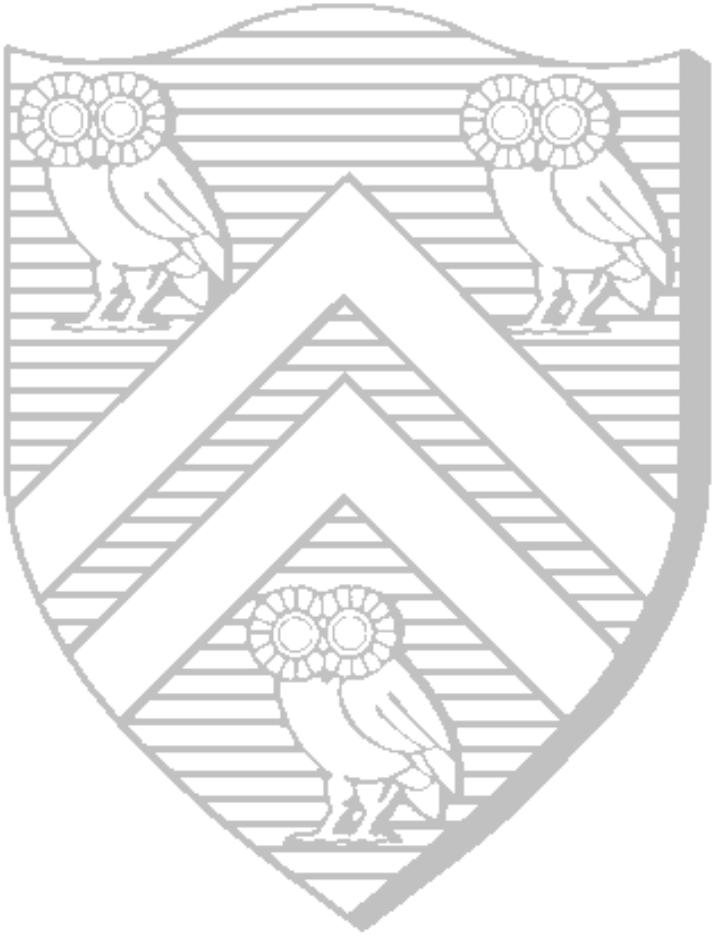
INVERSE PROBLEMS IN IMAGE PROCESSING

Ramesh Neelamani



Thesis: Doctor of Philosophy
Electrical and Computer Engineering
Rice University, Houston, Texas (June 2003)

RICE UNIVERSITY

Inverse Problems in Image Processing

by

Ramesh Neelamani

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

APPROVED, THESIS COMMITTEE:

Richard Baraniuk, Professor, Chair
Electrical and Computer Engineering

Robert Nowak, Associate Professor
Electrical and Computer Engineering

Michael Orchard, Professor
Electrical and Computer Engineering

Steven J. Cox, Professor
Computational and Applied Mathematics

HOUSTON, TEXAS

JUNE 2003

Inverse Problems in Image Processing

Ramesh Neelamani

Abstract

Inverse problems involve estimating parameters or data from inadequate observations; the observations are often noisy and contain incomplete information about the target parameter or data due to physical limitations of the measurement devices. Consequently, solutions to inverse problems are non-unique. To pin down a solution, we must exploit the underlying structure of the desired solution set. In this thesis, we formulate novel solutions to three image processing inverse problems: deconvolution, inverse halftoning, and JPEG compression history estimation for color images.

Deconvolution aims to extract crisp images from blurry observations. We propose an efficient, hybrid Fourier-Wavelet Regularized Deconvolution (ForWaRD) algorithm that comprises blurring operator inversion followed by noise attenuation via scalar shrinkage in both the Fourier and wavelet domains. The Fourier shrinkage exploits the structure of the colored noise inherent in deconvolution, while the wavelet shrinkage exploits the piecewise smooth structure of real-world signals and images. ForWaRD yields state-of-the-art mean-squared-error (MSE) performance in practice. Further, for certain problems, ForWaRD guarantees an optimal rate of MSE decay with increasing resolution.

Halftoning is a technique used to render gray-scale images using only black or white dots. Inverse halftoning aims to recover the shades of gray from the binary image and is vital to process

scanned images. Using a linear approximation model for halftoning, we propose the Wavelet-based Inverse Halftoning via Deconvolution (WInHD) algorithm. WInHD exploits the piece-wise smooth structure of real-world images via wavelets to achieve good inverse halftoning performance. Further, WInHD also guarantees a fast rate of MSE decay with increasing resolution.

We routinely encounter digital color images that were previously JPEG-compressed. We aim to retrieve the various settings—termed JPEG compression history—employed during previous JPEG operations. This information is often discarded en-route to the image's current representation. We discover that the previous JPEG compression's quantization step introduces lattice structures into the image. Our study leads to a fundamentally new result in lattice theory—nearly orthogonal sets of lattice basis vectors contain the lattice's shortest non-zero vector. We exploit this insight along with other known, novel lattice-based algorithms to effectively uncover the image's compression history. The estimated compression history significantly improves JPEG recompression.

Acknowledgments

I am indebted to my energetic advisor and mentor Prof. Richard Baraniuk for making my Rice experience truly enjoyable and enlightening. I am also grateful to my other thesis committee members Prof. Robert Nowak, Prof. Michael Orchard, and Prof. Steven Cox for stimulating discussions and valuable comments.

All my research collaborators, including Richard Baraniuk, Kathrin Berkner, Hyeokho Choi, Sanjeeb Dash, Zhigang Fan, Robert Nowak, Ricardo de Queiroz, Rudolf Riedi, and Justin Romberg, have played a critical role in broadening my research horizons. I truly appreciate the enthusiasm and the seemingly-infinite patience they displayed while interacting with me.

Thanks to all my friends and the Rice ECE group of professors and students for keeping me bright and sunny. I now have a home away from home.

I lack words to express my gratitude to amma, appa, and Anu for their unconditional love and support.

Sandhya has made this journey worthwhile.

Contents

Abstract ii

Acknowledgments iv

1 Introduction 1

1.1 ForWaRD: Fourier-Wavelet Regularized Deconvolution 2

1.2 WInHD: Wavelet-based Inverse Halftoning via Deconvolution 3

1.3 JPEG Compression History Estimation (CHEst) for Color Images 4

2 ForWaRD: Fourier-Wavelet Regularized Deconvolution 6

2.1 Introduction 6

2.1.1 Problem statement 6

2.1.2 Transform-domain shrinkage 8

2.1.3 Fourier-Wavelet Regularized Deconvolution (ForWaRD) 11

2.1.4 Related work 14

2.1.5 Chapter organization 15

2.2 Sampling and Deconvolution 15

2.3 Fourier-based Regularized Deconvolution (FoRD) 18

2.3.1 Framework 18

2.3.2 Strengths of FoRD 20

2.3.3 Limitations of FoRD 20

2.4 Wavelet-Vaguelette Deconvolution (WVD) 21

2.4.1 Framework 21

2.4.2 Strengths of WVD 21

2.4.3 Limitations of WVD 22

2.5 Fourier-Wavelet Regularized Deconvolution (ForWaRD) 22

2.5.1 ForWaRD algorithm 23

2.5.2 How ForWaRD works 23

2.5.3 Balancing Fourier and wavelet shrinkage in ForWaRD 24

2.5.4 Asymptotic ForWaRD performance and optimality 29

2.6 ForWaRD Implementation 35

2.6.1 Estimation of σ^2 35

2.6.2 Choice of Fourier shrinkage 35

2.6.3 Choice of wavelet basis and shrinkage 36

2.7 Results 37

2.7.1 Simulated problem 37

2.7.2 Real-life application: Magnetic Force Microscopy 38

3 WInHD: Wavelet-based Inverse Halftoning via Deconvolution 41

3.1 Introduction 41

3.2 Linear Model for Error Diffusion 46

3.3 Inverse Halftoning $\approx$ Deconvolution 48

3.3.1 Deconvolution 49

3.3.2 Inverse halftoning via Gaussian low-pass filtering (GLPF) 52

3.4 Wavelet-based Inverse Halftoning Via Deconvolution (WInHD) 53

3.4.1 WInHD algorithm 53

3.4.2 Asymptotic performance of WInHD 55

3.5 Results 57

4 JPEG Compression History Estimation (CHEst) for Color Images 60

4.1 Introduction 60

4.2 Color Spaces and Transforms 63

4.3 JPEG Overview 65

4.4 CHEst for Gray-Scale Images 66

4.4.1 Statistical framework 67

4.4.2 Algorithm steps 69

4.5 Dictionary-based CHEst for Color Images 70

4.5.1 Statistical framework 70

4.5.2 Algorithm steps 71

4.5.3 Dictionary-based CHEst results 72

4.6 Blind Lattice-based CHEst for Color Images 74

4.6.1 Ideal lattice structure of DCT coefficients 76

4.6.2 Lattice algorithms 78

4.6.3 LLL provides parallelepiped lattice's basis vectors 83

4.6.4 Estimating scaled linear component of the color transform 84

4.6.5 Estimating the complete color transform and quantization step-sizes 86

4.6.6 Round-offs perturb ideal geometry 87
4.6.7 Combating round-off noise 88
4.6.8 Algorithm steps 90
4.6.9 Lattice-based CHEst results 92
4.7 JPEG Recompression: A Sample Application of CHEst 95
4.7.1 JPEG recompression using dictionary-based CHEst 95
4.7.2 JPEG recompression using lattice-based CHEst 97

5 Conclusions 99

A Background on Wavelets 104

A.1 1-D and 2-D Wavelet Transforms 104
A.2 Economy of Wavelet Representations 106
A.3 Wavelet Shrinkage-based Signal Estimation 107

B Formal WVD Algorithm 109

C Derivation of Optimal Regularization Parameters for ForWaRD 111

D Decay Rate of Wavelet Shrinkage Error in ForWaRD 113

D.1 Besov Smoothness of Distorted Signal 113
D.2 Wavelet-domain Estimation Error: ForWaRD vs. Signal in White Noise 114

E Decay Rate of Total ForWaRD MSE 116

E.1 Bounding Wavelet Shrinkage Error 116

E.2 Bounding Fourier Distortion Error 116

F Decay Rate of WInHD's MSE 119

G Properties of Nearly Orthogonal Basis Vectors 120

G.1 Proof of Proposition 5 120

G.1.1 Proof for 2-D lattices 120

G.1.2 Proof for higher dimensional lattices 121

G.2 Proof of Proposition 6 122

G.2.1 Proof for 2-D lattices 122

G.2.2 Proof for higher dimensional lattices 124

Bibliography 128

Chapter 1

Introduction

There was neither non-existence nor existence then;

there was neither the realm of space nor the sky which is beyond.

–From the Rig Veda (translated by Prof. W. D. O'Flaherty)

Inverse problems estimate some parameter or data from a set of indirect noisy observations. Due to lack of sufficient information in the indirect observations, solutions to inverse problems are typically non-unique and, therefore, challenging. To tackle the inherent ambiguity of inverse problems solutions, we need to incorporate into our estimation a priori information about the structure of desired solution set.

In this thesis, we formulate solutions to deconvolution, inverse halftoning, and JPEG Compression History Estimation (CHEst) for color images. We constrain our deconvolution and inverse halftoning solutions by using wavelet representations to exploit the piece-wise smooth structure of typical real-world signals and images. JPEG's quantization operation creates lattice structures in the discrete cosine transform (DCT) coefficients. We exploit these lattice structures using novel algorithms from cryptography to perform color image JPEG CHEst.

1.1 ForWaRD: Fourier-Wavelet Regularized Deconvolution

Given an observation that is comprised of an input image first degraded by linear time-invariant (LTI) convolution with a known impulse response and then corrupted by additive noise, deconvolution aims to estimate the input image. Deconvolution is extremely important in applications such as satellite imaging and seismic imaging.

To solve the deconvolution problem, in Chapter 2, we propose a fast hybrid algorithm called Fourier-wavelet regularized deconvolution (ForWaRD) that comprises of convolution operator inversion followed by scalar shrinkage in both the Fourier domain and the wavelet domain. The Fourier shrinkage exploits the Fourier transform's economical representation of the colored noise inherent in deconvolution, while the wavelet shrinkage exploits the wavelet domain's economical representation of piecewise smooth signals and images. We derive the optimal balance between the amount of Fourier and wavelet regularization by optimizing an approximate mean-squared-error (MSE) metric and find that signals with more economical wavelet representations require less Fourier shrinkage. ForWaRD is applicable to all ill-conditioned deconvolution problems, unlike the purely wavelet-based Wavelet-Vaguelette Deconvolution (WVD); moreover, its estimate features minimal ringing, unlike the purely Fourier-based Wiener deconvolution. Even in problems for which the WVD was designed, we prove that ForWaRD's MSE decays with the optimal WVD rate as the number of samples increases. Further, we demonstrate that over a wide range of practical sample-lengths, ForWaRD improves upon WVD's performance.

1.2 WInHD: Wavelet-based Inverse Halftoning via Deconvolution

Digital halftoning is a common technique used to render a sampled gray-scale image using only black or white dots [1] (see Figures 3.3(a) and (b)); the rendered bi-level image is referred to as a halftone. Inverse halftoning is the process of retrieving a gray-scale image from a given halftone. Applications of inverse halftoning include enhancement and compression of facsimile images [2].

We propose a novel algorithm called *Wavelet-based inverse halftoning via deconvolution* (WInHD) to perform inverse halftoning of error-diffused halftones in Chapter 3. We realize that inverse halftoning can be formulated as a deconvolution problem by using the linear approximation model for error diffusion halftoning of Kite et al. In the linear approximation model, the error-diffused halftone is modeled as the gray-scale input blurred by a convolution operator and corrupted with additive colored noise; the convolution operator and noise coloring are determined by the error diffusion technique. WInHD adopts a wavelet-based deconvolution approach to perform inverse halftoning; it by first inverts the model-specified convolution operator and then attenuates the residual noise using scalar wavelet-domain operations. Since WInHD is model-based, it is easily adapted to different error diffusion halftoning techniques. Unlike previous inverse halftoning algorithms, under mild assumptions, we derive for images in a Besov space the minimum rate at which the WInHD estimate's MSE decays as the spatial resolution of the gray-scale image increases (i.e., number of samples $\rightarrow \infty$). We also prove that WInHD's MSE decay rate is optimal if the gray-scale image before halftoning is noisy. Using simulations, we verify that WInHD is competitive with state-of-the-art inverse halftoning techniques in the mean square error (MSE) sense and it also provides good visual performance.

1.3 JPEG Compression History Estimation (CHEst) for Color Images

We routinely encounter digital color images in bitmap (BMP) formats or in lossless compression formats such as Tagged Image File Format (TIFF). Such images are often subjected to JPEG (Joint Photographic Experts Group) compression and decompression operations before reaching their current representation. The settings used during the previous JPEG compression and decompression; such as the choice of color transformation, subsampling, and the quantization table; is not standardized [3]. We refer to such previous JPEG compression settings as the image's JPEG *compression history*. The compression history is lost during operations such as conversion from JPEG format to BMP or TIFF format. We aim to estimate this lost information from the image's current representation. The estimated JPEG compression history can be used for JPEG recompression, for covert message passing, or to uncover the compression settings used inside digital cameras.

In Chapter 4, we propose a new framework to estimate the compression history of color images. Due to quantization and dequantization operations during JPEG compression and decompression, the discrete cosine transform (DCT) coefficient histograms of previously JPEG-compressed images exhibit near-periodic structure with the period determined by the quantization step-size. For the general case, we derive a maximum likelihood approach that exploits the near-periodic DCT coefficient structure to select the compression color space from a dictionary of color spaces and to estimate the quantization table. If the transform from the color space used for JPEG compression to the current color space is affine and if no subsampling is employed during JPEG compression, then we no longer need a dictionary of color spaces to estimate the compression history. In this special case, we demonstrate that the DCT coefficients of the observed image nearly conform to a 3-dimensional parallelepiped lattice structure determined by the affine color transform. During

our study of lattices, we discover a new, fundamental property. We realize that a set of nearly orthogonal lattice basis vectors always contains the shortest non-zero vector in the lattice. Further, we also identify the conditions under which the set of nearly orthogonal basis vectors for a lattice is unique. Using our new insights and with novel applications of existing lattice algorithms, we exploit the lattice structure offered by our problem and estimate a color image's JPEG compression history, namely, the affine color transform and the quantization tables. We demonstrate the efficacy of our proposed algorithm via simulations. Further, we verify the utility of the estimated compression history in JPEG recompression; we demonstrate that the estimated information allows us to recompress an image with minimal distortion (large signal-to-noise-ratio) and simultaneously achieve a small file-size.

Chapter 2

ForWaRD: Fourier-Wavelet Regularized Deconvolution

You can't depend on your eyes when your imagination is out of focus.

–Mark Twain

2.1 Introduction

Deconvolution is a recurring theme in a wide variety of signal and image processing problems. For example, practical satellite images are often blurred due to limitations such as aperture effects of the camera, camera motion, or atmospheric turbulence [4]. Deconvolution becomes necessary when we wish a crisp deblurred image for viewing or further processing.

2.1.1 Problem statement

In this chapter, we treat the classical discrete-time deconvolution problem. The problem setup and solutions are described in 1-dimension (1-D), but everything extends directly to higher dimensions as well. The observed samples $y(n)$ consist of unknown desired signal samples $x(n)$ first degraded by circular convolution (denoted by $\circledast$) with a known impulse response $h(n)$ from a linear time-invariant (LTI) system $\mathcal{H}$ and then corrupted by zero-mean additive white Gaussian noise (AWGN)

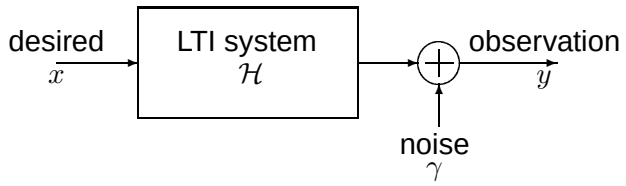


Figure 2.1: *Convolution model setup. The observation y consists of the desired signal x first degraded by the linear time-invariant (LTI) convolution system $\mathcal{H}$ and then corrupted by zero-mean additive white Gaussian noise (AWGN) γ.*

$\gamma(n)$ with variance σ^2 (see Fig. 2.1)

$$\begin{aligned} y(n) &:= \mathcal{H}x(n) + \gamma(n), \qquad n = 0, \ldots, N-1 \\ &:= (h \circledast x)(n) + \gamma(n). \end{aligned} \tag{2.1}$$

Given y and h, we seek to estimate x.

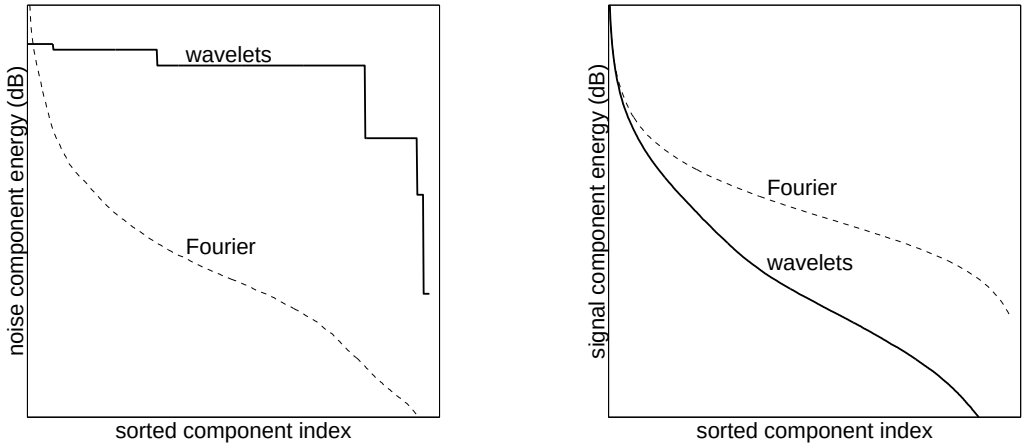
A naive deconvolution estimate $\widetilde{x}$ is obtained using the operator inverse $\mathcal{H}^{-1}$ as[1]

$$\widetilde{x}(n) := \mathcal{H}^{-1}y(n) = x(n) + \mathcal{H}^{-1}\gamma(n). \tag{2.2}$$

Unfortunately, the variance of the colored noise $\mathcal{H}^{-1}\gamma$ in $\widetilde{x}$ is large when $\mathcal{H}$ is ill-conditioned. In such a case, the mean-squared error (MSE) between x and $\widetilde{x}$ is large, making $\widetilde{x}$ an unsatisfactory deconvolution estimate.

In general, deconvolution algorithms can be interpreted as estimating x from the noisy signal $\widetilde{x}$ in (2.2). In this chapter, we focus on simple and fast estimation based on *scalar shrinkage* of

[1]For non-invertible $\mathcal{H}$, we replace $\mathcal{H}^{-1}$ by its pseudo-inverse and x by its orthogonal projection onto the range of $\mathcal{H}$ in (2.2) [5]. The estimate $\widetilde{x}$ in (2.2) continues to retain all the information that y contains about x.



(a) colored noise $\langle \mathcal{H}^{-1}\gamma, b_k \rangle$ (b) signal $\langle x, b_k \rangle$

Figure 2.2: *Economy of Fourier vs. wavelet representations. (a) Energies in dB of the Fourier and wavelet components of the noise $\mathcal{H}^{-1}\gamma$ colored by the pseudo-inverse of a 2-D 9×9 box-car smoothing operator. The components are sorted in descending order of energy from left to right. The colored noise energy is concentrated in fewer Fourier components than wavelet components. (b) Energies of the Fourier and wavelet components of the* Cameraman *image x. The signal energy is concentrated in fewer wavelet components than Fourier components.*

individual components in a suitable transform domain. Such a focus is not restrictive because transform-domain scalar shrinkage lies at the core of many traditional [6, 7] and modern [5, 8] deconvolution approaches.

2.1.2 Transform-domain shrinkage

Given an orthonormal basis $\{b_k\}_{k=0}^{N-1}$ for $\mathbb{R}^N$, the naive estimate $\widetilde{x}$ from (2.2) can be expressed as

$$\widetilde{x} = \sum_{k=0}^{N-1} \left(\langle x, b_k \rangle + \langle \mathcal{H}^{-1}\gamma, b_k \rangle \right) b_k. \tag{2.3}$$

An improved estimate $\widetilde{x}_\lambda$ can be obtained by simply shrinking the k-th component in (2.3) with a scalar λ_k, $0 \leq \lambda_k \leq 1$ [9]:

$$\widetilde{x}_\lambda := \sum_{k=0}^{N-1} \left(\langle x, b_k \rangle + \langle \mathcal{H}^{-1}\gamma, b_k \rangle \right) \lambda_k \, b_k \tag{2.4}$$

$$=: x_\lambda + \mathcal{H}^{-1}\gamma_\lambda. \tag{2.5}$$

The $x_\lambda := \sum_k \langle x, b_k \rangle \lambda_k \, b_k$ denotes the *retained part* of the signal x that the shrinkage preserves from (2.2), while $\mathcal{H}^{-1}\gamma_\lambda := \sum_k \langle \mathcal{H}^{-1}\gamma, b_k \rangle \lambda_k \, b_k$ denotes the *leaked part* of the colored noise $\mathcal{H}^{-1}\gamma$ that the shrinkage fails to attenuate. Clearly, we should set $\lambda_k \approx 1$ if the variance $\sigma_k^2 := \mathbb{E}(|\langle \mathcal{H}^{-1}\gamma, b_k \rangle|^2)$ of the k-th colored noise component is small relative to the energy $|\langle x, b_k \rangle|^2$ of the corresponding signal component and set $\lambda_k \approx 0$ otherwise. The shrinkage by λ_k can also be interpreted as a form of *regularization* for the deconvolution inverse problem [7].

The tradeoff associated with the choice of λ_k is easily understood: If $\lambda_k \approx 1$, then most of the k-th colored noise component leaks into $\widetilde{x}_\lambda$ with the corresponding signal component; the result is a distortion-free but noisy estimate. In contrast, if $\lambda_k \approx 0$, then most of the k-th signal component is lost with the corresponding colored noise component; the result is a noise-free but distorted estimate. Since the variance of the leaked noise $\mathcal{H}^{-1}\gamma_\lambda$ in (2.5) and the energy of the lost signal $x - x_\lambda$ comprise the MSE of the shrunk estimate $\widetilde{x}_\lambda$, judicious choices of the λ_k's help lower the estimate's MSE.

However, an important fact is that for a given transform domain, even with the best possible

λ_k's, the estimate $\widetilde{x}_\lambda$'s MSE is lower bounded by [8, 10, 11]

$$\frac{1}{2}\sum_{k=0}^{N-1}\min\left(|\langle x, b_k\rangle|^2, \sigma_k^2\right). \tag{2.6}$$

From (2.6), $\widetilde{x}_\lambda$ has small MSE only when most of the signal energy ($=\sum_k |\langle x, b_k\rangle|^2$) and colored noise energy ($=\sum_k \sigma_k^2$) is captured by just a few transform-domain coefficients—we term such a representation *economical*—and when the energy-capturing coefficients for the signal and noise are different. Otherwise, the $\widetilde{x}_\lambda$ is either excessively noisy due to leaked noise components or distorted due to lost signal components.

Traditionally, the Fourier domain (with sinusoidal b_k) is used to estimate x from $\widetilde{x}$. For example, the LTI Wiener deconvolution filter corresponds to (2.4) with each λ_k determined by the k-th component signal-to-noise ratio [6, 7]. The strength of the Fourier basis is that it most economically represents the colored noise $\mathcal{H}^{-1}\gamma$ (see Figure 2.2(a) and Section 2.3.2 for the details). However, the weakness of the Fourier domain is that it does not economically represent signals x with singularities such as images with edges (see Figure 2.2(b)). Consequently, as dictated by the MSE bound in (2.6), any estimate obtained via Fourier shrinkage is unsatisfactory with a large MSE; the estimate is either noisy or distorted for signals x with singularities (see Figure 2.4(c), for example).

Recently, the wavelet domain (with b_k shifts and dilates of a mother wavelet function) has been exploited to estimate x from $\widetilde{x}$; for example, Donoho's *Wavelet-Vaguelette Deconvolution* (WVD) [8]. The strength of the wavelet domain is that it economically represents classes of signals containing singularities that satisfy a wide variety of local smoothness constraints, including piecewise smoothness and Besov space smoothness (see Figure 2.2(b) and Section 2.4.2 for the details).

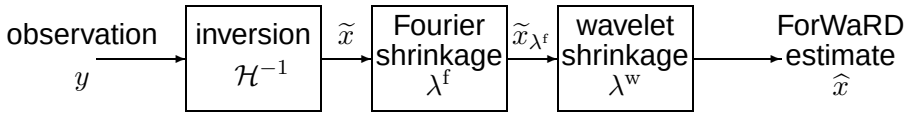


Figure 2.3: *Fourier-Wavelet Regularized Deconvolution (ForWaRD). ForWaRD employs a small amount of Fourier shrinkage (most $\lambda^{\mathrm{f}}_k \approx 1$) to partially attenuate the noise amplified during operator inversion. Subsequent wavelet shrinkage (determined by λ^{w}) effectively attenuates the residual noise.*

However, the weakness of the wavelet domain is that it typically does not economically represent the colored noise $\mathcal{H}^{-1}\gamma$ (see Figure 2.2(a)). Consequently, as dictated by the MSE bound in (2.6), any estimate obtained via wavelet shrinkage is unsatisfactory with a large MSE; the estimate is either noisy or distorted for many types of $\mathcal{H}$.

Unfortunately, no single transform domain can economically represent both the noise colored by a general $\mathcal{H}^{-1}$ and signals from a general smoothness class [8]. Hence, deconvolution techniques employing shrinkage in a single transform domain cannot yield adequate estimates in many deconvolution problems of interest.

2.1.3 Fourier-Wavelet Regularized Deconvolution (ForWaRD)

In this chapter, we propose a deconvolution scheme that relies on *tandem* scalar processing in both the Fourier domain, which economically represents the colored noise $\mathcal{H}^{-1}\gamma$, and the wavelet domain, which economically represents signals x from a wide variety of smoothness classes. Our hybrid *Fourier-Wavelet Regularized Deconvolution* (ForWaRD) technique estimates x from $\widetilde{x}$ by first employing a small amount of scalar Fourier shrinkage λ^{f} and then attenuating the leaked noise with scalar wavelet shrinkage λ^{w} (see Fig. 2.3) [12, 13].

Here is how it works: During operator inversion, some Fourier coefficients of the noise γ are

significantly amplified; just a small amount of Fourier shrinkage (most $\lambda_k^{\mathrm{f}} \approx 1$) is sufficient to attenuate these amplified Fourier noise coefficients with minimal loss of signal components. The leaked noise $\mathcal{H}^{-1}\gamma_{\lambda^{\mathrm{f}}}$ that Fourier shrinkage λ^{f} fails to attenuate (see (2.5)) has significantly reduced energy in all wavelet coefficients, but the signal part $x_{\lambda^{\mathrm{f}}}$ that Fourier shrinkage retains continues to be economically represented in the wavelet domain. Hence subsequent wavelet shrinkage effectively extracts the retained signal $x_{\lambda^{\mathrm{f}}}$ from the leaked noise $\mathcal{H}^{-1}\gamma_{\lambda^{\mathrm{f}}}$ and provides a robust estimate.

For an idealized ForWaRD system, we will derive the optimal balance between the amount of Fourier shrinkage and wavelet shrinkage by optimizing over an approximate MSE metric. We will find that signals with more economical wavelet representations require less Fourier shrinkage.

Figure 2.4 illustrates the superior overall visual quality and lower MSE of the ForWaRD estimate as compared to the LTI Wiener filter estimate [6, 7] for the 2-D box-car blur operator, which models rectangular scanning aperture effects [4], with impulse response $h(n_1, n_2) = \frac{1}{81}$ for $0 \leq n_1, n_2 \leq 8$ and 0 otherwise (see Section 2.7 for the details). For this operator, the WVD approach returns a near-zero estimate; scalar wavelet shrinkage cannot salvage the signal components since nearly all wavelet coefficients are corrupted with high-variance noise.

Indeed, even in problems for which the WVD was designed, we will prove that the ForWaRD MSE also decays with the same optimal WVD rate as the number of samples increases. Further, for such problems, we will experimentally demonstrate ForWaRD's superior MSE performance compared to the WVD over a wide range of practical sample sizes (see Fig. 2.6(a)).

(a) Desired x

(b) Observed y

(c) LTI Wiener filter estimate

(d) ForWaRD estimate

Figure 2.4: *(a) Desired* Cameraman *image* x *(*256×256 *samples). (b) Observed image* y*:* x *smoothed by a 2-D 9×9 box-car blur plus white Gaussian noise with variance such that the BSNR = 40 dB. (c) LTI Wiener filter estimate (SNR = 20.8 dB, ISNR = 5.6 dB). (d) ForWaRD estimate (SNR = 22.5 dB, ISNR = 7.3 dB). See Section 2.7 for further details.*

2.1.4 Related work

Kalifa and Mallat have proposed a mirror-wavelet basis approach that is similar to the WVD but employs scalar shrinkage in the mirror-wavelet domain adapted to the colored noise $\mathcal{H}^{-1}\gamma$ instead of shrinkage in the conventional wavelet domain [5]. Although the adapted basis improves upon the WVD performance in some "hyperbolic" deconvolution problems, similarly to the WVD, it provides inadequate estimates for arbitrary convolution operators. For example, for the ubiquitous box-car blur $\mathcal{H}$, again most signal components are lost during scalar shrinkage due to high-variance noise. Figure 2.7(b) illustrates that ForWaRD is competitive with the mirror-wavelet approach even for a hyperbolic deconvolution problem.

Similar to ForWaRD, Nowak and Thul [14] have first employed an under-regularized system inverse and subsequently used wavelet-domain signal estimation. However, they do not address the issue of optimal regularization or asymptotic performance.

Banham and Katsaggelos have applied a multiscale Kalman filter to the deconvolution problem [15]. Their approach employs an under-regularized, constrained-least-squares prefilter to reduce the support of the state vectors in the wavelet domain, thereby improving computational efficiency. The amount of regularization chosen for each wavelet scale is the lower bound that allows for reliable edge classification. While similar in spirit to the multiscale Kalman filter approach, ForWaRD employs simple Wiener or Tikhonov regularization in the Fourier domain to optimize the MSE performance. Also, ForWaRD employs simple scalar shrinkage on the wavelet coefficients in contrast to more complicated prediction on edge and non-edge quad-trees [15]. Consequently, as discussed in Section 2.5.4, ForWaRD demonstrates excellent MSE performance as the number of samples tends to infinity and is in fact asymptotically optimal in certain cases. Further, as demonstrated in

Section 2.7, ForWaRD yields better estimates than the multiscale Kalman filter approach.

There exists a vast literature on iterative deconvolution techniques; see [7, 16–18] and the references therein. In this chapter, we focus exclusively on non-iterative techniques for the sake of implementation speed and simplicity. Nevertheless, many iterative techniques could exploit the ForWaRD estimate as a seed to initialize their iterations; for example, see [19].

2.1.5 Chapter organization

We begin by providing a more precise definition of the convolution setup (2.1) in Section 2.2. We then discuss techniques that employ scalar Fourier shrinkage in Section 2.3. We introduce the WVD technique in Section 2.4. We present the hybrid ForWaRD scheme in Section 2.5 and discuss its practical implementation in Section 2.6. Illustrative examples lie in Section 2.7. A short overview of wavelets in Appendix A, a WVD review in Appendix B, and technical proofs in Appendices C–E complement this chapter.

2.2 Sampling and Deconvolution

Most real-life deconvolution problems originate in continuous time and are then sampled. In this section, we sketch the relationship between such a sampled continuous-time setup and the setup with discrete-time circular convolution considered in this chapter (see (2.1)).

Consider the following sampled continuous-time deconvolution setup: An unknown finite-energy desired signal $x(t)$ is blurred by linear convolution (denoted by $*$) with the known finite-energy impulse response $h(t)$ of an LTI system and then corrupted by an additive Gaussian process $\gamma(t)$ to form the observation $z(t) = (h * x)(t) + \gamma(t)$. For finite-support $x(t)$ and $h(t)$, the finite-

support $(h * x)(t)$ can be obtained using circular convolution with a sufficiently large period. For infinite-support $x(t)$ and $h(t)$, the approximation of $(h * x)(t)$ using circular convolution can be made arbitrarily precise by increasing the period. Hence, we assume that the observation $z(t)$ over a normalized unit interval can be obtained using circular convolution with a unit period, that is, $z(t) := (h \circledast x)(t) + \gamma(t)$ with $t \in [0, 1)$. Deconvolution aims to estimate $x(t)$ from the samples $z(n)$ of the continuous-time observation $z(t)$. For example, $z(n)$ can be obtained by averaging $z(t)$ over uniformly spaced intervals of length $\frac{1}{N}$

$$z(n) := N \int_{\frac{n}{N}}^{\frac{n+1}{N}} z(t)\, dt, \qquad n = 0, \ldots, N-1. \tag{2.7}$$

Other sampling kernels can also be used in (2.7);[2] see [20, 21] for excellent tutorials on sampling. Such a setup encapsulates many real-life deconvolution problems [4].

The observation samples $z(n)$ from (2.7) can be closely approximated by the observation $y(n)$ from setup (2.1) [4], that is,

$$z(n) \approx y(n) = (h \circledast x)(n) + \gamma(n), \quad n = 0, \ldots, N-1, \tag{2.8}$$

if the continuous-time variables $x(t)$, $\gamma(t)$, and $h(t)$ comprising $z(n)$ are judiciously related to the discrete variables $x(n)$, $\gamma(n)$, and $h(n)$ comprising $y(n)$. We choose to define $\gamma(n) := N \int_{\frac{n}{N}}^{\frac{n+1}{N}} \gamma(t)\, dt$. The $\gamma(n)$ so defined can be assumed to be AWGN samples with some non-zero variance σ^2; the large bandwidths of noise processes such as thermal noise justify the whiteness assumption [4]. Let $\mathcal{B}_{\frac{1}{N}}\, x(t)$ and $\mathcal{B}_{\frac{1}{N}}\, h(t)$ denote signals obtained by first making $x(t)$ and $h(t)$

[2]For example, impulse sampling samples at uniformly spaced time instants $t_n = \frac{n}{N}$ to yield $z(n) := z(t_n)$.

periodic and then band-limiting the resulting signals' Fourier series to the frequency $\frac{1}{2N}$ Hz (for anti-aliasing). We define $x(n) := N \int_{\frac{n}{N}}^{\frac{n+1}{N}} \mathcal{B}_{\frac{1}{N}}\, x(t)\, dt$ and define $h(n)$ as uniformly spaced (over $t \in [0, 1)$) impulse samples of $\frac{1}{N}\mathcal{B}_{\frac{1}{N}}\, h(t)$. With these definitions, we can easily show that the error $|z(n) - y(n)| \leq \|x(t) - \mathcal{B}_{\frac{1}{N}}\, x(t)\|_2 \|h(t) - \mathcal{B}_{\frac{1}{N}}\, h(t)\|_2$. For all finite-energy $x(t)$ and $h(t)$, both $\|x(t) - \mathcal{B}_{\frac{1}{N}}\, x(t)\|_2$ and $\|h(t) - \mathcal{B}_{\frac{1}{N}}\, h(t)\|_2$ decay to zero with increasing N because they denote the norm of the aliasing components of $x(t)$ and $h(t)$ respectively. Consequently, $|z(n) - y(n)|$ soon becomes negligible with respect to the noise variance σ^2 and can be ignored. Hence solutions to estimate $x(n)$ from the $y(n)$ in (2.1)—the focus of this chapter—can be directly applied to estimate $x(n)$ from $z(n)$. For a wide range of Besov space signals, the estimate of $x(n)$ can then be interpolated with minimal error to yield a continuous-time estimate of $x(t)$ as sought in (2.7) [22, 23].[3]

In Sections 2.4 and 2.5, we will analyze the MSE decay rate (in terms of N) of the WVD and ForWaRD solutions to the setup (2.1) as the number of samples $N \to \infty$. At each N, we assume that the corresponding $x(n)$ and $h(n)$ in (2.1) originate from an underlying continuous-time $x(t)$ and $h(t)$ as defined above. Further, we assume that the corrupting $\gamma(n)$ in (2.1) are AWGN samples with variance $\sigma^2 > 0$ that is invariant with N.

[3]The Besov space range is dictated by the smoothness of the sampling kernel. Let $x(t) \in$ Besov space $B^s_{p,q}$ (see Section A.2 for notations). Then, if the sampling kernel of (2.7) is employed, then the interpolation error is negligible with respect to the estimation error for the range $s > \frac{1}{p} - \frac{1}{2}$; the range decreases to $s > \frac{1}{p}$ if impulse sampling is employed [22, 23].

2.3 Fourier-based Regularized Deconvolution (FoRD)

2.3.1 Framework

The Fourier domain is the traditional choice for deconvolution [7] because convolution simplifies to scalar Fourier operations. That is, (2.1) can be rewritten as

$$Y(f_k) = H(f_k)\, X(f_k) + \Gamma(f_k) \tag{2.9}$$

with Y, H, X, and Γ the respective length-N discrete Fourier transforms (DFTs) of y, h, x, and γ, and $f_k := \frac{\pi k}{N}$, $k = -\frac{N}{2} + 1, \ldots, \frac{N}{2}$ (assuming N is even) the normalized DFT frequencies. Rewriting the pseudo-inversion operation (see (2.2)) in the Fourier domain

$$\widetilde{X}(f_k) \;:=\; \begin{cases} X(f_k) + \frac{\Gamma(f_k)}{H(f_k)}, & \text{if } |H(f_k)| > 0, \\ 0, & \text{otherwise} \end{cases} \tag{2.10}$$

with $\widetilde{X}$ the DFT of $\widetilde{x}$, clearly demonstrates that noise components where $|H(f_k)| \approx 0$ are particularly amplified during operator inversion.

Deconvolution via Fourier shrinkage, which we call *Fourier-based Regularized Deconvolution* (FoRD), attenuates the amplified noise in $\widetilde{X}$ with shrinkage

$$\lambda_k^{\mathrm{f}} = \frac{|H(f_k)|^2}{|H(f_k)|^2 \;+\; \Lambda(f_k)}. \tag{2.11}$$

The $\Lambda(f_k) \geq 0$, commonly referred to as *regularization terms* [7, 24], control the amount of shrink-

age. The DFT components of the FoRD estimate $\widetilde{x}_{\lambda^{\mathrm{f}}}$ are given by

$$\begin{aligned}\widetilde{X}_{\lambda^{\mathrm{f}}}(f_k) &:= \widetilde{X}(f_k)\,\lambda^{\mathrm{f}}_k,\\ &= X(f_k)\left(\frac{|H(f_k)|^2}{|H(f_k)|^2\,+\,\Lambda(f_k)}\right)+\frac{\Gamma(f_k)}{H(f_k)}\left(\frac{|H(f_k)|^2}{|H(f_k)|^2\,+\,\Lambda(f_k)}\right),\\ &=: X_{\lambda^{\mathrm{f}}}(f_k)+\frac{\Gamma_{\lambda^{\mathrm{f}}}(f_k)}{H(f_k)}. \end{aligned} \tag{2.12}$$

The $X_{\lambda^{\mathrm{f}}}$ and $\frac{\Gamma_{\lambda^{\mathrm{f}}}}{H}$ comprising $\widetilde{X}_{\lambda^{\mathrm{f}}}$ denote the respective DFTs of the retained signal $x_{\lambda^{\mathrm{f}}}$ and leaked noise $\mathcal{H}^{-1}\gamma_{\lambda^{\mathrm{f}}}$ components that comprise the FoRD estimate $\widetilde{x}_{\lambda^{\mathrm{f}}}$ (see (2.5)). Typically, the operator inversion in (2.10) and shrinkage in (2.12) are performed simultaneously to avoid numerical instabilities.

Different FoRD techniques such as LTI Wiener deconvolution [6, 7] and Tikhonov-regularized deconvolution [24] differ in their choice of shrinkage λ^{f} in (2.12). LTI Wiener deconvolution sets

$$\lambda^{\mathrm{f}}_k = \frac{|H(f_k)|^2}{|H(f_k)|^2\,+\,\alpha\,\frac{N\sigma^2}{|X(f_k)|^2}} \tag{2.13}$$

with *regularization parameter* $\alpha = 1$ to shrink more (that is, $\lambda^{\mathrm{f}}_k \approx 0$) at frequencies where the signal power $|X(f_k)|^2$ is small [6, 7]. Tikhonov-regularized deconvolution, which is similar to LTI Wiener deconvolution assuming a flat signal spectrum $|X(f_k)|^2$, sets

$$\lambda^{\mathrm{f}}_k = \frac{|H(f_k)|^2}{|H(f_k)|^2\,+\,\tau} \tag{2.14}$$

with $\tau > 0$ [24]. Later, in Section 2.5, we will put both of these shrinkage techniques to good use.

2.3.2 Strengths of FoRD

The Fourier domain provides the most economical representation of the colored noise $\mathcal{H}^{-1}\gamma$ in (2.2) because the Fourier transform acts as the Karhunen-Loeve transform [25] and decorrelates the noise $\mathcal{H}^{-1}\gamma$. Consequently, among all linear transformations, the Fourier transform captures the maximum colored noise energy using a fixed number of coefficients [26]. This economical noise representation enhances FoRD performance because the total FoRD MSE is lower bounded by $\frac{1}{2N}\sum_k \min\left(|X(f_k)|^2, \frac{N\sigma^2}{|H(f_k)|^2}\right)$ [8].[4] The best possible FoRD MSE $\frac{1}{N}\sum_k \frac{N\sigma^2|X(f_k)|^2}{|H(f_k)|^2|X(f_k)|^2+N\sigma^2}$ is achieved using the LTI Wiener deconvolution shrinkage λ^{f} of (2.13) in (2.12) [10]. When the signal x in (2.2) also enjoys an economical Fourier-domain representation (that is, when x is "smooth" and thus has rapidly decaying Fourier coefficients [10]), FoRD can provide excellent deconvolution estimates. For example, FoRD provides optimal estimates for signals x in L_2-Sobolev smoothness spaces [8].

2.3.3 Limitations of FoRD

Unfortunately, the Fourier domain does not provide economical representations for signals with singularities, such as images with edges, because the energy of the singularities spreads over many Fourier coefficients. Consequently, even with the best scalar Fourier shrinkage, the FoRD MSE is unsatisfactory, as dictated by the lower bound in (2.6). The estimation error becomes apparent in the form of distortions such as ringing around edge singularities (see Fig. 2.4(c)).

[4] The factor N arises because $\sum_k |X(f_k)|^2 = N\sum_k |x(k)|^2$ for any signal x.

2.4 Wavelet-Vaguelette Deconvolution (WVD)

2.4.1 Framework

The wavelet-vaguelette decomposition algorithm leverages wavelets' economical signal representation to solve some special linear inverse problems [8]. With a slight abuse of nomenclature, we will refer to the wavelet-vaguelette decomposition algorithm applied to deconvolution as *Wavelet-Vaguelette Deconvolution* (WVD).

In contrast to FoRD, the WVD algorithm conceptually extracts the signal x from $\widetilde{x}$ in (2.2) with scalar wavelet shrinkage λ^{w} such as hard thresholding [8, 27] to yield an estimate $\widetilde{x}_{\lambda^{\mathrm{w}}}$. For the reader's convenience, we provide a simple review of the WVD algorithm in Appendix B. We also refer the reader to Appendix A for a short overview of wavelets.

2.4.2 Strengths of WVD

The wavelet domain provides economical representations for a wide variety of signals x in (2.2). In fact, among all orthogonal transforms, the wavelet transform can capture the maximum (within a constant factor) signal energy using any fixed number of coefficients for the worst-case Besov space signal [11]. This economical signal representation enhances WVD's performance because the total WVD MSE can be bounded within a constant factor by $\sum_{j,\ell} \min\left(|w_{j,\ell}|^2, \sigma_j^2\right)$, where σ_j^2 is the wavelet-domain colored noise variance. When the colored noise $\mathcal{H}^{-1}\gamma$ in (2.2) also enjoys an economical wavelet-domain representation, the WVD can provide excellent deconvolution estimates. For example, consider a "scale-invariant" operator $\mathcal{H}$ with frequency response $|H(f_k)| \propto (|k|+1)^{-\nu}$, $\nu > 0$. Such a $\mathcal{H}$ yields colored noise $\mathcal{H}^{-1}\gamma$ that is nearly-diagonalized by the wavelet transform [5, 8] and is hence economically represented in the wavelet domain [8].

For such operators, the per-sample MSE of the WVD estimate $\widetilde{x}_{\lambda^{\mathrm{w}}}$ decays rapidly with increasing number of samples N as [5, 8, 28]

$$\frac{1}{N}\mathbb{E}\left(\sum_{n}|x(n)-\widetilde{x}_{\lambda^{\mathrm{w}}}(n)|^2\right)\leq C\,N^{\frac{-2s}{2s+2\nu+1}}, \tag{2.15}$$

with $C>0$ a constant. Further, no estimator can achieve better error decay rates for every $x(t)\in B^s_{p,q}$.

2.4.3 Limitations of WVD

Unfortunately, the WVD is designed to deconvolve only the very limited class of *scale-invariant* operators [8]. For other $\mathcal{H}$, the colored noise $\mathcal{H}^{-1}\gamma$ in (2.2) is not economically represented in the wavelet domain. For example, with the uniform box-car blur $\mathcal{H}$, the components of the colored noise $\mathcal{H}^{-1}\gamma$ corrupting most wavelet coefficients have extremely high variance due to zeros in H. Consequently, even with the best scalar wavelet shrinkage, the WVD MSE is unsatisfactory, as dictated by the lower bound in (2.6). Indeed, wavelet shrinkage will set most of the signal wavelet coefficients to zero when estimating x from $\widetilde{x}$ in (2.2) and yield an unsatisfactory, near-zero estimate.

2.5 Fourier-Wavelet Regularized Deconvolution (ForWaRD)

The hybrid ForWaRD algorithm estimates x from $\widetilde{x}$ in (2.2) by employing scalar shrinkage both in the Fourier domain to exploit its economical colored noise representation and in the wavelet domain to exploit its economical signal representation. The hybrid approach is motivated by our realization that shrinkage in a *single* transform domain cannot yield good estimates in many decon-

volution problems. This is because no single transform domain can economically represent both the colored noise $\mathcal{H}^{-1}\gamma$ with arbitrary $\mathcal{H}$ and signals x with arbitrary smoothness [8]. By adopting a hybrid approach, ForWaRD overcomes this limitation and provides robust solutions to a wide class of deconvolution problems.

2.5.1 ForWaRD algorithm

The ForWaRD algorithm consists of the following steps (see Figure 2.3):

1a) *Operator inversion*

Obtain Y and H by computing the DFTs of y and h. Then invert $\mathcal{H}$ to obtain $\widetilde{X}$ as in (2.10).

1b) *Fourier shrinkage*

Shrink $\widetilde{X}$ with scalars λ^{f} (using (2.13) or (2.14)) to obtain $\widetilde{X}_{\lambda^{\mathrm{f}}}$ as in (2.12). Compute the inverse DFT of $\widetilde{X}_{\lambda^{\mathrm{f}}}$ to obtain $\widetilde{x}_{\lambda^{\mathrm{f}}}$.

2) *Wavelet shrinkage*

Compute the DWT of the still noisy $\widetilde{x}_{\lambda^{\mathrm{f}}}$ to obtain $\widetilde{w}_{j,\ell;\lambda^{\mathrm{f}}}$. Shrink $\widetilde{w}_{j,\ell;\lambda^{\mathrm{f}}}$ with $\lambda^{\mathrm{w}}_{j,\ell}$ (using (A.7) or (A.8)) to obtain $\widehat{w}_{j,\ell} := \widetilde{w}_{j,\ell;\lambda^{\mathrm{f}}}\, \lambda^{\mathrm{w}}_{j,\ell}$. Compute the inverse DWT with the $\widehat{w}_{j,\ell}$ to obtain the ForWaRD estimate $\widehat{x}$.

For numerical robustness, the operator inversion in Step 1a and Fourier shrinkage in Step 1b are performed simultaneously.

2.5.2 How ForWaRD works

During operator inversion in Step 1a of the ForWaRD algorithm, some Fourier noise components are significantly amplified (see (2.10)). In Step 1b, ForWaRD employs a small amount of Fourier

shrinkage (most $\lambda^{\mathrm{f}}_k \approx 1$; $\lambda^{\mathrm{f}}_k \approx 0$ only when $|H(f_k)| \approx 0$) by choosing a small value for the regularization $\Lambda(f_k)$ that determines the λ^{f} in (2.11). Sections 2.5.3 and 2.6.2 contain details on the choice of λ^{f}. This minimal shrinkage is sufficient to significantly attenuate the amplified noise components with a minimal loss of signal components. Consequently, after the Fourier shrinkage step (see (2.12)), the leaked noise $\mathcal{H}^{-1}\gamma_{\lambda^{\mathrm{f}}}$ in the $\widetilde{x}_{\lambda^{\mathrm{f}}}$ has substantially reduced variances $\sigma^2_{j;\lambda^{\mathrm{f}}}$ in all wavelet coefficients. The variance $\sigma^2_{j;\lambda^{\mathrm{f}}}$ at wavelet scale j is given by

$$\begin{aligned}
\sigma^2_{j;\lambda^{\mathrm{f}}} &:= \mathbb{E}\left(|\langle \mathcal{H}^{-1}\gamma_{\lambda^{\mathrm{f}}}, \psi_{j,\ell}\rangle|^2\right) \\
&= \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} \frac{\sigma^2 |\Psi_{j,\ell}(f_k)|^2}{|H(f_k)|^2} |\lambda^{\mathrm{f}}_k|^2 \\
&= \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} \frac{\sigma^2 |H(f_k)\,\Psi_{j,\ell}(f_k)|^2}{\left(|H(f_k)|^2 \,+\, \Lambda(f_k)\right)^2}
\end{aligned} \tag{2.16}$$

with $\Psi_{j,\ell}$ the DFT of $\psi_{j,\ell}$. The retained signal part $x_{\lambda^{\mathrm{f}}}$ in $\widetilde{x}_{\lambda^{\mathrm{f}}}$ continues to be represented economically in the wavelet domain because $x_{\lambda^{\mathrm{f}}}$ lies in the same Besov space as the desired signal x (see Appendix D.1 for the justification). Therefore, the subsequent wavelet shrinkage in Step 2 effectively estimates the retained signal $x_{\lambda^{\mathrm{f}}}$ from the low-variance leaked noise $\mathcal{H}^{-1}\gamma_{\lambda^{\mathrm{f}}}$. Thus, ForWaRD's hybrid approach yields robust solutions to a wide variety of deconvolution problems (for example, see Fig. 2.4).

2.5.3 Balancing Fourier and wavelet shrinkage in ForWaRD

We now study the balance between the amount of Fourier shrinkage and wavelet shrinkage employed in the hybrid ForWaRD system so as to ensure low-MSE estimates. We consider an idealized ForWaRD system that performs Wiener-like Fourier shrinkage with α-parametrized λ^{f} as in

(2.13)—denoted by $\lambda^{\mathrm{f}}(\alpha)$ henceforth—and wavelet shrinkage with ideal oracle thresholding λ^{w} as in (A.6). The amounts of Fourier shrinkage and wavelet shrinkage are both automatically determined by simply choosing the α; the α also determines the wavelet shrinkage λ^{w} (see (A.6)) since it dictates the leaked noise variances $\sigma^2_{j;\lambda^{\mathrm{f}}(\alpha)}$ (see (2.16)).

The choice of α controls an interesting trade-off. On one hand, small values of α (so that most $\lambda^{\mathrm{f}}_k(\alpha) \approx 1$) are desirable to ensure that few signal components are lost during Fourier shrinkage, that is, to ensure that

$$\left\| X - X_{\lambda^{\mathrm{f}}(\alpha)} \right\|_2^2 \;=\; \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} |X(f_k)|^2 \, |1 - \lambda^{\mathrm{f}}_k(\alpha)|^2 \tag{2.17}$$

is minimized. But on the other hand, larger values of α result in smaller wavelet-domain noise variances $\sigma^2_{j;\lambda^{\mathrm{f}}(\alpha)}$ and thereby facilitate better estimation of the retained signal components $x_{\lambda^{\mathrm{f}}(\alpha)}$ via subsequent wavelet shrinkage. Ideally, we would like to set α such that the MSE of the final ForWaRD estimate is minimized.

An analytical expression for the optimal Fourier shrinkage determined by a single α is unfortunately intractable. Therefore, in this section, instead of minimizing the overall MSE via a single α, we will consider a more general ForWaRD system that employs a *different* Fourier shrinkage parameter α_j when computing the scale-j wavelet coefficients in the ForWaRD estimate. We desire to simultaneously set all the α_j's so that the overall MSE is minimized. Assuming an orthogonal DWT, the overall MSE is simply the sum of the MSEs at each wavelet scale. Thus, we can optimally set the α_j at each scale j independently of the other scales by minimizing the error in ForWaRD's scale-j wavelet coefficients. We then say that the amount of Fourier shrinkage and wavelet shrinkage is *balanced*.

Cost function

To determine the α_j that balances the amount of Fourier and wavelet shrinkage at scale j in ForWaRD, we use a cost function $\widetilde{\mathrm{MSE}}_j(\alpha_j)$ that closely approximates the actual scale-j MSE contribution $\mathrm{MSE}_j(\alpha_j)$

$$\begin{aligned}\widetilde{\mathrm{MSE}}_j(\alpha_j) &:= \frac{1}{N}\sum_{\ell=0}^{N_j-1}\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} |X(f_k)|^2|\Psi_{j,\ell}(f_k)|^2|1-\lambda_k^{\mathrm{f}}(\alpha_j)|^2 + \sum_{\ell=0}^{N_j-1}\min\Big(|w_{j,\ell}|^2,\, \sigma^2_{j;\lambda^{\mathrm{f}}(\alpha_j)}\Big) \\ &\approx \mathrm{MSE}_j(\alpha_j), \end{aligned} \tag{2.18}$$

with N_j the number of wavelet coefficients at scale j. The first term accounts for the signal components at scale j that are lost during Fourier shrinkage. The second term approximates the actual wavelet oracle thresholding error $\sum_\ell \min\Big(\big|\langle x_{\lambda^{\mathrm{f}}(\alpha_j)}, \psi_{j,\ell}\rangle\big|^2,\, \sigma^2_{j;\lambda^{\mathrm{f}}(\alpha_j)}\Big)$ [27]. (See also [13] for additional insights on the approximations.) We denote the $\widetilde{\mathrm{MSE}}_j(\alpha_j)$-minimizing regularization parameter by $\alpha_j^\star$ and the corresponding Fourier shrinkage by $\lambda^{\mathrm{f}}(\alpha_j^\star)$. As we shall soon see from the experimental results in Section 2.5.3, $\alpha_j^\star$ also nearly minimizes the actual error $\mathrm{MSE}_j(\alpha_j)$, thereby balancing the amount of Fourier and wavelet shrinkage.

Optimal Fourier shrinkage

We state the following result about the optimal $\alpha_j^\star$ that balances the amount of Fourier shrinkage and wavelet shrinkage at scale j (see Appendix C for the proof).

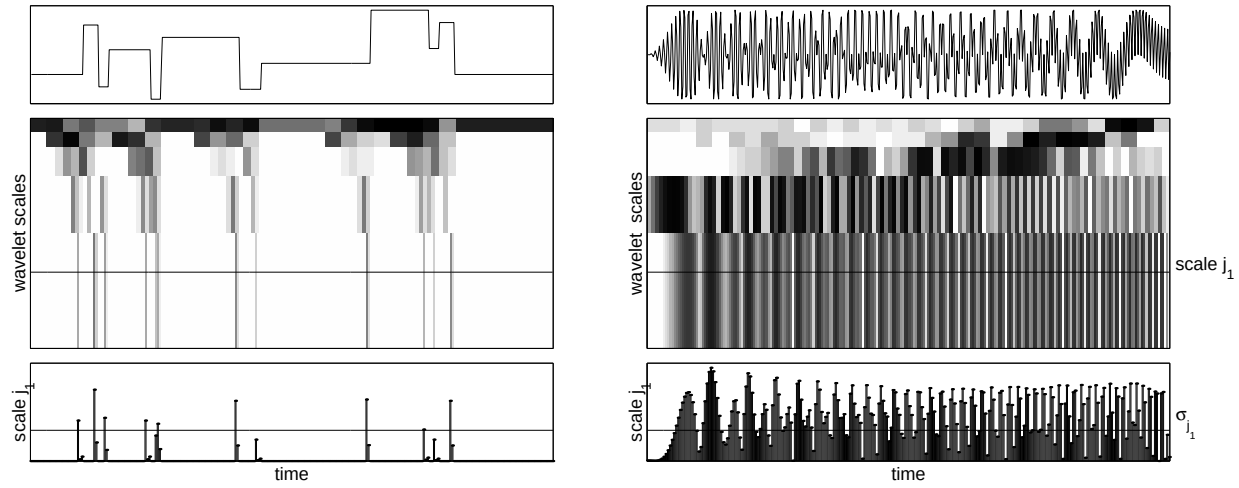
Proposition 1 *In a ForWaRD system employing Wiener-like Fourier shrinkage $\lambda^{\mathrm{f}}(\alpha_j)$ as in (2.13) and oracle wavelet shrinkage λ^{w} as in (A.6), the optimal scale-j regularization parameter $\alpha_j^\star$*

satisfies

$$\alpha_j^\star \;=\; \frac{1}{N_j}\#\left\{|w_{j,\ell}| > \sigma_{j;\lambda^{\mathrm{f}}(\alpha_j^\star)}\right\}. \tag{2.19}$$

Here, $\#\left\{|w_{j,\ell}| > \sigma_{j;\lambda^{\mathrm{f}}(\alpha_j^\star)}\right\}$ denotes the number of wavelet coefficients $w_{j,\ell}$ at scale j that are larger in magnitude than the noise standard deviation $\sigma_{j;\lambda^{\mathrm{f}}(\alpha_j^\star)}$. In words, (2.19) says that the approximate error in the scale-j wavelet coefficients is minimized when the regularization parameter determining the Fourier shrinkage equals the proportion of the desired signal wavelet coefficients with magnitudes larger than the corrupting noise standard deviation. Since the noise standard deviation is primarily determined by the Fourier structure of the convolution operator, we can infer that the balance between Fourier and wavelet shrinkage is simultaneously determined by the Fourier structure of the operator and the wavelet structure of the desired signal.

Proposition 1 quantifies the intuition that signals with more economical wavelet representations should require less Fourier shrinkage. To better understand Proposition 1, see Fig. 2.5, which displays the *Blocks* and the *TwoChirps* test signals and their wavelet coefficient time-scale plots. The *Blocks* signal has an economical wavelet-domain representation, and so only a small number of wavelet coefficient magnitudes would exceed a typical noise standard deviation σ_{j_1} at scale j_1. For *Blocks*, (2.19) would advocate a small $\alpha_{j_1}^\star \approx 0$ and thus most $\lambda_k^{\mathrm{f}}(\alpha_{j_1}^\star) \approx 1$ (see (2.13)); hence most Fourier components would be retained during Fourier shrinkage. However, a substantial amount of noise would also leak through the Fourier shrinkage. Therefore, many $\lambda_{j,\ell}^{\mathrm{w}} \ll 1$ and only the few dominant wavelet components would be retained during subsequent wavelet shrinkage. On the other hand, for a signal with an uneconomical wavelet representation like *TwoChirps*, (2.19) would advocate a large $\alpha_{j_1}^\star \approx 1$ and thus most $\lambda_k^{\mathrm{f}}(\alpha_{j_1}^\star) \ll 1$ and most $\lambda_{j_1,\ell}^{\mathrm{w}} \approx 1$. To summarize, (2.19)



(a) economical wavelet representation (b) uneconomical wavelet representation

Figure 2.5: *Effect of economical wavelet-domain signal representation on optimal Fourier shrinkage in ForWaRD. (a) The* Blocks *test signal (top) wavelet coefficient time-scale plot (middle) is illustrated with a darker shade indicating a larger magnitude for the coefficient* $w_{j,\ell}$ *corresponding to the wavelet basis function at scale* j *and localized at time* $2^{-j}\ell$*. The wavelet coefficient magnitudes at the scale* j_1*, marked by a solid horizontal line in the middle plot, are illustrated in the bottom plot. Since the number of coefficients exceeding a typical noise standard deviation* σ_{j_1}*, marked by a solid horizontal line in the bottom plot, is small for the economically represented* Blocks *signal, the* $\alpha^\star_{j_1}$ *would be* ≈ 0*. (b) In contrast, the* TwoChirps *test signal (top) has an uneconomical wavelet representation. Hence, the optimal amount of Fourier shrinkage will be large with* $\alpha^\star_{j_1} \approx 1$*.*

would recommend less Fourier and more wavelet shrinkage for signals with economical wavelet representations and vice versa for signals with uneconomical wavelet representations. Thus (2.19) balances the amount of Fourier shrinkage and wavelet shrinkage in ForWaRD based on the economy of the desired signal wavelet representation with respect to the corrupting noise variance.

We clarify that while Proposition 1 provides valuable intuition, it cannot be employed in a practical ForWaRD system because (2.19) requires knowledge of the desired signal's wavelet coefficient magnitudes.

Experimental verification

We now experimentally verify that the optimal $\alpha_j^\star$'s predicted by Proposition 1 balance the amount of Fourier and wavelet shrinkage in ForWaRD and lead to low overall MSE. The experimental setup consists of the desired image, blurring function, and noise level described in Section 2.7. We assume complete knowledge of the desired image's wavelet coefficient magnitudes to perform oracle thresholding and to compute the optimal $\alpha_j^\star$'s by (2.19). The first column in Table 2.1 specifies the 2-D wavelet subbands at each scale j—high-pass vertically and horizontally (HH), high-pass vertically and low-pass horizontally (HL), and low-pass vertically and high-pass horizontally (LH). The second column lists the optimal $\widetilde{\mathrm{MSE}}_j(\alpha_j)$-minimizing $\alpha_j^\star$ computed using (2.19) for each scale and subband. The third column lists the α_j's that minimize the actual MSE in each subband at scale j. The fourth column lists the $\%$ increase in the actual MSE due to using the $\alpha_j^\star$'s instead of the α_j's minimizing the actual MSE. Even for the worst case (1st row), the MSE performance with the $\alpha_j^\star$ differs from the best possible MSE performance by less than 7%. Thus, the experiment verifies that the $\alpha_j^\star$ from (2.19) nearly minimize the actual MSE in ForWaRD.

2.5.4 Asymptotic ForWaRD performance and optimality

We now analyze the asymptotic ForWaRD MSE performance (as the number of signal samples $N \to \infty$) and prove its optimality in recovering Besov space signals. Considering asymptotic performance is natural because with technological advances, the resolution of signals and images is continually increasing. We will perform our analysis using a number of steps. We assume a ForWaRD system that employs Fourier-Tikhonov shrinkage as in (2.14) and employs wavelet hard thresholding as in (A.7). For such a system, assuming that the Fourier-Tikhonov shrinkage

Table 2.1: *Experimental verification that (2.19) balances Fourier and wavelet shrinkage in ForWaRD.*

$\{j$, subband$\}$	$\alpha_j^\star$: $\widetilde{\mathrm{MSE}_j(\alpha_j)}$ minimizer from (2.19)	$\mathrm{MSE}_j(\alpha_j)$ minimizer	$\%$ Increase in $\mathrm{MSE}_j(\alpha_j)$
$\{5$, HH$\}$	0.16	0.060	6.5
$\{5$, HL$\}$	0.16	0.14	0.47
$\{5$, LH$\}$	0.23	0.16	0.47
$\{4$, HH$\}$	0.18	0.16	0.57
$\{4$, HL$\}$	0.29	0.35	0.23
$\{4$, LH$\}$	0.34	0.35	0.12
$\{3$, HH$\}$	0.33	0.55	5.4
$\{3$, HL$\}$	0.50	0.65	6.2
$\{3$, LH$\}$	0.55	0.75	6.0
$\{2$, HH$\}$	0.84	0.75	1.0
$\{2$, HL$\}$	0.95	1.0	0.011
$\{2$, LH$\}$	0.93	0.65	2.0
$\{1$, HH$\}$	0.98	0.55	3.2
$\{1$, HL$\}$	1.0	1.0	0.0
$\{1$, LH$\}$	1.0	1.0	0.0

remains unchanged with N and assuming mild conditions on $\mathcal{H}$, we first establish in Proposition 2 the behavior of the distortion due to Fourier shrinkage and the error due to wavelet shrinkage as $N \to \infty$. Then, in Proposition 3, by allowing the Fourier-Tikhonov shrinkage to decay with N, we prove that for scale-invariant deconvolution problems, ForWaRD also enjoys the optimal rate of MSE decay like the WVD.

Proposition 2 *For a ForWaRD system with Fourier-Tikhonov shrinkage λ^{f} as in (2.14) with a fixed $\tau > 0$ and wavelet hard thresholding λ^{w} as in (A.7), the per-sample distortion due to loss of signal components during Fourier shrinkage*

$$\frac{1}{N}\sum_{n=0}^{N-1} |x(n) - x_{\lambda^{\mathrm{f}}}(n)|^2 \;\to\; C_1 \tag{2.20}$$

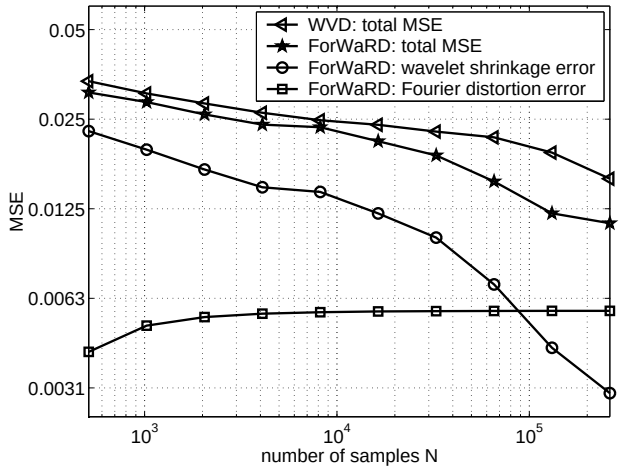


Figure 2.6: *MSE performance of ForWaRD compared to the WVD [8] at different N. The MSE incurred by ForWaRD's wavelet shrinkage step decays much faster than the WVD's MSE with increasing N, while the Fourier distortion error saturates to a constant.*

as $N \to \infty$, with $C_1 > 0$ a constant. Further, if the underlying continuous-time $x(t) \in B^s_{p,q}$, $s > \frac{1}{p} - \frac{1}{2}$, $1 < p, q < \infty$ and $\mathcal{H}$ is a convolution operator whose squared-magnitude frequency response is of bounded variation over dyadic frequency intervals, then the per-sample wavelet shrinkage error in estimating the signal part $x_{\lambda^{\mathrm{f}}}$ retained during Fourier shrinkage with λ^{f} decays with $N \to \infty$ as

$$\frac{1}{N}\,\mathbb{E}\left(\sum_{n=0}^{N-1} |x_{\lambda^{\mathrm{f}}}(n) - \widehat{x}(n)|^2\right) \leq C_2\, N^{\frac{-2s}{2s+1}} \tag{2.21}$$

with $C_2 > 0$ a constant and $\widehat{x}$ the ForWaRD estimate.

Refer to Appendix D for the proof of (2.21); the proof of (2.20) is immediate. The bounded variation assumption is a mild smoothness requirement that is satisfied by a wide variety of $\mathcal{H}$. The bound (2.21) in Proposition 2 asserts that ForWaRD's wavelet shrinkage step is extremely effective, but it comes at the cost of a constant per-sample distortion (assuming τ is kept constant with N).

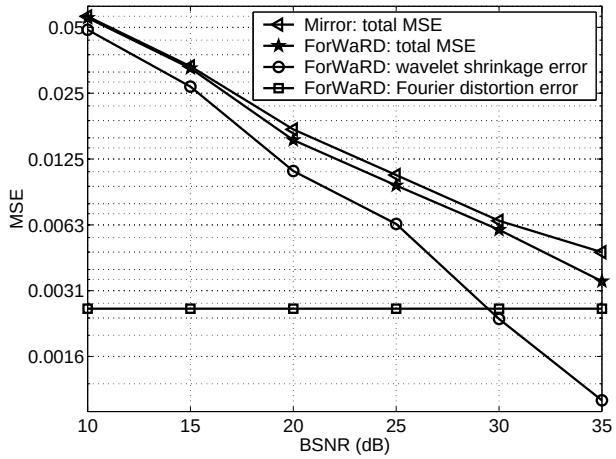


Figure 2.7: *MSE performance of ForWaRD compared to the mirror-wavelet basis approach [5] at different BSNRs.*

Consider an example using $\mathcal{H}$ with frequency response $|H(f_k)| \propto (|k|+1)^{-\nu}$, $\nu > 0$, for which the WVD is optimal. The per-sample ForWaRD MSE (assuming constant $\tau > 0$ with N) decays with a rapid rate of $N^{\frac{-2s}{2s+1}}$ but converges to a non-zero constant. In contrast, the per-sample WVD MSE decays to zero but with a slower rate of $N^{\frac{-2s}{2s+2\nu+1}}$ (see (2.15)) [8]. Thus, the drawback of the asymptotic bias is offset by the much improved ForWaRD MSE performance at small sample lengths. To experimentally verify ForWaRD's asymptotic performance and compare it with WVD, we blurred the 1-D, zero-mean *Blocks* test signal (see Fig. 2.5(a) (top)) using $\mathcal{H}$ with a DFT response $H(f_k) = 200/(|k|+1)^{-2}$ and added noise with variance $\sigma^2 = 0.01$ for N ranging from 2^9 to 2^{18}. To obtain the ForWaRD estimate, we employ Fourier-Tikhonov shrinkage using (2.14) with $\tau = 5 \times 10^{-4}$. For both the ForWaRD and WVD estimate, we employ wavelet shrinkage using (A.7) with $\rho_j = \sqrt{2 \log N}$ [10, 27]. Figure 2.6(a) verifies that ForWaRD's Fourier distortion error stays unchanged with N; the smaller the Fourier shrinkage (smaller τ), the smaller the distortion. However, ForWaRD's wavelet shrinkage error decays significantly faster with increasing N than the overall WVD error. Consequently, the overall ForWaRD MSE remains below the WVD MSE

over a wide range of sample lengths N that are of practical interest.

If τ is kept *fixed* with increasing N, then the WVD MSE will eventually catch up and improve upon the ForWaRD MSE. We will now show that if the τ controlling the Fourier shrinkage in ForWaRD is *tuned* appropriately at each N, then as stated in Proposition 3, ForWaRD will also enjoy an asymptotically optimal MSE decay rate like the WVD.

Proposition 3 *Let $\mathcal{H}$ be an operator with frequency response $|H(f_k)| \propto (|k|+1)^{-\nu}$, $\nu > 0$. Let $x(t) \in B^s_{p,q}$, $s > \min\left(0, \left(\frac{2}{p}-1\right)\nu, \frac{1}{p}-\frac{1}{2}\right)$, $1 < p, q < \infty$. Consider a ForWaRD system with Fourier-Tikhonov shrinkage λ^{f} as in (2.14) and wavelet hard thresholding λ^{w} as in (A.7). If the τ parameterizing λ^{f} is such that*

$$\tau \leq C_3 N^{-\beta}, \tag{2.22}$$

with

$$\beta > \frac{s}{2s+2\nu+1} \max\left(1, \frac{4\nu}{\min\left(2s, 2s+1-\frac{2}{p}\right)}\right), \tag{2.23}$$

for some constant $C_3 > 0$, then the per-sample ForWaRD MSE decays as

$$\frac{1}{N}\,\mathbb{E}\left(\sum_{n=0}^{N-1} |x(n) - \widehat{x}(n)|^2\right) \leq C_4\, N^{\frac{-2s}{2s+2\nu+1}} \tag{2.24}$$

with $N \to \infty$ for some constant $C_4 > 0$. Further, no estimator can achieve a faster error decay rate than ForWaRD for every $x(t) \in B^s_{p,q}$.

The basic idea behind the proof is to show that both the wavelet shrinkage error (2.21) and the Fourier distortion error (2.20) decay as $N^{\frac{-2s}{2s+2\nu+1}}$ (see Appendix E for the details). It is easy to infer that the wavelet shrinkage error decays as fast as the WVD error due to the relatively lower

noise levels after Fourier shrinkage. The Fourier distortion error monotonically increases with τ. We prove that a τ that decays as $N^{-\beta}$ drives the Fourier distortion error to also decay as $N^{\frac{-2s}{2s+2\nu+1}}$. For example, Proposition 3 guarantees that if $x(t) \in B^1_{1,1}$, $\nu = 0.5$, and $\tau \propto N^{-\beta}$ at each N with $\beta > 0.5$, then the per-sample ForWaRD MSE will decay at the optimal rate of $N^{\frac{-2}{\tau}}$ as $N \to \infty$.

Further, tuning τ to precisely *minimize* the ForWaRD MSE at each N would ensure that the ForWaRD MSE curve remains below (or at least matches) the WVD's MSE curve at all sample lengths for scale-invariant $\mathcal{H}$. This follows from the fact that for $\tau = 0$, ForWaRD is trivially equivalent to the WVD.

ForWaRD can also match or improve upon the performance of adapted mirror-wavelet deconvolution [5]. To experimentally compare the MSE performance of ForWaRD with mirror-wavelets, we blurred the 1-D, zero-mean *Blocks* test signal (see Fig. 2.5(a) (top)) using $\mathcal{H}$ with a DFT response $H(f_k) = (1 - \frac{2|k|}{N})^2$. The mirror-wavelet approach is designed to optimally tackle such a hyperbolic deconvolution problem [5]. We fixed the number of samples at $N = 1024$ and varied the amount of additive noise so that the blurred signal-to-noise ratios (BSNRs) ranged from 10 dB to 35 dB. The BSNR is defined as $10 \log_{10} \left(\frac{\|(x \circledast h) - \mu(x \circledast h)\|_2^2}{N\sigma^2} \right)$, where $\mu(x \circledast h)$ denotes the mean of the blurred image $x \circledast h$ samples. To obtain the ForWaRD estimate at each BSNR, we employed Fourier-Tikhonov shrinkage using $\tau = 10^{-2}$. In the wavelet and mirror-wavelet domain, we employed shrinkage using (A.7) with $\rho_j = \sqrt{2 \log N}$ to obtain the ForWaRD and mirror-wavelet estimate respectively. In Fig. 2.7(b), the MSE incurred by ForWaRD's wavelet shrinkage step decays much faster than the mirror-wavelet's MSE with increasing BSNR (that is, with reducing noise), while the Fourier distortion error stays constant. The overall ForWaRD MSE stays below the mirror-wavelet MSE over the entire BSNR range. The ForWaRD performance demonstrated in

Fig. 2.7(b) gives us reason to conjecture that ForWaRD with appropriately chosen Fourier shrinkage should match mirror-wavelet's optimal asymptotic performance in hyperbolic deconvolution problems.

2.6 ForWaRD Implementation

To ensure good results with ForWaRD, the noise variance σ^2, the Fourier shrinkage, and the wavelet shrinkage on ForWaRD need to be set appropriately.

2.6.1 Estimation of σ^2

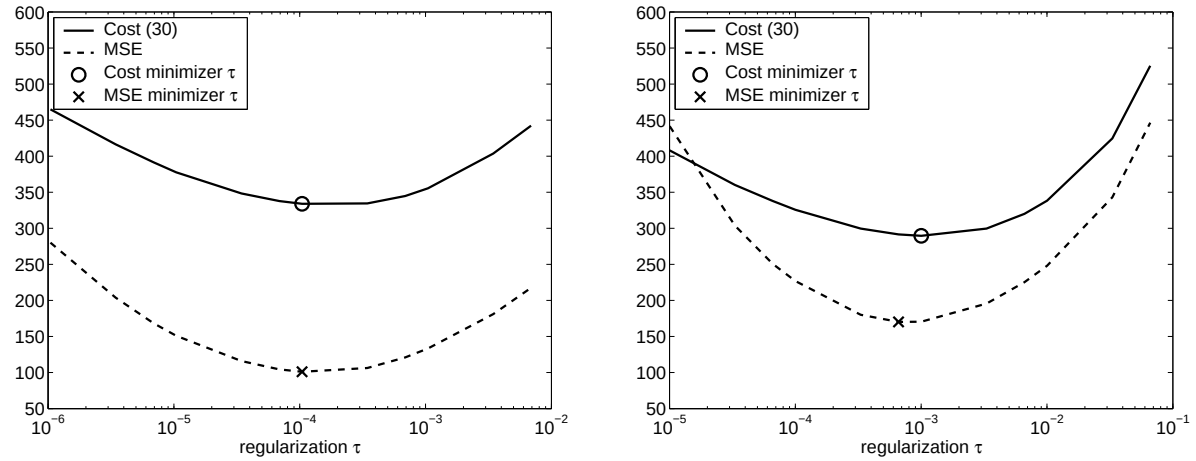
The variance σ^2 of the additive noise γ in (2.1) is typically unknown in practice and must be estimated from the observation y. The noise variance can be reliably estimated using a median estimator on the finest scale wavelet coefficients of y [22].

2.6.2 Choice of Fourier shrinkage

In practice, we employ Fourier-Tikhonov shrinkage λ^{f} (see (2.14)) with the parameter $\tau > 0$ set judiciously. We desire to choose the τ that minimizes the ForWaRD MSE $\|x - \widehat{x}\|_2^2$. However, since x is unknown, we set τ such that the ForWaRD estimate agrees well with the observation y. That is, we choose the τ that minimizes the observation-based cost

$$\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} \frac{|H(f_k)|^2}{|H(f_k)|^2+\eta} \frac{1}{|H(f_k)|} \left| H(f_k)\widehat{X}(f_k) - Y(f_k) \right|^2 \tag{2.25}$$

with $\eta := \frac{N\sigma^2}{\|y-\mu(y)\|_2^2}$ and $\mu(y) := \sum_n \frac{y(n)}{N}$ the mean value of y. The term $\frac{|H(f_k)|^2}{|H(f_k)|^2+\eta}$ (see (2.12)) simply weighs the error $\left| H(f_k)\widehat{X}(f_k) - Y(f_k) \right|^2$ between the blurred estimate and the observation at



(a) 40 dB BSNR (b) 30 dB BSNR

Figure 2.8: *Choice of Fourier-Tikhonov regularization parameter τ. In each plot, the solid line denotes the observation-based cost (2.25) and the dashed lines denotes the actual MSE; the respective minima are marked by "$\circ$" and "$\times$". The plots illustrate that the cost (2.25)-minimizing τ's at 40 dB and 30 dB BSNRs yield estimates whose MSEs are within 0.1 dB of the minimum possible MSE.*

the different frequencies to appropriately counter-balance the effect of $\frac{1}{H(f_k)}$. The τ that minimizes the cost (2.25) provides near-optimal MSE results for a wide variety of signals and convolution operators. For example, for the problem setup described in Section 2.7 with 40 dB and 30 dB BSNRs, Fig. 2.8(a) and (b) illustrate that the τ's minimizing (2.25) yield estimates whose MSEs are within 0.1 dB of the minimum possible MSEs. Since the MSE performance of ForWaRD is insensitive to small changes around the MSE-optimal τ, a logarithmically spaced sampling of the typically observed τ-range $[0.01, 10] \times \frac{N\sigma^2}{\|y-\mu(y)\|_2^2}$ is sufficient to efficiently estimate the best τ and determine the Fourier shrinkage λ^{f}.

2.6.3 Choice of wavelet basis and shrinkage

Estimates obtained by shrinking DWT coefficients are not shift-invariant, that is, translations of y will result in different ForWaRD estimates. We exploit the redundant, shift-invariant DWT to

obtain improved shift-invariant estimates [10] by averaging over all possible shifts at $O(N \log N)$ computational cost for N-sample signals. (Complex wavelets can also be employed to obtain near shift-invariant estimates at reduced computational cost [29, 30]). We shrink the redundant DWT coefficients using the WWF (see (A.8)) rather than hard thresholding due to its superior performance.

2.7 Results

2.7.1 Simulated problem

We illustrate the performance of ForWaRD (implemented as described in Section 2.6) using a 2-D deconvolution problem described by Banham et al. [15]. A self-contained Matlab implementation of ForWaRD is available at *www.dsp.rice.edu/software* to facilitate easy reproduction of the results. We choose the 256×256 *Cameraman* image as the x and the 2-D 9×9-point box-car blur $\mathcal{H}$ with discrete-time system response $h(n_1, n_2) = \frac{1}{81}$ for $0 \leq n_1, n_2 \leq 8$ and 0 otherwise. We set the additive noise variance σ^2 such that the BSNR is 40 dB.

Figure 2.4 illustrates the desired x, the observed y, the LTI Wiener filter estimate, and the ForWaRD estimate. The regularization $\tau = 3.4 \times 10^{-4}$ determining the Fourier-Tikhonov shrinkage is computed as described in Section 2.6.2. The $|X(f_k)|^2$ required by the LTI Wiener filter is estimated using the iterative technique of [6]. As we see in Fig. 2.4, the ForWaRD estimate, with signal-to-noise ratio (SNR) = 22.5 dB, clearly improves upon the LTI Wiener filter estimate, with SNR = 20.8 dB; the smooth regions and most edges are well-preserved in the ForWaRD estimate. In contrast, the LTI Wiener filter estimate displays visually annoying ripples because the underlying Fourier basis elements have support over the entire spatial domain. The ForWaRD es-

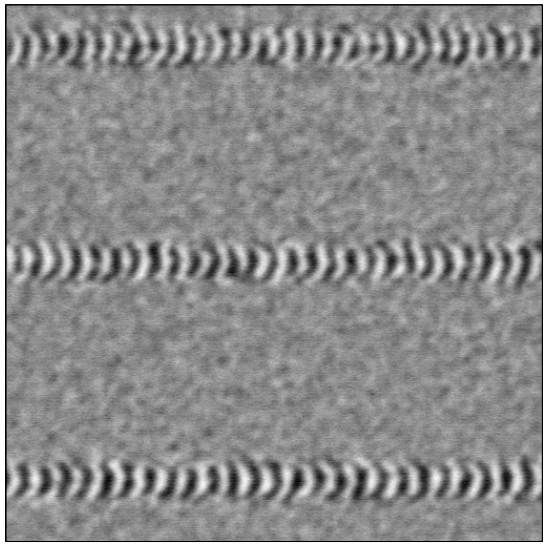
timate also improves on the multiscale Kalman estimate proposed by Banham et al. [15] in terms of improvement in signal-to-noise-ratio (ISNR) $:= 10\log_{10}(\|x-y\|_2^2/\|x-\widehat{x}\|_2^2)$. (During ISNR calculations, the y is aligned with the estimate $\widehat{x}$ by undoing the shift caused by the convolution operator. For the 9×9 box-car operator, y is cyclically shifted by coordinates $(4,4)$ toward the top-left corner to the minimize the ISNR [19].) Banham et al. report an ISNR of 6.7 dB; ForWaRD provides an ISNR of 7.3 dB. For the same experimental setup but with a substantially higher noise level of BSNR = 30 dB, ForWaRD provides an estimate with SNR = 20.3 dB and ISNR = 5.1 dB compared to the LTI Wiener filter estimate's SNR = 19 dB and ISNR = 3.8 dB. Both the WVD and mirror-wavelet basis approaches [5] are not applicable in these cases since the box-car blur used in the example has multiple frequency-domain zeros. Wiener SNR = 18.1 dB and ISNR = 3 dB

2.7.2 Real-life application: Magnetic Force Microscopy

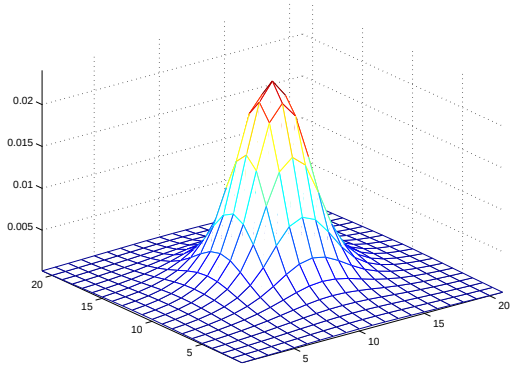
We employ ForWaRD on experimental data obtained using Magnetic Force Microscopy (MFM) to illustrate that it provides good performance on real-life problems as well.[5] Figure 2.9(a) shows a measured MFM image of the surface of a magnetic disk; the tracks of black and white patches are measured magnetic equivalents of bits 0 and 1. Due to instrument limitations, the MFM observation in Fig. 2.9(a) is blurred and noisy. The MFM measurement can modeled as being comprised of a desired image blurred by the *sensitivity field function* of Fig. 2.9(b) and additive, possibly colored, noise. Figure 2.9(c) illustrates the ForWaRD estimate obtained assuming that the additive noise is white; the implementation is described in Section 2.6. The Fig. 2.9(d) estimate is obtained using a modified ForWaRD implementation that does not assume any knowledge about the noise

[5]The Magnetic Force Microscopy image and sensitivity field function were provided courtesy of Dr. Frank M. Candocia and Dr. Sakhrat Khizroev from Florida International University, Department of Electrical and Computer Engineering.

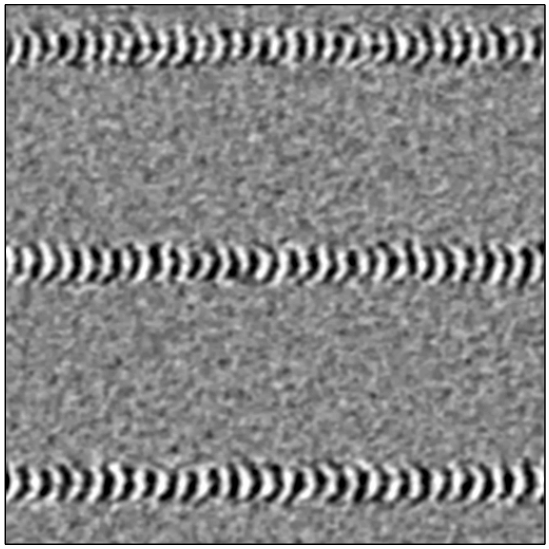
spectrum. The modified algorithm estimates the additive noise's variance at *each* wavelet scale using a median estimator [22]; in contrast, for the white noise case, a common noise variance is estimated for *all* wavelet scales (see Section 2.6.1). We believe that both Figs. 2.9(c) and (d) provide fairly accurate and consistent representations of the three bit-tracks in the magnetic disk. However, the estimates of the background media differ significantly depending on the assumptions about the additive colored noise's structure.



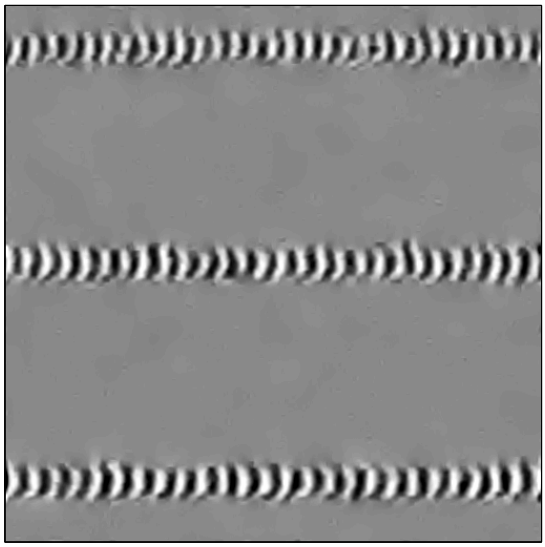
(a) MFM measurement



(b) Sensitivity field function



(c) ForWaRD estimate (white noise)



(d) ForWaRD estimate (colored noise)

Figure 2.9: *(a) Blurred and noisy Magnetic Force Microscopy (MFM) measurement (*512×512 *samples). (b) Sensitivity field function. The MFM observation in (a) is believed to have been blurred by this function. (c) and (d) ForWaRD estimates obtained assuming that the MFM measurement contains additive white noise and additive colored colored respectively. Data courtesy of Dr. Frank M. Candocia and Dr. Sakhrat Khizroev from Florida International University, Department of Electrical and Computer Engineering.*

Chapter 3

WInHD: Wavelet-based Inverse Halftoning via Deconvolution

Once you become informed, you start seeing complexities and shades of gray.

You realize that nothing is as clear and simple as it first appears.

–Bill Watterson

3.1 Introduction

Digital halftoning is a common technique used to render a sampled gray-scale image using only black or white dots [1] (see Figures 3.3(a) and (b)); the rendered bi-level image is referred to as a halftone. *Inverse halftoning* is the process of retrieving a gray-scale image from a given halftone. Applications of inverse halftoning include rehalftoning, halftone resizing, halftone tone correction, and facsimile image compression [2, 31]. In this chapter, we focus on inverse halftoning images that are halftoned using popular error diffusion techniques such as those of Floyd et al. [32], and Jarvis et al. [33] (hereby referred to as Floyd and Jarvis respectively).

Error-diffused halftoning is non-linear because it uses a quantizer to generate halftones. Recently, Kite et al. proposed an accurate linear approximation model for error diffusion halftoning (see Figure 3.4) [34, 35]. Under this model, the halftone $y(n_1, n_2)$ is expressed in terms of the

original gray-scale image $x(n_1, n_2)$ and additive white noise $\gamma(n_1, n_2)$ as (see Figure 3.1)

$$\begin{aligned} y(n_1, n_2) &= \mathcal{P}x(n_1, n_2) + \mathcal{Q}\gamma(n_1, n_2) \\ &= (p * x)(n_1, n_2) + (q * \gamma)(n_1, n_2), \end{aligned} \tag{3.1}$$

with $*$ denoting convolution and (n_1, n_2) indexing the pixels. The $\mathcal{P}$ and $\mathcal{Q}$ are the linear time-invariant (LTI) systems with respective impulse responses $p(n_1, n_2)$ and $q(n_1, n_2)$ determined by the error diffusion technique.

From (3.1), we infer that inverse halftoning can be posed as the classical *deconvolution* problem because the gray-scale image $x(n_1, n_2)$ can be obtained from the halftone $y(n_1, n_2)$ by deconvolving the filter $\mathcal{P}$ in the presence of the colored noise $\mathcal{Q}\gamma(n_1, n_2)$. Conventionally, deconvolution is performed in the Fourier domain. The Wiener deconvolution filter, for example, would estimate $x(n_1, n_2)$ by inverting $\mathcal{P}$ and *regularizing* the resulting noise with scalar Fourier shrinkage. As we will see, inverse halftoning using a Gaussian low-pass filter (GLPF) [36] can be interpreted as a naive Fourier deconvolution approach to inverse halftoning.

Unfortunately, all Fourier-based deconvolution techniques induce ringing and blurring artifacts due to the fact that the energy of edge discontinuities spreads over many Fourier coefficients. As a result of this uneconomical representation, the desirable edge Fourier coefficients are easily confounded with those due to the noise [8, 10, 11].

In contrast, the wavelet transform provides an economical representation for images with sharp edges [37]. This economy makes edge wavelet coefficients easy to distinguish from those due to the noise and has led to powerful image estimation algorithms based on scalar wavelet shrinkage [10, 27].

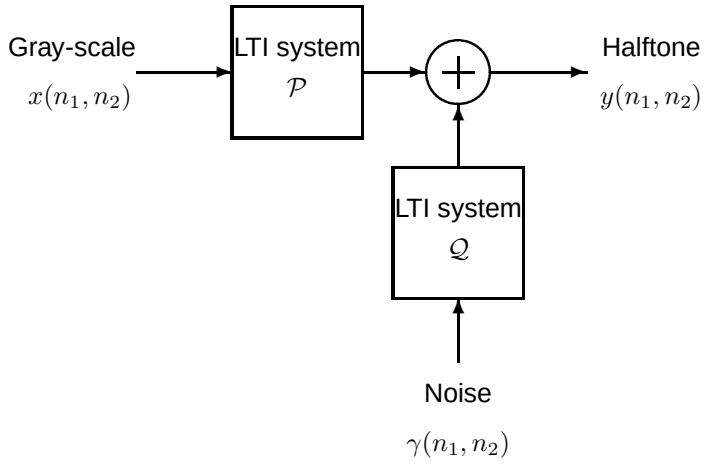


Figure 3.1: *Linear approximation for error diffusion halftoning. Under the linear model of [34, 35], the error-diffused halftone $y(n_1, n_2)$ comprises the original gray-scale image $x(n_1, n_2)$ passed through an LTI system $\mathcal{P}$ and white noise $\gamma(n_1, n_2)$ colored by an LTI system $\mathcal{Q}$. The systems $\mathcal{P}$ and $\mathcal{Q}$ are determined by the error diffusion technique.*

The wavelet transform was first exploited in inverse halftoning by J. Luo et al. [38]. Xiong et al. extended this algorithm using non-orthogonal, redundant wavelets to obtain improved results for error-diffused halftones [39]. Both these algorithms rely on a variety of steps such as clipping and edge-adapted noise attenuation in the wavelet subbands to exploit different empirical observations. However, these steps and their implications are not theoretically well-justified.

To simultaneously exploit the economy of wavelet representations and the interplay between inverse halftoning and deconvolution, we propose the *Wavelet-based Inverse Halftoning via Deconvolution* (WInHD) algorithm (see Figure 3.2) [40]. WInHD provides robust estimates by first inverting the convolution operator $\mathcal{P}$ determined by the linear model (3.1) for error diffusion and then effectively attenuating the residual colored noise using wavelet-domain scalar shrinkage operations [22, 27]. Since WInHD is model-based, it easily adapts to different error diffusion halftoning techniques. See Figure 3.3 for simulation results.

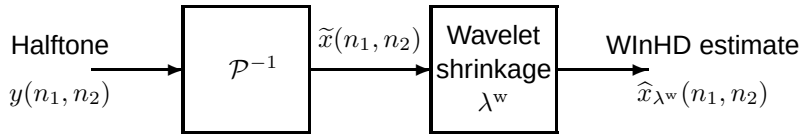


Figure 3.2: *Wavelet-based Inverse Halftoning via Deconvolution (WInHD). WInHD inverts the convolution operator $\mathcal{P}$ to obtain a noisy estimate $\widetilde{x}(n_1, n_2)$ of the gray-scale image. Subsequent scalar shrinkage with λ^{w} in the wavelet domain (for example, level-dependent hard thresholding) effectively attenuates the residual noise corrupting $\widetilde{x}(n_1, n_2)$ to yield the WInHD estimate $\widehat{x}_{\lambda^{\mathrm{w}}}(n_1, n_2)$.*

Unlike previous inverse halftoning algorithms, we can analyze the theoretical performance of WInHD under certain conditions. For images in a Besov smoothness space, we derive the minimum rate at which the WInHD estimate's mean-squared-error (MSE) decays as the resolution increases; that is, as number of pixels in the gray-scale image tends to infinity. We assume that the linear model for error diffusion (3.1) is exact and that the noise $\gamma(n_1, n_2)$ is Gaussian. Further, if the gray-scale image $x(n_1, n_2)$ contains some additive noise (say, scanner noise) before halftoning that is Gaussian, then we show that the MSE decay rate enjoyed by WInHD in estimating the noise-free $x(n_1, n_2)$ is optimal; that is, no other inverse halftoning algorithm can have a better error decay rate for every Besov space image as the number of image pixels increases.

Section 3.2 describes Kite et al.'s linear model for error diffusion halftoning from [34, 35]. Section 3.3 clarifies the equivalence between inverse halftoning and deconvolution and also analyzes Fourier-domain inverse halftoning. Section 3.4 discusses the proposed WInHD algorithm and its theoretical performance. Section 3.5 illustrates the experimental performance of WInHD. A short overview of wavelets in Appendix A and a technical proof in Appendix F complement this chapter.

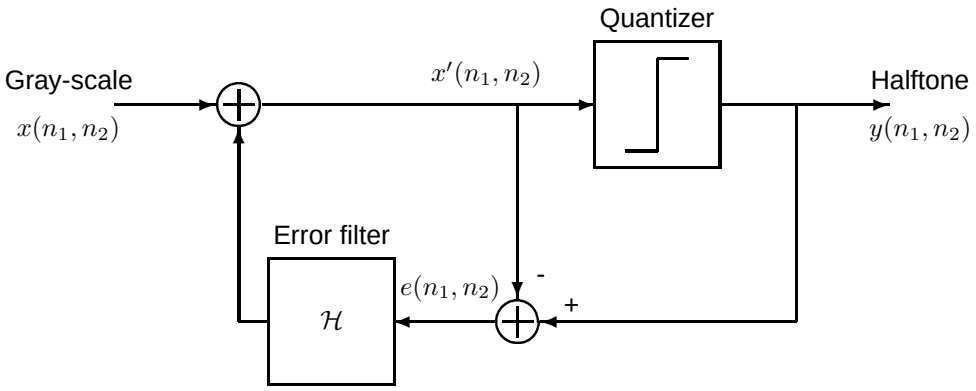
(a) Original $x(n_1, n_2)$

(b) Floyd halftone $y(n_1, n_2)$

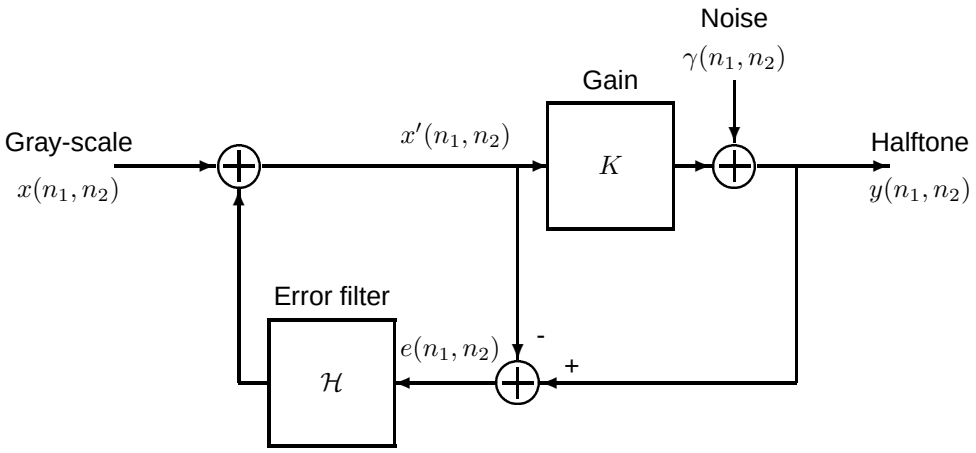
(c) Gradient estimate [41] (PSNR = 31.3 dB)

(d) WInHD estimate (PSNR = 32.1 dB)

Figure 3.3: *(a) Original Lena image (*512×512 *pixels). (b) Floyd halftone. (c) Multiscale gradient-based estimate [41], PSNR = 31.3 dB. (d) WInHD yields competitive PSNR performance (32.1 dB) and visual performance. (All documents including the above images undergo halftoning during printing. To minimize the halftoning effect, the images have been reproduced at the maximum size possible.) See Figure 3.8 for image close-ups.*



(a) Error diffusion halftoning



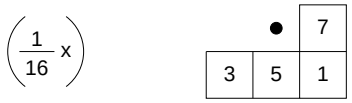
(b) Linear approximation

Figure 3.4: *(a) Error diffusion halftoning. The gray-scale image pixels* $x(n_1, n_2)$ *are quantized to yield* $y(n_1, n_2)$ *and the quantization error* $e(n_1, n_2)$ *is diffused over a causal neighborhood by the error filter* $\mathcal{H}$*. (b) The linear model approximates the quantizer with gain* K *and additive white noise* $\gamma(n_1, n_2)$ *[34].*

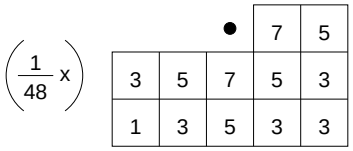
3.2 Linear Model for Error Diffusion

In this section, we describe the non-linear error diffusion halftoning and the linear approximation proposed in [34, 35].

Digital halftoning is a process that converts a given gray-scale digital image (for example, each pixel value $\in [0, 1, \ldots, 255]$) into a bi-level image (for example, each pixel value $= 0$ or 255) [1]. Error diffusion halftoning is one popular approach to perform digital halftoning. The idea is to take



$h(n_1, n_2)$ for Floyd [32]



$h(n_1, n_2)$ for Jarvis [33]

Figure 3.5: *Error filters $h(n_1, n_2)$ for Floyd [32] and Jarvis [33] error diffusion. The quantization error at the black dot is diffused over a causal neighborhood according the displayed weights.*

the error from quantizing a gray-scale pixel to a bi-level pixel and diffuse this quantization error over a causal neighborhood. The error diffusion ensures that the spatially-localized average pixel values of the halftone and original gray-scale image are similar; therefore, the halftone visually resembles the gray-scale image. Figure 3.4(a) illustrates the block diagram for error diffusion halftoning. The $x(n_1, n_2)$ denote the pixels of the input gray-scale image and $y(n_1, n_2)$ denote the bi-level pixels of the output halftone. The $x'(n_1, n_2)$, which yields $y(n_1, n_2)$ after quantization, is obtained by diffusing the quantization error $e(n_1, n_2)$ over a causal neighborhood of $x(n_1, n_2)$ using the error filter $\mathcal{H}$. The quantizer makes error-diffused halftoning a non-linear technique. Error diffusion techniques such as Floyd [32] and Jarvis [33] are characterized by their choice of $\mathcal{H}$'s impulse response $h(n_1, n_2)$ (see Figure 3.5).

Recently, Kite et al. proposed an accurate linear model for error diffusion halftoning [34, 35]. This model accurately predicts the "blue noise" (high-frequency noise) and edge sharpening effects observed in various error-diffused halftones. As shown in Figure 3.4(b), this model approximates the effects of quantization using a gain K followed by the addition of white noise $\gamma(n_1, n_2)$. The

halftone $y(n_1, n_2)$ can then be written in terms of the gray-scale image $x(n_1, n_2)$ and the additive white noise $\gamma(n_1, n_2)$ as in (3.1); the error diffusion technique determines the 2-dimensional (2-D) frequency responses of the LTI filters $\mathcal{P}$ and $\mathcal{Q}$ as

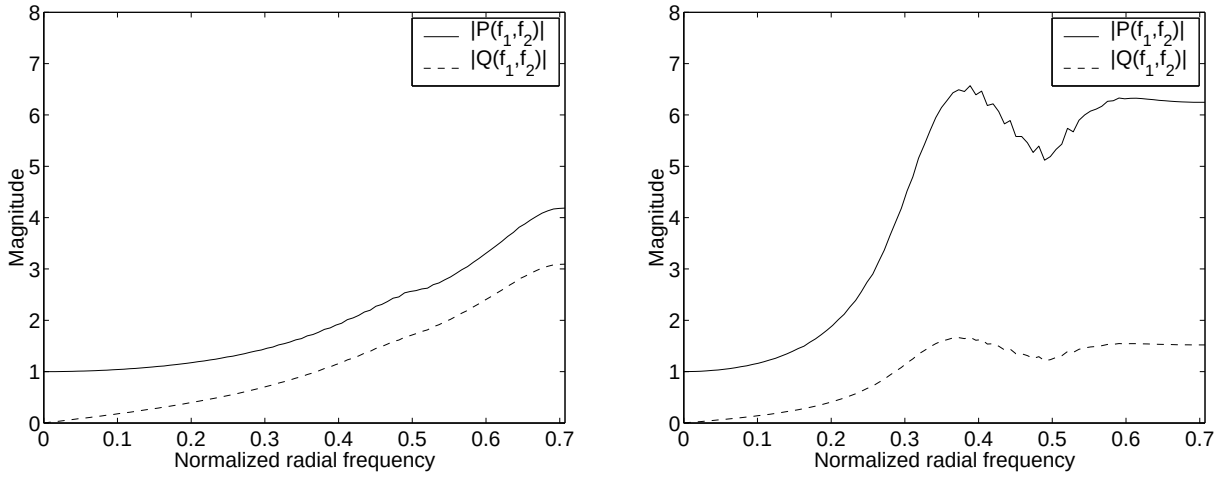
$$P(f_1, f_2) := \frac{K}{1 + (K-1)H(f_1, f_2)}, \tag{3.2}$$

$$Q(f_1, f_2) := \frac{1 - H(f_1, f_2)}{1 + (K-1)H(f_1, f_2)} \tag{3.3}$$

with $P(f_1, f_2)$, $Q(f_1, f_2)$, and $H(f_1, f_2)$ denoting the respective 2-D Fourier transforms of $p(n_1, n_2)$, $q(n_1, n_2)$, and $h(n_1, n_2)$. For any given error diffusion technique, Kite et al. found that the gain K is almost constant for different images. However, the K varied with the error diffusion technique [34]; for example, $K = 2.03$ for Floyd, while $K = 4.45$ for Jarvis. Figure 3.6 (a) and (b) illustrate the radially-averaged frequency response magnitudes of the filters $\mathcal{P}$ and $\mathcal{Q}$ for Floyd and Jarvis respectively; these responses are obtained by averaging over an annulus of constant radius in the 2-D frequency domain [1]. In [35], Kite et al. further generalized the linear model of (3.1) by using different gains K_s and K_n in the signal transfer function $P(f_1, f_2)$ and the noise transfer function $Q(f_1, f_2)$ respectively: $P(f_1, f_2) := \frac{K_s}{1+(K_s-1)H(f_1,f_2)}$ and $Q := \frac{1-H(f_1,f_2)}{1+(K_n-1)H(f_1,f_2)}$. In this chapter, we assume a single gain factor K for both the signal and noise transfer functions as proposed in [34].

3.3 Inverse Halftoning $\approx$ Deconvolution

Given a halftone $y(n_1, n_2)$ (see Figure 3.4(a)), inverse halftoning aims to estimate the gray-scale image $x(n_1, n_2)$. In the classical deconvolution problem, given the blurred and noisy observation



(a) Floyd (b) Jarvis

Figure 3.6: *Plots (a) and (b) respectively illustrate the radially-averaged frequency response magnitudes $|P(f_1, f_2)|$ (solid line) and $|Q(f_1, f_2)|$ (dotted line) for Floyd and Jarvis. The high-pass $|P(f_1, f_2)|$ explains the sharpened edges, while the high-pass $|Q(f_1, f_2)|$ explains the "blue noise" behavior in the halftones.*

$y(n_1, n_2)$ as in (3.1) with known LTI filters responses $p(n_1, n_2)$ and $q(n_1, n_2)$, we seek to estimate $x(n_1, n_2)$. Thus, under the linear model of [34, 35], inverse halftoning can be posed as a deconvolution problem.

3.3.1 Deconvolution

Due to the interplay between inverse halftoning and deconvolution, the well-studied deconvolution literature [4, 7, 13] can be exploited to understand inverse halftoning as well. Deconvolution algorithms conceptually consist of the following steps:

1. *Operator inversion*

   Invert the convolution operator $\mathcal{P}$ to obtain a noisy estimate $\widetilde{x}(n_1, n_2)$ of the input signal[1]

$$\widetilde{x}(n_1, n_2) := \mathcal{P}^{-1} y(n_1, n_2) = x(n_1, n_2) + \mathcal{P}^{-1}\mathcal{Q}\gamma(n_1, n_2). \tag{3.4}$$

2. *Transform-domain shrinkage*

   Attenuate the colored noise $\mathcal{P}^{-1}\mathcal{Q}\gamma(n_1, n_2)$ by expressing $\widetilde{x}(n_1, n_2)$ in terms of a chosen orthonormal basis $\{b_k\}_{k=0}^{N-1}$ and shrinking the k-th component with a scalar λ_k, $0 \leq \lambda_k \leq 1$ [9]

$$\widehat{x}_\lambda := \sum_k \langle \widetilde{x}, b_k \rangle \lambda_k \, b_k = \sum_k \left(\langle x, b_k \rangle + \langle \mathcal{P}^{-1}\mathcal{Q}\gamma, b_k \rangle \right) \lambda_k \, b_k \tag{3.5}$$

   to obtain the deconvolution estimate $\widehat{x}_\lambda$.

The $\sum_k \langle x, b_k \rangle \lambda_k \, b_k$ in (3.5) denotes the *retained part* of the signal $x(n_1, n_2)$ that shrinkage preserves from (3.4), while $\sum_k \langle \mathcal{P}^{-1}\mathcal{Q}\gamma, b_k \rangle \lambda_k \, b_k$ denotes the *leaked part* of the colored noise $\mathcal{P}^{-1}\mathcal{Q}\gamma(n_1, n_2)$ that shrinkage fails to attenuate. Clearly, we should set $\lambda_k \approx 1$ if the variance $\sigma_k^2 := \mathbb{E}(|\langle \mathcal{P}^{-1}\mathcal{Q}\gamma, b_k \rangle|^2)$ of the k-th colored noise component is small relative to the energy $|\langle x, b_k \rangle|^2$ of the corresponding signal component and set $\lambda_k \approx 0$ otherwise. The shrinkage by λ_k can also be interpreted as a form of *regularization* for the deconvolution inverse problem [7].

The choice of transform domain to perform the shrinkage in deconvolution (see Step 2 above) critically influences the MSE of the deconvolution estimate. An important fact is that for a given transform domain, even with the best possible λ_k's, the estimate $\widehat{x}_\lambda$'s MSE is lower-bounded within

[1]For non-invertible $\mathcal{P}$, we replace $\mathcal{P}^{-1}$ by its pseudo-inverse and $x(n_1, n_2)$ by its orthogonal projection onto the range of $\mathcal{P}$ in (3.4).

a factor of 2 by [8, 10, 11]

$$\sum_k \min\left(|\langle x, b_k\rangle|^2, \sigma_k^2\right). \tag{3.6}$$

From (3.6), $\widetilde{x}_\lambda$ can have small MSE only when most of the signal energy ($= \sum_k |\langle x, b_k\rangle|^2$) and colored noise energy ($= \sum_k \sigma_k^2$) is captured by just a few transform-domain coefficients — we term such a representation *economical* — and when the energy-capturing coefficients for the signal and noise are different. Otherwise, the $\widetilde{x}_\lambda$ is either excessively noisy due to leaked noise components or distorted due to lost signal components.

Traditionally, the Fourier domain (with sinusoidal b_k) is used to estimate $x(n_1, n_2)$ from $\widetilde{x}(n_1, n_2)$ because it represents the colored noise $\mathcal{P}^{-1}\mathcal{Q}\gamma(n_1, n_2)$ in (3.4) most economically. That is, among orthonormal transforms, the Fourier transform captures the maximum colored noise energy using a fixed number of coefficients because it diagonalizes convolution operators [26]. Fourier-based deconvolution performs both the operator inversion and the shrinkage simultaneously in the Fourier domain as

$$\widehat{X}_{\lambda^{\mathrm{f}}} := Y(f_1, f_2)\ \frac{1}{P(f_1, f_2)}\ \lambda^{\mathrm{f}}_{f_1, f_2} \tag{3.7}$$

with shrinkage

$$\lambda^{\mathrm{f}}_{f_1, f_2} := \frac{|P(f_1, f_2)|^2}{|P(f_1, f_2)|^2\ +\ \Upsilon(f_1, f_2)|Q(f_1, f_2)|^2} \tag{3.8}$$

at the different frequencies. The $Y(f_1, f_2)$ and $\widehat{X}_{\lambda^{\mathrm{f}}}(f_1, f_2)$ denote the 2-D Fourier transforms of $y(n_1, n_2)$ and the deconvolution estimate $\widehat{x}_{\lambda^{\mathrm{f}}}(n_1, n_2)$ respectively. The $\Upsilon(f_1, f_2)$ in (3.8) is called the *regularization term* and is set appropriately during deconvolution [7]. For example, using the signal to noise ratio to set $\Upsilon(f_1, f_2) = \frac{\mathbb{E}(|\Gamma(f_1, f_2)|^2)}{|X(f_1, f_2)|^2}$ in (3.7) yields the Wiener deconvolution esti-

mate [25]; the $\Gamma(f_1, f_2)$ and $X(f_1, f_2)$ denote the respective Fourier transforms of $\gamma(n_1, n_2)$ and $x(n_1, n_2)$. The $\frac{1}{P(f_1,f_2)}\ \lambda^{\mathrm{f}}_{f_1,f_2}$ in (3.7) constitutes the frequency response of the so-called *deconvolution filter*.

Fourier-based deconvolution suffers from the drawback that its estimates for images with sharp edges are unsatisfactory either due to excessive noise or due to distortions such as blurring or ringing. Since the energy due to the edge discontinuities spreads over many image Fourier coefficients, as dictated by the MSE bound in (3.6), any estimate obtained via Fourier-domain shrinkage suffers from a large MSE.

3.3.2 Inverse halftoning via Gaussian low-pass filtering (GLPF)

Conventionally, inverse halftoning has been performed using a finite impulse response (FIR) Gaussian filter with coefficients $g(n_1, n_2) \propto \exp[-(n_1^2 + n_2^2)/(2\sigma_g^2)]$, where $-4 \leq n_1, n_2 \leq 4$, and σ_g determines the bandwidth [36]. We can interpret inverse halftoning using GLPF as a naive Fourier-domain deconvolution approach to inverse halftoning. This is substantiated by our observation that the deconvolution filter $\frac{1}{P(f_1,f_2)}\ \lambda^{\mathrm{f}}_{f_1,f_2}$ (see (3.7) and (3.8)) constructed with the linear model filters $\mathcal{P}$ and $\mathcal{Q}$ for Floyd and with regularization $\Upsilon(f_1, f_2) \propto \frac{1}{f_1^2+f_2^2}$ has a frequency response that closely matches the frequency response of the GLPF (see Figure 3.7) [36]. The corresponding inverse halftone estimates obtained using simulations are also nearly identical. Predictably, GLPF estimates suffer from the same drawbacks that afflict any Fourier-based deconvolution estimate — excessive noise (when σ_g is small) or significant blurring (when σ_g is large). Exploiting the insights provided by the deconvolution perspective, we can infer that unsatisfactory GLPF estimates result because the Fourier domain does not economically represent images with edges.

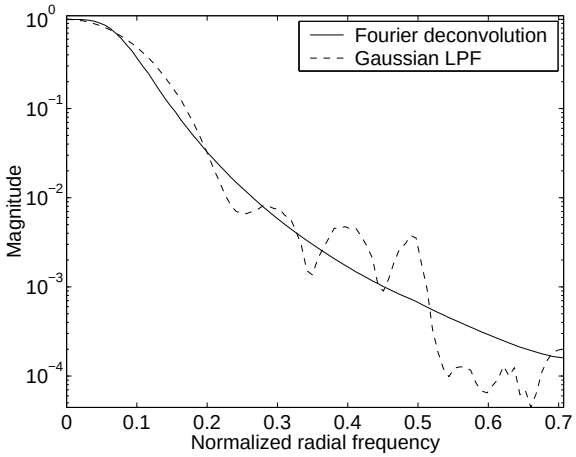


Figure 3.7: *Comparison of radially-averaged frequency response magnitudes of the FIR GLPF (dashed line) used for inverse halftoning in [36] with the response of the deconvolution filter (solid line) constructed with filters $\mathcal{P}$ and $\mathcal{Q}$ for Floyd and with $\Upsilon(f_1, f_2) \propto \frac{1}{f_1^2+f_2^2}$ (see (3.7) and (3.8)). Ripples in the GLPF frequency response result because the filter is truncated in space.*

3.4 Wavelet-based Inverse Halftoning Via Deconvolution (WInHD)

To simultaneously exploit the economy of wavelet representations (refer Appendix A for an overview of wavelets) and our realization about the interplay between inverse halftoning and deconvolution, we propose the WInHD algorithm [40]. WInHD adopts the wavelet-based deconvolution approach of [8] to perform inverse halftoning.

3.4.1 WInHD algorithm

WInHD employs scalar shrinkage in the wavelet domain to perform inverse halftoning as follows (see Figure 3.2):.

1. *Operator inversion*

   As in (3.4), obtain a noisy estimate $\widetilde{x}(n_1, n_2)$ of the input image by inverting $\mathcal{P}$.

2. *Wavelet-domain shrinkage*

   Employ scalar shrinkage in the wavelet domain to attenuate the noise $\mathcal{P}^{-1}\mathcal{Q}\gamma(n_1, n_2)$ in $\widetilde{x}(n_1, n_2)$ and obtain the WInHD estimate $\widehat{x}_{\lambda^{\mathrm{w}}}(n_1, n_2)$ as follows:

   (a) Compute the DWT of the noisy $\widetilde{x}$ to obtain $\widetilde{w}_{j,\ell} := \langle \widetilde{x}\ , \psi_{j,\ell} \rangle$.

   (b) Shrink the noisy $\widetilde{w}_{j,\ell}$ with scalars $\lambda^{\mathrm{w}}_{j,\ell}$ (using (A.7) or (A.8)) to obtain $\widehat{w}_{j,\ell;\lambda^{\mathrm{w}}} := \widetilde{w}_{j,\ell}\, \lambda^{\mathrm{w}}_{j,\ell}$. The colored noise variance at each scale j determining the $\lambda^{\mathrm{w}}_{j,\ell}$ is given by $\sigma_j^2 := \mathbb{E}\left(|\langle \mathcal{P}^{-1}\, \mathcal{Q}\, \gamma\ , \psi_{j,\ell} \rangle|^2 \right)$.

   (c) Compute the inverse DWT with the shrunk $\widehat{w}_{j,\ell;\lambda^{\mathrm{w}}}$ to obtain the WInHD estimate $\widehat{x}_{\lambda^{\mathrm{w}}}(n_1, n_2)$.

For error diffusion systems, $\mathcal{P}^{-1}$ is an FIR filter. Hence, the noisy estimate $\widetilde{x}(n_1, n_2)$ obtained in Step 1 using $\mathcal{P}^{-1}$ is well-defined. The subsequent wavelet-domain shrinkage in Step 2 effectively extracts the few dominant wavelet components of the desired gray-scale image $x(n_1, n_2)$ from the noisy $\widetilde{x}(n_1, n_2)$ because the residual noise $\mathcal{P}^{-1}\, \mathcal{Q}\, \gamma(n_1, n_2)$ corrupting the wavelet components is not excessive.

WInHD can be easily adapted to different error diffusion techniques simply by choosing the gain K recommended by [34] and the error filter response $h(n_1, n_2)$ for the target error diffusion technique. K and $h(n_1, n_2)$ determine the filters $\mathcal{P}$ and $\mathcal{Q}$ (see (3.2) and (3.3)) required to perform WInHD. In contrast, the gradient-based inverse halftoning method [41] adapts to a given error diffusion technique by employing a set of smoothing filters that need to be designed carefully.

3.4.2 Asymptotic performance of WInHD

With advances in technology, the spatial resolution of digital images (controlled by the number of pixels N) has been steadily increasing. Hence any inverse halftoning algorithm should not only perform well at a fixed resolution but should also guarantee good performances at higher spatial resolutions. In this section, under some assumed conditions, we deduce the rate at which the per-pixel MSE for WInHD decays as number of pixels $N \to \infty$.

Invoking established results in wavelet-based image estimation in Gaussian noise, we prove the following proposition in Appendix F about the asymptotic performance of WInHD.

Proposition 4 *Let $x(n_1, n_2)$ be a N-pixel gray-scale image obtained as in (A.3) by uniformly sampling a continuous-space image $x(t_1, t_2) \in B^s_{p,q}$ with $t_1, t_2 \in [0, 1)$, $s > \frac{2}{p} - 1$, and $1 < p, q, < \infty$. Let $p(n_1, n_2)$ and $q(n_1, n_2)$ denote known filter impulse responses that are invariant with N and with Fourier transform magnitudes $|P(f_1, f_2)| \geq \epsilon > 0$ and $|Q(f_1, f_2)| < \infty$. Let $y(n_1, n_2)$ be observations obtained as in (3.1) with $\gamma(n_1, n_2)$ zero-mean AWGN samples with variance σ^2. Then, the per-pixel MSE of the WInHD estimate $\widehat{x}(n_1, n_2)$ obtained from $y(n_1, n_2)$ using hard thresholding behaves as*

$$\frac{1}{N}\mathbb{E}\left(\sum_{n_1,n_2} |\widehat{x}(n_1, n_2) - x(n_1, n_2)|^2\right) \leq C\, N^{\frac{-s}{s+1}}, \qquad N \to \infty, \tag{3.9}$$

with constant $C > 0$ independent of N.

The above proposition affirms that the per-pixel MSE of the WInHD estimate decays as $N^{\frac{-s}{s+1}}$ with increasing spatial resolution ($N \to \infty$) under the mild assumptions discussed below.

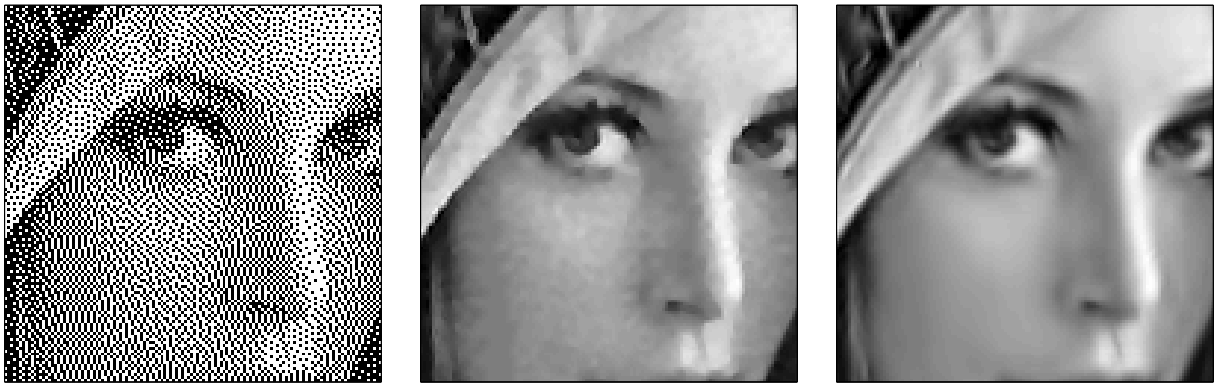
The central assumption in Proposition 4 is that the linear model (3.1) for error diffusion

is accurate. This is well-substantiated in [34, 35]. The conditions $|P(f_1, f_2)| \geq \epsilon > 0$ and $|Q(f_1, f_2)| < \infty$ respectively ensure that $\mathcal{P}$ is invertible and that the variance of the colored noise $Q\gamma(n_1, n_2)$ is bounded. We have verified that for common error diffusion halftoning techniques such as Floyd and Jarvis, the filters $\mathcal{P}$ and $\mathcal{Q}$ recommended by the linear model of Kite et al. satisfy these conditions (see Figure 3.6). The final assumption is that the noise $\gamma(n_1, n_2)$ is Gaussian; this is required to invoke the established results on the asymptotics of wavelet-based estimators [22]. However, recently, wavelet-domain thresholding has been shown to be optimal for many other noise distributions as well [42, 43]. Hence the noise Gaussianity assumption in Proposition 4 could be relaxed.

Often, gray-scale digital images are corrupted with some noise before being subjected to halftoning. For example, sensor noise corrupts images captured by charged coupled device (CCD) based digital cameras. In such cases as well, WInHD can effectively estimate the noise-free gray-scale image with an MSE decay rate of $N^{\frac{-s}{s+1}}$ as in Proposition 4. Further, WInHD's MSE decay rate can be shown to be optimal. The noise-free gray-scale image and resulting halftone can be related using the linear model of [34, 35] as

$$y(n_1, n_2) = \mathcal{P}\left[x(n_1, n_2) + \beta(n_1, n_2)\right] + \mathcal{Q}\gamma(n_1, n_2), \tag{3.10}$$

with $\beta(n_1, n_2)$ denoting the noise corrupting the gray-scale image before halftoning. If the $\beta(n_1, n_2)$ is AWGN with non-zero variance, then we can easily infer that the residual noise after inverting $\mathcal{P}$ in Step 1 of WInHD can be analyzed like white noise because its variance is bounded but non-zero [8]. Hence we can invoke well-established results on the performance of wavelet-based signal estimation in the presence of white noise [22, 23, 44] to conclude that no estimator



(a) Floyd halftone (b) Gradient estimate (c) WInHD estimate

Figure 3.8: *Close-ups (128 $\times$ 128 pixels) of (a) Floyd halftone, (b) Gradient estimate [41], and (c) WInHD estimate.*

can achieve a better error decay rate than WInHD for every Besov space image. Thus, WInHD is an optimal estimator for inverse halftoning error-diffused halftones of noisy images.

3.5 Results

We illustrate WInHD's performance using 512×512-pixel *Lena* and *Peppers* test images halftoned using the Floyd algorithm [32] (see Figure 3.3 and 3.8). All WInHD estimates and software are available at `www.dsp.rice.edu/software`. We set the gain $K = 2.03$, as calculated for Floyd in [34, 35], and use the Floyd error filter response $h(n_1, n_2)$ (see Figure 3.5) to characterize the impulse responses $p(n_1, n_2)$ and $q(n_1, n_2)$. Inverting the operator $\mathcal{P}$ (Step 2) requires $O(N)$ operations and memory for a N-pixel image since $\mathcal{P}^{-1}$ is FIR. To perform the wavelet-domain shrinkage (Step 2), we choose the WWF because it yields better estimates compared to schemes such as hard thresholding.

Estimates obtained by shrinking DWT coefficients are not shift-invariant; that is, translations

of $y(n_1, n_2)$ will result in different estimates. Hence, we exploit the *complex wavelet transform* (CWT) instead of the usual DWT to perform the WWF. The CWT expands images in terms of shifted and dilated versions of *complex-valued* basis functions instead of the real-valued basis functions used by the DWT [29, 30]; the expansion coefficients are also complex-valued. Wavelet-domain shrinkage using WWF on the CWT coefficient magnitudes yields significantly improved near shift-invariant estimates with just $O(N)$ operations and memory. (The redundant, shift-invariant DWT can also be used instead of the CWT to obtain shift-invariant estimates [10], but the resulting WInHD algorithm requires $O(N \log N)$ operations and memory.) The standard deviation of the noise $\gamma(n_1, n_2)$, which is required during wavelet shrinkage, is calculated using the standard deviation of $y(n_1, n_2)$'s finest scale CWT coefficients.

Figures 3.3 and 3.8 compares the WInHD estimate with the multiscale gradient-based estimate [41] for the Lena image. We quantify the WInHD's performance by measuring the peak signal-to-noise ratio PSNR $:= 20 \log_{10} \frac{512 \times 255}{\|\widehat{x} - x\|_2}$ (for 512×512-pixel images with gray levels $\in [0, 1, \ldots, 255]$) with $\widehat{x}(n_1, n_2)$ the estimate. Table 3.1 summarizes the PSNR performance and computational complexity of WInHD compared to published results for inverse halftoning with Gaussian filtering [36], kernel estimation [31], gradient estimation [41], and wavelet denoising with edge-detection [39]. We can see that WInHD is competitive with the best published results.

The WInHD estimate yields competitive visual performance as well. We quantify visual performance using two metrics: weighted SNR (WSNR) [45, 46] and the *Universal Image Quality Index* (UIQI) [47]. Both metrics were computed using the halftoning toolbox of [48]. The WSNR is obtained by weighting the SNR in the frequency domain according to a linear model of the human visual system [45, 46]. The WSNR numbers in Table 3.2 are calculated at a spatial Nyquist

Table 3.1: *PSNR and computational complexity of inverse halftoning algorithms (N pixels).*

Inverse halftoning algorithm	PSNR (dB)		Computational complexity
	Lena	*Peppers*	
Gaussian [36]	28.6	27.6	$O(N)$
Kernel [31]	32.0	30.2	$O(N)$
Gradient [41]	31.3	31.4	$O(N)$
Wavelet denoising [39]	31.7	30.7	$O(N \log N)$
WInHD	32.1	31.2	$O(N)$

Table 3.2: *Visual metrics for inverse halftoned estimates of Lena.*

Algorithm	WSNR (dB)	UIQI
Gradient [41]	34.0	0.62
WInHD	35.9	0.62

frequency of 60 cycles/degree. The recently proposed UIQI metric of Wang et al. effectively models image distortion with a combination of correlation loss, luminance distortion, and contrast distortion [47]; UIQI $\in [-1, 1]$ with larger values implying better image quality. For the Lena image, WInHD's performance in terms of both the visual metrics is competitive with the gradient estimate's performance (see Table 3.2).

Chapter 4

JPEG Compression History Estimation (CHEst) for Color Images

We live in a rainbow of chaos.

–Paul Cezanne

4.1 Introduction

A digital color image is a collection of pixels, with each pixel typically a 3-dimensional (3-D) color vector. The vector entries specify the color of the pixel with respect to a chosen color space; for example, *RGB*, *YCbCr*, et cetera. JPEG is a commonly used standard to compress digital color images [3]. It achieves compression by quantizing the discrete cosine transform (DCT) coefficients of the image's three color planes; see Fig. 4.1 for an overview of JPEG. However, the various settings used during JPEG compression and decompression are not standardized [3]. The following JPEG settings can be chosen by the user such as an imaging device: (1) the color space used to independently compress the image's three color planes; (2) the subsampling employed on each color plane during compression and the complementary interpolation used during decompression; and (3) the quantization table used to compress each color plane. We refer to the settings used during JPEG operations as the image's JPEG *compression history*.

An image's compression history is often not directly available from its current representation.

For example, JPEG images are often imported into Microsoft Powerpoint or Word documents using graphics programs such as Microsoft Clip Gallery and then stored internally using an de-compressed format. JPEG images are also routinely converted to lossless-compression formats such as bitmap (BMP) format (say, to create a background image for Windows or to feed a printing driver) or Tagged Image File Format (TIFF). In such cases, the JPEG compression settings are discarded after decompression.

The compression history, if available, can be used for a variety of applications. A BMP or TIFF image's file-size is significantly larger than the previous JPEG file. The JPEG compression history enables us to effectively recompress such BMP and TIFF images; JPEG-compressing the image with previous JPEG settings yields significant file-size reduction without introducing any additional distortion. The JPEG compression history can also be used by "smart" print servers to reduce artifacts, such as blocking due to previous JPEG compression, from received BMP images. To alleviate such artifacts by adapting techniques described in [49–55], the print server would need the image's JPEG compression history. An image's JPEG compression history can also potentially be used as an authentication feature, for covert messaging, or to uncover the compression settings used inside digital cameras. Hence, the problem of Compression History Estimation (CHEst) is useful.

The CHEst problem is relatively unexplored. Fan and de Queiroz have proposed a statistical framework to perform CHEst for gray-scale images [56]; for a gray-scale image, the compression history comprises only the quantization table employed during previous JPEG operations. However, CHEst for color images remains unsolved. In this chapter, we propose new frameworks to perform CHEst for color images.

We first derive a statistical framework to perform CHEst. We realize that due to JPEG's quantization operation, the DCT coefficient histograms of previously JPEG-compressed images exhibit near-periodic structure. We statistically characterize this near-periodicity for a single color plane. The resulting framework can be exploited to estimate a gray-scale image's compression history, namely, its quantization table. We extend the statistical framework to color images and design a dictionary-based CHEst algorithm that provides the *maximum a priori* (MAP) estimate of a color image's compression history

$$\left\{\widehat{G}, \widehat{S}, \widehat{Q}\right\} = \arg\max_{G,S,Q} P\left(\text{Image}, G, S, Q\right), \tag{4.1}$$

with $\widehat{G}$, $\widehat{S}$, $\widehat{Q}$ denoting the estimated compression color space, the subsampling and associated interpolation, and the quantization tables.

For the case when the transform from the color space used by JPEG to perform quantization to the color space of the image's current representation is affine and when no subsampling is employed during JPEG compression, we develop a novel, blind, lattice-based CHEst algorithm. In this case, we make a fundamental observation that the DCT coefficients of the observed image closely conform to a 3-D parallelepiped lattice structure determined by the affine color transform. During our quest to exploit the inherent lattice structure, we discover a fundamentally new contribution to the theory of lattices. We derive the geometric conditions under which a set of lattice basis vectors contains the shortest non-zero lattice vector. Further, we also discover conditions to characterize the uniqueness of such basis vector sets. By exploiting our new-found insights and via novel applications of existing lattice algorithms, we estimate the color image's compression history, namely, the affine color transform and the quantization tables.

Our proposed CHEst algorithms demonstrate excellent performance in practice. Further, we verify that CHEst allows us to recompress an image with minimal distortion (large signal-to-noise-ratio (SNR)) and simultaneously achieve a small file-size.

The rest of this chapter is organized as follows. We first provide a brief overview of color transforms and JPEG in Sections 4.2 and 4.3. We derive the statistical framework for CHEst for gray-scale images in Section 4.4 and extend this framework to design a dictionary-based CHEst algorithm for color images in Section 4.5. In Section 4.6, we present our new contributions to the theory of lattices and then design a lattice-based CHEst algorithm for cases when the compression color space is related to the current color space by an affine transformation and demonstrate the algorithm's estimation performance using simulations. In Section 4.7, we demonstrate the utility of CHEst in JPEG recompression.

4.2 Color Spaces and Transforms

Color perception is a sensation produced when incident light excites the receptors in the human retina. Color can be described by specifying the spectral power distribution of the light. Such a description is highly redundant because the human retina has only three types of receptors that influence color perception.[1] Consequently, three numerical components are sufficient to describe a color; this is termed the *trichromatic theory* [57].

Based on the trichromatic theory, digital color imaging devices use three parameters to specify any color; the three parameters can be viewed as a 3-D vector. The *color space* is the reference coordinate system with respect to which the 3-D vector describes color. There exist many different

[1] A fourth type of receptor is also present in the retina, but it does not affect color perception since it is effective only at extremely low light levels

coordinate systems or color spaces according to which a color can be specified. For example, the Commission Internationale de L'Éclairage (CIE) defined the CIE XYZ color space to specify all visible colors using positive X, Y, and Z values [57, 58]. Other examples include different varieties of *RGB* (Red (*R*), Green (*G*), and Blue (*B*)) and *YCbCr* (luminance *Y*, and chrominances *Cb* and *Cr*) color spaces. These color spaces are related to each other and to reference color spaces such as the CIE XYZ via linear or non-linear color transformations. For example, the popular Independent JPEG Group (IJG) JPEG implementation [59] converts the $0-255$-valued *R*, *G*, and *B* components of a digital color image to $0-255$-valued *Y*, *Cb*, and *Cr* components using the following transformation

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.558 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}. \tag{4.2}$$

The resulting *YCbCr* space is also referred to as the *ITU.BT-601 YCbCr* space. The inverse color transformation from the *ITU.BT-601 YCbCr* space to the *RGB* space is given by

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 1.40 \\ 1.0 & -0.344 & -0.714 \\ 1.0 & 1.77 & 0.0 \end{bmatrix} \left(\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \right). \tag{4.3}$$

The transforms in both (4.2) and (4.3) are *affine*; we will henceforth refer to the 3×3 matrix and the 3×1 shift (for example, $[0\ 128\ 128]^T$ in (4.3)) as the affine transform's *linear component* and the *additive component* respectively.

Later in this chapter, we will invoke a variety of color spaces that are inter-related by affine or non-linear transforms. We refer the reader to [57, 58] for good tutorials on color and for overviews of the different color spaces and transforms.

4.3 JPEG Overview

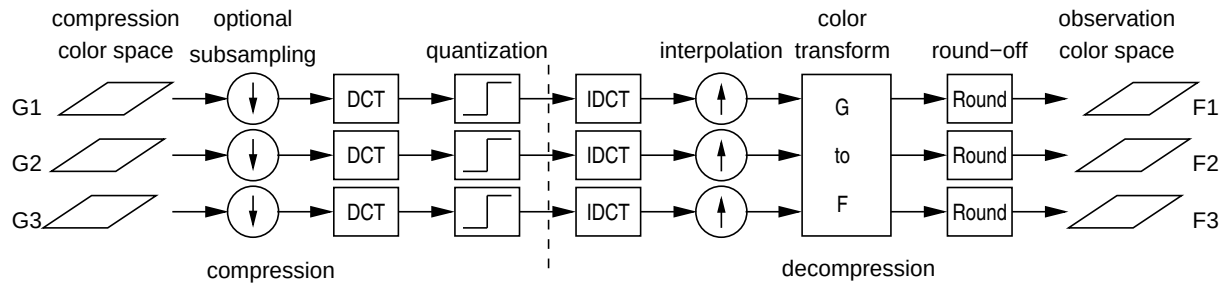


Figure 4.1: *Overview of JPEG compression and decompression.*

In this section, we review JPEG compression and decompression [3]. Consider a color image that is currently represented in the F color space (see Fig. 4.1); $F1$, $F2$, and $F3$ denote the three color planes. We refer to the F space as the *observation color space*. Assume that the image was previously JPEG-compressed in the G color space—termed *compression color space*. JPEG compression performs the following operations independently on each color plane $G1$, $G2$, and $G3$ in the G space:

1. Optionally downsample each color plane (for example, retain alternate pixels to downsample by a factor of two); this process is termed *subsampling*.

2. Split each color plane into 8×8 blocks. Take the DCT of each block.

3. Quantize the DCT coefficients at each frequency to the closest integer multiple of the quantization step-size corresponding to that frequency. For example, if X denotes an arbitrary

DCT coefficient and q denote the quantization step-size for the corresponding DCT frequency, then the quantized DCT coefficient $\overline{X}_q$ is obtained by

$$\overline{X}_q := \text{round}\left(\frac{X}{q}\right) q. \tag{4.4}$$

See Fig. 4.2 for examples of quantization tables; each entry in the 8×8 quantization table is the quantization step-size for an 8×8 image block's corresponding DCT coefficient.

JPEG decompression performs the following operations:

1. Take the inverse DCTs of the 8×8 blocks of quantized coefficients.

2. Interpolate the downsampled color planes by repetition followed by optional spatial smoothing with a low-pass filter. The popular IJG JPEG implementation [59] uses a $\frac{1}{4} \times [1\ 2\ 1]$ impulse response filter to smooth in the horizontal and vertical directions.

3. Transform the decompressed image to the desired color space F using the appropriate G to F transformation.

4. Round-off resulting pixel values to the nearest integer so that they lie in the 0–255 range.

4.4 CHEst for Gray-Scale Images

For gray-scale images, JPEG compression and decompression replicates the steps outlined in Section 4.3 for a single color plane but without subsampling and interpolation. Due to JPEG's quantization operations, the discrete cosine transform (DCT) coefficient histograms of previously JPEG-compressed gray-scale images exhibit a near-periodic structure with the period determined by the

10	7	6	10	14	24	31	37
7	7	8	11	16	35	36	33
8	8	10	14	24	34	41	34
8	10	13	17	31	52	48	37
11	13	22	34	41	65	62	46
14	21	33	38	49	62	68	55
29	38	47	52	62	73	72	61
43	55	57	59	67	60	62	59

Quantization table 1

10	11	14	28	59	59	59	59
11	13	16	40	59	59	59	59
14	16	34	59	59	59	59	59
28	40	59	59	59	59	59	59
59	59	59	59	59	59	59	59
59	59	59	59	59	59	59	59
59	59	59	59	59	59	59	59
59	59	59	59	59	59	59	59

Quantization table 2

Figure 4.2: *Example of JPEG quantization tables.*

quantization step-size. In this section, we derive a statistical framework, which characterizes the near-periodic structure, to estimate the quantization table.

4.4.1 Statistical framework

An arbitrary DCT coefficient $\widetilde{X}$ of a previously JPEG-compressed gray-scale image can be obtained by adding to the corresponding quantized coefficient $\overline{X}_q$ (see (4.4)) a round-off error term Γ

$$\widetilde{X} = \overline{X}_q + \Gamma. \tag{4.5}$$

As described in [56], we can model Γ using a truncated Gaussian distribution

$$P\left(\Gamma = t\right) = \frac{1}{\Upsilon} \exp\left(-\frac{t^2}{2\sigma^2}\right), \qquad \text{for } t \in [-\zeta, \zeta], \tag{4.6}$$

with σ^2 the variance of the Gaussian, $[-\zeta, \zeta]$ the support of the truncated Gaussian, and Υ the normalizing constant. Further, based on studies in [3, 60], we model the DCT coefficients using a

zero-mean Laplacian distribution.[2]

$$P(X = t) = \frac{\lambda}{2} \exp\left(-\lambda|t|\right). \tag{4.7}$$

We have assumed that the parameter λ is known; in practice, we estimate λ from the previously compressed image for each DCT frequency as described later in this section. From (4.7), we have

$$L_\lambda(kq) := P\big(\overline{X}_q = kq \mid q,\ k \in \mathbb{Z}\big) = \int_{(k-0.5)q}^{(k+0.5)q} \frac{\lambda}{2} \exp\left(-\lambda|\tau|\right)\, d\tau. \tag{4.8}$$

Hence,

$$P\big(\overline{X}_q = t \mid q\big) = \sum_{k\in\mathbb{Z}} \delta\left(t - kq\right) L_\lambda(kq). \tag{4.9}$$

Now, assuming that the round-off error Γ is independent of X and q, the distribution of $\widetilde{X}$ is obtained by convolving the distributions for $\overline{X}$ and Γ (see Fig. 4.3). That is,

$$P\Big(\widetilde{X} = t \mid q\Big) = \int P\big(\overline{X}_q = \tau \mid q\big)\, P(\Gamma = t - \tau)\, d\tau \tag{4.10}$$

$$= \begin{cases} \sum_{k\in\mathbb{Z}} \frac{1}{\Upsilon} \exp\left(-\frac{|t-kq|^2}{2\sigma^2}\right) L_\lambda(kq), \\ \qquad\qquad \text{for } |t - kq| \in [-\zeta, \zeta], \\ 0, \qquad\quad \text{otherwise.} \end{cases} \tag{4.11}$$

Given a set $\mathcal{D}$ of DCT coefficients at a particular frequency that are obtained from a previously compressed image, we can obtain the MAP estimate $\widehat{q}$ of the quantization step used during previous

[2]DC components are typically modeled using Gaussian distributions with non-zero mean. To avoid the errors associated with estimating the Gaussian's mean and for simplicity, we assume that the DC coefficient can also be modeled using a zero-mean Laplacian distribution with zero mean.

compression assuming the DCT coefficients are independent as

$$\widehat{q} = \arg\max_{q\in\mathbb{Z}^+} P(\mathcal{D}, q) \tag{4.12}$$

$$= \arg\max_{q\in\mathbb{Z}^+} \left(\prod_{\widetilde{X}\in\mathcal{D}} P(\widetilde{X} \mid q) P(q) \right), \tag{4.13}$$

where $P(q)$ denotes the prior on q.

4.4.2 Algorithm steps

Based on the statistical framework derived in the previous section, we can estimate the quantization step-sizes using the following steps:

1. Compute set of the desired frequency DCT coefficients $\mathcal{D}$ from the previously compressed image.

2. Estimate the parameter λ from the observations as

$$\lambda = \frac{N}{\sum_{\widetilde{X}\in\mathcal{D}} |\widetilde{X}|},$$

   with N the number of coefficients in the set $\mathcal{D}$.

3. Assuming all quantization step-sizes are equally likely, use (4.11) with suitable parameters σ^2 and ζ to estimate

$$\widehat{q} = \arg\max_{q\in\mathbb{Z}^+} \left(\prod_{\widetilde{X}\in\mathcal{D}} P(\widetilde{X} \mid q) \right). \tag{4.14}$$

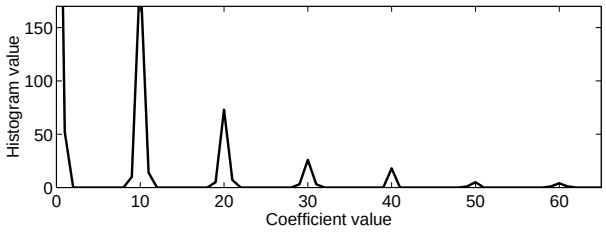


Figure 4.3: *Histogram of quantized DCT coefficients. The DCT coefficients from DCT frequency (4,4) of the gray-scale Lenna image were subjected to quantization with step-size* $q = 10$ *during JPEG compression and then decompressed. Due to roundoff errors, the DCT coefficients are perturbed from integer multiples of* 10*.*

The above algorithm is a refinement of the technique proposed by Fan and de Queiroz in [56]. While the core ideas remain the same, the final derived equation (4.11) differs because of significant variations in the starting points for the derivation and in the intermediate assumptions. Further, our derivation explicitly accounts for all the normalization constants, thereby allowing us to extend the above approach to estimate the compression history of color images.

4.5 Dictionary-based CHEst for Color Images

In this section, we build on the quantization step-size estimation algorithm for gray-scale images to perform CHEst for color images.

4.5.1 Statistical framework

For color images, in addition to quantization, JPEG performs color transformation and subsampling along with the complementary interpolation. We realize that the DCT coefficient histograms of each color plane exhibit the near-periodic structure of Fig. 4.3 introduced by quantization only when the image is transformed to the original compression color space and the all interpolation effects are inverted. Otherwise, the near-periodic structure is not visible. This realization enables

us to obtain the MAP estimate of a color image's compression history as in (4.1) by extending the statistical framework for gray-scale images a straightforward way. Let $\mathcal{D}_{G,S}$ denote the set of DCT coefficients $\widetilde{X}_{G,S}$ obtained by first transforming the image from F to the G color space representation, then undoing the interpolation S, and finally taking the DCT of the color planes. Then,

$$\begin{aligned}\left\{\widehat{G},\widehat{S},\widehat{Q}\right\} &= \arg\max_{G,S,Q} P\left(\text{Image}|G,S,Q\right)P(G,S,Q)\\ &= \arg\max_{G,S,Q} P(\mathcal{D}_{G,S}|G,S,Q)P(G)P(S)P(Q)\\ &= \arg\max_{G,S,Q} \prod_{\widetilde{X}_{G,S}\in\mathcal{D}_{G,S}} P\left(\widetilde{X}_{G,S}|G,S,Q\right)P(G)P(S)P(Q), \qquad (4.15)\end{aligned}$$

assuming that the G, S, and Q are independent. In (4.15), the conditional probability $P(\widetilde{X}_{G,S}|G,S,Q)$ of the DCT coefficients is set to (4.11), which is a metric for how well the image DCT coefficients conform to a near-periodic structure. Hence, if G, S, and Q were actually employed during the previous JPEG compression, then the histogram of $\mathcal{D}_{G,S}$ exhibits near-periodicity and the associated $P(\widetilde{X}_{G,S}|G,S,Q)$ would be large. Consequently, the MAP estimate would be accurate.

4.5.2 Algorithm steps

Ideally, the MAP estimation of (4.15) would require a search over all G and S. For practical considerations, we constrain our search to a dictionary of commonly employed compression color spaces and interpolations. The steps of our simple dictionary-based CHEst algorithm are as follows:

1. Transform the observed color image to a test color space G.

2. Undo the effects of the test interpolation S. To undo interpolation by simple repetition, simply downsample the color plane. To undo interpolation by repetition and smoothing, first deconvolve the smoothing using a simple Tikhonov-regularized deconvolution filter [24], and then downsample the color plane.

3. Employ the quantization step-size estimation algorithm of Section 4.4 on the coefficients at each DCT frequency and each color plane.

4. Output the color transform and interpolation yielding the maximum conditional probability (see (4.15)) along with the associated quantization tables from Step 3.

The computational complexity of the dictionary-based CHEst algorithm is determined by the size of the image, the number of the test color spaces, and the number of test subsamplings and interpolations. We can easily prune the number of test color spaces and interpolations to reduce the computational complexity by using a small part of the image. The quantization table estimates can be perfected using the entire image after the color space and interpolation is quickly determined.

4.5.3 Dictionary-based CHEst results

Our dictionary-based CHEst algorithm precisely estimates the compression history of a previously JPEG-compressed color image. We demonstrate the performance of our algorithm using the 512×512 *Lenna* color image [61]. We JPEG-compressed *Lenna* in the 8-bit CIELab color space using the sRGB to 8-bit CIELab color transformation [58] and employed $2 \times 2, 1 \times 1, 1 \times 1$ subsampling; that is, the luminance *L* color plane is not downsampled, while the chrominance planes *a* and *b* are downsampled by a factor of 2 in the horizontal and vertical directions. We employed quantization table 1 from Fig. 4.2 for the *L* plane and quantization table 2 for the both the *a* and *b* planes. During

decompression, the *a* and *b* planes are interpolated by first upsampling using repetition and then smoothing in the horizontal and vertical directions using a $\frac{1}{4} \times [1\ 2\ 1]$ impulse response filter.

To solve the CHEst problem, we tested all color transforms from a dictionary consisting of RGB to YCbCr, Computer RGB to ITU.BT-601 YCbCr, Studio RGB to ITU.BT-601 YCbCr, RGB to Kodak PhotoYCC, sRGB to Linear RGB, sRGB to 8-bit CIELab, and sRGB to CMY transforms [58]. For each transform, we considered subsampling factors $2 \times 2, 1 \times 1, 1 \times 1$ (with and without smoothing during interpolation) and $1 \times 1, 1 \times 1, 1 \times 1$.

During the computations of the conditional probabilities (4.15), we assumed that all color transforms and quantization step-sizes are equally likely; that is, $P(G) = P(Q) = 1$. During our experiments, we set $\zeta = 6$ (see (4.11)). To test if a color plane was subsampled by a factor of 2 and then smoothed during interpolation, we set the $\sigma^2 = 0.8$ (see (4.11)) during the quantization table estimation step. To test if no smoothing is employed during interpolation, we set $\sigma^2 = 0.75$, and to test if no subsampling is employed, we reduced the σ^2 to 0.5. Further, we set the prior $P(S) = 0.55$ for the $2\times2, 1\times1, 1\times1$ with smoothing, $P(S) = 0.35$ for the $2\times2, 1\times1, 1\times1$ without smoothing, and $P(S) = 0.1$ for the $1 \times 1, 1 \times 1, 1 \times 1$ subsampling. When larger subsampling factors and smoothing are employed, the DCT coefficients deviate further from their quantized values resulting in relatively lower conditional probabilities. The adapted σ^2 and priors $P(S)$ help level this effect and correctly detect the compression history of the image.

By comparing the logarithms of the conditional probabilities listed in Table 4.1, we precisely identify that the sRGB to 8-bit CIELab color transformation was employed with $2 \times 2, 1 \times 1, 1 \times 1$ subsampling during the previous compression, and smoothing was employed during the decompression; the corresponding conditional probability value (enclosed by a $\square$ in the table) is the

Table 4.1: *Conditional probabilities' logarithms ($\times 10^6$) for different color transforms and interpolations*

Color transform	$1\times1, 1\times1, 1\times1$	$2\times2, 1\times1, 1\times1$ (without smoothing)	$2\times2, 1\times1, 1\times1$ (with smoothing)
RGB to YCbCr	-0.88	-0.88	-0.8
Computer RGB to YCbCr	-0.83	-0.83	-0.75
Studio RGB to YCbCr	-0.88	-0.89	-0.81
RGB to Kodak YCC	-0.79	-0.78	-0.69
sRGB to Linear RGB	-1.5	-1.9	-1.8
sRGB to 8-bit CIELab	-0.73	-0.71	$\boxed{\mathbf{-0.53}}$
sRGB to CMY	-1.5	-1.9	-1.8

largest. Our quantization table estimates are given in Fig. 4.4. (see Fig. 4.2 for the actual tables). An $\times$ indicates that the quantization step-size estimation for the corresponding DCT frequency was not possible because all the DCT coefficients were quantized to zero during compression. Our quantization step-size estimates, especially at the most important low frequencies, is accurate. The estimates for the a and the b planes suffer from seemingly large errors. For example, we incur an error of 52 while estimating an entry in the a plane's quantization table; the actual quantization step-size was 59 but our algorithm's estimate was 7. The error is a result of additional noise (compared to the no subsampling case) introduced by the deconvolution step (algorithm's Step 2)), which is necessary to undo the interpolation . However, the estimation error does not adversely affect applications such as recompression because, in reality, all the DCT coefficients at the corresponding frequency were set to zero during quantization.

4.6 Blind Lattice-based CHEst for Color Images

The efficacy of dictionary-based CHEst approach is limited by the richness of the color space dictionary. If an unknown proprietary color space is used to perform the JPEG compression, then

10	7	6	10	14	24	31	37
7	7	8	11	16	35	36	32
8	8	10	14	24	34	40	33
8	10	13	17	31	51	47	35
11	13	22	34	40	63	×	44
14	21	32	37	47	×	×	51
28	35	44	49	×	×	×	×
×	×	×	×	×	×	×	×

L's table

10	11	14	28	54	9	×	9
11	13	15	36	50	9	7	7
14	15	30	9	9	5	×	7
26	34	48	×	×	×	7	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×

a's table

10	11	14	26	×	7	×	×
11	13	15	35	×	×	×	×
13	15	31	48	×	×	×	×
25	34	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×

b's table

Figure 4.4: *Quantization tables estimates using the dictionary-based CHEst algorithm for the L, a and b color planes. An $\times$ indicates that estimation was not possible because all coefficients were quantized to zero.*

the dictionary-based approach will fail to detect it. In this section, we develop an approach that does not need a dictionary of color spaces, and is hence *blind*, to estimate the compression history if the transform from the compression color space to the current color space is affine and if no subsampling is employed during JPEG compression.

Our fundamental observation is that the DCT coefficients of the observed image closely conform to a 3-D parallelepiped lattice structure determined by the affine color transform. We discover some fascinating, new, and relevant properties of lattices during our theoretical study. We rely on these new insights and novel applications of existing lattice algorithms to exploit the lattice structure of our problem and estimate the color image's compression history.

For the sake of clarity, we first describe the estimation of the image's compression history assuming that no round-off errors are present; that is, assuming that the DCT coefficients exactly conform to a 3-D parallelepiped lattice. We then adjust the estimate to combat the round-off noise.

4.6.1 Ideal lattice structure of DCT coefficients

In the absence of round-off noise, the 3-D vector of DCT coefficients of an previously JPEG-compressed color image conform to a regular parallelepiped lattice structure. This inherent structure is a result of the quantization undergone during previous JPEG compression.

Consider an arbitrary 8×8 color image block that the DCT acts on during JPEG compression in the G color space. Let X_{G1}, X_{G2}, and X_{G3} denote the respective i^{th} frequency DCT coefficients of the $G1$, $G2$, and $G3$ planes in the chosen 8×8 color image block, and let $q_{i,1}$, $q_{i,2}$, and $q_{i,3}$ denote the respective quantization step-sizes. JPEG quantizes the DCT coefficients of the each plane *independently* to $\overline{X}_{q_{i,1}} := \text{round}\left(\frac{X_{G1}}{q_{i,1}}\right) q_{i,1}$, $\overline{X}_{q_{i,2}} := \text{round}\left(\frac{X_{G2}}{q_{i,2}}\right) q_{i,2}$, and $\overline{X}_{q_{i,3}} := \text{round}\left(\frac{X_{G3}}{q_{i,3}}\right) q_{i,3}$ respectively. The 3-D vector of quantized DCT coefficients $\left[\, \overline{X}_{q_{i,1}} \; \overline{X}_{q_{i,2}} \; \overline{X}_{q_{i,3}} \right]^T$, with superscript T denoting matrix transpose, along with all the other vectors of quantized i^{th} DCT frequency coefficients (see Fig. 4.5), lies on a rectangular box lattice (see Fig. 4.6) with edge-lengths equal to the quantization step-sizes.

When an image with previously quantized DCT coefficients is represented in another color space F that is related to G by an affine transformation, then the corresponding 3-D vectors of DCT coefficients in the F space do not lie on a rectangular box lattice, but rather on a *parallelepiped*[3] lattice, assuming that no round-off is performed during JPEG decompression in the G space. The

[3] A solid with six faces, each of which is a parallelogram.

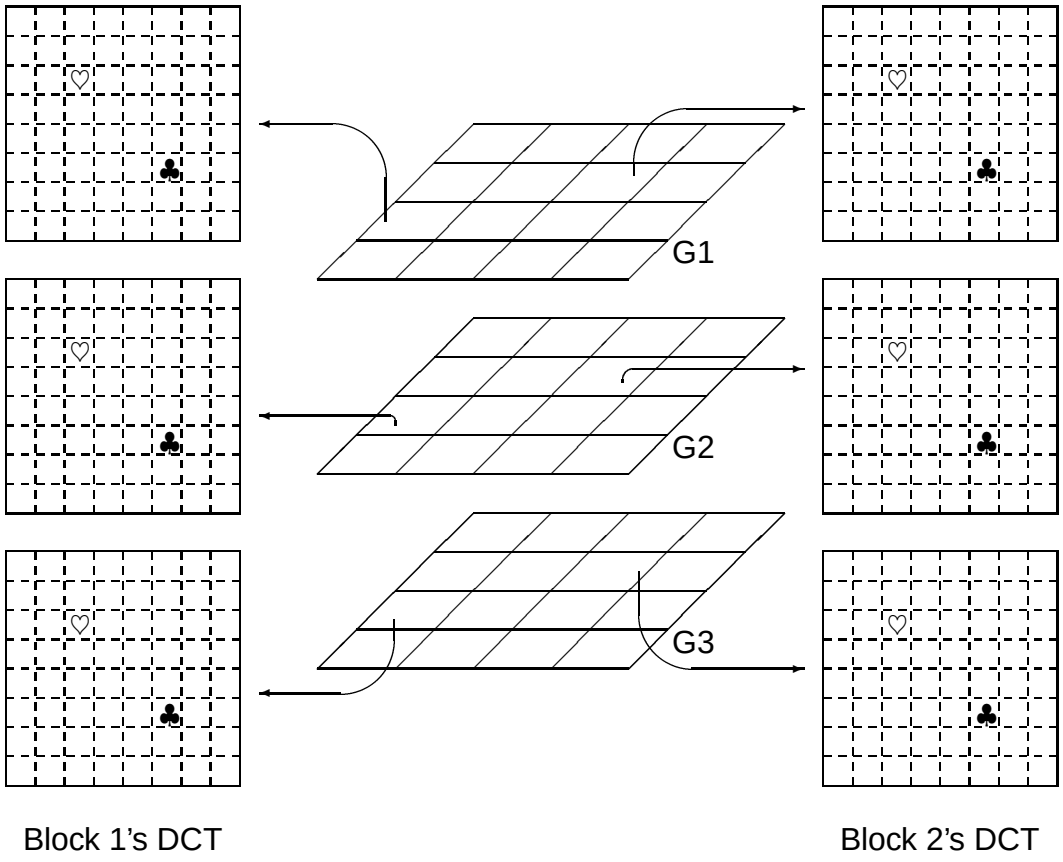
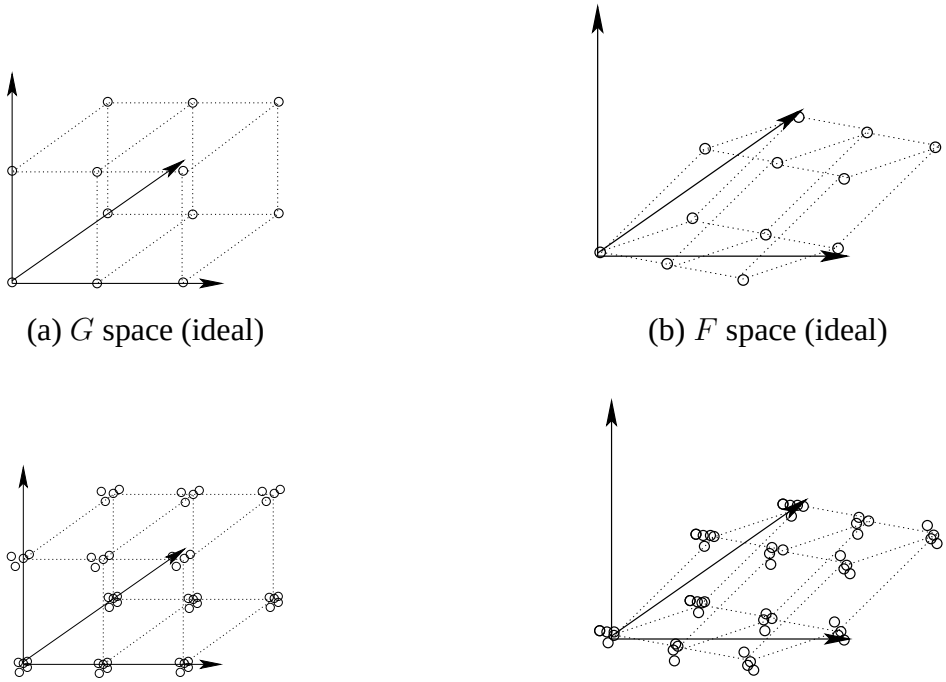


Figure 4.5: *3-D vectors from different* 8×8 *blocks. The vertically stacked triplet of boxes, each with dotted lines, represent the DCT coefficients of an* 8×8 *image block. The 3-D vectors* $[\heartsuit_{G1}\ \heartsuit_{G2}\ \heartsuit_{G3}]^T$ *of quantized DCT coefficients from frequency* $(3, 3)$ *of Block 1 and Block 2 lie on the same rectangular lattice but at possibly different locations. A different rectangular lattice contains the* $(5, 6)$ *frequency 3-D vectors* $[\clubsuit\ \clubsuit\ \clubsuit]^T$ *of quantized DCT coefficients.*

edges of the parallelepiped are determined by the column vectors of the linear component of the affine color transform from G to F (for example, see (4.3)), which we henceforth denote by $\mathcal{T}$. (The additive component of the affine color transform only reflects as a shift in the DC frequency coefficients.) As we shall soon see, the parallelepiped structure of the DCT coefficients can be exploited to solve the CHEst problem.



(a) G space (ideal)

(b) F space (ideal)

(c) G space (with round-off)

(d) F space (with round-off)

Figure 4.6: *Lattice structures in the previously JPEG-compressed color image: (a) Assuming round-off errors are absent, all 3-D vectors of G space's DCT coefficients from the different 8×8 image blocks but same DCT frequency lie exactly on the vertices of a rectangular lattice. The 3-D vectors are denoted by small circles. (b) In the F space, the 3-D vectors of DCT coefficients lies exactly on the vertices of a parallelepiped lattice. (c) and (d) Round-off errors slightly perturb the 3-D vectors of DCT coefficients in the G and F color spaces from the vertices of the rectangular lattice and parallelepiped lattice respectively.*

4.6.2 Lattice algorithms

Lattices are regular arrangements of points in space and their study arises in a number of fields such as coding theory, number theory, and crystallography. Consider an ordered set of m vectors $\{b_1, b_2, \ldots, b_m\}$. The *lattice* $\mathcal{L}$ spanned by these vectors consists of all *integral* linear combinations $u_1 b_1 + u_2 b_2 + \ldots + u_m b_m, \forall u_i \in \mathbb{Z}$. The structure in Figures 4.6(a) and (b) are both examples of 3-D lattices. Any set $\{b_1, b_2, \ldots, b_m\}$ forms a basis for a given $\mathcal{L}$ if the following two conditions

hold:

1. Every vector in $\mathcal{L}$ can be expressed as $u_1 b_1 + u_2 b_2 + \ldots + u_m b_m$, $u_i \in \mathbb{Z}$ for $i = 1, \ldots, m$.

2. If $u_1 b_1 + u_2 b_2 + \ldots + u_m b_m = 0$, $u_i \in \mathbb{Z}$, then $u_i = 0$ for $i = 1, \ldots, m$.

For any given lattice $\mathcal{L}$, there exist many different sets of basis vectors. If $\mathcal{B} := [b_1\ b_2\ \ldots\ b_m]$ is a matrix whose columns form a basis for a lattice $\mathcal{L}$, then any arbitrary basis vector set can be expressed as the columns of $\mathcal{BU}$, where $\mathcal{U}$ is an integer matrix with determinant equal to ± 1.

Properties of nearly orthogonal lattice basis vectors

Later in this section, we will need to understand and exploit the properties of lattice basis vectors that are nearly orthogonal. This is because the column vectors of an affine color transform's linear component, which determine the structure of the parallelepiped lattice in the *F* space, are typically nearly orthogonal. We have discovered two new, fundamental properties of such nearly orthogonal lattice basis vectors.

To formally describe the two properties, we first need to define the following terms:

- *Weak θ-orthogonality:* We define an ordered set of vectors $\{b_1, b_2, \ldots, b_m\}$ to be weakly θ-orthogonal if for any $i = 2, \ldots, m$, the angle between b_i and any non-zero linear combination of $\{b_1, \ldots, b_{i-1}\}$ lies in the range $[\theta, \pi - \theta]$; that is,

$$\cos^{-1}\left(\frac{|\langle b_i, \sum_{j=1}^{i-1} \lambda_i\, b_i\rangle|}{\|b_i\|_2 \left\|\sum_{j=1}^{i-1} \lambda_i\, b_i\rangle\right\|_2}\right) \geq \theta, \text{ for all } \sum_j |\lambda_j| > 0, \text{ with } \lambda_j \in \mathbb{R}. \tag{4.16}$$

- *Strong θ-orthogonality:* We define a set of vectors $\{b_1, b_2, \ldots, b_m\}$ to be strongly θ-orthogonal if all its permutations are weakly θ-orthogonal.

Our first observation is that a nearly orthogonal set of basis vectors contains the shortest non-zero lattice vector.

Proposition 5 *Let $\mathcal{B} := [b_1, b_2, \ldots, b_m]$ be a matrix whose columns form a ordered set of basis vectors for a lattice $\mathcal{L}$. If the columns of $\mathcal{B}$ are weakly $\left(\frac{\pi}{3} + \epsilon\right)$-orthogonal, $0 \leq \epsilon \leq \frac{\pi}{6}$, then $\mathcal{B}$'s shortest column is the shortest non-zero vector in the lattice $\mathcal{L}$; that is,*

$$\min_{j \in 1,\ldots,m} \|b_j\|_2 \leq \left\| \sum_{i=1}^{m} u_i b_i \right\|_2, \qquad \textit{for all } \sum_{i=1}^{m} |u_i| \geq 1, \textit{ with } u_i \in \mathbb{Z}. \tag{4.17}$$

Further, if $\epsilon > 0$, then

$$\min_{j \in 1,\ldots,m} \|b_j\|_2 < \left\| \sum_{i=1}^{m} u_i b_i \right\|_2, \qquad \textit{for all } \sum_{i=1}^{m} |u_i| > 1, \textit{ with } u_i \in \mathbb{Z}. \tag{4.18}$$

Proposition 5's proof, which is described in Appendix G.1, follows by induction; it is easy to first verify that the proposition holds for 2-dimensional (2-D) lattices and then extend the proposition to higher dimensions. An immediate corollary is

Corollary 1 *A strongly $\frac{\pi}{3}$-orthogonal set of basis vectors contains the shortest non-zero lattice vector.*

Proposition 5 and its corollary are significant contributions to the theory of lattices because the problem of finding the shortest non-zero vector in an arbitrary lattice is believed to be a NP-hard problem. To the best of our knowledge, Proposition 5 is the first result that can provide a certificate that a set of vectors contains the shortest non-zero lattice vector.

Our second observation describes the conditions under which a lattice contains a unique (modulo permutations and sign changes) set of nearly orthogonal lattice basis vectors.

Proposition 6 *Let $\mathcal{B} := [b_1\ b_2\ \ldots\ b_m]$ be a matrix whose columns form a basis for a lattice $\mathcal{L}$ and are strongly θ-orthogonal. For all $i \in 1, \ldots, m$, if*

$$\min_{j \in 1,\ldots,m} \|b_j\|_2 \leq \|b_i\|_2 < \min_{j \in 1,\ldots,m} \|b_j\|_2\ \eta(\theta), \tag{4.19}$$

with

$$\eta(\theta) = \left(\frac{\sqrt{3}\sin(\theta) - 3|\cos(\theta)|}{4\sin^2(\theta) - 3} \right) \tag{4.20}$$

then any strongly θ-orthogonal set of basis vectors can be obtained by simply permuting and changing the signs of the $\mathcal{B}$'s columns.

Proposition 6's proof is described in Appendix G.1. In words, Proposition 6 claims that a nearly orthogonal set of basis vectors is unique when the lengths of all the basis vectors are nearly equal. For example, both Fig. 4.7(a) and (b) illustrate 2-D lattices that can be spanned by orthogonal basis vectors. For the lattice in Fig. 4.7(a), the ratio of the basis vector's lengths is less than $\eta\left(\frac{\pi}{2}\right) = \sqrt{3}$. Hence, there exists only one set (modulo permutations and sign changes) of basis vectors such that the angle between them is greater than $\frac{\pi}{3}$. In contrast, the lattice in Fig. 4.7(b) contains many sets of strongly $\frac{\pi}{3}$-orthogonal basis vectors.

Lattice reduction

A celebrated lattice problem of interest to us is the so-called *lattice reduction* problem, which can be stated as follows: Given a set of vectors b_i's that span a lattice $\mathcal{L}$, find an ordered set of *basis*

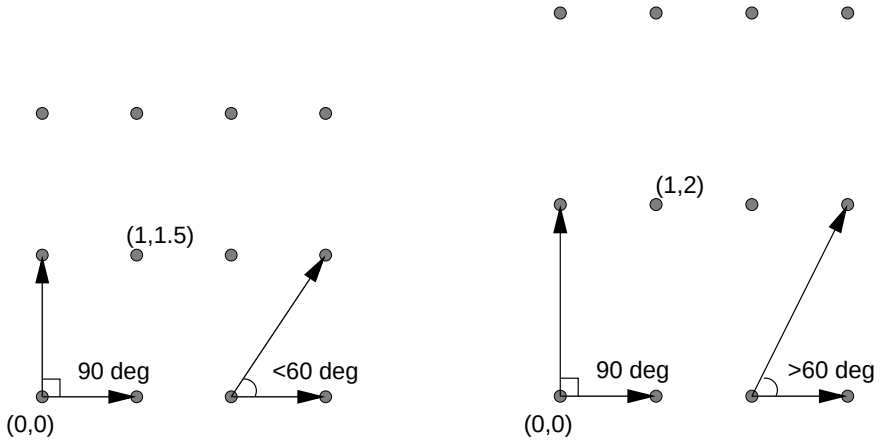


Figure 4.7: *(a) The vectors comprising the lattice are denoted by filled circles. The lattice contains two orthogonal basis vectors with lengths* 1 *and* 1.5 *respectively. Since* $1.5 < \sqrt{3}$*, the lattice contains no other set of basis vectors such that the angle between them is greater than* $\frac{\pi}{3}$ *radians (60 degrees). (b) The lattice contains two orthogonal basis vectors with lengths* 1 *and* 2 *respectively. The figure illustrates two sets of strongly* $\frac{\pi}{3}$*-orthogonal basis vectors for the same lattice.*

vectors for $\mathcal{L}$ such that [62]

1. the basis vectors are nearly orthogonal,

2. the shorter basis vectors appear first in the ordering.

A major breakthrough in the lattice theory area was the discovery of a polynomial time algorithm to perform lattice reduction by Lenstra, Lenstra, and Lovasz; this algorithm is referred to as the LLL algorithm henceforth. The LLL algorithm, which can be intuitively understood as an adaptation of Gram-Schmidt orthogonalization, performs lattice reduction by sequentially processing the vectors b_i's and performing a combination of very simple operations such as

1. Change the order of the basis vectors.

2. Add to one of the vectors b_i an integral multiple of another vector b_j. Note that the vectors resulting from such integral operations still lie on the same lattice.

3. Delete any resulting zero vectors.

The LLL algorithm has since proved invaluable in many areas of mathematics and computer science, especially in algorithmic number theory and cryptography [62, 63].

Closest vector problem

The *closest vector problem* (CVP), another famous lattice problem of interest to us, is closely related to the lattice reduction problem and can be stated as follows: Given a lattice $\mathcal{L}$ and an arbitrary vector v, find the vector on the lattice $\mathcal{L}$ that is closest to v. For a comprehensive semi-tutorial paper on the CVP and algorithms to solve the same, we refer the reader to [64].

4.6.3 LLL provides parallelepiped lattice's basis vectors

The LLL algorithm, when employed on the 3-D vectors of F space's AC frequency DCT coefficients (assuming round-off errors are absent), produces a set of nearly orthogonal, norm-sorted basis vectors for the parallelepiped lattice containing the 3-D vectors. (Since the additive component of the affine color transformation causes an unknown shift in the DC coefficients, the DC coefficients are handled separately.) In fact, during our experiments, we have observed that the set of basis vectors returned by LLL is always strongly $\frac{\pi}{3}$-orthogonal; this conforms with the popular notion that, in practice, the LLL performs significantly better than what is guaranteed theoretically [62, 63]. Any basis that spans the parallelepiped lattice of 3-D vectors of F space's i^{th}-frequency

DCT coefficients can be written as

$$\mathcal{B}_i := \begin{bmatrix} & & \\ & \mathcal{T} & \\ & & \end{bmatrix} \begin{bmatrix} q_{i,1} & 0 & 0 \\ 0 & q_{i,2} & 0 \\ 0 & 0 & q_{i,3} \end{bmatrix} \begin{bmatrix} & & \\ & \mathcal{U}_i & \\ & & \end{bmatrix} \tag{4.21}$$

$$=: \mathcal{T}\,\mathcal{Q}_i\,\mathcal{U}_i \tag{4.22}$$

The $\mathcal{U}_i$, which is a unit-determinant matrix with integer entries, accounts for the non-uniqueness of lattice basis vectors.

4.6.4 Estimating scaled linear component of the color transform

The columns of $\mathcal{B}_i$ that is returned by the LLL (see (4.22)) after acting on i-th frequency 3-D vectors are sometimes misaligned with the columns of $\mathcal{T}\mathcal{Q}_i$, due to a non-identity $\mathcal{U}_i$. In this subsection, with the help of Propositions 5 and 6, we determine the appropriate $\mathcal{U}_i$'s, and thereby extract the $\mathcal{T}\mathcal{Q}_i$'s from the different $\mathcal{B}_i$'s.

Sometimes, the quantization step-sizes for the three components are chosen to be equal, that is, $q_{i,1} \approx q_{i,2} \approx q_{i,3}$, especially if the compression color space G is a member of RGB family. Further, the columns of typical $\mathcal{T}$'s are strongly θ-orthogonal, $\frac{\pi}{3} \leq \theta \leq \frac{2\pi}{3}$. Hence, by invoking Proposition 6, we can infer that $\mathcal{T}\,\mathcal{Q}_i$ is the only set of $\frac{\pi}{3}$-orthogonal basis for the lattice. Hence, in this case, the LLL algorithm directly estimates the columns of $\mathcal{T}\mathcal{Q}_i$ as the LLL-reduced basis $\mathcal{B}_i$; that is, the $\mathcal{U}_i$ is just permutation a matrix.

Often, the quantization step-sizes at the i^{th} frequency are chosen to be different for the three components of G. For example, if G is a member of $YCbCr$ family, then the quantization step-

size for the luminance component Y is typically chosen to be much smaller than that for the chrominance components Cb and Cr. In this case, Proposition 5 helps us infer that the first vector of $\mathcal{B}_i$ is the smallest vector in the lattice and that this vector would necessarily be aligned with one of $\mathcal{T}$'s columns. The remaining two $\mathcal{B}_i$ columns could be misaligned with the columns of $\mathcal{T}$ due to the addition of a small integer multiple of the smallest vector. For example, $\mathcal{U}_i$ could be

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

in which case, the third column vector of the $\mathcal{B}_i$ would be misaligned with the third column of $\mathcal{T}\,\mathcal{Q}_i$ due to the addition of $\mathcal{T}\,\mathcal{Q}_i$'s first column. In this case, we can obtain the columns of $\mathcal{T}\,\mathcal{Q}_i$ for the different DCT frequencies by comparing the $\mathcal{B}_i$'s across frequencies and estimating the $\mathcal{U}_i$'s.

From (4.22), we can infer that all of the $\mathcal{U}_i$'s satisfy the property that $\mathcal{U}_i\mathcal{B}_i^{-1}\mathcal{B}_j\mathcal{U}_j^{-1}$ is a diagonal matrix. Since a typical $\mathcal{T}$'s columns are nearly orthonormal, the actual $\mathcal{U}_i$'s add or subtract at most a unit multiple of the first column (smallest vector) of $\mathcal{B}_i$ to the second and third columns of $\mathcal{B}_i$. If $\frac{q_{j,1}}{q_{i,1}} \neq \frac{q_{j,2}}{q_{i,2}}$ and $\frac{q_{j,1}}{q_{i,1}} \neq \frac{q_{j,3}}{q_{i,3}}$, which is a reasonable assumption in practice,[4] then a unique combination of $\mathcal{U}_i$ and $\mathcal{U}_j$ ensures that $\mathcal{U}_i\mathcal{B}_i^{-1}\mathcal{B}_j\mathcal{U}_j^{-1}$ is a diagonal matrix. Hence, we simply search for the $\mathcal{U}_i$ and $\mathcal{U}_j$, $i \neq j$, such that $\mathcal{U}_i\mathcal{B}_i^{-1}\mathcal{B}_j\mathcal{U}_j^{-1}$ is a diagonal matrix. This yields the columns of $\mathcal{B}_i\,\mathcal{U}_i^{-1} = \mathcal{T}\,\mathcal{Q}_i$ for the different AC frequencies. (If an rough estimate of $\mathcal{T}$ is known, then it is straightforward to search for a $\mathcal{U}_i$ that aligns $\mathcal{B}_i\mathcal{U}_i^{-1}$ with the columns of $\mathcal{T}$.)

[4]Typically, different quantization tables are used for the luminance and chrominance components.

4.6.5 Estimating the complete color transform and quantization step-sizes

Once we estimate the scaled linear component of the $\mathcal{T}\,\mathcal{Q}_i$ for the different AC DCT frequencies, we can extract the color transform $\mathcal{T}$ and quantization step-sizes $q_{i,j}$.

To extract the linear component $\mathcal{T}$ of the color transform from $\mathcal{T}\,\mathcal{Q}_i$, we need to estimate the norms of each of the three columns of $\mathcal{T}$. The columns norms of $\mathcal{T}$ are the largest numbers such that the respective column vectors of $\mathcal{T}\,\mathcal{Q}_i$ are integer multiples of $\mathcal{T}$. Let $\mathcal{T}(:,k)$ and $(\mathcal{T}\,\mathcal{Q}_i)\,(:,k)$, $k = \{1,2,3\}$, denote the k^{th} column vectors of $\mathcal{T}$ and $\mathcal{T}\,\mathcal{Q}_i$ respectively. We set the norms of $\mathcal{T}$'s columns as the solution to the following penalized least-squares cost function,

$$\|\mathcal{T}(:,k)\|_2 = \arg\min_{\kappa} \sum_i \left(\left(\|(\mathcal{T}\,\mathcal{Q}_i)\,(:,k)\|_2 - \kappa\,\mathrm{round}\left(\frac{\|(\mathcal{T}\,\mathcal{Q}_i)\,(:,k)\|_2}{\kappa} \right) \right)^2 + \beta\,\mathrm{round}\left(\frac{\|(\mathcal{T}\,\mathcal{Q}_i)\,(:,k)\|_2}{\kappa} \right) \right). \quad (4.23)$$

The first term ensures that the column norms of $\mathcal{T}\,\mathcal{Q}_i$ conform to integer multiples of the column norm of $\mathcal{T}$, and the second term ensures that the column norm of $\mathcal{T}$ is large; β controls the tradeoff between the two terms. Thus, we can estimate $\mathcal{T}$ from the different $\mathcal{T}\,\mathcal{Q}_i$'s.

Once we know the column norm of $\mathcal{T}$, we can estimate the quantization step-sizes for all the AC DCT frequencies as $q_{i,k} = \frac{\|(\mathcal{T}\,\mathcal{Q}_i)(:,k)\|_2}{\|\mathcal{T}(:,k)\|_2}$.

With the knowledge of $\mathcal{T}$, we can also estimate the DC frequency quantization step-sizes and the additive component of the color transform from the DC frequency DCT coefficients. We first transform the 3-D vectors of F space's DC coefficients using $\mathcal{T}^{-1}$. This yields the DC frequency DCT coefficients of the G space but the coefficients are shifted by an unknown amount due to the

additive component of the color transform. Due to quantization, the respective 1-D histograms of coefficients from each component of the obtained $\mathcal{T}^{-1}$-transformed vectors show a similar behavior to that illustrated in Fig. 4.3. To estimate the quantization step-size, we first compute the magnitude of the Discrete Fourier Transform (DFT) of the histogram, which is immune to the unknown shift. The near-periodic behavior of the populated histogram bins ensures that the frequency at which the histogram's DFT magnitude first peaks away from zero frequency corresponds to the quantization step-size.

The color transform's additive component's coefficients can only be such that the histogram of estimated G space's DC frequency DCT coefficients is populated at zero. Further, the typical role of the additive component of the color transform is to ensure that the component values conform to the 0–255 range after transformation. Hence, we simply choose the additive component's coefficients to simultaneously ensure both these properties. We clarify that many solutions can satisfy the above two criteria. However, fortunately, the mis-estimations in the additive component of the color transform do not affect typical CHEst applications such as recompression and enhancement.

4.6.6 Round-offs perturb ideal geometry

Round-offs employed during JPEG decompression (see Fig. 4.1 and Section 4.3) perturb the DCT coefficient values. Consequently, the 3-D vectors of DCT coefficients in the G and F color spaces lie only approximately on the rectangular lattice and parallelepiped lattice respectively (see Fig. 4.6).

Let V denote the 3-D error vector that captures the perturbation of a 3-D DCT coefficient vector from the parallelepiped lattice in the F color space due to round-offs. Based on (4.6), we

can statistically model the distribution of the norm of the perturbation vector as

$$P\left(\|V\|_2 = t\right) = \frac{1}{\Upsilon} \exp\left(-\frac{t^2}{2\sigma^2}\right), \qquad \text{for } t \in [-\zeta, \zeta]. \tag{4.24}$$

We must fuse this knowledge with the LLL algorithm to combat the round-off errors.

4.6.7 Combating round-off noise

In Sections 4.6.3–4.6.5, for the sake of clarity, we designed our algorithm assuming round-off noise was absent. In this section, we clarify the modifications required to combat round-off noise.

Our goal is use the LLL algorithm to estimate a set of nearly orthogonal, norm-sorted basis vectors $\widehat{\mathcal{B}}_i$ such that all the observed 3-D vectors lie *close* to the parallelepiped lattice spanned by $\widehat{\mathcal{B}}_i$. The LLL algorithm *sequentially* sorts through a set of input vectors and maintains a basis spanning the sorted vectors at any instant. However, if the vectors are noisy, then the noise gets amplified and propagates during the sequence of LLL's arithmetic operations. Fortunately, for many parallelepiped lattice locations, we observe multiple realizations of 3-D DCT vectors perturbed due to round-off because even reasonable-sized images contain many 8×8 blocks of pixels. To reduce round-off noise propagation, we initiate the LLL algorithm with the least noisy vector first. The order of the inputs to the LLL algorithm is determined by first computing the 3-D histogram of the 3-D DCT coefficient vectors from the different 8×8 blocks and then sorting the vectors in the descending order of the histogram values.

We can leverage our knowledge about the round-off errors' distribution to incorporate additional noise attenuation steps. During the sequence of LLL operations, when encountered with a vector that lies within a distance of ζ from the closest point on the lattice spanned by an estimate $\widetilde{\mathcal{B}}_i$

of the lattice basis, we assume that the the current noisy vector lies in the "span" of $\widetilde{\mathcal{B}}_i$. The closest point can be computed using an algorithm to solve the CVP [64]. We simply use the current noisy realization to update the $\widetilde{\mathcal{B}}_i$ and obtain $\widehat{\mathcal{B}}_i$. Let $\mathcal{D}_i$ denote a $3 \times m$ matrix of m 3-D DCT coefficient vectors that have been sorted through including the current noisy vector. Let $\widetilde{\mathcal{B}}_i\mathcal{I}_i$ denote the lattice points closest to the respective columns of $\mathcal{D}_i$, where $\mathcal{I}_i$ is a $3 \times m$ matrix with integer entries. If $\widetilde{\mathcal{B}}_i$'s entries are close to the basis $\mathcal{B}$ spanning the actual underlying lattice, then the $\mathcal{I}_i$ entries are exact. Assuming the perturbation vectors are independent of each other, we have using (4.24)

$$P\left(\widetilde{\mathcal{B}}_i|\mathcal{I}_i\right) \propto \exp\left(-\frac{1}{2\sigma^2}\left\|\mathcal{D}_i - \widetilde{\mathcal{B}}_i\mathcal{I}_i\right\|_2^2\right), \tag{4.25}$$

where $\|.\|_2^2$ denotes the sum of squares of all entries in the matrix. We can update the estimate $\widetilde{\mathcal{B}}_i$ by differentiating the exponent $\left\|\mathcal{D}_i - \widetilde{\mathcal{B}}_i\mathcal{I}_i\right\|_2^2$ with respect to the entries of $\widetilde{\mathcal{B}}_i$ and setting them to zero to obtain

$$\widehat{\mathcal{B}}_i = \left(\mathcal{D}_i\mathcal{I}_i^T\right)\left(\mathcal{I}_i\,\mathcal{I}_i^T\right)^{-1}. \tag{4.26}$$

The above equation updates the $\widetilde{\mathcal{B}}_i$ to ensure that the round-off error between the lattice spanned by $\widehat{\mathcal{B}}_i$ and the observations $\mathcal{D}$ is minimized. We have ignored the finite support of the distribution during the update assuming that the round-off error norms stay less than ζ. Combining the update step with the LLL algorithm successfully curbs the propagation and amplification of the round-off errors during the LLL's arithmetic operations.

The presence of round-off noise prods us to slightly modify the estimation of the unit-determinant $\widehat{\mathcal{U}}_i$'s as well. Due to round-off, the final estimate $\widehat{\mathcal{B}}_i$ obtained using (4.26) is not exactly equal to $\mathcal{B}_i$, but contains some small errors. Consequently, unlike $\mathcal{U}_i\mathcal{B}_i^{-1}\mathcal{B}_j\mathcal{U}_j^{-1}$ which is

exactly diagonal (see Section (4.6.4)), $\mathcal{U}_i\widehat{\mathcal{B}}_i^{-1}\widehat{\mathcal{B}}_j\mathcal{U}_j^{-1}$ is only diagonally dominant. Hence, we define a new measure for the "diagonality" of a matrix as the ratio of the norm of the diagonal elements to the norm of all elements of the matrix; the measure is equal to one if and only if a matrix is exactly diagonal. We estimate $\widehat{\mathcal{U}}_i$ and $\widehat{\mathcal{U}}_j$ using $\widehat{\mathcal{B}}_i$ and $\widehat{\mathcal{B}}_j$, we choose the combination of $\widehat{\mathcal{U}}_i$ and $\widehat{\mathcal{U}}_j$ such that the diagonality measure of $\widehat{\mathcal{U}}_i\widehat{\mathcal{B}}_i^{-1}\widehat{\mathcal{B}}_j\widehat{\mathcal{U}}_j^{-1}$ is maximized.

Ideally, any non-zero β in (4.23) will estimate the column norm of the actual $\mathcal{T}$. However, since we only have an estimate $\widehat{\mathcal{B}}_i\widehat{\mathcal{U}}_i^{-1}$ of $\mathcal{T}\mathcal{Q}_i$, we need to set β prudently; in practice, we set $\beta := \frac{0.2}{\text{mean}\left(\|(\mathcal{T}\mathcal{Q}_i)(:,k)\|_2\right)}$.

4.6.8 Algorithm steps

We now have all the pieces to estimate the compression history, namely, the affine color transform from G space to F space and the quantization tables.

1. *Estimate all the lattices for the AC frequency components.*

   We fuse the LLL algorithm with noise attenuation to estimate the different lattices as follows:

   (a) Choose an AC DCT frequency i.

   (b) Take the 3-D histogram of the 3-D DCT coefficient vectors from the different 8×8 blocks and sort the locations of the histogram bins in descending order of the histogram values obtained in Step 1a.

   (c) Choose the first location vector on the sorted list that lies outside the sphere with radius $\zeta = 5$ as a basis vector to the lattice. Any vector within the sphere could potentially be a noisy realization of origin $[0\ 0\ 0]^T$, and is hence ignored.

(d) Choose the next location vector. If there are no more vectors left in the list, then go to Step 1h.

(e) Calculate the error vector between the currently chosen vector and the closest vector that lies on the lattice spanned by the current set of basis vectors.

(f) If the error vector calculated in Step 1e lies outside the sphere with radius ζ, then add the currently chosen vector to list of basis vectors. Perform LLL on this set of basis vectors. Go to Step 1d.

(g) If the error vector lies inside the sphere with radius ζ, then update the basis vectors to minimize the cumulative probability of error (see Section 4.6.7 for details). Go to Step 1d.

(h) Output the computed set of basis vectors as the $\widehat{\mathcal{B}}_i$. Go to Step 1a to choose the next AC frequency.

2. *Estimate the $\mathcal{T}\ \mathcal{Q}_i$ for all the AC frequencies.*

(a) Choose a $\widehat{\mathcal{B}}_i$ from Step 1.

(b) As described in Section 4.6.4, choose $\widehat{\mathcal{U}}_i$ and $\widehat{\mathcal{U}}_j$, $i \neq j$, such that $\widehat{\mathcal{U}}_i\, \widehat{\mathcal{B}}_i^{-1}\, \widehat{\mathcal{B}}_j\, \widehat{\mathcal{U}}_j^{-1}$ is nearly diagonal for all j. (If the norms of the columns of $\widehat{\mathcal{B}}_i$ are with a factor of 2 of each other, then we can immediately infer that $\widehat{\mathcal{B}}_i$ is already aligned with $\mathcal{T}\ \mathcal{Q}_i$ due to the properties of the LLL algorithm and typical $\mathcal{T}$'s, that is, set the $\widehat{\mathcal{U}}_i$ to be the identity matrix.)

(c) Estimate the $\mathcal{T}\ \mathcal{Q}_i$ for each AC frequency as $\widehat{\mathcal{B}}_i\, \widehat{\mathcal{U}}_i^{-1}$.

3. *Estimate the color transform and the quantization tables.*

   The details of the following three steps are described in Section 4.6.5 and Section 4.6.7.

   (a) Obtain the $\widehat{\mathcal{T}}$ from the $\widehat{\mathcal{B}}_i\widehat{\mathcal{U}}_i^{-1}$'s by estimating the matrix's column norms using (4.23).

   (b) Estimate the quantization tables for all the frequencies using the $\widehat{\mathcal{B}}_i\widehat{\mathcal{U}}_i^{-1}$ and the $\widehat{\mathcal{T}}$.

   (c) Estimate the additive component the color transform using the estimated DC frequency quantization step-size and the observed DC frequency DCT coefficients.

4.6.9 Lattice-based CHEst results

To verify the efficacy of the lattice-based CHEst algorithm, we used the 512×512 *Lenna* color image [61]. We JPEG-compressed the *Lenna* image in the *ITU.BT-601 YCbCr* space (see (4.3)). The luminance plane Y's DCT coefficients were quantized using table 1 from Fig. 4.2 and the chrominance planes Cb's and Cr's DCT coefficients were quantized using table 2 from Fig. 4.2. The *Cb* and *Cr* planes were not subsampled during compression. The image was then decompressed and then transformed to the *RGB* space. Our algorithm operated in this *RGB* space, and tried to estimate the affine transformation from *ITU.BT-601 YCbCr* to current *RGB* space (see (4.3)).

We estimated the lattices using Step 1 of the algorithm described in Section 4.6.8. For the AC frequencies $[1, 2]$ and $[1, 3]$, the estimated lattices were

$$\widehat{\mathcal{B}}_{[1,2]} = \begin{bmatrix} -7.00 & 15.50 & -6.96 \\ -7.01 & -7.81 & -10.72 \\ -7.00 & 0.02 & 12.66 \end{bmatrix} \quad \text{and} \quad \widehat{\mathcal{B}}_{[1,3]} = \begin{bmatrix} -6.01 & 13.64 & -5.95 \\ -5.99 & -15.91 & -10.90 \\ -6.00 & -5.91 & 18.94 \end{bmatrix}. \tag{4.27}$$

The first columns of $\widehat{\mathcal{B}}_{[1,2]}$ and $\widehat{\mathcal{B}}_{[1,3]}$ are the smallest of the respective three vectors. Further, as

expected, the first columns are both aligned with one of the columns of the linear component $\mathcal{T}$ of the *ITU.BT-601 YCbCr* to *RGB* transformation. However, $\widehat{\mathcal{B}}_{[1,2]}$'s third column and $\widehat{\mathcal{B}}_{[1,2]}$'s second and third column are not scaled versions of any of $\mathcal{T}$'s columns due to the addition of respective first columns. By using Step 2 of the lattice-based CHEst algorithm, we obtain

$$\widehat{\mathcal{U}}_{[1,2]} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \widehat{\mathcal{U}}_{[1,3]} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.28}$$

Hence, our estimate for the scaled linear component of the color transform for the AC frequencies $[1, 2]$ and $[1, 3]$ was

$$\widehat{\mathcal{B}}_{[1,2]}\widehat{\mathcal{U}}_{[1,2]}^{-1} = \begin{bmatrix} -7.00 & 15.50 & 0.04 \\ -7.01 & -7.81 & -3.70 \\ -7.00 & 0.02 & 19.66 \end{bmatrix} \quad \text{and} \quad \widehat{\mathcal{B}}_{[1,3]}\widehat{\mathcal{U}}_{[1,3]}^{-1} = \begin{bmatrix} -6.01 & 19.65 & 0.06 \\ -5.99 & -9.92 & -4.91 \\ -6.00 & 0.09 & 24.94 \end{bmatrix}. \tag{4.29}$$

Similarly, we computed the $\widehat{\mathcal{B}}_i\widehat{\mathcal{U}}_i^{-1}$ for all AC frequencies. We use (4.23) to estimate $\mathcal{T}$'s column norms. The signs of columns were chosen to ensure that the largest magnitude entry is positive. See (4.30) for the obtained estimate of the 3×3 matrix $\widehat{\mathcal{T}}$. We have appropriately ordered the three columns to facilitate easy comparison with the actual $\mathcal{T}$ transformation in (4.3).

As outlined in Step 3 of lattice-based CHEst algorithm, we estimated that the quantization tables shown in Fig. 4.8 were employed to compress the Y, Cb, and Cr color planes. An $\times$ indicates that the quantization step-size estimation was not possible because all the DCT coefficients were quantized to zero. Our estimate for the additive component of the transform from *ITU.BT-601*

10	7	6	10	14	24	31	×
7	7	8	11	16	35	36	33
8	8	10	14	24	34	×	×
8	10	13	17	31	×	×	×
11	13	22	34	41	×	×	×
14	21	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×

Y plane

10	11	14	28	×	×	×	×
11	13	16	×	×	×	×	×
14	16	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×

Cb plane

10	11	14	28	×	×	×	×
11	13	16	×	×	×	×	×
14	16	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×
×	×	×	×	×	×	×	×

Cr plane

Figure 4.8: *Quantization table estimates using lattice-based CHEst.*

YCbCr to *RGB* transformation was $[3\ 88\ 138]^T$ (see (4.30)).

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.00 & 0.00 & 1.41 \\ 1.00 & -0.35 & -0.71 \\ 1.00 & 1.78 & 0.00 \end{bmatrix} \left(\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} 3 \\ 88 \\ 138 \end{bmatrix} \right). \tag{4.30}$$

As we can verify, both the estimated affine color transform (4.30) and the estimated quantization tables in Fig. 4.8 conform well with the actual previous compression settings (see (4.3) and Fig. 4.2).

4.7 JPEG Recompression: A Sample Application of CHEst

When a given TIFF or BMP image's file-size needs to be reduced, the conventional approach is to naively employ JPEG with an arbitrary choice of compression color space, subsampling factor, and quantization table. Reasonable choices for the color transformations include RGB to YCbCr, Computer RGB to ITU.BT-601 YCbCr, RGB to Kodak PhotoYCC, and sRGB to 8-bit CIELab. Some common subsampling factors are $2 \times 2, 1 \times 1, 1 \times 1$ and $1 \times 1, 1 \times 1, 1 \times 1$. The quantization tables are often set by adjusting a so-called Quality Factor (QF). The QF is a reference number between 1 to 100 used by the IJG JPEG implementation; QF=100 set all the quantizer steps are unity and thus yields the best quality JPEG can possibly achieve. Any combination of the above choices would yield a JPEG image file with a certain file-size. Smaller file-sizes are typically accompanied by increased distortions in the recompressed image. In this section, we demonstrate that using CHEst to recompress a previously JPEG-compressed color image offers significant benefits over a naive recompression approach.

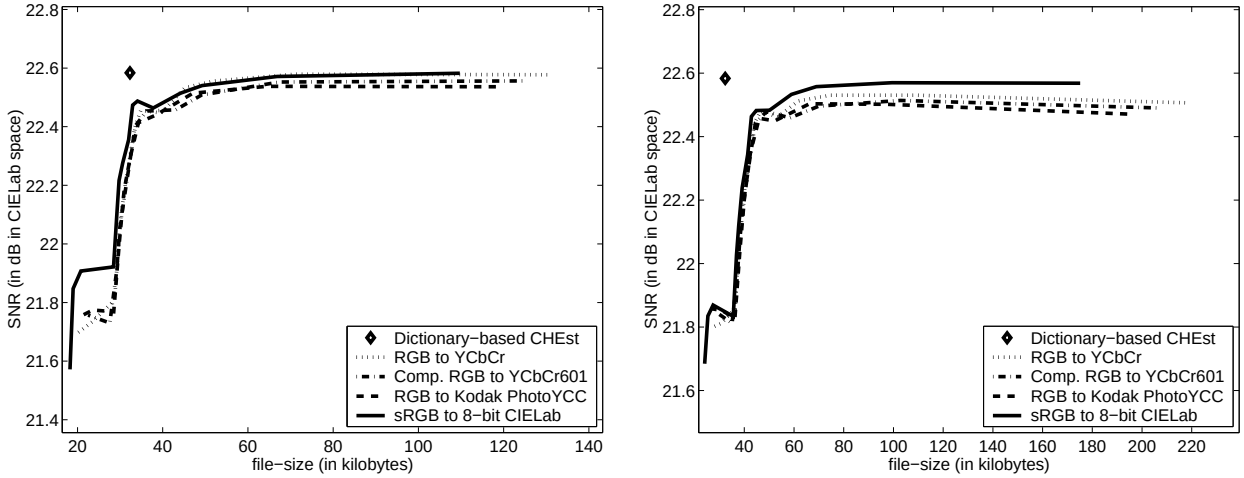
4.7.1 JPEG recompression using dictionary-based CHEst

To demonstrate the dictionary-based CHEst's benefits in JPEG recompression, we choose the exact same test image described in Section 4.5.3—*Lenna* color image previously JPEG-compressed in the 8-bit CIELab color space with $2 \times 2, 1 \times 1, 1 \times 1$ subsampling using quantization table 1 and 2 from Fig. 4.2. As described in Section 4.5.3, the dictionary-based CHEst algorithm accurately estimates the compression history of this test image.

To perform recompression using the dictionary-based CHEst information, we first transform the observed image into the compression color space using the estimated sRGB to 8-bit CIELab

color transformation. Then, we deconvolve the effect of the smoothing employed during previous decompression on the *a* and *b* color planes. After performing $2 \times 2, 1 \times 1, 1 \times 1$ subsampling, using the IJG JPEG implementation [59], we JPEG-compress the 8-bit CIELab color planes with the estimated quantization tables in Fig. 4.4 (setting the $\times$ entries to 100). Our recompression yields a JPEG image with file-size 32.31 kilobytes (KB) with an SNR of 22.58 dB; the SNR is computed in dB with respect to the original *Lenna* image in the perceptually-uniform CIELab color space.

For comparison, we also recompress the image using a variety of settings. We JPEG-compress the test BMP image using the RGB to YCbCr, Computer RGB to ITU.BT-601 YCbCr, RGB to Kodak PhotoYCC, and sRGB to 8-bit CIELab color transforms using $2 \times 2, 1 \times 1, 1 \times 1$ and also using $1 \times 1, 1 \times 1, 1 \times 1$ subsampling. For each chosen color transform and subsampling, we varied the quantization tables using the QF value and noted the resulting JPEG image's file-size (in KB) and the incurred distortion in SNR (in dB in the CIELab space). Each curve in Fig. 4.9 illustrates the behavior of file-size versus SNR as we vary the QF for a particular choice of color space; Fig. 4.9(a) curves are obtained using $2 \times 2, 1 \times 1, 1 \times 1$ subsampling and Fig. 4.9(b) using $1 \times 1, 1 \times 1, 1 \times 1$ subsampling. The curves demonstrate a "knee-point" trend—the SNR remains flat for a broad range of file-sizes, but decreases rapidly for small changes in file-size thereafter. The file-size SNR pair (32.31 KB, 22.58 dB) associated with the image recompressed using dictionary-based CHEst results is marked using the "$\diamondsuit$" symbol. Both the plots confirm that exploiting the dictionary-based CHEst enables us to strike a desirable file-size versus distortion trade-off—we attain the nearly minimum file-size without introducing any significant additional distortion.



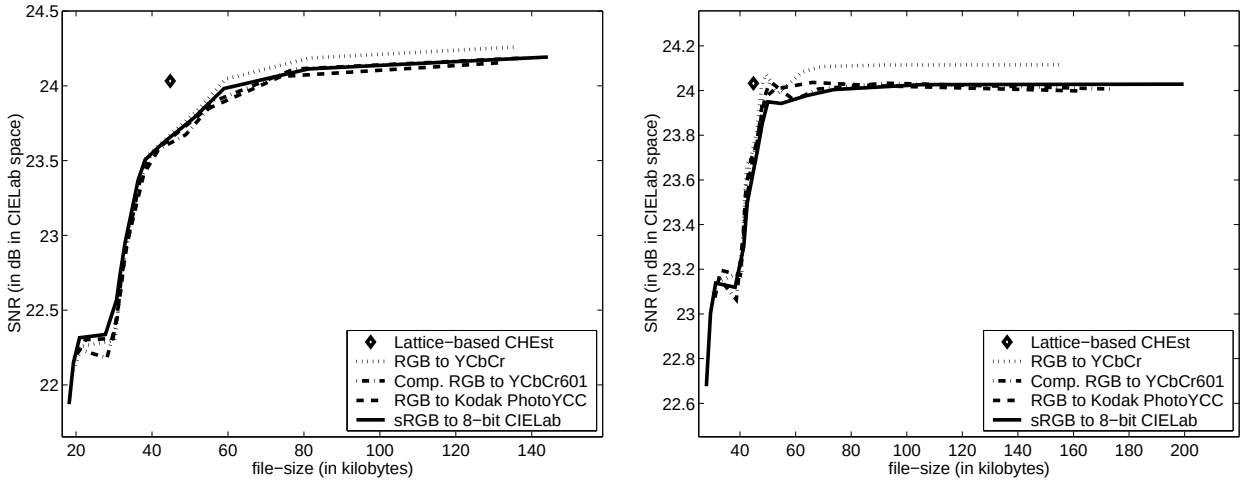
(a) $2 \times 2, 1 \times 1, 1 \times 1$ subsampling (b) $1 \times 1, 1 \times 1, 1 \times 1$ subsampling

Figure 4.9: *Recompression using dictionary-based CHEst. (a) The "♢" marks the file-size SNR pair (*32.31 *KB,* 22.58 *dB) obtained by using dictionary-based CHEst information for JPEG recompression. The curves illustrate the file-size versus SNR behavior obtained by varying the QF. Each curve corresponds to JPEG recompression in the denoted color spaces when* $2 \times 2, 1 \times 1, 1 \times 1$ *subsampling is employed. (b) The plot compares the file-size SNR pair (*32.31 *KB,* 22.58 *dB) with curves that are obtained using* $1 \times 1, 1 \times 1, 1 \times 1$ *subsampling in the denoted color spaces.*

4.7.2 JPEG recompression using lattice-based CHEst

We demonstrate the lattice-based CHEst's benefits in JPEG recompression using the test image described in Section 4.6.9—*Lenna* color image previously JPEG-compressed in the *ITU.BT-601 YCbCr* color space with $1 \times 1, 1 \times 1, 1 \times 1$ subsampling using quantization table 1 and 2 from Fig. 4.2. From, Section 4.6.9 the lattice-based CHEst algorithm accurately estimates the compression history of the test image.

To perform recompression using the lattice-based CHEst information, we transform the observed image to the estimated *ITU.BT-601 YCbCr* space using the inverse of estimated *ITU.BT-601 YCbCr* to *RGB* transformation. We JPEG-compress the three planes with the estimated quantization tables in Fig. 4.2 (setting the $\times$ entries to 100) to obtain an image with file-size=44.81 KB and



(a) $2 \times 2, 1 \times 1, 1 \times 1$ subsampling (b) $1 \times 1, 1 \times 1, 1 \times 1$ subsampling

Figure 4.10: *Recompression using lattice-based CHEst. Plots (a) and (b) are similar to Fig. 4.9 but for lattice-based CHEst. Recompression performed by exploiting lattice-based CHEst information yields a JPEG image whose file-size is* 44.81 *KB and SNR is* 24.03 *dB; a "♢" marks this file-size SNR pair.*

SNR=24.03 dB.

Figure 4.10(a) and (b) compares the file-size SNR pair for lattice-based CHEst recompression with file-size versus SNR curves for naive JPEG recompression in different color spaces at different QFs for $2 \times 2, 1 \times 1, 1 \times 1$ and $1 \times 1, 1 \times 1, 1 \times 1$ subsampling. Figure 4.10 verifies that lattice-based CHEst results also enables us to strike a desirable file-size versus distortion trade-off during JPEG recompression.

Chapter 5

Conclusions

Don't let it end like this. Tell them I said something.

–Pancho Villa

In this thesis, we have exploited smoothness and lattice structures inherent in images to develop novel solutions to deconvolution, inverse halftoning, and JPEG Compression History Estimation (CHEst).

In Chapter 2, we proposed an efficient, hybrid *Fourier-Wavelet Regularized Deconvolution* (ForWaRD) algorithm that effectively combines and balances scalar Fourier shrinkage and wavelet shrinkage. The motivation for the hybrid approach stems from the realization that deconvolution techniques relying on scalar shrinkage in a single transform domain—for example, the LTI Wiener deconvolution filter or the WVD—are inadequate to handle the wide variety of practically encountered deconvolution problems. ForWaRD can be potentially employed in a wide variety of applications, including satellite imaging, seismic deconvolution, and channel equalization.

Theoretical analysis of an idealized ForWaRD algorithm reveals that the balance between the amount of Fourier and wavelet shrinkage is simultaneously determined by the Fourier structure of the convolution operator and the wavelet structure of the desired signal. By analyzing the ForWaRD's MSE decay rate as the number of samples increases, we prove that ForWaRD is also asymptotically optimal like the WVD for certain deconvolution problems.

In 2-D simulations, ForWaRD outperforms the LTI Wiener filter in terms of both visual quality and MSE performance. Further, even for problems suited to the WVD, ForWaRD demonstrates improved performances over a wide range of practical sample-lengths.

There are several avenues for future ForWaRD related research. An interesting twist to ForWaRD would be to first exploit the wavelet domain to estimate $x \circledast h$ from the noisy observation y and then invert the convolution operator. This technique, called the *Vaguelette-Wavelet Decomposition* (VWD), has been studied by Silverman and Abramovich [65]. The salient point of such a technique is that the wavelet-domain estimation now deals with white noise instead of colored noise. However, like the WVD, this technique is also not adequate for all types of $\mathcal{H}$ (for example, a box-car blur). Construction of a universally-applicable deconvolution scheme lying between WVD and VWD appears promising but challenging.

In ForWaRD, we have assumed knowledge of the convolution operator. However, in many cases, the convolution operator is unknown. Such blind deconvolution problems are extremely challenging. For example, if an image undergoes blurring twice before being observed, then which estimate should a blind deconvolution system aim to extract? Why? Finding a meaningful solution to such blind problems is impossible, unless we severely constrain the structure of the convolution operator and the desired input signal. ForWaRD can tractably incorporate a variety of such constraints because of its hybrid nature. Hence we believe that adapting the ForWaRD to tackle blind deconvolution problems is an interesting and promising avenue for future work.

In Chapter 3, we used the linear error diffusion model of [34, 35] to show that inverse halftoning can be posed as a deconvolution problem in the presence of colored noise. Exploiting this new perspective, we proposed the simple *Wavelet-based Inverse Halftoning via Deconvolution* (WInHD)

algorithm based on wavelet-based deconvolution to perform inverse halftoning. Since WInHD is model-based, it is easily tunable to the different error diffusion halftoning techniques. WInHD yields state-of-the-art performance in the MSE sense and visually.

WInHD also enjoys desirable theoretical properties under certain mild conditions. For images in a Besov space, WInHD estimate's MSE is guaranteed to decay rapidly as the spatial resolution of the input gray-scale image increases. Further, if the gray-scale image lies in a Besov space and is noisy before halftoning, then WInHD's MSE decay rate cannot be improved upon by any estimator.

WInHD assumes *a priori* knowledge of the error diffusion filter. However, sometimes the error diffusion filter is not known. From WInHD's perspective, inverse halftoning reduces to blind deconvolution under such circumstances. One straightforward approach to such blind inverse halftoning problems is to first estimate the error diffusion filter coefficients using adaptive techniques such as the one proposed by Wong [66] and the, perform WInHD. An interesting, albeit challenging, area for future research would be to invoke ideas from blind deconvolution and perform the error diffusion filter estimation and inverse halftoning in tandem.

To facilitate efficient hardware implementation, in addition to requiring minimal memory and computations, an inverse halftoning algorithm should also be compatible with fixed-point digital signal processors. For example, the gradient-based algorithm [41] is optimized for hardware implementation while still obtaining good inverse halftoning results. Since our focus in this thesis has been primarily theoretical, we have not specifically addressed any hardware optimization issues. The design of a hardware-compatible inverse halftoning algorithm based on WInHD is also a topic of interesting future study.

In Chapter 4, we introduced the problem of JPEG CHEst for color images and its potential applications. We discovered that when an image is subjected to JPEG, it coefficients no longer assume arbitrary values. JPEG leaves its signature by quantizing the image's DCT coefficients and forcing them to closely conform to periodic structures.

We first formulated a statistical framework to characterize these near-periodic structures and solve the CHEst problem for grayscale and color images. Essentially, the statistical framework chooses from a dictionary the best model, which comprises the compression history elements, that explains the regular structure of the observed image coefficients.

For special cases when affine color transformations are used by JPEG, we have formulated a blind CHEst scheme that no longer relies on a finite dictionary. We realize that, surprisingly, the DCT coefficients of a previously JPEG-compressed image actually conform to 3-D lattice structures. Our quest to understand such lattice structures helped us make a fundamentally new observation that a nearly orthogonal set of basis vectors always contains the shortest non-zero lattice vector. By exploiting such insights and by employing existing lattice algorithms, we provide a novel, blind solution to the CHEst problem.

JPEG recompression performed using the estimated compression history information introduces minimal distortion (large signal-to-noise-ratio (SNR)) and simultaneously achieves a small file-size.

An interesting question that remains unanswered is how can we extend our blind CHEst approach to also tackle non-linear color transforms. In such cases, the DCT coefficients would *locally* conform to a lattice-like structure. In fact, the mathematical tools that need to be developed to tackle such a problem are fascinating by themselves.

We also envision that with some additional research CHEst can enable a variety of intriguing applications. For example, CHEst could potentially help us uncover proprietary compression settings used by imaging devices. CHEst could also contribute to applications such as covert message passing and image authentication.

In this thesis, we tackled a variety of inverse problems. At first glance, each problem looks formidable with a myriad of solutions. The key to successfully negotiating these problems is to not only understand the structure offered by the problem but to also comprehend the structure of our desired solution set.

Appendix A

Background on Wavelets

This chapter overviews wavelet transforms, their ability to provide economical representations for a diverse class of signals and images including those with singularities [10, 37], and their utility in signal and image estimation.

A.1 1-D and 2-D Wavelet Transforms

The 1-D discrete wavelet transform (DWT) represents a 1-D continuous-time signal $x(t) \in L^2([0,1))$, $t \in [0,1)$, in terms of shifted versions of a low-pass scaling function ϕ and shifted and dilated versions of a prototype band-pass wavelet function ψ [10]. For special choices of ϕ and ψ, the functions $\psi_{j,\ell}(t) := 2^{j/2}\,\psi(2^j t - \ell)$ and $\phi_{j,\ell}(t) := 2^{j/2}\,\phi(2^j t - \ell)$ with $j, \ell \in \mathbb{Z}$ form an orthonormal basis. The j parameter corresponds to the *scale* of the analysis, while the ℓ parameter corresponds to the *location*. A finite-resolution approximation x^J to x is given by

$$x^J(t) = \sum_{\ell=0}^{N_{j_0}-1} s_{j_0,\ell}\,\phi_{j_0,\ell}(t) + \sum_{j=j_0}^{J} \sum_{\ell=0}^{N_j-1} w_{j,\ell}\,\psi_{j,\ell}(t),$$

with the scaling coefficients $s_{j_0,\ell} := \langle x, \phi_{j_0,\ell}\rangle$ and wavelet coefficients $w_{j,\ell} := \langle x, \psi_{j,\ell}\rangle$. The parameter J controls the resolution of the wavelet reconstruction x^J of x. In fact, the L_2 norm $\|x^J - x\|_2 \to 0$ as $J \to \infty$.

Multidimensional DWTs are computed by wavelet-transforming alternately along each dimen-

sion [10, 22]. The 2-D DWT represents a spatially-continuous image $x(t_1, t_2) \in L^2([0,1)^2)$ in terms of shifted versions of a low-pass scaling function $\phi_{j,\ell_1,\ell_2}(t_1, t_2) := 2^j\phi(2^j t_1 - \ell_1, 2^j t_2 - \ell_2)$, shifted and dilated versions of prototype bandpass wavelet functions $\psi^b_{j,\ell_1,\ell_2} := 2^j\psi^b(2^j t_1 - \ell_1, 2^j t_2 - \ell_2)$ with $b \in \mathcal{B} := \{LH, HL, HH\}$, where the LH, HL, and HH denote the *subbands* of the wavelet decomposition. Again, a finite-resolution approximation $x^J(t_1, t_2)$ to $x(t_1, t_2)$ is given by

$$x^J(t_1, t_2) = \sum_{\ell_1,\ell_2\in\mathbb{Z}} s_{j_0,\ell_1,\ell_2}\phi_{j_0,\ell_1,\ell_2}(t_1, t_2) + \sum_{b\in\mathcal{B}}\sum_{j=j_0}^{J}\sum_{\ell_1,\ell_2\in\mathbb{Z}} w^b_{j,\ell_1,\ell_2}\psi^b_{j,\ell_1,\ell_2}(t_1, t_2),$$

with scaling coefficients $s_{j_0,\ell_1,\ell_2} := \langle x, \phi_{j_0,\ell_1,\ell_2}\rangle$, wavelet coefficients $w^b_{j,\ell_1,\ell_2} := \langle x, \psi^b_{j,\ell_1,\ell_2}\rangle$, and J controlling the resolution of the wavelet reconstruction.

The DWT can be extended to transform sampled signals and images as well. Consider a discrete-time 1-D signal with N samples obtained as in (2.7) or a sampled image obtained by sampling $x(t_1, t_2)$ uniformly as

$$x(n_1, n_2) = N\int_{\frac{n_2}{\sqrt{N}}}^{\frac{n_2+1}{\sqrt{N}}}\int_{\frac{n_1}{\sqrt{N}}}^{\frac{n_1+1}{\sqrt{N}}} x(t_1, t_2)\, dt_1\, dt_2, \qquad 0 \le n_1, n_2 \le \sqrt{N} - 1. \tag{A.3}$$

For such N-pixel signals and images, the N wavelet coefficients can be efficiently computed in $O(N)$ operations using a filter bank consisting of low-pass filters, high-pass filters, and decimators [10]. For periodic signals, which are natural when analyzing circular convolution, filter-banks implementing circular convolution are employed.

Purely for notational convenience, we henceforth refer to the location parameters ℓ_1, ℓ_2 by ℓ and do not explicitly specify the different wavelet subbands: w^b_{j,ℓ_1,ℓ_2} and ψ^b_{j,ℓ_1,ℓ_2} for $b \in \mathcal{B} :=$

$\{LH, HL, HH\}$ will be referred to simply as $w_{j,\ell}$ and $\psi_{j,\ell}$. Further, we discuss the processing of only the wavelet coefficients, but all steps are replicated on the scaling coefficients as well.

A.2 Economy of Wavelet Representations

Wavelets provide economical representations for signals and images in smoothness spaces such as *Besov spaces* [11]. Roughly speaking, a Besov space $B^s_{p,q}$ contains functions with "s derivatives in L_p," with q measuring finer smoothness distinctions [37]. Besov spaces with different s, p, and q characterize many classes of signals in addition to L_2-Sobolev space signals; for example, $B^1_{1,1}$ contains piece-wise polynomial signals [10, 67]. Further, unlike L_2-Sobolev spaces, Besov spaces also contains images with edges [37]. The wavelet coefficients computed from samples of a continuous-time signal or image $x \in B^s_{p,q}$, $s > d\left(\frac{1}{p} - \frac{1}{2}\right)$, $1 \leq p, q \leq \infty$, with $d = 1$ for 1-D and $d = 2$ for 2-D functions, satisfy for all N

$$\frac{1}{\sqrt{N}} \left(\sum_{j \geq j_0} 2^{jq(s+d(\frac{1}{2}-\frac{1}{p}))} \left(\sum_{l} |w_{j,\ell}|^p \right)^{\frac{q}{p}} \right)^{\frac{1}{q}} < \infty, \tag{A.4}$$

assuming sufficiently smooth wavelet basis functions [8, 22, 23].[1] The condition for higher-dimensional Besov space signals is a straightforward extension of (A.4) [8, 22]. From (A.4), we can infer that the wavelet coefficients of Besov space signals decay exponentially fast with increasing scale j.

[1]The traditional Besov space characterizing equation in [8, 22, 23] assumes L_2-normalized wavelet coefficients $w_{j,\ell}$, that is, $\sum_{j,\ell} |w_{j,\ell}|^2 = \|x(t)\|_2^2$. Because the $w_{j,\ell}$ used in (A.4) are computed using signal samples $x(n)$ that satisfy $\sum_{j,\ell} |w_{j,\ell}|^2 = \sum_n |x(n)|^2 \approx N\|x(t)\|_2^2$, a normalization factor of $\sqrt{N}$ appears.

A.3 Wavelet Shrinkage-based Signal Estimation

The wavelet transform's economical representations have been exploited in many fields [10]. The wavelet transform's economical signal representation facilitates an effective solution to the problem of estimating a signal from AWGN-corrupted observations [22, 23, 27, 44]. For example, consider consider estimating 1-D signal samples $x(n)$ from noisy observations

$$\widetilde{x}(n) = x(n) + \gamma(n). \tag{A.5}$$

Simple shrinkage in the wavelet domain with scalars λ^{w} can provide excellent estimates of x.

Oracle thresholding shrinks with

$$\lambda^{\mathrm{w}}_{j,\ell} \;=\; \begin{cases} 1, & \text{if } |w_{j,\ell}| > \sigma_j, \\ 0, & \text{if } |w_{j,\ell}| \leq \sigma_j, \end{cases} \tag{A.6}$$

with σ_j^2 the noise variance at wavelet scale j. Oracle thresholding provides excellent estimation results [27] but is impractical because it assumes knowledge of the wavelet coefficients $w_{j,\ell}$ of the desired x. *Hard thresholding*, which closely approaches oracle thresholding's performance and is also practical [23], employs

$$\lambda^{\mathrm{w}}_{j,\ell} \;=\; \begin{cases} 1, & \text{if } |\widetilde{w}_{j,\ell}| > \rho_j \sigma_j, \\ 0, & \text{if } |\widetilde{w}_{j,\ell}| \leq \rho_j \sigma_j, \end{cases} \tag{A.7}$$

with $\widetilde{w}_{j,\ell} := \langle \widetilde{x}\,, \psi_{j,\ell} \rangle$ and ρ_j a scale-dependent threshold factor (see [10, p. 442] for choices of ρ_j). When the underlying continuous-time $x(t) \in B^s_{p,q}$ with $s > d\left(\frac{1}{p} - \frac{1}{2}\right)$ and $1 \leq p, q \leq \infty$,

both oracle and hard thresholding (with judiciously chosen ρ_j [44]) provide estimates whose MSE-per-sample decays at least as fast as $N^{\frac{-2s}{2s+1}}$ with increasing number of samples $N \to \infty$ [22, 23]. Further, no estimator can achieve a better error decay rate for every $x \in B^s_{p,q}$. If the threshold factor ρ_j is chosen to be scale-independent, then the MSE decay rate is decelerated by an additional $\log N$ factor.

In practice, the *Wavelet-domain Wiener Filter* (WWF) improves on the MSE performance of hard thresholding by employing Wiener estimation on each wavelet coefficient [68]. WWF chooses

$$\lambda^{\mathrm{w}}_{j,\ell} = \frac{|w_{j,\ell}|^2}{|w_{j,\ell}|^2 + \sigma_j^2}. \tag{A.8}$$

However, like in oracle thresholding, the coefficients $w_{j,\ell}$ required to construct the $\lambda^{\mathrm{w}}_{j,\ell}$ are unknown. Hence, a "pilot" estimate of the unknown signal is first computed using hard thresholding (with, say, $\rho_j = 3$ for 256×256 images). Then, using λ^{w} constructed with the pilot estimate's wavelet coefficients in (A.8), WWF shrinkage is performed. Sufficiently different wavelet basis functions must be used in the two steps [68].

Appendix B

Formal WVD Algorithm

We briefly review the WVD algorithm as applied to deconvolve discrete-time circular convolution operators $\mathcal{H}$ [8]. WVD relies on functionals called *vaguelettes* $u_{j,\ell}$ to simultaneously invert $\mathcal{H}$ and compute the wavelet transform. The $u_{j,\ell}$ act on the noiseless data $x \circledast h$ to yield the wavelet coefficients $w_{j,\ell}$ of the signal x [8]

$$\langle x \circledast h, \kappa_j \, u_{j,\ell} \rangle := \langle x, \psi_{j,\ell} \rangle = w_{j,\ell}. \tag{B.1}$$

Here κ_j is a scale-dependent parameter that normalizes the vaguelette norm $\|u_{j,\ell}\|_2$. For example, $\kappa_j \approx 2^{j\nu}$, when $|H(f_k)| \propto (|k|+1)^{-\nu}$ [8]. Since inner products are preserved under orthogonal transformations, (B.1) can be re-written using the Karhunen-Loeve transform for discrete-time circular convolution (the DFT) as

$$\langle HX, \, \kappa_j \, U_{j,\ell} \rangle = \langle X, \Psi_{j,\ell} \rangle, \tag{B.2}$$

with $\Psi_{j,\ell}(f_k)$ and $U_{j,\ell}(f_k)$ denoting the respective DFT representations of $\psi_{j,\ell}$ and $u_{j,\ell}$. Since (B.2) holds for any x, we can infer that each DFT component of $u_{j,\ell}$ can be expressed as

$$U_{j,\ell}(f_k) = \kappa_j^{-1} \frac{\Psi_{j,\ell}(f_k)}{\overline{H}(f_k)} \tag{B.3}$$

with $\overline{H}(f_k)$ the complex conjugate of $H(f_k)$.

The WVD employs the vaguelettes $u_{j,\ell}$ to perform deconvolution as follows:

1) *Project the observation onto the vaguelettes to compute the noisy wavelet coefficients.*

Compute the wavelet coefficients $\widetilde{w}_{j,\ell}$ of the noisy $\widetilde{x}$ in (2.2) as (see (B.1))

$$\widetilde{w}_{j,\ell} \;=\; \langle y, \kappa_j\, u_{j,\ell}\rangle = w_{j,\ell} + \kappa_j\, \langle \gamma, u_{j,\ell}\rangle. \tag{B.4}$$

2) *Shrink the noisy wavelet coefficients.*

Compute $\widetilde{w}_{j,\ell;\lambda^{\mathrm{w}}} := \widetilde{w}_{j,\ell}\lambda^{\mathrm{w}}_{j,\ell}$ using shrinkage $\lambda^{\mathrm{w}}_{j,\ell}$. For example, employ hard thresholding (A.7) [8, 27] with the σ_j^2 computed as in (2.16) but with all $\lambda^{\mathrm{f}}_k = 1$.

3) *Invert the wavelet transform to compute the WVD estimate.*

Reconstruct the WVD estimate as $\widetilde{x}_{\lambda^{\mathrm{w}}} = \sum_{j,\ell} \widetilde{w}_{j,\ell;\lambda^{\mathrm{w}}}\, \psi_{j,\ell}$.

Thus, the WVD algorithm performs deconvolution by first inverting the convolution operator and then employing scalar shrinkage in the wavelet domain.

Appendix C

Derivation of Optimal Regularization Parameters for ForWaRD

Our goal is to prove Proposition 1. We will find the optimal regularization parameter $\alpha_j^\star$ by differentiating $\widetilde{\mathrm{MSE}}_j(\alpha_j)$ from (2.18) with respect to α_j and setting the derivative equal to zero.

The $\widetilde{\mathrm{MSE}}_j(\alpha_j)$ in (2.18) can be rewritten as

$$\widetilde{\mathrm{MSE}}_j(\alpha_j) = N_j \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} \frac{\alpha_j^2\, N\sigma^4 |X(f_k)|^2 |\Psi_{j,\ell}(f_k)|^2}{(|H(f_k)|^2|X(f_k)|^2 \;+\; \alpha_j N\sigma^2)^2} + \sum_{\ell=0}^{N_j-1} \min\Big(|w_{j,\ell}|^2,\, \sigma^2_{j;\lambda^{\mathrm{f}}(\alpha_j)}\Big). \quad \text{(C.1)}$$

Differentiating the first term in (C.1) with respect to α_j, we have

$$\frac{d}{d\alpha_j}\left(N_j \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} \frac{\alpha_j^2 N\sigma^4 |X(f_k)|^2 |\Psi_{j,\ell}(f_k)|^2}{(|H(f_k)|^2|X(f_k)|^2 \;+\; \alpha_j N\sigma^2)^2} \right)$$
$$= N_j \alpha_j \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} \frac{2\sigma^4 N |H(f_k)|^2 |X(f_k)|^4 |\Psi_{j,\ell}(f_k)|^2}{(|H(f_k)|^2|X(f_k)|^2 + \alpha_j N\sigma^2)^3}. \quad \text{(C.2)}$$

Differentiating the second term in (C.1) with respect to α_j, we have for almost every $\alpha_j > 0$ (in the measure-theoretic sense)

$$\frac{d}{d\alpha_j}\left(\sum_{\ell=0}^{N_j-1} \min\Big(|w_{j,\ell}|^2,\, \sigma^2_{j;\lambda^{\mathrm{f}}(\alpha_j)}\Big) \right) = \#\left\{ |w_{j,\ell}| > \sigma_{j;\lambda^{\mathrm{f}}(\alpha_j)} \right\} \frac{d\sigma^2_{j;\lambda^{\mathrm{f}}(\alpha_j)}}{d\alpha_j}. \quad \text{(C.3)}$$

Using (2.16) with $\Lambda(f_k) = \alpha_j \frac{N\sigma^2}{|X(f_k)|^2}$, we have

$$\frac{d\sigma^2_{j;\lambda^{\mathrm{f}}(\alpha_j)}}{d\alpha_j} = -\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} \frac{2\sigma^4 N|H(f_k)|^2|X(f_k)|^4|\Psi_{j,\ell}(f_k)|^2}{(|H(f_k)|^2|X(f_k)|^2 + \alpha_j N\sigma^2)^3}. \tag{C.4}$$

Hence, from (C.3) and (C.4), for almost every $\alpha_j > 0$, we have

$$\frac{d}{d\alpha_j}\left(\sum_{\ell=0}^{N_j-1} \min\Big(|w_{j,\ell}|^2,\, \sigma^2_{j;\lambda^{\mathrm{f}}(\alpha_j)}\Big)\right) = -\#\left\{|w_{j,\ell}| > \sigma_{j;\lambda^{\mathrm{f}}(\alpha_j)}\right\} \times \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} \frac{2\sigma^4|H(f_k)|^2|X(f_k)|^4|\Psi_{j,\ell}(f_k)|^2}{(|H(f_k)|^2|X(f_k)|^2 + \alpha_j\sigma^2)^3}. \tag{C.5}$$

The terms obtained by differentiating $\widetilde{\mathrm{MSE}}_j(\alpha_j)$ from (C.1) with respect to α_j are given by (C.2) and (C.5). Setting the derivative of $\widetilde{\mathrm{MSE}}_j(\alpha_j)$ to zero and denoting the satisfying solution by $\alpha_j^\star$, we have

$$N_j\,\alpha_j^\star - \#\left\{|w_{j,\ell}| > \sigma_{j;\lambda^{\mathrm{f}}(\alpha_j^\star)}\right\} = 0 \tag{C.6}$$

which yields the expression (2.19) for the optimal regularization parameter. □

Appendix D

Decay Rate of Wavelet Shrinkage Error in ForWaRD

Here we will bound the asymptotic error (2.21) in estimating the signal part $x_{\lambda^{\mathrm{f}}}$ retained during Fourier shrinkage via wavelet scalar shrinkage. The estimation problem solved by the wavelet shrinkage step in ForWaRD (see Step 2 in Section 2.5.1) is the following: Estimate the retained signal $x_{\lambda^{\mathrm{f}}}(n)$ from the noisy observation (see also (2.12))

$$\widetilde{x}_{\lambda^{\mathrm{f}}}(n) = x_{\lambda^{\mathrm{f}}}(n) + \mathcal{H}^{-1}\gamma_{\lambda^{\mathrm{f}}}(n). \tag{D.1}$$

To deduce (2.21), we first justify in Appendix D.1 that the continuous-time retained signal $x_{\lambda^{\mathrm{f}}}(t) \in B^s_{p,q}$ when the desired signal $x(t) \in B^s_{p,q}$. Then, in Appendix D.2, we prove (2.21) by invoking established bounds on the MSE performance of wavelet-domain scalar estimation of $B^s_{p,q}$ signals observed in white Gaussian noise [22, 23].

D.1 Besov Smoothness of Distorted Signal

We will show that if $x(t) \in B^s_{p,q}$, then for a wide variety of $\mathcal{H}$, including those with a smooth frequency response, $x_{\lambda^{\mathrm{f}}}(t) \in B^s_{p,q}$. The $x_{\lambda^{\mathrm{f}}}(t)$ is obtained by the action on $x(t)$ of a circular convolution operator $\mathcal{D}$ with frequency response $D(f_k) := \frac{|H(f_k)|^2}{|H(f_k)|^2+\tau}$. Consider the variation of $D(f_k)$ over dyadic frequency intervals—defined as $\sum_{k\in(2^j,2^{j+1}]} |D(f_{k+1}) - D(f_k)|$. When the variation of $D(f_k)$ over each dyadic interval $k \in (2^j, 2^{j+1}]$ is bounded, then $\mathcal{D}$ lies in the set

$\mathcal{C}_{L_p}$ of operators that map an L_p signal to another L_p signal according to the Marcinkeiwicz's Multiplier Theorem [69, pg. 148]. From [70, pg. 131–132, Theorems 3 and 4] the set $\mathcal{C}_{L_p} \subset \mathcal{C}_{B^s_{p,q}}$, $1 < p < \infty$, with $\mathcal{C}_{B^s_{p,q}}$ denoting the set of operators that map any $B^s_{p,q}$ signal into another $B^s_{p,q}$ signal. Hence, we can infer that if $\mathcal{D}$'s frequency response has bounded variation over dyadic intervals, then $x_{\lambda^{\mathrm{f}}}(t) \in B^s_{p,q}$. Further, it is easy to show that if the squared-magnitude frequency response $|H(f_k)|^2$ enjoys bounded variation over dyadic intervals, then so does $\mathcal{D}$'s frequency response. The bounded variation condition is simply a smoothness constraint. Hence, we can infer from the previous argument that if the frequency response of $\mathcal{H}$ is smooth and if $x(t) \in B^s_{p,q}$, then $x_{\lambda^{\mathrm{f}}}(t) \in B^s_{p,q}$. For many other $\mathcal{D}$ also, $x_{\lambda^{\mathrm{f}}}(t) \in B^s_{p,q}$. The rich set $\mathcal{C}_{B^s_{p,q}}$ of $\mathcal{D}$ can be precisely characterized by the necessary and sufficient condition in [70, pg. 132, Theorem 4]. Hence the retained signal $x_{\lambda^{\mathrm{f}}}(t) \in B^s_{p,q}$ when $x(t) \in B^s_{p,q}$.

D.2 Wavelet-domain Estimation Error: ForWaRD vs. Signal in White Noise

The estimation problem (D.1) is similar to the well-studied setup (A.5) of signal estimation in white noise but with colored corrupting noise $\mathcal{H}^{-1}\gamma_{\lambda^{\mathrm{f}}}$. The variance of $\mathcal{H}^{-1}\gamma_{\lambda^{\mathrm{f}}}$ is bounded at all wavelet scales because we can easily infer from (2.16) that for Fourier-Tikhonov shrinkage

$$\sigma^2_{j;\lambda^{\mathrm{f}}} \leq \frac{\sigma^2}{4 \min_{f_k} \Lambda(f_k)} = \frac{\sigma^2}{4\tau}. \tag{D.2}$$

Because the estimation error due to wavelet thresholding is monotone with respect to the noise variance [8], the error in estimating $x_{\lambda^{\mathrm{f}}}$ from (D.1) using wavelet-domain scalar thresholding is less than the error in estimating $x_{\lambda^{\mathrm{f}}}$ when observed in white noise of variance $\frac{\sigma^2}{4\tau}$. Further, $x_{\lambda^{\mathrm{f}}}(t) \in B^s_{p,q}$, $s > \frac{1}{p} - \frac{1}{2}$, $1 < p, q < \infty$, from Appendix D.1. Hence, the per-sample MSE in estimating $x_{\lambda^{\mathrm{f}}}$

from (D.1) can be bounded with the decay rate $N^{\frac{-2s}{2s+1}}$ established for the white noise setup (see Section A.3). This yields (2.21) with constant $C_2 > 0$. □

Appendix E

Decay Rate of Total ForWaRD MSE

Our proof of Proposition 3 proceeds by individually bounding the wavelet shrinkage error and the Fourier distortion error. The $C_k > 0$ with different k's denote constants in the proof.

E.1 Bounding Wavelet Shrinkage Error

It is straightforward to infer that the per-sample wavelet shrinkage error in ForWaRD decays at least as fast as the WVD error, that is,

$$\frac{1}{N}\,\mathbb{E}\left(\sum_{n=0}^{N-1}|x_{\lambda^{\mathrm{f}}}(n)-\widehat{x}(n)|^2\right) \leq C_5\, N^{\frac{-2s}{2s+2\nu+1}}. \tag{E.1}$$

This follows because firstly, for any $\tau \geq 0$, the noise variance encountered by wavelet shrinkage in ForWaRD at all scales is less than or equal to that encountered in WVD for the same setup. Further, for reasons similar to those outlined in Appendix D.1, $x_{\lambda^{\mathrm{f}}} \in B^s_{p,q}$.

E.2 Bounding Fourier Distortion Error

The per-sample Fourier distortion error, which we will now bound, can be expressed as

$$\frac{1}{N}\sum_{n=0}^{N-1}|x(n)-x_{\lambda^{\mathrm{f}}}(n)|^2 = \frac{1}{N^2}\sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}}\frac{\tau^2|X(f_k)|^2}{(|H(f_k)|^2+\tau)^2} \leq 2\left(\frac{1}{N^2}\sum_{k=0}^{\frac{N}{2}}\frac{\tau^2|X(f_k)|^2}{(|H(f_k)|^2+\tau)^2}\right). \tag{E.2}$$

Since this error increases monotonically with τ, we merely need to show that for $\tau = N^{-\beta}$, the error decays like $N^{\frac{-2s}{2s+2\nu+1}}$. Setting $\tau = N^{-\beta}$ and $|H(f_k)| = (|k|+1)^{-\nu}$ in (E.2), we have

$$\begin{aligned}
\frac{1}{N^2}\sum_{k=0}^{\frac{N}{2}} \frac{N^{-2\beta}|X(f_k)|^2}{((|k|+1)^{-2\nu}+N^{-\beta})^2} \;&\leq\; \frac{1}{N^2}\left(\sum_{k=0}^{N^{\frac{\beta}{2\nu}}-1} \frac{N^{-2\beta}|X(f_k)|^2}{((|k|+1)^{-2\nu}+N^{-\beta})^2}\right. \\
&\qquad\left. + \sum_{k=N^{\frac{\beta}{2\nu}}}^{\frac{N}{2}} \frac{N^{-2\beta}|X(f_k)|^2}{((|k|+1)^{-2\nu}+N^{-\beta})^2}\right) \\
&\leq\; \frac{1}{N^2}\left(\sum_{k=0}^{N^{\frac{\beta}{2\nu}}-1} N^{-2\beta}(|k|+1)^{4\nu}|X(f_k)|^2\right. \\
&\qquad\left. + \sum_{k=N^{\frac{\beta}{2\nu}}}^{\frac{N}{2}} |X(f_k)|^2\right). \qquad \text{(E.3)}
\end{aligned}$$

The second summation in (E.3) captures the total energy of the high-frequency components of $x(t)$ convolved with the sampling kernel. For any signal, the total energy of the high-frequency components can be bounded using the energy of the signal's fine-scale Shannon or Meyer wavelet coefficients [10]. The energy of any $B^s_{p,q}$ signal's fine-scale wavelet coefficients can in turn be bounded using [23, Lemma 2.2]. The $x(t)$ convolved with typical sampling kernels $\in B^s_{p,q}$ (for the same reasons outlined in Appendix D.1). Hence we can bound the second summation in (E.3) using [23, Lemma 2.2] and then using (2.23) as

$$\frac{1}{N^2}\sum_{k=N^{\frac{\beta}{2\nu}}}^{\frac{N}{2}} |X(f_k)|^2 \;\leq\; C_6\left(N^{\frac{\beta}{2\nu}}\right)^{-\min\left(2s,2s+1-\frac{2}{p}\right)} \;\leq\; C_6\, N^{\frac{-2s}{2s+2\nu+1}}. \qquad \text{(E.4)}$$

The zero-frequency term of the first summation in (E.3) can also be easily bounded using (2.23) as

$$\left(\frac{1}{N^2}N^{-2\beta}|X(f_0)|^2\right) \leq \|x(t)\|_2^2\, N^{-2\beta} \leq C_7 N^{\frac{-2s}{2s+2\nu+1}}. \tag{E.5}$$

The non-zero frequency terms of the first summation in (E.3) can be written as

$$\begin{aligned}
\frac{1}{N^2}\sum_{k=1}^{N^{\frac{\beta}{2\nu}}-1} N^{-2\beta}(|k|+1)^{4\nu}|X(f_k)|^2 &= \frac{1}{N^2}\sum_{j=0}^{\log_2(N^{\frac{\beta}{2\nu}})-1}\sum_{k=2^j}^{2^{j+1}-1} N^{-2\beta}(|k|+1)^{4\nu}|X(f_k)|^2 \\
&\leq \frac{N^{-2\beta}}{N^2}\sum_{j=0}^{\log_2(N^{\frac{\beta}{2\nu}})-1} 2^{4(j+1)\nu}\sum_{k=2^j}^{\frac{N}{2}}|X(f_k)|^2 \\
&\leq C_8\, N^{-2\beta}\sum_{j=0}^{\log_2(N^{\frac{\beta}{2\nu}})-1} 2^{4(j+1)\nu}2^{-j\min\left(2s,2s+1-\frac{2}{p}\right)} \\
&\leq C_9 \times \begin{cases} N^{-2\beta}\log_2(N^{\frac{\beta}{2\nu}}), \\ \qquad \text{if } 4\nu \leq \min\left(2s, 2s+1-\frac{2}{p}\right), \\ N^{-\frac{\beta}{2\nu}\left(\min\left(2s,2s+1-\frac{2}{p}\right)\right)}, \quad \text{otherwise} \end{cases} \\
&\leq C_{10}\, N^{\frac{-2s}{2s+2\nu+1}}, \ \ (\text{using (2.23)}).
\end{aligned} \tag{E.6}$$

Using (E.2)–(E.6), we can thus infer that the Fourier shrinkage term also decays as $N^{\frac{-2s}{2s+2\nu+1}}$ with increasing N. Since the total ForWaRD MSE can be bounded using twice the sum of the wavelet shrinkage error and the Fourier distortion, we can infer (2.24). Further, since the ForWaRD MSE decay rate matches the WVD MSE decay rate (see (2.15)), which is optimal for this setup (see Section 2.4.2), we can also infer that no estimator can achieve a faster MSE decay rate than ForWaRD for every $x(t) \in B^s_{p,q}$. □

Appendix F

Decay Rate of WInHD's MSE

We deduce the asymptotic performance of WInHD as claimed in Proposition 4.

Instead of analyzing the problem of estimating $x(n_1, n_2)$ from $y(n_1, n_2)$, we can equivalently analyze the estimation of $x(n_1, n_2)$ from the noisy observation $\widetilde{x}(n_1, n_2)$ obtained after inverting $\mathcal{P}$ (see (3.4)). The reduction is equivalent because $P(f_1, f_2)$ is known and invertible (since $|P(f_1, f_2)| \geq \epsilon > 0$).[1]

The frequency components of the colored noise $\mathcal{P}^{-1}\mathcal{Q}\gamma(n_1, n_2)$ corrupting the $\widetilde{x}(n_1, n_2)$ in (3.4) is given by $\frac{Q(f_1,f_2)\Gamma(f_1,f_2)}{P(f_1,f_2)}$. These frequency components are independent and Gaussian because the Fourier transform diagonalizes convolution operators. Since $|P(f_1, f_2)|$ is strictly non-zero and $|Q(f_1, f_2)|$ is bounded, the variance of $\frac{Q(f_1,f_2)\Gamma(f_1,f_2)}{P(f_1,f_2)}$ is uniformly bounded — say with variance ς^2 — at all frequencies.

Because the estimation error due to wavelet-domain hard thresholding is monotone with respect to noise variance [8], the error in estimating $x(n_1, n_2)$ from (3.4) using wavelet-domain hard thresholding is less than the error in estimating $x(n_1, n_2)$ observed in white noise as in (A.5) but with variance ς^2. Hence the per-pixel MSE in estimating $x(n_1, n_2)$ from (3.4) can be bounded with the decay rate $N^{\frac{-s}{s+1}}$ established for the white noise setup (see Section A.3) to yield (3.9) with a constant $C > 0$ independent of N [23, 44]. □

[1]Since the filter $\mathcal{P}^{-1}$ is FIR for error diffusion systems, boundary effects are negligible asymptotically because only a finite number of boundary pixels are corrupted.

Appendix G

Properties of Nearly Orthogonal Basis Vectors

We will now prove Propositions 5 and 6.

G.1 Proof of Proposition 5

Our approach will be first prove Proposition 5 for 2-D lattices and then tackle the proof for higher dimensional lattices via induction.

G.1.1 Proof for 2-D lattices

Consider a lattice with basis vectors b_1 and b_2. By rotating the lattice, the basis vectors can be expressed as the columns of

$$\begin{bmatrix} \|b_1\|_2 & \|b_2\|_2 \cos(\theta) \\ 0 & \|b_2\|_2 \sin(\theta) \end{bmatrix},$$

with θ the angle between b_1 and b_2. Any non-zero vector in the lattice can be expressed as

$$\begin{bmatrix} \|b_1\|_2 & \|b_2\|_2 \cos(\theta) \\ 0 & \|b_2\|_2 \sin(\theta) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} u_1\|b_1\|_2 + u_2\|b_2\|_2 \cos(\theta) \\ u_2\|b_2\|_2 \sin(\theta) \end{bmatrix}$$

where $u_1, u_2 \in \mathbb{Z}$ and $|u_1| + |u_2| > 0$. The squared-length of the lattice vector is equal to

$$\begin{aligned}
&\left(u_1\|b_1\|_2 + u_2\|b_2\|_2\cos(\theta)\right)^2 + \left(u_2\|b_2\|_2\sin(\theta)\right)^2 \\
&\quad = |u_1|^2\|b_1\|_2^2 + |u_2|^2\|b_2\|_2^2 + 2u_1u_2\|b_1\|_2\|b_2\|_2\cos(\theta) \\
&\quad \geq |u_1|^2\|b_1\|_2^2 + |u_2|^2\|b_2\|_2^2 - 2|u_1||u_2|\|b_1\|_2\|b_2\|_2\cos\left(\frac{\pi}{3}\right) \\
&\quad = \left(|u_1|\|b_1\|_2 - |u_2|\|b_2\|_2\right)^2 + |u_1||u_2|\|b_1\|_2\|b_2\|_2 \\
&\quad \geq \min\left(\|b_1\|_2^2,\ \|b_2\|_2^2\right),
\end{aligned}$$

with equality possible only when $\theta = \frac{\pi}{3}$ or $\frac{2\pi}{3}$. This proves Proposition 5 for 2-D lattices.

G.1.2 Proof for higher dimensional lattices

For the sake of induction, assume that Proposition 5 holds true for the $(k-1)$-dimensional lattice with basis vectors $b_1, b_2, \ldots, b_{k-1}$. We need to prove that Proposition 5 will hold true for the k-dimensional lattice spanned $b_1, b_2, \ldots, b_k$.

Consider any non-zero vector $\sum_{i=1}^{k} u_i\, b_i$, $u_i \neq 0$ for some $i = 1, \ldots, k$, contained by the k-dimensional lattice. If $u_k = 0$, then $\sum_{i=1}^{k} u_i\, b_i$ is contained by the $(k-1)$-dimensional lattice. By our assumption for the $(k-1)$-dimensional lattice, we have

$$\left\|\sum_{i=1}^{k} u_i\, b_i\right\|_2 = \left\|\sum_{i=1}^{k-1} u_i\, b_i\right\|_2 \geq \min_{j\in 1,\ldots,k-1} \|b_j\|_2 \geq \min_{j\in 1,\ldots,k} \|b_j\|_2 .$$

If $u_k \neq 0$ and $u_i = 0$ for $i = 1, \ldots, k-1$, then again

$$\left\|\sum_{i=1}^{k} u_i\, b_i\right\|_2 \geq \|b_k\|_2 \geq \min_{j\in 1,\ldots,k} \|b_j\|_2 .$$

Hence assume that $u_k \neq 0$ and $u_i \neq 0$ for some $i = 1, \ldots, k-1$. Now $\sum_{i=1}^{k} u_i\, b_i$ is contained by the lattice with basis vectors $\sum_{i=1}^{k-1} u_i\, b_i$ and $u_k b_k$. Since the ordered set $\{b_1, b_2, \ldots, b_k\}$ is weakly $\left(\frac{\pi}{3} + \epsilon\right)$-orthogonal, the angle between the non-zero vectors $\sum_{i=1}^{k-1} u_i\, b_i$ and $u_k b_k$ lies in the closed interval $\left[\frac{\pi}{3} + \epsilon, \frac{2\pi}{3} - \epsilon\right]$. Invoking Proposition 5, which holds true for 2-D lattices, we have

$$\begin{aligned} \left\| \sum_{i=1}^{k} u_i\, b_i \right\|_2 &\geq \min\left(\left\| \sum_{i=1}^{k-1} u_i\, b_i \right\|_2,\ \|u_k b_k\|_2 \right) \\ &\geq \min\left(\min_{j \in 1,\ldots,k-1} \|b_j\|_2,\ \|u_k b_k\|_2 \right) \\ &\geq \min_{j \in 1,\ldots,k} \|b_j\|_2 \end{aligned} \tag{G.3}$$

with equality possible only if $\epsilon = 0$. Thus, the set of basis vectors $\{b_1, b_2, \ldots, b_k\}$ contains the shortest non-zero vector in the k-dimensional lattice. By induction, the proof for Proposition 5 is now complete. □

G.2 Proof of Proposition 6

Similar to the proof for Proposition 5, our approach will be to first prove Proposition 6 for 2-D lattices. We will tackle the proof for higher dimensional lattices by contradiction. Before proceeding further, note that $\frac{\pi}{3} < \theta < \frac{2\pi}{3}$ to ensure that $\eta(\theta) > 1$.

G.2.1 Proof for 2-D lattices

Consider a lattice with basis vectors b_1 and b_2. Without loss of generality (WLOG), assume that $1 = \|b_1\|_2 \leq \|b_2\|_2$. By rotating the 2-D lattice, the basis vectors can be expressed as the columns

of

$$\begin{bmatrix} 1 & \|b_2\|_2 \cos(\theta) \\ 0 & \|b_2\|_2 \sin(\theta) \end{bmatrix},$$

with θ the angle between b_1 and b_2. Let $\left\{\widetilde{b}_1, \widetilde{b}_2\right\}$ denote another strongly $\frac{\pi}{3}$-orthogonal set of basis vectors for the same 2-D lattice. Then, using Proposition 5, both $\{b_1, b_2\}$ and $\left\{\widetilde{b}_1, \widetilde{b}_2\right\}$ contain the shortest vector in the lattice, namely $\pm b_1$. [1] Hence we can express

$$\begin{bmatrix} \widetilde{b}_1 & \widetilde{b}_2 \end{bmatrix} = \begin{bmatrix} 1 & \|b_2\|_2 \cos(\theta) \\ 0 & \|b_2\|_2 \sin(\theta) \end{bmatrix} \begin{bmatrix} \pm 1 & u \\ 0 & \pm 1 \end{bmatrix}, \qquad \text{with } u \in \mathbb{Z}.$$

Then, the smallest angle between $\widetilde{b}_1$ and $\pm\widetilde{b}_2$ is given by

$$\tan^{-1}\left(\left|\frac{\|b_2\|_2 \sin(\theta)}{\|b_2\|_2 \cos(\theta) \pm u}\right|\right).$$

If $\|b_2\|_2 < \eta(\theta)$, then it is straightforward to verify that for all $u \neq 0$

$$\left|\frac{\|b_2\|_2 \sin(\theta)}{\|b_2\|_2 \cos(\theta) \pm u}\right|^2 < \tan^2\left(\frac{\pi}{3}\right) = 3,$$

by analyzing the roots of the quadratic equation in $\|b_2\|_2$. Thus, to ensure that $\left\{\widetilde{b}_1, \widetilde{b}_2\right\}$ is strongly $\frac{\pi}{3}$-orthogonal, $u = 0$. This proves Proposition 6 for 2-D lattices.

[1] $\left\{\widetilde{b}_1, \widetilde{b}_2\right\}$ can contain $\pm b_2$ but not $\pm b_1$ only if $\|b_1\|_2 = \|b_2\|_2$. In such a case, we can interchange b_1 and b_2 WLOG and proceed further.

G.2.2 Proof for higher dimensional lattices

Let $\mathcal{B}$ be a matrix as described in Proposition 6. Let $\widetilde{\mathcal{B}} = \left[\widetilde{b}_1\ \widetilde{b}_2\ \ldots\ \widetilde{b}_k\right]$ be another matrix whose columns form a basis for the lattice spanned by $\mathcal{B}$. Then, we can express $\widetilde{\mathcal{B}} = \mathcal{B}\mathcal{U}$, with $\mathcal{U}$ an integer matrix such that its determinant $= \pm 1$. Let us further assume that $\widetilde{\mathcal{B}}$'s columns are strongly $\frac{\pi}{3}$-orthogonal. Then, we need to prove that $\mathcal{U}$ is a permutation of a diagonal matrix with the non-zero elements $= \pm 1$. We only need to show that each columns of $\mathcal{U}$ has just one non-zero entry; the determinant$(\mathcal{U}) = \pm 1$ condition will ensure that the non-zero entries are $= \pm 1$.

Assume, for the sake of contradiction, that $\mathcal{U}$ contains atleast one column vector with multiple non-zero entries. WLOG, by appropriately rearranging the columns of $\mathcal{B}$ and $\widetilde{\mathcal{B}}$, we can ensure that $\mathcal{U}$'s first column's elements $u_{j,1} \neq 0$ only for $j = 1, \ldots, \ell$, $2 \leq \ell \leq m$ and $u_{j,1} = 0$ for $j > \ell$; $u_{j,i}$ denotes the element from $\mathcal{U}$'s j-th row and i-th column.

We will now construct a 2-D lattice subspace that will be spanned by two sets of basis vectors; one set comprises integer combinations of $\mathcal{B}$'s columns and the second set of $\widetilde{\mathcal{B}}$'s columns. From our assumption in the previous paragraph, we can express

$$\widetilde{b}_1 = \sum_{j=1}^{\ell} u_{j,1}\, b_j. \tag{G.8}$$

Further, since the columns of $\widetilde{\mathcal{B}}$ form a basis, there exist $\widetilde{u}_{j,i} \in \mathbb{Z}$, with j and $i \in \{1, \ldots, m\}$, such

that

$$\begin{aligned}
\sum_{j=1}^{m} \widetilde{u}_{j,k}\,\widetilde{b}_j &= b_k \\
\sum_{j=2}^{m} \widetilde{u}_{j,k}\,\widetilde{b}_j &= b_k - \widetilde{u}_{1,k}\widetilde{b}_1 \\
&= (1-\widetilde{u}_{1,k}u_{k,1})\,b_k - \widetilde{u}_{1,k}\left(\widetilde{b}_1 - u_{k,1}b_k\right) \\
&= (1-\widetilde{u}_{1,k}u_{k,1})\,b_k - \widetilde{u}_{1,k}\sum_{\substack{j=1\\ j\neq k}}^{\ell} u_{j,1}\,b_j \qquad \text{(using (G.8))}. \qquad \text{(G.9)}
\end{aligned}$$

Using (G.9), we have

$$\begin{bmatrix} b_k & \sum_{\substack{j=1\\ j\neq k}}^{\ell} u_{j,1}\,b_j \end{bmatrix} \begin{bmatrix} u_{k,1} & 1-\widetilde{u}_{1,k}\,u_{k,1} \\ 1 & -\widetilde{u}_{1,k} \end{bmatrix} = \begin{bmatrix} \widetilde{b}_1 & \sum_{j=2}^{m} \widetilde{u}_{j,k}\,\widetilde{b}_j \end{bmatrix}$$

Denoting

$$\begin{aligned}
\mathcal{B}_k &:= \begin{bmatrix} b_k & \sum_{\substack{j=1\\ j\neq k}}^{\ell} u_{j,1}\,b_j \end{bmatrix} \\
\mathcal{U}_k &:= \begin{bmatrix} u_{k,1} & 1-\widetilde{u}_{1,k}\,u_{k,1} \\ 1 & -\widetilde{u}_{1,k} \end{bmatrix} \\
\widetilde{\mathcal{B}}_k &:= \begin{bmatrix} \widetilde{b}_1 & \sum_{j=2}^{m} \widetilde{u}_{j,k}\,\widetilde{b}_j \end{bmatrix},
\end{aligned}$$

we have $\mathcal{B}_k\,\mathcal{U}_k = \widetilde{\mathcal{B}}_k$. Because the determinant of the integer matrix $\mathcal{U}_k$ is equal to -1, the columns of both $\mathcal{B}_k$ and $\widetilde{\mathcal{B}}_k$ span the same lattice subspace, which we denote as $\mathcal{L}_k$.

We can exploit the properties of $\mathcal{B}$ and $\widetilde{\mathcal{B}}$ to deduce the range of values that the elements of $\mathcal{U}_k$

can assume. Clearly, the columns of $\mathcal{B}_k$ and $\widetilde{\mathcal{B}}_k$ enjoy the strong θ-orthogonality of $\mathcal{B}$'s columns and the strong $\frac{\pi}{3}$-orthogonality of $\widetilde{\mathcal{B}}$'s columns respectively. Hence, by invoking Proposition 5, we can infer that the columns of both $\mathcal{B}_k$ and $\widetilde{\mathcal{B}}_k$ contain the shortest vector in the lattice $\mathcal{L}_k$. Further, since $u_{k,1} \neq 0$ and $\frac{\pi}{3} < \theta < \frac{2\pi}{3}$, $\widetilde{b}_1 = \left(u_{k,1}\, b_k + \sum_{\substack{j=1 \\ j\neq k}}^{\ell} u_{j,1}\, b_j \right)$ cannot be the shortest vector in $\mathcal{L}_k$. Consequently, $\sum_{j=2}^{m} \widetilde{u}_{j,k}\, \widetilde{b}_j$ must be the shortest vector and either

$$\sum_{j=2}^{m} \widetilde{u}_{j,k} \widetilde{b}_j = b_k, \quad \Rightarrow \quad \mathcal{U}_k = \begin{bmatrix} u_{k,1} & 1 \\ \\ 1 & 0 \end{bmatrix} \tag{G.11}$$

or

$$\sum_{j=2}^{m} \widetilde{u}_{j,k} \widetilde{b}_j = \pm \left(\widetilde{b}_1 - u_{k,1} b_k \right) \quad \Rightarrow \quad \mathcal{U}_k = \begin{bmatrix} \pm 1 & 0 \\ \\ 1 & \pm 1 \end{bmatrix} \tag{G.12}$$

We claim that (G.12) holds for some $k \in 1, \ldots, \ell$. Otherwise, if (G.11) holds true for all $k = 1, \ldots, \ell$, then

$$\sum_{k=1}^{\ell} u_{k,1} \left(\sum_{j=2}^{m} \widetilde{u}_{j,k}\, \widetilde{b}_j \right) = \sum_{k=1}^{\ell} u_{k,1} b_k = \widetilde{b}_1 \quad (\text{using (G.8)}).$$

This creates a contradiction because $\left\{ \widetilde{b}_1, \ldots, \widetilde{b}_m \right\}$ are independent of each other. Hence, there exists a $k \in 1, \ldots, \ell$ such that (G.12) holds.

For the k such that (G.12) holds, $\sum_{j=2}^{m} \widetilde{u}_{j,k}\, \widetilde{b}_j = \pm \left(\sum_{\substack{j=1 \\ j\neq k}}^{\ell} u_{j,1}\, b_j \right)$ is the shortest vector in $\mathcal{L}_k$. Consequently,

$$\left\| \sum_{\substack{j=1 \\ j\neq k}}^{\ell} u_{j,1}\, b_j \right\|_2 \leq \|b_k\|_2$$

Further, b_k and $\sum_{\substack{j=1 \\ j\neq k}}^{\ell} u_{j,1}\, b_j$ are both non-zero vectors contained by the lattice spanned by $\mathcal{B}$'s columns. Hence, using (4.19),

$$\min_{j\in 1,\ldots,m} \|b_j\|_2 \leq \left\| \sum_{\substack{j=1 \\ j\neq k}}^{\ell} u_{j,1}\, b_j \right\|_2 \leq \|b_k\|_2 < \min_{j\in 1,\ldots,m} \|b_j\|_2 \;\; \eta(\theta). \tag{G.14}$$

We have thus shown that for some k the two columns of $\mathcal{B}_k$ satisfy all the requirements of Proposition 6, namely, strongly θ-orthogonality and (4.19). Since Proposition 6 holds true for 2-D lattices, we can infer that any strongly $\frac{\pi}{3}$-orthogonal set of basis vectors can be obtained only by permuting and changing the signs of $\mathcal{B}_k$'s columns. However, the strongly $\frac{\pi}{3}$-orthogonal set of $\widetilde{\mathcal{B}}_k$'s columns is related to $\mathcal{B}_k$ via the $\mathcal{U}_k$ in (G.11). Thus, we have contradiction, which proves Proposition 6. □

Bibliography

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

[2] M. Y. Ting and E. A. Riskin, "Error-diffused image compression using a binary-to-gray scale decoder and predictive pruned tree-structured vector quantization," *IEEE Trans. Image Processing*, vol. 3, pp. 854–857, Nov. 1994.

[3] W. Pennebaker and J. Mitchell, *JPEG, Still Image Data Compression Standard*. Van Nostrand Reinhold, 1993.

[4] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[5] J. Kalifa and S. Mallat, "Thresholding estimators for linear inverse problems," *Ann. Statist.*, vol. 31, Feb. 2003.

[6] A. D. Hillery and R. T. Chin, "Iterative Wiener filters for image restoration," *IEEE Trans. Signal Processing*, vol. 39, pp. 1892–1899, Aug. 1991.

[7] A. K. Katsaggelos (Ed.), *Digital Image Restoration*. New York: Springer-Verlag, 1991.

[8] D. L. Donoho, "Nonlinear solution of linear inverse problems by Wavelet-Vaguelette Decomposition," *Appl. Comput. Harmon. Anal.*, vol. 2, pp. 101–126, 1995.

[9] W. James and C. Stein, "Estimation with quadratic loss," in *Proc. Fourth Berkeley Symp. Math. Statist. Probab.*, vol. 1, pp. 361–380, Univ. California Press, 1961.

[10] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

[11] D. L. Donoho, "Unconditional bases are optimal bases for data compression and for statistical estimation," *Appl. Comput. Harmon. Anal.*, vol. 1, pp. 100–115, Dec. 1993.

[12] R. Neelamani, H. Choi, and R. G. Baraniuk, "Wavelet-based deconvolution for ill-conditioned systems," in *Proc. IEEE ICASSP '99*, vol. 6, (Phoenix, AZ), pp. 3241–3244, Mar. 1999.

[13] R. Neelamani, H. Choi, and R. G. Baraniuk, "Wavelet-based deconvolution using optimally regularized inversion for ill-conditioned systems," in *Wavelet Applications in Signal and Image Processing VII, Proc. SPIE*, vol. 3813, pp. 58–72, July 1999.

[14] R. D. Nowak and M. J. Thul, "Wavelet-Vaguelette restoration in photon-limited imaging," in *Proc. IEEE ICASSP '98*, (Seattle, WA), pp. 2869–2872, 1998.

[15] M. R. Banham and A. K. Katsaggelos, "Spatially adaptive wavelet-based multiscale image restoration," *IEEE Trans. Image Processing*, vol. 5, pp. 619–634, Apr. 1996.

[16] Y. Wan and R. D. Nowak, "A Bayesian multiscale approach to joint image restoration and edge detection," in *Wavelet Applications in Signal and Image Processing VII, Proc. SPIE*, vol. 3813, pp. 73–84, July 1999.

[17] M. Figueiredo and R. D. Nowak, "Image restoration using the EM algorithm and wavelet-based complexity regularization," *IEEE Trans. Image Processing*, July 2003. To appear.

[18] A. Jalobeanu, L. Blanc-Feŕaud, and J. Zerubia, “Adaptive parameter estimation for satellite image deconvolution,” Tech. Rep. 3956, INRIA, 2000.

[19] P. de Rivaz and N. Kingsbury, “Bayesian image deconvolution and denoising using complex wavelets,” in *Proc. IEEE ICIP ’01*, vol. 2, (Thessaloniki, Greece), pp. 273–276, Oct. 7–10 2001.

[20] M. Unser, “Sampling—50 Years after Shannon,” *Proc. IEEE*, vol. 88, pp. 569–587, Apr. 2000.

[21] M. Unser and A. Aldroubi, “A general sampling theory for nonideal acquisition devices,” *IEEE Trans. Signal Processing*, vol. 42, pp. 2915–2925, Nov. 1994.

[22] D. L. Donoho, “De-noising by soft-thresholding,” *IEEE Trans. Inform. Theory*, vol. 41, pp. 613–627, May 1995.

[23] D. L. Donoho and I. M. Johnstone, “Asymptotic minimaxity of wavelet estimators with sampled data,” *Statist. Sinica*, vol. 9, no. 1, pp. 1–32, 1999.

[24] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*. Washington D.C.: V. H. Winston & Sons, 1977.

[25] K. R. Castleman, *Digital Image Processing*. New Jersey: Prentice Hall, 1996.

[26] G. Davis and A. Nosratinia, “Wavelet-based image coding: An overview,” in *Appl. Comput. Control Signals Circuits* (B. N. Datta, ed.), vol. 1, Birkhauser, 1999.

[27] D. L. Donoho and I. M. Johnstone, “Ideal spatial adaptation via wavelet shrinkage,” *Biometrika*, vol. 81, pp. 425–455, 1994.

[28] I. M. Johnstone, “Wavelet shrinkage for correlated data and inverse problems: Adaptivity results,” *Statist. Sinica*, no. 9, pp. 51–83, 1999.

[29] J. K. Romberg, H. Choi, R. G. Baraniuk, and N. G. Kingsbury, “A hidden Markov tree model for the complex wavelet transform,” *IEEE Trans. Signal Processing*, 2002. Submitted.

[30] N. G. Kingsbury, “Complex wavelets for shift invariant analysis and filtering of signals,” *Appl. Comput. Harmon. Anal.*, vol. 10, pp. 234–253, May 2001.

[31] P. Wong, “Inverse halftoning and kernel estimation for error diffusion,” *IEEE Trans. Image Processing*, vol. 6, pp. 486–498, Apr. 1995.

[32] R. W. Floyd and L. Stienberg, “An adaptive algoritm for spatial grayscale,” *Proc. Soc. Image Display*, vol. 17, no. 2, pp. 75–77, 1976.

[33] J. Jarvis, C. Judice, and W. Ninke, “A survey of techniques for the display of continuous tone pictures on bilevel displays,” *Comput. Graph and Image Process.*, vol. 5, pp. 13–40, 1976.

[34] T. D. Kite, B. L. Evans, A. C. Bovik, and T. L. Sculley, “Digital halftoning as 2-D delta-sigma modulation,” *Proc. IEEE ICIP ’97*, vol. 1, pp. 799–802, Oct. 26–29 1997.

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

[36] 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.

[37] R. A. DeVore, B. Jawerth, and B. J. Lucier, “Image compression through wavelet transform coding,” *IEEE Trans. Inform. Theory*, vol. 38, pp. 719–746, Mar. 1992.

[38] 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. 1179–1184, Apr. 1998.

[39] Z. Xiong, M. T. Orchard, and K. Ramchandran, “Inverse halftoning using wavelets,” *IEEE Trans. Signal Processing*, vol. 8, pp. 1479–1482, Oct. 1999.

[40] R. Neelamani, R. D. Nowak, and R. G. Baraniuk, “Model-based inverse halftoning with Wavelet-Vaguelette Deconvolution,” in *Proc. IEEE ICIP ’00*, (Vancouver, Canada), pp. 973–976, Sept. 2000.

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

[42] R. Averkamp and C. Houdre, “Wavelet thresholding for non (necessarily) Gaussian noise: Functionality,” *Ann. Statist.*, vol. 31, Feb. 2003.

[43] H.-Y. Gao, “Choice of thresholds for wavelet shrinkage estimate of the spectrum,” *Journal of Time Series Analysis*, vol. 18, pp. 231–251, 1997.

[44] D. L. Donoho and I. Johnstone, “Minimax estimation by wavelet shrinkage,” *Ann. Statist.*, vol. 26, pp. 879–921, 1998.

[45] T. Mitsa and K. Varkur, “Evaluation of contrast sensitivity functions for the formulation of quality measures incorporated in halftoning algorithms,” in *Proc. IEEE ICASSP ’93*, vol. 5,

pp. 301–304, 1993.

[46] T. D. Kite, N. Damera-Venkata, B. L. Evans, and A. C. Bovik, “Image quality assessment based on a degradation model,” *IEEE Trans. Image Processing*, vol. 9, pp. 636–650, Apr. 2000.

[47] Z. Wang and A. C. Bovik, “A universal image quality index,” *IEEE Signal Processing Lett.*, vol. 9, pp. 81–84, Mar. 2002.

[48] V. Monga, N. Damera-Venkata, and B. L. Evans., “Halftoning toolbox for MATLAB,” 2002. `www.ece.utexas.edu/~bevans/projects/halftoning/toolbox`.

[49] R. Rosenholtz and A. Zakhor, “Iterative procedures for reduction of blocking effects in transform image coding,” *IEEE Trans. Cir. Sys. for Video Tech.*, vol. 2, pp. 91–95, Mar. 1992.

[50] S. Minami and A. Zakhor, “An optimization approach for removing blocking effectsin transform coding,” *IEEE Trans. Cir. Sys. for Video Tech.*, vol. 5, pp. 74–82, Apr. 1995.

[51] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, “Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images,” *IEEE Trans. Cir. Sys. for Video Tech.*, vol. 3, pp. 421–432, Dec. 1993.

[52] K. T. Tan and M. Ghanbari, “Blockiness detection for MPEG-2-coded video,” *IEEE Signal Processing Lett.*, vol. 7, pp. 213–215, Aug. 2000.

[53] Z. Fan and R. Eschbach, “JPEG decompression with reduced artifacts,” in *Proc. IS&T/SPIE Symp. Electronic Imaging: Image and Video Compression*, (San Jose, CA), Feb. 1994.

[54] J. Chou, M. Crouse, and K. Ramachandran, "A simple algorithm for removing blocking artifacts in block-transform coded images," *IEEE Signal Processing Lett.*, vol. 5, pp. 33–35, Feb. 1998.

[55] J. Luo, C. W. Chen, K. J. Parker, and T. S. Huang, "Artifact reduction in low bit rate DCT-based image compression," *IEEE Trans. Image Processing*, vol. 5, pp. 1363–1368, 1996.

[56] Z. Fan and R. de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Trans. Image Processing*, vol. 12, pp. 230–235, Feb. 2003.

[57] G. Sharma and H. Trussell, "Digital color imaging," *IEEE Trans. Image Processing*, vol. 6, pp. 901–932, July 1997.

[58] C. Poynton, *A Technical Introduction to Digital Video*. New York: Wiley, 1996.

[59] *Independent JPEG Group Library*. www.ijg.org.

[60] R. J. Clarke, *Transform Coding of Images*. London, England: Academic Press, 1985.

[61] "The USC-SIPI image database." `sipi.usc.edu/services/database/Database.html`.

[62] A. Joux and J. Stern, "Lattice reduction: A toolbox for the cryptanalyst," *Journal of Cryptology*, vol. 11, no. 3, pp. 161–185, 1998.

[63] P. Nguyen and J. Stern, "Lattice reduction in cryptology: An update," in *Lecture notes in Comp. Sci.*, vol. 1838, pp. 85–112, Springer Verlag, 2000.

[64] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inform. Theory*, vol. 48, pp. 2201–2214, Aug. 2002.

[65] F. Abramovich and B. W. Silverman, “Wavelet decomposition approaches to statistical inverse problems,” *Biometrika*, vol. 85, pp. 115–129, Oct. 1998.

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

[67] K. Berkner, M. J. Gormish, and E. L. Schwartz, “Multiscale sharpening and smoothing in Besov spaces with applications to image enhancement,” *Appl. Comput. Harmon. Anal.*, vol. 11, pp. 2–31, July 2001.

[68] S. Ghael, A. M. Sayeed, and R. G. Baraniuk, “Improved wavelet denoising via empirical Wiener filtering,” in *Wavelet Applications in Signal and Image Processing V, Proc. SPIE*, vol. 3169, pp. 389–399, Oct. 1997.

[69] R. Edwards and G. Gaudry, *Littlewood-Paley and Multiplier Theory*. Berlin: Springer-Verlag, 1977.

[70] J. Peetre, *New Thoughts on Besov Spaces*. Durham, N.C.: Duke University Mathematics Series, 1976.