mathbf{E}(\mathbf{z}) &= \mathbf{B}(\mathbf{z})-\mathbf{X}(\mathbf{z}) \\ &= \tilde{\mathbf{K}}[\tilde{\mathbf{I}}+\tilde{\mathbf{L}}]\mathbf{X}(\mathbf{z})-\tilde{\mathbf{K}}\tilde{\mathbf{H}}(\mathbf{z})\mathbf{E}(\mathbf{z}) \qquad (3.34)\end{aligned}$$

This implies that

$$[\tilde{\mathbf{I}}-\tilde{\mathbf{H}}(\mathbf{z})+\tilde{\mathbf{K}}\tilde{\mathbf{H}}(\mathbf{z})]\mathbf{E}(\mathbf{z})=[\tilde{\mathbf{K}}(\tilde{\mathbf{I}}+\tilde{\mathbf{L}})-\tilde{\mathbf{I}}]\mathbf{X}(\mathbf{z}) \qquad (3.35)$$

By taking the limit as $\mathbf{z}\to\mathbf{1}$,

$$\begin{aligned}\mathbf{E}(\mathbf{1}) &= [\tilde{\mathbf{I}}+\tilde{\mathbf{L}}-\tilde{\mathbf{K}}^{-1}]\mathbf{X}(\mathbf{1}) \\ &= \mathbf{0} \qquad (3.36)\end{aligned}$$

By analyzing Fig. 3.4 in a similar manner, $\lim_{\mathbf{z}\to\mathbf{1}}=-\tilde{\mathbf{L}}\ \mathbf{X}(\mathbf{1})$

From (3.18) and (3.25), the two block-diagrams are equivalent at DC. At other frequencies for which $[\tilde{\mathbf{I}}-\tilde{\mathbf{H}}(\mathbf{z})]$ might not be invertible, a similar analysis using the matrix gain model may be applied to show that Figs. 3.4 and 3.5 are equivalent. However, the *exact* analysis may be in error to some extent when $[\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z})]$ is not invertible.

3.5 Conclusion

This chapter introduces a new modeling approach for color error diffusion. By regarding the quantization noise injection as the estimation error of a linear minimum mean squared error estimator, I linearize error diffusion. Separate matrix-valued signal shaping and noise shaping filters are determined which accurately predict the signal and noise shaping behavior of color error diffusion. Using the model, I show that a linear shift-invariant prefilter can eliminate the signal sharpening effects of error diffusion. Further, I show that such a prefilter can be incorporated with low-complexity by modifying the error diffusion system to feed a linear transformation of the quantizer input. Finally, the mathematical validity of the modeling is discussed. The next chapter designs a matrix-valued error filter based on the noise shaping predicted by the matrix gain model. The error filter is designed so that the visibility of the halftone noise is minimized.

Chapter 4

Optimum Error Filter Design for Color Error Diffusion Halftoning

4.1 Introduction

In designing the color error filter coefficients, I use the matrix gain model along with a sophisticated model for human color vision. The formulation results in an uncorrelated noise image replacing the highly correlated error image in the objective function of [23]. Thus, the optimization becomes less restrictive since I do not compensate for or try to minimize correlated signal components in the error image. Recall from Chapter 3 that the correlated signal components in the error image produced a sharpening effect which is usually desirable. I assume the *uncorrelated* noise image is a white noise process as in [21]. I minimize the visual impact of the quantization noise by incorporating the matrix gain model into the optimization along with a linear model for human color vision. I show that the optimal filter may be obtained by a solution of a matrix version of the Yule-Walker equations [40]. Because the error filter does not need to minimize correlated signal components, the filter can be solely optimized for optimal noise shaping.

Section 4.2 describes a linear color model for the human visual system and shows that it may be represented as a linear transformation followed by spatial filtering. Section 4.3 formulates the design problem as a quadratic optimization and derives the optimal solution. The optimal solution is compared quantitatively and qualitatively to separable Floyd-Steinberg error diffusion. Section 4.4 concludes the chapter by summarizing the central ideas governing the design algorithm.

4.2 Linear color model for the human visual system

To obtain a true matrix linear color model, one needs to model the color processing of the human visual system as a convolution with a matrix-valued filter $\tilde{\mathbf{v}}(\mathbf{m})$. The development of such a model is beyond the scope of this paper and a topic for future research. Instead, I use a pattern-color separable model for the human visual system based on the work of Poirson and Wandell [36, 41]. The pattern-color separable color vision model forms the basis for the S-CIELab color space, which has become an industry standard [42]. The pattern-color separable model first transforms device dependent RGB values (where R, G and B are coefficients of standard spectral tristimulus basis functions) into a space with basis functions represented by the normalized color sensitivities of the three fundamental cones responsible for human color vision. The three cones are called the L, M and S cones respectively, to denote long (L), medium (M), and short (S) wavelength sensitivities. Thus, at each pixel an RGB value is transformed into the corresponding cone photoreceptor absorption rates. The L, M and S basis functions are referred to in the literature as the Smith-Pokorny cone fundamentals [43]. The LMS coordinates are then transformed using a color transformation into an opponent representation [44].

The three opponent visual pathways are the white-black (or the luminance pathway), red-green and blue-yellow pathways (chrominance pathways). The "-" in red-green and blue-yellow should be read as "minus" and not confused with a hyphen. Thus white and black are in opposition, red and green are in opposition, and blue and yellow are in opposition. Such a representation is very different from early RGB models where it was believed that humans respond to the three primary colors [44]. Strong support for the opponent representation comes from the fact that humans do not perceive colors that are reddish green or yellowish blue since the red-green and yellow-blue visual pathways are opponent channels. Poirson and Wandell [36, 41] found that spatial frequency sensitivity to color patterns could be modeled as spatial frequency sensitivity of the three channels in the opponent representation.

Thus, the linear color model consists of

1. A linear transformation $\tilde{\mathbf{T}}$ and,
2. Separable spatial filtering on each channel using a different spatial filter on each channel. This operation may be regarded as a matrix multiplication in the frequency domain by a diagonal matrix $\tilde{\mathbf{D}}(\mathbf{z})$.

Thus, $\mathbf{v}(\mathbf{m})$ is computed as

$$\mathbf{v}(\mathbf{m}) = \tilde{\mathbf{d}}(\mathbf{m})\tilde{\mathbf{T}} \tag{4.1}$$

I now describe the the computation of the model parameters for viewing RGB images on a monitor. First, one needs to account for the fact that the 8-bit values that are put in the frame buffer to trigger the red, green, and blue guns of the CRT are *not* the RGB tristimulus values of the colors displayed on the monitor. This is because the CRT has a nonlinear response to frame buffer values. Thus, I need to pass the RGB values of the image through this nonlinearity to obtain the RGB coordinates of the colors displayed on the monitor. This corresponds to the inverse of gamma correction. The color images are first pre-processed with this point-nonlinearity before they are halftoned. This ensures that the colors in the halftone are closest to the color actually rendered on the monitor.

The linear transformation $\tilde{\mathbf{T}}$ is computed as the composition of two linear transformations $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{O}}$. The transformation $\tilde{\mathbf{C}}$ is the transformation that converts linear RGB values into Smith-Pokorny cone absorption rates. $\tilde{\mathbf{C}}$ is a monitor dependent transformation. The transformation $\tilde{\mathbf{O}}$ that transforms the LMS coordinates into the opponent representation is given in [36, 41, 42] and is monitor independent.

The spatial frequency weighting functions for the three opponent visual pathways were obtained for viewing images displayed on the monitor at 72 dpi (dots per inch) at a "normal" viewing distance of 18 inches using the parameters given in [42].

The next section uses the visual model developed in this section to find optimal matrix-valued error filter coefficients for vector color error diffusion.

4.3 Designing the Error Filter

This section considers the design of the matrix valued coefficients of the error filter. The matrix gain model is incorporated in the optimization process. The noise shaping predicted by the matrix gain model is

$$\mathbf{B}_n(\mathbf{z}) = [\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z})]\mathbf{N}(\mathbf{z}) \tag{4.2}$$

In the spatial domain, (4.2) may be expressed as:

$$\mathbf{b}_n(\mathbf{m}) = [\tilde{\mathbf{I}} - \tilde{\mathbf{h}}(\mathbf{m})] * \mathbf{n}(\mathbf{m}) \tag{4.3}$$

The filter coefficients $\tilde{\mathbf{h}}_{opt}(\mathbf{m})$ must be chosen such that

$$E\left[\| \mathbf{b}_n^{(\tilde{\mathbf{v}})}(\mathbf{m}) \|^2\right] = E\left[\| \tilde{\mathbf{v}}(\mathbf{m}) * [\tilde{\mathbf{I}} - \tilde{\mathbf{h}}(\mathbf{m})] * \mathbf{n}(\mathbf{m}) \|^2\right] \tag{4.4}$$

is minimized. Here, $\tilde{\mathbf{v}}(\mathbf{m})$ represents a filter with matrix-valued coefficients that models the linear approximation of the human visual system response. To ensure that all of the error is diffused (lossless error diffusion) from each image plane I introduce the constraint

$$\tilde{\mathbf{\Gamma}}\mathbf{1}_{3M\times 1} = \mathbf{1}_{3\times 1} \tag{4.5}$$

where $\mathbf{1}_{3M\times 1}$ is a $3M \times 1$ column vector with all elements equal to unity, and $\mathbf{1}_{3\times 1}$ is a 3×1 column vector with all of its elements equal to unity:

$$\tilde{\mathbf{\Gamma}} = \left[\tilde{\mathbf{h}}_0 \mid \tilde{\mathbf{h}}_1 \mid \ldots \mid \tilde{\mathbf{h}}_{M-1}\right] \tag{4.6}$$

I assume that the cardinality of $\mathcal{S}$, denoted $\mid \mathcal{S} \mid$, is M. I can then order the elements of $\mathcal{S}$ according to rows, from $0 \ldots M-1$. $\tilde{\mathbf{\Gamma}}$ represents a matrix formed by concatenating the M matrices $\tilde{\mathbf{h}}(\mathbf{k})$, $\mathbf{k} \in \mathcal{S}$, ordered according to rows. The matrix $\tilde{\mathbf{\Gamma}}$ proves useful in deriving an efficient algorithm for implementing the error filter, as discussed in Section 2.4. The constraint set may also be expressed as

$$\mathcal{C} = \left\{\tilde{\mathbf{h}}(\mathbf{i}),\ \mathbf{i} \in \mathcal{S} \mid \sum_{\mathbf{i}} \tilde{\mathbf{h}}(\mathbf{i})\mathbf{1} = \mathbf{1}\right\} \tag{4.7}$$

For the purpose of deriving the optimal filter coefficients, I will not assume the pattern-color separable model, but instead use the most general linear spatially-invariant weighting possible (viz: a spatial filter with unstructured matrix-valued coefficients). I rewrite (4.4) as

$$\begin{aligned} E\left[\| \mathbf{b}_n^{(\tilde{\mathbf{v}})}(\mathbf{m}) \|^2\right] &= E\left[\| \tilde{\mathbf{a}}(\mathbf{m}) - \tilde{\mathbf{v}}(\mathbf{m}) * \tilde{\mathbf{h}}(\mathbf{m}) * \mathbf{n}(\mathbf{m}) \|^2\right] \\ &= E\left[\| \tilde{\mathbf{a}}(\mathbf{m}) - \sum_{\mathbf{k}} \sum_{\mathbf{k}'} \tilde{\mathbf{v}}(\mathbf{k}')\tilde{\mathbf{h}}(\mathbf{k})\mathbf{n}(\mathbf{m}-\mathbf{k}'-\mathbf{k}) \|^2\right] \\ &= E\ Tr\left[\tilde{\mathbf{a}}(\mathbf{m}) - \sum_{\mathbf{k}} \sum_{\mathbf{k}'} \tilde{\mathbf{v}}(\mathbf{k}')\tilde{\mathbf{h}}(\mathbf{k})\mathbf{n}(\mathbf{m}-\mathbf{k}'-\mathbf{k})\right] \\ &\quad \left[\tilde{\mathbf{a}}(\mathbf{m}) - \sum_{\mathbf{k}} \sum_{\mathbf{k}'} \tilde{\mathbf{v}}(\mathbf{k}')\tilde{\mathbf{h}}(\mathbf{k})\mathbf{n}(\mathbf{m}-\mathbf{k}'-\mathbf{k})\right]^T \end{aligned} \tag{4.8}$$

where I have substituted $\mathbf{a}(\mathbf{m}) = \tilde{\mathbf{v}}(\mathbf{m}) * \mathbf{n}(\mathbf{m})$ and used the fact that for a vector $\mathbf{x}$, $\| \mathbf{x} \|^2 = Tr\left[\mathbf{x}\mathbf{x}^T\right]$, where Tr denotes the trace operation. Also since the trace is a linear functional, (4.8) may be further simplified as

$$E\left[\| \mathbf{b}_n^{(\tilde{\mathbf{v}})}(\mathbf{m}) \|^2\right] = \Theta_1 + \Theta_2 + \Theta_3 + \Theta_4 \tag{4.9}$$

where

$$\Theta_1 = E\ Tr\left[\tilde{\mathbf{a}}(\mathbf{m})\tilde{\mathbf{a}}^T(\mathbf{m})\right] \tag{4.10}$$

$$\Theta_2 = -E\ Tr\left[\tilde{\mathbf{a}}(\mathbf{m}) \sum_{\mathbf{k}} \sum_{\mathbf{k}'} \mathbf{n}^T(\mathbf{m}-\mathbf{k}'-\mathbf{k})\tilde{\mathbf{h}}^T(\mathbf{k})\tilde{\mathbf{v}}^T(\mathbf{k}')\right] \tag{4.11}$$

$$\Theta_3 = -E\ Tr\left[\sum_{\mathbf{k}} \sum_{\mathbf{k}'} \tilde{\mathbf{v}}(\mathbf{k}')\tilde{\mathbf{h}}(\mathbf{k})\mathbf{n}(\mathbf{m}-\mathbf{k}'-\mathbf{k})\tilde{\mathbf{a}}^T(\mathbf{m})\right] \tag{4.12}$$

$$\Theta_4 = E\ Tr\left[\sum_{\mathbf{p}} \sum_{\mathbf{q}} \sum_{\mathbf{r}} \sum_{\mathbf{s}} \tilde{\mathbf{v}}(\mathbf{s})\tilde{\mathbf{h}}(\mathbf{r})\mathbf{n}(\mathbf{m}-\mathbf{s}-\mathbf{r})\mathbf{n}^T(\mathbf{m}-\mathbf{p}-\mathbf{q})\tilde{\mathbf{h}}^T(\mathbf{p})\tilde{\mathbf{v}}^T(\mathbf{q})\right] \tag{4.13}$$

By taking the first partial derivatives of (4.4) with respect to $\tilde{\mathbf{h}}(\mathbf{i})$ for all $\mathbf{i} \in \mathcal{S}$ and setting them to zero, I obtain the first-order necessary conditions for an optimum solution. This requires that a scalar function be differentiated with respect to a matrix. To do this, some results from linear algebra are required.

The following results are stated here without proof. For proofs of the following, please see [40]:

$$\frac{d}{d\tilde{\mathbf{X}}^T} f(\tilde{\mathbf{X}}) = \left(\frac{d}{d\tilde{\mathbf{X}}} f(\tilde{\mathbf{X}})\right)^T \tag{4.14}$$

$$\frac{d}{d\tilde{\mathbf{X}}} Tr(\tilde{\mathbf{A}}\tilde{\mathbf{X}}) = \tilde{\mathbf{A}}^T \tag{4.15}$$

$$\frac{d}{d\tilde{\mathbf{X}}} Tr(\tilde{\mathbf{A}}\tilde{\mathbf{X}}\tilde{\mathbf{B}}) = \tilde{\mathbf{A}}^T\tilde{\mathbf{B}}^T \tag{4.16}$$

$$\frac{d}{d\tilde{\mathbf{X}}} Tr(\tilde{\mathbf{X}}^T\tilde{\mathbf{A}}\tilde{\mathbf{X}}\tilde{\mathbf{B}}) = \tilde{\mathbf{A}}\tilde{\mathbf{X}}\tilde{\mathbf{B}} + \tilde{\mathbf{A}}^T\tilde{\mathbf{X}}\tilde{\mathbf{B}}^T \tag{4.17}$$

$$Tr(\tilde{\mathbf{A}}\tilde{\mathbf{B}}) = Tr(\tilde{\mathbf{B}}\tilde{\mathbf{A}}) \tag{4.18}$$

Let us consider the terms Θ_1, Θ_2, Θ_3 and Θ_4.

$$\frac{d}{d\tilde{\mathbf{h}}(\mathbf{i})}\Theta_1 = 0 \tag{4.19}$$

By using (4.14) and (4.16),

$$\begin{aligned}
\frac{d}{d\tilde{\mathbf{h}}(\mathbf{i})}\Theta_2 &= \left(\frac{d}{d\tilde{\mathbf{h}}^T(\mathbf{i})}\Theta_2\right)^T \\
&= -E\left[\frac{d}{d\tilde{\mathbf{h}}^T(\mathbf{i})} Tr \sum_{\mathbf{k}'} \mathbf{a}(\mathbf{m})\mathbf{n}^T(\mathbf{m}-\mathbf{k}'-\mathbf{i})\tilde{\mathbf{h}}^T(\mathbf{i})\tilde{\mathbf{v}}^T(\mathbf{k}')\right]^T \\
&= -E\left[\sum_{\mathbf{k}'} \mathbf{n}(\mathbf{m}-\mathbf{k}'-\mathbf{i})\mathbf{a}^T(\mathbf{m})\tilde{\mathbf{v}}(\mathbf{k}')\right]^T \\
&= -E\left[\sum_{\mathbf{k}'} \tilde{\mathbf{v}}^T(\mathbf{k}')\mathbf{a}(\mathbf{m})\mathbf{n}^T(\mathbf{m}-\mathbf{k}'-\mathbf{i})\right] \\
&= -\sum_{\mathbf{k}} \tilde{\mathbf{v}}^T(\mathbf{k})\tilde{\mathbf{r}}_{\mathbf{an}}(-\mathbf{i}-\mathbf{k})
\end{aligned} \tag{4.20}$$

By using (4.16),

$$\begin{aligned}
\frac{d}{d\tilde{\mathbf{h}}(\mathbf{i})}\Theta_3 &= -E\left[\frac{d}{d\tilde{\mathbf{h}}(\mathbf{i})}Tr\sum_{\mathbf{k}'}\tilde{\mathbf{v}}(\mathbf{k}')\tilde{\mathbf{h}}(\mathbf{i})\mathbf{n}(\mathbf{m}-\mathbf{k}'-\mathbf{i})\mathbf{a}^T(\mathbf{m})\right] \\
&= -E\left[\sum_{\mathbf{k}'}\tilde{\mathbf{v}}^T(\mathbf{k}')\mathbf{a}(\mathbf{m})\mathbf{n}^T(\mathbf{m}-\mathbf{k}'-\mathbf{i})\right] \\
&= -\sum_{\mathbf{k}}\tilde{\mathbf{v}}^T(\mathbf{k})\tilde{\mathbf{r}}_{\mathbf{an}}(-\mathbf{i}-\mathbf{k}) \qquad (4.21)
\end{aligned}$$

By considering Θ_4, $\tilde{\mathbf{h}}(\mathbf{i})$ only occurs in three terms Θ_{41}, Θ_{42} and Θ_{43} where

$$\begin{aligned}
\Theta_{41} &= E\left[Tr\sum_{\mathbf{p}\neq\mathbf{i}}\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{v}}(\mathbf{s})\tilde{\mathbf{h}}(\mathbf{i})\mathbf{n}(\mathbf{m}-\mathbf{s}-\mathbf{i})\mathbf{n}^T(\mathbf{m}-\mathbf{p}-\mathbf{q})\tilde{\mathbf{h}}^T(\mathbf{p})\tilde{\mathbf{v}}^T(\mathbf{q})\right] \\
\Theta_{42} &= E\left[Tr\sum_{\mathbf{r}\neq\mathbf{i}}\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{v}}(\mathbf{s})\tilde{\mathbf{h}}(\mathbf{r})\mathbf{n}(\mathbf{m}-\mathbf{s}-\mathbf{r})\mathbf{n}^T(\mathbf{m}-\mathbf{i}-\mathbf{q})\tilde{\mathbf{h}}^T(\mathbf{i})\tilde{\mathbf{v}}^T(\mathbf{q})\right] \\
\Theta_{43} &= E\left[Tr\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{v}}(\mathbf{s})\tilde{\mathbf{h}}(\mathbf{i})\mathbf{n}(\mathbf{m}-\mathbf{s}-\mathbf{i})\mathbf{n}^T(\mathbf{m}-\mathbf{i}-\mathbf{q})\tilde{\mathbf{h}}^T(\mathbf{i})\tilde{\mathbf{v}}^T(\mathbf{q})\right] \qquad (4.22)
\end{aligned}$$

By using (4.16),

$$\begin{aligned}
\frac{d}{d\tilde{\mathbf{h}}(\mathbf{i})}\Theta_{41} &= E\left[\sum_{\mathbf{p}\neq\mathbf{i}}\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{v}}^T(\mathbf{s})\{\mathbf{n}(\mathbf{m}-\mathbf{s}-\mathbf{i})\mathbf{n}^T(\mathbf{m}-\mathbf{p}-\mathbf{q})\tilde{\mathbf{h}}^T(\mathbf{p})\tilde{\mathbf{v}}^T(\mathbf{q})\}^T\right] \\
&= E\left[\sum_{\mathbf{p}\neq\mathbf{i}}\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{v}}^T(\mathbf{s})\tilde{\mathbf{v}}(\mathbf{q})\tilde{\mathbf{h}}(\mathbf{p})\mathbf{n}(\mathbf{m}-\mathbf{p}-\mathbf{q})\mathbf{n}^T(\mathbf{m}-\mathbf{s}-\mathbf{i})\right] \\
&= \sum_{\mathbf{p}\neq\mathbf{i}}\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{v}}^T(\mathbf{s})\tilde{\mathbf{v}}(\mathbf{q})\tilde{\mathbf{h}}(\mathbf{p})\tilde{\mathbf{r}}_{\mathbf{nn}}(\mathbf{p}+\mathbf{q}-\mathbf{s}-\mathbf{i}) \qquad (4.23)
\end{aligned}$$

By using (4.14) and (4.16) as in (4.20),

$$\begin{aligned}
\frac{d}{d\tilde{\mathbf{h}}(\mathbf{i})}\Theta_{42} &= \left[\frac{d}{d\tilde{\mathbf{h}}^T(\mathbf{i})}\Theta_{42}\right]^T \\
&= E\left[\sum_{\mathbf{r}\neq\mathbf{i}}\sum_{\mathbf{q}}\sum_{\mathbf{s}}\{\tilde{\mathbf{v}}(\mathbf{s})\tilde{\mathbf{h}}(\mathbf{r})\mathbf{n}(\mathbf{m}-\mathbf{s}-\mathbf{r})\mathbf{n}^T(\mathbf{m}-\mathbf{i}-\mathbf{q})\}^T\tilde{\mathbf{v}}(\mathbf{q})\right]^T \\
&= E\left[\sum_{\mathbf{r}\neq\mathbf{i}}\sum_{\mathbf{q}}\sum_{\mathbf{s}}\mathbf{n}(\mathbf{m}-\mathbf{i}-\mathbf{q})\mathbf{n}^T(\mathbf{m}-\mathbf{s}-\mathbf{r})\tilde{\mathbf{h}}^T(\mathbf{r})\tilde{\mathbf{v}}^T(\mathbf{s})\tilde{\mathbf{v}}(\mathbf{q})\right]^T
\end{aligned}$$

$$\begin{aligned}
&= E\left[\sum_{\mathbf{r}\neq\mathbf{i}}\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{v}}^T(\mathbf{q})\tilde{\mathbf{v}}(\mathbf{s})\tilde{\mathbf{h}}(\mathbf{r})\mathbf{n}(\mathbf{m}-\mathbf{s}-\mathbf{r})\mathbf{n}^T(\mathbf{m}-\mathbf{q}-\mathbf{i})\right] \\
&= \sum_{\mathbf{r}\neq\mathbf{i}}\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{v}}^T(\mathbf{q})\tilde{\mathbf{v}}(\mathbf{s})\tilde{\mathbf{h}}(\mathbf{r})\tilde{\mathbf{r}}_{\mathbf{nn}}(\mathbf{r}+\mathbf{s}-\mathbf{q}-\mathbf{i}) \\
&= \sum_{\mathbf{p}\neq\mathbf{i}}\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{v}}^T(\mathbf{s})\tilde{\mathbf{v}}(\mathbf{q})\tilde{\mathbf{h}}(\mathbf{p})\tilde{\mathbf{r}}_{\mathbf{nn}}(\mathbf{p}+\mathbf{q}-\mathbf{s}-\mathbf{i})
\end{aligned} \tag{4.24}$$

To simplify $\frac{d}{d\tilde{\mathbf{h}}(\mathbf{i})}\Theta_{43}$, I use (4.18) and (4.17)

$$\Theta_{43} = E\left[Tr\sum_{\mathbf{q}}\sum_{\mathbf{s}}\underbrace{\tilde{\mathbf{v}}(\mathbf{s})}_{\tilde{\mathbf{U}}_{\mathbf{s}}}\underbrace{\tilde{\mathbf{h}}(\mathbf{i})}_{\tilde{\mathbf{X}}}\underbrace{\mathbf{n}(\mathbf{m}-\mathbf{s}-\mathbf{i})\mathbf{n}^T(\mathbf{m}-\mathbf{i}-\mathbf{q})}_{\tilde{\mathbf{V}}_{\mathbf{q},\mathbf{s}}}\underbrace{\tilde{\mathbf{h}}^T(\mathbf{i})}_{\tilde{\mathbf{X}}^T}\underbrace{\tilde{\mathbf{v}}^T(\mathbf{q})}_{\tilde{\mathbf{W}}_{\mathbf{q}}}\right] \tag{4.25}$$

By using (4.18) in (4.25) and applying (4.17),

$$\begin{aligned}
\frac{d}{d\tilde{\mathbf{X}}}\Theta_{43} &= \frac{d}{d\tilde{\mathbf{X}}}E\left[Tr\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{U}}_{\mathbf{s}}\tilde{\mathbf{X}}\tilde{\mathbf{V}}_{\mathbf{q},\mathbf{s}}\tilde{\mathbf{X}}^T\tilde{\mathbf{W}}_{\mathbf{q}}\right] \\
&= \frac{d}{d\tilde{\mathbf{X}}}E\left[Tr\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{X}}^T\tilde{\mathbf{W}}_{\mathbf{q}}\tilde{\mathbf{U}}_{\mathbf{s}}\tilde{\mathbf{X}}\tilde{\mathbf{V}}_{\mathbf{q},\mathbf{s}}\right] \\
&= E\left[\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{W}}_{\mathbf{q}}\tilde{\mathbf{U}}_{\mathbf{s}}\tilde{\mathbf{X}}\tilde{\mathbf{V}}_{\mathbf{q},\mathbf{s}}+\tilde{\mathbf{U}}_{\mathbf{s}}^T\tilde{\mathbf{W}}_{\mathbf{q}}^T\tilde{\mathbf{X}}\tilde{\mathbf{V}}_{\mathbf{q},\mathbf{s}}^T\right] \\
&= E\left[\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{v}}^T(\mathbf{q})\tilde{\mathbf{v}}(\mathbf{s})\tilde{\mathbf{h}}(\mathbf{i})\mathbf{n}(\mathbf{m}-\mathbf{s}-\mathbf{i})\mathbf{n}^T(\mathbf{m}-\mathbf{i}-\mathbf{q})\right] \\
&\quad +E\left[\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{v}}^T(\mathbf{s})\tilde{\mathbf{v}}(\mathbf{q})\tilde{\mathbf{h}}(\mathbf{i})\mathbf{n}(\mathbf{m}-\mathbf{i}-\mathbf{q})\mathbf{n}^T(\mathbf{m}-\mathbf{s}-\mathbf{i})\right] \\
&= 2E\left[\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{v}}^T(\mathbf{s})\tilde{\mathbf{v}}(\mathbf{q})\tilde{\mathbf{h}}(\mathbf{i})\mathbf{n}(\mathbf{m}-\mathbf{i}-\mathbf{q})\mathbf{n}^T(\mathbf{m}-\mathbf{s}-\mathbf{i})\right] \\
&= 2\sum_{\mathbf{q}}\sum_{\mathbf{s}}\tilde{\mathbf{v}}^T(\mathbf{s})\tilde{\mathbf{v}}(\mathbf{q})\tilde{\mathbf{h}}(\mathbf{i})\tilde{\mathbf{r}}_{\mathbf{nn}}(\mathbf{q}-\mathbf{s})
\end{aligned} \tag{4.26}$$

Finally, combining (4.26) with (4.24) and (4.23) and combining (4.21) and (4.20) yields the first-order necessary conditions for an optimum solution to the minimization of (4.4)

$$\sum_{\mathbf{k}} \tilde{\mathbf{v}}^T(\mathbf{k})\tilde{\mathbf{r}}_{\mathbf{an}}(-\mathbf{i}-\mathbf{k}) = \sum_{\mathbf{p}}\sum_{\mathbf{q}}\sum_{\mathbf{s}} \tilde{\mathbf{v}}^T(\mathbf{s})\tilde{\mathbf{v}}(\mathbf{q})\tilde{\mathbf{h}}(\mathbf{p})\tilde{\mathbf{r}}_{\mathbf{nn}}(\mathbf{p}+\mathbf{q}-\mathbf{s}-\mathbf{i}) \quad (4.27)$$

These equations may be regarded as a generalization of the Yule-Walker equations [40] from linear prediction theory to the matrix case, with a generalized linear spatially-invariant weighting. The above set of generalized Yule-Walker equations may be solved for the optimal filter subject to the constraints of (4.5) using the steepest descent algorithm [40].

I use a white noise image as an approximation to the uncorrelated noise image $\mathbf{n}(\mathbf{m})$. Thus, the required autocorrelation matrices are approximated as

$$\tilde{\mathbf{r}}_{\mathbf{nn}}(\mathbf{k}) = E\left[\mathbf{n}(\mathbf{m})\mathbf{n}^T(\mathbf{m}+\mathbf{k})\right] \approx \delta(\mathbf{k}) \quad (4.28)$$

$$\tilde{\mathbf{r}}_{\mathbf{an}}(\mathbf{k}) = E\left(\sum_{\mathbf{t}} \tilde{\mathbf{v}}(\mathbf{t})\mathbf{n}(\mathbf{m}-\mathbf{t})\right)\mathbf{n}^T(\mathbf{m}+\mathbf{k}) \approx \sum_{\mathbf{t}} \tilde{\mathbf{v}}(\mathbf{t})\delta(\mathbf{k}+\mathbf{t}) = \tilde{\mathbf{v}}(-\mathbf{k}) \quad (4.29)$$

where $\delta(\mathbf{k})$ is the two-dimensional Kronecker delta function [40]. In the optimization, the constraint is enforced by projection onto the convex constraint set. The convergence behavior of this algorithm is discussed in [45]. The algorithm is guaranteed to converge if the convergence parameter in the descent algorithm is chosen to be small enough [45].

The descent algorithm may be formulated as

$$(\nabla J)(\tilde{\mathbf{h}}(\mathbf{i})) \triangleq -\sum_{\mathbf{k}} \tilde{\mathbf{v}}^T(\mathbf{k})\tilde{\mathbf{v}}(\mathbf{i}+\mathbf{k}) + \sum_{\mathbf{p}}\sum_{\mathbf{q}}\sum_{\mathbf{s}} \tilde{\mathbf{v}}^T(\mathbf{s})\tilde{\mathbf{v}}(\mathbf{q})\tilde{\mathbf{h}}(\mathbf{p})\delta(\mathbf{p}+\mathbf{q}-\mathbf{s}-\mathbf{i}) \quad (4.30)$$

$$\tilde{\mathbf{f}}^{(\theta)}(\mathbf{i}) \triangleq \tilde{\mathbf{h}}^{(\theta)}(\mathbf{i}) - \alpha(\nabla J)(\tilde{\mathbf{h}}^{(\theta)}(\mathbf{i})) \quad (4.31)$$

$$\tilde{\mathbf{h}}^{(\theta+1)}(\mathbf{i}) = \mathcal{P}\left(\tilde{\mathbf{f}}^{(\theta)}(\mathbf{i})\right) \quad (4.32)$$

where θ refers to the iteration number, and $\mathcal{P}$ is the projection operator that projects the iterate into the constraint set $\mathcal{C}$, defined by (4.7). I use the convergence parameter $\alpha = 0.005$ in my simulations. The projection operator is defined as [46]

$$\mathcal{P}\left(\tilde{\mathbf{f}}^{(\theta)}(\mathbf{i})\right) \triangleq \tilde{\mathbf{f}}^{(\theta)}(\mathbf{i}) - \frac{1}{3\mid\mathcal{S}\mid}\left(\sum_{\mathbf{i}\in\mathcal{S}} \tilde{\mathbf{f}}^{(\theta)}(\mathbf{i}) - \tilde{\mathbf{I}}\right)\mathbf{1} \tag{4.33}$$

Several random initial guesses were tried, and the descent algorithm was terminated when the changes in the objective function were below a predefined threshold. Using this method, one may explore different minimizers (solutions that result in nearly the same objective function value). The uniformity in the dot distributions produced by different initial guesses was different. It has been shown [47, 48, 49] that frequency weighted mean squared error alone cannot guarantee optimum dot distributions. This problem can be alleviated by using threshold modulation [14]. For the purpose of this work, since I am only concerned with the noise shaping behavior of error diffusion, I chose a solution that had a reasonably uniform dot distribution.

My calibration data used a monitor display $\gamma \approx 2.2$, and a monitor dependent transformation matrix $\tilde{\mathbf{O}}$

$$\tilde{\mathbf{O}} = \begin{pmatrix} 2.0935 & 7.6018 & 1.1235 \\ 0.7921 & 7.6394 & 1.6264 \\ 0.0894 & 0.8020 & 7.6618 \end{pmatrix} \tag{4.34}$$

The optimal filter coefficients obtained for this monitor were

$$\begin{aligned} \tilde{\mathbf{h}}(0,1) &= \begin{pmatrix} 0.6316 & -0.1306 & 0.0323 \\ -0.0430 & 0.3993 & 0.0327 \\ -0.0167 & -0.1082 & 0.7379 \end{pmatrix} \\ \tilde{\mathbf{h}}(1,1) &= \begin{pmatrix} -0.1949 & 0.1289 & -0.0242 \\ 0.0817 & -0.0730 & 0.0645 \\ 0.0454 & 0.1585 & -0.4017 \end{pmatrix} \end{aligned}$$

$$\tilde{\mathbf{h}}(1,0) = \begin{pmatrix} 0.3598 & -0.0549 & 0.0403 \\ -0.0018 & 0.2906 & 0.0173 \\ -0.0080 & -0.0895 & 0.4867 \end{pmatrix}$$

$$\tilde{\mathbf{h}}(1,-1) = \begin{pmatrix} 0.2181 & -0.0112 & 0.0047 \\ 0.0222 & 0.1515 & 0.0580 \\ 0.0129 & 0.0213 & 0.1614 \end{pmatrix}$$

The optimal filter that was obtained based on our calibrated color monitor and was tested on five standard color test images (*lenna, peppers, pasta, fruits, hats*). In each case, to evaluate the noise shaping behavior, I produced undistorted halftones using the color signal distortion canceling method developed in Section 3.3.2. Section 3.3.2 showed that according to the matrix gain model, the quantization error image in the distortion canceling method is in fact the uncorrelated noise injection into the halftoning system. Therefore, I used the error image produced while halftoning the set of test images with distortion canceling schemes using the optimal filter and the Floyd-Steinberg error filter, respectively, as the noise image in the objective function of (4.4). The effective noise shaping gain (in dB) of the optimal filter over the separable Floyd-Steinberg filter may be computed as

$$NG = 10\log_{10}\left(\frac{E\left[\|\ \mathbf{b}^{(\tilde{\mathbf{v}})}_{n-fs}(\mathbf{m})\ \|^2\right]}{E\left[\|\ \mathbf{b}^{(\tilde{\mathbf{v}})}_{n-opt}(\mathbf{m})\ \|^2\right]}\right) \tag{4.35}$$

where the numerator and denominator in the argument of the log function are the objective functions computed by using (4.4) for the optimal filter and the Floyd-Steinberg filter, respectively. Sample averages were used to estimate the expectations. Table 4.1 tabulates the noise gain of the optimal filter over using a separable Floyd-Steinberg error filter.

Fig. 4.1(a) shows the *pasta* image halftoned using Floyd-Steinberg halftoning on each color plane. Fig. 4.1(b) shows a magnified view of a portion

image	Noise gain (dB)
lenna	1.8691
peppers	1.5071
fruits	0.7529
pasta	0.7161
hats	0.1442

Table 4.1: Noise gain of the optimal filter on standard test images.

of the image. Figs. 4.2(a) and 4.2(b) show the corresponding results for halftoning with the optimal error filter. The optimal filter results in less visible halftone noise. It significantly reduces color impulses when compared with scalar error diffusion using filters with scalar coefficients. The halftone noise patterns produced by conventional Floyd-Steinberg error scalar filter were significantly more visible when observed on the calibrated monitor as compared to the noise patterns produced by the optimal filter. However the proposed design procedure does not guarantee that the distribution of the color dots is the most regular possible. It must be emphasized that since the optimal filter coefficients are dependent on a particular monitor configuration, the above design process must be applied on a case-by-case basis. These figures must be viewed on a monitor, rather than printed, for a fair assessment of the image quality[1]. Since our color model is defined in a device independent color space, our preceding discussion holds for other color spaces as well. For example, if we are working in a Cyan Magenta Yellow (CMY) color space (for a printing application), then we can convert CMY into corresponding CIE XYZ coordinates [44] and then into the opponent representation. Thus, using a new color

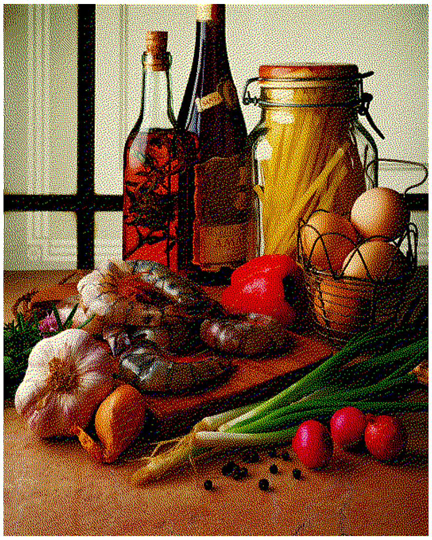
---

[1] Sample images are available at http://signal.ece.utexas.edu/~damera/col-vec.html

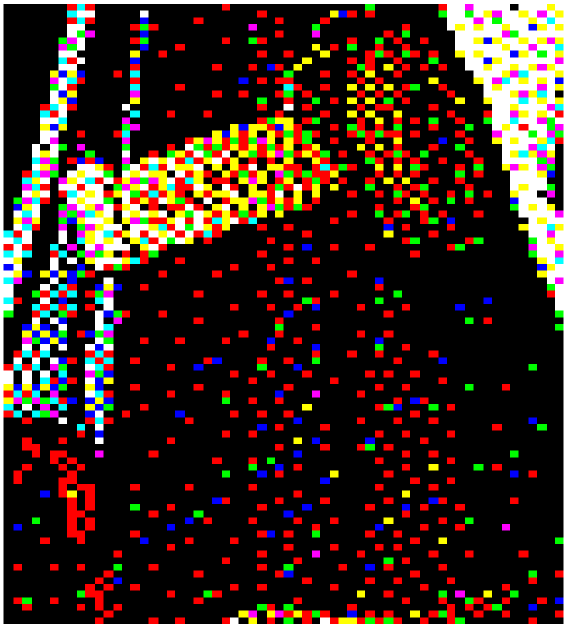
transformation matrix $\tilde{\mathbf{T}}$, the optimal filter for this case can be calculated using the method described in this section.

4.4 Conclusion

This chapter formulates the error filter design problem as the minimization of a quadratic objective function formed by weighting the shaped noise in the halftone with a linear model for the human visual system. The linear model for the human visual system was based on the Poirson and Wandell [36] opponent representation which models human sensitivity to color patterns. Using such a model enables one to tune the error filter design for a specific calibrated display device. This is because the excitations of the cone photoreceptors depends on the specific spectral profile of the color being viewed. The spectral statistics of the colors are taken into account by means of a calibration transformation $\tilde{\mathbf{T}}$. The spatial sensitivities of individual opponent pathways is incorporated by a linear filter $\tilde{\mathbf{d}}(\mathbf{m})$. The optimal solution was shown to satisfy a matrix version of the Yule-Walker equations [40]. A steepest descent approach was used to solve the system of equations and explore the design space for more regular halftone dot distributions. The designed error filter outperforms separable solutions like the method using the popular Floyd-Steinberg error filter in terms of color halftone noise visibility both qualitatively and quantitatively. However, it was noticed by inspection that the dot spacing between color dots was not visually optimum.

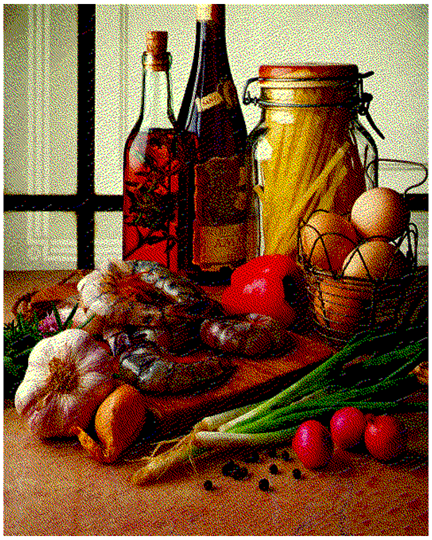


(a) *pasta* halftoned with Floyd-Steinberg filter.

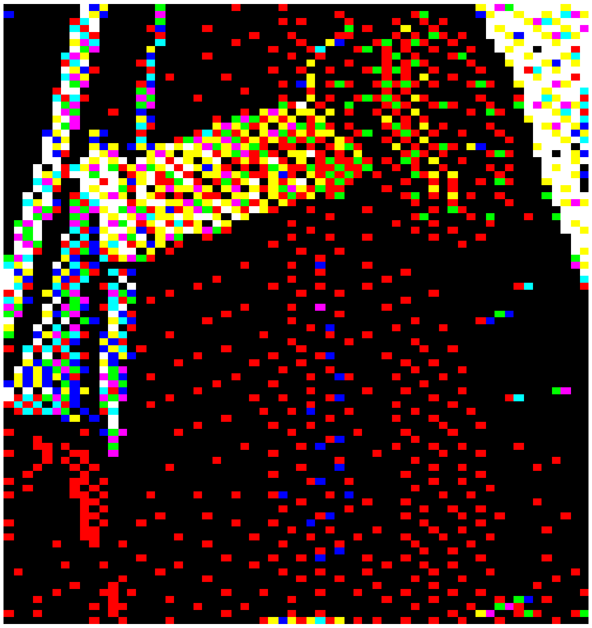


(b) Magnified portion of halftone.

Figure 4.1: Performance of the separable Floyd-Steinberg filter (images available at `http://signal.ece.utexas.edu/~damera/col-vec.html`).



(a) *pasta* halftoned with optimal filter.



(b) Magnified portion of halftone.

Figure 4.2: Performance of the optimal filter (images available at `http://signal.ece.utexas.edu/~damera/col-vec.html`).

Chapter 5

Clustered Dot Error Diffusion Halftoning

5.1 Introduction

This chapter exploits the duality between a multifilter and a block filter as discussed in Section 2.3 to design algorithms for FM halftoning with canonical dot shapes and AM-FM halftoning with user defined dot shape and dot size modulation. An overview of current AM-FM halftoning methods is provided in Section 1.2.4.

This chapter generalizes conventional error diffusion halftoning to produce FM halftones with user controlled dot size and shape. Unlike conventional FM halftones, dot clusters of more than one pixel are allowed. The generated FM halftones can be designed to have very low dot size/shape variation, while their spacing is modulated depending on the underlying grayscale image. This is different from the AM-FM methods discussed above that have variable dot shape/size. FM halftones with clustered dots may be used to provide robust printed-dots over all graylevel values. The AM-FM halftoning methods discussed in Section 1.2.4 do not have control over the dot shape, although the dot size may be controlled to some extent. Furthermore, the method presented in this chapter is single pass unlike [27] and uses the conventional raster scan unlike [9, 26]. Unlike the previous methods, the method presented in this chapter has user tunable sharpness control and multiresolution embedding capability inbuilt into the framework. Further, with minor modifications, I show how AM-FM halftones with/without user controlled amplitude modulation may be produced with this framework. Finally, I show that the framework may be implemented efficiently, since it can take advantage of a parallel polyphase filterbank implementation. I refer to the framework as block error diffusion since the notion of a pixel in conventional error diffusion is replaced with the concept of a pixel block in our frame work. The quantization error at each pixel in the pixel block is diffused to suitably selected pixels in neighboring blocks in suitably selected proportions. Thus, an entire block of quantization error is diffused.

Section 5.2 introduces the basic framework of block error diffusion by formulating the underlying equations. Section 5.3 demonstrates the effect of designing the block error filter from well known scalar error filter prototypes. I also discuss how FM halftones with user-controlled dot shape and size may be produced. Section 5.4 shows how to modify block error diffusion to incorporate sharpness control and embedded multiresolution rendering capability. I extend block error diffusion to generate AM-FM halftones with/without user controlled AM dots in Section 5.5. Section 5.6 derives an efficient parallel implementation for block error diffusion. This implementation is more efficient than the general parallel implementation of a multifilter discussed in Section 2.4 due to the special structure of the designed block error filters. Section 5.7 concludes the chapter by summarizing the contributions.

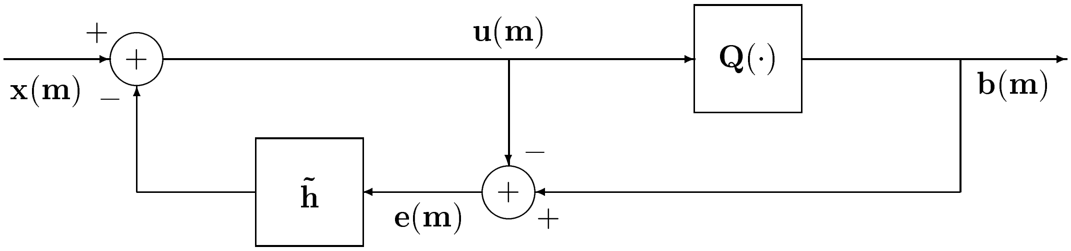


Figure 5.1: System block diagrams for block error diffusion halftoning where $\tilde{\mathbf{h}}$ represents a fixed 2-D nonseparable FIR error filter with matrix valued coefficients. The vector $\mathbf{m}$ represents the 2-D index (m_1, m_2).

5.2 Block Error Diffusion

Fig. 5.1 shows the block diagram illustrating block error diffusion. Although the basic diagram resembles conventional error diffusion halftoning, there are key differences. The input is an $N \times N$ block of pixels (called a pixel-block) as opposed to a single pixel in conventional error diffusion. Consider each block to be ordered into an N^2-element vector as discussed in Section 2.2.

The quantizer output for each pixel in a pixel-block is exactly one element from the discrete set $\mathcal{O} = \{-1, 1\}$. Here, -1 represents black and $+1$ represents white. I quantize each pixel-block using a scalar quantizer. The quantizer $\mathbf{Q}(\cdot)$ is defined by

$$\mathbf{Q}(\mathbf{u}) = \begin{pmatrix} Q(u_1) \\ Q(u_2) \\ \vdots \\ Q(u_{N^2}) \end{pmatrix} \tag{5.1}$$

$$Q(u_i) = \begin{cases} 1 & u_i \geq 0 \\ -1 & u_i < 0 \end{cases} \tag{5.2}$$

The filter in the feedback loop has matrix-valued coefficients. The filter operates on the quantization error sequence $\mathbf{e}(\mathbf{m})$ to produce the feedback

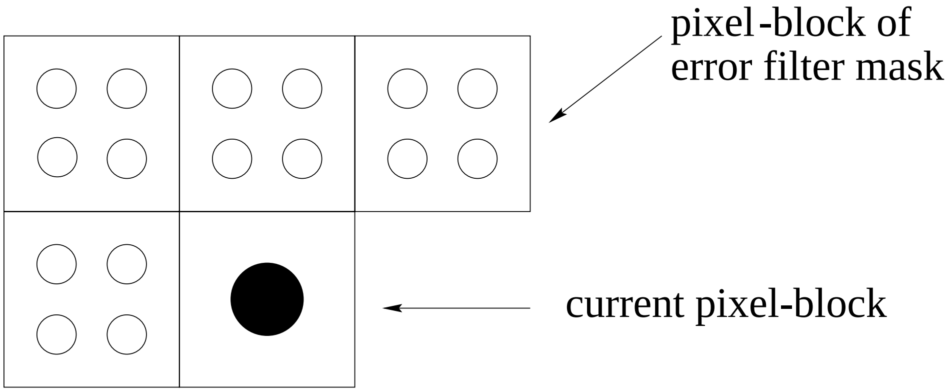


Figure 5.2: Block error filter operating on pixel-blocks of 2 × 2 pixels. The shaded circle indicates the current pixel-block. The unfilled circles indicate the error image pixels underlying the block filter mask. The pixels in the output pixel-block are computed using 4 linear combinations of all 16 error pixels within the error filter mask.

signal sequence $\mathbf{f}(\mathbf{m})$ according to

$$\mathbf{f}(\mathbf{m}) = \sum_{\mathbf{k}\in\mathcal{S}} \tilde{\mathbf{h}}(\mathbf{k})\mathbf{e}(\mathbf{m}-\mathbf{k}) \tag{5.3}$$

where $\mathbf{m}$, $\mathbf{k}$ are two-dimensional index vectors, $\tilde{\mathbf{h}}(\cdot)$ is an $N^2 \times N^2$ matrix-valued sequence, and $\mathcal{S}$ is the filter support. As before, I assume a four-tap filter support defined by (horizontal, vertical) offsets to the current pixel being processed as $\mathcal{S} = \{(0,1),(1,0),(1,1),(1,-1)\}$ unless specified otherwise.

In terms of block filtering, the operation described by (5.3) can be described with the help of Fig. 5.2 which illustrates a block error filter operating on pixel-blocks of 2 × 2 pixels. The output pixel-block is computed by forming 4 different linear combinations of all pixels in the pixel-block mask which consists of 16 pixels. Each linear combination produces a single output pixel of the output pixel-block.

Section 5.3 shows how the matrix-valued coefficients of error filter may be designed to promote user defined minority pixel clustering in the generated halftone image.

5.3 FM halftoning via block error diffusion

The block error filter in the feedback loop governs how quantization error is diffused to the neighboring pixel-blocks. For conventional error diffusion, one only needs to decide how much of the quantization error is to be diffused to each neighboring pixel under the constraint that all of the quantization error be diffused. The case of block error diffusion also requires that all of the quantization error be diffused. This imposes the constraints

$$\tilde{\mathbf{\Gamma}}\mathbf{1} = \mathbf{1} \tag{5.4}$$

$$\tilde{\mathbf{\Gamma}} \geq 0 \tag{5.5}$$

where $\mathbf{1}$ represents an $N^2M^2 \times 1$ column vector with all of its elements equal to one. These conditions correspond to the assertion that the elements of the matrix-valued error filter coefficients be non-negative so that each row sums to unity.

5.3.1 Error filter design

In designing the coefficients $\tilde{\mathbf{\Gamma}}$, I start with the coefficients of a conventional error filter and map them into corresponding block filters. By starting with a scalar filter with the same support as the multifilter or block filter, and by representing its coefficients by the row vector $\tilde{\gamma}$, where

$$\tilde{\gamma} = \Big[g(0) \mid g(1) \mid \ldots \mid g(M^2-1) \Big] \tag{5.6}$$

a multifilter $\tilde{\mathbf{\Gamma}}$ may be derived from $\tilde{\gamma}$ as follows

$$\tilde{\mathbf{\Gamma}} = \tilde{\gamma} \bigotimes \tilde{\mathbf{D}} \tag{5.7}$$

where $\otimes$ denotes the Kronecker product operation. $\tilde{\mathbf{D}}$ is an $N^2 \times N^2$ matrix which we call the diffusion matrix. Since the element of $\tilde{\gamma}$ are the coefficients of a conventional error filter they are non-negative and sum to one. Thus to satisfy the constraints imposed by equation (5.4), the diffusion matrix must satisfy the constraints

$$\tilde{\mathbf{D}}\mathbf{1} = \mathbf{1} \tag{5.8}$$

$$\tilde{\mathbf{D}} \geq 0 \tag{5.9}$$

where again $\mathbf{1}$ represents an $N^2 \times 1$ column vector with all its elements equal to one. Thus, by imposing structure on $\tilde{\mathbf{\Gamma}}$, one only needs to design the $N^2 \times N^2$ diffusion matrix $\tilde{\mathbf{D}}$.

I will now show that the decomposition of (5.7) is a natural and intuitive way of designing suitable error filters to generate FM halftones via block error diffusion. The physical meaning of deriving the block filter from a given conventional error filter via (5.7) is that the quantization error incurred at the current pixel block is diffused to the neighboring pixel-blocks in the same proportions that a conventional error filter diffuses error to its neighboring pixels. The diffusion matrix $\tilde{\mathbf{D}}$ governs the proportions in which errors are to be distributed within the pixels of a block. According to our proposed structure, these proportions are constant independent of the relative position of the pixel-blocks to which errors are diffused. This enforces a local isotropy constraint. The constraints on the diffusion matrix simply indicate that *all*

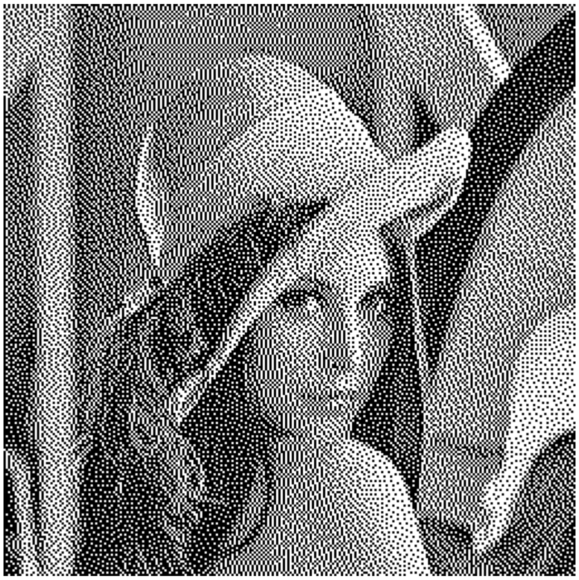


Figure 5.3: Halftone generated by pixel replication induced block clustering. Here the original image is filtered (to prevent aliasing) and downsampled. The downsampled image is then halftoned using conventional error diffusion. Pixel clustering is then induced by replicating each pixel to form a pixel-block. Note the loss of high frequency information and blurred appearance.

of the quantization error that is diffused to a pixel block must be diffused among pixels that make up the block. Thus the pixel-blocks in the block-error diffusion framework are made to behave like pixels in conventional error diffusion and the block errors are diffused in much the same way as pixel errors in conventional error diffusion.

5.3.2 FM halftoning

This subsection will attempt to produce FM-halftones with dot clusters greater than one pixel. One method of achieving dot-clustering would be to halftone a downsampled version of the grayscale image and then using pixel replication to get a halftone of the same size as the grayscale image. For 2×2 minority pixel dot clusters, one could filter the original grayscale image with a halfband filter, downsample the original grayscale image by retaining every alternate sample in the horizontal and vertical directions, halftone the downsampled grayscale image, and interpolate it to the resolution of the original grayscale image by pixel replication. This process is identical to halftoning the filtered image after replicating the leftmost uppermost sample in each block to all samples and using the identity diffusion matrix $\tilde{\mathbf{D}} = \tilde{\mathbf{I}}_{4\times 4}$. Fig. 5.3 shows a sample halftone obtained using a halfband pre-filter followed by block error diffusion with the identity diffusion matrix. Clearly, the spatial resolution of the image suffers due to the pixel replication and pre-filtering.

My approach to FM halftoning relies on forming minority pixel dot clusters by diffusing the quantization error from each pixel block equally to all of the samples within a pixel block. The error diffused to each block within the block error filter mask will, however, be unequal since it is governed by the corresponding conventional error filter coefficients $\tilde{\gamma}$. Thus, I use the diffusion matrix

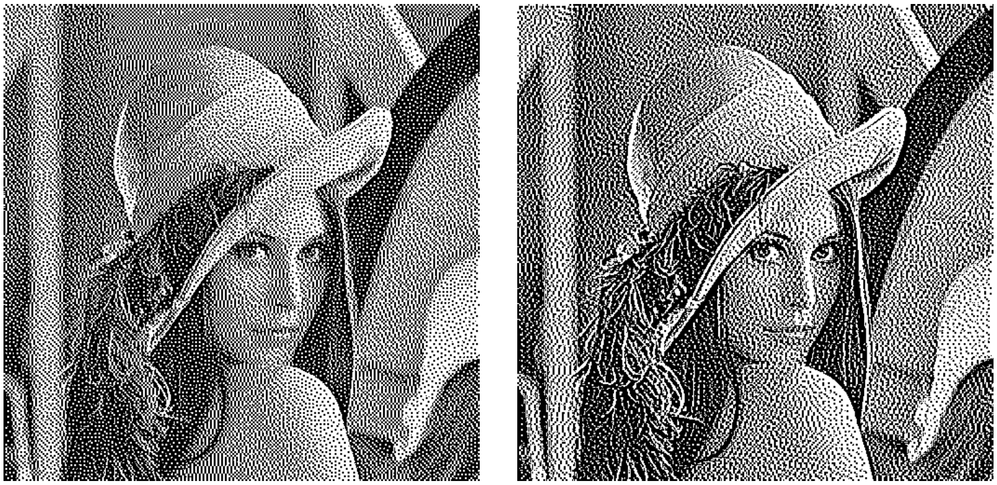
$$\tilde{\mathbf{D}} = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{5.10}$$

The motivation for this is that the error at any sample within the current pixel-block that is diffused to an adjacent pixel-block will be spread to all the samples within the pixel-block equally. This means that the quantization decisions of all pixels within the modified pixel-block will be biased in the same direction. Intuitively, this should result in the halftoned samples of that pixel-block organizing themselves into a pixel-block cluster.

Figs. 5.4(a) and (b) show sample halftones obtained using block error diffusion with the above diffusion matrix and using $\tilde{\gamma} = \left[\frac{1}{16}\ \frac{5}{16}\ \frac{3}{16}\ \frac{7}{16}\right]$ and $\tilde{\gamma} = \left[\frac{1}{48}\ \frac{3}{48}\ \frac{5}{48}\ \frac{3}{48}\ \frac{1}{48}\ \frac{3}{48}\ \frac{5}{48}\ \frac{7}{48}\ \frac{5}{48}\ \frac{3}{48}\ \frac{5}{48}\ \frac{7}{48}\right]$, respectively. This corresponds to the well known Floyd-Steinberg [2] and Jarvis [11] error filters, respectively, with support shown in Fig. 1.4. For the rest of the chapter, I fix $\tilde{\gamma} = \left[\frac{1}{16}\ \frac{5}{16}\ \frac{3}{16}\ \frac{7}{16}\right]$. There is no need to use a pre-filter to prevent spatial aliasing. By visual inspection of Fig. 5.4 the spatial resolution of the grayscale image is not compromised. The halftones even exhibits sharpening, characteristic of conventional Floyd-Steinberg and Jarvis error diffusion [19]. From visual inspection, the dots are clustered into 2×2 blocks.

Using the method described above, it is possible to cluster the halftone dots into user defined shapes and sizes. Halftones with dot clusters of 3×3 and 4×4 are produced using 9×9 and 16×16 diffusion matrices having all of their elements equal to $\frac{1}{9}$ and $\frac{1}{16}$, respectively. Figs. 5.5(a) and (b) show halftones with non-rectangular dot clusters.

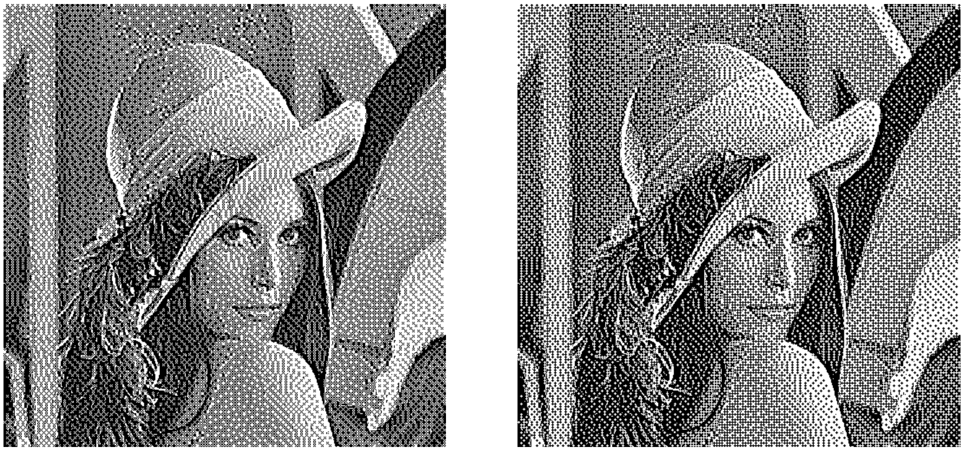
Figs. 5.5(a) and (b) were produced using the diffusion matrix $\frac{1}{9}\tilde{\mathbf{1}}_{9\times 9}$ similar to square block clustered dot FM halftones described earlier. However, I quantize the minority pixel-blocks by replacing them with the desired dot shape, while the majority pixel-blocks are quantized as usual by using (5.2). Figs. 5.6(a) and 5.6(b) show the pixels within a pixel-block (shaded) that are part of the dot shapes corresponding to the halftones of Figs. 5.5(a) and (b). The shaded pixels are to be interpreted as having a value -1, while the unshaded pixels are to be interpreted as having a value $+1$. The quantization error is computed at each pixel-block location as the vector difference between



(a) Floyd-Steinberg (b)Jarvis

Figure 5.4: Block error diffusion with block error filters derived from conventional Floyd-Steinberg and Jarvis filters. Note the improved performance over pixel replication induced block clustering.

the output shape (if we are quantizing a minority pixel) or the standard output pixel-block (if we are quantizing a majority pixel). Thus, at each pixel-block location one needs to determine if the current pixel-block to be quantized will form a minority pixel-block or a majority pixel-block. This may be estimated by comparing the majority pixel type ($+1$ or -1) in the current pixel-block quantized at mid-gray against the majority pixel type in the corresponding pixel-block in the original grayscale image that is quantized at mid-gray. If the two are not equal, then the pixel-block is a minority block and the output pixel-block is replaced with the desired dot-shape. If the two are equal, then the current pixel-block is a majority pixel-block and the output pixel block is simply the quantization of the current pixel-block at mid-gray.



(a) "multiply" dots (b)"plus" dots

Figure 5.5: Block error diffused halftones with user controlled dot shapes.

5.4 Enhancements to FM-halftoning via block error diffusion

This section shows that one may modify block error diffusion to produce

1. FM halftones with user-defined sharpness (Section 5.4.1) and
2. Embedded multiresolution FM halftones (Section 5.4.2).

5.4.1 FM halftoning with user-defined sharpness control

Eschbach and Knox [50] show that the sharpness of a halftone may be changed by adding a fraction L of the input image to the quantizer input. Kite, Evans and Bovik [19] showed that conventional error diffusion halftoning modified as described above is equivalent to pre-filtering the original image with a tunable filter. I extend these results to block error diffusion, and hence, show that FM halftones with user controlled sharpness may be be generated with low complexity using the framework described in Section 5.2.

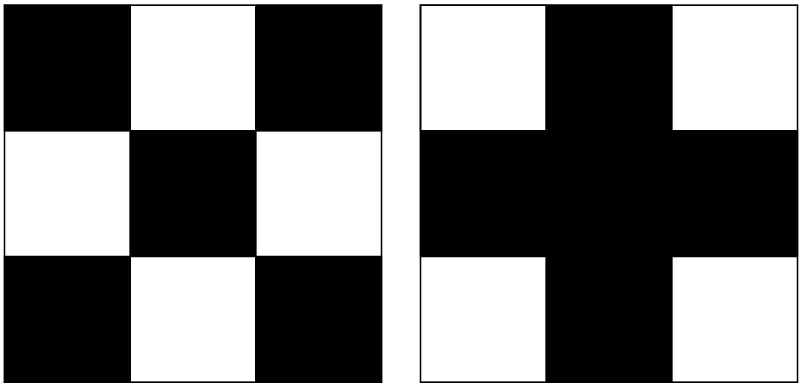

(a) "multiply" dot (b)"plus" dot

Figure 5.6: FM halftone dot shapes. The shaded pixels indicate the pixels in the pixel-blocks that are part of the halftone dot shape.

Fig. 5.7 shows the block diagram for block error diffusion with user-defined sharpness control. Here, I add a linear transformation of the pixels in the input pixel-block to the input to the quantizer. This section will show that this approach is equivalent to pre-filtering the original image with a tunable block filter $\tilde{\mathbf{g}}$ as shown in Fig. 5.8. This is similar to the claims of Proposition 1 of Chapter 3. However, I do not need to invoke the matrix gain model for the following discussion.

By analyzing Fig. 5.8, the input to the quantizer $\mathbf{u}(\mathbf{m})$ in the z-domain is

$$\mathbf{U}(\mathbf{z}) = \tilde{\mathbf{G}}(\mathbf{z})\mathbf{X}(\mathbf{z}) - \tilde{\mathbf{H}}(\mathbf{z})\mathbf{E}(\mathbf{z}) \tag{5.11}$$

$$\mathbf{E}(\mathbf{z}) = \mathbf{B}(\mathbf{z}) - \mathbf{U}(\mathbf{z}) \tag{5.12}$$

From (5.11) and (5.12),

$$\mathbf{E}(\mathbf{z}) = [\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z})]^{-1}[\mathbf{B}(\mathbf{z}) - \tilde{\mathbf{G}}(\mathbf{z})\mathbf{X}(\mathbf{z})] \tag{5.13}$$

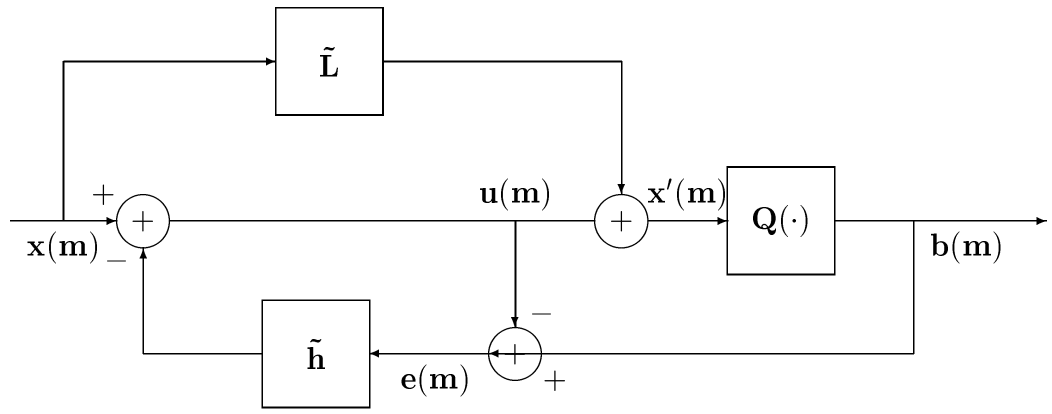


Figure 5.7: System block diagram for modified block error diffusion halftoning. $\tilde{\mathbf{L}}$ represents a constant linear transformation.

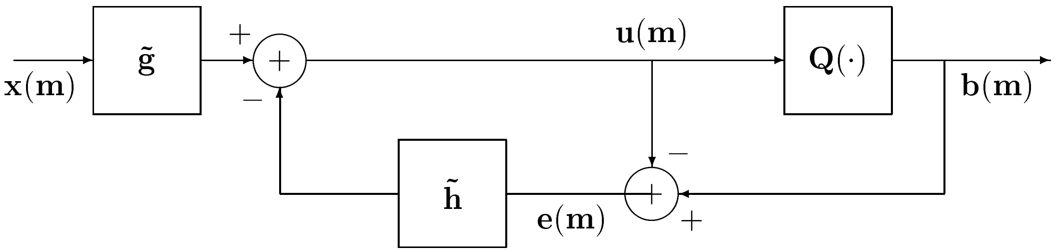


Figure 5.8: System block diagram for block error diffusion halftoning with a fixed pre-filter $\tilde{\mathbf{g}}$ with matrix valued coefficients.

Substituting for $\mathbf{E(z)}$ into (5.11) yields

$$\mathbf{U(z)} = [\tilde{\mathbf{I}} + \tilde{\mathbf{H}}\mathbf{(z)}(\tilde{\mathbf{I}} - \tilde{\mathbf{H}}\mathbf{(z)})^{-1}]\tilde{\mathbf{G}}\mathbf{(z)}\mathbf{X(z)} - \tilde{\mathbf{H}}\mathbf{(z)}(\tilde{\mathbf{I}} - \tilde{\mathbf{H}}\mathbf{(z)})^{-1}\mathbf{B(z)} \tag{5.14}$$

By analyzing Fig. 5.7,

$$\mathbf{U(z)} = \mathbf{X(z)} - \tilde{\mathbf{H}}\mathbf{(z)}\mathbf{E(z)} \tag{5.15}$$

$$\mathbf{E(z)} = \mathbf{B(z)} - \mathbf{U(z)} \tag{5.16}$$

From (5.15) and (5.16),

$$\mathbf{E}(\mathbf{z}) = [\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z})]^{-1}[\mathbf{B}(\mathbf{z}) - \mathbf{X}(\mathbf{z})] \tag{5.17}$$

Also, since

$$\mathbf{X}'(\mathbf{z}) = [\tilde{\mathbf{I}} + \tilde{\mathbf{L}}]\mathbf{X}(\mathbf{z}) - \tilde{\mathbf{H}}(\mathbf{z})\mathbf{E}(\mathbf{z}) \tag{5.18}$$

By substituting (5.17) into (5.18)

$$\mathbf{X}'(\mathbf{z}) = [(\tilde{\mathbf{I}} + \tilde{\mathbf{L}}) + \tilde{\mathbf{H}}(\mathbf{z})[\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z})]^{-1}]\mathbf{X}(\mathbf{z}) - \tilde{\mathbf{H}}(\mathbf{z})(\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z}))^{-1}\mathbf{B}(\mathbf{z}) \tag{5.19}$$

By comparing (5.14) and (5.19), we conclude that Figs. 5.8 and 5.7 are equivalent in the sense that they have the same quantizer input and hence output if

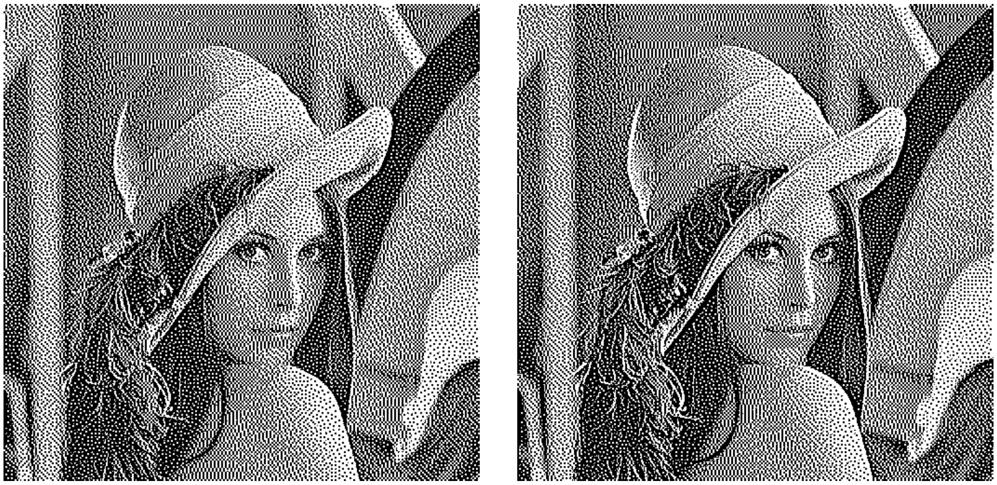
$$[(\tilde{\mathbf{I}} + \tilde{\mathbf{L}}) + \tilde{\mathbf{H}}(\mathbf{z})[\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z})]^{-1}] = [\tilde{\mathbf{I}} + \tilde{\mathbf{H}}(\mathbf{z})(\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z}))^{-1}]\tilde{\mathbf{G}}(\mathbf{z}) \tag{5.20}$$

Thus, if $\tilde{\mathbf{G}}(\mathbf{z}) = [\tilde{\mathbf{I}} + \tilde{\mathbf{H}}(\mathbf{z})(\tilde{\mathbf{I}} - \tilde{\mathbf{H}}(\mathbf{z}))^{-1}]^{-1}\tilde{\mathbf{L}} + \tilde{\mathbf{I}}$, Figs. 5.8 and 5.7 are equivalent. This means that the pre-filter $\tilde{\mathbf{g}}$ is tunable by varying $\tilde{\mathbf{L}}$. This produces varying sharpness in the output halftone. For reduced complexity, $\tilde{\mathbf{L}}$ is taken to be a diagonal matrix of the form $\tilde{\mathbf{L}} = l\tilde{\mathbf{I}}$. Figs. 5.9(a) and (b) show the effect of tuning the parameter l on the sharpness of the resulting FM-halftones. As l increases, the sharpness of the halftone increases.

5.4.2 Embedded multiresolution FM halftoning

Conventional embedded multiresolution halftoning [49] generates halftones at successively lower resolutions from a given grayscale image x. The goal of embedded multiresolution halftoning is to generate the embedded set $\mathcal{E}$ of halftones b_j

$$\mathcal{E} = \left\{ b^{(j)}, j \geq 0 : b^{(j+1)} = b^{(j)} \downarrow_M \right\} \tag{5.21}$$



(a) $l = 0.2$ (b) $l = 0.6$

Figure 5.9: Block error diffused halftones with user controlled sharpness.

where $\downarrow_M$ denotes downsampling by M. Wong [49] generates the set of halftones $\mathcal{E}$ from x by adapting the error filter. Suitably downsampled versions of x are halftoned subject to the constraint that some of the pixels of the output at that resolution were already determined by the halftone at a lower resolution by the condition $b^{(j+1)} = b^{(j)} \downarrow_M$.

I embed low-resolution halftones with smaller dot clusters into higher resolution halftones with larger dot clusters. The reduction in dot cluster size compensates for the loss in spatial resolution due to subsampling. The embedding may be accomplished by designing the diffusion matrix directly. Thus, block error diffusion using the new diffusion matrix will yield the desired embedding. Without loss of generality, I will assume a separable subsampling matrix Λ of the form

$$\Lambda = \begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix} \tag{5.22}$$

I will also assume without loss of generality that $M = 2$.

Let $\mathcal{S}_{higher} = diag(\tilde{\mathbf{D}}_{higher})$ represent the ordered set of the diagonal elements of $\tilde{\mathbf{D}}_{higher}$ ordered according to increasing rows, where $\tilde{\mathbf{D}}_{higher}$ represents the diffusion coefficients of the halftone at a higher resolution. The location of the non-zero elements of $\mathcal{S}_{higher}$ indicates the shape of the higher resolution halftone dot clusters. The embedding process requires that the dot shape of the halftone at the lower resolution be such that, when upsampled to the higher resolution, the location of the non-zero elements of $\mathcal{S}_{lower}$ denoted by $NZ((\uparrow 2)\mathcal{S}_{lower})$ is a proper subset of $NZ(\mathcal{S}_{higher})$. Thus

$$NZ((\uparrow 2)\mathcal{S}_{lower}) \subset NZ(\mathcal{S}_{higher}) \tag{5.23}$$

This requirement is illustrated in Fig. 5.10 where a 2×2 dot cluster is embedded into a 4×4 dot cluster.

The diffusion matrix that will perform the embedding is designed as follows. First the diffusion matrices of the multiresolution halftones are designed subject to the constraint given by (5.23). The embedding matrix has the same dimensions as the diffusion matrix of the highest resolution halftone. The highest resolution halftone pixel locations that are also present in the lower resolution halftone are considered. The quantization error at these pixels is *not* diffused equally to all of the pixels in the higher resolution dot shape as with usual FM halftoning. Instead, the diffusion matrix does not have all of its elements equal. The quantization error at a given pixel location is only diffused equally to all of the pixels that will be included in the lowest resolution halftone containing the given pixel. Thus, the quantization error at the pixel location *LMH* in Fig. 5.10 is exclusively diffused to pixels of the neighboring pixel-blocks that are marked *L*. The usual constraints on the diffusion matrix

LMH	H	MH	H
H	H	H	H
MH	H	MH	H
H	H	H	H

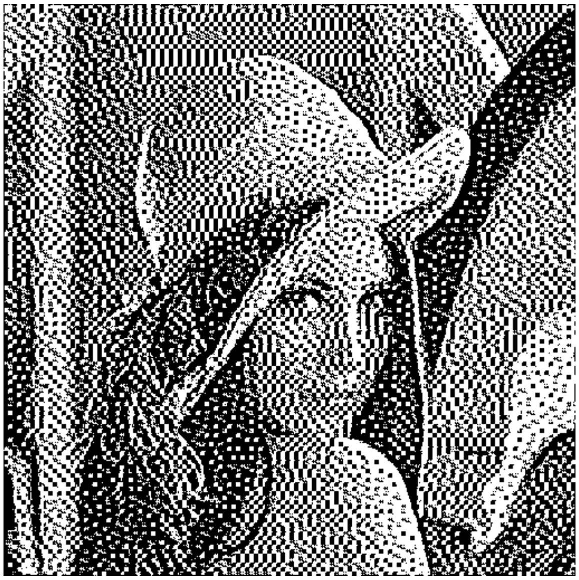
Figure 5.10: Halftone dot shape embedding. Pixels belonging to the highest resolution halftone (4 × 4 pixel dot shape) are indicated with the letter H. Pixels belonging to the medium resolution halftone (2×2 pixel dot shape) are indicated with the letter M. Pixels belonging to the lowest resolution halftone (1 × 1 pixel dot shape) are indicated with the letter L. Note that a pixel belonging to a lower resolution dot shape also belongs to the higher resolution dot shapes.

as defined by (5.8) still apply. Thus, when the output halftone is subsampled, the resulting dots will again cluster together into dots of the lower resolution halftone.

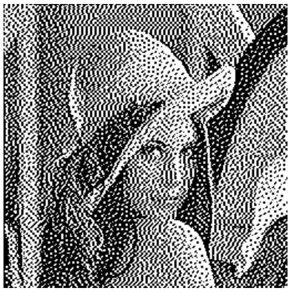
An example of the embedding process is illustrated in Fig. 5.10. In this case

$$\tilde{\mathbf{D}}_3 = \frac{1}{16}\left(\tilde{\mathbf{1}}_{16\times 16}\right) \quad \tilde{\mathbf{D}}_2 = \tfrac{1}{4}\left(\tilde{\mathbf{1}}_{4\times 4}\right) \quad \tilde{\mathbf{D}}_1 = (1) \tag{5.24}$$

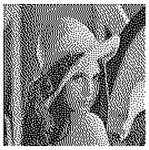
where $\tilde{\mathbf{1}}_{n\times n}$ denotes an $n \times n$ matrix with each element equal to one. The



(a) 512 × 512 halftone with 4 × 4 dot clusters.



(b)256 × 256 halftone with 2 × 2 dot clusters.



(c) 128 × 128 halftone with 1 × 1 dot clusters.

Figure 5.11: FM halftones embedded in halftones with a larger dot size.

diffusion matrix that performs the embedding is

$$\tilde{\mathbf{D}}_{embed} = \begin{pmatrix} \mathbf{l} & \mathbf{h} & \mathbf{m} & \mathbf{h} & \mathbf{h} & \mathbf{h} & \mathbf{h} & \mathbf{h} & \mathbf{m} & \mathbf{h} & \mathbf{m} & \mathbf{h} & \mathbf{h} & \mathbf{h} & \mathbf{h} & \mathbf{h} \end{pmatrix}^T \tag{5.25}$$

where

$$\mathbf{l} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \tag{5.26}$$

$$\mathbf{m} = \begin{pmatrix} 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \tag{5.27}$$

$$\mathbf{h} = \begin{pmatrix} 0 & \frac{1}{12} & 0 & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & 0 & \frac{1}{12} & 0 & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} & \frac{1}{12} \end{pmatrix}^T \tag{5.28}$$

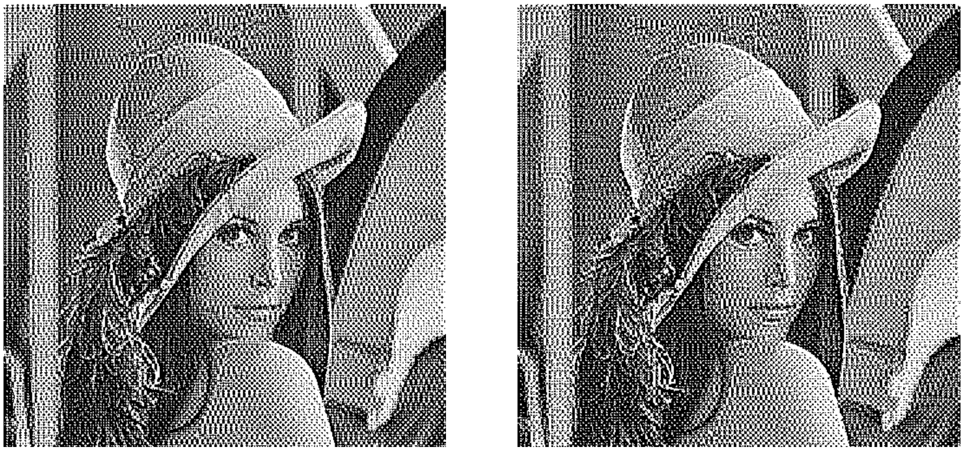
Fig. 5.11 shows the effect of halftoning using the diffusion matrix $\tilde{\mathbf{D}}_{embed}$. Fig. 5.11(a) shows the highest resolution halftone with 4×4 block clusters, while Figs. 5.11(b) and (c) show successively downsampled versions of (a) by the downsampling matrix Λ. Typically, the lowest resolution thumbnail image could be displayed on a monitor, and the higher resolution halftones could be used to print the higher resolution versions of the thumbnail image.

5.5 AM-FM halftoning via block error diffusion

With a given block size, two different methods may be employed to produce AM-FM halftones by allowing user control of dot shape and dot size, respectively, as discussed next in Sections 5.5.1 and 5.5.2.

5.5.1 AM-FM halftoning with variable dot shape

AM-FM halftoning varies the shape/size of the dot clusters are varied depending on the graylevel value to be rendered. The dot shape is varied with user control. This is similar to amplitude modulation in communication theory, in



(a) AM dot modulation with array $\mathbf{A} = \begin{pmatrix} 7 & 4 & 8 \\ 2 & 0 & 3 \\ 5 & 1 & 6 \end{pmatrix}$

(b)AM dot modulation with array $\mathbf{B} = \begin{pmatrix} 5 & 7 & 4 \\ 2 & 3 & 8 \\ 0 & 1 & 6 \end{pmatrix}$

Figure 5.12: Block error diffused AM-FM halftones. The AM dot modulation is induced by a user defined dither mask. However the AM dots are distributed as with usual FM halftoning.

which the amplitude of the carrier sinusoid is modulated in accordance with a message signal [51].

An array of thresholds is used to introduce AM modulation while quantizing each pixel-block. Thus the shape of the dot at each pixel-block is not predetermined as in the case of FM halftoning, but depends on the graylevel values of the pixels in the pixel-block. The AM dot clusters must be distributed using FM. This can be accomplished using the block error diffusion framework.

The key to producing user-controlled AM dot clusters is to allow the user to define a clustered-dot dither mask as the AM modulating array. Thus the quantizer function of (5.2) is modified as follows.

$$Q_m^+(u_i) = \begin{cases} 1 & u_i \geq T_i, \\ -1 & u_i < T_i \end{cases} \tag{5.29}$$

$$Q_m^-(u_i) = \begin{cases} 1 & u_i \geq -T_i, \\ -1 & u_i < -T_i \end{cases} \tag{5.30}$$

where $Q_m^+(\cdot)$ and $Q_m^-(\cdot)$ are used depending on whether the minority pixel-block is white $(+1)$ or black (-1) respectively. T_i is a user defined threshold. The AM dot modulation is only applied in pixel-block locations where a minority dot cluster is to be formed. Using the quantizer function defined in (5.29) on majority pixel-block locations results in undesirable dot modulation at the majority pixel-block locations, which causes periodic artifacts similar to those observed with screened halftones. In order to apply the dot modulation selectively, one needs to determine if the current pixel block will give rise to a minority pixel-block cluster. This may be estimated by comparing the majority pixel type $(+1$ or $-1)$ in a pixel-block quantized at mid-gray against the majority pixel type in the corresponding pixel-block in the original gray-scale image that is quantized at mid-gray. If the two are not equal, then the pixel-block is a minority block and the current pixel-block is requantized using (5.29). If the two are equal, then the current pixel-block is a majority pixel-block, and no requantization is necessary. In either case, the quantization error is diffused as with usual FM halftoning as described in Section 5.3.

Figs. 5.12(a) and (b) show example images halftoned using the method described above with two different AM modulation threshold arrays. The desired variation in the minority pixel dot clusters that are themselves frequency modulated is exhibited. The limit cycle artifacts above the hat region of the

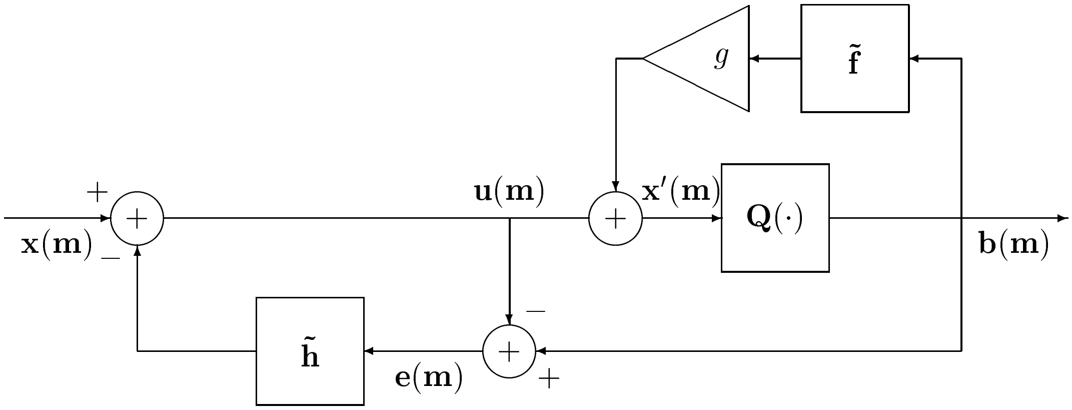
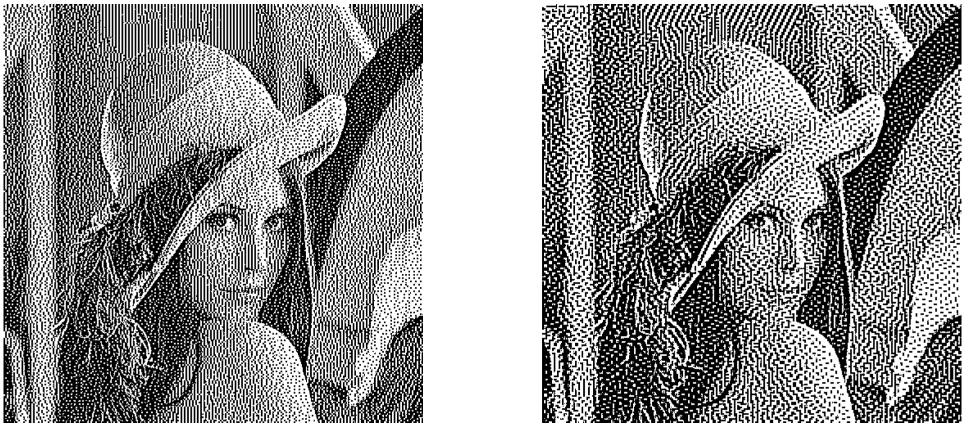


Figure 5.13: Block green noise error diffusion halftoning with hysteresis 2-D block FIR filter $\tilde{\mathbf{f}}$, and scalar gain g.

images are a characteristic of Floyd-Steinberg error filter which was used to derive the block error filters we used. By viewing the images from a distance, one can affirm that the gray tones of the original image are indeed reproduced by block AM-FM error diffusion halftoning.

5.5.2 AM-FM halftoning with user defined dot size

A method to generate AM-FM halftones with user-defined dot size but uncontrollable dot shape is an extension of conventional error diffusion with output dependent feedback due to Levien [9] to the case of block error diffusion. The resulting structure is a blocked version of green noise error diffusion. Fig. 5.13 shows the block diagram for block green noise error diffusion. Here a linear transformation of the output vector is filtered using a multifilter and fed back into the input to the quantizer. A scaling factor applied to the feedback from the output determines the size of the dots as in [25]. This promotes



(a) AM dot modulation with $g = 0.2$ (b)AM dot modulation with $g = 0.75$

Figure 5.14: Block error diffused AM-FM halftones. The AM dot modulation is induced by output dependent feedback. This results in smaller pixel clusters further clustering together to form "super" pixel-block clusters. The user cannot control the AM dot modulation although control of the gross dot size is possible.

the clustering of dot pixel-blocks into *super pixel-blocks* which are a collection of minority dot pixel-blocks. The scaling parameter controls the pixel-block clustering. The feedback hysteresis filter is designed by deriving it from a conventional scalar filter via methods described in Section 5.3. Fig. 5.14(a) and Fig. 5.14(b) show sample halftones halftoned with scaling parameter $g = 0.2$ and $g = 0.75$, respectively. The increased dot clustering with increase in g is evident.

5.6 Efficient parallel implementation of block error diffusion

In this section, I show that an error filter with matrix-valued coefficients has a parallel implementation, which can increase throughput by a factor N^2 for a block size of $N \times N$. A filter with matrix valued coefficients can be implemented with *conventional* filtering operations applied in parallel to each component of the vector sequence being filtered [32][1].

While it is possible to derive the parallel implementation directly from space domain block filtering considerations as discussed in [32] so as to explicitly incorporate the relationship between multifiltering and block filtering I derive the parallel implementation from a multifilter perspective only.

By converting (5.3) to the frequency domain,

$$\begin{pmatrix} F_0(\mathbf{z}) \\ F_1(\mathbf{z}) \\ \vdots \\ F_{N^2-1}(\mathbf{z}) \end{pmatrix} = \begin{pmatrix} H_0^{(0)}(\mathbf{z}) & H_1^{(0)}(\mathbf{z}) & \cdots & H_{N^2-1}^{(0)}(\mathbf{z}) \\ H_0^{(1)}(\mathbf{z}) & H_{(1)}^{(1)}(\mathbf{z}) & \cdots & H_{N^2-1}^{(1)}(\mathbf{z}) \\ \vdots & \vdots & \vdots & \vdots \\ H_0^{(N^2-1)}(\mathbf{z}) & H_{(N^2-1)}^{(N^2-1)}(\mathbf{z}) & \cdots & H_{N^2-1}^{(N^2-1)}(\mathbf{z}) \end{pmatrix} \begin{pmatrix} E_0(\mathbf{z}) \\ E_1(\mathbf{z}) \\ \vdots \\ E_{N^2-1}(\mathbf{z}) \end{pmatrix} \tag{5.31}$$

I represent the filtering using Fig. 5.15. Each of the filters H_j^i, for $i, j = 0, 1, \cdots, N^2 - 1$, represents a conventional scalar filter that can be derived from the filter coefficients of the multifilter $\tilde{\mathbf{\Gamma}}$ using the polyphase decomposition [32]. Since $\tilde{\mathbf{\Gamma}}$ is fixed, the polyphase components of its rows may be precomputed. N^4 polyphase filters are required for the implementation.

The result is a set of conventional filters with scalar coefficients, which enables the components of the input signal vector sequence to be buffered and filtered independently of the other components. Since each filter H_j^i has fewer coefficients than a row of $\tilde{\mathbf{\Gamma}}$ by a factor N^2, the throughput is increased by a factor of N^2 when the filtering is implemented in parallel on N^2 processors.

Due to the redundancy in the diffusion matrix further savings in com-

---

[1] The corrected version of this paper is available at `http://www.ece.utexas.edu/~bevans/papers/2000/multifilter/index.html`

putation can be achieved. This is because in the case where the diffusion matrix has all of its elements equal (e.g. for block FM and block AM-FM halftoning), $F_0(\mathbf{z}) = F_1(\mathbf{z}) = \cdots = F_{N^2-1}(\mathbf{z})$. Thus, only one set of parallel scalar filtering operations is required. In fact, each parallel filter is identical to the scalar error filter prototype used to design the block filter. By neglecting the cost of determining if the current pixel-block is a minority pixel block or not, then the block error diffusion could be up to N^2 times faster than conventional error diffusion. This is because each scalar polyphase filter operates on an image of $\frac{1}{N^2}$ times smaller than the image a conventional scalar error filter operates on in regular error diffusion halftoning. If the number of pixels in a block N^2 is a power of two, then the summation of the outputs of the parallel filters can be accomplished with a single shift operation.

5.7 Conclusion

This chapter introduces a general framework called block error diffusion for producing FM and AM-FM halftones with user controlled shape, size and sharpness. The framework is also able to embed FM halftones in other FM/AM-FM halftones. Further, owing to the structure of the block error filter, I give a parallel implementation for block error diffusion. The structure imposed on the block error filter encapsulates the properties of conventional scalar error filters. Thus, improved grayscale error filters with reduced artifacts would result in better block error filters. Block error diffusion may be considered a natural extension of scalar grayscale error diffusion to FM halftoning with dot clusters of more than one pixel.

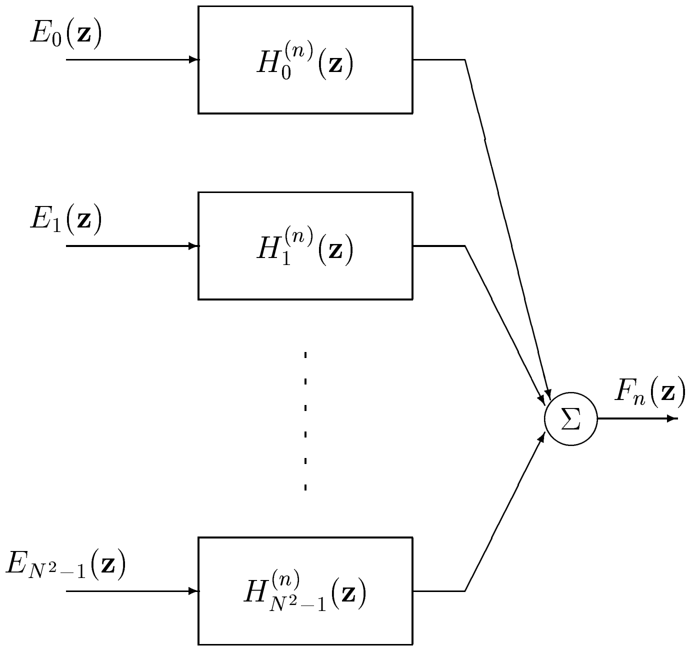


Figure 5.15: A parallel implementation of a multifilter which computes the nth component of the output signal N^2-vector sequence. N^4 filters are required for the computation of all of the components of the output vector sequence.

Chapter 6

Conclusion

This dissertation has develops new tools for the analysis, design, and implementation of vector error diffusion halftoning systems. Here, I summarize the contributions of the dissertation and suggest opportunities for future work in this area.

Chapter 2 introduces the general theory of filtering vector-valued signals. I show the duality that exists between the processing of vector-valued signals and scalar-valued "blocked" signals. In both cases, the generalized linear spatially invariant filtering may be accomplished via a convolution using a filter with matrix-valued coefficients. These filters, when put into polyphase form, have an efficient parallel implementation. I also show that such an implementation was especially valuable when using conventional embedded digital signal processor architectures. Since the error filter in vector error diffusion has matrix-valued coefficients, the results of this chapter apply directly to efficient implementations of vector error diffusion.

Chapter 3 formalizes and generalizes the idea that error diffusion may be approximated as a system that produces frequency distortion and adds additive noise [52]. The modeling approach generalizes modeling methods for scalar error diffusion [19] to the vector case. I linearize vector error diffusion based on a “matrix gain model” for the quantizer that accounts for correlations among the components of the vector error being diffused. I use this modeling to predict the linear signal distortion and noise shaping effects of vector error diffusion. Based on the model, I derive a low-complexity compensation method to eliminate signal frequency distortion in vector error diffusion. The model could potentially be used for color halftone compression, in which one may decide to allocate bits according to the signal distortion and noise injection profiles predicted by the model. Spatially varying extensions of the model could be used to optimize our inverse halftoning approach of [53, 54] and extend it to color images. Spatially varying extensions to the distortion canceling method presented in Chapter 3 have been analyzed in our work on optimal threshold modulation [55] for the specific case of grayscale image halftoning.

Chapter 4 develops a model-based error filter design method in which the objective is to minimize the visual effect of the additive noise injection produced by vector error diffusion. I cast the optimal error filter design problem as a generalized weighted linear prediction problem and derived the set of equations that may be regarded as a generalization of the Yule-Walker equations. The solution of the generalized set of equations results in color error filters with visually optimum noise shaping. The explicit modeling for the human visual system incorporates a generalized linear spatially invariant matrix-valued weighting and is not restricted to the pattern-color separable model [36] that is used to obtain our filters. Thus, more general linear visual models could be used if they were available. Future work could explore the role of the constraints in designing optimal color error filters. Better results were obtained when the lossless diffusion constraints were not strictly observed. Symbolic optimization such as the approach of [56] could be used to explore constrained design spaces in an automated framework where the error filter can be simultaneously optimized to satisfy several constraints.

Chapter 5 exploits the idea that error diffusion on “blocked” grayscale images is equivalent to vector error diffusion. The objective of this “block error diffusion” is to produce FM halftones with clustered dots. The error filter in this case was derived from a conventional error filter prototype. Several extensions to basic FM halftoning were suggested including halftoning with user-controlled dot shapes, embedded mutiresolution halftoning, halftoning with user-controlled sharpness and AM-FM halftoning with and without dot shape and dot size control. Since the block error filter has a well structured and redundant form, I show that a vector error diffusion implementation of block error diffusion may be accomplished at a fraction of the computational complexity of regular vector error diffusion. Possible extensions of this work are to design optimal error filters depending on the dot shape or AM modulation and extend the framework to color halftoning.

Future research in vector color error diffusion could focus on using the matrix gain model to solve practical halftoning problems. In this dissertation I suggested a few applications such as sharpness control and error filter design for optimal noise shaping. Future applications of the matrix gain model in the area of color halftone compression, inverse halftoning of color halftones and optimal spatially varying sharpness control seem promising. For example, JPEG quantization tables or subband quantizer bit allocations in JPEG 200 may be optimized based on the signal and noise transfer functions and the visual model developed in this thesis. One may allocate fewer bits to image frequencies where the color halftoning introduces high distortion. A method to design JPEG quantization tables based on empirically determined frequency distortions for grayscale image halftoning is described in [57]. In color error filter design there are two major open problems which need to be addressed by future research. First, the constraints that are necessary for good image quality in terms of color smoothness (reduced irregularity in the halftone noise) and halftone dot spacing need to be formulated. Second, a flexible sufficient condition for the numerical stability of vector error diffusion needs to be established. The design process suggested for color halftoning in this dissertation does not guarantee optimal dot distributions or numerical stability. The stability of the nonlinear feedback system governed by the error diffusion equations requires that the quantization error be bounded for all input colors in the input color gamut. For block error diffusion, one may be able to design optimal error filters that are appropriate for a given shape. However, the speed and efficiency of using the structured error filters proposed in this dissertation will be compromised. The design of optimal AM screens to perform the AM modulation in block error diffusion is also an open issue. The AM modulation may be performed with different screens depending on the graylevel being rendered. Finally, the framework could be extended to halftone color images using color clustered-dots.

Bibliography

[1] T. Rao, G. Arce, and J. Allebach, "Analysis of ordered dither for arbitrary sampling lattices and screen periodicities," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 38, pp. 1981–1999, Nov. 1981.

[2] R. Floyd and L. Steinberg, "An adaptive algorithm for spatial grayscale," *Proc. Soc. Image Display*, vol. 17, no. 2, pp. 75–77, 1976.

[3] D. Anastassiou, "Error diffusion coding for A/D conversion," *IEEE Trans. Circuits and Systems*, vol. 36, pp. 1175–1186, Sept. 1989.

[4] T. Bernard, "From Σ-Δ modulation to digital halftoning of images," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 2805–2808, May 1991.

[5] R. Ulichney, "Dithering with blue noise," *Proc. IEEE*, vol. 76, pp. 56–79, Jan. 1988.

[6] T. Mitsa and K. Parker, "Digital halftoning using a blue noise mask," *J. Opt. Soc. Am. A*, vol. 9, pp. 1920–1929, Nov. 1992.

[7] R. Ulichney, "The void-and-cluster method for dither array generation," *Proc. SPIE Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, pp. 332–343, Feb. 1993.

[8] M. Analoui and J. Allebach, "Model based halftoning using direct binary search," *Proc. SPIE Human Vision, Visual Processing, and Digital Display III*, vol. 1666, pp. 109–121, Feb. 1992.

[9] R. Levien, "Output dependent feedback in error diffusion halftoning," *IS&T Imaging Science and Technology*, vol. 1, pp. 115–118, May 1993.

[10] D. L. Lau, G. R. Arce, and N. C. Gallagher, "Green noise digital halftoning," *Proc. IEEE Conf. Image Processing*, vol. 2, pp. 39–43, Oct. 1998.

[11] J. Jarvis, C. Judice, and W. Ninke, "A survey of techniques for the display of continuous tone pictures on bilevel displays," *Computer Graphics and Image Processing*, vol. 5, pp. 13–40, May 1976.

[12] J. Mannos and D. Sakrison, "The effects of a visual fidelity criterion on the encoding of images," *IEEE Trans. Information Theory*, vol. 20, pp. 525–536, July 1974.

[13] K. Knox, "Error diffusion: a theoretical view," *Proc. SPIE Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, pp. 326–331, Feb. 1993.

[14] K. Knox and R. Eschbach, "Threshold modulation in error diffusion," *J. Electronic Imaging*, vol. 2, pp. 185–192, July 1993.

[15] S. Norsworthy, R. Schreier, and G. Temes, eds., *Delta-Sigma Data Converters.* IEEE Press, 1997.

[16] Z. Fan and R. Eschbach, "Limit cycle behavior of error diffusion," *Proc. IEEE Conf. Image Processing*, vol. 2, pp. 1041–1045, Nov. 1994.

[17] Z. Fan, “Stability analysis of error diffusion,” *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 5, pp. 321–324, Apr. 1993.

[18] K. Knox, “Error image in error diffusion,” *Proc. SPIE Image Processing Algorithms and Techniques III*, vol. 1657, pp. 268–279, Feb. 1992.

[19] T. D. Kite, B. L. Evans, and A. C. Bovik, “Modeling and quality assessment of halftoning by error diffusion,” *IEEE Trans. Image Processing*, vol. 9, pp. 909–922, May 2000.

[20] S. Ardalan and J. Paulos, “An analysis of nonlinear behavior in delta-sigma modulators,” *IEEE Trans. Circuits and Systems*, vol. 34, pp. 593–603, June 1987.

[21] B. Kolpatzik and C. Bouman, “Optimized error diffusion based on a human visual model,” *Proc. SPIE Human Vision, Visual Processing, and Digital Display III*, vol. 1666, pp. 152–164, Feb. 1992.

[22] T. D. Kite, *Design and Quality Assessment of Forward and Inverse Error-Diffusion Halftoning Algorithms.* PhD thesis, Dept. of Electrical and Comp. Eng., The University of Texas at Austin, Austin, TX, Aug. 1998.

[23] P. Wong and J. Allebach, “Optimum error diffusion kernel design,” *Proc. SPIE/IS&T Sym. on Electronic Imaging*, Jan. 1997. Invited paper.

[24] L. Akarun, Y. Yardimici, and A. E. Cetin, “Adaptive methods for dithering color images,” *IEEE Trans. Image Processing*, vol. 6, pp. 950–955, July 1997.

[25] D. L. Lau, G. R. Arce, and N. C. Gallagher, "Green-noise digital halftoning," *Proceedings of the IEEE*, vol. 86, pp. 2424–2442, Dec. 1998.

[26] L. Velho and J. M. Gomez, "Digital halftoning with space filling curves," *Computer Graphics*, vol. 25, pp. 81–90, July 1991.

[27] T. Scheermesser and O. Bryngdahl, "Control of texture in image halftoning," *J. Opt. Soc. Am. A*, vol. 13, pp. 81–90, Aug. 1996.

[28] T. Mitsa and K. Parker, "Digital halftoning using a blue noise mask," *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 2809–2812, May 1991.

[29] M. A. Coudray, "Causes and corrections of dot gain on press," *Screen Printing: The Journal of Technology and Management*, vol. 86, pp. 18–26, Aug. 1996.

[30] R. Ulichney, *Digital Halftoning.* Cambridge, MA: MIT Press, 1987.

[31] N. Damera-Venkata and B. L. Evans, "Design and analysis of vector color error diffusion halftoning systems," *IEEE Trans. Image Processing*, submitted.

[32] N. Damera-Venkata and B. L. Evans, "Parallel implementation of multifilters," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3335–3338, June 2000.

[33] N. Damera-Venkata and B. L. Evans, "Clustered dot FM halftoning via block error diffusion," *IEEE Trans. Image Processing*, submitted.

[34] T. D. Kite, B. L. Evans, A. C. Bovik, and T. L. Sculley, “Digital halftoning as 2-D delta-sigma modulation,” *Proc. IEEE Conf. Image Processing*, vol. 1, pp. 799–802, Oct. 1997.

[35] G. Strang and T. Q. Nguyen, *Wavelets and Filter Banks.* Cambridge, MA: Wellesley-Cambridge Press, 1996.

[36] A. B. Poirson and B. A. Wandell, “Appearance of colored patterns: Pattern-color separability,” *J. Opt. Soc. Am. A*, vol. 10, pp. 2458–2470, Dec. 1993.

[37] P. P. Vaidyanathan, *Multirate Systems and Filter Banks.* Prentice-Hall, Inc., 1993.

[38] H. Stark and J. W. Woods, *Probability, Random Processes and Estimation Theory for Engineers.* Englewood Cliffs, NJ: Prentice-Hall, 1986.

[39] T. D. Kite. Personal communication: Sept. 25, 1999.

[40] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing.* New Jersey: Prentice-Hall International, Inc., 2000.

[41] A. B. Poirson and B. A. Wandell, “Pattern-color separable pathways predict sensitivity to simple colored patterns,” *Vision Research.*, vol. 36, pp. 515–526, Dec. 1996.

[42] X. Zhang and B. A. Wandell, “A spatial extension of cielab for digital color image reproduction,” *SID 96 Digest of Technical Papers*, pp. 731–734, 1996.

[43] V. Smith and J. Pokorny, "Spectral sensitivity of the foveal cone photopigments between 400 and 500nm," *Vision Research*, vol. 15, pp. 161–171, 1975.

[44] M. D. Fairchild, *Color Appearance Models.* Addison Wesley Longman, 1998.

[45] P. L. Combettes and P. Bondon, "Adaptive linear filtering with convex constraints," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1372–1375, May 1995.

[46] H. Stark and Y. Yang, *Vector Space Projections.* New York, NY: John Wiley and Sons, Inc., 1998.

[47] P. Wong, "Image halftoning using multipath tree coding," *Proc. IEEE Conf. Image Processing*, vol. 2, pp. 31–35, Nov. 1994.

[48] P. Wong, "Entropy constrained halftoning using multipath tree coding," *IEEE Trans. Image Processing*, vol. 6, pp. 1567–1569, Nov. 1997.

[49] P. Wong, "Adaptive error diffusion and its application in multiresolution rendering," *IEEE Trans. Image Processing*, vol. 5, pp. 1184–1196, July 1996.

[50] R. Eschbach and K. Knox, "Error-diffusion algorithm with edge enhancement," *J. Opt. Soc. Am. A*, vol. 8, pp. 1844–1850, Dec. 1991.

[51] S. Haykin, *Communication Systems.* New York, NY: John Wiley and Sons, 1994.

[52] N. Damera-Venkata, T. D. Kite, W. S. Geisler, B. L. Evans, and A. C. Bovik, "Image quality assessment based on a degradation model," *IEEE Trans. Image Processing*, vol. 9, pp. 636–651, Apr. 2000.

[53] T. D. Kite, N. Damera-Venkata, B. L. Evans, and A. C. Bovik, "A high quality, fast inverse halftoning algorithm for error diffused halftones," *Proc. IEEE Conf. Image Processing*, vol. 1, pp. 1583–1593, Oct. 1998.

[54] T. D. Kite, N. Damera-Venkata, B. L. Evans, and A. C. Bovik, "A fast high-quality inverse halftoning algorithm for error diffused halftones," *IEEE Trans. Image Processing*, vol. 9, pp. 1583–1592, Sept. 2000.

[55] N. Damera-Venkata and B. L. Evans, "Adaptive threshold modulation for error diffusion halftoning," *IEEE Trans. Image Processing*, Jan. 2001. to appear.

[56] N. Damera-Venkata and B. L. Evans, "An automated framework for multicriteria optimization of analog filter designs," *IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 46, pp. 981–990, Aug. 1999.

[57] R. A. V. Kam, P. W. Wong, and R. M. Gray, "JPEG-compliant perceptual coding for a grayscale image printing pipeline," *IEEE Trans. Image Processing*, vol. 8, pp. 1–14, Jan. 1999.

Vita

Niranjan Damera-Venkata was born in Madras, India, on December 4, 1976. He received the B.E. degree (majoring in Electronics and Communication Engineering) from the University of Madras, Madras, India, in July 1997 and the M.S.E.E. degree from The University of Texas, Austin, Texas, in May 1999. During the summer of 1999, he interned at Hewlett-Packard Laboratories in Palo Alto, California, where he worked on embedded digital camera image processing. Since July 2000, he has been working as a research engineer with the halftoning and image processing group at Hewlett-Packard Laboratories. Niranjan Damera-Venkata was awarded a bronze medal for academic performance by the University of Madras in 1997, and won a Texas Telecommunications Engineering Consortium (TXTEC) graduate fellowship from The University of Texas in 1999. He is a student member of the IEEE and a member of Sigma Xi.

Permanent address: 15, Kasturi Ranga Road,
Madras, Tamil Nadu 600018,
INDIA

This thesis was typeset with LaTeX[‡] by the author.

[‡] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.