DESIGN AND QUALITY ASSESSMENT OF FORWARD AND INVERSE ERROR DIFFUSION HALFTONING ALGORITHMS

APPROVED BY
DISSERTATION COMMITTEE:

Supervisor: _____

Supervisor: _____

_____

_____

_____

_____

This thesis is dedicated to my parents,

and to the memory of my wonderful Gran.

DESIGN AND QUALITY ASSESSMENT OF FORWARD AND INVERSE ERROR DIFFUSION HALFTONING ALGORITHMS

by

THOMAS DAVID KITE, B.A., 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

August 1998

Acknowledgments

First of all, I would very much like to thank my advisors, Al Bovik and Brian Evans (in alphabetical order) for all their help during my time in graduate school. I believe that their complementary styles have given this work a unique character. They have broadened my horizons in totally different ways outside school, and have been pillars of strength in a sometimes precarious landscape. I am indebted to them both.

Both have a collection of fine graduate students, who have brightened each day with far-reaching, eclectic, and occasionally academic conversation: Joebob, Bill, Dave, Hung-Ta, Kartick, Dong, Sanghoon and Marios (without whom the lab will never be quite the same) from Al's lab, and Wade, Niranjan, Güner and Biao from Brian's. I owe special thanks to Biao for making sure that preparations for the defense went smoothly while I was out of town. I wish them all the best of luck in everything they do.

Many people have contributed in less tangible ways. My great friends Paul Calamia and Eric Rosenberg have made my last two years in school a time to remember, by providing me not just entertainment of boundless variety, but also friendship that is comfortingly familiar, yet dazzlingly unpredictable. Brent Bliven and Jim Haley continue to enrich my life with their astounding intelligence, perception, and affection. John Post, Robin Cleveland, Pete Ziev-

ers, Greg Woodward, Rudy Bauss, Rebecca Nowlin, Aashlesha Patel, Stacy Genovese, Stacy Manning, Nina Bhattacharya, and many others have not only made Austin memorable for me, but have also subtly changed who I am by their friendship. Thank you!

I am very grateful to Adela Baines and Melanie Gulick for helping me negotiate the extraordinary obstacle course known as University Procedure. Without them, nothing would get done.

I am proud to be able to call Dr. John Cogdell and Dr. Elmer L. Hixson my friends, as well as my advisors of one sort or another. They have been particularly kind during my time at the University, especially at the times when I needed the most help.

I would like to thank all the members of my committee for agreeing to take part, for providing valuable comments on the dissertation, and for putting up with the rather dry banana bread I served them at the qualifying exam. I hope that the offerings at the defense were more up to their expectations.

Finally, my love and inexpressible gratitude go to my family, for their support throughout all the stages of my life. As I embark on the next, I know that they will be there to guide me once again.

Thomas Kite
August, 1998

DESIGN AND QUALITY ASSESSMENT OF FORWARD AND INVERSE ERROR DIFFUSION HALFTONING ALGORITHMS

Publication No. _____

Thomas David Kite, Ph. D.
The University of Texas at Austin, 1998

Supervisors: Alan C. Bovik, Brian L. Evans

Digital halftoning is the process by which a continuous-tone image is converted to a binary image, or halftone, for printing or display on binary devices. Error diffusion is a halftoning method which employs feedback to preserve the local image intensity and reduce low frequency quantization noise. It is a highly nonlinear process, and it is therefore difficult to analyze mathematically. In this work, a linear gain model for the quantizer is presented which accurately predicts the *edge sharpening* and *noise shaping* effects of error diffusion. The model is used to construct a residual image that has a low correlation with the original image. By weighting this residual with a model of the human visual system, a measure of the subjective effect of the quantization noise on the viewer is obtained. A distortion metric for the halftoning scheme is also computed. By characterizing the edge sharpening, noise shaping, and distortion of an error diffusion scheme, *objective measures* of *subjective quality* of halftones are obtained. This permits the comparison of halftoning schemes.

A new, efficient inverse halftoning scheme for error diffused halftones is presented that produces results comparable to the best current methods, but at a fraction of the computational cost. A method of modeling inverse halftoning schemes is demonstrated, and is used to generate residual images, which are weighted with the human visual system model. An effective transfer function for the inverse halftoning scheme is also computed. By characterizing the degree of blurring and the noise content, objective measures of subjective quality of inverse halftones are obtained. This allows competing inverse halftoning algorithms to be compared. The linear gain model is further used to design and analyze the performance of applications which include error diffusion. The model of the human visual system is again used to obtain objective measures of the quality of images produced by these applications.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Since the advent of the printing press, it has been desirable to reproduce grayscale (multi-bit) imagery on inherently binary (one-bit) media. Simple truncation of the grayscale image gives visually unacceptable results; instead, a specialized binarization procedure must be used that attempts to preserve image features and graylevels. This process is known in the printing industry as *halftoning*. By judiciously applying dots to the paper in patterns of varying density, it is possible to achieve the illusion of grayscale.

Many digital halftoning methods exist, each with its own strengths and weaknesses. In this chapter, an overview of the more common schemes is presented, and their advantages and disadvantages are described. The focus is on error diffusion, and important concepts and mathematical results are introduced that will be used throughout the rest of this work. The delta-sigma modulator, which is the one-dimensional equivalent of error diffusion, is discussed, as is inverse halftoning; that is, the process by which a grayscale image can be estimated from its halftone representation. Finally, the organization of the rest of the dissertation is presented.

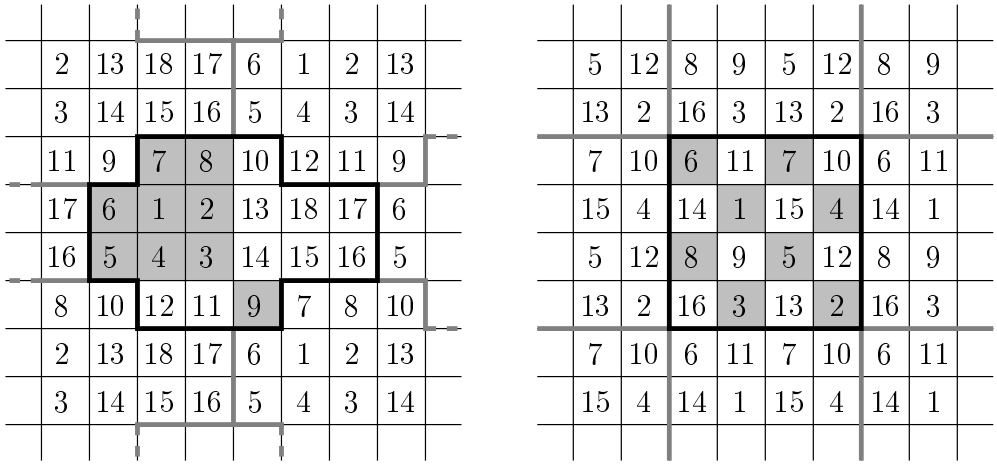
1.1 Common halftoning methods

Devices such as printing presses, ink jet printers and laser printers cannot print in shades of gray. They can only apply (or not apply) ink to the paper at every point. Low-cost liquid crystal displays (LCDs) have the same limitation. To reproduce grayscale imagery using these devices, it is necessary to halftone the grayscale image to produce a binary image that gives the impression of grayscale when viewed by a human being. In this section, the four most common halftoning methods in current use are examined.

1.1.1 Classical screening

The oldest and simplest halftoning method is *screening*, also known as *ordered dithering* [1]. A periodic mask of thresholds, or "screen", is constructed, which is of the same size as the grayscale image. Pixels with intensities below the corresponding screen threshold become zero (black) in the halftone, whereas pixels with intensities higher than the threshold become one (white).

Figure 1.1 shows two classical screens. The thin dark lines represent the borders between image pixels. The area surrounded by the thick black line is the screen itself; areas surrounded by thick gray lines are replications of the screen. The screen thresholds range linearly from 0 to 1, but to simplify the figure, the *ordering* of the thresholds is shown, rather than their grayscale values. For instance, in Figure 1.1(a), the threshold labeled '1' has a value of 1/19, the threshold labeled '2' has a value of 2/19, and so on, up to the threshold labeled '18', whose value is 18/19. If this screen were used to halftone a constant image of graylevel 1/2, then the pixels covered by thresholds 1 to



2	13	18	17	6	1	2	13	
3	14	15	16	5	4	3	14	
11	9	7	8	10	12	11	9	
17	6	1	2	13	18	17	6	
16	5	4	3	14	15	16	5	
8	10	12	11	9	7	8	10	
2	13	18	17	6	1	2	13	
3	14	15	16	5	4	3	14	

(a) 19-level, clustered-dot.

5	12	8	9	5	12	8	9	
13	2	16	3	13	2	16	3	
7	10	6	11	7	10	6	11	
15	4	14	1	15	4	14	1	
5	12	8	9	5	12	8	9	
13	2	16	3	13	2	16	3	
7	10	6	11	7	10	6	11	
15	4	14	1	15	4	14	1	

(b) 16-level, dispersed-dot.

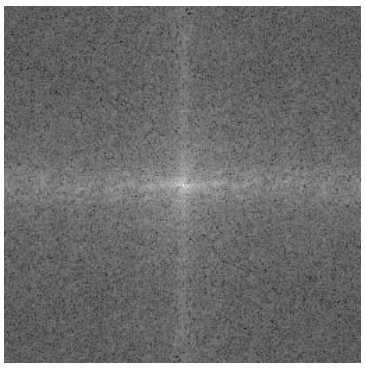
Figure 1.1: Threshold masks for two common screening methods. The ordering of the threshold values is shown, not their actual values. Shaded pixels show the screened output for a uniform input of 1/2.

9 would be dark in the final image, and the pixels covered by thresholds 10 to 18 would be light. The halftone would be perceived as having a graylevel of 1/2. This is indicated by the shaded pixels.

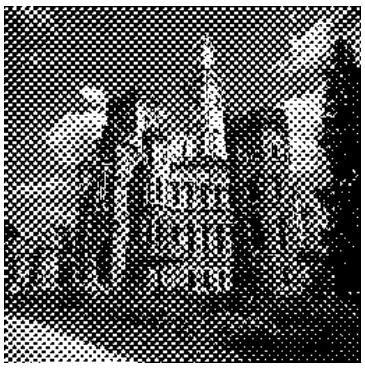
If a screen is of size N pixels, it can support $N + 1$ graylevels, since any integer number of pixels in the screen from zero to N can be made dark. The difference between the two screens in Figure 1.1 (apart from the minor difference in the number of graylevels they can support) is that the clustered-dot screen shown in Figure 1.1(a) forms clumps, or clusters, of dots, while the dispersed-dot screen shown in Figure 1.1(b) keeps the dots as far apart as possible. This can be seen in the shaded pixels of Figure 1.1.

The ordering of the thresholds in the screen determines the characteristics of the screen, and has a large effect on the visual quality of the halftone.

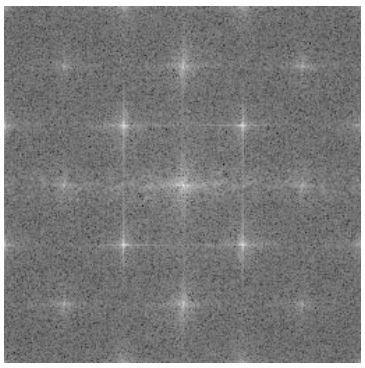
(a) Original *castle* image.



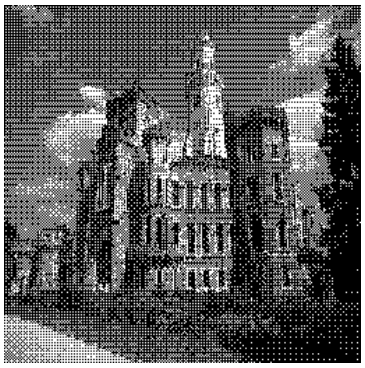
(b) Fourier transform.



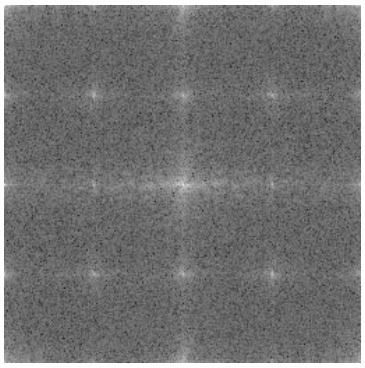
(c) Clustered-dot halftone.



(d) Fourier transform.



(e) Dispersed-dot halftone.



(f) Fourier transform.

Figure 1.2: Screened halftones and their discrete Fourier transforms.

Figure 1.2(a) shows the original *castle* image.[1] Figures 1.2(c) and 1.2(e) show the clustered-dot and dispersed-dot halftones, respectively. Both halftones contain noticeable artifacts, most notably *contouring* (false edges) due to the low number of graylevels, and a loss of spatial resolution because of the large screen size. The number of graylevels can be increased at the expense of a loss of spatial resolution by using a larger screen. Both methods also suffer from Moiré patterns, which are caused by halftoning an image with a strong component at a frequency close to the screen frequency.

Clustered-dot screening produces a much coarser, more visually objectionable image than dispersed-dot screening. The advantage of the clustered-dot technique is that it is more resistant to the phenomenon known as ink spread [2]. When a laser printer applies toner to the paper, the resulting dot is not a perfect square. It is usually round, with considerable overlap with neighboring pixels; furthermore, the toner tends to spread out on the paper, producing a dot that is larger than one would like. The result is that the toner covers more area than expected, and images therefore appear darker than they should. Knowledge of the pixel size and the ink spread function allows this to be pre-corrected [3], but it requires individual calibration, taking into account the characteristics of the printer, toner, and paper; a simpler solution is to apply large blobs of toner. The effect of non-square pixels and ink spread is only seen at the edges of dark areas, so the fractional increase in area,

[1] Images referred to as "original" have been halftoned by the printing process used to render this work. All images are therefore of low spatial resolution (256×256 pixels unless otherwise stated) and have been reproduced at as large a size as possible, to mitigate the effect of the printer. This produces grainy halftones. The graininess can be reduced by holding the page further from the eye. This effect will be explained in Chapter 2.

and hence the error in the graylevel, is smaller for the larger, clustered dots. Clustered-dot screening is therefore more robust than dispersed-dot screening; the improvement in consistency across toner and paper types outweighs the loss in performance due to the increased dot size.

The screening process can be modeled as a pointwise multiplication of the original image with a periodic dot pattern [4]. The result is that the Fourier transform of the halftone consists of the spectrum of the grayscale image, and multiple aliased copies of this spectrum spread over the entire frequency plane [5, 6]. Figure 1.2(b) shows the discrete Fourier transform of the original *castle* image, while Figures 1.2(d) and 1.2(f) show the discrete Fourier transforms of the clustered-dot and dispersed-dot halftones, respectively. All three spectra have similar low frequency regions, near the center of the images. However, copies of the spectrum of the original image appear throughout the transforms of the halftones. As will be explained in Chapter 2, the human visual system can be modeled as a lowpass filter. Low frequency image components are therefore more visible than high frequency components. The fact that the aliased spectra are higher in frequency for the dispersed-dot halftone than for the clustered-dot halftone explains the more pleasing appearance of the dispersed-dot halftone, since they are more strongly attenuated by the lowpass human visual system.

The primary advantage of screening is its simplicity. It is a *point process*, that is, only the graylevel of the current pixel, and not its neighbors, is required to compute the output. A single threshold operation per output pixel is required, and, since the screens themselves are small, little memory is

needed. Furthermore, the resistance of clustered-dot screening to ink spread and dot positioning errors makes it attractive for lower cost printers.

1.1.2 Dithering with blue noise

In 1987, Ulichney introduced the concepts of *blue noise* and *principal frequency* to characterize halftoning algorithms [1]. All halftoning algorithms introduce error into an image; this error is known as *quantization error*, since it is due to reducing the wordlength from a typical value of eight bits to one bit. Under certain circumstances, referring to this error as *quantization noise* is justified. Use of the term "noise" in this context implies that the quantization error has a random character. Ulichney proposed that noise with a highpass characteristic ("blue noise") was the ideal error from a perceptual point of view [7]. He also showed that halftones created by error diffusion, which is described in Section 1.1.4, have such a characteristic.

Figure 1.3 shows the noise spectrum produced by halftoning a uniform (DC) input image with a halftoning scheme that has ideal blue noise characteristics. Here, f_r refers to radial spatial frequency, defined as $\sqrt{f_x^2 + f_y^2}$, where f_x and f_y are the spatial frequencies in the x and y directions, respectively, and the the noise power is assumed to be isotropic. Ulichney showed that images with isotropic noise spectra have a higher perceived quality than images whose noise power is not isotropic. The spectral distribution is characterized by low noise power at low radial frequencies, a sharp transition to a peak at the principal frequency f_g, and a flat power spectrum above f_g. The principal

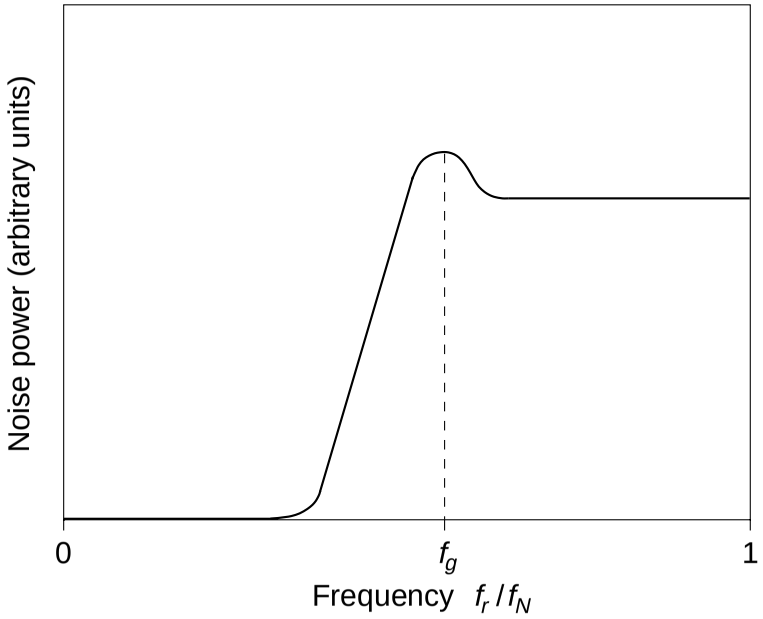


Figure 1.3: Blue noise characteristic. f_r and f_N refer to radial frequency and the Nyquist frequency, respectively. The principal frequency of graylevel g is denoted f_g.

frequency is given by

$$f_g = \begin{cases} \sqrt{g}\,, & 0 \le g \le \frac{1}{2} \\ \sqrt{1-g}\,, & \frac{1}{2} < g \le 1 \end{cases}, \tag{1.1}$$

where g is the graylevel of the uniform image. The relation in (1.1) arises from the fact that, for $g \le \frac{1}{2}$, that is, when white pixels are in the minority, an average proportion g of the pixels in a unit area are white. There are therefore $\sqrt{g}$ white pixels per unit length, on average. This is the principal frequency. For $\frac{1}{2} < g \le 1$, when black pixels are in the minority, the same argument applies to the black pixels, giving a principal frequency of $\sqrt{1-g}$.

Mitsa and Parker combined the blue noise concept and classical screening to form the *blue noise mask*, a screen whose thresholds are arranged to produce a halftone with blue noise characteristics [8]. Much larger screens are

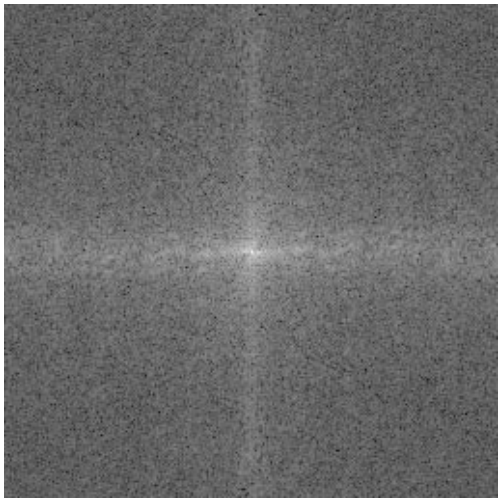
required than for classical screening (typically 128×128 or 256×256 pixels) so that the periodicity of the screen is not noticeable. In psychophysical testing, halftones generated by the blue noise mask rate much higher than halftones created by ordered dithering, for the same computational cost [9]. Before their work, it was assumed that the blue noise characteristic could only be achieved by a *neighborhood process*, that is, one that requires knowledge of the gray-levels of the current pixel and its neighbors to compute an output pixel. Such processes are discussed next.

1.1.3 Direct binary search

Direct binary search (DBS) methods, first introduced in [10] by Analoui and Allebach, create halftones by directly manipulating the pixels in the halftone to minimize a distortion measure, such as the weighted mean squared error. (An example of such a metric is discussed in detail in Chapter 2.) The modification of pixels is governed by a heuristic that allows a small number of manipulations, such as pixel toggling and pixel swapping with a neighbor. The procedure is iterative, and can require thousands of passes through the image if the starting point is not chosen carefully. It is therefore very slow. Furthermore, convergence on the optimal image is not guaranteed.

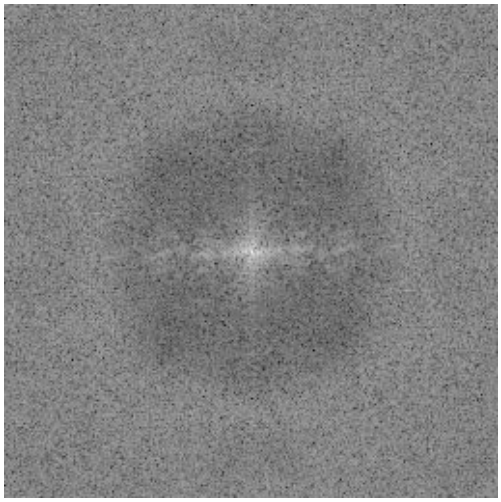
Figure 1.4(a) shows the original *castle* image, while Figure 1.4(c) shows the DBS halftone. The halftone has a pleasing, isotropic arrangement of dots in the shadow areas, with little artificial texture. The apparent noise level is low, and the edges are sharp. Its discrete Fourier transform, shown in Figure 1.4(d), resembles the discrete Fourier transform of the original image at low frequencies, but is swamped by quantization noise as the frequency increases.

(a) Original *castle* image.



(b) Fourier transform.

(c) DBS halftone.



(d) Fourier transform.

Figure 1.4: 512×512 direct binary search halftone and its discrete Fourier transform. The original image and the DBS halftone were provided by Professor Jan P. Allebach and David J. Lieberman, Purdue University. Their assistance is gratefully acknowledged.

The noise is almost perfectly isotropic.

Related to DBS are iterative techniques for designing stochastic halftoning screens. In this application, the high computational cost of the search is unimportant, because the screen is computed off-line. A conventional screening technique is used to generate the halftone itself [11, 12]. Models can be incorporated into the design of the screen to match the human visual system, and to improve performance on a given printer [13].

Although DBS is slow, algorithms exist which are reasonably efficient for a search-based scheme. The halftones it produces are close to the best possible; techniques such as simulated annealing can give a slight improvement, but at enormously increased computational cost. Therefore, DBS serves a useful function in establishing a practical upper bound on the visual quality of a halftone. The purpose of all other halftoning schemes is to approach this limit as closely as possible, at the lowest computational cost.

1.1.4 Error diffusion

Error diffusion was introduced in 1976 by Floyd and Steinberg [14]. It was a completely new method of image halftoning that produced much higher quality images than screening, though at increased computational cost. The algorithm relies on distributing the quantization error from thresholding to neighbors of the current pixel. As the image is scanned (usually in raster fashion, i.e., from left to right, and top to bottom), the quantization error "diffuses" across and down the image, giving the algorithm its name. Qualitatively speaking, error diffusion accurately reproduces the graylevel in a local region by driving the

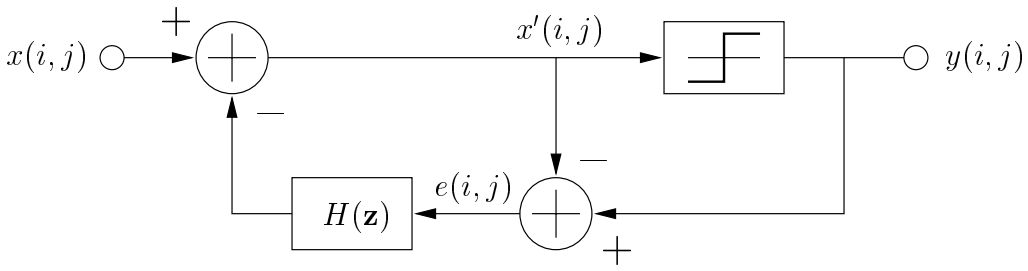


Figure 1.5: Equivalent circuit of error diffusion, also known as a noise shaping feedback coder. The graylevel input image is denoted $x(i,j)$; the one-bit output is denoted $y(i,j)$.

average error to zero through the use of feedback.

The equivalent circuit of error diffusion is shown in Figure 1.5. The process is described mathematically as follows. Assume an input image $x(i,j)$ of size $M \times N$ pixels, with pixel values ranging from 0 to 1. As the algorithm proceeds, each input pixel is effectively modified by the weighted errors diffused from previous pixels; this modified input is denoted $x'(i,j)$. For the first pixel in the image, $x'(i,j) = x(i,j)$. The modified input $x'(i,j)$ is thresholded to produce an output pixel $y(i,j)$:

$$y(i,j) = \begin{cases} 0, & x'(i,j) < 0.5 \\ 1, & x'(i,j) \geq 0.5 \end{cases} . \tag{1.2}$$

The quantization error is given by

$$e(i,j) = y(i,j) - x'(i,j) \ , \tag{1.3}$$

and is subtracted from neighboring pixels according to

$$x'(k,l) = x(k,l) - h(k-i, l-j)\, e(i,j) \ , \quad \begin{cases} 0 < k < M-1 \\ 0 < l < N-1 \end{cases} , \tag{1.4}$$

where $h(i,j)$ is known as the *error filter*. The filter is denoted $H(\mathbf{z})$ in Figure 1.5, where $\mathbf{z}$ refers to the two-dimensional vector (z_1, z_2) in the z-transform

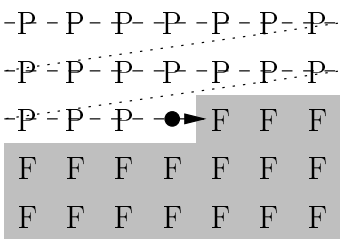


Figure 1.6: Definition of past and future for raster ordering.

plane. The definition of (1.4) is general; any function $h(i,j)$ is allowed. In practice, $h(i,j)$ is non-zero only for those pixels defined to be ahead of the current pixel for the scan used, that is, for those pixels that have not yet been thresholded. For instance, the raster scan defines an ordering shown in Figure 1.6. The scan is indicated by the dashed line. The current pixel is depicted by the black disk; pixels defined to be in the past are labeled 'P', while those in the future are labeled 'F' and shaded. The error filter $h(i,j)$ is non-zero only for the 'F' pixels. Thus the weighted quantization error is distributed only to those pixels which have yet to be visited by the scan.

Floyd and Steinberg designed the following four-tap error filter:

$$h(1,0) = \tfrac{7}{16}; \quad h(-1,1) = \tfrac{3}{16}; \quad h(-1,0) = \tfrac{5}{16}; \quad h(1,1) = \tfrac{1}{16}\ . \tag{1.5}$$

They arrived at these coefficients "mostly by trial and error" [14]. However, they give good visual results, and it has proved difficult to improve on their performance without increasing the computation required. The filter coefficients in (1.5) are indexed relative to the current pixel. The filter is shown schematically in Figure 1.7.

Figure 1.8 shows two examples of halftoning by error diffusion. The original *castle* image is shown in Figure 1.8(a). The Floyd-Steinberg halftone

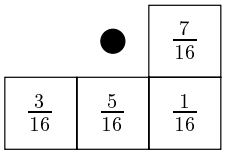


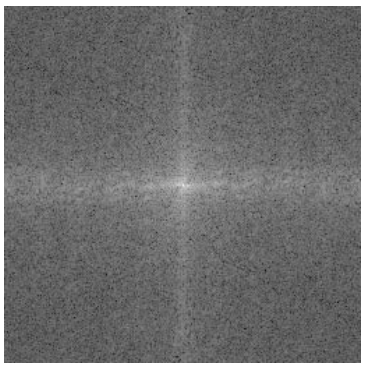
Figure 1.7: Floyd-Steinberg error filter. The current pixel is indicated by the black disk.

is shown in Figure 1.8(c), while the halftone generated using a filter due to Jarvis *et al.* [15] is shown in Figure 1.8(e). The halftones show good rendition of grayscale, sharp edges, and low apparent noise. However, artifacts due to the raster order of processing can be seen in the dark tree in the right foreground, and in parts of the sky. Both halftones are sharper than the original image; this effect will be examined in Chapter 3.

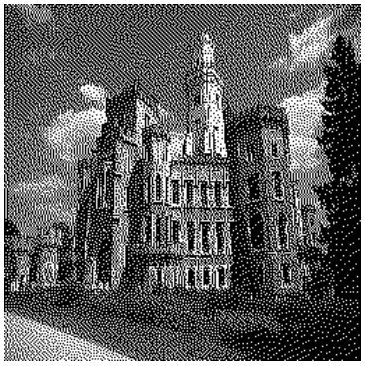
In a similar manner to DBS halftones, the Fourier transform of an error diffused halftone consists of the original image immersed in a bed of noise whose power rises with increasing spatial frequency. Figure 1.8(b) shows the discrete Fourier transform of the original image, while Figures 1.8(d) and 1.8(f) show the discrete Fourier transforms of the two halftones. The low frequency spectra of the halftones are almost identical to the original image. At high frequencies, the quantization noise swamps the image power. The noise is not completely isotropic, especially for the Jarvis image. The anisotropy is consistent with the directional artifacts seen in the halftones.

In psychophysical tests, error diffused halftones rate higher than those produced by screening, including screening with a blue noise mask [16]. The improvement comes at the expense of an increase in computation, since error diffusion is a neighborhood process. However, it produces the best images

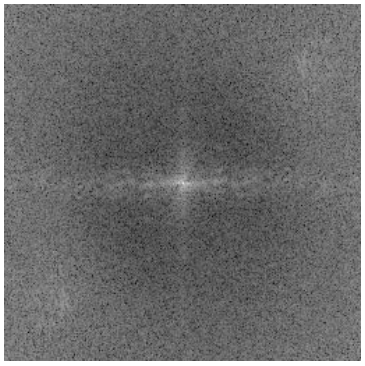
(a) Original *castle* image.



(b) Fourier transform.

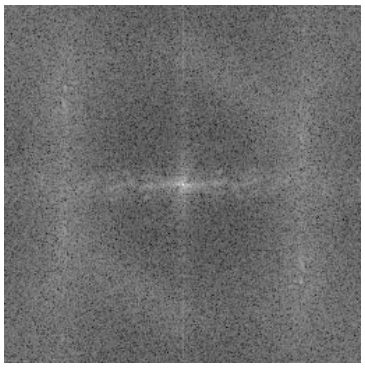


(c) Floyd-Steinberg halftone.



(d) Fourier transform.

(e) Jarvis *et al.* halftone.



(f) Fourier transform.

Figure 1.8: Error diffused halftones and their discrete Fourier transforms.

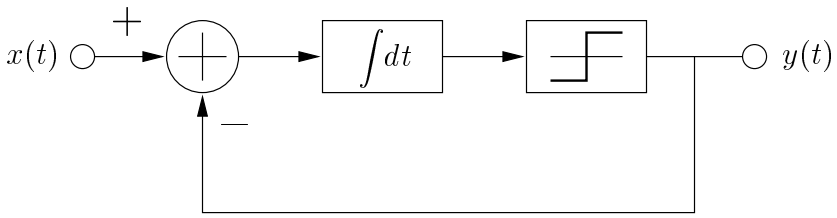


Figure 1.9: Equivalent circuit of first-order delta-sigma modulator.

possible in reasonable time on printers which are capable of reliably and repeatably placing dots at specific points on the page.

1.2 Error diffusion and delta-sigma modulation

Delta-sigma modulation has become popular in the last decade as a way of building high quality, low cost data converters in VLSI technology. It permits the use of a low resolution converter in a high resolution application by feeding back the quantization error to linearize the converter and reduce the in-band quantization noise [17]. A first-order delta-sigma modulator is shown in Figure 1.9. The total noise *power* introduced by quantization is a function of the coarseness of the quantizer. However, by spreading the noise power over a larger range of frequencies using *oversampling*, the noise *density* is lowered, and much of the noise power falls outside the passband [18].

Oversampling by a factor of four reduces the total noise power in the passband by a factor of four, and therefore the noise *voltage* is reduced by a factor of two, or 6 dB. This is equivalent to one extra bit of resolution [19]. This is a low rate of return; to increase the resolution of a one-bit converter to 16 bits, for instance, an oversampling ratio of 4^{15} (over one billion) would

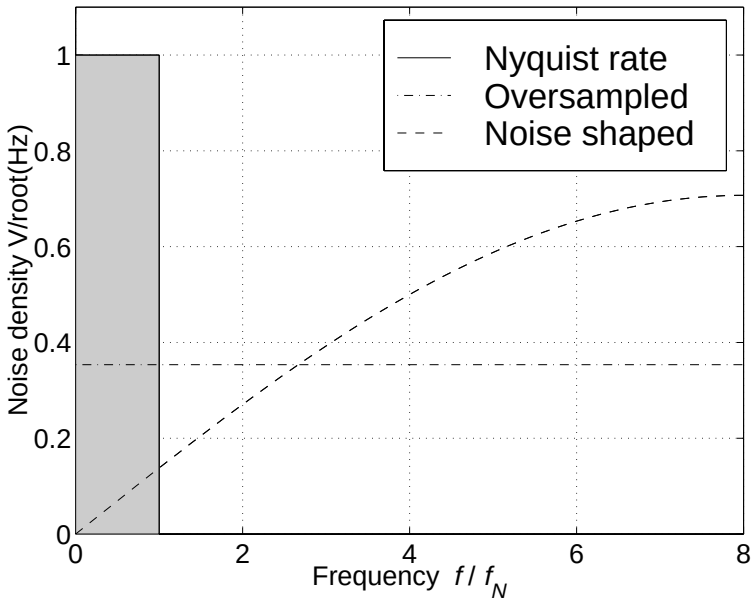


Figure 1.10: Effect of oversampling on quantization noise spectrum. The Nyquist rate is denoted f_N. The oversampling ratio is eight times in this figure. Noise shaping is first order. The passband is shown shaded.

be required. The solution is to employ delta-sigma modulation, in which the quantization noise is *shaped*, reducing its power at low frequencies at the expense of the power at high frequencies.

Figure 1.10 shows the effect of noise shaping on the quantization noise spectrum. The solid line shows the noise density for the Nyquist rate system, normalized to unity. The shaded area represents the passband noise power. Oversampling by a factor of eight (dot-dashed line) spreads the noise power over a wider bandwidth, reducing the in-band noise density to $1/\sqrt{8} \approx 0.35$ of its value in the Nyquist rate system. Noise shaping (dashed line) further reduces the in-band noise density, at the expense of out-of-band noise. In a digital-to-analog conversion application, the oversampled bitstream is filtered by a low-order analog lowpass filter to remove the out-of-band noise. By using

a sufficient oversampling factor, and high-order noise shaping, resolution that is limited only by the analog noise of the surrounding circuitry may be achieved. Delta-sigma modulation has become synonymous with the use of a one-bit converter operated at high oversampling rates, although longer wordlengths are possible and are in common use [18].

The analogy between digital halftoning and delta-sigma modulation was first made explicit by Anastassiou in 1989 [20]. He discusses features common to both systems, including the nature of quantization error and the effect of the error filter on stability, and uses results from the literature on delta-sigma modulation to explain effects seen in error diffusion. Bernard provided further insight in 1991 [21]. However, the focus of both of these papers is exploring efficient halftoning methods in hardware, rather than analyzing or improving error diffusion.

The delta-sigma modulator topology shown in Figure 1.9 is used in analog-to-digital converters; the block labeled $\int dt$ is a discrete-time analog integrator. As explained in Chapter 3, systems employing one-bit quantizers are difficult to analyze mathematically, because of the non-linearity of the quantizer. However, a comparison of the time-averaged Fourier transforms of the input and output signals of Figure 1.9 shows that the delta-sigma modulator effectively shapes the spectrum of the quantization noise by placing a zero at DC in the noise transfer function [18]. This reduces low frequency noise at the expense of increased high frequency noise. In an oversampled system, only the low frequency portion of the spectrum is of interest; noise-shaping therefore reduces the *in-band* noise level.

An alternative form of the delta-sigma modulator, known as the *noise shaping feedback coder*, is used for wordlength reduction. The equivalent circuit of error diffusion, shown in Figure 1.5, is a two-dimensional, single-bit version of this coder. The objective is to reduce the wordlength of an input stream while retaining as much information as possible, without changing the sampling rate. Simple truncation results in signals smaller than the least significant bit (LSB) being lost, and introduces correlated error. To avoid correlated error, *dither* is added to the input before wordlength reduction. Dither is a random signal, usually with a triangular probability density function[2], which decorrelates the quantization noise and allows signals below the LSB to be recovered [19]. It is an essential component of a digital audio system, and was used by Roberts to reduce correlated quantization error in images, which manifests itself as contouring [22].

1.3 Inverse halftoning

Inverse halftoning attempts to recover a grayscale image from its halftone representation. It has become important now that manipulation of digital images is possible on inexpensive embedded hardware and desktop computers. A document which is printed and subsequently optically scanned may contain a mixture of text, graphics, and halftones. If the scanned image is resized or rotated, the quality of the halftones will be degraded [23]. It is necessary to convert the halftones to grayscale before manipulating them. They can be

[2]Triangular pdf dither is commonly used because it perfectly linearizes the quantizer, and results in an ideal noise floor, that is, one that is not modulated by the input signal. It is believed to be the optimal dither signal in this regard [19].

re-halftoned for printing, if needed. A side benefit is that the re-halftoning scheme can be tailored to the user's local printer for the best visual results.

Information is lost when converting a grayscale image to a halftone, since the wordlength is reduced to one bit, and oversampling is not generally used. Thus, exact recovery of a grayscale image from its halftone is impossible. However, by using known characteristics of the halftoning scheme and typical images, it is possible to reconstruct a visually acceptable image.

Halftones produced by error diffusion or direct binary search have a highpass quantization noise spectrum. Since most natural images have a lowpass spectrum [24], lowpass filtering would appear to be the solution to inverse halftoning. However, the image and noise spectra overlap, and it is impossible to find a cutoff frequency for the lowpass filter that suppresses noise sufficiently without unacceptably blurring the image. Instead, an adaptive scheme must be used that varies the effective cutoff frequency of the lowpass filter according to the local image content. Halftones produced by screening have strong artifacts, because aliased images of the Fourier transform appear at low spatial frequencies, where they obscure important image components. It is much more difficult to achieve good grayscale reconstructions from screened halftones than from error diffused halftones.

Several inverse halftoning algorithms have appeared in the literature. However, those yielding high quality are computationally expensive [25, 26, 27]. There is therefore a strong motivation to devise inverse halftoning schemes capable of high quality at a reasonable computational cost.

1.4 Organization of the dissertation

The remainder of this dissertation focuses on error diffusion. A model that predicts important features of error diffused halftones is presented. The importance of modeling error diffusion to obtain accurate measures of halftone visual quality is demonstrated. A new, fast inverse halftoning method for error diffused halftones is presented, and it is shown that modeling is also important for measuring the quality of inverse halftones. Ideas from the analysis of error diffusion and the inverse halftoning algorithm are used to design and analyze novel applications of forward and inverse halftoning. Finally, conclusions and ideas for further research are presented. The work is organized as follows:

Chapter 2: Image Quality Metrics

The peak signal-to-noise ratio (PSNR) measure commonly used for image quality is inadequate for all but the simplest degradations. A model for the human visual system that is used to derive *objective measures* of the *subjective quality* of halftones and inverse halftones is presented. The need to first obtain a residual image that has low correlation with the original image before computing a weighted signal-to-noise ratio (WSNR) is demonstrated, and the WSNR measure is used to assess the quality of halftones and inverse halftones.

Chapter 3: Error Diffusion

A mathematical analysis of error diffusion that uses a *linear gain model* for the quantizer is presented. The model, whose accuracy is demonstrated in three novel, independent ways, predicts the edge sharpening intrinsic to error diffusion. It decouples the edge sharpening from the noise shaping, allowing

the two effects to be quantified independently. A distortion metric that characterizes the tonality of halftoning schemes is also presented. The model also provides a framework for the design of error filters for specific applications. The human visual system model from Chapter 2 is used to assess the quality of halftones.

Chapter 4: Inverse Halftoning

A new inverse halftoning method is presented which produces inverse halftones whose quality is equal to, or better than, images produced by existing methods, but at a fraction of the computational cost. A model for inverse halftoning is presented which decouples the intrinsic blurring from the quantization noise, allowing each to be quantified independently. The human visual system model from Chapter 2 is used to assess the quality of inverse halftones.

Chapter 5: Applications

Results from Chapters 2, 3 and 4 are used to devise novel applications of error diffusion. By introducing an approximation to the digital frequency, optimum values for the sharpness parameter in modified error diffusion are derived. Rehalftoning and oversampling schemes are thereby designed, with the emphasis on high visual quality and low computational cost.

Chapter 6: Conclusions

The original contributions of this dissertation are summarized, and ideas for future work are presented.

Chapter 2

Image Quality Metrics

Algorithms such as halftoning, inverse halftoning, and image restoration result in an image which visually resembles a benchmark image, commonly referred to as the "original image". The performance of these algorithms must be quantified to allow comparison between competing schemes. Conducting psychovisual tests under controlled conditions is time-consuming and error-prone. There is therefore a strong incentive to develop a method of *computationally* estimating image quality. A *distance measure* is required that numerically expresses the perceived visual difference between an original image and a processed version.

Traditionally, signal-to-noise ratio (SNR) and peak signal-to-noise ratio (PSNR) have been used as distance measures. In this chapter, their deficiencies will be demonstrated, especially when they are used to assess halftones and inverse halftones. A distance measure will be described that incorporates a model of the human visual system. This measure has a higher correlation with psychovisual data than both SNR and PSNR. It will also be shown that it is necessary to first account for image distortions before computing the distance measure, to obtain accurate results.

2.1 Distance measures

Image processing algorithms often produce an image which is intended to visually resemble another image. Image restoration, for instance, attempts to recover an image corrupted by blurring, noise, and possibly other distortions; to test the accuracy of the restoration algorithm, the restored image is compared to a known original. In lossy image compression, the aim is to compress an image in such a way that, for a given bit rate, the processed image is as similar as possible to the original. In digital halftoning, one attempts to create a binary image which resembles the original image closely when viewed by a human being. In inverse halftoning, the aim is to re-create a grayscale image from a halftone that visually resembles the original.

To quantify the performance of such algorithms, one must define a measure of image quality. Signal-to-noise ratio (SNR) and peak signal-to-noise ratio (PSNR) are commonly used. Both are mean-squared (l_2-norm) error metrics. For an image of size $M \times N$ pixels, SNR is given by

$$\text{SNR (dB)} = 10 \log_{10} \left(\frac{\sum_{i,j} x(i,j)^2}{\sum_{i,j} (x(i,j) - y(i,j))^2} \right), \quad \left\{ \begin{array}{l} 0 < i < M-1 \\ 0 < j < N-1 \end{array} \right. , \quad (2.1)$$

where $x(i,j)$ denotes pixel (i,j) of the original ("clean") image, and $y(i,j)$ denotes pixel (i,j) of the noisy image. PSNR, being a peak measure, depends on the wordlength of the image pixels. For 8-bit images, PSNR is given by

$$\text{PSNR (dB)} = 10 \log_{10} \left(\frac{D^2 MN}{\sum_{i,j} (x(i,j) - y(i,j))^2} \right), \quad \left\{ \begin{array}{l} 0 < i < M-1 \\ 0 < j < N-1 \end{array} \right. , \quad (2.2)$$

where x and y are defined as before, and D is the maximum peak-to-peak swing of the signal. For 8-bit images, $D = 255$ typically. SNR is defined as

the ratio of the average signal power to the average noise power. PSNR is defined as the ratio of the peak signal power to the average noise power.

The SNR and PSNR measures are mathematically tractable and have historical appeal. Much work already exists to minimize the l_2-norm of an error, such as the LMS algorithm in adaptive filtering [28] and rate-distortion theory [29]; the attraction of the l_2-norm is therefore great. However, the correlation between SNR or PSNR and visual quality is known to be poor [30]. Nevertheless, PSNR is almost universally quoted as a figure of merit for images. Furthermore, despite the fact that PSNR is a *noise* measure, and therefore should only be applied to images whose sole degradation is due to additive noise, it is used in the literature to evaluate images with degradations that are not noise-like. The blocking artifacts of the Joint Photographic Experts Group (JPEG) compression scheme operated at high compression rates, for instance, cannot be adequately quantified by PSNR; neither can the so-called "mosquito noise" of wavelet compression algorithms, since neither is additive noise. Yet PSNR is still quoted for the images produced by such schemes.

Ultimately, most images are intended for human consumption (although images processed automatically by computer vision algorithms are a notable exception). What is therefore required is an error measure which is correlated to visual difference. That is, a processed image which appears very similar to the original should have a small error relative to it. Furthermore, as visual quality degrades, the error should increase monotonically. Neither of these criteria is met by either SNR or PSNR.

The lack of a good alternative to PSNR is probably due in part to the

fact that many image distortions are possible, and characterizing each distortion in terms of its effect on visual quality, let alone actually determining the level of each distortion in a particular image, is daunting. Fortunately, some image processing operations result in an image being modified by a small set of characterizable distortions. The effect of each operation can then be quantified, allowing comparison between schemes which are attempting to achieve the same result. For instance, a block-based image compression scheme might be characterized by the level of blocking and the degree of blurring at a given bit rate. Block-based compression schemes could then be compared using these two criteria. In this chapter, the degradation of halftones and inverse halftones are separated into *noise injection* and *frequency distortion*. This allows both effects to be quantified, permitting comparison of competing schemes.

2.2 Human visual system

To devise a satisfactory measure of the visual quality of an image, it is necessary to understand the mechanisms involved in human vision. The human visual system (HVS) is a complicated, spatially-varying, non-linear system; distilling its multiple characteristics into a single equation, especially one that is linear, is a gross over-simplification. Nevertheless, experiments have been carried out that indicate that, over a limited range of inputs, the HVS can be treated as a linear system [31]. Certain visual anomalies can be at least partially explained by such a treatment. These include the nonlinear relationship between intensity and brightness, and the Mach band effect, which causes edges between large, uniform regions to appear sharper than they actually are. Furthermore, assuming that the HVS is linear leads to the simplification of

any analysis which depends on the response of the HVS to a particular stimulus. It is therefore reasonable to assess how applicable the linear model is to halftones and inverse halftones.

The front end of the HVS consists of an optical system composed of the cornea, iris, lens, and retina [32]. Incoming light is focused onto the retina by the cornea and lens, whose thickness is adjusted by the ciliary muscles to accommodate for object distance. The iris controls the amount of light entering the eye by varying the size of the aperture through which light passes. The retina is covered with a mosaic of photoreceptors, with the coverage being densest in a small region close to the visual axis known as the fovea. Electrical impulses generated by the photoreceptors in response to light are transmitted, via synaptic connections to bipolar and ganglion cells in the retina, down the optic nerve to the brain.

When an object is imaged by the eye, an inverted and reduced image of the object falls on the retina. The size of the retinal image is determined by the visual angle θ subtended by the object, given approximately by

$$\theta = \frac{l}{d} \text{ radians} , \tag{2.3}$$

where l is the size of the object, and d is the distance of the object from the nodal point of the eye. (This is effectively equal to the distance between the object and the observer for reasonable object distances.) The approximation in (2.3) stems from the fact that $\tan(\theta) \approx \theta$ for small values of θ.

As an object recedes from the viewer (i.e., as $d \to \infty$), the visual angle subtended at the eye by the object tends to zero. Consider a sine-wave grating

situated at $z = 0$ in the plane formed by the x and y axes of a Cartesian co-ordinate system. Let the intensity I of the grating be given by

$$I(x, y) = 1 + c \sin(\omega_g x) \ , \tag{2.4}$$

where c is the contrast of the grating ($0 \leq c \leq 1$), and ω_g is the angular frequency of the grating in radians/m. It is assumed without loss of generality that the grating intensity does not depend on y. Assume also that the observer moves along the z axis, oriented in such a way that he or she perceives the grating to be vertical. Since the grating is infinite, the observer will not see any change in the *size* of the grating as he or she moves; however, the *angular frequency* subtended by the grating at the observer's eye will change, in a reciprocal manner to (2.3). Specifically, when the observer is at a distance d from the grating, the angular frequency at the eye is given by

$$\begin{aligned} f_a &= \omega_g d \quad \text{radians/radian} \\ &= \frac{\omega_g d}{360} \quad \text{cycles/degree} \ . \end{aligned} \tag{2.5}$$

The wavelength of light, and the quality of the human optical system, place a limit on the resolving power of the eye, that is, on the maximum angular frequency that can be resolved. This limit occurs at about 60 cycles/degree [33]. Below this limit, gratings are resolved if they are of sufficient contrast. The contrast sensitivity function (CSF) is the contrast required to resolve a grating of a particular angular frequency. Under the assumption that the HVS is linear, the CSF corresponds to the transfer function (angular frequency response) of the system. It therefore determines the visibility of individual Fourier components of an image, as seen by a human viewer.

The CSF is measured using a two-alternative forced-choice method under *threshold conditions*, i.e., at signal levels which cause a response in the ganglion cells that is asymptotically zero. The HVS can be assumed to be the most linear at low signal levels; extrapolation of the CSF to normal (supra-threshold) viewing conditions is somewhat difficult to justify. However, the success of the CSF in explaining the non-linear relationship between brightness and intensity [31] suggests that the CSF model is justified under certain supra-threshold circumstances.

Several analytic approximations to the CSF have appeared in the literature [34, 35]. The CSF due to Mannos and Sakrison [34] is

$$H(f_r) = 2.6(0.0192 + 0.114 f_r)\exp(-(0.114 f_r)^{1.1}) \ , \tag{2.6}$$

where f_r is the radial angular frequency in cycles/degree, given by

$$f_r = \sqrt{f_x^2 + f_y^2} \ , \tag{2.7}$$

where f_x and f_y are angular frequencies in the x and y directions, respectively. The CSF of (2.6) is radially symmetric. A simple modification by Sullivan, Miller, and Pios [36] accounts for the mild drop in visual sensitivity in the diagonal directions. The angular modification of f_r is

$$f_r' = \frac{f_r}{s(\phi)} \ , \tag{2.8}$$

where ϕ is the angle measured from the x axis, defined by $\phi = \tan^{-1}(f_y/f_x)$. The function $s(\phi)$ is given by

$$s(\phi) = \frac{1-w}{2}\cos(4\phi) + \frac{1+w}{2} \ , \tag{2.9}$$

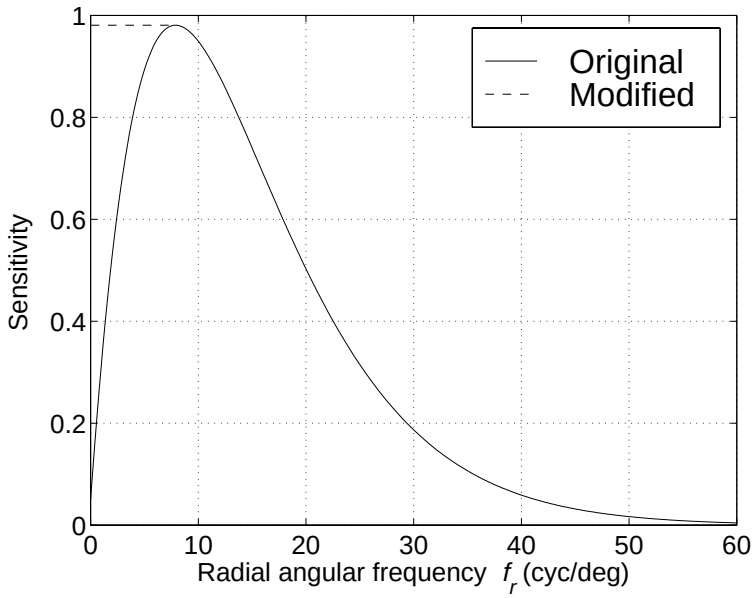


Figure 2.1: On-axis radial contrast sensitivity function. Solid: Original function due to Mannos and Sakrison [34]. Dotted: Modification due to Mitsa and Varkur [16].

where w, the *symmetry parameter*, is chosen to be 0.7. The $s(\phi)$ function varies from a value of 1 along the x and y axes to 0.7 along the lines defined by $y = \pm x$. Thus the effective radial frequency is increased somewhat off-axis, causing a faster decrease in visual sensitivity than along the axes. Figure 2.1 shows the CSF along the x and y axes.

A further modification was suggested by Mitsa and Varkur [16]. They advocate flattening the CSF at low angular frequencies to provide a lowpass, rather than a bandpass, characteristic. The modified CSF is shown by the dotted line in Figure 2.1. At high angular frequencies, the unmodified CSF drops off because of physical limitations imposed by the lens system of the human eye. The drop-off in contrast sensitivity for low frequencies, however, is due to *lateral inhibition* [37]. In the retina, lateral connections made by

horizontal and amacrine cells cause a reduction in the firing rate of a ganglion cell when its surrounding ganglion cells are exposed to the same stimulus, that is, when the stimulus has a low angular frequency.

However, the unmodified CSF is measured with the subject fixated at one point. When examining a real image, the viewer continually changes the point of fixation to examine features in the image. This movement introduces a temporal factor into the contrast sensitivity. Spatio-temporal CSFs have been published [33], and show that the CSF is flattened at low angular frequencies if the contrast of the stimulus varies slowly with time. Cornsweet [31] demonstrates the low contrast sensitivity to low angular frequencies when the fixation point is stationary, but also shows that even small movements of the fixation point restore the lost sensitivity. Furthermore, sharp edges in the image, which contain components at higher frequencies where the HVS is more sensitive, enhance the effect. The result is that contrast sensitivity does not fall off appreciably at low angular frequencies when a viewer is not forced to fixate at a single point. This is especially true when viewing halftones, since they contain large amounts of high frequency quantization noise, even in areas that were smooth in the original image [16]. Flattening the CSF at low angular frequencies is therefore justified. The two-dimensional CSF defined by (2.6) and (2.9), together with the flattening of Figure 2.1, is shown in Figure 2.2. The decreased sensitivity along the diagonals and the flattening at low angular frequencies are visible.

It is easy to demonstrate that the human CSF is not flat. The *lena* image in Figure 2.3(a) has been corrupted by Gaussian white noise, so that its

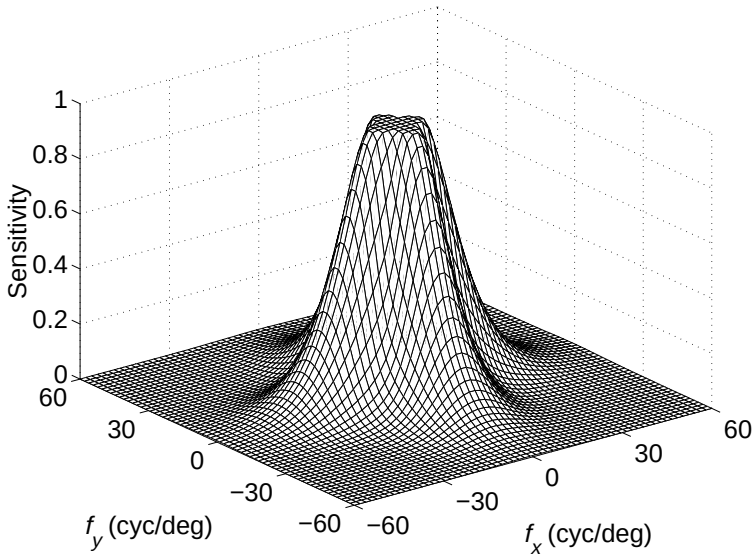


Figure 2.2: Two-dimensional contrast sensitivity function computed according to models of Mannos and Sakrison [34] (radial dependence) and Sullivan [36] (angular dependence).

(a) White noise.

(b) Highpass noise.

Figure 2.3: Effect of the frequency distribution of noise on its visibility. The SNR of both images is 10.0 dB. The PSNR of both images is 15.7 dB. At normal viewing distances, (a) is visibly noisier than (b).

SNR relative to the original image is 10.0 dB. The image in Figure 2.3(b) has been corrupted with highpass Gaussian noise (generated by filtering Gaussian white noise), so that its SNR relative to the original image is also 10.0 dB. At normal viewing distances, Figure 2.3(a) is visibly noisier than Figure 2.3(b), despite the fact that their SNRs are identical. This is because the bulk of the noise power in Figure 2.3(b) falls at higher frequencies, which are attenuated by the CSF. The subjective difference between the two images reduces as the images are brought closer to the eye, as predicted by the CSF of Figure 2.2.

2.3 Weighted noise measurements

Because the CSF is a function of angular frequency, the size and viewing distance of the image must be taken into account when determining the response of the HVS. For discretized images, such as those displayed on a computer screen or printed on paper, one can compute the maximum angular frequency at the retina for a given image and viewing distance. The arrangement is shown in Figure 2.4. The following analysis refers only to the horizontal direction. An analogous formulation applies to the vertical direction.

The angle subtended by the image at the eye in the horizontal direction is $\theta = 2\tan^{-1}(l/2d) \approx l/d$ radians, for small values of θ. The maximum angular frequency in the discrete image is termed the *Nyquist frequency*; at this frequency, neighboring pixels alternate from black to white, giving an angular frequency of one cycle per two pixels, or π radians per pixel. Since there are N pixels in the image horizontally, a component at the Nyquist frequency has $N/2$ cycles, or $N\pi$ radians, across the image. There are therefore $N\pi$ cycles

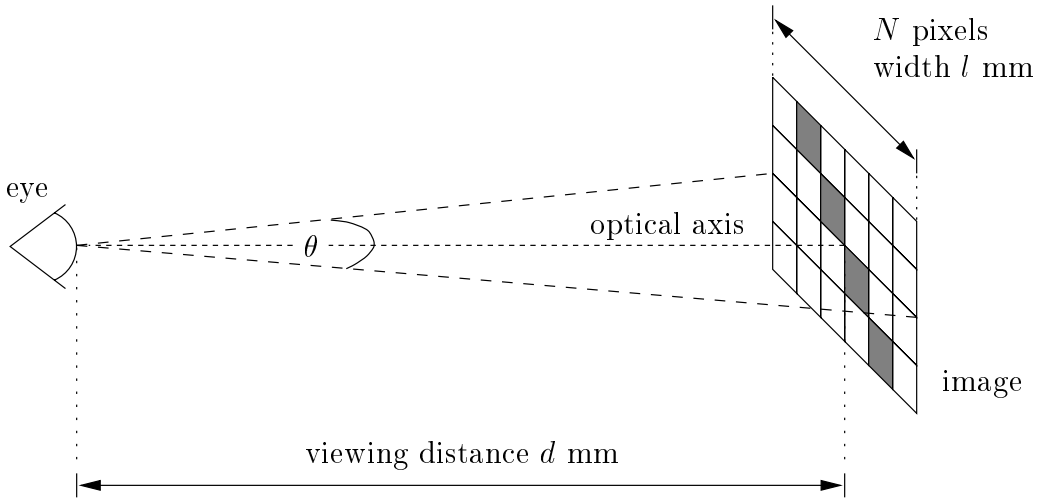


Figure 2.4: Computation of angular frequency at the eye. Horizontal (x) direction is shown; vertical (y) direction is analogous.

contained in an angle of l/d radians; the angular frequency is given by

$$\begin{aligned} f_a &= \frac{N\pi d}{l} \quad \text{radians/radian} \\ &= \frac{N\pi d}{360l} \quad \text{cycles/degree .} \end{aligned} \tag{2.10}$$

Thus a knowledge of the number of pixels in an image, the size of the image, and the viewing distance allows the maximum angular frequency at the eye to be computed. As an example, for an image of size 512×512 pixels, printed 100 mm on a side, and held at a normal viewing distance of 400 mm, the maximum angular frequency is approximately 18 cycles/degree.

By assuming that the HVS is linear, the effect on the viewer of a particular image component can be assessed using the following procedure. A two-dimensional discrete Fourier transform (DFT) of the image is performed. The maximum angular frequency of the image is computed using (2.10), and an appropriate CSF is constructed using (2.6), (2.8) and (2.9). The DFT of

the image is then multiplied point-for-point with the CSF, so that an image component at a particular angular frequency is weighted by the value of the CSF at that frequency. The result is the DFT of an image that would lead to the same response when viewed by a visual system with a flat CSF as the original image leads to when viewed by the HVS.

Given two versions of an image of size $M \times N$ pixels, one clean (denoted x) and the other corrupted by noise (denoted y), the weighted signal-to-noise ratio (WSNR) of the noisy image is computed as follows:

$$\text{WSNR (dB)} = 10 \log_{10} \left(\frac{\sum_{u,v} |(X(u,v)C(u,v)|^2}{\sum_{u,v} |(X(u,v) - Y(u,v))C(u,v)|^2} \right), \qquad (2.11)$$

where $X(u,v)$, $Y(u,v)$ and $C(u,v)$ represent the DFT of the input image, output image, and CSF, respectively, and $0 < u < M-1$, $0 < v < N-1$. In the same way that SNR is defined as the ratio of average signal power to average noise power, WSNR is defined as the ratio of average *weighted* signal power to average *weighted* noise power, where the weighting is derived from the CSF. Weighting is common in the audio industry, where the noise performance of devices is often measured by employing "A-weighting" [38]. This de-emphasizes the noise at high and low frequencies to account for the reduced sensitivity of the auditory system at the limits of the spectrum, giving a better measure of the true audibility of the noise. For images, the high spatial frequencies are de-emphasized using the CSF to give a better measure of the true visibility of the noise.

Table 2.1 shows computed values of the WSNR for the two images shown in Figure 2.3, for different viewing distances. The first column lists

Distance d (mm)	Maximum f_a (cyc/deg)	White WSNR (dB)	Highpass WSNR (dB)
200	6.9	10.1	10.3
400	13.7	11.7	13.6
600	20.6	13.9	18.0
800	27.5	16.1	22.4
1000	34.4	17.9	26.5

Table 2.1: Weighted SNR measurements for noisy *lena* images of Figure 2.3, relative to the original image. Normal viewing distance ≈ 400 mm.

the viewing distance in mm. The second column shows the angular frequency in cycles/degree corresponding to the Nyquist frequency for that viewing distance. The third and fourth columns list the computed WSNR measures for Figures 2.3(a) and 2.3(b), respectively.

At the shortest viewing distance of 200 mm, both images have a WSNR of approximately 10 dB, since the Nyquist frequency for this distance and image size is 6.9 cycles/degree, which is almost entirely inside the flattened passband of the CSF. The WSNR of both images increases with viewing distance, since the noise is attenuated by the dropoff in the CSF at high angular frequencies. However, the WSNR of the image corrupted with highpass noise increases faster, as expected.

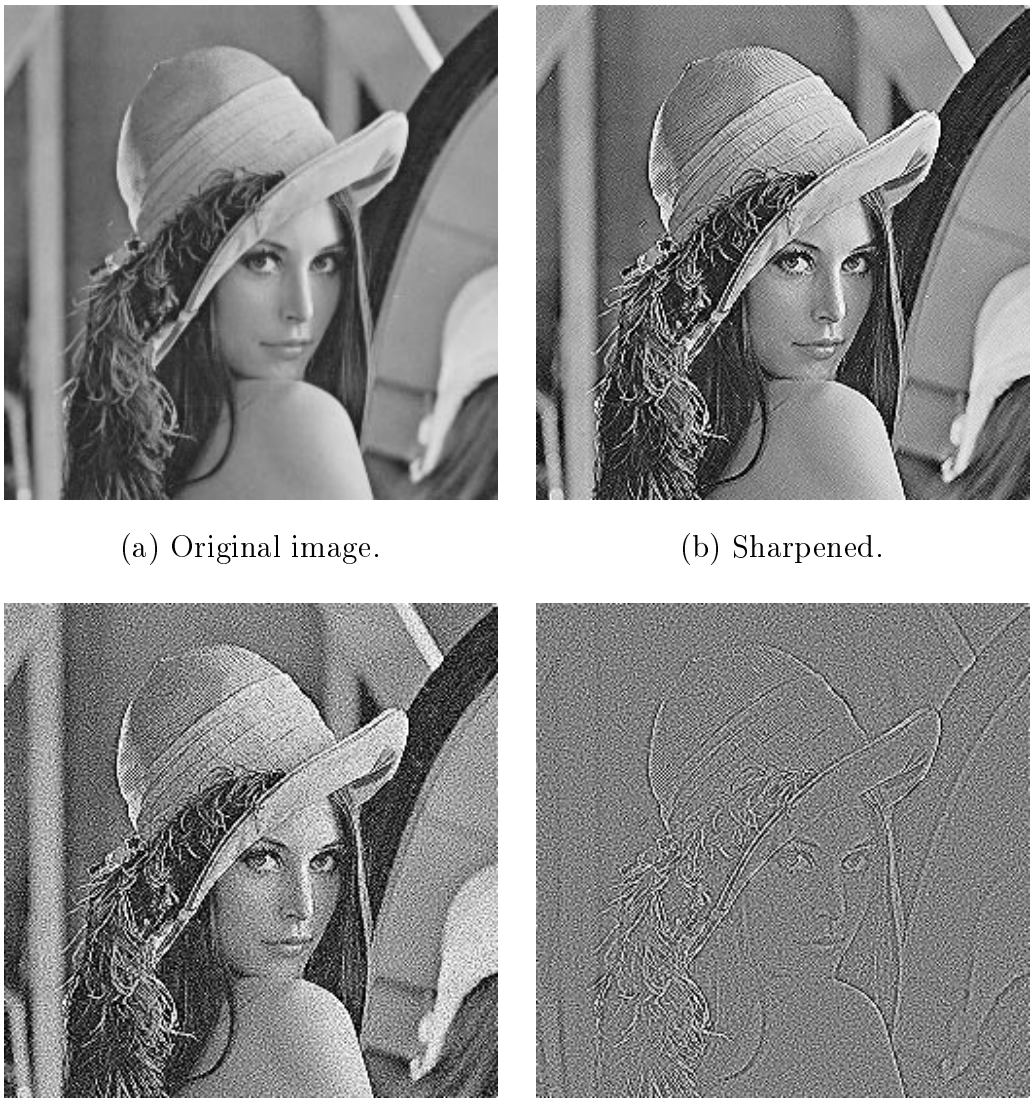
2.4 Accounting for other image degradations

As mentioned in Section 2.1, SNR and PSNR are commonly used as measures of image quality. Noise-based measurements are appropriate in situations where degradations are noise-like. For instance, a camera using a charge-coupled device (CCD) as the light-sensing element produces a noisy image

when operated under low-light conditions, because of the high gain needed in the video amplifier. It would therefore be appropriate to use a noise-based measure, such as WSNR, to assess the quality of images from the camera.

When an image has been corrupted by other factors as well as noise, it is necessary to account for these degradations before computing the WSNR; otherwise, they will be erroneously incorporated into the weighted noise figure. Figure 2.5 shows an example. Figure 2.5(a) is the original *lena* image. Figure 2.5(b) has been sharpened with a filter of size 3×3 pixels. (This amount of sharpening is similar to that seen in some error diffusion halftoning algorithms, as will be shown in Chapter 3.) Figure 2.5(c) shows the sharpened image with highpass noise added to give an SNR of 10.0 dB relative to the clean, sharpened image. Figure 2.5(d) shows the difference between Figure 2.5(c) and Figure 2.5(a). This difference image is referred to as the *residual*. Because it is correlated with the original image, and is therefore not signal-independent noise, it is inappropriate to compute the SNR (or PSNR, or WSNR) of Figure 2.5(c) relative to Figure 2.5(a). However, it *is* appropriate to compute a noise-based measure for Figure 2.5(c) relative to Figure 2.5(b), since the difference between them is noise that is independent of the original image.

Table 2.2 lists WSNR figures for the image in Figure 2.5(c) for five viewing distances. The third column shows the WSNR relative to Figure 2.5(a), while the fourth column shows the WSNR relative to Figure 2.5(b). As expected, the values in the third column are considerably lower than those in the fourth column, because the residual includes power from the original image. The WSNR figures relative to Figure 2.5(b) are correct, because the



(a) Original image.

(b) Sharpened.

(c) Sharpened + highpass noise.

(d) Residual (c) − (a).

Figure 2.5: Effect of sharpening on WSNR measurement. The residual (d) contains information from the original image (a), thereby making it unsuitable for use in a measurement of WSNR. The residual (c) − (b) consists of independent noise, and therefore can be used to compute WSNR.

Distance d (mm)	Maximum f_a (cyc/deg)	WSNR (dB)	
		Ref. original	Ref. sharpened
200	6.9	5.8	10.2
400	13.7	8.8	13.1
600	20.6	12.5	17.3
800	27.5	15.7	21.5
1000	34.4	18.1	25.6

Table 2.2: Measures of weighted signal-to-noise ratio computed using inappropriate (third column) and appropriate (fourth column) residuals for the images in Figure 2.5. The first and second columns show the viewing distance and maximum angular frequency, respectively. Figures in the third column were generated using a residual correlated with the original image. Figures in the fourth column were generated using an uncorrelated residual.

residual is uncorrelated with the original image. The results of Table 2.2 show the importance of removing as much image power as possible from the residual before computing the WSNR of an image.

2.4.1 Correlation of the residual with the original image

To quantify the degree to which a residual image R is correlated with an original image I, a correlation measure between them must be defined. The magnitude of the correlation coefficient, C_{RI}, is given by [39]

$$C_{RI} = \frac{|\mathrm{Cov}[R, I]|}{\sigma_R \, \sigma_I} \ , \tag{2.12}$$

where Cov refers to covariance, and σ_R and σ_I are the standard deviations of images R and I, respectively. An absolute value in the numerator ensures that $0 \leq C_{RI} \leq 1$, with 0 indicating no correlation, and 1 indicating linear correlation. Thus C_{RI} can be considered to be a measure of linear correlation

between two images. The covariance is defined as

$$\mathrm{Cov}[R, I] = E[(R - \mu_R)(I - \mu_I)] \ , \tag{2.13}$$

where $E[\cdot]$ denotes expectation, and μ_R and μ_I denote the means of R and I, respectively.

Ideally, a residual image consists of independent additive noise, and therefore has zero correlation with the original image. In practice, the correlation will not be exactly zero, and noise-based measures such as WSNR may be in error. It is therefore important to determine the effect on WSNR caused by varying degrees of correlation. To this end, two images were generated: an "original image" I, composed of lowpass filtered noise, and a white noise image N of the same size. A noisy, corrupted image J is created as follows:

$$J = \alpha\, I + N \ , \tag{2.14}$$

where α is a gain factor. The residual image R is given by $R = (\alpha - 1)I + N$. By choosing α, one can force a prescribed linear correlation between R and I. The correlation is measured for a given α, and the SNR and WSNR for J relative to I are computed.

Table 2.3 shows the results for values of α ranging from 1.000 to 1.030. As expected, the correlation C_{RI} increases, and the SNR and WSNR decrease, as G increases above 1. The WSNR falls by approximately 3 dB as the correlation increases from zero to 0.100. This large variation underlines the importance of keeping the correlation of the residual and the original image to a minimum, preferably $C_{RI} < 0.020$, for the WSNR figure to be accurate.

Gain α	C_{RI}	SNR (dB)	WSNR (dB)
1.000	0.000	28.0	31.7
1.005	0.019	27.9	31.6
1.010	0.038	27.7	31.1
1.015	0.058	27.4	30.5
1.020	0.077	27.0	29.7
1.025	0.096	26.5	28.9
1.030	0.115	26.0	28.0

Table 2.3: Variation of SNR and WSNR with correlation of the residual and the original image, C_{RI}. The WSNR is computed assuming a maximum angular frequency of 20 cycles/degree. The first row shows the actual values of SNR and WSNR for the given image, relative to the noiseless original. The other rows show SNR and WSNR for increasing correlation between the residual and the original image.

2.4.2 Application to error diffused halftones

It was shown in Chapter 1 that error diffused halftones have non-flat noise spectra, because of the noise shaping property of error diffusion. Meaningful perceptual noise figures for halftones can be obtained by using WSNR. The unweighted SNR of error diffused images is typically 1–2 dB, and the PSNR is 6–7 dB, regardless of the scheme used. These low figures stem from the one-bit quantization inherent to all halftoning schemes, and give no indication of visual quality. Different error diffusion schemes have different noise shaping properties, however, and WSNR is able to distinguish between them.

It was also mentioned briefly in Chapter 1 that an error diffused halftone is sharper than the original image, with the degree of sharpness being dependent on the error diffusion scheme. This sharpening will be examined further in Chapter 3. If the WSNR of an error diffused halftone is computed relative to the original image, the result will be in error, because the residual

between the sharpened halftone and the original is correlated with the original image. It is therefore necessary to remove the sharpening before computing the WSNR, as discussed in Section 2.4.1. In Chapter 3, a new model of error diffusion is developed that solves this problem in one of two ways: either by constructing a "clean" image that is sharpened in an identical way to the halftone, or by modifying the input image itself so that the resulting halftone is not sharpened. Both methods produce residuals having a low correlation with the original image. The correlation is lower for the second method, however, and it is therefore used exclusively to determine WSNR.

In [16] it was reported that predictions of halftone quality using the low-pass CSF presented in Section 2.2 correlated well with psychovisual measurements. By removing sharpening first, the applicability of WSNR is extended to halftones created by schemes that exhibit strong sharpening. To demonstrate this, conventional (sharpened) halftones of the *barbara* image were computed using the Floyd-Steinberg and Jarvis error filters. The Jarvis filter sharpens more than the Floyd-Steinberg filter, as shown in Figure 1.8(e). Unsharpened halftones were also created using the same error filters. Table 2.4 shows computed values of WSNR for these halftones, for various viewing distances.

For the Floyd-Steinberg halftones, the correlation between the original image and the halftone residuals was 0.030 for the sharpened halftone, and 0.001 for the unsharpened halftone. For the Jarvis images, the correlation was 0.094 for the sharpened halftone, and 0.025 for the unsharpened halftone. Table 2.4 shows that the discrepancy in WSNR between the sharpened halftones and the unsharpened halftones increases with maximum angular frequency,

Maximum	Floyd-Steinberg		Jarvis *et al.*	
f_a (cyc/deg)	W_{SH} (dB)	W_{NS} (dB)	W_{SH} (dB)	W_{NS} (dB)
20	8.1	8.2	5.7	6.0
30	14.9	15.2	11.1	11.9
40	21.1	21.6	16.5	18.0
50	26.0	26.9	21.1	23.9
60	29.7	31.0	24.6	29.3

Table 2.4: Weighted SNR measurements for halftoned *barbara* images at different viewing distances. W_{SH} is the WSNR between the conventional (sharpened) halftone and the original image. W_{NS} is the WSNR between the modified (non-sharpened) halftone and the original image.

and that this discrepancy is larger for the Jarvis filter than the Floyd-Steinberg filter, as expected. By using an accurate model for halftoning, one ensures that the WSNR figures are accurate. The WSNR measure is used in Chapters 3 and 5 to assess the quality of halftones.

2.4.3 Application to inverse halftones

It was mentioned in Chapter 1 that an inverse halftone is a grayscale image created from a halftone. It is blurred relative to the original image, and contains quantization noise whose spectrum has been shaped by both the halftoning and inverse halftoning processes. WSNR can be used to assess the perceptual effect of the shaped noise in an inverse halftone. The fact that an inverse halftone is blurred relative to the original image indicates that the blurring must be taken into account before the WSNR is computed, to avoid error. In Chapter 4, a model of inverse halftoning is presented that greatly reduces the correlation of the residual, thereby allowing the application of WSNR.

Modified Floyd-Steinberg error diffusion was used to create an unsharp-

Maximum f_a (cyc/deg)	WSNR (dB)	
	Ref. original	Ref. modeled
10	18.2	31.7
20	20.5	32.3
30	24.1	33.4
40	27.6	34.8
50	30.5	36.3

Table 2.5: Weighted SNR measurements for inverse halftoned *barbara* images at different viewing distances. The second column shows the WSNR relative to the original image. The third column shows the WSNR relative to the modeled inverse halftone.

ened halftone from the *barbara* image, and this halftone was then inverse halftoned. A model inverse halftone was also created which exhibits the blurring of the inverse halftone, but without the noise. The correlation of the original image and the residual between the inverse halftone and the original is 0.365, which is high enough to cause large errors in WSNR (see Table 2.3). The correlation of the original image and the residual between the inverse halftone and the model inverse halftone is 0.008.

Table 2.5 shows WSNR figures for the inverse halftone at various viewing distances. The second column shows the WSNR relative to the original image, while the third column shows the WSNR relative to the modeled inverse halftone. The large difference between the two WSNR figures shows that modeling the blur of inverse halftoning is extremely important to obtain true weighted noise measurements. An inverse halftone, being a grayscale image, is likely to be held closer to the eye than a halftone; halftones rely on the lowpass filtering action of the HVS to achieve high visual quality, whereas inverse halftones do not. Thus, the maximum angular frequency subtended at

the eye by an inverse halftone is likely to be lower than that of a halftone. It is in this region that the discrepancy between the two WSNR figures is at its greatest. The WSNR measure is used in Chapter 4 to assess the quality of inverse halftones.

2.5 Summary

A contrast sensitivity function (CSF) from the literature that has been shown to be a good predictor of visual quality for halftones has been modified for use with all error diffused halftones, including those produced by schemes that greatly sharpen the image. The CSF has also been applied to inverse halftones, which are blurred compared to the original image. A weighted signal-to-noise ratio (WSNR) is thereby obtained that is a measure of the perceptual impact on the human visual system of noise in the image. The technique relies on modeling the frequency shaping of the process in question, thus reducing the correlation of the residual with the original image. It was shown that this correlation must be close to zero to obtain an accurate perceptual noise figure, thus allowing schemes to be compared.

WSNR is dependent on the size of an image, the number of pixels it contains, and the viewing distance. To achieve high visual quality, halftones must be viewed so that the Nyquist frequency f_N subtends a large angular frequency f_a at the eye. The quantization noise is then greatly attenuated by the lowpass CSF of the human visual system. Inverse halftones have no such restriction, and typical maximum angular frequencies are likely to be lower than for halftones. This will be taken into account in subsequent chapters.

Chapter 3

Error Diffusion

Digital halftoning quantizes a grayscale image to one bit per pixel, and is a non-linear, spatially-varying system. In this chapter, a *linear gain model* for the quantizer in error diffusion halftoning systems is presented that permits analysis using linear methods. The model provides an accurate description of the two primary effects of error diffusion: edge sharpening and noise shaping. The accuracy of this model is demonstrated in three new ways.

As discussed in Chapter 2, it is necessary to account for distortions, such as sharpening or blurring, before computing the weighted signal-to-noise ratio (WSNR) of a processed image. This is important for error diffusion schemes which greatly sharpen the image. The linear gain model accurately quantifies and models this sharpening. By quantifying the sharpening, one obtains an objective measure of a subjective image enhancement; by modeling it, one obtains an accurate WSNR measure. In addition, a distortion metric can be computed which quantifies the degree of tonality in the halftone. Thus, the linear gain model permits *objective measures* of the *subjective quality* of halftones to be made. It also makes possible the design and analysis of novel halftoning schemes, which will be examined in Chapter 5.

3.1 Previous work

As explained in Chapter 1, error diffusion is a digital halftoning method which employs feedback to minimize the local weighted error introduced by quantization. The image is scanned and the current pixel is quantized by thresholding. The quantization error is subtracted from neighboring pixels in fixed proportions according to the *error filter*.

Error diffusion research can be classified into two broad groups: work aimed at improving the visual quality of halftones, and work aimed at analyzing the error diffusion process itself. The primary objection to the quality of error diffused halftones is the presence of visually annoying artifacts, such as idle tones. Section 3.1.1 describes approaches for reducing or eliminating these artifacts at minimal computational cost. A thorough understanding of error diffusion is essential to make improvements that are not purely *ad hoc*. Section 3.1.2 describes previous analyses of error diffusion.

3.1.1 Reducing artifacts in error diffused halftones

The performance of an error diffusion scheme depends on the choice of the error filter. Two factors drive its design: the need for high quality halftones, and the desire to minimize computational cost. That is, the smallest filter which achieves adequate visual quality is preferred. Computation can be reduced further if the filter coefficients are fixed-point, or if they are dyadic, i.e., if they can be applied using bit shifts rather than multiplications. In 1975, Floyd and Steinberg asserted that a four-coefficient filter was the smallest that gave good results [1], and this appears to have been verified by later work. As a

side benefit, the Floyd-Steinberg filter is dyadic.

In 1976, Jarvis, Judice and Ninke published a survey of halftoning methods which included an error diffusion scheme with a 12-coefficient error filter [15]. A similar filter was later published by Stucki [40]. The motivation behind these larger filters is to improve image quality by reducing directional artifacts in the image. These artifacts (or "worms"), which depend on the scan, can be broken up by using a different scan. For instance, the serpentine scan, which is similar to the raster scan except that even rows are scanned from right to left, can break up worms; however, this solution comes at the expense of creating other worms that did not exist with the raster scan [41]. The recursively defined Peano-Hilbert scan has also been used [42, 43], although its pseudo-random nature leads to halftones with a noisy appearance.

Worms are the result of the quantization error being correlated with the input signal. The quantization error can be decorrelated by dithering; however, this reduces the signal-to-noise ratio (SNR) at the output. For a multi-bit system with a large dynamic range, such as the compact disc audio standard, the loss of a few dB of SNR because of dither is worth the improvement in subjective quality obtained by decorrelating the quantization error [22]. For a critically sampled one-bit system such as error diffusion, the SNR is already so low that a dithered image may appear worse than one that is not dithered. Kolpatzik and Bouman's locally dithered error diffusion (LDED) adds dither only in smooth regions of the image to reduce contouring without greatly increasing the perceived noise level [44]. Because of computational considerations, however, it is more common not to use dither in halftoning.

Instead, the goal of a halftoning algorithm is to make the quantization error as visually benign as possible. The direct binary search (DBS) halftone of Figure 1.4 shows what can be achieved: the quantization error is not objectionable, and its Fourier transform is smooth and isotropic.

Ulichney showed that perturbing the weights of the error filter in a random fashion reduces worm artifacts and contouring [1]. Visual noise is increased, because perturbing the error filter is equivalent to dithering the system [45]. A beneficial side-effect of this scheme is that the size of the error filter can be reduced, thus lowering the cost of the algorithm. However, unless a table of pseudo-random numbers has been pre-computed and stored in memory, it may be more computationally expensive to generate and apply the random weights than to use standard error diffusion with a larger filter.

Knox and Eschbach reduced artifacts by modulating the quantizer threshold [46]. The threshold is usually set to mid-gray; by varying it about this point, artifacts can be reduced. Varying the threshold is equivalent to adding dither *at the quantizer*. Section 3.2 shows that the transfer function from the input of the quantizer to the output of the system is highpass. White noise added at the quantizer therefore becomes high frequency ("blue") at the output, thereby making it more pleasing to the eye [1]. Dithering at the quantizer is used extensively in delta-sigma modulation for audio [18].

Fan addressed the problem of directional artifacts by using a two-pass error diffusion technique, which distributes quantization error symmetrically in the horizontal direction [47]. This results in a halftone with a more isotropic distribution of dots in uniform areas of the image. The two-pass method

doubles the computational complexity of the algorithm.

Wong and Allebach design an optimal error filter using a model of the human visual system [48]. They begin by assuming that the quantization error can be modeled as additive white noise, and construct an error filter that minimizes its visual impact. They halftone the image with this error filter, and use the actual quantization error to compute a new error filter. This procedure is repeated until the change in the error filter from one iteration to the next falls below a threshold. Using a set of test images, they design a filter that is the same size as the Floyd-Steinberg filter, but gives better subjective results.

Wong used an adaptive technique to improve the quality of error diffused halftones [23]. At each pixel, the error filter is updated using the least mean squares (LMS) algorithm [28] to minimize a local error criterion. The resulting images are of high quality, but computational complexity is increased. Wong also used the technique to embed reduced-size halftones inside the halftone, which enables simple multiresolution rendering. This could be extended to embed data in a halftone, e.g., for identification or security purposes.

3.1.2 Analysis of error diffusion

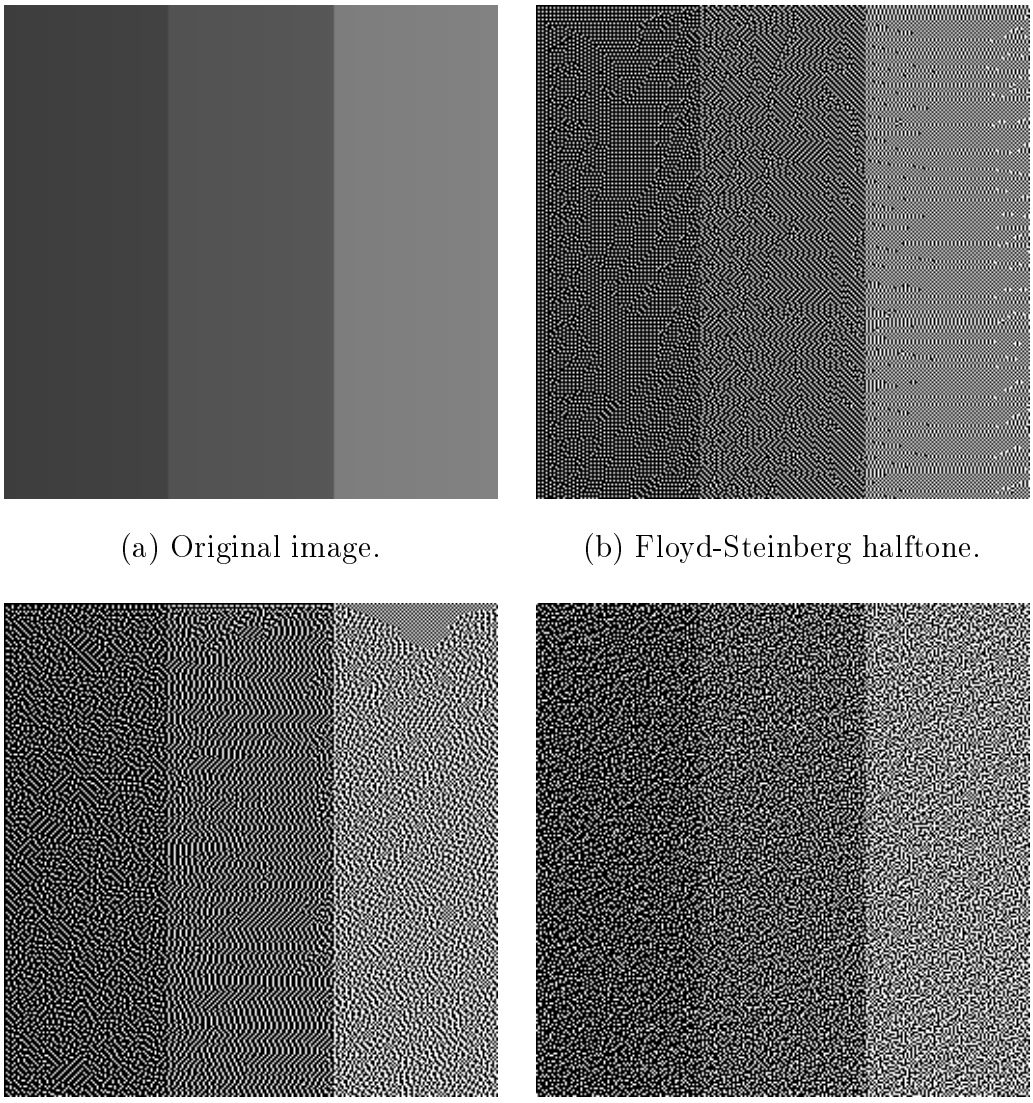
Following the 1989 paper by Anastassiou examining the analogy between error diffusion and delta-sigma modulation [20], Knox published results in 1992 which showed that the *error image* (the image composed of the quantization error at each pixel) is correlated with the input image. Section 3.2 presents a model for the quantizer which is derived from the assumption that the quantization error is additive white noise that is uncorrelated with the input. Knox

showed that this assumption is false, and noted that the sharpness of halftones increased as the correlation of the error image with the input increased.

In 1993, Knox published an analysis of error diffusion using a serpentine scan [49]. He showed that the serpentine scan results in a more symmetric error spectrum than the raster scan. This coincides with the fact that artifacts are less directional in serpentine-scanned halftones than in raster-scanned halftones. Fan analyzed the stability of error diffusion for generalized error filters [50]. Generally, the error filter coefficients are non-negative and sum to one to guarantee stability. Stability is not guaranteed for all inputs if these conditions are not met. One-dimensional delta-sigma modulators can suffer from instability, and steps must be taken to ensure that the system is stable for all expected input sequences [18]. Reducing the input level improves stability at the expense of SNR. This is not really an option in halftoning, since the SNR is already so low. Error diffusion schemes must therefore be stable for full-scale inputs.

The worms mentioned in Section 3.1.1 are also seen in audio applications of delta-sigma modulation. In audio, they are known as *limit cycles* or *idle tones*, since they result from the system cycling periodically through a finite set of states when the input is constant. If the period is long, then the tones fall in the audio band, where they are easily discerned by human listeners, even if they fall below the noise floor [19]. Part of audio delta-sigma modulator design is ensuring that limit cycles either do not occur (because of the modulator design itself, or because dither is used), or are inaudible [18].

In halftones, limit cycles appear as strong patterns. These patterns may



(a) Original image.

(b) Floyd-Steinberg halftone.

(c) Jarvis *et al.* halftone.

(d) Floyd-Steinberg dithered halftone.

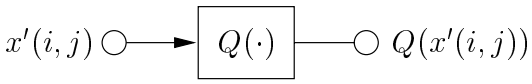
Figure 3.1: Limit cycles in error diffusion [51]. The original image is composed of three constant regions of graylevel $\frac{1}{4}$, $\frac{1}{3}$, and $\frac{1}{2}$, from left to right. Strong idle tones are visible in the undithered halftones (b) and (c).

not themselves be visually annoying, but when they change (e.g., because of a disturbance caused by noise) they are easily noticed, and can be interpreted by the viewer as false texture. Figure 3.1(a) shows a grayscale image composed of three constant regions. Figure 3.1(b) shows the Floyd-Steinberg halftone. Although the average graylevel in each region is faithfully reproduced, strong tones are visible. Two tones predominate in the leftmost region. In the middle region, a single, diagonal idle tone dominates. In the rightmost region, the checkerboard pattern is most common, although vertical stripes also appear.

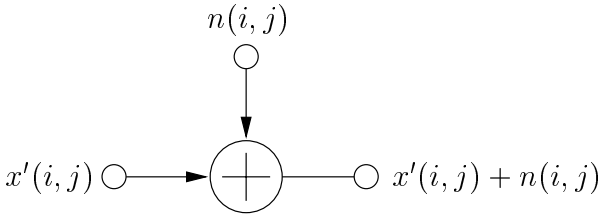
Figure 3.1(c) shows the effect of the larger error filter due to Jarvis *et al.* [15]. In 1994, Fan and Eschbach analyzed the limit cycle behavior of error diffusion [51]. They showed that the dominant tones for a particular constant input can be predicted from the transfer function of the error filter. These tones can be broken up by using a larger error filter, or by applying dither. The limit cycles produced by the Jarvis filter are reduced in the leftmost and rightmost regions of Figure 3.1(c), but are quite disturbing in the center region. The boundary between the checkerboard and the more random pattern at the top of the rightmost region is distracting. Figure 3.1(d) shows the result of using the Floyd-Steinberg filter, with dither having a triangular probability distribution function added at the quantizer [39]. The limit cycles have completely vanished, but the image is visually noisy.

3.2 Quantizer models

Quantized systems are non-linear, and are difficult to analyze, except for restricted classes of inputs. To obtain general results, it is necessary to model



(a) Quantizer.



(b) Linear model.

Figure 3.2: Quantizer (a) and the simple linear model (b). The quantizer is assumed to add white noise that is uncorrelated with the input signal.

the quantizer with a tractable element. In this section, two quantizer models are examined. Section 3.2.1 discusses a simple linear model, and shows that it fails to account for image sharpening. Section 3.2.2 introduces the *linear gain model*, which overcomes this deficiency. This model was used by Ardalan and Paulos in the one-dimensional case [52], but has not been applied to error diffusion previously.

3.2.1 Simple linear model

As a first approximation, the quantizer is treated as a linear element whose output is equal to the sum of its input and uniformly distributed, uncorrelated white noise, as shown in Figure 3.2. (This substitution will be referred to as the *uncorrelated white noise assumption.*) Referring to the noise shaping feedback coder shown in Figure 1.5, one obtains

$$e(i,j) \;=\; n(i,j) \tag{3.1}$$

$$x'(i,j) = x(i,j) - h(i,j) * e(i,j) \tag{3.2}$$

$$y(i,j) = x'(i,j) + n(i,j) \ . \tag{3.3}$$

By taking z-transforms of (3.1)–(3.3), one obtains

$$Y(\mathbf{z}) = X(\mathbf{z}) + N(\mathbf{z})(1 - H(\mathbf{z})) \ . \tag{3.4}$$

This is the linearized governing equation for error diffusion. The signal transfer function (STF), which is defined as $\dfrac{Y(\mathbf{z})}{X(\mathbf{z})}$, is unity. The noise transfer function (NTF), which is defined as $\dfrac{Y(\mathbf{z})}{N(\mathbf{z})}$, is given by $1 - H(\mathbf{z})$. This filtering effect is known as *noise shaping*. Since $H(\mathbf{z})$ is generally lowpass, the NTF is highpass. As was shown in Chapter 2, the human visual system can be modeled as a lowpass filter. By highpass filtering the quantization noise, its visibility is reduced, thereby improving the perceived image quality. The linearized equations (3.1)–(3.4) predict the following results:

- The difference, or *residual*, between the output image and the input image, $y(i,j) - x(i,j)$, is filtered noise uncorrelated with the input;
- The error image, $e(i,j)$, is white noise uncorrelated with the input; and
- The noise shaping function is given by $1 - H(\mathbf{z})$.

These predictions are examined for two filters: the Floyd-Steinberg filter defined in (1.5), and the filter due to Jarvis *et al.*, which was introduced in [15]. The coefficients of this filter are shown in Figure 3.3.

The first prediction is tested using the *bridge* image. Figure 3.4(a) shows the original image. Figures 3.4(b) and 3.4(c) show the Floyd-Steinberg and Jarvis halftones, respectively. Figures 3.4(d) and 3.4(e) show the corresponding residuals, and the correlation of these residuals with Figure 3.4(a),

		●	$\frac{7}{48}$	$\frac{5}{48}$
$\frac{3}{48}$	$\frac{5}{48}$	$\frac{7}{48}$	$\frac{5}{48}$	$\frac{3}{48}$
$\frac{1}{48}$	$\frac{3}{48}$	$\frac{5}{48}$	$\frac{3}{48}$	$\frac{1}{48}$

Figure 3.3: Error filter due to Jarvis *et al.* [15]. The black disk indicates the current pixel.

computed using (2.12). Both residuals are correlated with the input, because the halftones are sharper than the original image. Thus, the first prediction is not met, although the correlation is small for the Floyd-Steinberg filter.

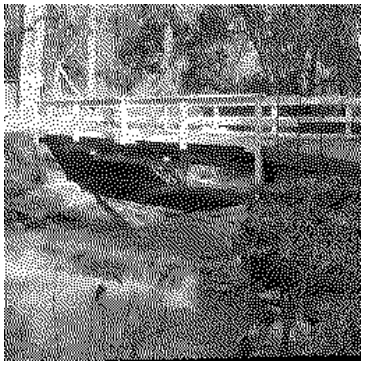
The second prediction is examined in Figure 3.5. The Floyd-Steinberg error image is shown in Figure 3.5(a), while the Jarvis error image appears in Figure 3.5(b). Both images are highly correlated with the input, as the correlation coefficients show. Thus, the second prediction is not met. The correlation of the error image with the input was first noted by Knox [53].

The third prediction is tested as follows. A noise image is halftoned, and the NTF is estimated by dividing the discrete Fourier transform (DFT) of the residual by the DFT of the error image. This is repeated for N images, and the results averaged:

$$\mathrm{NTF} = \frac{1}{N} \sum_{n=1}^{N} \frac{\mathrm{DFT}\left[(y_n - x_n)\right]}{\mathrm{DFT}\left[(y_n - x'_n)\right]} \; . \tag{3.5}$$

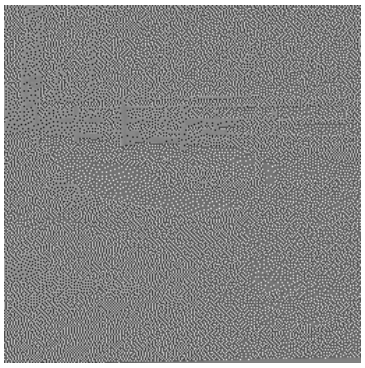
Figure 3.6 compares the measured NTF with the prediction of $1 - H(\mathbf{z})$. Figures 3.6(a) and 3.6(c) show the predicted NTFs for the Floyd-Steinberg and Jarvis schemes, respectively. Figures 3.6(b) and 3.6(d) show the corresponding measured NTFs. Both schemes show excellent agreement. This concurs with data from one-dimensional quantizers [18].

(a) Original *bridge* image.

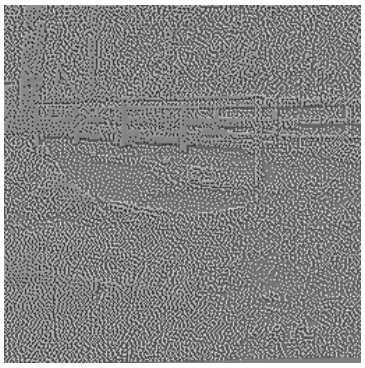


(b) Floyd-Steinberg halftone.

(c) Jarvis *et al.* halftone.

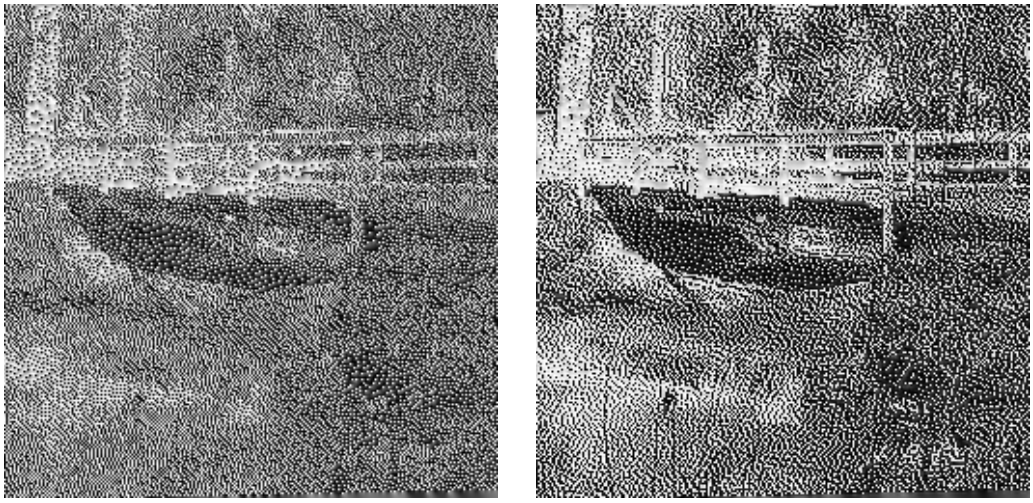


(d) Residual (b) − (a). $C_{RI} = 0.029$.



(e) Residual (c) − (a). $C_{RI} = 0.093$.

Figure 3.4: Residual images from error diffused halftones.



(a) Floyd-Steinberg. $C_{RI} = 0.309$. (b) Jarvis *et al.* $C_{RI} = 0.438$.

Figure 3.5: Error images from error diffused *bridge* halftones. The residual covers the range $(-0.5, 0.5)$; it is brought into the range $(0, 1)$ by adding 0.5. C_{RI} is the correlation of the residual with the input.

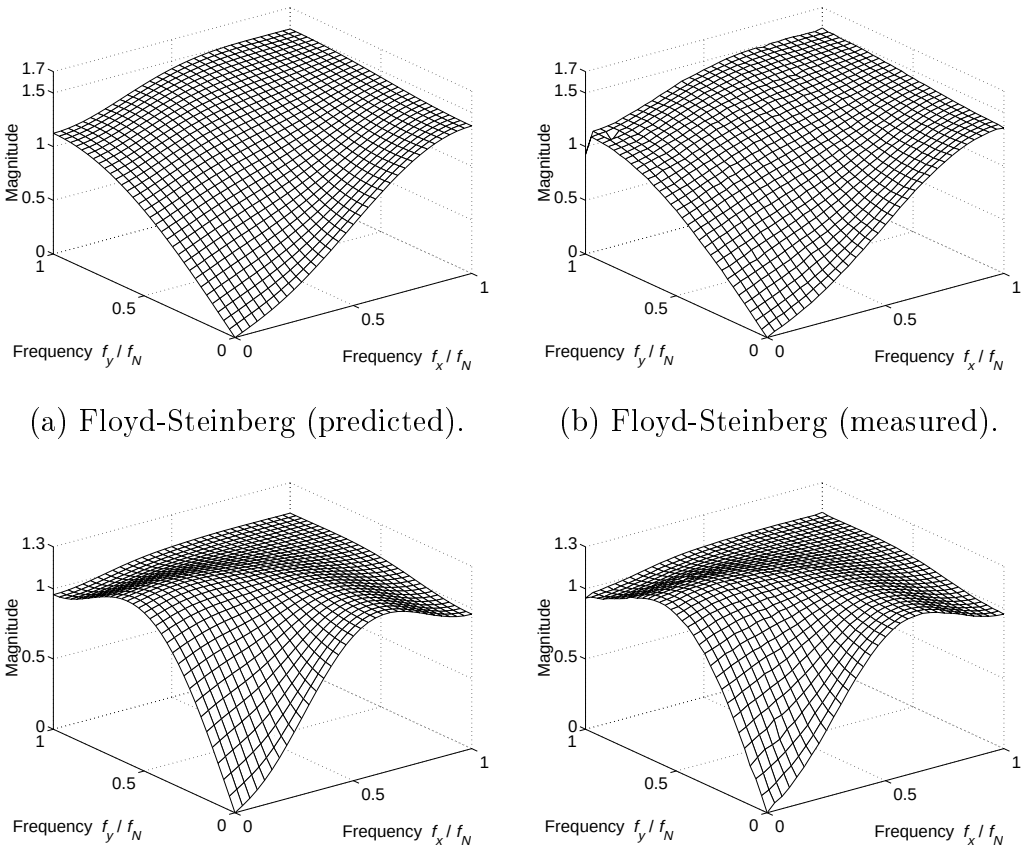
Two of the three predictions of the uncorrelated white noise assumption are therefore not met. Both of these predictions rely on the error introduced by the quantizer being uncorrelated with the input, which is clearly not true. The correlated nature of quantization error is in fact well known [18, 54]. One must therefore find an alternative model for the quantizer.

3.2.2 Linear gain model

The error images in Figure 3.5 are correlated with the input. The error image is given by $e(i,j) = y(i,j) - x'(i,j)$, which can be rewritten as

$$e(i,j) = Q(x'(i,j)) - x'(i,j) \ . \tag{3.6}$$

Since $e(i,j)$ is correlated with $x(i,j)$, and $x(i,j)$ is correlated with $x'(i,j)$ from (3.2), it follows that $e(i,j)$ is correlated with $x'(i,j)$. From (3.6), this



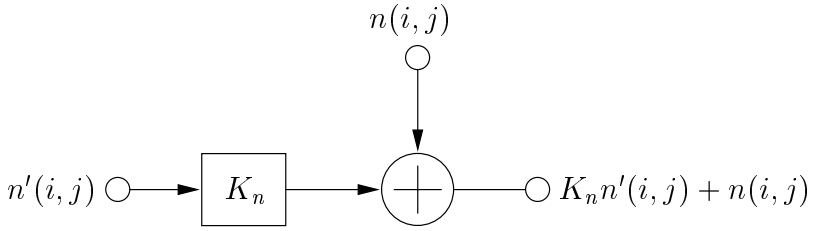
(a) Floyd-Steinberg (predicted).

(b) Floyd-Steinberg (measured).

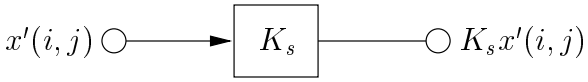
(c) Jarvis *et al.* (predicted).

(d) Jarvis *et al.* (measured).

Figure 3.6: Predicted and measured noise transfer functions. The predictions are derived from $1 - H(\mathbf{z})$, where $H(\mathbf{z})$ is the z-transform of the error filter. The measured responses are averaged over 5000 images. Mean squared error: 0.0090 (Floyd-Steinberg), 0.0056 (Jarvis *et al.*).



(a) Noise path.



(b) Signal path.

Figure 3.7: 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.

implies that $Q(x'(i,j)) - x'(i,j)$ is correlated with $x'(i,j)$. One can model this correlation if one assumes that

$$Q(x'(i,j)) = Kx'(i,j) + n(i,j) \ , \tag{3.7}$$

where K is a constant to be determined, and $n(i,j)$ is independent white noise. As K increases above 1, the correlation between $Q(x'(i,j))$ and $x'(i,j)$ increases. The relation in (3.7) models the quantizer as a cascade of a gain block of gain K and an additive, uncorrelated white noise source. For generality, the input to the quantizer is conceptually separated into signal and noise components, and gains K_s and K_n are assigned to the signal path and noise path, respectively. The quantizer model is shown in Figure 3.7.

This model is inserted into the noise shaping feedback coder shown in Figure 1.5 by using independent circuits for the signal and the noise. This idea was used by Ardalan and Paulos to model quantizers embedded in delta-sigma

modulators [52]. Analysis of the signal path leads to

$$\begin{aligned} e(i,j) &= (K_s - 1)x'(i,j) && (3.8)\\ x'(i,j) &= x(i,j) - h(i,j) * e(i,j) && (3.9)\\ y_s(i,j) &= K_s x'(i,j)\ , && (3.10) \end{aligned}$$

where $y_s(i,j)$ refers to the component of the output due to the signal. The signal transfer equation is obtained by taking z-transforms of (3.8)–(3.10):

$$Y_s(\mathbf{z}) = \frac{K_s}{1 + (K_s - 1)H(\mathbf{z})} X(\mathbf{z})\ . \tag{3.11}$$

Analysis of the noise circuit leads to

$$\begin{aligned} e(i,j) &= (K_n - 1)n'(i,j) + n(i,j) && (3.12)\\ n'(i,j) &= -h(i,j) * e(i,j) && (3.13)\\ y_n(i,j) &= K_n n'(i,j) + n(i,j)\ , && (3.14) \end{aligned}$$

where $y_n(i,j)$ refers to the component of the output due to the noise. The noise transfer equation is obtained by taking z-transforms of (3.12)–(3.14):

$$Y_n(\mathbf{z}) = \frac{1 - H(\mathbf{z})}{1 + (K_n - 1)H(\mathbf{z})} N(\mathbf{z})\ . \tag{3.15}$$

The transfer equation for the system is given by the sum of (3.11) and (3.15):

$$Y(\mathbf{z}) = \underbrace{\frac{K_s}{1 + (K_s - 1)H(\mathbf{z})}}_{\text{STF}} X(\mathbf{z}) + \underbrace{\frac{1 - H(\mathbf{z})}{1 + (K_n - 1)H(\mathbf{z})}}_{\text{NTF}} N(\mathbf{z})\ , \tag{3.16}$$

where STF and NTF are the signal and noise transfer functions, respectively, and constants K_s and K_n are still to be determined.

Referring to (3.15), one can see that if $K_n = 1$, one recovers the uncorrelated white noise result of (3.4), namely, that $\dfrac{Y(\mathbf{z})}{N(\mathbf{z})} = 1 - H(\mathbf{z})$. Section

3.2.1 shows that the uncorrelated white noise assumption accurately predicts the noise spectrum. Therefore, $K_n = 1$.

The signal gain K_s refines the linearization based on the uncorrelated white noise assumption, which has been shown to be inaccurate. Physically, the value of K_s at any pixel is given by the ratio of the output of the quantizer to its input. Because the input to the quantizer may vary continuously over a finite range, whereas the output is binary, K_s varies with the input. Thus a model which assumes a constant K_s must be in error to some extent. Nevertheless, by finding a value for K_s that minimizes the mean-squared error between the true halftone and the output of the model, progress can be made.

A halftone is related to the quantizer input by K_s (3.10). An image is halftoned, and the quantizer input is saved. A least-squares fit of $x'(i,j)$ to $y(i,j)$ is computed; this gives the value of K_s which leads to the minimum squared error between the halftone and the model output in a global image sense. In the following analysis, the output of the quantizer is assumed to be in the set $\{-0.5, 0.5\}$ rather than $\{0, 1\}$ to simplify the mathematics.

The quantizer output is ± 0.5. Consider pixels where the output is positive. The squared error over these pixels is minimized by finding

$$\min_{K_s} \left(\sum_{i,j} (K_s x'(i,j) - 0.5)^2 \right) \quad \forall\, (i,j) \text{ s.t. } y(i,j) = 0.5 \ . \tag{3.17}$$

Differentiating (3.17) with respect to K_s gives

$$\sum_{i,j} 2\,(K_s x'(i,j) - 0.5)\, x'(i,j) = 0 \ , \tag{3.18}$$

which leads to

$$K_s = 0.5\, \frac{\sum_{i,j} x'(i,j)}{\sum_{i,j} x'(i,j)^2} \ . \tag{3.19}$$

Input Image	Error filter		
	Floyd-Steinberg	Jarvis *et al.*	Stucki
barbara	2.01	3.76	3.62
boats	1.98	4.93	4.28
lena	2.09	5.32	4.49
mandrill	2.03	3.45	3.38
Average	2.03	4.37	3.94

Table 3.1: Computed values of the optimum quantizer signal gain K_s for various error filters and test images. Image size is 512×512 pixels.

For values of (i, j) for which the output is negative, the sign of the numerator in (3.19) changes. By combining (3.19) and the equivalent equation for negative outputs, one obtains

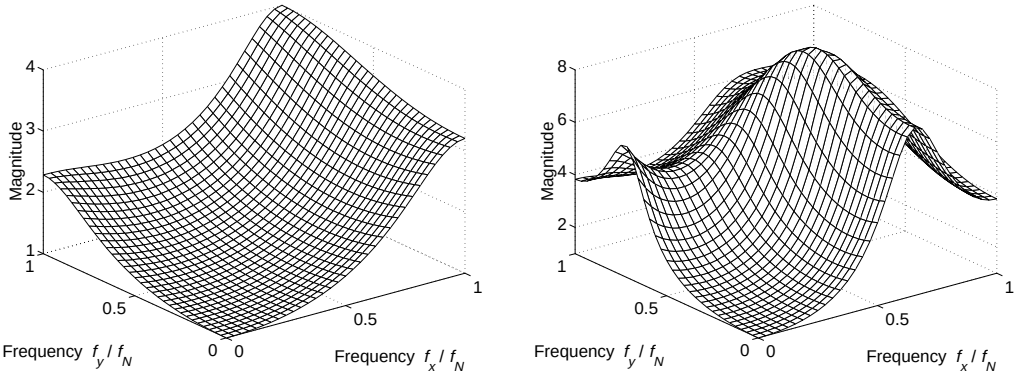
$$K_s = 0.5 \frac{\sum_{i,j} |x'(i,j)|}{\sum_{i,j} x'(i,j)^2} \quad \forall\,(i,j)\ . \tag{3.20}$$

This can also be expressed as

$$K_s = \frac{E[|x'(i,j)|]}{2E[x'(i,j)^2]}\ , \tag{3.21}$$

where $E[\cdot]$ denotes expectation. Measurements for four test images and three error diffusion filters are shown in Table 3.1. The value of K_s varies somewhat from image to image for a given error filter, although it is quite stable for images produced by Floyd-Steinberg error diffusion.

The STFs for two error filters computed using (3.11) are shown in Figure 3.8. Both have unity gain at DC; the gain rises at high frequency to 4 for the Floyd-Steinberg STF and 9 for the Jarvis STF. This qualitatively explains the image sharpening inherent to error diffusion. It must now be examined whether there is also good quantitative agreement.



(a) Floyd-Steinberg. $K_s = 2.00$. (b) Jarvis *et al.* $K_s = 4.37$.

Figure 3.8: Signal transfer functions computed from (3.11) using average values for the signal gain from Table 3.1.

3.3 Validation of the linear gain model

The linear gain model predicts an STF that is dependent on K_s and $H(\mathbf{z})$. Three ways of determining the accuracy of this model are presented. In Section 3.3.1, the model is used to generate a sharpened original image, and the correlation coefficient of the residual and the original image, computed using (2.12), is shown to be small. In Section 3.3.2, modified error diffusion, which introduces a sharpness parameter, is presented. The linear gain model is used to set this parameter to give an unsharpened halftone, whose residual has a low correlation with the original image. In Section 3.3.3, a frequency domain approach is used to examine the reduction in correlation. Section 3.3.1 is an extension of the work presented in [55].

3.3.1 Validation by constructing a sharpened original

Given an image and an error diffusion scheme, a halftone is constructed and the optimal value of K_s is computed from (3.21). The *original* image is then

Residual Image	Correlation Coefficient $C_{\text{original,difference}}$				
	barbara	*boats*	*bridge*	*lena*	*mandrill*
Halftone − Original	0.124	0.077	0.093	0.060	0.227
Halftone − Model, $K_s = K_{\text{ave}}$	0.099	0.033	0.022	0.020	0.163
Halftone − Model, $K_s = K_{\text{opt}}$	0.048	0.025	0.004	0.019	0.021

Table 3.2: Correlation coefficients for gain model residuals for the Jarvis *et al.* filter. The first row shows the correlation of the original image and the (halftone − original) residual image. The next two rows show the correlation of the original image and the (halftone − gain model) residual image, using the average K_s for this filter (K_{ave}), and the optimum K_s for this filter and each image (K_{opt}).

processed using the equivalent circuit of Figure 1.5, with the signal-only gain model substituted for the quantizer. This modifies the image using the STF of the error diffusion scheme without adding quantization noise. A "clean" image is created that has the sharpness of the halftone. The residual between this image and the halftone should therefore be quantization noise.

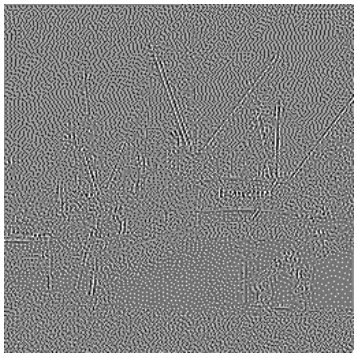
Figure 3.9 shows the results from this test. The original image is shown in Figure 3.9(a). Figure 3.9(b) shows the Jarvis halftone. There is a noticeable increase in sharpness over the original image, which is especially visible around the masts of the boat in the foreground. Figure 3.9(c) shows the image processed by the gain model. It has similar sharpness to the halftone. For this figure, $K_s = 4.93$, which is the optimal value for this image in the mean-squared sense. Figure 3.9(d) shows the residual between the halftone and the processed image. Figure 3.9(e) shows the image processed with $K_s = 4.37$, the average value for the Jarvis filter from Table 3.1. Figure 3.9(f) shows the corresponding residual.

Table 3.2 shows computed values of three correlation coefficients for five

(a) Original *boats* image.

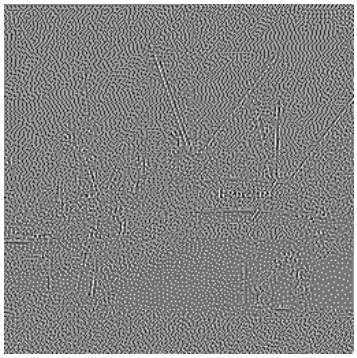
(b) Halftone.

(c) Gain model. $K_s = 4.93$.



(d) Residual (c) $-$ (b). $C_{RI} = 0.025$.

(e) Gain model. $K_s = 4.37$.



(f) Residual (e) $-$ (b). $C_{RI} = 0.033$.

Figure 3.9: Gain model validation using the Jarvis error filter. K_s is the quantizer gain. C_{RI} is the correlation coefficient for the residual.

test images. In terms of the images of Figure 3.9, they are, in order: between (a) and (b) − (a), between (a) and (f), and between (a) and (d). For all five test images, the correlation between the original image and the (halftone − original) residual is higher than the correlation between the original image and the (halftone − original modified by the gain model) residual, with K_{opt} giving a slightly lower correlation than K_{ave}.

Using images sharpened by the gain model, the correlation of the original to the residual is, on average, 0.51 that of the original to the unmodified residual when $K_s = K_{\text{ave}}$. When $K_s = K_{\text{opt}}$, the average correlation of the original to the residual falls to 0.25 that of the original to the unmodified residual. The reduction in correlation of the residual indicates that sharpening is accurately modeled by the linear gain model.

3.3.2 Validation by constructing an unsharpened halftone

In 1991, Eschbach and Knox published a method to control the sharpening of error diffusion by means of a multiplicative parameter L [56]. Positive values of L increase sharpening over the unmodified output, while negative values decrease sharpening. Because only an extra multiplication and addition per input pixel are required, it is a computationally simple way to adjust sharpness. Later, Knox and Eschbach published work on threshold modulation, and included an analysis of the sharpening technique [46]. Here, the technique, referred to as *modified error diffusion*, is analyzed using the notation of this chapter, and used to corroborate the linear gain model.

The modified error diffusion algorithm is shown in Figure 3.10. It will

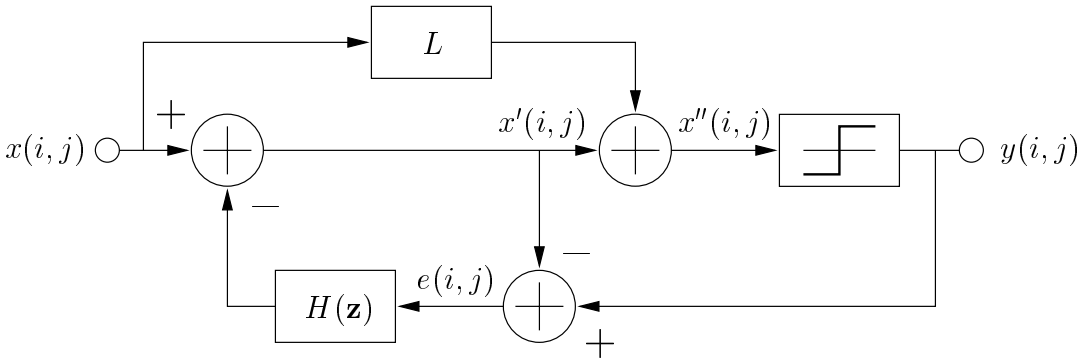


Figure 3.10: Modified error diffusion circuit for sharpness manipulation due to Eschbach and Knox [56]. The parameter L controls the degree of sharpening. The circuit reduces to standard error diffusion when $L = 0$.

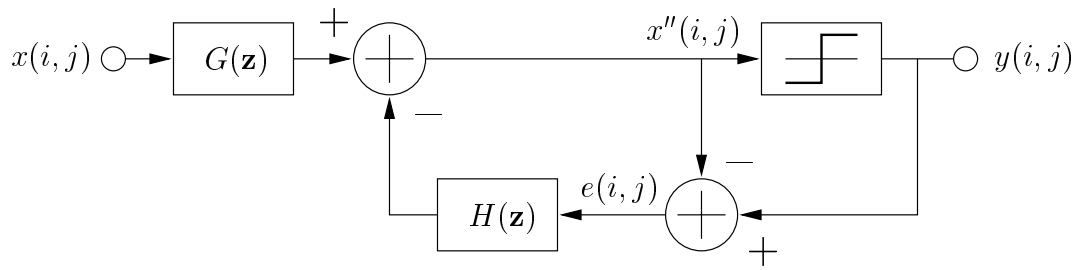


Figure 3.11: Modified error diffusion equivalent circuit. $G(\mathbf{z})$ is a pre-equalizer whose form is dependent on L and $H(\mathbf{z})$.

be shown to be equivalent to the circuit shown in Figure 3.11, with $G(\mathbf{z})$ being a function of L and $H(\mathbf{z})$. From Figure 3.10,

$$\begin{aligned} e(i,j) &= y(i,j) - x'(i,j) && (3.22)\\ x'(i,j) &= x(i,j) - h(i,j) * e(i,j) && (3.23)\\ x''(i,j) &= x'(i,j) + L\, x(i,j) && (3.24)\\ y(i,j) &= Q(x''(i,j)) \ . && (3.25) \end{aligned}$$

Combining (3.23) and (3.24), and taking z-transforms, leads to

$$X''(\mathbf{z}) = X(\mathbf{z})(1+L) - H(\mathbf{z})E(\mathbf{z}) \ . \tag{3.26}$$

Combine (3.22) and (3.23) and taking the z-transform gives

$$E(\mathbf{z}) = \frac{Y(\mathbf{z}) - X(\mathbf{z})}{1 - H(\mathbf{z})} \ . \tag{3.27}$$

Combining (3.26) and (3.27) leads to

$$X''(\mathbf{z}) = X(\mathbf{z}) \underbrace{\left(L + \frac{1}{1 - H(\mathbf{z})} \right)}_{M(\mathbf{z})} - Y(\mathbf{z}) \underbrace{\left(\frac{H(\mathbf{z})}{1 - H(\mathbf{z})} \right)}_{M'(\mathbf{z})} \ . \tag{3.28}$$

Let $m(i,j)$ and $m'(i,j)$ denote the inverse z-transforms of $M(\mathbf{z})$ and $M'(\mathbf{z})$, respectively. Now take the inverse z-transform of (3.28):

$$x''(i,j) = m(i,j) * x(i,j) - m'(i,j) * Q(x''(i,j)) \ , \tag{3.29}$$

and apply (3.25) to see that

$$y(i,j) = Q\left(m(i,j) * x(i,j) - m'(i,j) * Q(x''(i,j)) \right) \ . \tag{3.30}$$

This is the output for the modified system shown in Figure 3.10.

The equivalent circuit of Figure 3.11 can be analyzed in a similar way:

$$e(i,j) = y(i,j) - x''(i,j) \tag{3.31}$$

$$x'(i,j) = g(i,j) * x(i,j) - h(i,j) * e(i,j) \tag{3.32}$$

$$y(i,j) = Q(x''(i,j)) \ , \tag{3.33}$$

where $g(i,j)$ is the impulse response of the pre-equalizer $G(\mathbf{z})$. Combining (3.31) and (3.32) and taking z-transforms leads to

$$E(\mathbf{z}) = \frac{Y(\mathbf{z}) - G(\mathbf{z})X(\mathbf{z})}{1 - H(\mathbf{z})} \ . \tag{3.34}$$

Inserting this into the z-transform of (3.32) gives

$$X''(\mathbf{z}) = X(\mathbf{z}) \underbrace{\left(\frac{G(\mathbf{z})}{1 - H(\mathbf{z})} \right)}_{N(\mathbf{z})} - Y(\mathbf{z}) \underbrace{\left(\frac{H(\mathbf{z})}{1 - H(\mathbf{z})} \right)}_{M'(\mathbf{z})} \ . \tag{3.35}$$

Let $n(i,j)$ denote the z-transform of $N(\mathbf{z})$. Take the inverse z-transform of (3.35) and make use of (3.33) to see that

$$y(i,j) = Q\left(n(i,j) * x(i,j) - n'(i,j) * Q(x''(i,j)) \right) \ , \tag{3.36}$$

which is identical to (3.30) if the impulse responses $m(i,j)$ and $n(i,j)$ are the same. From (3.28) and (3.35), this condition is satisfied when

$$L + \frac{1}{1 - H(\mathbf{z})} = \frac{G(\mathbf{z})}{1 - H(\mathbf{z})} \ , \tag{3.37}$$

or, equivalently,

$$G(\mathbf{z}) = 1 + L\left(1 - H(\mathbf{z})\right) \ . \tag{3.38}$$

Thus, halftoning an image with the modified circuit is *exactly* equivalent to halftoning a version of the image that has been pre-filtered by the function $G(\mathbf{z}) = 1 + L\left(1 - H(\mathbf{z})\right)$.

The linear gain model predicts the STF given by (3.11). If $G(\mathbf{z})$ is made equal to the reciprocal of this STF, then the composite STF of the system will be flat. This is achieved when

$$1 + L\,(1 - H(\mathbf{z})) = \frac{1 + (K_s - 1)H(\mathbf{z})}{K_s}\ , \tag{3.39}$$

or, equivalently,

$$L = \frac{1 - K_s}{K_s}\ . \tag{3.40}$$

This allows the gain model to be corroborated. Values of K_s from Table 3.1 are used to compute L from (3.40), and images are halftoned using the circuit of Figure 3.10. If the gain model is accurate, the STF will be flat, and the residual between the original image and the halftone will consist solely of noise.

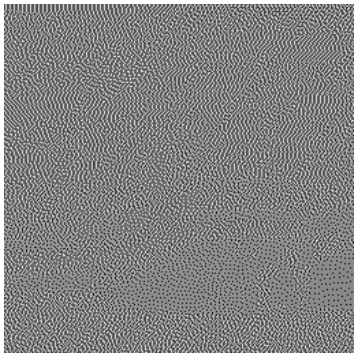
Figures 3.12(a) and 3.12(b) show the original image and its Jarvis halftone, respectively. Figure 3.12(c) shows the modified halftone using $L = -0.80$ ($K_s = 4.93$). Its sharpness is similar to the original image, but its quantization noise structure is similar to Figure 3.12(b). Figure 3.12(d) shows that components of the original image are at a very low level in the residual (c) $-$ (a). Figure 3.12(e) shows the modified halftone using $L = -0.77$, computed from the average K_s from Table 3.1. Figure 3.12(f) shows the corresponding residual. It also consists almost entirely of noise.

Table 3.3 shows computed values of the correlation coefficient for various images and residuals. The trend is similar to that of Table 3.2, except that the reduction in correlation using modified error diffusion is substantially larger than that obtained using the gain model alone. On average, the correlation of the original to the residual is 0.11 that of the original to the unmodified

(a) Original *boats* image.

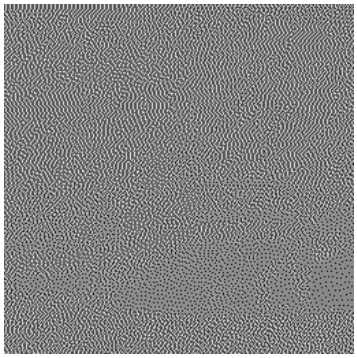
(b) Halftone.

(c) Mod. error diffusion. $L = -0.80$.



(d) Residual (c) $-$ (a). $C_{RI} = 0.005$.

(e) Mod. error diffusion. $L = -0.77$.



(f) Residual (e) $-$ (a). $C_{RI} = 0.008$.

Figure 3.12: Gain model validation using modified Jarvis error diffusion. L is the sharpness parameter. C_{RI} is the correlation coefficient for the residual.

Residual Image	Correlation Coefficient $C_{\text{original,difference}}$				
	barbara	*boats*	*bridge*	*lena*	*mandrill*
Halftone − Original	0.124	0.077	0.093	0.060	0.227
Halftone − Model, $L = L_{\text{ave}}$	0.019	0.008	0.005	0.007	0.030
Halftone − Model, $L = L_{\text{opt}}$	0.010	0.005	0.003	0.002	0.020

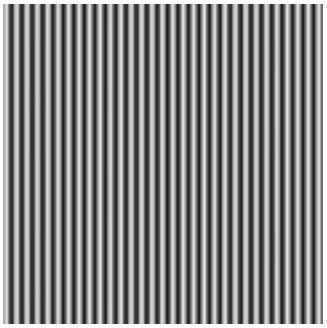
Table 3.3: Correlation coefficients for modified halftone residuals for the Jarvis filter. The first row shows the correlation of the original image and the (halftone − original) residual. The next two rows show the correlation of the original image and the (modified halftone − original) residual, using the average L for this filter, and the optimum L for this filter and each image.

residual when $L = L_{\text{ave}}$. When $L = L_{\text{opt}}$, the average correlation of the original to the residual falls to 0.06 that of the original to the unmodified residual. This lends strong support to the validity of the gain model.

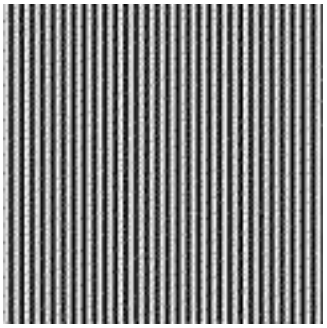
3.3.3 Validation by using sinusoidal inputs

By applying standard and modified error diffusion to a sinusoidal input image and finding the Fourier transform of the residuals, one can see the individual distortion components introduced by halftoning (which have been referred to as "noise"), and measure how strongly each is suppressed in the modified residual. Figure 3.13(a) shows a vertical sine wave grating of size 256×256 pixels with frequency $0.24 f_N$, where f_N refers to the Nyquist frequency. Figures 3.13(b) and 3.13(c) show the standard and modified ($L = L_{\text{opt}}$) halftones, respectively. Each residual (b) − (a) and (c) − (a) is averaged over its rows to produce a 256-element vector, and its Fourier transform is computed.

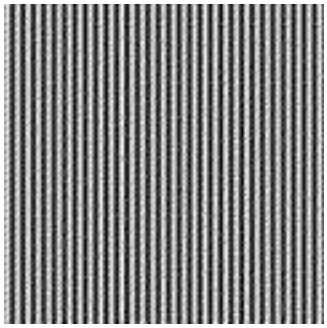
The results are shown in Figure 3.13(d). Both spectra have been scaled by the same factor, so that the level of the fundamental component in the unmodified residual is unity. The residual spectra consist of multiple lines; these



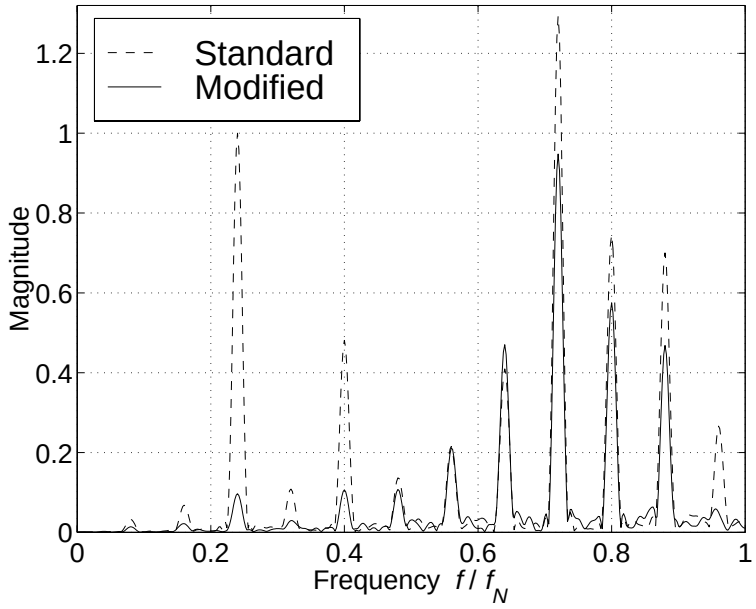
(a) Input.



(b) Halftone.



(c) Modified halftone.



(d) Magnitude spectra of residual images (row averaged).

Figure 3.13: Gain model validation using a sinusoidal input with a horizontal frequency $f_h = 0.24 f_N$, halftoned with the Floyd-Steinberg error filter. Each row vector is Hanning windowed before computing a 501-point Fourier transform, to obtain a sample every $0.004 f_N$.

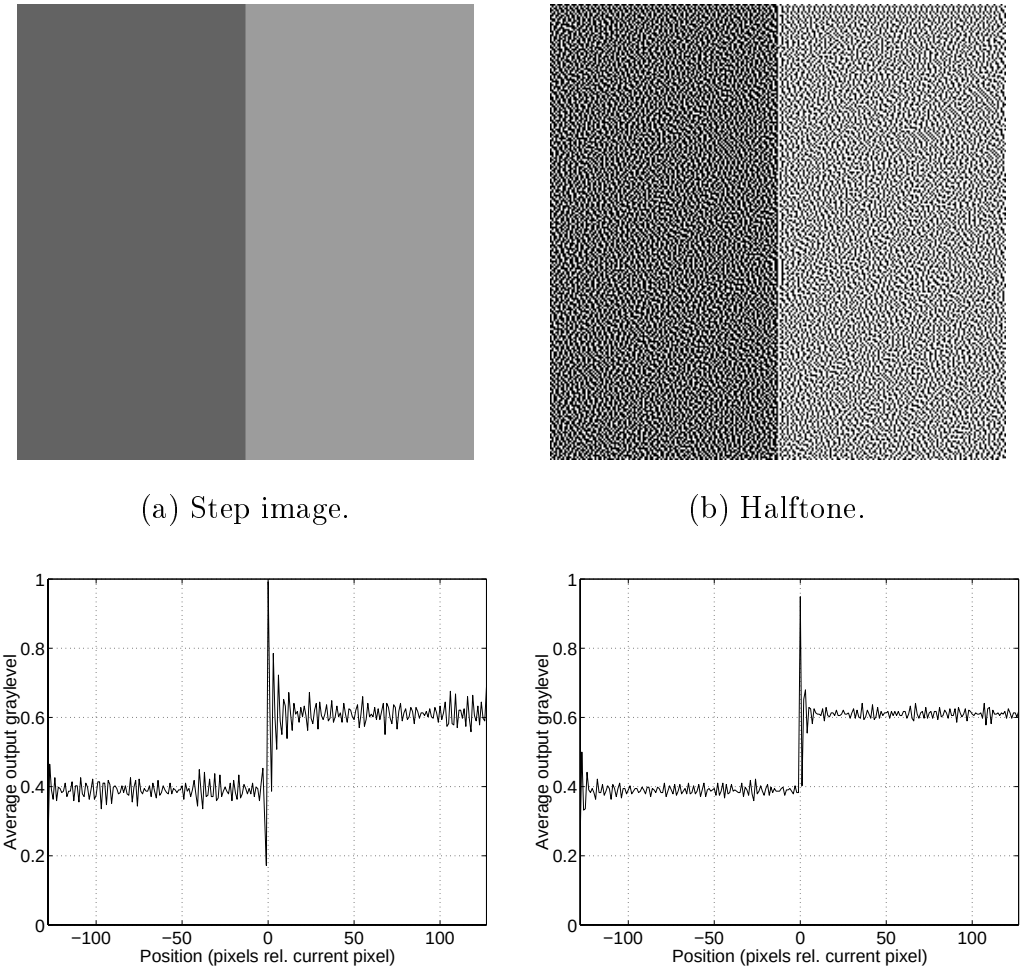
lines are harmonics of the fundamental. Some have been aliased about f_N, and therefore do not have a harmonic relationship to the fundamental. The third harmonic, at $0.72f_N$, and the fifth, aliased to $0.80f_N$ from $1.20f_N$, are particularly strong, because of the symmetric distortion characteristic of the quantizer. The form of the spectrum of a one-bit modulator with a sinusoidal input has been rigorously analyzed by Gray, Chou and Wong [57]. Nearly all of the harmonic products are attenuated in the modified residual; the fundamental is attenuated by a factor of more than 10. This concurs with the large reduction in residual correlation obtained when using modified error diffusion, and lends further weight to the accuracy of the linear gain model.

3.4 Physical reason for sharpening

The correlation of the quantization error with the input image led to the linear gain model for the quantizer, which accurately predicts the edge sharpening of error diffusion. However, this does not explain *why* sharpening occurs; it merely models it. In fact, the means by which sharpening occurs has not been addressed before. In this section, this effect is explained. Section 3.4.1 shows that decorrelating the quantization error eliminates edge sharpening. Section 3.4.2 explains how the finite size of the error filter leads to sharpening. Section 3.4.3 shows how the signal gain K_s can be predicted from the error filter.

3.4.1 Correlation of the quantization error

The linear gain model shows that sharpening results from the correlation between the quantization error and the input. This implies that if the quantization error is decorrelated using dither, sharpening will disappear. To quantify



(a) Step image.

(b) Halftone.

(c) Horizontal step response.

(d) Vertical step response.

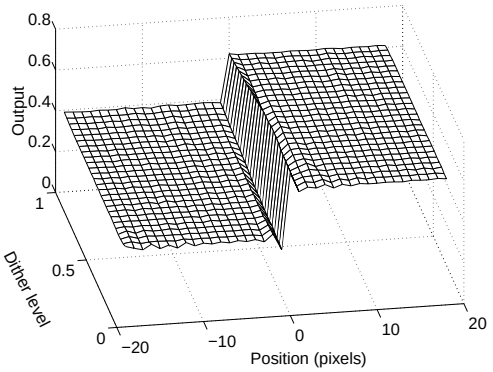
Figure 3.14: Measuring the step response of Jarvis error diffusion. The halftone shown in (b) is used to measure the horizontal step response; a halftoned, rotated input image is used to measure the vertical response. Row-averaged horizontal and vertical outputs are shown.

sharpening, the *effective step response* of the halftoning scheme is measured. A step image is generated, as shown in Figure 3.14(a). The graylevels chosen are not rational numbers, to reduce the likelihood of idle tones in the halftone [51]. The step image is halftoned and its rows averaged to form the one-dimensional effective horizontal step response shown in Figure 3.14(c). Strong ringing is evident, causing the step to be exaggerated, i.e., sharpened. Figure 3.14(d) shows the vertical step response. It exhibits one-sided ringing at the step. Both responses are noisy because of the low number of averages used for these plots (256). In practice, several thousand rows are averaged to obtain a low-noise measurement. The difference between the vertical and horizontal step responses will be explained in Section 3.4.2.

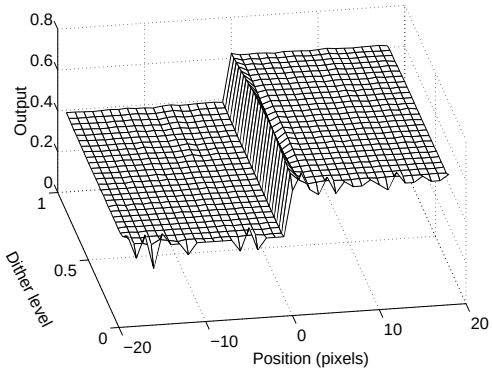
To decorrelate the quantization error, dither with a rectangular probability distribution function is added to the input image. The dither level is varied from zero to one (a full quantization step). As dither increases, the correlation between the quantization error and the input decreases, until zero correlation is achieved at a level of one [19]. Figure 3.15 shows the resulting measured step responses. The x axis denotes the distance in pixels from the step. The y axis denotes the level of dither. In all the plots, the response converges to an ideal, unsharpened step as the dither level increases. This new result confirms the hypothesis suggested by the linear gain model that sharpening is due to correlated quantization error.

3.4.2 Finite size of the error filter

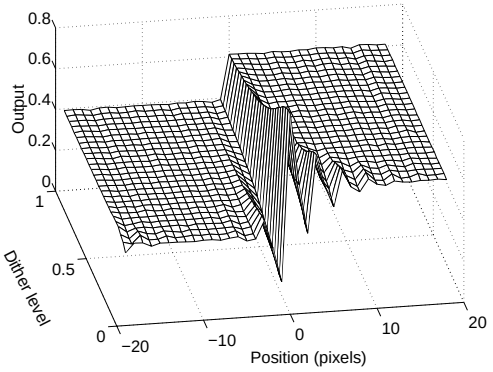
The feedback in error diffusion acts to reduce the average graylevel error of the halftone to zero in smooth regions of the image. Near edges, however, it



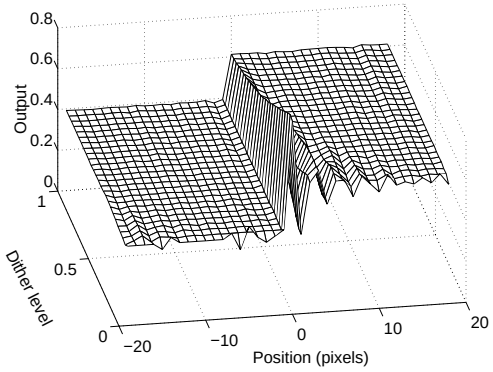
(a) Floyd-Steinberg, horizontal.



(b) Floyd-Steinberg, vertical.



(c) Jarvis *et al.*, horizontal.



(d) Jarvis *et al.*, vertical.

Figure 3.15: Dithered step response results. Measurements were made using the method shown in Figure 3.14, averaging over 16384 rows.

attempts to correct for the graylevel error of pixels on the *far* side of the edge that fall within the support of the filter. This causes errors in the average graylevel on the *near* side. This error sharpens the edge, as will be shown.

Consider a vertical step edge, with a graylevel of 0.4 on the left and 0.6 on the right, and assume that the Jarvis filter is used. The upper part of Figure 3.16(a) shows the first row of the input image, and the position of the error filter. (Most of the filter falls outside the image and is not shown.) The

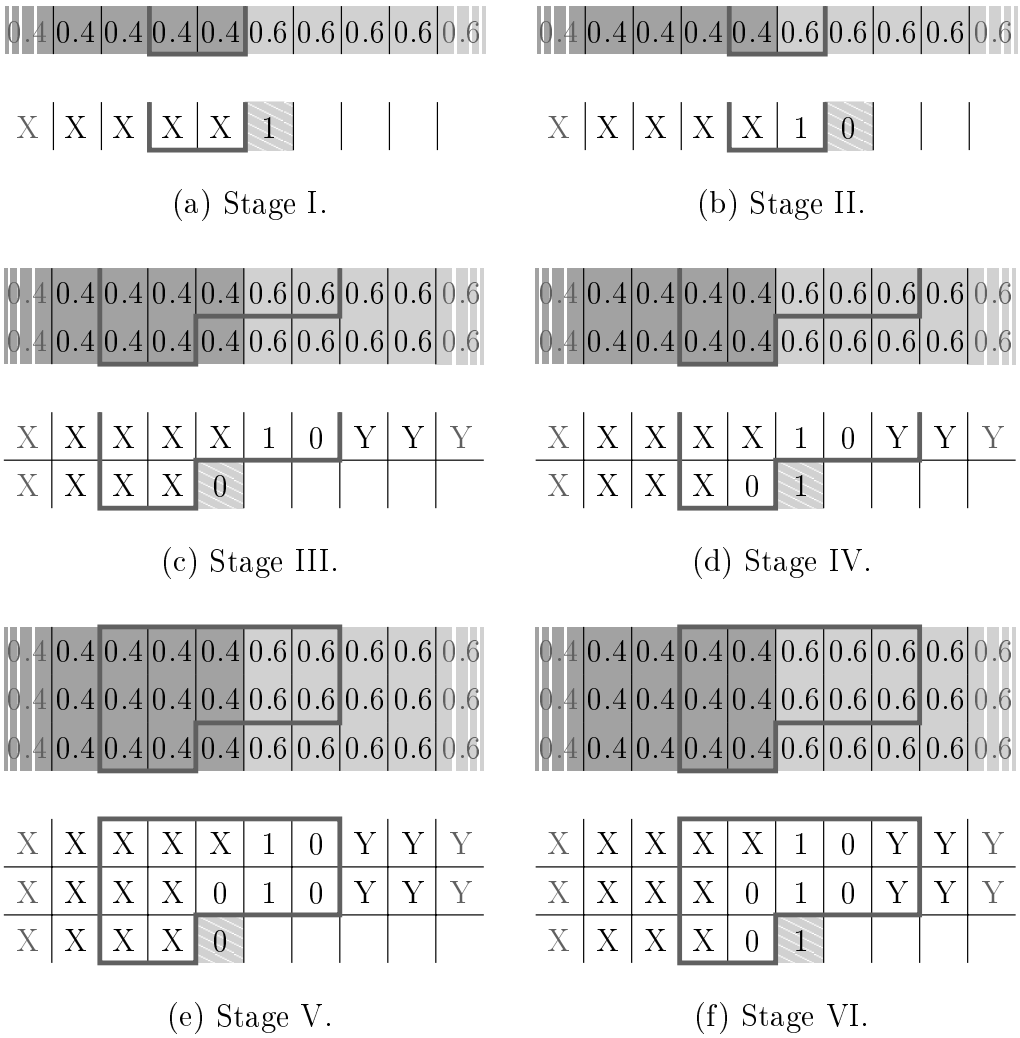
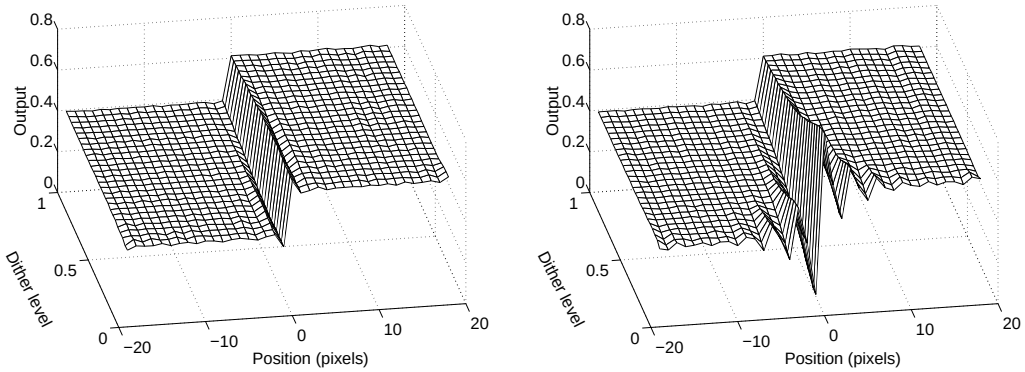


Figure 3.16: Edge enhancement. In Stage I, the pixel to the right of the edge is forced high. In Stage II, its neighbor is forced low to compensate. In Stage III, the pixel to the left of the edge is forced low because the filter extends to the right of the edge. In Stage IV, the pixel to the right of the edge is forced to 1 because the pixels to its left and upper right outweigh the pixel above it. In subsequent stages, the edge sharpening perpetuates and spreads horizontally.

lower part shows the first row of the halftone, with the current pixel striped. Blank pixels have not yet been quantized. The average quantization error is driven to 0 by feedback for the pixels marked 'X', whose average graylevel is 0.4. Since the input is above the threshold, the current output is forced to 1. (When the output is described as being *forced* to a state, this means that quantization is statistically more likely to result in that state than the opposite state.) In Figure 3.16(b), the current pixel is forced to 0 to counteract the positive quantization error of the previous pixel. This tends to overcorrect, forcing the next pixel to 1 to compensate. As halftoning proceeds along the row, the ringing due to overcorrection dies away, and the average quantization error falls to zero. Pixels marked 'Y' have an average graylevel of 0.6.

In Figure 3.16(c), the current pixel is on the second row, to the left of the edge. The error filter covers five pixels to the left of the edge, whose quantization error is close to zero, and two pixels to the right of the edge. The nearer of these two pixels has more weight in the error filter, making the total quantization error positive; the output is therefore forced to 0. Figure 3.16(d) shows that the next pixel is forced to 1 because of the 0 pixel to its left. The clustering of zeros to the left of the edge and ones to the right continues down the image, and spreads horizontally. The finite filter size therefore causes an initial overcorrection in the output near an edge, which is then compensated for in succeeding pixels, leading to an oscillatory step response.

The ringing in the horizontal step response (Figure 3.15) begins *before* the edge; this is because the error filter extends horizontally ahead of the current pixel on the rows above. The vertical step response is one-sided, because



(a) Floyd-Steinberg. (b) Jarvis *et al.*

Figure 3.17: Horizontal step responses using a serpentine scan. Dither increases along the y axis. The responses have horizontal symmetry, as expected, unlike those for the raster scan (Figure 3.15).

the error filter does not extend vertically beyond the current pixel.

The serpentine scan offers insight into edge sharpening. Because the direction of the scan reverses on each row, one expects the horizontal step response to be symmetric. Figures 3.17(a) and 3.17(b) show the horizontal step responses for the Floyd-Steinberg and Jarvis filters, respectively. They are indeed symmetric. Vertical step response is unaffected by the scan.

3.4.3 Predicting K_s from the error filter

The error filter $H(\mathbf{z})$ is a lowpass filter with a maximum gain of unity at DC. For wideband inputs, the standard deviation of the filter output is therefore smaller than the standard deviation of its input. The ratio of the input and output standard deviations is given by

$$R = \left(\frac{\int_{-\pi}^{\pi} \mathcal{F}[g(i,j)] \mathcal{F}^*[g(i,j)]\, dx\, dy}{\int_{-\pi}^{\pi} \mathcal{F}[g(i,j)] \mathcal{F}^*[g(i,j)] \mathcal{F}[h(i,j)] \mathcal{F}^*[h(i,j)]\, dx\, dy} \right)^{\frac{1}{2}} , \qquad (3.41)$$

Computed Parameter	Error filter		
	Floyd-Steinberg	Jarvis *et al.*	Stucki
R (3.41)	1.91	3.89	3.58
K_{ave}	2.03	4.37	3.94

Table 3.4: Comparison of error filter ratio R and K_{ave}. Values of R are computed from (3.41). The K_{ave} figures are taken from Table 3.1.

where $g(i,j)$ is the input to the error filter, $h(i,j)$ are the coefficients of the error filter, $\mathcal{F}[x]$ denotes the Fourier transform of x, and * denotes complex conjugation. If the spectrum of the quantization error is $1-H(\mathbf{z})$, as predicted by the linear gain model, then $g(i,j)$ is given by

$$g(i,j) = \mathcal{F}^{-1}[\mathcal{F}[1 - h(i,j)]] \; . \tag{3.42}$$

The numerator in (3.41) is the signal power at the filter input, while the denominator is the output power. It was found empirically that computing the Fourier transforms in (3.41) on a grid of size 6×5 points was sufficient to give an accurate value for R. The integrals become summations, and computation is therefore very simple. Table 3.4 shows computed values of R for three error filters, together with average values of K_s from Table 3.1. There is a strong linear correlation between R and K_{ave}. One can define

$$K_{\text{est}} = 1.17R - 0.2 \; , \tag{3.43}$$

to obtain an estimate for K_s that is accurate to approximately 1 % for the three schemes shown. This provides a simple way to estimate K_s from the error filter alone, and greatly speeds up filter design procedures, since the effect of filter sharpening can be predicted without halftoning test images.

Max. freq. (cyc/deg)	Error filter	WSNR (dB)				
		barbara	*boats*	*bridge*	*lena*	*mandrill*
30	Floyd	15.1	16.9	15.4	16.1	16.2
	Jarvis	11.8	13.2	11.9	12.4	12.4
	Stucki	14.4	15.7	14.2	15.2	15.3
60	Floyd	30.0	31.6	29.2	30.8	30.8
	Jarvis	26.3	27.3	24.5	26.8	26.9
	Stucki	27.6	28.5	25.7	28.2	28.3
90	Floyd	36.0	37.8	34.3	36.5	36.8
	Jarvis	30.7	31.5	28.0	30.9	31.3
	Stucki	31.7	32.5	29.0	32.2	32.4

Table 3.5: WSNR measurements using three error diffusion schemes at different viewing distances. Modified error diffusion was used to compute unsharpened versions of each input image, thereby creating a residual with a low correlation to the original. 'Floyd' refers to Floyd-Steinberg.

3.5 Weighted noise measurements of halftones

The weighted signal-to-noise ratio (WSNR) measure was introduced in Chapter 2. Image noise is weighted according to the human contrast sensitivity function to estimate its perceptual effect. It was shown that it is necessary to first remove image distortions that are not additive noise. By using modified error diffusion with an appropriate value of L, an unsharpened halftone is created. The low correlation of the residual with the original image allows an accurate WSNR figure to be determined.

Table 3.5 lists WSNR figures for five test images, three error filters, and three values of the maximum angular frequency. For an image of size 512×512 pixels at a viewing distance of 400 mm, the three values of maximum angular frequency listed correspond to printed image sizes of 60 mm, 30 mm, and 20 mm on a side, respectively. In Figure 3.18, the WSNR figures for the same

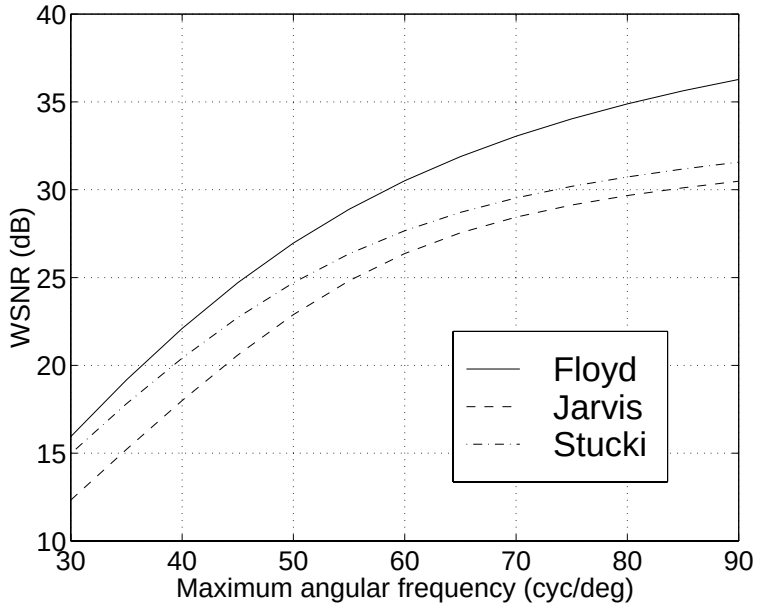


Figure 3.18: Perceptually weighted signal-to-noise figures for three halftoning schemes, averaged over five test images. Solid: Floyd-Steinberg. Dashed: Jarvis *et al.* Dot-dashed: Stucki.

five images are averaged, for values of the maximum angular frequency from 30 to 90 cycles per degree. WSNR is plotted against maximum angular frequency for the three error filters. It is clear from both Table 3.5 and Figure 3.18 that the Floyd-Steinberg error filter achieves consistently higher values of WSNR than both the Jarvis and the Stucki filters, although the Stucki scheme is comparable at small viewing distances. This concurs with psychovisual experience: Floyd-Steinberg halftones appear less noisy than halftones produced by the larger error filters.

3.5.1 Quantifying the effect of idle tones

It was mentioned in Section 3.1.1 that the motivation for the Jarvis and Stucki error filters was to reduce idle tones, which result from a combination of feed-

Dither Level	Error filter		
	Floyd-Steinberg	Jarvis *et al.*	Stucki
0.0	0.16	0.095	0.083
0.2	0.13	0.086	0.069
0.4	0.10	0.081	0.065
0.6	0.088	0.082	0.065
0.8	0.083	0.081	0.070
1.0	0.081	0.076	0.071

Table 3.6: Distortion D for three error filters, with dither applied, measured by averaging sinewave gratings over a range of frequencies: $0.05\,\omega_N \leq \omega_f \leq 0.45\,\omega_N$. A dither level of one corresponds to a full quantization step.

back and non-linearity in the quantizer, and thus are not taken into account by the linear gain model. Because idle tones affect the visual quality of a halftone, a method of measuring their level is presented here.

In Section 3.3.3, idle tones and distortion products were examined by halftoning a sinusoidal grating, averaging over its rows, and computing the Fourier transform of the resulting vector. By analogy with *total harmonic distortion* (THD) [58], the *total distortion* T for the halftone is defined as

$$T = \left[\frac{1}{Y(e^{j\omega_f})Y^*(e^{j\omega_f})} \sum_{\omega \in \{\omega_d\}} Y(e^{j\omega})Y^*(e^{j\omega}) \right]^{\frac{1}{2}}, \tag{3.44}$$

where $Y(e^{j\omega})$ is the discrete Fourier transform of the row-averaged grating image y, ω_f is the radian frequency of the grating, and $\{\omega_d\}$ denotes the set of frequencies of the distortion products. T is equivalent to THD if the set $\{\omega_d\}$ contains only distortion products that are harmonically related to the fundamental. Because some distortion products are aliased back into the passband, they may not be *harmonically* related to the fundamental.

To obtain an expected distortion measure D for a halftoning scheme, T

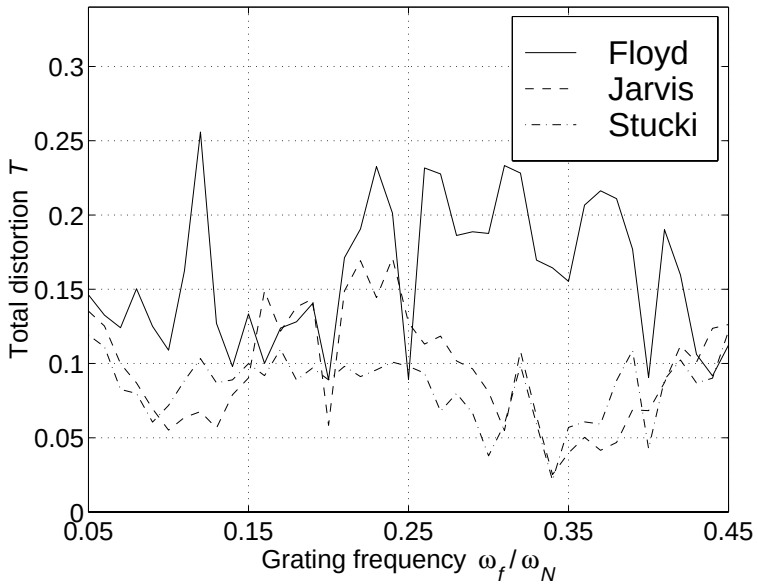


Figure 3.19: Computed harmonic distortion T for three error diffusion schemes, for a range of sinusoidal input frequencies. Solid: Floyd-Steinberg. Dashed: Jarvis *et al.* Dot-dashed: Stucki.

is computed for N values of the grating frequency and the results are averaged: $D = \frac{1}{N}\sum_{n=1}^{N} T(\omega_n)$. Figure 3.19 shows the variation of T with ω_f for three error diffusion schemes. The Floyd-Steinberg distortion is consistently higher than the distortion for the larger filters. Table 3.6 shows computed values of D for the same schemes, with various levels of dither applied. The undithered result confirms that the Floyd-Steinberg error filter is more tonal than the larger filters, on average. This agrees with results from the delta-sigma modulation literature, namely, that lower-order modulators exhibit higher tonality than higher-order systems [18]. As more dither is applied, D decreases until it reaches the noise floor defined by the level of dither. This also agrees with one-dimensional results [19].

Use of the serpentine scan reduces tonality to 0.12 for the Floyd-

Steinberg scheme, compared to 0.16 for the raster scan. This was predicted by Fan [47], and corresponds with the higher subjective quality of serpentine scanned halftones over raster scanned halftones. The large reduction in tonality indicates that the serpentine scan should be used for small error filters such as the Floyd-Steinberg filter. The serpentine scan has no measurable effect on the tonality of the Jarvis and Stucki schemes.

3.6 Summary

The linear gain model of the quantizer has been shown to accurately predict both the sharpening and the noise shaping characteristics of error diffusion. The accuracy of the model was demonstrated in three independent ways by measuring the correlation of residual images. The combination of modified error diffusion and the linear gain model allows unsharpened halftones to be created with *any* error filter. This allows schemes to be compared by the use of perceptually weighted noise metrics, such as WSNR.

The physical mechanism by which edges are sharpened was explained, and a method of predicting the effective quantizer signal gain K_s from the error filter was presented. A distortion measure that quantifies the tonality of halftoning schemes was also introduced. By characterizing *edge sharpening*, *noise shaping*, and *tonality* separately, one is able to obtain objective measures of the subjective quality of halftones. This allows meaningful comparisons of the results of error diffusion schemes to be made. Sharpening has received little attention in the literature. The new results presented here explain its origin and permit the design of novel halftoning schemes, some of which will

be examined in Chapter 5.

Of the three classic filters, the Floyd-Steinberg filter has the lowest computational requirement, since it has four taps rather than twelve. The results indicate that it also gives the best WSNR performance at any viewing distance. However, the larger filters have lower tonality, and consequently fewer artifacts. The serpentine scan should be used with the Floyd-Steinberg error filter to reduce tonality. The Floyd-Steinberg filter gives a neutral rendering, because it only mildly sharpens the image. When added sharpness is desirable, the Stucki filter gives results that are only slightly worse in WSNR terms than the Floyd-Steinberg filter, with much lower tonality.

Chapter 4

Inverse Halftoning

Inverse halftoning algorithms recover grayscale images from halftones. A scanned document that contains halftones cannot be scaled, sharpened or rotated without causing severe degradation to the halftones. In addition, halftones do not compress efficiently. Halftones are therefore converted to grayscale by using inverse halftoning. A side benefit is that inverse halftoning produces images that are visually superior to their halftoned versions.

In this chapter, a new inverse halftoning scheme based on anisotropic diffusion is presented. It produces high quality images from error diffused halftones at a low computational cost, and with a very small memory requirement. The algorithm varies the trade-off between spatial resolution and grayscale resolution at each pixel to obtain a sharp image with a low perceived noise level. A model for inverse halftoning is also presented that enables objective measures of the noise content and blurring of inverse halftones to be made. The perceptually weighted signal-to-noise ratio (WSNR) of Chapter 2 is used to permit quantitative comparison of the results of inverse halftoning schemes.

4.1 Introduction

In general, halftones and other binary images cannot be manipulated without causing severe degradation. Exceptions include cropping, rotation through multiples of 90°, and logical operations. Another exception, due to Wong [23], is a halftoning scheme in which smaller halftones are embedded within larger ones, thereby allowing the image to be downsampled by a pre-determined rational factor. However, no other image manipulations can be performed. Halftones are difficult to compress, either losslessly or lossily; grayscale images, on the other hand, can be compressed efficiently [59, 60]. Inverse halftoning, which converts a halftone to a grayscale equivalent, permits the application of a wide range of image processing operations to halftones.

Inverse halftoning attempts to recreate a grayscale image with a typical wordlength of eight bits from a halftone with a wordlength of one bit. The problem is therefore underdetermined; an essentially infinite number of possible grayscale images could have led to the given halftone, even if the halftoning scheme were known. Several methods for inverse halftoning have been described in the literature [25, 26, 61, 62]. They can be divided into two broad groups: schemes designed for error diffused images, and schemes designed for screened images. At least one published scheme makes use of a parameter that allows it to be used with both screened and error diffused images [5]. However, most schemes are optimized for only one type of halftone, because error diffusion and screening produce outputs with greatly differing artifacts, as shown in Chapter 1. In this chapter, only error diffused halftones are considered.

Section 4.2 surveys existing work in the field. Section 4.3 discusses the

trade-offs inherent to inverse halftoning. Section 4.4 presents the proposed inverse halftoning algorithm, which uses estimated local image gradients to vary the cutoff frequency of a variable smoothing filter. The design of the filter is described in Section 4.5, and the design of the gradient estimators is described in Section 4.6. Section 4.7 explains how the inverse halftone is constructed, and discusses the computational requirements of the algorithm. Section 4.8 presents results and compares them with results from existing schemes. A model for inverse halftoning is also presented that enables objective measures of the noise content and blurring of inverse halftones to be made. The weighted signal-to-noise ratio (WSNR) metric presented in Chapter 2 is used. It was shown in Chapter 2 that peak signal-to-noise ratio (PSNR) is inadequate as a measure of visual quality of inverse halftones. However, it is often the only metric given, and it is therefore quoted where available. Finally, Section 4.9 summarizes the contributions of the chapter. A condensed version of this chapter can be found in [63].

4.2 Previous work

The simplest inverse halftoning method consists of filtering the halftone with a fixed lowpass filter. This removes quantization noise, but also removes important high frequency image information in the halftone, such as edges and texture. The spectrum of the original image, which is typically lowpass, overlaps the highpass spectrum of the quantization noise. If the cutoff frequency of the lowpass filter is too low, then the inverse halftone is blurry; if it is too high, then the inverse halftone is noisy. In Chapter 5, inverse halftoning by linear lowpass filtering is shown to be sufficient if the inverse halftone is sub-

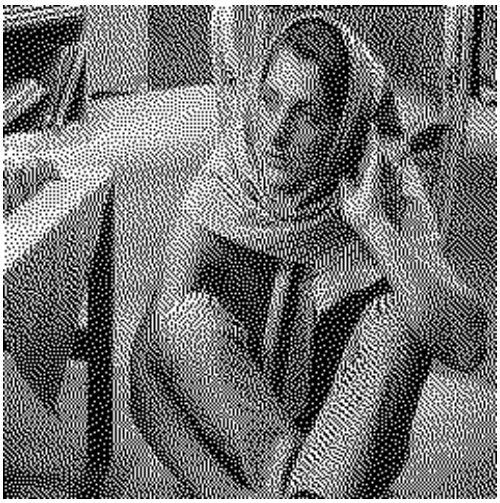
sequently re-halftoned. For producing grayscale images of high visual quality, however, it is inadequate.

Figures 4.1(a) and 4.1(b) show the *barbara* image and its halftone, respectively. Figure 4.1(c) shows the result of inverse halftoning with a fixed, separable, lowpass filter with cutoff frequency $f_c = 0.10 f_N$ in each direction, where f_N denotes the Nyquist frequency. The image is smooth, but blurry. Figure 4.1(d) shows the result of changing the cutoff frequency to $f_c = 0.40 f_N$. The image is sharp, but noisy. It is possible to design an optimal linear space-invariant (Wiener) filter for this application, but the results are poor because of the spectral overlap of signal and noise.

Screened images, especially those dithered with clustered dot screens, are generally of lower quality than error diffused images. Several methods for inverse halftoning screened images have been reported in the literature [62, 64, 65]. Fan [64] estimates the dither matrix (screen), finds a grayscale image that leads to the given halftone when dithered with the estimated screen, and then constrains this image using "logic filtering" (a form of non-linear filter) to provide smoothing without blurring edges. Analoui and Allebach [62] use the theory of projection onto convex sets (POCS) to restrict the inverse halftone to the set of all bandlimited grayscale images that lead to the given halftone under the assumed halftoning scheme. No PSNR figures are given.

Inverse halftoning methods designed specifically for error diffused images have also been published. Ting and Riskin [60] employ vector quantization (VQ). Using a suite of test images, a 512-element codebook is constructed which maps 3×3 neighborhoods of pixels in the halftone to the pixel in the

(a) Original image.



(b) Floyd-Steinberg halftone.

(c) Lowpass filtered, $f_c = 0.10 f_N$.

(d) Lowpass filtered, $f_c = 0.40 f_N$.

Figure 4.1: Inverse halftones generated using linear space-invariant lowpass filtering. The original *barbara* image is halftoned using Floyd-Steinberg error diffusion, and inverse halftoned with a fixed lowpass filter whose cutoff frequency f_c is shown as a fraction of the Nyquist frequency f_N. The filters follow the design described in Section 4.5.

inverse halftone that lies at the center of the neighborhood. Once the codebook has been constructed, inverse halftoning proceeds by table lookup, and is therefore extremely fast. The gray level corresponding to a particular binary input is the most likely value of the pixel in the original image at that point, given the values of the halftone in the surrounding neighborhood. A PSNR of 30.41 dB for the *lena* image is quoted.

Stevenson [66] and Schweizer and Stevenson [27] present an inverse halftoning scheme for error diffused images that uses maximum *a posteriori* (MAP) estimation. This is a Bayesian method that assumes a Markov random field model for the image. Geman and Geman performed *simulated annealing* for image restoration using this model [67]. With an appropriate choice of parameters, the scheme produces good inverse halftones. However, it is slow, because an energy function must be computed over the entire image at each iteration, and many iterations may be required. Furthermore, the resulting image is not guaranteed to be of high quality. No PSNR figures are given.

Hein and Zakhor [61, 68] present a reconstruction method based on POCS. The inverse halftoning process is subject to a spatial domain constraint (the inverse halftone must lead to the given halftone when halftoned with the known error diffusion kernel) and a frequency domain constraint (the inverse halftone is bandlimited). The two intersecting sets of images satisfying these constraints are convex. Following the theory of POCS, an image can be found iteratively that is a member of both of these sets. The search for one of these images is shown to be a quadratic programming problem. It is computationally intensive, and a heuristic for terminating the process must be devised. The

spatial constraint is academically appealing, because one can be sure that the inverse halftone *could* have been the original image, but it is unnecessarily restrictive. A PSNR of 30.40 dB is given for the *lena* image.

Wong [25] describes two iterative inverse halftoning methods. The first method applies halfband lowpass filtering and adaptive statistical (non-linear) smoothing alternately to reconstruct the grayscale image. The lowpass filtering removes some of the quantization noise, while statistical smoothing is used to smooth the image without excessively blurring its edges.

Wong's second method makes use of kernel (error filter) estimation. The error filter that produced the original halftone is estimated by an iterative process consisting of the following steps:

- Inverse halftone the halftone using lowpass filtering;
- Estimate the error filter from the halftone and inverse halftone; and
- Inverse halftone the image using the newly estimated error filter.

The last two steps are repeated until an acceptable inverse halftone is obtained. No proof of convergence is given, although the algorithm converges on all the test images. The results are of high quality; as a side benefit, one obtains an estimate of the error diffusion kernel. However, the procedure has a high computational cost, and a heuristic is needed to terminate the iteration. A PSNR of 32.0 dB is quoted for the *lena* image after eight iterations.

Xiong, Orchard, and Ramchandran describe a scheme employing wavelets [26]. In this work, inverse halftoning is treated as a de-noising problem. An overcomplete, discrete wavelet transform decomposes the image into a lowpass

subband and two highpass subbands. Edges are extracted from the highpass subbands using a Gaussian lowpass filter. The lowpass subband is transformed again, and the resulting lowpass subband is once more transformed. Noise is removed without blurring edges by correlating the wavelet coefficients across the lowest two scales; edges tend to correlate across scales, whereas noise does not [69]. A map of edge pixels is obtained by thresholding, and is used to suppress noise in smooth regions. Finally, the inverse wavelet transform is used to reconstruct the inverse halftone. The inverse halftones have a natural appearance, with a good range of smooth and sharp regions. A PSNR of 31.47 dB is quoted for the *lena* image.

The disadvantage of the wavelet-based method is that a great deal of computation and memory are needed to perform the overcomplete wavelet transform, which uses large filters and floating-point arithmetic. Nine floating-point images equal in size to the halftone, not counting the halftone and inverse halftone themselves, must be in memory at one time. This makes the wavelet method unattractive in standalone, low-cost applications. Since it has produced arguably the best inverse halftones to date, however, its results are used as a comparison with the scheme presented here. It will be shown that the proposed scheme provides comparable image quality, with execution time and memory requirements that are orders of magnitude lower.

4.3 Trade-offs in inverse halftoning

As discussed in Chapter 3, error diffusion is equivalent to a two-dimensional form of delta-sigma modulator [55]. The process of halftoning can be viewed as

spatially-interactive wordlength reduction, usually from eight bits per pixel to one bit per pixel. Inverse halftoning can therefore be interpreted as spatially-interactive wordlength expansion. This section describes wordlength expansion and the trade-off between grayscale resolution and spatial resolution in inverse halftoning.

In 1-D (audio) applications, such as analog-to-digital (A/D) converters, a delta-sigma converter operating at a high sampling rate produces a one-bit data stream whose spectrum consists of the low frequency signal of interest and shaped quantization noise [18]. The data stream is decimated for further processing and storage. To avoid aliasing, it is first lowpass filtered to remove images above half of the target sampling frequency; this filtering increases the wordlength of each output sample. Thus the wordlength is increased at the same time that decimation is performed. Linear filtering can be used because the high oversampling factor (typically at least 64 times) ensures that the bulk of the noise power falls outside the passband. Recently, more complex methods of decoding oversampled delta-sigma modulated data streams have appeared that give better results than simple linear filtering [70, 71]. It is possible that these techniques could be applied to the problem at hand.

In inverse halftoning, which is a two-dimensional extension of wordlength expansion, one generally assumes an oversampling factor of 1, that is, the number of pixels in the halftone and the inverse halftone are equal. Thus, no decimation is performed. When using linear filtering, the wordlength can only be increased by averaging over many samples, and therefore the inverse halftone contains correlated data. This also follows from the fact that, for

an array of size $M \times N$ pixels, there are 2^{MN} possible binary images, but 256^{MN} possible 8-bit images. Since, for a given deterministic inverse halftoning scheme, there is at most one unique grayscale image for each halftone, a maximum of 2^{MN} grayscale images from the much larger set of 256^{MN} possible images can be produced. Each of these images is therefore highly redundant.

Wordlength is increased by averaging over a neighborhood of samples. For instance, averaging 16 (2^4) samples produces an output wordlength of four bits; in general, N samples must be averaged to obtain a wordlength of $\log_2(N)$ bits. This averaging blurs out features that are within the support of the filter. Therefore, a trade-off exists between grayscale resolution (wordlength) and spatial resolution (detail). A simple lowpass (averaging) filter imposes a fixed relationship between the increase in grayscale resolution and the decrease in spatial resolution. By *varying* the trade-off over the halftone between increasing grayscale resolution and decreasing spatial resolution, a large improvement in inverse halftoning performance is obtained. In smooth regions of the grayscale image, more pixels are included in the average, increasing the wordlength that can be achieved. Near edges, fewer pixels are included in the average, thus preserving the edge. Smooth regions (with many levels of gray) *and* sharp edges (with fewer levels of gray) can therefore be obtained.

4.4 Proposed algorithm

The inverse halftoning algorithm described here is a form of *anisotropic diffusion*, which is a tool introduced by Perona and Malik principally to implement robust multi-scale edge detection [72]. Anisotropic diffusion estimates image

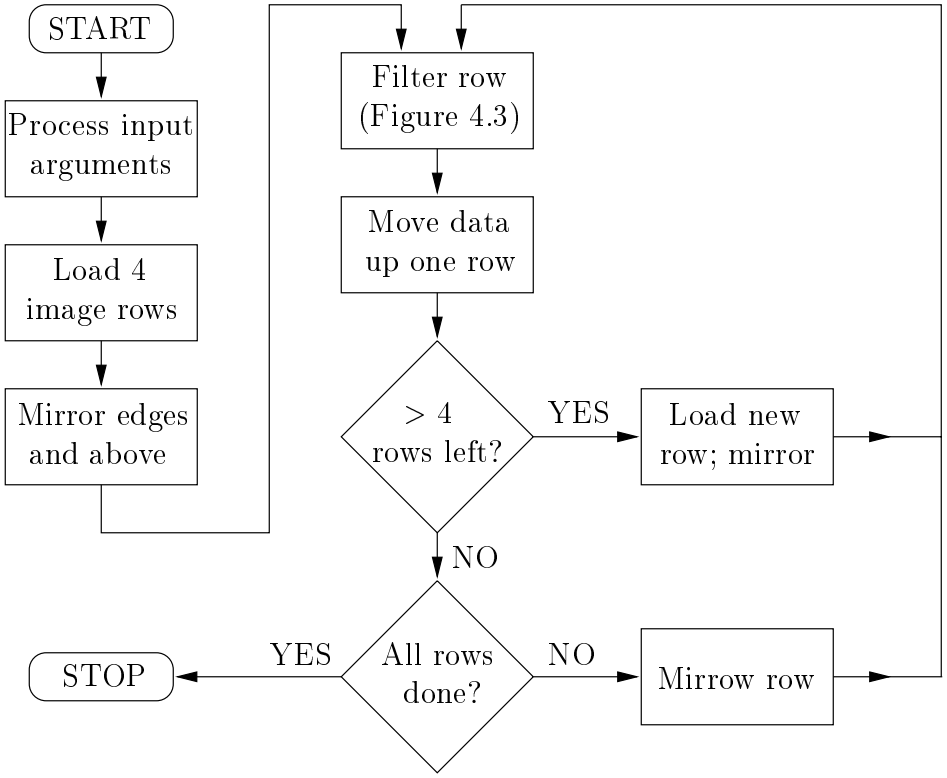


Figure 4.2: Block diagram of the dataflow of the inverse halftoning algorithm. The filter applied at each pixel is determined by operations shown in Figure 4.3. Because all operations are local, the algorithm is well-suited for implementation in VLSI or embedded software.

gradients to compute a diffusion coefficient that governs smoothing. A nonlinear relationship between the estimated gradient and the diffusion coefficient encourages smoothing inside regions, but not between them. To perform inverse halftoning, image gradients are estimated from the halftone, and control functions are derived that vary the cutoff frequency of a smoothing filter. Figure 4.2 shows the dataflow of the algorithm. Only seven rows of the image need to be kept in memory.

Figure 4.3 shows the algorithm in more detail. Gradient estimation

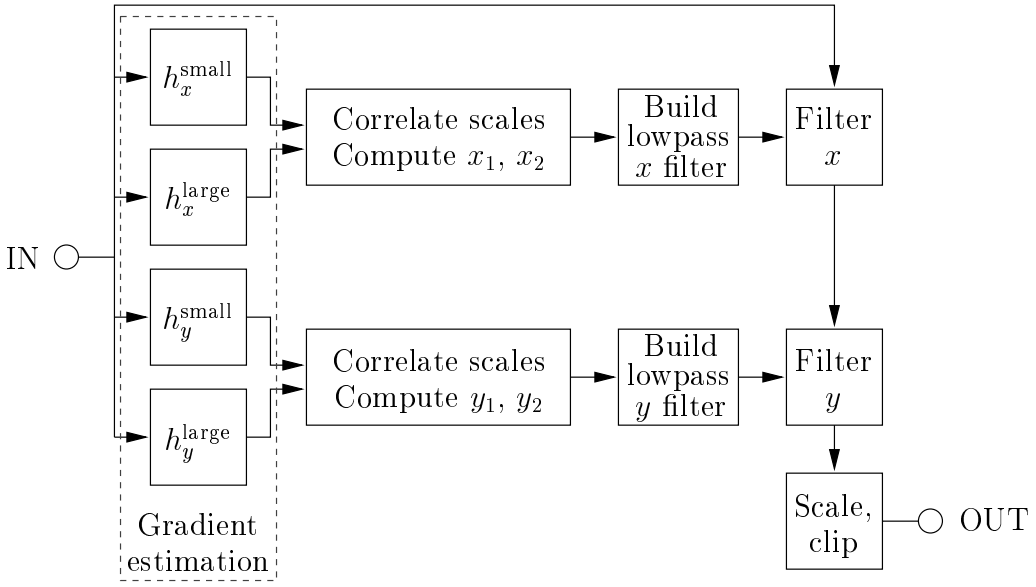


Figure 4.3: Details of the inverse halftoning algorithm. Stages, from left to right, are: 1. Gradient estimation; 2. Gradient correlation and filter parameter construction; 3. Filter construction; and 4. Lowpass filtering.

(stage 1), gradient correlation (stage 2), and filter construction (stages 2 and 3) dominate the computation. In the final stage of Figure 4.3, the halftone is filtered to generate the inverse halftone. The inverse halftoning is performed in the spatial domain using local operations. This obviates the need for computationally expensive and memory-hungry transforms, as execution proceeds in a raster fashion. Raster processing makes better use of a processor's memory cache, since only a small number of image pixels are kept in memory at once. This reduces execution time. Because all operations are local, the algorithm is well-suited for implementation in VLSI or embedded software.

As described in Chapter 1, halftones have a very low signal-to-noise ratio (SNR) because of the one-bit quantization, with most of the noise power

falling in the high frequencies. Multiscale gradient estimation, described in Section 4.6, is used to obtain robust estimates of the image gradients. Gradients are computed at two scales in both the horizontal (x) and vertical (y) directions. The gradient estimates are correlated to give maximum output when a large gradient appears in both scales, such as at a sharp edge. The correlated gradient estimates are referred to as *control functions*.

The control functions are used to construct a separable FIR filter of size 7×7 pixels. The separability of the filter allows it to be constructed and applied independently in the x and y directions, thereby reducing execution time. The smoothing ability of the filter is designed to increase as the image gradient decreases; thus, smoothing is greatest in smooth regions of the original image. Near edges, smoothing is reduced. Because the gradient estimation and filtering occur independently in the x and y directions, smoothing occurs parallel to horizontal and vertical edges, but not across them. Thus edges are preserved in one direction, while grayscale resolution is increased in the orthogonal direction.

4.5 Smoothing filter design

The inverse halftone is constructed from the halftone using a variable smoothing filter. The general criteria for the filter are:

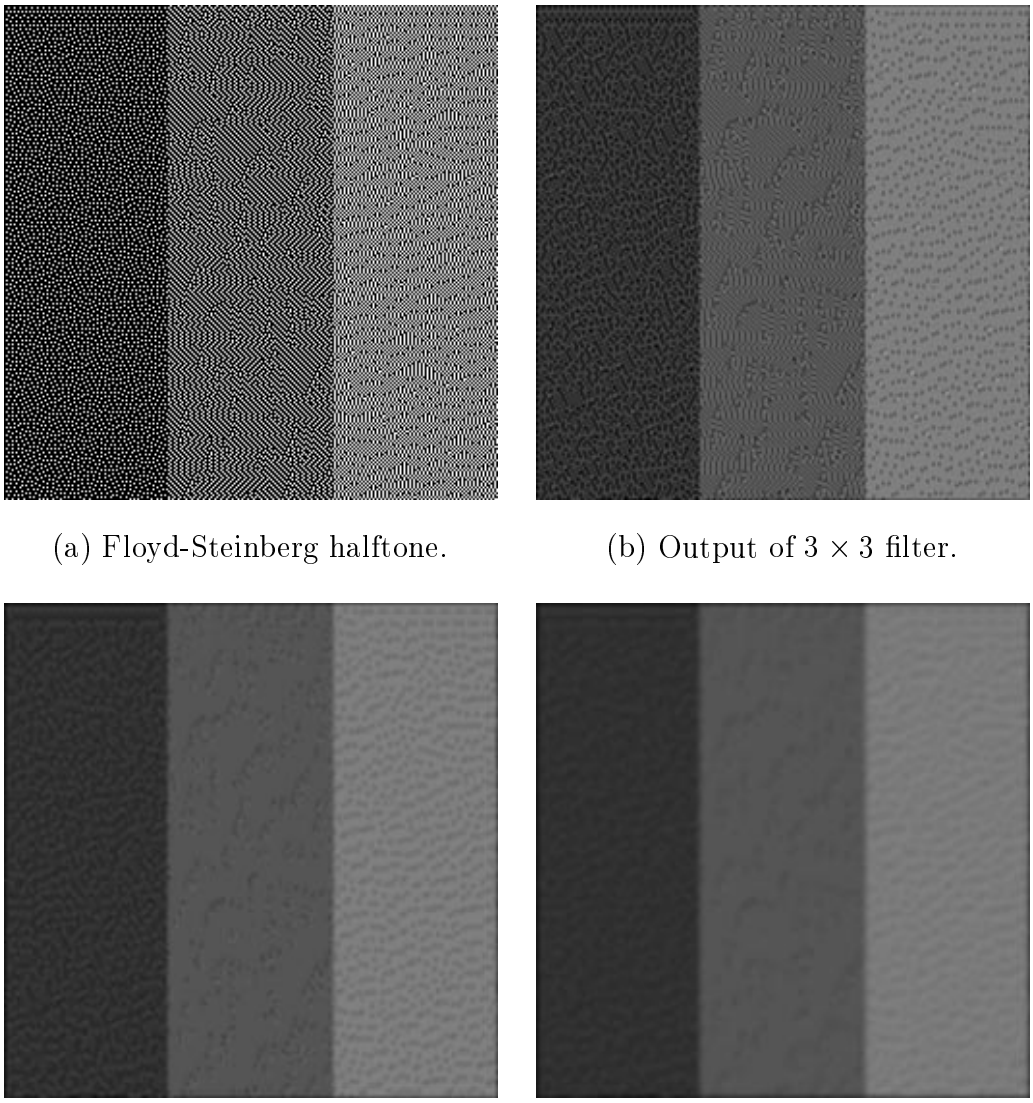
- Small fixed size, FIR
- Simple to generate
- Separable
- Cutoff frequency determined by a single parameter

- Frequency response tailored for halftones

An FIR filter is guaranteed to be stable, and its output can be computed quickly when its extent is small and the filter size is fixed. Computation is reduced by making the filter simple to generate. By making the filter separable, it can be designed, constructed and applied independently in each direction, thereby further reducing execution time. The cutoff frequency of the filter in each direction is determined by one parameter, namely, the control function in that direction. The frequency response of the filter is constrained to account for the particular characteristics of error diffused halftones. Section 4.5.1 describes these characteristics and derives the filter specifications. Section 4.5.2 describes the filter design procedure.

4.5.1 Filter specifications

Because of the reciprocal nature of the Fourier transform, a filter with a large region of support can be designed with a lower cutoff frequency than one with a smaller region of support, and can therefore smooth more. Figure 4.4 shows the effect of filter size on the smoothness of an inverse halftone. The halftone of Figure 4.4(a) is filtered with separable lowpass filters of size 3×3, 5×5 and 7×7 pixels. Each has the narrowest passband for its size, and all meet the same passband and stopband specifications. Image noise reduces steadily as the filter size increases. Testing on a set of eight natural images showed that a 7×7 filter provided enough smoothing to give good results. Extreme smoothness is not required because natural (rather than computer generated) images do not generally contain large, perfectly smooth regions. For computer generated imagery, a larger filter might be desirable, if the penalty in execution



(a) Floyd-Steinberg halftone.

(b) Output of 3×3 filter.

(c) Output of 5×5 filter.

(d) Output of 7×7 filter.

Figure 4.4: Effect of filter size on inverse halftoning performance. The halftone shown in (a) is inverse halftoned using separable FIR filters of different sizes, all with unity gain at DC and line zeros at $(f_N, -)$ and $(-, f_N)$. Passband ripple $< \pm 0.07$. Stopband gain < 0.05.

time can be accommodated.

Worm artifacts (limit cycles) are often present in halftones, as shown in Figure 4.4(a). These strong tones should be suppressed in the inverse halftone, else they will lead to undesirable texture. In Floyd-Steinberg error diffusion, they are particularly likely to occur at (f_N, f_N), $(f_N, 0)$, and, to a lesser extent, $(0, f_N)$ [51], where (f_h, f_v) denotes horizontal and vertical spatial frequency, respectively. These tones are suppressed in the inverse halftone by placing zeros in the smoothing filter at these frequencies. Halftones produced using Jarvis error diffusion are less likely to contain these tones [51].

In general, it is not possible to determine whether high frequency tones in a halftone are caused by quantization noise or by information from the original image. Since natural images tend to be lowpass [24], it is more likely that these tones are artifacts of the halftoning process. It is therefore appropriate to suppress them. Because the smoothing filter is separable, a zero in the one-dimensional prototype becomes a two-dimensional (line) zero in the two-dimensional composite filter. By placing a zero at f_N in the x filter, for instance, one obtains a line zero at $(f_N, -)$ in the composite filter.

The gain of the filter should be unity at DC, to preserve the image mean (brightness). The filter is therefore constrained at DC and f_N. A symmetric filter ensures linear phase; it is well-known that this is critical for good performance of image processing filters [73]. Two parameters are free to determine the filter response. To choose these parameters, the maximum passband ripple and stopband gain are specified.

The maximum passband ripple is constrained to ensure that the inverse

halftone is a faithful reproduction of the original image. A filter with an excessively peaked passband produces falsely sharpened images. It was found empirically that restricting the ripple to ± 0.07 (± 0.59 dB) produced high quality images that were not falsely sharpened. The maximum stopband gain was specified as 0.05 (-26 dB), so that the total noise power in the filter output decreases monotonically as the cutoff frequency of the filter is lowered. If the maximum stopband gain is not specified, it is possible to design a filter a whose cutoff frequency is lower than that of filter b, yet whose output has a higher noise power for the same input. This produces poor inverse halftones, since the reduction of quantization noise is no longer inversely proportional to the local image gradient.

4.5.2 Filter design

The class of one-dimensional, linear phase filters satisfying the criteria of unity gain at DC and a zero at f_N has the form

$$h(n) = \frac{1}{4(x_2+2)} \left[\; x_2 - x_1 + 2, \;\; x_2, \;\; x_1, \;\; 4, \;\; x_1, \;\; x_2, \;\; x_2 - x_1 + 2 \; \right], \quad (4.1)$$

where x_1 and x_2 are parameters that must be chosen so that $h(n)$ satisfies the passband and stopband specifications. (4.1) follows by assuming a filter of the form $[a, b, c, d, c, b, a]$, imposing the constraints at DC and f_N, and simplifying to a form that requires the least computation. This class of filters is referred to as the *one-dimensional prototype* class. Two filters from the class are constructed at each pixel of the input image, one for each of the x and y directions. The following analysis refers exclusively to the x filter. The y filter is constructed in the same way.

A family of lowpass filters that met the specifications was designed using the sequential quadratic programming (SQP) algorithm [74] in the MATLAB optimization toolbox. This algorithm varies parameters (x_1 and x_2) to minimize a cost function, subject to a constraint. The passband ripple was used as the constraint, and the maximum stopband gain as the cost function. These definitions lead to equiripple filters in principle, and near-equiripple filters in practice. Thus the filters are near-optimal in achieving the lowest transition width for the given filter size, passband ripple, and stopband gain.

Ten filters were designed by specifying a desired cutoff frequency f_c, fixing the passband ripple at ± 0.05, and adjusting the stopband edge f_s to the lowest value possible, subject to a maximum stopband gain of 0.03. That is, the filter with the shortest transition width which satisfied the passband and stopband constraints was found. The passband and stopband specifications are slightly better than the targets mentioned in the previous section, to allow for an approximation described later in this section. The actual f_c and f_s of the designed filter were then calculated. Table 4.1 shows the filter parameters.

The cutoff frequency of the filter should be determined by a single parameter, as explained in Section 4.5. A functional relationship between x_1 and x_2 must therefore be found. Figure 4.5 plots x_2 against x_1 from the data in Table 4.1, along with a best fit cubic polynomial; this was found to be the lowest order polynomial that gave an adequate fit. The cubic function is

$$x_2 = 0.4631x_1^3 - 2.426x_1^2 + 4.660x_1 - 3.612 \ . \tag{4.2}$$

The continuous set defined by (4.1) and (4.2) consists of filters whose cutoff frequencies vary from $0.066f_N$ to $0.502f_N$. All the filters have unity gain at

f_c (specified)	f_c (achieved)	f_s achieved	x_1	x_2
0.05	0.066	0.428	3.351	2.192
0.10	0.104	0.627	2.948	0.871
0.15	0.148	0.668	2.705	0.437
0.20	0.205	0.686	2.592	0.247
0.25	0.252	0.699	2.508	0.128
0.30	0.299	0.701	2.462	0.0326
0.35	0.352	0.793	2.145	−0.199
0.40	0.400	0.906	1.707	−0.427
0.45	0.455	0.930	1.452	−0.554
0.50	0.502	0.938	1.309	−0.621

Table 4.1: Parameters of the smoothing filters. The first column gives the specified cutoff frequency. The second column shows the actual cutoff frequency of the designed filter, defined as the lowest frequency for which the gain $G < 0.95$. The third column shows the stopband edge, defined as the highest frequency for which $G > 0.03$. The last two columns show the computed values of x_1 and x_2.

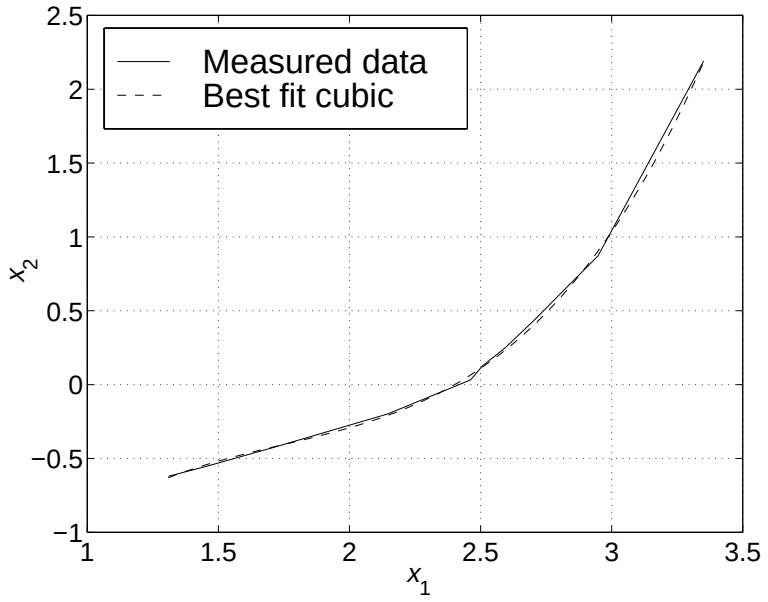


Figure 4.5: Functional relationship between filter parameters x_1 and x_2. Data from Table 4.1 is shown solid. Best fit cubic is shown dashed.

DC and a zero at f_N. The ripple in the passband for any filter is no greater than $\pm 6.2\%$ (± 0.52 dB), and the maximum stopband gain is 0.045 (-27 dB). Thus, the performance of the entire family is within the original specifications, despite the approximation of (4.2).

Figure 4.6 shows the *lena* image filtered with four filters chosen from this family. The same f_c is used for the x and y directions. The suppression of the components at $(f_N, 0)$, and (f_N, f_N) is visible above the hat (where the checkerboard pattern at (f_N, f_N) is prominent) and in the cheek (where vertical stripes at $(f_N, 0)$ are particularly objectionable). Also obvious is the increasing smoothness of the filtered image with decreasing f_c. The shoulder in Figure 4.6(d) is filtered enough to appear smooth, while the feathers and eyes in Figure 4.6(a) are clear and sharp. The filter family therefore provides a range of smoothness needed to produce good inverse halftones.

Figure 4.7 shows the magnitude responses of four two-dimensional filters from the family. Figure 4.7(a) would be used at a vertical edge, as it smooths mainly in the y direction. Figure 4.7(b) would be used in reasonably smooth, isotropic regions of the image. Figure 4.7(c) would be used at a reasonably strong horizontal edge, while Figure 4.7(d) would be used in smooth, isotropic regions. The line zeros at $(f_N, -)$ and $(-, f_N)$ are evident, and the equiripple nature of the filters is visible in Figures 4.7(a) and 4.7(d).

4.6 Derivation of the control functions

As mentioned in Section 4.4, the amount of smoothing applied at a particular pixel is driven by the value of the local image gradient. Because of the presence

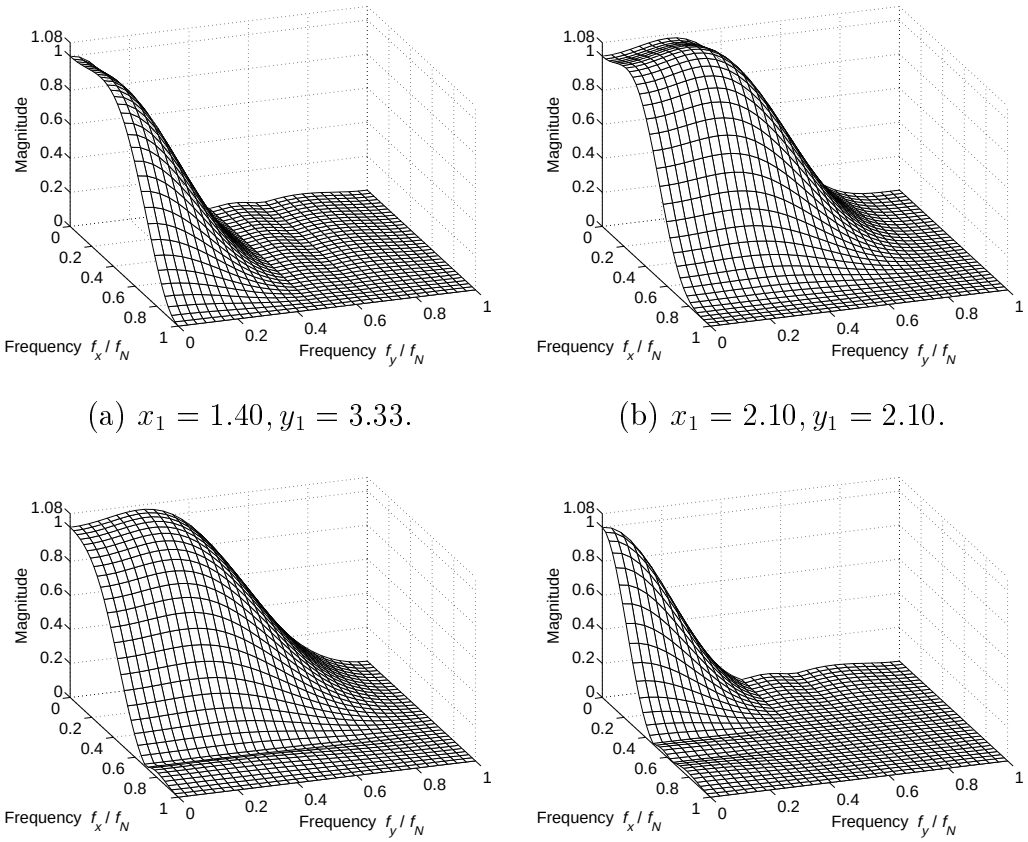
(a) $x_1 = 1.40, f_c = 0.46 f_N$.

(b) $x_1 = 2.07, f_c = 0.37 f_N$.

(c) $x_1 = 2.73, f_c = 0.15 f_N$.

(d) $x_1 = 3.40, f_c = 0.065 f_N$.

Figure 4.6: Effect of the filter cutoff frequency on image smoothness. The *lena* halftone is filtered with four filters from the family, with the parameters shown. The filter parameter x_2 is computed from x_1 using (4.2).



(a) $x_1 = 1.40, y_1 = 3.33.$

(b) $x_1 = 2.10, y_1 = 2.10.$

(c) $x_1 = 2.70, y_1 = 1.80.$

(d) $x_1 = 3.33, y_1 = 3.33.$

Figure 4.7: Magnitude responses of four lowpass filters from the possible range of $(x_1, y_1) \in [1.4, 3.4]$.

of high frequency noise in error diffused halftones, a robust method of gradient estimation is required. This section describes the theory and design of the gradient estimators used in the algorithm, and the method by which their outputs are correlated to derive the smoothing filter control functions.

4.6.1 Gradient estimator design

Consider a continuous image $I(x, y)$. The gradient of the image in the x direction is given by $\partial I/\partial x$. The *gradient operator* $\partial/\partial x$ is a linear filter with frequency response $j\omega$, that is, a response that rises linearly with spatial frequency. If the image I is discretized spatially, the continuous gradient can be approximated by using a digital filter with a frequency response similar to $j\omega$. The frequency response of the discrete difference operator $\Delta_x = I(x+1, y) - I(x, y)$, for instance, is given by $\widetilde{\Delta}_x(e^{j\omega}) = 1 - e^{-j\omega}$, which is approximately $j\omega$ at low frequencies.

Gradient estimation by discrete difference is not robust to noise, because high frequencies are amplified. This is a problem in error diffused halftones, where most of the noise power is at high frequencies. Perona and Malik estimated gradients in grayscale images with discrete differences. They acknowledged that while they obtained good results with these estimators, they were not robust against noise [72]. Catté, Lions, Morel and Coll [75] address the problem of robustness to noise by pre-smoothing the gradient estimate with a discrete approximation to a Gaussian lowpass filter. The reason for using the Gaussian is as follows.

The product of the spatial domain variance (effective filter size) and

frequency domain variance (effective filter bandwidth) for any filter is subject to the uncertainty relation

$$\Delta x \, \Delta \omega \geq \frac{\pi}{4} \ , \tag{4.3}$$

where $\Delta x, \Delta \omega$ are the variances in the spatial and frequency domains, respectively. The filter forms a spatial average over a region of effective width Δx; minimizing Δx therefore improves the localization of the gradient, which is uncertain to within Δx. Similarly, $\Delta \omega$ defines the range of scales over which gradients are estimated; minimizing $\Delta \omega$ restricts this range [76].

For continuous signals, the relation in (4.3) is an equality only for the Gaussian. The Gaussian is therefore the optimal pre-smoothing filter for gradient estimation in continuous signals, in the sense that it provides the best localization of image gradients for a given range of scales. In halftones, however, large amounts of high frequency noise power and strong idle tones introduce additional requirements of the pre-smoothing filter. The conjoint minimization of spatial domain and frequency domain variances is therefore not the only factor determining the filter response. The additional requirements are addressed by designing the pre-smoothing filters according to the characteristics described in Section 4.5. Although no claims are made about the optimality of these filters, they give better performance than Gaussians of the same size. Their impulse responses are very similar to truncated Gaussians, however, and the impulse responses of the resulting gradient estimators are similar to those proposed as optimal by Canny [77].

To improve robustness to noise further, gradients are estimated at two scales and the results are correlated across scales. Large, sharp edges appear

across scales, whereas noise does not [69]. It was found that gradient estimation at two scales gave the best performance for the test images used; the inclusion of a third, smaller scale increased noise in the inverse halftone. The specifications of the gradient estimation filters are as follows:

- Line zeros at $(-, 0)$, $(f_N, -)$, and $(-, f_N)$
- Maximum stopband gain of 0.03
- Peak passband gain of 1
- Narrowest possible passband for a given filter size

The specifications on the line zeros and the maximum stopband gain arise from considerations described in Section 4.5. The peak passband gain is defined to be unity, so the bounds of the filter output are known. The filter passband is made as narrow as possible to best distinguish between the two scales.

Each filter is separable. In the direction in which gradients are estimated, the filter is bandpass, with zeros at DC and the Nyquist frequency. The free parameters are chosen to give the narrowest passband possible, subject to the maximum stopband gain being 0.030. In the direction perpendicular to the direction of gradient estimation, the filter is lowpass, designed according to the criteria of Section 4.5. The parameters are chosen to give the smallest possible passband for the filter size to maximize noise rejection.

Since the peak passband gain of the filters is known, one can find fast integer implementations. Each filter is scaled and its coefficients rounded to fit into one byte. Since the halftone is binary, only integer additions are needed

to compute the output of each filter. The x filters are given by

$$h_x^{\text{small}} = \tfrac{1}{1024}\begin{bmatrix} -19 & -32 & 0 & 32 & 19 \\ -55 & -92 & 0 & 92 & 55 \\ -72 & -120 & 0 & 120 & 72 \\ -55 & -92 & 0 & 92 & 55 \\ -19 & -32 & 0 & 32 & 19 \end{bmatrix},$$

$$h_x^{\text{large}} = \tfrac{1}{2048}\begin{bmatrix} -12 & -27 & -25 & 0 & 25 & 27 & 12 \\ -30 & -68 & -64 & 0 & 64 & 68 & 30 \\ -45 & -103 & -96 & 0 & 96 & 103 & 45 \\ -54 & -124 & -114 & 0 & 114 & 124 & 54 \\ -45 & -103 & -96 & 0 & 96 & 103 & 45 \\ -30 & -68 & -64 & 0 & 64 & 68 & 30 \\ -12 & -27 & -25 & 0 & 25 & 27 & 12 \end{bmatrix},$$

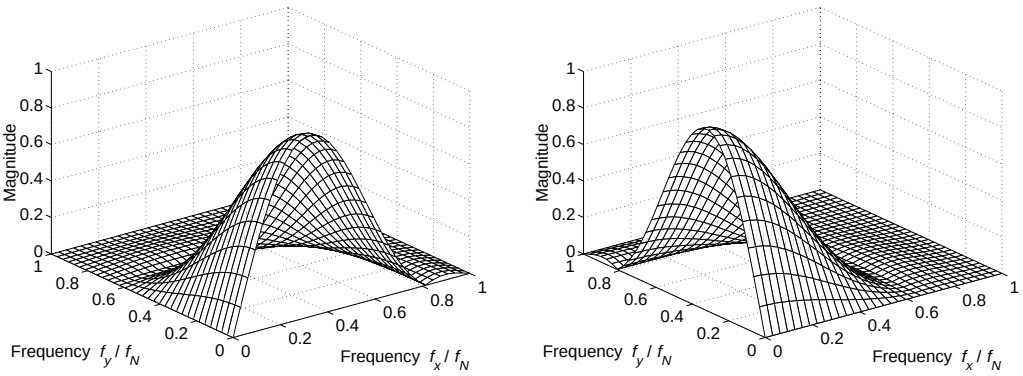
where the superscripts 'small' and 'large' refer to the scale. The y filters are transposes of the x filters. The frequency responses of the four filters are shown in Figure 4.8. The near-linear rise of the response with frequency close to DC conforms to the $j\omega$ response of gradient estimators. The line zeros at the band edges are evident, as is the the equiripple behavior of the large-scale filters shown in Figures 4.8(c) and 4.8(d).

4.6.2 Correlation across scales

At each pixel of the input image, gradients are estimated from the halftone using the filters h_x^{small}, h_y^{small}, h_x^{large}, and h_y^{large} to produce outputs e_x^{small}, e_y^{small}, e_x^{large}, and e_y^{large}, respectively. To correlate the gradients across scales, the control functions are computed according to the products

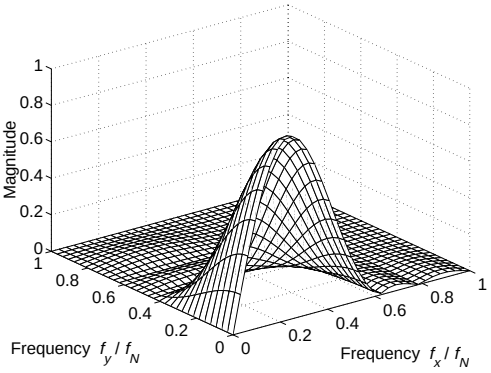
$$e_x^{\text{cf}} = \left| e_x^{\text{small}} \times e_x^{\text{large}} \times e_x^{\text{large}} \right|^{1/3}, \quad e_y^{\text{cf}} = \left| e_y^{\text{small}} \times e_y^{\text{large}} \times e_y^{\text{large}} \right|^{1/3}, \tag{4.4}$$

where $|\cdot|$ denotes absolute value. The large-scale gradients are weighted more heavily than the small-scale gradients to suppress small-scale noise. This pro-

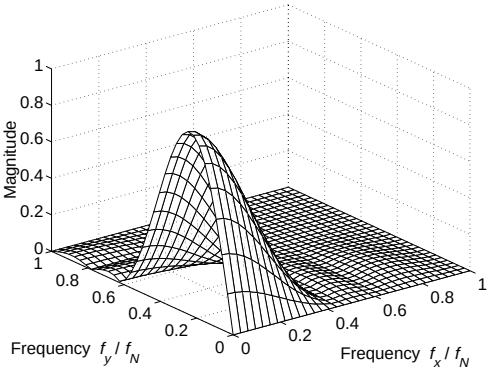


(a) h_x^{small}.

(b) h_y^{small}.



(c) h_x^{large}.



(d) h_y^{large}.

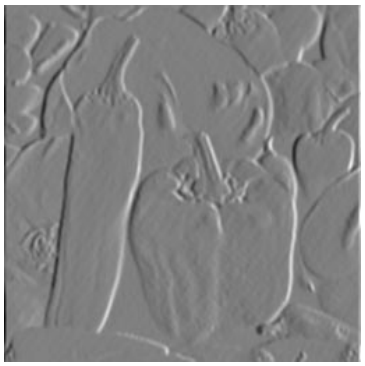
Figure 4.8: Magnitude responses of the gradient estimation filters. The small-scale estimators have peak response at approximately $0.32 f_N$, and a lowpass cutoff frequency of approximately $0.090 f_N$. The large-scale estimators have peak response at approximately $0.24 f_N$, and a lowpass cutoff frequency of approximately $0.066 f_N$.

duces slightly smoother, better quality inverse halftones than if equal weighting is used. Since each gradient estimator is linear, its output is proportional to its input. Each product in (4.4) is therefore proportional to the cube of the true image gradient. The cube root of the product is computed, so that the control function varies linearly with the gradient.

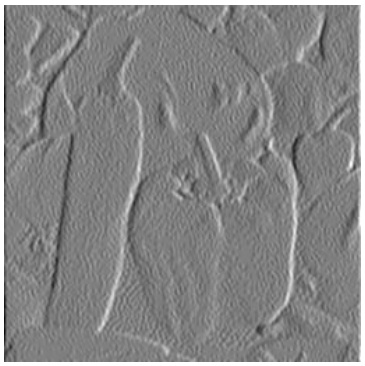
To quantify the accuracy of the gradient estimates, the results of estimating gradients in a halftone are compared with the results of estimating gradients in the original grayscale image. A perfect multiscale detector would produce identical estimates from both images. The output of a practical detector, however, is contaminated by noise in the halftone. This is demonstrated in Figure 4.9, which shows gradients estimated from the original and halftoned versions of the *peppers* image. Modified Floyd-Steinberg halftoning is used to give an unsharpened halftone, as described in Section 3.3.2.

Figure 4.9(a) shows the small-scale x direction gradients computed from the original image, while Figure 4.9(b) shows the same gradients computed from the halftone. Figure 4.9(b) is noticeably noisier than Figure 4.9(a). Figure 4.9(c) shows the large-scale y direction gradients computed from the original image, while Figure 4.9(d) shows the same gradients computed from the halftone. The noise is less obvious in Figure 4.9(d) than in Figure 4.9(b), because the large-scale filter removes more of the quantization noise than the small-scale filter. Figures 4.9(e) and 4.9(f) show the x direction control functions computed from the original image and the halftone, respectively.

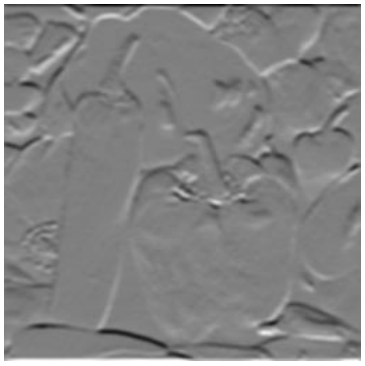
The accuracy of the gradient images obtained from the halftone can be quantified by computing their signal-to-noise ratio (SNR) relative to the



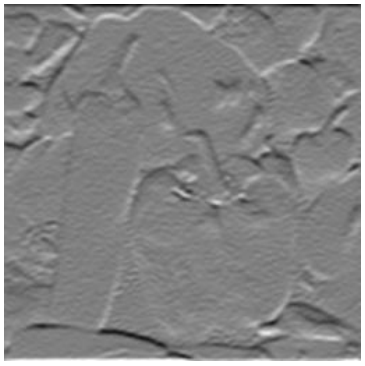
(a) e_x^{small} (original image).



(b) e_x^{small} (halftone).

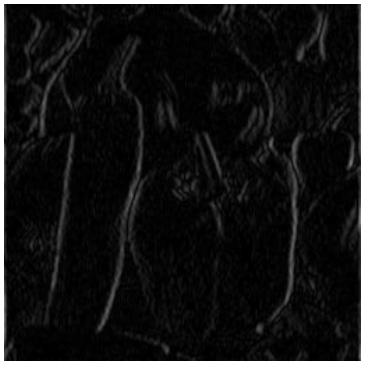


(c) e_y^{large} (original image).



(d) e_y^{large} (halftone).

(e) e_x^{cf} (original image).



(f) e_x^{cf} (halftone).

Figure 4.9: Gradients estimated from original and halftoned *peppers* images.

Image	Description	SNR (dB)
e_x^{small}	Small-scale x	3.25
e_x^{large}	Large-scale x	12.10
e_x^{cf}	Composite x	8.74
e_y^{small}	Small-scale y	2.49
e_y^{large}	Large-scale y	9.91
e_y^{cf}	Composite y	7.53

Table 4.2: SNR of gradients estimated from modified Floyd-Steinberg halftone, relative to gradients estimated from original *peppers* image.

gradients obtained from the grayscale image. The use of SNR is justified because the difference between the images is filtered noise. Table 4.2 gives results for the *peppers* image. The small-scale images, e_x^{small} and e_y^{small}, have an average SNR of approximately 2.9 dB. The large amount of quantization noise in the small-scale images computed from the halftone leads to the low SNR figure; however, the images are sharp. The large-scale images, e_x^{large} and e_y^{large}, have an average SNR of approximately 11 dB. However, they are not as sharp as the small-scale images. The control functions, e_x^{cf} and e_y^{cf}, have an average SNR of approximately 8.1 dB, an improvement of more than 5 dB over the small-scale figure. Furthermore, they are sharp. Thus, by correlating gradients across scales, one obtains most of the noise rejection of the large-scale gradient image, while retaining the sharpness of the small-scale image.

4.7 Inverse halftone construction

The x and y control functions, e_x^{cf} and e_y^{cf}, determine the cutoff frequencies of a separable smoothing filter, whose characteristics are described in Section 4.5. Section 4.7.1 describes how the filter is constructed and applied, and

how computation is minimized for high speed. The computational cost of the algorithm is presented in Section 4.7.2.

4.7.1 Filtering the halftone

Section 4.5.2 showed how the filter parameters x_1 and x_2 determine the cutoff frequency of the one-dimensional prototype filter, and presented a relation between x_2 and x_1. A relation between e_x^{cf} and x_1 is now required. To reduce computation, consider the linear relation (the y relation is analogous)

$$x_1 = a + b\, e_x^{\mathrm{cf}} \ , \tag{4.5}$$

where constants a and b are yet to be determined. When the gradient magnitude is low, the image is smooth, and therefore the cutoff frequency of the filter should be low. This requires x_1 to be at the top of the allowable range: $x_1 \approx 3.4$ (see Table 4.1). When the gradient magnitude is high, x_1 should be at the bottom of the allowable range: $x_1 \approx 1.4$. By varying a and b from their starting values ($a = 3.4,\ b = -10$) and monitoring the visual quality of test images, the optimum values $a = 3.33$ and $b = -5.7$ were obtained.

The parameter x_2 is derived from x_1 using Horner's form of (4.2)

$$x_2 = -3.612 + x_1(4.660 + x_1(-2.426 + 0.4631x_1)) \ , \tag{4.6}$$

which uses 3 multiplications instead of 5. A prototype filter is then constructed according to (4.1), ignoring for the moment the factor of $1/(4(x_2 + 2))$. Each coefficient is a floating-point number in the approximate range $(-0.5, 4)$. Each coefficient is scaled by the factor 1024 (2^{10}), and converted to an integer by discarding the fractional part. This results in at most a 13-bit signed integer,

apart from the fixed central coefficient, which is 14-bit. The reason for this conversion is to permit application of the filter using integer arithmetic, which is quicker than floating-point arithmetic on most hardware.

The x and y prototype filters are applied separably to the 7×7 neighborhood centered on the current pixel. At the boundaries of the image, three pixels are replicated by mirroring to simplify the filtering. Applying the filters separably obviates the need to construct the equivalent two-dimensional filter. A two-dimensional filter would require 49 integer multiplications for its construction, and 48 integer additions for its application, per pixel. Applying the filters separably requires 42 integer additions in the x direction, followed by 7 integer multiplications and 6 integer additions in the y direction, per pixel. Thus 42 integer multiplications per pixel are saved.

Each of the 7 outputs of the x filter is at most a 16-bit signed integer; each is multiplied by one coefficient from the y filter, yielding at most a 29-bit signed integer product, apart from the central product, which may be 30-bit. The 7 products are then summed, yielding at most a 32-bit signed result, which is a common integer wordlength for general purpose hardware. (Fixed-point digital signal processors typically use 16-bit or 24-bit words.) The coefficient quantization has no measurable effect on the final results.

The filtered output pixel is converted to a `float` and scaled. The scaling simultaneously accounts for the ignored factor $1/(4(x_2 + 2))$ from (4.1) (and the corresponding factor from the y filter), the scaling factor used in converting the filter coefficients to integers, and the requirement that the output pixels be in the range $(0, 255)$. Clipping enforces this range, before the pixel is rounded

to the nearest integer and converted to an `unsigned char` (single byte).

4.7.2 Computation and memory requirements

The following arithmetic operations are required per pixel:

- 303 increments (`++`)
- 30–226 integer additions
- 7 integer multiplications
- 34 floating-point additions
- 21 floating-point multiplications
- 5 floating-point divisions

The number of integer additions depends on the image. A halftone composed solely of black pixels would require 30 integer additions per pixel, whereas an all-white halftone would require 226. A typical image is mid-gray on the average, and therefore requires approximately 128 integer additions. The increment operator is listed separately, because some hardware architectures can perform this operation as a special addressing mode, with zero time penalty. The number of floating-point operations, particularly divisions, has been kept to a minimum to increase speed. For an image of size 512×512 pixels, the entire inverse halftoning process takes 2.9 seconds to execute on a 167 MHz Sun UltraSparc 2 machine, and 6.8 seconds on a Sparc 10.

In (4.4), it was shown that two cube roots must be computed to derive the x and y control functions. The cube root is computed using an initial bilinear approximation, followed by two iterations of Newton-Raphson approximation. Over the entire input range, the result is accurate to better than

0.4%; for more than 90% of the input range, the accuracy is better than 0.01%. A total of 4 additions, 7 multiplications, and 2 divisions (all floating-point) are required to compute each cube root.

Execution proceeds in raster fashion, one row at a time. Seven image rows are required for the filters; they are kept in the *image storage area*, a pre-allocated array of memory of size $7\,(c+6)$ bytes, where c is the number of image columns. There are 6 more columns in the storage area than in the image itself, because of the mirroring extension of 3 pixels at the image boundaries. The image pixels themselves take up one byte each. For an image of size 512×512 pixels, 3626 bytes of memory are allocated for image storage.

After an entire row has been inverse halftoned, rows 2–7 of the image storage area are moved upwards into the positions occupied by rows 1–6, and a new image row is written into the row 7 position. If circular buffering were available (as on dedicated digital signal processors), the block move could be avoided. However, the time penalty due to the move is small, because of the small block size, and because only one shift is needed for each row.

4.8 Results

It was mentioned in Section 4.2 that arguably the best inverse halftoning results to date have been produced by the wavelet-based method of Xiong, Orchard, and Ramchandran [26]. In this section, results from the proposed algorithm and the wavelet-based algorithm are compared. In addition, a model for inverse halftoning is presented that allows the perceptually weighted signal-to-noise (WSNR) metric given in Chapter 2 to be applied to inverse halftones.

The images displayed in this section, all of which are of size 512×512 pixels, have been reproduced at a large size to reduce the effect of the halftoning that occurs in the printer used to reproduce them.

4.8.1 Visual evaluation

Figure 4.10(a) shows the original *lena* image, while Figure 4.10(b) shows the Floyd-Steinberg halftone. Artifacts above the hat (containing tones close to (f_N, f_N)) and in the cheek (containing tones close to $(f_N, 0)$) are visible. Figure 4.11(a) shows the result of inverse halftoning the *lena* image using the proposed algorithm. The image shows a range of smooth and sharp areas; compare, for instance, the appearance of the interior of the shoulder with that of its edge where it overlaps the mirror. Artifacts are still visible in the area above the hat, where the Floyd-Steinberg halftone is quasi-periodic.

Figure 4.11(b) shows the result of wavelet-based algorithm. Its appearance is similar to Figure 4.11(a), although its artifacts are different in quality, with the image appearing better in some areas and worse in others. Overall, the wavelet image looks a little more natural, but it is noisier than the image produced by the proposed algorithm, and the edges are not as sharp. The increased noise is particularly visible in the cheek and nose.

Figure 4.12(a) shows the original *peppers* image, while Figure 4.12(b) shows the Floyd-Steinberg halftone. Figures 4.13(a) and 4.13(b) show the inverse halftones generated by the proposed scheme and the wavelet scheme, respectively. The image produced by the proposed scheme has sharper edges: the chile pepper at the left is more distinct, as is the stalk of the bell pepper.

(a) Original image.

(b) Floyd-Steinberg halftone.

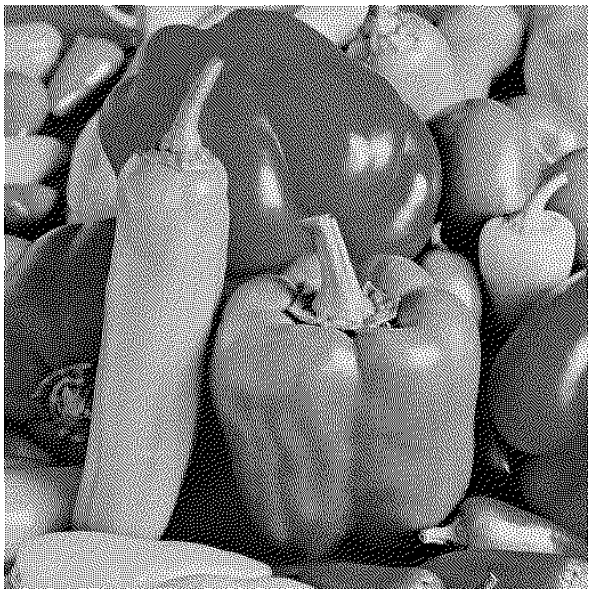
Figure 4.10: Original *lena* image and its halftone.

(a) Proposed algorithm. PSNR 31.34 dB.

(b) Wavelet algorithm. PSNR 31.47 dB.

Figure 4.11: Inverse halftoned *lena* images.

(a) Original image.



(b) Floyd-Steinberg halftone.

Figure 4.12: Original *peppers* image and its halftone.

(a) Proposed algorithm. PSNR 31.43 dB.

(b) Wavelet algorithm. PSNR 30.40 dB.

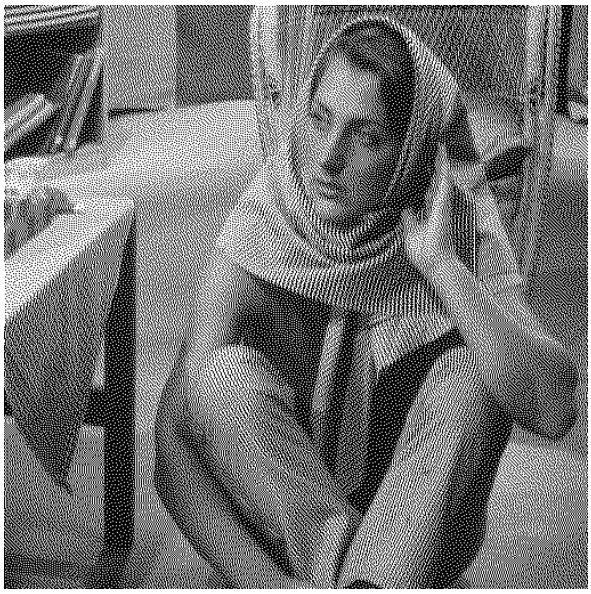
Figure 4.13: Inverse halftoned *peppers* images.

In the shadows, the wavelet image appears to have slightly lower noise.

Figure 4.14(a) shows the original *barbara* image, while Figure 4.14(b) shows the Floyd-Steinberg halftone. Figures 4.15(a) and 4.15(b) show the inverse halftones generated by the proposed scheme and the wavelet scheme, respectively. The *barbara* image is difficult to inverse halftone, because it contains strong high frequencies that effectively cannot be recovered from the halftone. The stripes in the trousers, for instance, have completely disappeared from both inverse halftones. However, the image produced by the proposed algorithm retains the sharp edges of the table leg and the books, and the skin on the face and arms is quite smooth. The edges in the wavelet image are not as sharp, and the smooth areas are noisier.

The preceding results have shown that inverse halftones recovered from Floyd-Steinberg halftones tend to be blurry. As an experiment, a halftone that was created with the filter due to Jarvis *et al.* was inverse halftoned using the proposed algorithm. Figure 4.16(a) shows the original *lena* image, while Figure 4.16(b) shows the inverse halftone computed from the Jarvis halftone. It is very similar in appearance to the original image, and in fact the two images must be examined closely before differences can be discerned. The absence of artifacts in the Jarvis halftone leads to accurate reproduction of the smooth region above the hat; compare Figure 4.16(b) to the results of Figure 4.11. Despite the mediocre PSNR of 28.46 dB, this result suggests that excellent results can be achieved by using Jarvis error diffusion for images that are likely to be subsequently inverse halftoned.

(a) Original image.



(b) Floyd-Steinberg halftone.

Figure 4.14: Original *barbara* image and its halftone.

(a) Proposed algorithm. PSNR 24.61 dB.

(b) Wavelet algorithm. PSNR 24.14 dB.

Figure 4.15: Inverse halftoned *barbara* images.

(a) Original image.

(b) Inverse halftone recovered from Jarvis *et al.* halftone.

Figure 4.16: Original *lena* image and its inverse halftone.

Algorithm & citation	Memory usage	Computational complexity	PSNR (dB)	
			lena	*peppers*
Wavelet [26]	$36N^2$	Medium	31.5	30.4
Kernel est. [25]	$8N^2$	Medium	32.0	30.2
Bayes [27]	$8N^2$	High	–	–
POCS [61]	$8N^2$	High	30.4	–
Proposed	$7N$	Low	31.3	31.4

Table 4.3: Comparison of inverse halftoning schemes. The memory requirements are byte estimates, assuming an image size of $N \times N$ pixels. Computational complexity is estimated from algorithm information given in the cited paper. "Low" complexity denotes fewer than 500 operations per pixel, "medium" denotes 500–2000 operations per pixel, and "high" denotes more than 2000 operations per pixel. PSNR figures are taken directly from the publications, where available.

4.8.2 Comparison with existing schemes

Table 4.3 compares the performance of four inverse halftoning schemes from the literature with the proposed algorithm. The schemes are compared on memory usage, computational complexity, and visual quality. Data on memory usage and computational complexity are often not given by the authors; these figures are estimated from the nature of the algorithm. PSNR is usually the only measure of image quality that is quoted, and figures are therefore reproduced here, despite the fact that PSNR is a poor indicator of image quality.

Table 4.3 shows that the proposed algorithm uses by far the least memory of any scheme, since it is the only scheme whose memory requirement increases linearly with N, rather than quadratically. Furthermore, it does not store copies of the image, as iterative schemes do. The computational complexity of the proposed algorithm is also considerably lower than the other schemes, all of which make heavy use of floating-point arithmetic. Neverthe-

less, the PSNR achieved for the standard images is comparable to the best schemes. (The large improvement in PSNR for the *peppers* image is due in part to an error in the original image. This error was corrected for this work, and was reported to the authors of the wavelet-based scheme [26].)

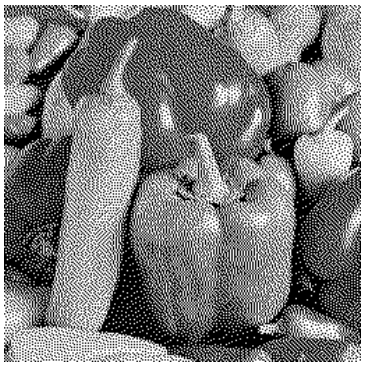
4.8.3 Measurements

It was discussed in Chapter 2 that noise-based metrics, such as SNR and PSNR, are inappropriate when the image distortion is not additive noise. An inverse halftone is not only corrupted by filtered quantization noise from the halftoning process, it is also blurred relative to the original image. Furthermore, the blurring is image-dependent and spatially-varying. Nevertheless, PSNR is often quoted as a figure of merit for inverse halftones.

A simple method of modeling the blurring of inverse halftoning was devised, with the aim of obtaining a residual between the inverse halftone and the modeled inverse halftone that is additive noise. This allows the level of the noise to be determined, and the blurring to be quantified. During inverse halftoning, the filter parameters x_1 and y_1 are saved, thereby keeping a record of the filter used at each pixel. This information is used to filter the *original* image using the same filters that were used to create the inverse halftone. This results in a noiseless image which has the same spatial blur as the inverse halftone. An example is shown in Figure 4.17.

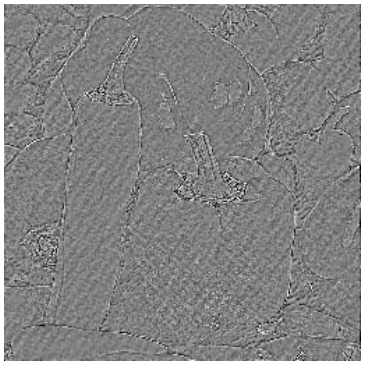
Figure 4.17(a) shows the original *peppers* image. Figure 4.17(b) shows the modified Floyd-Steinberg halftone, with the parameter L chosen to give a flat signal transfer function. The inverse halftone, shown in Figure 4.17(c),

(a) Original image.



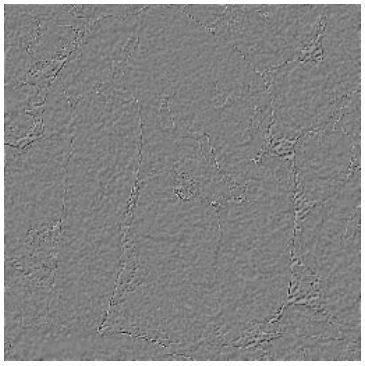
(b) Modified Floyd-Steinberg halftone.

(c) Inverse halftone.



(d) Residual (c) − (a). Gain of 4 applied.

(e) Model inverse halftone.



(f) Residual (c) − (e). Gain of 4 applied.

Figure 4.17: Result of modeling inverse halftoned *peppers* image.

Difference Image	Correlation Coefficient $C_{\text{original,difference}}$				
	barbara	*boats*	*lena*	*mandrill*	*peppers*
Inverse halftone − Original	0.364	0.259	0.189	0.546	0.229
Inverse halftone − Model	0.007	0.006	0.011	0.018	0.007

Table 4.4: Correlation coefficients for inverse halftone residuals. The first row shows the correlation coefficient between the original image and the (inverse halftone − original) residual. The second row shows the correlation coefficient between the original image and the (inverse halftone − inverse halftone model) residual.

is computed, and the filter parameters at each pixel are saved. The residual between the inverse halftone and the original image is shown in Figure 4.17(d). Strong image edges can be seen, because the inverse halftone is blurred. Figure 4.17(e) shows the modeled inverse halftone, computed from Figure 4.17(a) using the same filter set used to create Figure 4.17(c). Figure 4.17(f) shows the residual between the inverse halftone and the model. The image components are greatly reduced relative to Figure 4.17(d).

The quality of the results produced by the inverse halftoning model are evaluated using the correlation measure of (2.12). Table 4.4 shows the correlation between the original image and two residual images: the difference between the inverse halftone and the original image, and the difference between the inverse halftone and the modeled inverse halftone. On average, the correlation for the actual residual is 0.317, while the correlation for the modeled residual is 0.010. Image components are therefore suppressed by a factor of 33 in the modeled residual, on average. The low correlation of the original image and the modeled residual permits the use of modeled inverse halftones as a basis for perceptually weighted signal-to-noise (WSNR) measurements.

Reference Image	WSNR (dB)				
	barbara	*boats*	*lena*	*mandrill*	*peppers*
Original	20.47	25.36	26.93	19.02	27.69
Model	32.29	33.02	32.74	31.93	31.77

Table 4.5: WSNR measures for inverse halftones, $f_N = 20$ cycles/degree. The first row shows the WSNR between the inverse halftone and the original image. The second row shows the WNSR between the inverse halftone and the modeled inverse halftone.

Table 4.5 shows WSNR measurements for five test images, assuming a maximum spatial frequency in the x and y directions of 20 cycles/degree, which corresponds to a typical combination of image resolution, size, and viewing distance. The first row shows the WSNR of the inverse halftone relative to the original image, while the second row shows the WSNR of the inverse halftone relative to the modeled inverse halftone. The second of these figures is a true measure of the weighted noise content of the inverse halftones, since the first figure includes image distortions. As expected, WSNR is higher when the inverse halftone is compared to the modeled inverse halftone. It is also more stable across images, varying by 1.25 dB over the test set, compared to a variation of over 8.5 dB when image distortion is not taken into account.

By creating a clean image whose blur is identical to that of the inverse halftone, the blurring may be quantified by computing an effective transfer function for the inverse halftoning system, as follows:

- Compute the two-dimensional fast Fourier transform (FFT) of the original image and the modeled inverse halftone;
- Divide the model FFT by the original image FFT point-for-point, for spatial frequencies where the original image FFT is non-zero;

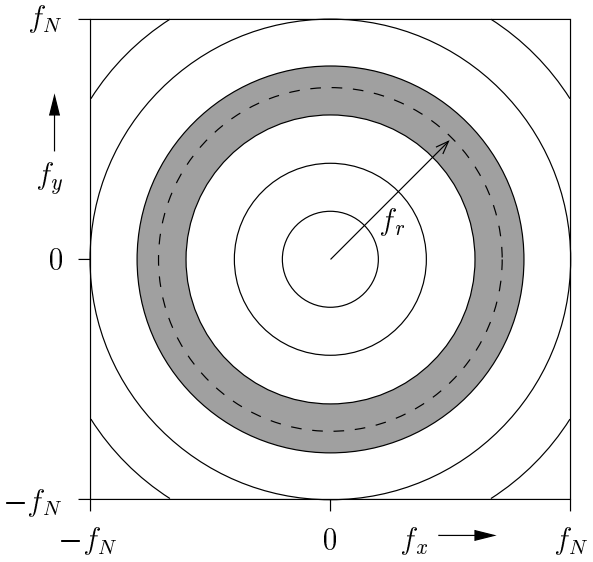


Figure 4.18: Radial averaging of the transfer function of the inverse halftoning system. The image is assumed to be square. The transfer function is averaged over each annulus (shown wider than actual size). The average magnitude over the shaded annulus is assigned to the radial frequency f_r.

- Compute the absolute value (magnitude) of the complex quotient to find the two-dimensional transfer function; and
- Radially average the transfer function over annuli of radius f_r to obtain a one-dimensional transfer function.

The radial averaging of the transfer function is depicted in Figure 4.18. The result is a one-dimensional transfer function that indicates the degree to which image components are suppressed in the inverse halftone.

Figure 4.19 shows the transfer functions for the *lena*, *peppers*, and *barbara* images. All show the marked high frequency suppression that is characteristic of blurring. It would be desirable to condense the transfer function into a single number to describe its shape, to permit easy comparison between

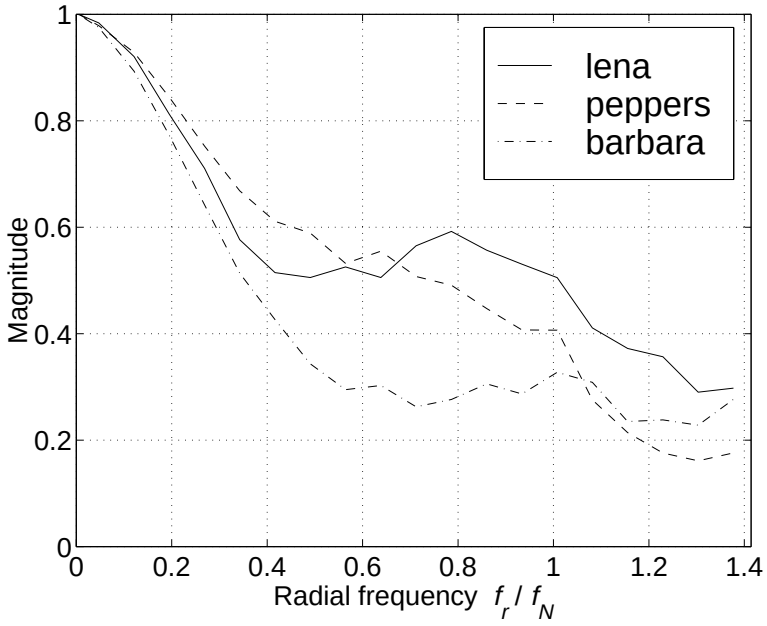


Figure 4.19: Radial transfer function of the proposed inverse halftoning scheme for the *lena*, *peppers*, and *barbara* images. Radial frequency $f_r = \sqrt{f_x^2 + f_y^2}$. The magnitude at f_r is the average transfer function magnitude over an annulus in the frequency domain with average radius f_r.

competing inverse halftoning schemes. Possible candidates are the radial frequency at which the response drops to a certain level, and the equivalent noise bandwidth of the transfer function [58]. However, not enough test images have been examined in this way to determine whether such a sparse description can account for all typical transfer functions. In addition, other inverse halftoning algorithms cannot be tested without modification to the code, which may be unavailable. Further research will show whether it is indeed possible to quantify halftoning performance by WSNR and equivalent noise bandwidth, rather than by measures such as PSNR.

4.9 Summary

A new inverse halftoning scheme based on anisotropic diffusion has been presented that produces high quality images from error diffused halftones at low computational cost. By combining work in gradient estimation and multiscale edge detection, a multiscale gradient estimator designed specifically for halftones was obtained. The control functions derived from the gradient estimates determine the cutoff frequencies of a specially-designed smoothing filter.

Inverse halftoning is modeled by filtering the original image and the halftone identically. This technique can be applied to any inverse halftoning scheme, and permits a true noise residual to be obtained, from which WSNR can be calculated. The modeled inverse halftone can be used to compute an effective transfer function for the scheme. This permits the meaningful comparison of competing schemes based on the amount of noise suppression and blurring they exhibit.

Chapter 5

Applications

This chapter develops and optimizes new algorithms for rehalftoning and interpolated halftoning. Rehalftoning converts one halftone into another type of halftone. Interpolated halftoning resizes an image before halftoning.

The rehalftoning algorithm presented here greatly reduces computation over conventional inverse halftoning followed by halftoning by using a simple inverse halftoning scheme and modified error diffusion. Blurring in the inverse halftone is corrected by designing the sharpness parameter for a flat system response, while noise is masked by the halftoning quantization noise. The linear gain model described in Chapter 3, and a polynomial approximation to the digital frequency $z = e^{j\omega}$, are used to derive the optimum value of the sharpness parameter. The weighted SNR metric described in Chapter 2 is used to assess the quality of the rehalftoned images.

The interpolated halftoning algorithm uses simplified interpolation to create high quality interpolated halftones. Computation is reduced over more complicated interpolation methods for the same visual quality. The linear gain model and the digital frequency approximation are again used to derive an optimum value for the sharpness parameter to flatten the system response.

5.1 Introduction

The purpose of rehalftoning is to convert a halftone created by one method to one created by a different method. For instance, a user might want to render a scanned error diffused halftone on a printer which performs best with screened halftones. The user may also wish to perform operations on the image at the same time, such as rotation or scaling, in which case the halftoning scheme used to generate the output may be the same as the one used for the input. Rehalftoning is needed for digital copiers, facsimile machines, and other devices which scan printed images and re-print them. Interpolation is used to resize images. The number of pixels in an image is increased by interpolating new pixels between existing pixels. The quality of the resulting image is strongly dependent on the interpolation scheme used.

This chapter describes new algorithms for rehalftoning and interpolated halftoning. For both applications, computation and memory usage are reduced over conventional methods by exploiting the characteristics of error diffused halftones. Specifically, the quantization noise introduced by the final halftoning step is used to mask artifacts due to the previous processing. Furthermore, modified error diffusion, described in Chapter 3, is used to create halftones which have a similar sharpness to the original images. The systems are analyzed with the linear gain model so that the sharpness parameter may be chosen to give a flat system transfer function.

Section 5.2 describes the design and analysis of a rehalftoning system for error diffused halftones that produces high quality results with minimal computation. Section 5.3 presents a combined interpolation and error diffu-

sion scheme that produces high quality interpolated halftones using simple interpolation methods. Section 5.4 concludes the chapter.

5.2 Rehalftoning

To perform rehalftoning, the halftone must in general be inverse halftoned, and then rehalftoned. Inverse halftoning attempts to recover a visually acceptable grayscale image from a halftone in reasonable time. Chapter 4 presented a new method of inverse halftoning that drastically reduces execution time over existing methods, for the same visual quality. Nevertheless, the required computation is still substantial. If it is known in advance that the inverse halftone will be rehalftoned, the requirements on visual quality of the inverse halftone can be relaxed, thereby reducing the computation.

Eschbach [78] has demonstrated a method of resizing images of arbitrary wordlength using printer and scanner models followed by adaptive error diffusion. The scanner model implements crude inverse halftoning by averaging pixels that fall within the assumed aperture of the scanner. The adaptive error diffusion rehalftones the image in a way that avoids pixel clumping that would occur with standard error diffusion.

In this section, a rehalftoning method designed for error diffused halftones is presented that has a far lower computational cost than conventional inverse halftoning followed by halftoning. Section 5.2.1 introduces rehalftoning. Section 5.2.2 describes the design procedure for the inverse halftoning filter. In Section 5.2.3, the entire rehalftoning system is analyzed by making use of the linear gain model from Chapter 3, and by using a polynomial

approximation to the digital frequency $z = e^{j\omega}$. The rehalftoning quality is rated by first compensating for the frequency distortion, and then applying the perceptually weighted SNR (WSNR) metric described in Chapter 2 to the residual image. Section 5.2.4 demonstrates examples of the processing that can be performed on the intermediate inverse halftone, and Section 5.2.5 evaluates the computational requirements of the algorithm.

5.2.1 Rehalftoning fundamentals

The quantization artifacts of a particular halftoning scheme can be used to mask the deficiencies of an inverse halftone. For instance, high frequency artifacts in an inverse halftone may be masked in the halftone by quantization noise. Thus, the inverse and forward halftoning schemes must be designed together to achieve optimum performance. Converting one error diffused halftone to another is useful in the following situations:

- When manipulation, such as rotation or scaling, must be performed on the halftone;
- When the halftone is too sharp or too dull; or
- When the rendering device is optimized for an error filter that differs from the one used to create the halftone.

Manipulation of sharpness is listed separately from other operations, because it can be accomplished while halftoning by using modified error diffusion [56].

To produce a high quality grayscale image from a halftone, the noise in smooth regions must be suppressed, while retaining sharpness in edge and textured regions. As discussed in Chapter 4, the effective support of the

smoothing filter must be large in the smooth regions to provide adequate noise suppression. Furthermore, the filter must be adaptive, else edges will be blurred. At the same time, one seeks computationally simple algorithms.

If a linear lowpass filter is used to perform the inverse halftoning, the inverse halftone will be either too smooth or too noisy, depending on the cutoff frequency of the filter. If the inverse halftone is subsequently halftoned using error diffusion, however, then the quantization noise introduced partially masks the noise that leaked through the linear lowpass filter, and the image is sharpened, which partially counteracts the blurring introduced by the filter. It is therefore possible to obtain a high quality halftone without employing an expensive inverse halftoning scheme.

5.2.2 Filter design

If the input to an error diffusion algorithm is itself a halftone, then the output is identical to the input, since the two input levels 0 (black) and 1 (white) are exactly equal to the two possible quantized output levels. Similarly, if the halftone is subject to screening, it will also be unchanged, because the input level 0 is less than all the thresholds in the screen, and the input level 1 is greater. In general, the output of standard error diffusion is equal to the input at pixels where the input is 0 or 1. This is because the quantization error is in the range $(-0.5, 0.5)$, and the error filter has a maximum gain of unity. Thus the feedback error is never large enough to force the input to the quantizer to cross the threshold when the input is 0 or 1. For any input image, therefore, the output is pre-determined to be 0 when the input is 0, and 1 when the input is 1. For intermediate values of the input, the output can be 0 or 1, depending

on previous outputs.

An image quantized to a short wordlength has a greater proportion of pixels with values 0 or 1 than if a longer wordlength is used. Thus, error diffusion has less freedom to disperse output pixels for short wordlength inputs, since the output is already fixed at many pixels. The loss of the ability to optimally disperse the output pixels leads to pixel clumping, with consequent artifacts. It is therefore important to use an input image of sufficient wordlength to obtain high quality halftones.

Figure 5.1 shows the result of using Floyd-Steinberg error diffusion to halftone *lena* images which have previously been reduced in wordlength to B bits using a Floyd-Steinberg coder with a 2^B-level quantizer. For instance, the grayscale original image used to obtain Figure 5.1(a) has four possible graylevels, while the grayscale image used for Figure 5.1(f) is the original 8-bit image. As the wordlength of the grayscale image increases, the detail in the halftone improves, and the apparent noise level goes down. The change in detail is especially noticeable in the eyes, lips, and feathers. Figures 5.1(e) and 5.1(f) are nearly identical, and are slightly better visually than Figure 5.1(d). This indicates that a wordlength of approximately 6 bits is sufficient to produce high quality error diffused halftones.

To minimize computation, a simple linear lowpass filter is used to perform inverse halftoning. As stated in Chapter 4, the lowpass filter should be short and FIR for ease of computation. It should also be symmetric, and have zeros at the band edges to eliminate the strong tones in halftones. In this instance, separability is unnecessary because the filter is applied non-separably.

(a) $B = 2$ bits.

(b) $B = 3$ bits.

(c) $B = 4$ bits.

(d) $B = 5$ bits.

(e) $B = 6$ bits.

(f) $B = 8$ bits.

Figure 5.1: Halftones obtained from original images of wordlength B. Key quality differences are in the lips, eyes, and feathers.

The filter must satisfy the following requirements:

- Small, FIR
- Zeros at $(f_N, -)$, $(-, f_N)$
- 6-bit output resolution

The output resolution is measured by computing the filter output for each possible binary input (of which there are $2^{(M^2)}$ for a filter of size $M \times M$ pixels), and counting the number of distinct outputs N. The output resolution R in bits is given by $R = \log_2(N)$. For instance, a boxcar averaging filter of size 2×2 pixels has five possible outputs, and a consequent output resolution of $\log_2 5 \approx 2.3$ bits. The only 3×3 filter that satisfies the first two criteria has a resolution of 4.1 bits.

The smallest filter which satisfies all of the criteria is of size 4×4 pixels. A filter was designed separably that balances the trade-off between sharpness and noise suppression, to give a reasonably artifact-free, sharp inverse halftone. The integer version of this filter is given by

$$h = \frac{1}{1024} \begin{bmatrix} 10 & 41 & 41 & 10 \\ 41 & 164 & 164 & 41 \\ 41 & 164 & 164 & 41 \\ 10 & 41 & 41 & 10 \end{bmatrix} .$$

This filter has 107 possible outputs, i.e., an average resolution of 6.74 bits.

Figure 5.2(b) shows the inverse halftone generated from the halftone of Figure 5.2(a). It is slightly blurred and somewhat noisy, as expected. Figure 5.2(c) shows the Floyd-Steinberg halftone computed from Figure 5.2(b). It is more blurred than the original halftone, but its noise level appears similar.

(a) Original *boats* halftone.

(b) Inverse halftone (4×4 filter).

(c) Floyd-Steinberg, $L = 0.0$.

(d) Floyd-Steinberg, $L = 1.5$.

(e) Jarvis, $L = 0.0$.

(f) Jarvis, $L = 0.8$.

Figure 5.2: 512×512 rehalftones obtained from linear lowpass inverse halftone.

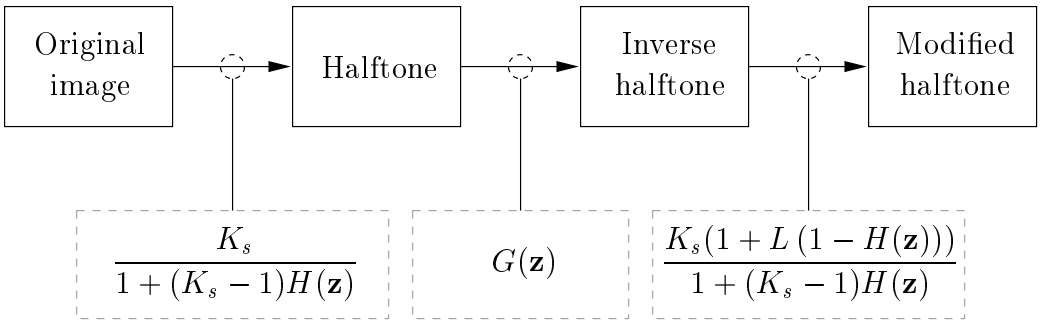


Figure 5.3: Signal modification in the rehalftoning chain. The dashed boxes show the signal transfer function at each step. K_s is the effective signal gain, which is dependent on the error filter, $H(\mathbf{z})$. $G(\mathbf{z})$ is the transfer function of the inverse halftoning filter. L is a free parameter that controls edge sharpening.

Figure 5.2(d) shows the modified error diffusion halftone using a sharpening factor $L = 1.5$. This image is about as sharp as the original halftone. Figures 5.2(e) and 5.2(f) show Jarvis halftones computed from Figure 5.2(b), with different values of L. Both are sharper than the original halftone and exhibit the low tonality that is characteristic of the Jarvis scheme. All four rehalftoned images in Figure 5.2 are free of artifacts, and show no signs of pixel clumping.

5.2.3 Analysis and measurements

Figure 5.3 shows the steps of the rehalftoning chain, and the signal transfer functions (STFs) associated with them. The linear gain model from Chapter 3 can be used to derive the STFs for the two halftoning steps. The STF of the system is given by the product of the three STFs shown in Figure 5.3. The following low frequency approximation for the digital frequency is used to obtain a polynomial expression for the STF:

$$z = e^{j\omega} \approx 1 + j\omega - \frac{\omega^2}{2}\ , \tag{5.1}$$

which is obtained by using the series formula $e^x = 1 + x + \frac{x^2}{2!} + \ldots$. The expression in (5.1) is accurate to approximately 10% (real part) and 20% (imaginary part) at $\omega = 1$ radian/sample. Since most of the energy in natural images falls in the lower spatial frequencies, and noise power from halftoning swamps image noise at higher frequencies, the use of (5.1) is justified.

The STFs in Figure 5.3 can be simplified by assuming that Floyd-Steinberg halftoning is used, and that $K_s = 2$. The transfer function of the system is

$$T(e^{j\vec{\omega}}) = \frac{4G(e^{j\vec{\omega}})(1 + L(1 - H(e^{j\vec{\omega}})))}{(1 + H(e^{j\vec{\omega}}))^2} \ , \tag{5.2}$$

where $\vec{\omega} = (\omega_x, \omega_y)$, the two-dimensional frequency vector. The error filter is given by $H(e^{j\vec{\omega}}) = \frac{1}{16}[7e^{-j\omega_x} + e^{-j\omega_y}(e^{-j\omega_x} + 5 + 7e^{j\omega_x})]$.

$T(\vec{\omega})$ is found by inserting (5.1) into (5.2) and retaining up to quadratic terms in (ω_x, ω_y). The reciprocal of the denominator is evaluated using the expansion $(1 + x)^{-1} = 1 - x + x^2 + \ldots$, and multiplied by the numerator. Considerable algebra leads to the intermediate result

$$\begin{aligned} T(\vec{\omega}) \ = \ & G(e^{j\vec{\omega}}) \Big(1 \ + \ \frac{5}{16} j\omega_x (1 + L) + \frac{9}{16} j\omega_y (1 + L) \\ & + \ \frac{\omega_x^2}{1024}(277 + 252L) + \frac{\omega_y^2}{1024}(45 - 36L) \\ & - \ \frac{\omega_x \omega_y}{1024}(398 + 488L) + O(\omega^3) \Big) \ . \end{aligned} \tag{5.3}$$

The transfer function of the inverse halftoning filter is

$$\begin{aligned} G(e^{j\vec{\omega}}) \ = \ & \frac{1}{1024} \Big(\ (e^{-j\omega_y} + e^{2j\omega_y})[10(e^{-j\omega_x} + e^{2j\omega_x}) + 55(1 + e^{j\omega_x})] \\ & + \ (1 + e^{j\omega_y})[41(e^{-j\omega_x} + e^{2j\omega_x}) + 164(1 + e^{j\omega_x})] \Big) \ . \end{aligned} \tag{5.4}$$

By inserting (5.4) into (5.3) and retaining up to quadratic terms in (ω_x, ω_y), one obtains

$$\begin{aligned} T(\vec{\omega}) &= 1 + \frac{j\omega_x}{16}(13+5L) + \frac{j\omega_y}{16}(17+9L) \\ &\quad + \frac{\omega_x^2}{1024}(-343+92L) + \frac{\omega_y^2}{1024}(-703-324L) \\ &\quad + \frac{\omega_x\omega_y}{1024}(-1102-936L) \ . \end{aligned} \tag{5.5}$$

This equation can be solved for L to achieve an approximately flat response in the ω_x and ω_y directions independently. When $\omega_y = 0$, (5.5) becomes

$$T(\omega_x) = 1 + \frac{j\omega_x}{16}(13+5L) + \frac{\omega_x^2}{1024}(-343+92L) \ , \tag{5.6}$$

which has the form $T(\omega_x) = 1 + aj\omega_x + b\,\omega_x^2$. It is required that $|T(\omega_x)| = 1$, and therefore that $|1 + aj\omega_x + b\,\omega_x^2| = 1$. Thus

$$\Big((1+b\,\omega_x^2)^2 + (a\omega_x)^2\Big)^{\frac{1}{2}} = 1 \ , \quad \omega_x \ll 1 \ . \tag{5.7}$$

Squaring both sides of (5.7) and expanding it in powers of ω_x up to $O(\omega^2)$ gives $1 + 2b\,\omega_x^2 + a^2\omega_x^2 = 1$. The solution of (5.7) is therefore

$$a^2 + 2b = 0 \ , \tag{5.8}$$

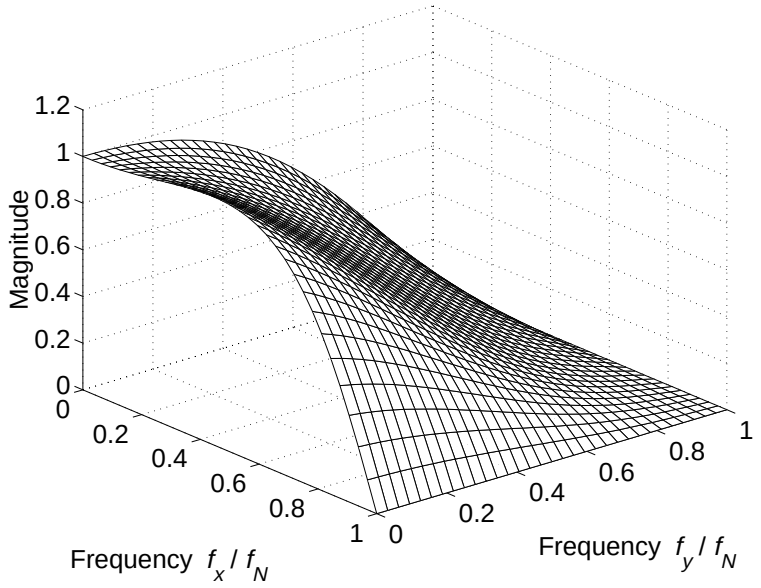
where a and b are the coefficients of $j\omega_x$ and ω_x^2 in (5.5), respectively. The same equation holds for ω_y. Solving (5.8) for L, one obtains the result

$$\left.\begin{array}{rcll} L &=& 0.014 & (x \text{ direction}) \\ L &=& 0.361 & (y \text{ direction}) \end{array}\right\} \ L = 0.188, \text{ on average} \ . \tag{5.9}$$

Although it is not possible to choose a value of L that simultaneously flattens $T(e^{j\vec{\omega}})$ in ω_x and ω_y, the average value of L in (5.9) gives good results, since the spread of the optimum values for ω_x and ω_y is small. Figure 5.4(a) shows

(a) Original *food* image.

(b) Rehalftone ($L = 0.188$).



(c) Transfer function $T(e^{j\vec{\omega}})$.

Figure 5.4: 512×512 rehalftone computed using $L = 0.188$ to give the flattest spectrum around DC. Floyd-Steinberg error diffusion is used [14].

an original image named *food*, while Figure 5.4(b) shows the rehalftone, computed using $L = 0.188$. It has a similar sharpness to the original, as expected. Figure 5.4(c) shows the magnitude response of the corresponding signal transfer function $T(e^{j\vec{\omega}})$. It is quite flat around DC. Because the average value of L in (5.9) is higher than the optimum value for ω_x, $T(e^{j\vec{\omega}})$ rises slightly along the ω_x axis (labeled f_x). Similarly, $T(e^{j\vec{\omega}})$ falls slightly along the ω_y (f_y) axis, because the average L is lower than the optimum value for ω_y.

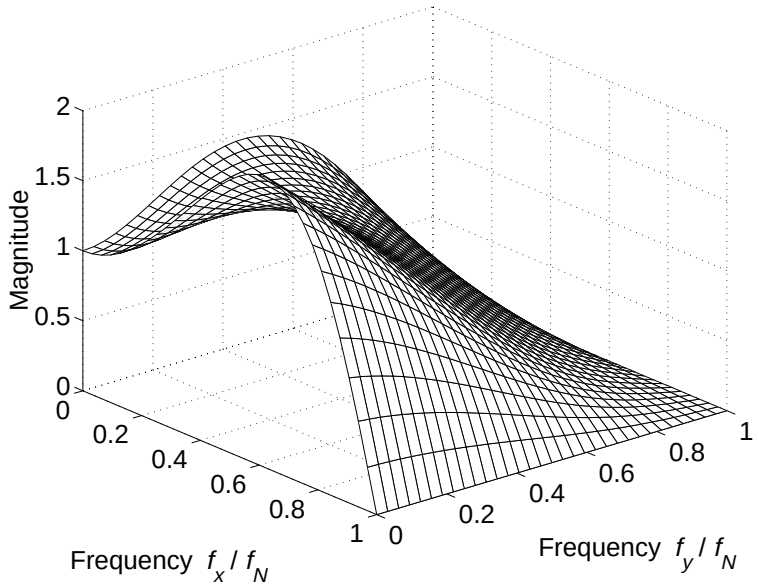
The image can be sharpened by increasing L beyond the value for optimum flatness. Figure 5.5(b) shows the rehalftoning result for $L = 1.5$. The image is somewhat sharper than the original. The corresponding signal transfer function is shown in Figure 5.5(c). There are peaks in the midband along each axis that increase the apparent sharpness of the rehalftone.

The WSNR of rehalftones is measured using a combination of the methods presented in Chapters 3 and 4. Modified error diffusion with a flat STF ($L = -0.5$) is used for the two halftoning steps, and a model inverse halftone is constructed by filtering the original image with the inverse halftoning filter. The residual between this model and the rehalftone has a very low correlation with the original image, averaging less than 0.001 for the images tested.

Table 5.1 shows measured values of WSNR for five rehalftoned images, compared to the WSNR of the original halftones. For all the tested images, the rehalftone has a slightly poorer WSNR than the direct halftone, as expected. However, the difference is small, amounting to less than 3 dB at large viewing distances. These results indicate that the fixed inverse halftoning filter is adequate for rehalftoning.

(a) Original *food* image.

(b) Rehalftone ($L = 1.5$).



(c) Transfer function $T(e^{j\vec{\omega}})$.

Figure 5.5: 512×512 rehalftone computed using $L = 1.5$ to give a sharper image. Floyd-Steinberg error diffusion is used.

Max. ang. freq. (cyc/deg)	WSNR (dB)									
	boats		*barbara*		*food*		*lena*		*mandrill*	
	RH	OH	RH	OH	RH	OH	RH	OH	RH	OH
20	10.2	10.4	8.6	8.7	11.0	11.3	9.4	9.5	9.5	9.7
40	22.1	23.0	20.6	21.3	21.7	22.5	21.6	22.4	21.7	22.5
60	29.5	31.5	28.1	29.9	28.2	29.7	29.0	31.0	28.9	30.9
80	33.3	36.1	32.0	34.5	32.0	33.9	32.7	35.4	32.7	35.4

Table 5.1: WSNR measurements of rehalftones, compared to direct halftones, for various viewing distances. Columns labeled 'RH' show the WSNR in dB of the rehalftone, relative to the original image blurred by the inverse halftoning filter, $G(\mathbf{z})$. Columns labeled 'OH' show the WSNR in dB of the original halftone, relative to the original image.

5.2.4 Intermediate processing

Although the intermediate inverse halftone in rehalftoning is noisy, and therefore not suitable for display in its own right, it has sufficient wordlength that operations such as rotation, scaling, and contrast adjustment may be successfully applied. In general, these operations are not possible with halftones [23], because they either cause wordlength expansion, or destroy the quality of the halftone.

Figure 5.6 shows examples of operations that can be performed on the intermediate inverse halftone. Figure 5.6(a) shows the original *food* halftone. Figure 5.6(b) shows the image resized to two-thirds of its original size, Figure 5.6(c) shows a rotation of 25 degrees, and Figure 5.6(d) shows a nonlinear contrast reduction. Although the first two operations can be performed on halftones using the technique described in [78], the resulting image quality suffers. The contrast adjustment cannot be performed.

(a) Original halftone, 512×512.

(b) Resized to 340×340 pixels.

(c) Rotated by 25 degrees.

(d) Contrast reduced.

Figure 5.6: Halftones obtained by processing the intermediate inverse halftones generated from the *food* original halftone.

5.2.5 Computational requirements

The inverse halftoning portion of the rehalftoning algorithm has a far lower computational requirement than even the efficient inverse halftoning algorithm presented in Chapter 4, since it consists solely of a small, fixed FIR filter. Only four rows of the image need to be stored in memory at one time. The computational requirement of error diffusion is also small. Computation is further reduced by exploiting the fact that some operations are common to both parts of the algorithm, such as looping over the image and writing results to the output. The rehalftoning algorithm requires the following number of operations per pixel:

- 34 increments (`++`)
- 12–28 integer additions
- 4 integer multiplications
- 2 bit shifts

No floating-point operations are needed. The number of additions varies according to the input, and is equal to 20 on the average for a mid-gray image. For an image of size 512×512 pixels, the entire rehalftoning process requires approximately 16 million operations. The C implementation takes less than 0.4 seconds to execute on a 167 MHz Sun UltraSparc 2 machine, and less than 1.2 seconds on a Sparc 10, for a 512×512 halftone. This implementation requires $4(c+3)$ bytes of storage for the image, where c is the number of image columns. Thus only 2060 bytes of memory are allocated for a 512×512 image. Because all operations are local and use integer arithmetic, the algorithm is ideal for implementation in embedded software.

5.3 Interpolation

An image often needs to be resized for printing, so that it appears at the correct size on the page. For instance, an image which is sized correctly for a printer with a resolution of 300 dpi (dots per inch) will be half the size when printed on a 600 dpi printer. In such instances, the image must be resized by interpolation before halftoning. Several interpolation methods are in common use, and are listed here in order of computational complexity [79]:

- Nearest neighbor interpolation;
- Bilinear interpolation; and
- Higher order functions, such as bicubic interpolation, lowpass filtering, cubic spline interpolation, etc.

Interpolation assumes that the pixel values of an image represent samples on an integer grid of an intensity function, $I(x, y)$, that is defined over the entire plane. To resize the image, a grid of output points is constructed, and $I(x, y)$ is interpolated at these new points. The interpolation scheme defines how the intensity at each pixel is constructed.

The interpolation method used depends on the required quality of the resulting image, and the computation power available. If an image is intended for printing, it makes little sense to perform a computationally expensive interpolation, since improvements in the resulting image will probably be masked by the halftoning process. Furthermore, modified error diffusion can be used to sharpen images which are blurred by simple interpolation schemes. This section shows how simultaneous design of the interpolation scheme and the sharpness parameter in error diffusion leads to high quality images at low

computational cost. Section 5.3.1 describes common interpolation methods, and Section 5.3.2 presents transfer functions and example images for two of these methods. Section 5.3.3 optimizes the design of a combined interpolation and error diffusion system for two interpolation schemes. Section 5.3.4 evaluates the computational requirements of the algorithm.

5.3.1 Common interpolation methods

Nearest neighbor interpolation uses the intensity at the pixel nearest to the new pixel; that is, it assumes that $I(x, y)$ is constant between input pixels. It is equivalent to replicating pixels if the image size is increased, and deleting pixels if the image size is reduced, and is therefore very fast. However, the interpolated image usually appears blocky, because of aliasing. Bilinear interpolation assumes that $I(x, y)$ varies linearly in the x and y directions over the rectilinear area between four neighboring input pixels; that is, the area with an input pixel at each corner. The interpolated output $I'(x, y)$ is computed as follows:

$$\begin{aligned} x' &= x - \lfloor x \rfloor \\ y' &= y - \lfloor y \rfloor \\ I_1 &= y' I(\lfloor x \rfloor, \lfloor y \rfloor + 1) + (1 - y') I(\lfloor x \rfloor, \lfloor y \rfloor) \\ I_2 &= y' I(\lfloor x \rfloor + 1, \lfloor y \rfloor + 1) + (1 - y') I(\lfloor x \rfloor + 1, \lfloor y \rfloor) \\ I'(x, y) &= x' I_2 + (1 - x') I_1 \ . \end{aligned}$$

The intermediate intensities I_1 and I_2 have been interpolated in the y direction. In the last step, the intensity is interpolated in the x direction between I_1 and I_2. The assumption that $I(x, y)$ varies linearly between pixels fails at sharp edges, and the interpolated image therefore appears smoother than

the original. Higher order functions, such as bicubic interpolation and spline interpolation, assume that $I(x, y)$ is a higher order function of (x, y) than linear. This requires more neighboring pixels to be included in the estimate of the interpolated output, thereby increasing computation time. However, the resulting image is generally sharper than if bilinear interpolation were used.

5.3.2 One-dimensional analysis

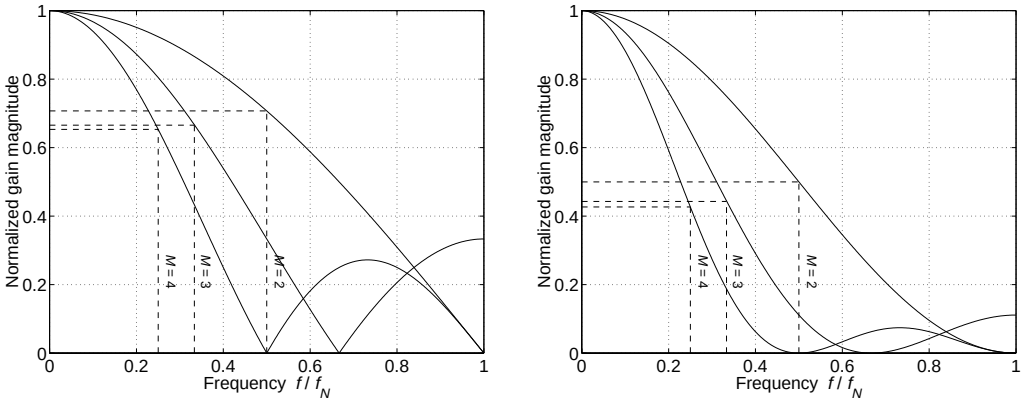
In the special case where an image is increased in size by an integer factor along each dimension, a proportion of the output pixels will be exactly equal to the input pixels, and do not need to be interpolated. In this instance, interpolation is equivalent to *upsampling* the original image by an integer factor in each direction, followed by filtering with an equivalent interpolating filter.

In one dimension, a signal $x[n]$ is upsampled by an integer factor M by inserting $M-1$ zeros between each sample, to produce the upsampled signal $y[n]$. The two signals are related in the frequency domain by

$$Y(e^{j\omega}) = \frac{1}{M} X(e^{j\omega M}) \ . \tag{5.10}$$

The effect of upsampling, apart from the gain of $\frac{1}{M}$, is to compress the spectrum of $x[n]$ by a factor of M, so that it occupies the baseband of $Y(e^{j\omega})$ from DC to $\frac{f_N}{M}$. The spectrum from $\frac{f_N}{M}$ to f_N is filled with $M-1$ images of the baseband spectrum. The ideal interpolator removes these images, while leaving the baseband spectrum intact [54]. Linear filtering approximates the ideal lowpass interpolator at a reasonable computational cost.

In one dimension, nearest neighbor interpolation is equivalent to convolving the upsampled signal with an FIR filter of length M with unity coef-



(a) Nearest neighbor. (b) Linear.

Figure 5.7: Frequency responses of two common interpolation functions, for upsampling ratios of 2, 3, and 4. The passband edges and their associated gains are shown dashed.

ficients. Such a filter has the frequency response

$$\begin{aligned} H_{\mathrm{NN}}(e^{j\omega}) &= 1 + e^{-j\omega} + e^{-2j\omega} + \ldots + e^{-(M-1)j\omega} \\ &= \frac{1 - e^{-jM\omega}}{1 - e^{j\omega}} \ . \end{aligned} \tag{5.11}$$

The magnitude of $H_{\mathrm{NN}}(e^{j\omega})$, normalized by the gain at DC, is plotted in Figure 5.7(a) for $M = \{2, 3, 4\}$. There are $\lfloor \frac{M}{2} \rfloor$ zeros spaced by $\frac{2f_N}{M}$ throughout the band, with the first zero being at $\frac{f_N}{M-1}$. The response falls off monotonically from DC, and is equal to 0.5 (-3 dB) at the passband edge for $M = 2$, where the passband is defined as frequencies below $\frac{f_N}{M}$. As $M \to \infty$, the gain at the passband edge approaches $\frac{2}{\pi} \approx 0.637$ (-3.9 dB). When the interpolator is applied separably in two dimensions, the normalized gain at the passband edge is halved over the one-dimensional case. The response is therefore -6 dB for $M = 2$, falling to -7.8 dB as M becomes large. This leads to blurring of the image.

The one-dimensional linear interpolator is formed by convolving the nearest neighbor interpolator with itself, giving it a triangular impulse response. Its frequency response is given by

$$\begin{aligned} H_{\mathrm{LI}}(e^{j\omega}) &= 1 + \frac{M-1}{M}(e^{j\omega} + e^{-j\omega}) + \frac{M-2}{M}(e^{2j\omega} + e^{-2j\omega}) + \ldots \\ &\quad + \frac{1}{M}(e^{(M-1)j\omega} + e^{-(M-1)j\omega}) \\ &= \left(\frac{1 - e^{-jM\omega}}{1 - e^{j\omega}}\right)^2 . \end{aligned} \tag{5.12}$$

This response is plotted for $M = \{2, 3, 4\}$ in Figure 5.7(b). It is the square of the nearest neighbor response. The stopband suppression is therefore greater, at the expense of the passband gain, which is more sharply rolled off than the nearest neighbor response. The normalized response at the passband edge is 0.25 (-6 dB) for $M = 2$. As $M \to \infty$, the gain asymptotically approaches $\frac{4}{\pi^2} \approx 0.405$ (-7.8 dB). When the bilinear interpolator is applied separably in two dimensions, the gain is reduced by 12 dB at the passband edge for $M = 2$, and by 13.9 dB in the limit as M becomes large. The blurring of the image is obvious, but the blocking artifacts that arise with nearest neighbor interpolation from inadequate suppression of baseband images are greatly reduced.

Figure 5.8(a) shows the original *cameraman* image. In Figures 5.8(b) and 5.8(c), the central part of the image has been zoomed by a factor of 2, using nearest neighbor and bilinear interpolation, respectively. Figures 5.8(d) and 5.8(e) zoom by a factor of 3. The nearest neighbor interpolated images are blockier than the bilinear interpolated images, but they are also sharper. In addition, they require far less time to construct.

(a) Original image.

(b) Nearest neighbor, 2×.

(c) Bilinear, 2×.

(d) Nearest neighbor, 3×.

(e) Bilinear, 3×.

Figure 5.8: Interpolated *cameraman* images. All images are 256×256.

5.3.3 Halftoning interpolated images

If an interpolated image is halftoned by error diffusion, the blocking artifacts will be masked to a certain extent by the quantization noise. Furthermore, the sharpness parameter in modified error diffusion can be used to correct for the blurring of the interpolation filter. Here, the $M = 2$ case is considered, since printer resolutions tend to be related by factors of 2 (300, 600 and 1200 dpi being current common values of print resolution), and therefore a doubling of image size is likely to be used more often than scaling by a different factor. The analysis is analogous for other scaling factors.

The low frequency approximation of (5.1) is used to analyze the compound system of interpolation followed by modified error diffusion. The STF for modified Floyd-Steinberg error diffusion, assuming that $K_s = 2$, is

$$H_{\mathrm{FS}}(e^{j\vec{\omega}}) = \frac{2(1 + L(1 - H(e^{j\vec{\omega}})))}{1 + H(e^{j\vec{\omega}})} \ , \tag{5.13}$$

where L is the sharpness parameter, and $\vec{\omega} = (\omega_x, \omega_y)$. After applying the approximation of (5.1) and retaining up to quadratic terms in (ω_x, ω_y), (5.13) becomes

$$H_{\mathrm{FS}}(\vec{\omega}) = 1 + \frac{1 + 2L}{1024}\left(160j\omega_x + 288j\omega_y + 151\omega_x^2 + 63\omega_y^2 - 154\omega_x\omega_y\right) \ . \tag{5.14}$$

Note that if $L = -0.5$, (5.14) reduces to $H_{\mathrm{FS}}(\vec{\omega}) = 1$; that is, the frequency response is flat. This agrees with (3.40), which is not an approximation.

The frequency response of the two-dimensional nearest neighbor interpolator for $M = 2$ is

$$H_{\mathrm{NN}}(e^{j\vec{\omega}}) = (1 + e^{-j\omega_x})(1 + e^{-j\omega_y}) \ , \tag{5.15}$$

which becomes, after applying (5.1) and retaining up to quadratic terms in (ω_x, ω_y),

$$H_{\mathrm{NN}}(\vec{\omega}) = 4 - 2j(\omega_x + \omega_y) - \omega_x^2 - \omega_y^2 - \omega_x\omega_y \ . \tag{5.16}$$

The frequency response of the bilinear interpolator for $M = 2$ is

$$H_{\mathrm{BI}}(e^{j\vec{\omega}}) = (\tfrac{1}{2}e^{j\omega_x} + 1 + \tfrac{1}{2}e^{-j\omega_x})(\tfrac{1}{2}e^{j\omega_y} + 1 + \tfrac{1}{2}e^{-j\omega_y}) \ , \tag{5.17}$$

which approximates to

$$H_{\mathrm{BI}}(\vec{\omega}) = 4 - \omega_x^2 - \omega_y^2 \ . \tag{5.18}$$

The transfer functions $H_{\mathrm{NN}}(e^{j\vec{\omega}})$ and $H_{\mathrm{BI}}(e^{j\vec{\omega}})$ have a gain of 4 at DC to compensate for the upsampling gain of 0.25. In the following analysis, the upsampling gain is combined with the transfer function of the interpolator, so the system has unity gain at DC.

The system composed of nearest neighbor interpolation followed by error diffusion has a response given by the product of (5.14) and (5.16):

$$\begin{aligned} H_{\mathrm{NN-FS}}(\vec{\omega}) \ = \ & 1 + \frac{j\omega_x}{1024}(-352 + 320L) + \frac{j\omega_y}{1024}(-224 + 576L) \\ & + \frac{\omega_x^2}{1024}(-25 + 462L) + \frac{\omega_y^2}{1024}(-49 + 414L) \\ & + \frac{\omega_x\omega_y}{1024}(-186 + 140L) \ . \end{aligned} \tag{5.19}$$

The response of the bilinear interpolation and error diffusion system is given by the product of (5.14) and (5.18):

$$\begin{aligned} H_{\mathrm{BI-FS}}(\vec{\omega}) \ = \ & 1 + \frac{5j\omega_x}{32}(1 + 2L) + \frac{9j\omega_y}{32}(1 + 2L) \\ & + \frac{\omega_x^2}{1024}(-105 + 302L) + \frac{\omega_y^2}{1024}(-193 + 126L) \\ & + \frac{\omega_x\omega_y}{1024}(-154 - 308L) \ . \end{aligned} \tag{5.20}$$

For each scheme, one can find the value of L that maximizes the flatness of the STF at low frequency by applying (5.8). For the nearest neighbor interpolator, one obtains

$$\left.\begin{array}{rcll} L & = & -0.102 & (x \text{ direction}) \\ L & = & 0.0813 & (y \text{ direction}) \end{array}\right\} \; L = -0.0105, \text{ on average .} \tag{5.21}$$

For the bilinear interpolator, the result is

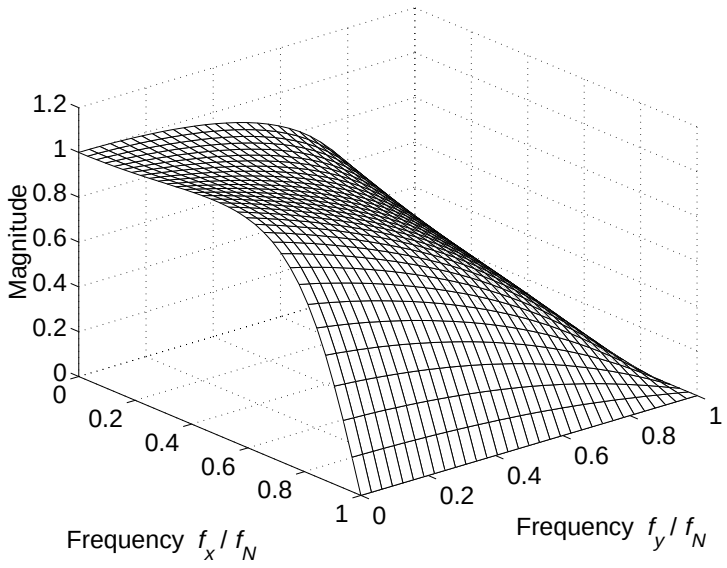
$$\left.\begin{array}{rcll} L & = & 0.254 & (x \text{ direction}) \\ L & = & 0.427 & (y \text{ direction}) \end{array}\right\} \; L = 0.340, \text{ on average .} \tag{5.22}$$

The combined interpolation and halftoning systems were tested by creating an image of size 256×256 pixels by filtering and subsampling a 512×512 original image. The smaller image was then scaled by a factor of two in each direction and interpolated, to obtain a 512×512 approximation to the original image. Since spectral energy above $\frac{f_N}{2}$ in the original image is lost when creating the 256×256 image, and cannot be recovered by interpolation, the interpolated image looks blurred with respect to the original, regardless of the interpolation scheme used. Therefore, a halfband filtered version of the original 512×512 image was created for comparison by using a lowpass filter with approximately unity gain from DC to $\frac{f_N}{2}$, and zero gain from $\frac{f_N}{2}$ to f_N. This allows the two interpolation schemes to be compared more easily.

Figure 5.9(b) shows the result of nearest neighbor interpolation, followed by modified error diffusion, using the average L defined in (5.21). Figure 5.9(a) shows the halftoned, halfband filtered original. The two images appear very similar. Some blockiness can be seen in the interpolated image, but the effect is slight. Figure 5.9(c) shows the transfer function of the system. As predicted by (5.21), it is substantially flat around DC, with a slight rise along

(a) Halftoned, filtered *food* image.

(b) Nearest neighbor ($L = -0.0105$).



(c) Transfer function $T(e^{j\vec{\omega}})$.

Figure 5.9: A 512×512 halftone with maximally flat spectrum around DC. It is computed by interpolating the 256×256 original using nearest neighbor interpolation, followed by modified Floyd-Steinberg error diffusion.

the ω_x axis and a slight drop along the ω_y axis, since L falls between the optimum value for each direction.

Figure 5.10 shows the corresponding results for bilinear interpolation, using the average L from (5.22). The interpolated image in Figure 5.10(b) also has similar sharpness to the halftoned, halfband filtered original. No blockiness can be discerned. The system transfer function in Figure 5.10(c) is again substantially flat around DC, although the response falls off quicker than the nearest neighbor response. This gives the interpolated halftone a slightly smoother look. However, the difference is small, and could be corrected perceptually by increasing L above its optimum value.

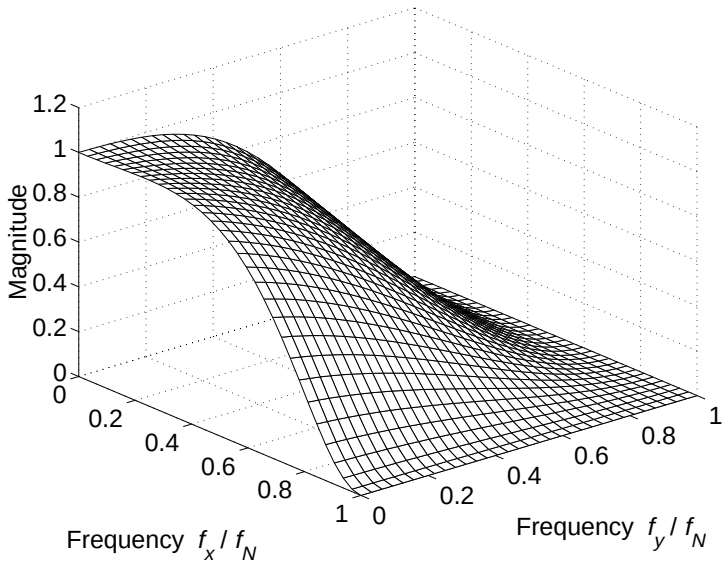
5.3.4 Computational requirements

The computational efficiency of interpolated halftoning stems from the use of simple interpolation schemes. Nearest neighbor interpolation has essentially no overhead; to convert a halftoning algorithm to an interpolated halftoning algorithm, only the order in which image pixels are addressed need be changed. Bilinear interpolation requires 7 additions and 6 multiplications to compute each output pixel. For interpolation by a factor of two, this reduces to an average of 1.67 additions and 1 bit shift per pixel. Both interpolated halftoning methods use modified error diffusion for the halftoning step. However, the optimum value for the sharpness parameter for nearest neighbor interpolation is so close to zero that conventional error diffusion may be used with no effect on visual quality. For bilinear interpolation by a factor of two, the algorithm requires the following number of operations per pixel:

- 2 increments (++)

(a) Halfband filtered *food* image.

(b) Bilinear ($L = 0.340$).



(c) Transfer function $T(e^{j\vec{\omega}})$.

Figure 5.10: A 512×512 halftone with maximally flat spectrum around DC. It is computed by interpolating the 256×256 original using bilinear interpolation, followed by modified Floyd-Steinberg error diffusion.

- 9.67 (7) integer additions
- 4 (3) integer multiplications
- 3 (2) bit shifts

where numbers in parentheses refer to nearest neighbor interpolation. No floating point operations are needed. Two rows of the image need to be stored for bilinear interpolation, while only one row is needed for nearest neighbor interpolation. Because all operations are local and use integer arithmetic, the algorithm is ideal for implementation in embedded software.

5.4 Summary

This chapter developed and optimized new, fast algorithms for rehalftoning and interpolated halftoning. The algorithms are suitable for implementation in embedded hardware, for use in printers, facsimile machines, scanners, and so on. In both applications, a polynomial approximation to the digital frequency $z = e^{j\omega}$, together with results from Chapters 3 and 4, were used to derive optimum values for the sharpness parameter in modified error diffusion to flatten the system transfer function.

The linear gain model and the digital frequency approximation allow the signal transfer function of an error diffusion scheme to be expressed as a polynomial in (ω_x, ω_y). This leads to solutions for L that are quadratic in ω. It was shown that these solutions are extremely accurate in all instances. Although the Floyd-Steinberg scheme was used for simplicity, the method can be applied to any error filter.

The rehalftoning scheme presented in this chapter is useful now that

scanning, processing, and re-printing documents is common. Commercial digital copiers must perform inverse halftoning on halftones that are embedded in documents before they can re-print them. This chapter has shown that good results can be obtained by using a combination of simple linear filtering and modified error diffusion. This greatly reduces the computation required. The WSNR quality metric from Chapter 2 was applied to rehalftones generated using the new method, and indicated that they are only slightly noisier than the original halftones.

The interpolated halftoning scheme presented in this chapter demonstrates that accurate images can be obtained with nearest neighbor and bilinear interpolation, without the need to use more costly schemes. If computation is at a premium, good results can be achieved with nearest neighbor interpolation, since the blockiness is masked to a certain extent by the quantization noise of error diffusion. Excellent quality results can be obtained at a slightly higher cost by using bilinear interpolation, with the blurring corrected by choosing the correct value of L. This produces halftones that are essentially indistinguishable from those produced by the best interpolation methods.

Chapter 6

Conclusions

Although this dissertation has covered a lot of ground, two themes are common to the entire work. First, the importance of modeling the effects of an image processing system to account for the distortions that it introduces has been demonstrated. This enables a residual image to be generated that has a low correlation with the original image, which in turn permits the use of noise-based visual quality metrics, such as the weighted signal-to-noise ratio (WSNR) presented in Chapter 2. Second, it has been demonstrated that the model itself allows the distortions of the image processing system to be characterized separately from the noise injected by the system. Metrics generated from the model can then be provided along with WSNR results to quantify the performance of the system. In other words, by modeling the system, one is able to obtain *objective measures* of the *subjective quality* of images.

Chapter 2 demonstrates the need to model the frequency distortion in an image before applying a weighted noise metric. The linear contrast sensitivity function (CSF) that is used to weight the residual image is a simple model of the human visual system. A possible way to improve the correlation between the noise metric and visual quality would be to use a more detailed

model, such as that described by Peli [80]. This model takes into account effects such as local contrast and frequency masking. When combined with models for forward and inverse halftoning, it should provide noise figures which are in excellent agreement with subjective results.

Chapters 3 and 4 demonstrate ways of modeling forward and inverse halftoning so that separate figures of merit for noise content and frequency distortion (and tonality, for halftones) can be obtained. Other visual quality metrics, such as that described by Lubin [81], combine all image distortions into one figure. However, as was shown in Chapter 3, halftoning by error diffusion is accurately modeled as a process which sharpens an image and adds noise, i.e., a process which introduces a small, predictable set of distortions. Applying to halftones a quality metric that is designed for any type of image therefore seems unnecessarily complicated, and discards information about the process used to generate the image. Furthermore, characterizing halftone sharpening independently is desirable, because of its highly subjective and viewer-dependent effect on the quality of an image.

A similar argument applies to inverse halftones. Obtaining separate figures for blurring and noise content can be accomplished by modeling the inverse halftoning process. Since these two distortions have greatly different effects on the human visual system, it would seem to be sensible to characterize them separately. In Chapter 4, the blurring was characterized by computing an effective transfer function for the inverse halftoning system; ideally, this transfer function would be further reduced to a single number. It may transpire that all inverse halftoning schemes can be adequately characterized by

the bandwidth of the effective transfer function, by its rate of rolloff, or by some other simple measure. However, this can only be determined by examining results from other inverse halftoning schemes, with a larger suite of representative images.

Ultimately, one would probably desire a *single* figure of merit for a forward or inverse halftone, and the frequency distortion and noise figures would therefore have to be combined. To obtain a meaningful quality measure, it will be necessary to determine an appropriate weighting for the two components. This will almost certainly require psychovisual testing under controlled conditions. It would then be possible to quantify the benefit of using a quality metric which separates the effects of frequency distortion and noise, assesses them individually, and combines the results in a weighted fashion to produce a single figure of merit, over using a conventional combined measure such as that presented in [81].

The linear gain model for the quantizer in error diffusion that was presented in Chapter 3 is a simple approximation to a difficult non-linear problem. The accuracy of the model is certainly high enough to obtain a residual image that has a very low correlation with the original image, permitting the use of WSNR to determine the visual quality of halftones. The WSNR results are in accordance with results obtained from visual inspection of halftones. The tonality metric provides useful additional information about halftone quality, and itself relies on the linear gain model to obtain a low-correlation residual. However, most halftones are currently produced by clustered-dot screening, which is widely used in commercial printers, chiefly because of its speed.

As the spatial resolution of printers increases, the relatively poor quality of screened halftones is becoming less of a concern. The primary trade-off in screening is well-known [1], namely, between grayscale resolution and spatial resolution. The designer must therefore merely decide how many shades of gray can be obtained without allowing the screen frequency to drop too low, which would introduce visible artifacts into the halftone. This becomes an increasingly easy task as printer resolution improves. Furthermore, the quality of a screened halftone is probably directly predictable from the screen alone; the visual quality metric presented in this work is almost certainly unnecessary for screened halftones.

The field of inverse halftoning is relatively new, and there will probably be large improvements in the performance of algorithms in the future. Since the primary use of inverse halftoning is in recovering grayscale images from optically scanned halftones, it will be necessary to understand the degradations introduced by the scanning process if high quality inverse halftones are to be obtained. At present, all inverse halftoning schemes, including the one presented in Chapter 4, assume that a perfect copy of the halftone is available. In general, this will not be true. Printer ink spread, document skew, scanner resolution, and many other factors will affect the quality of the halftone that is actually available.

A scanned halftone is likely to have been screened, rather than error diffused, because of the popularity of screening in commercial printers. Fast, high quality algorithms are required for these halftones. Already, at least one technique exists for inverse halftoning *any* type of halftone [82]; however, it

is a wavelet-based technique, and therefore suffers from the implementation difficulties described in Chapter 4 in connection with [26]. A better method would be to first detect the halftoning method that was used, and then apply a fast algorithm designed specifically for that type of halftone. This requires robust detection methods. However, ordered (e.g., clustered-dot screened) and stochastic (e.g., error diffused) halftones have radically different spatial properties, and it should not be difficult to devise a simple algorithm to differentiate between them.

In a rehalftoning application, the inverse halftoning problem is simplified, as was shown in Chapter 5. It is quite difficult to obtain high quality grayscale images from screened halftones. However, if the inverse halftone is subsequently re-screened (or error diffused), the artifacts are largely masked by the halftoning error. Simple inverse halftoning methods for screened halftones should therefore be investigated, with a view towards producing rehalftones of acceptable quality. A comb filter with zeros placed at multiples of the screen frequency is a possible example. This requires knowledge of the screen period, but this can be ascertained from the halftone by correlation methods.

Because grayscale images can be compressed much more efficiently than halftones [59], rehalftoning may become important in applications such as facsimile transmission. Connection time can be reduced if the halftones in a faxed document are converted to grayscale at the transmitter, compressed, and rehalftoned at the receiver. An additional benefit is that the receiver can use a different halftoning method from the one used to create the original image, thus giving the best possible final image. In this case, the intermediate

inverse halftone must be of sufficient quality that a good quality halftone can be derived from it using several different methods; the requirements on the inverse halftone need to be ascertained. This should be performed with the intended compression scheme in mind, so that good compression performance and high quality can be achieved simultaneously.

In summary, separating frequency distortion from noise injection for forward and inverse halftones appears sound, and further work will show whether the concept can be applied to other image processing operations, such as compression. Furthermore, it would appear that the field of inverse halftoning, including rehalftoning, is currently more open than that of forward halftoning. New problems are appearing that must be solved. It is hoped that the ideas presented in this work will be of use to those intending to continue work in the area.

Bibliography

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

[2] C. Dong, T. Pappas, and D. Neuhoff, "Measurement of printer parameters for model-based halftoning," *Proc. SPIE, Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, pp. 355–366, Feb. 1993.

[3] C. Rosenberg, "Measurement-based evaluation of a printer dot model for halftone algorithm tone correction," *J. Electronic Imaging*, vol. 2, pp. 205–212, July 1993.

[4] J. Allebach and B. Liu, "Analysis of halftone dot profile and aliasing in the discrete binary representation of images," *J. Opt. Soc. Am.*, vol. 67, pp. 1147–1154, Sept. 1977.

[5] Y. Kim, G. Arce, and N. Grabowski, "Inverse halftoning using binary permutation filters," *IEEE Trans. Image Processing*, vol. 4, pp. 1296–1311, Sept. 1995.

[6] D. Neuhoff and T. Pappas, "Perceptual coding of images for halftone display," *IEEE Trans. Image Processing*, vol. 3, pp. 341–354, July 1994.

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

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

[9] 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.

[10] 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.

[11] 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.

[12] J. Sullivan, L. Ray, and R. Miller, "Design of minimum visual modulation halftone patterns," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 21, pp. 33–38, Jan. 1991.

[13] M. Schulze and T. Pappas, "Blue noise and model-based halftoning," *Proc. SPIE, Human Vision, Visual Processing, and Digital Display V*, vol. 2179, pp. 182–194, Feb. 1994.

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

[15] 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, 1976.

[16] T. Mitsa and K. Varkur, "Evaluation of contrast sensitivity functions for the formulation of quality measures incorporated in halftoning algorithms," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 5, pp. 301–304, Apr. 1993.

[17] H. Spang and P. Schultheiss, "Reduction of quantizing noise by use of feedback," *IRE Trans. Communications Systems*, pp. 373–380, Dec. 1962.

[18] S. Norsworthy, R. Schreier, and G. Temes, eds., *Delta-Sigma Data Converters*. New York, NY: IEEE Press, 1997.

[19] K. Pohlmann, *Principles of Digital Audio*. New York, NY: McGraw-Hill, 3rd ed., 1995.

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

[21] 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.

[22] L. Roberts, "Picture coding using pseudo-random noise," *IRE Trans. Information Theory*, pp. 145–154, Feb. 1962.

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

[24] N. Jayant, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[25] P. Wong, "Inverse halftoning and kernel estimation for error diffusion," *IEEE Trans. Image Processing*, vol. 4, pp. 486–498, Apr. 1995.

[26] Z. Xiong, M. Orchard, and K. Ramchandran, "Inverse halftoning using wavelets," *Proc. IEEE Conf. Image Processing*, pp. 569–572, Sept. 1996.

[27] S. Schweizer and R. Stevenson, "A Bayesian approach to inverse halftoning," *Proc. SPIE, Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, pp. 282–292, Feb. 1993.

[28] B. Widrow and S. Stearns, *Adaptive Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1985.

[29] C. Shannon and W. Weaver, *The Mathematical Theory of Communication.* Urbana, IL: University of Illinois Press, 1949.

[30] T. Mitsa, "Image quality metrics for halftone images," *Proc. SPIE, Imaging Technologies and Applications*, vol. 1778, pp. 196–207, Mar. 1992.

[31] T. Cornsweet, *Visual Perception.* New York, NY: Academic Press, 1970.

[32] W. Geisler, *Class Notes for Psychology 380E: Vision Systems.* Austin, TX: The University of Texas at Austin, 1996.

[33] B. Wandell, *Foundations of Vision.* Sunderland, MA: Sinauer Associates, 1995.

[34] 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.

[35] P. Barten, "Evaluation of subjective image quality with the square-root integral method," *J. Opt. Soc. Am. A*, vol. 7, pp. 2024–2031, Oct. 1990.

[36] J. Sullivan, R. Miller, and G. Pios, "Image halftoning using a visual model in error diffusion," *J. Opt. Soc. Am. A*, vol. 10, pp. 1714–1724, Aug. 1993.

[37] A. Netravali and B. Haskell, *Digital Pictures: Representation, Compression, and Standards.* New York, NY: Plenum Press, 2nd ed., 1995.

[38] L. Beranek, *Acoustics.* New York, NY: American Institute of Physics, 1986.

[39] R. Williams, *Electrical Engineering Probability.* St. Paul, MN: West, 1991.

[40] P. Stucki, "MECCA—a multiple-error correcting computation algorithm for bilevel hardcopy reproduction," Research Report RZ1060, IBM Research Laboratory, Zurich, Switzerland, 1981.

[41] K. Knox, "Threshold modulation in error diffusion on non-standard rasters," *Proc. SPIE, Human Vision, Visual Processing, and Digital Display V*, vol. 2179, pp. 159–169, Feb. 1994.

[42] I. Witten and R. Neal, "Using Peano curves for bilevel display of continuous-tone images," *Proc. IEEE Computer Graphics and Applications*, pp. 47–51, May 1982.

[43] T. Agui, T. Nagae, and M. Nakajima, "Digital halftoning using a generalized peano scan," *Proc. SPIE, Visual Communications and Image Processing*, vol. 1606, pp. 912–916, Nov. 1991.

[44] 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.

[45] R. Eschbach, "Reduction of artifacts in error diffusion by means of input-dependent weights," *J. Electronic Imaging*, vol. 2, pp. 352–358, Oct. 1993.

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

[47] Z. Fan, "Error diffusion with a more symmetric error distribution," *Proc. SPIE, Human Vision, Visual Processing, and Digital Display V*, vol. 2179, pp. 150–158, Feb. 1994.

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

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

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

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

[52] 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.

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

[54] A. Oppenheim and R. Schafer, *Discrete-Time Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

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

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

[57] R. Gray, W. Chou, and P. Wong, "Quantization noise in single-loop sigma-delta modulation with sinusoidal inputs," *IEEE Trans. Communications*, vol. 37, pp. 956–967, Sept. 1989.

[58] P. Horowitz and W. Hill, *The Art of Electronics.* Cambridge, England: Cambridge University Press, 1980.

[59] D. Neuhoff and T. Pappas, "Perceptual coding of images for halftone display," *IEEE Trans. Image Processing*, vol. 3, pp. 1–13, Jan. 1994.

[60] M. Ting and E. Riskin, "Error-diffused image compression using a binary-to-grayscale decoder and predictive pruned tree-structured vector quantization," *IEEE Trans. Image Processing*, vol. 3, pp. 854–858, Nov. 1994.

[61] S. Hein and A. Zakhor, "Halftone to continuous-tone conversion of error-diffusion coded images," *IEEE Trans. Image Processing*, vol. 4, pp. 208–216, Feb. 1995.

[62] M. Analoui and J. Allebach, "New results on reconstruction of continuous-tone from halftone," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 3, pp. 313–316, Mar. 1992.

[63] T. Kite, N. Damera-Venkata, B. Evans, and A. Bovik, "A high quality, fast inverse halftoning algorithm for error diffused halftones," *Proc. IEEE Conf. Image Processing*, Oct. 1998. To appear.

[64] Z. Fan, "Retrieval of gray images from digital halftones," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 2477–2480, May 1992.

[65] R. Kern, T. Stockham, and D. Strong, "Descreening via linear filtering and iterative techniques," *Proc. SPIE, Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, pp. 299–309, Feb. 1993.

[66] R. Stevenson, "Inverse halftoning via MAP estimation," *IEEE Trans. Image Processing*, vol. 6, pp. 574–583, Apr. 1997.

[67] D. Geman and S. Geman, "Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, Nov. 1984.

[68] S. Hein and A. Zakhor, "Reconstruction of continuous tone images from their error-diffused halftone version," *Proc. SPIE, Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, pp. 310–324, Feb. 1993.

[69] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, pp. 710–732, July 1992.

[70] S. Hein and A. Zakhor, "Optimal decoding for data acquisition applications of sigma delta modulators," *IEEE Trans. Signal Processing*, vol. 41, pp. 602–616, Feb. 1993.

[71] N. Thao and M. Vetterli, "Optimal MSE reconstruction in oversampled A/D conversion using convexity," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 4, pp. 165–168, Mar. 1992.

[72] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, pp. 629–639, July 1990.

[73] T. Huang, J. Burnett, and A. Deczky, "The importance of phase in image processing filters," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 23, pp. 529–542, Dec. 1975.

[74] P. Gill, W. Murray, and M. Wright, *Practical Optimization.* New York, NY: Academic Press, 1981.

[75] F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll, "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM J. Numerical Analysis*, vol. 29, pp. 182–193, Feb. 1992.

[76] D. Marr and E. Hildreth, "The theory of edge detection," *Proc. Royal Society of London. Series B: Biological Sciences*, vol. 207, pp. 187–217, 1980.

[77] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, Nov.

1986.

[78] R.Eschbach, "Pixel quantization with adaptive error diffusion." United States Patent, May 1993. Patent Number: 5,208,871.

[79] R. Gonzalez and R. Woods, *Digital Image Processing.* Reading, MA: Addison-Wesley, 1993.

[80] E. Peli, "Contrast in complex images," *J. Opt. Soc. Am. A*, vol. 7, pp. 2032–39, Oct. 1990.

[81] J. Lubin, "A visual discrimination model for imaging system design and evaluation," in *Vision Models for Target Detection and Recognition* (E. Peli, ed.), pp. 245–283, Singapore: World Scientific, 1995.

[82] J. Luo, R. de Queiroz, and Z. Fan, "A robust technique for image descreening based on the wavelet transform," *IEEE Trans. Signal Processing*, vol. 46, pp. 1179–1194, Apr. 1998.

Vita

Thomas David Kite was born in Chester, England on July 27, 1970, to Janet and David Kite. He was described as a 'thoroughly naughty boy' and 'indefatigable' at various times during his school career, both of which were only partially true. He obtained a degree in physics from Oxford University in June, 1991, and a Master's degree in electrical engineering (specializing in acoustics) from The University of Texas at Austin in December, 1993. During his time at the University, he was a teaching assistant for classes in logic design, senior design lab, digital signal processing, and digital image processing. He received an engineering Ramshorn award in 1996, ostensibly for outstanding work both teaching and in the lab, although he suspects that it was actually in recognition of countless hours of service washing coffee cups and keeping the fridge defrosted. He would not be without his cat, Pete, who deserves a Ramshorn himself for Tireless Devotion to Purring.

Permanent address: Smithy Cottage, Grange Lane, Whitegate,
Cheshire, England, CW8 2BQ.

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.