UNIVERSITY OF CALIFORNIA
SANTA CRUZ

GRAPH-BASED IMAGE RESTORATION

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

Amin Kheradmand

September 2016

The Dissertation of Amin Kheradmand
is approved:

Professor Peyman Milanfar, Chair

Professor Alex Pang

Professor Claire Gu

Dean Tyrus Miller
Vice Provost and Dean of Graduate Studies

Contents

List of Figures **vi**

List of Tables **x**

Abstract **xi**

Dedication **xiii**

Acknowledgments **xiv**

1 Introduction **1**
1.1 Imaging Pipeline and Different Sources of Distortion 1
1.2 Image formation model 3
1.3 Contributions 4
1.4 Existing Methods for Image Deblurring, Denoising, And Sharpening . . 7

2 A Graph-based Regularized Framework for Image Deblurring **11**
2.1 Introduction 12
2.2 Related Graph-based Regularization Approaches 16
2.3 Derivation of Building Block Matrices of The Proposed Algorithm . . . 21
2.3.1 Kernel Similarity Matrix K And Filtering Matrix W 21
2.3.2 Normalized Graph Laplacian Matrix 23
2.4 Proposed Deblurring Method 25
2.4.1 Spectral Analysis of The Overall Deblurring Algorithm 29
2.5 Image Sharpening As a Special Case of The Proposed Objective Function 30
2.6 Implementation Details 31
2.7 Numerical Experiments 34
2.7.1 Symmetric Blurs 36
2.7.2 Synthetic Motion Blur 44
2.7.3 Real Motion Deblurring 46
2.7.4 Comparison With Traditional Normalized Graph Laplacian . . . 49

3 Graph-based Denoising And Smoothing: Interpretation Based on Diffusion and Boosting Iterations **54**
3.1 Introduction . . . 55
3.2 Kernel Formulation for Denoising . . . 56
3.2.1 Case 1: Un-normalized Laplacian . . . 56
3.2.2 Case 2: Normalized Laplacian . . . 57
3.2.3 Spectral Analysis Based on Eigenvectors of Laplacian . . . 59
3.3 Diffusion and Boosting As Special Cases of the Proposed Algorithm . . 60
3.4 Graph-based multi-layer image decomposition and its interpretation based on diffusion and boosting iterations . . . 63
3.5 Experimental Results . . . 65
3.5.1 Comparisons With NLM Denoising Algorithm . . . 65
3.5.2 Comparison With Global Denoising Algorithm . . . 67

4 Data-adaptive Image Sharpening Based on The Difference of Smoothing Operators **69**
4.1 Introduction . . . 70
4.2 Problem Formulation and Proposed Sharpening Filter . . . 71
4.2.1 Underlying model . . . 71
4.2.2 Classical DoG Operator . . . 72
4.2.3 Structure-aware Sharpening Filter . . . 73
4.3 Interpretation of The Proposed Filter As Data-adaptive Difference of Smoothing Operator . . . 76
4.3.1 Special Cases of The Proposed Sharpening Framework . . . 78
4.4 Reducing Color Artifacts . . . 79
4.5 Experimental Results . . . 81
4.5.1 Investigating The Effect of Different Parameters via Synthetic Examples . . . 83
4.5.2 Real Sharpening Examples . . . 88

5 A Discussion on The Range of The Eigenvalues of Different Normalized Graph Laplacians **91**
5.1 Definitions and Theorems Used in Analyses . . . 91
5.1.1 Spectral Radius of Stochastic Matrices . . . 93
5.1.2 Irreducible matrices and connected graphs . . . 94
5.1.3 Perron-Frobenius theorem . . . 95
5.1.4 Gershgorin's theorem . . . 95
5.1.5 A Theorem on Characterizing The Graphs Preserving Positive Definiteness Upon Thresholding . . . 96
5.2 Investigation of The Spectral Range of Normalized Graph Laplacian Matrices . . . 97
5.2.1 Random Walk Laplacian $I - D^{-1}K$ . . . 98
5.2.2 Traditional Normalized Laplacian $I - D^{-1/2}KD^{-1/2}$ . . . 98

5.2.3 Sinkhorn Normalized Laplacian 99
5.2.4 Spectrum of Different Graph Structures: Full Versus Sparse Similarity Matrices 99

6 Conclusions and Future Work **105**
6.1 Conclusions 105
6.2 Future Work and Extensions 107
6.2.1 Optimal Eigenbasis for Representing Any Given Image 107
6.2.2 A Unified Framework for Estimating The Filtering Matrix and The Latent Image 108
6.2.3 Patch Ordering Idea As Sorting Nodes In A Graph 109

Bibliography **113**

List of Figures

1.1 Block diagram of camera imaging pipeline. 4

2.1 Block diagram of our proposed iterative deblurring method. K is the kernel similarity matrix, and W is the doubly stochastic filtering matrix. $\hat{\mathbf{z}}^{(k)}$ is the estimate corresponding to optimization of the objective function at the kth outer iteration of the algorithm. 16

2.2 Graph representation of images and construction of kernel similarity matrix K, un-normalized Laplacian $D - K$ and normalized Laplacian $I - C^{-1/2}KC^{-1/2}$. 17

2.3 (a) From left to right: original 41×41 image, and the eigenvectors of $F(A, W)$ corresponding to the four largest eigenvalues for $\beta = 0.7$ and $\eta = 0.2$. 30

2.4 Set of color images used for evaluation of our method: (a)Building image (480×640), (b)Bikes image (494×494) , (c)Girl image (496×700), (d)Street image (480×640), (e)Boat image (420×520), and (f)Book shelf image (580×520). 32

2.5 Deblurring example with Gaussian blur: (a)clean Bikes image, (b)blurred noisy image, (c)output of [22], and (d)output of our algorithm. 36

2.6 Deblurring example with out-of-focus blur: (a)clean Girl image, (b)blurred noisy image, (c)output of [22], and (d)output of our algorithm. 37

2.7 Real sharpening examples: left column: input blurred noisy image, middle column: output of Focus Magic software at http://www.focusmagic.com, and right column: output of our algorithm ($\eta = 0.1$ and $\beta = 0.06$ for both images). 38

2.8 The relationship between the patch size and blur kernel width for Girl image and out-of-focus blur with different radii. 41

2.9 The effect of the smoothing parameter h in the kernel similarity function for Girl image and out-of-focus blur with radius 7 for two different noise variances. 42

2.10 Convergence plots of the CG iterations for different initializations. . . . 43

2.11 Real motion deblurring example: (a)input blurred noisy image, (b)output of hyper-Laplacian algorithm [58], (c)output of [97], and (d)output of our algorithm ($\eta = 0.031, \beta = 0.6$). 46
2.12 Real motion deblurring example: (a)input blurred noisy image, (b)output of hyper-Laplacian algorithm [58], (c)output of [97], and (d)output of our algorithm ($\eta = 0.25, \beta = 2.5$). 47
2.13 Real motion deblurring example: (a)input blurred noisy image, (b)output of hyper-Laplacian algorithm [58], (c)output of [97], and (d)output of our algorithm ($\eta = 0.018, \beta = 0.9$). 48
2.14 Real motion deblurring example: (a)input blurred noisy image, (b)output of hyper-Laplacian algorithm [58], (c)output of [97], and (d)output of our algorithm ($\eta = 0.031, \beta = 0.6$). 49
2.15 Real motion deblurring example: (a)input blurred noisy image, (b)output of hyper-Laplacian algorithm [58], (c)output of [97], and (d)output of our algorithm ($\eta = 0.032, \beta = 0.6$). 50
2.16 Deblurring examples with blurred noisy Girl image by out-of-focus blur with radius 7 and additive white Gaussian noise with standard deviation $\sigma = 1$: (a)clean image, (b)blurred noisy image, (c)output of the deblurring algorithm with the corresponding traditional normalized Laplacian (PSNR = 29.40dB, SSIM = 0.8734), and (d)output of our proposed deblurring algorithm (PSNR = 30.58dB, SSIM = 0.9058). 51
2.17 Motion deblurring examples with blurred noisy Street image by synthetic motion blur kernel and additive white Gaussian noise with standard deviation $\sigma = 1$: (a)clean image, (b)blurred noisy image, (c)output of the deblurring algorithm with the corresponding traditional normalized Laplacian (PSNR = 27.93dB), and (d)output of our proposed deblurring algorithm (PSNR = 29.75dB). 52

3.1 Denoising experiment on 31×31 piece-wise constant synthetic patch (a)clean patch, (b)noisy patch by adding white Gaussian noise with $\sigma = 25$ (PSNR = 20.31dB) , (c)output of iterative algorithm (3.8) (PSNR= 35.96dB, $\eta = 19$), and (d)spectrum of filter matrices W (λ_i's) and W' (λ_i''s). 60
3.2 Block diagram of the proposed graph-based approach for multi-layer image decomposition. 61
3.3 Set of images used for evaluating the performance of our denoising algorithm with respect to standard NLM. 62
3.4 Denoising experiment on 256×256 House image, (a)clean image, (b)noisy image ($\sigma = 20$), (c)standard NLM output image (PSNR = 30.80dB), (d)output of iterative algorithm (3.3) (PSNR= 32.16dB, $\eta = 0.67$), and (e)output of iterative algorithm (3.8) (PSNR= 32.37dB, $\eta = 0.82$). . . . 62

3.5 Denoising experiment on 256 × 256 color parrot image, (a)original image, (b)noisy image ($\sigma = 20$, PSNR= 17.06dB), (c)standard NLM output image (PSNR= 27.68dB), and (d)output of iterative algorithm (3.8) (PSNR= 28.54dB). 65
3.6 Set of images used for evaluating the performance of our denoising algorithm with respect to global denoising algorithm. 66
3.7 Denoising experiment on 256×256 Aerial image, (a)original image, (b)noisy image ($\sigma = 20$, PSNR= 22.08dB), (c)output of global denoising algorithm in [107] (PSNR= 26.67dB), and (d)output of iterative algorithm (3.8) (PSNR= 26.89dB). 67

4.1 Block diagram of the proposed enhancement algorithm. $\mathbf{y}$ is the input image and $\hat{\mathbf{z}}$ is the enhanced output image. 76
4.2 (a)top left: original 91×91 image, (a)top right: output with $f(\lambda) = 3\lambda^2 - 2\lambda^3$, (a)bottom left: output with $f(\lambda) = 6\lambda^2 - 5\lambda^3$, (a)bottom left: output with $f(\lambda) = 11\lambda^2 - 10\lambda^3$, and (b)the eigenvalues of the smoothing matrix W constructed from (a)top left along with the eigenvalues corresponding to sharpened smoothing filter $3W^2 - 2W^3$ and band pass filters $6W^2 - 5W^3$ and $11W^2 - 10W^3$. 79
4.3 First row: (a)input JPEG image and its different channels: (b)luminance Y channel, (c)chrominance Cb channel, and (d)chrominance Cr channel. Second row: (e)output of our sharpening filter and its different enhanced channels: (f)enhanced luminance Y channel, (g)enhanced chrominance Cb channel, and (h)enhanced chrominance Cr channel. 80
4.4 Set of color images used in synthetic experiments. 82
4.5 Effect of the parameter h_1 via synthetic examples with $\beta = 1.5, k = 3$: (a)input noisy and blurry image, (b)output of our sharpening algorithm with $h_1 = 0.8$, (c)output image with $h_1 = 1.2$, and (d)output image with $h_1 = 2$. 82
4.6 Effect of the parameter β via synthetic examples with $h_1 = 1.4, k = 3$: (a)input noisy and blurry image, (b)output of our sharpening algorithm with $\beta = 1.5$, (c)output image with $\beta = 2.5$, and (d)output image with $\beta = 3.5$. 83
4.7 Effect of the parameter k via synthetic examples with $h_1 = 1.4, \beta = 1.5$: (a)input noisy and blurry image, (b)output of our sharpening algorithm with $k = 2$, (c)output image with $k = 3.5$, and (d)output image with $k = 6$. 84
4.8 (a)input image, (b)output of [5], (c)output of [43], (d)output of [128], and (e)output of our sharpening algorithm with $h_1 = 0.7, k = 4, \beta = 1.6$. 85
4.9 (a)input image, (b)output of [5], (c)output of [43], (d)output of [128], and (e)output of our sharpening algorithm with $h_1 = 2.7, k = 4, \beta = 1.7$. 86
4.10 (a)input image, (b)output of [5], (c)output of [43], (d)output of [128], and (e)output of our sharpening algorithm with $h_1 = 1.4, k = 4.5, \beta = 1.2$. 87

5.1 41×41 image used in the experiments and patches around pixels i and j of the image. 92
5.2 Spectrum of different full normalized Laplacians: (a)spectrum of $I - D^{-1}K$, (b)spectrum of $I - D^{-1/2}KD^{-1/2}$, and (c)spectrum of $I - C^{-1/2}KC^{-1/2}$. 100
5.3 Spectrum of different sparse normalized Laplacians: (a)spectrum of $I - D^{-1}K$, (b)spectrum of $I - D^{-1/2}KD^{-1/2}$, and (c)spectrum of $I - C^{-1/2}KC^{-1/2}$. 101
5.4 Spectrum of different sparse normalized Laplacians in terms of scaling parameter h in (5.1): (a)spectrum of $I - D^{-1}K$ for different values of h, and (b)spectrum of $I - C^{-1/2}KC^{-1/2}$ for different values of h. 101

6.1 Original image used for computing the similarity matrix K. 110
6.2 (a)Pixel intensity values (pixels in raster-scan order), and (b)pixel intensity values when pixels are ordered using the permutation from sorting the second eigenvector of the graph Laplacian matrix. 110
6.3 (a)Kernel similarity matrix from pixels indexed by raster-scan ordered version of the input image $\mathbf{z}$, and (b)kernel similarity matrix indexed by the permutation matrix P. 112

List of Tables

2.1 Properties of different normalized graph Laplacians. Last row is our definition. 19
2.2 Condition number of $(A^T(I+\beta(I-W))A+\eta(I-W))$ for different values of η and β and blurring matrix A corresponding to out-of-focus blur with radius 7. The condition number of A^TA is 5.74×10^{20}. 30
2.3 SSIM and PSNR performance of the kernel similarity-based algorithm and IDDBM3D [22] for Gaussian blur kernel of size 25×25 with standard deviation 1.6 and out-of-focus blur generated using disk function of radius 7. In each cell, the first number denotes SSIM value, and the second number represents PSNR value in dB. 33
2.4 Set of parameters in different synthetic color image deblurring examples in this chapter. nInner is the maximum number of the inner CG iterations. 35
2.5 PSNR oracle performance of the proposed algorithm vs. the low-noise case. 43
2.6 ISNR values for the Girl image and out-of-focus blur with different noise standard deviation when the weights are computed from the input noisy blurred image. 44
2.7 SSIM and PSNR performance of the kernel similarity-based algorithm in comparison with those of motion deblurring methods in [58] and [97] for synthetic camera motion blur. In each cell, the first number denotes SSIM value, and the second number represents PSNR value. 45

3.1 Denoising PSNR and SSIM performance of the iterative algorithms (3.3) and (3.8). 63
3.2 Denoising performance and run-time in seconds of the iterative algorithm (3.8) and global denoising [107]. 66

Abstract

Graph-based Image Restoration

by

Amin Kheradmand

Digital photography has experienced great progress during the past decade. A lot of people are recording their moments via digital hand-held cameras. Pictures taken with digital cameras usually undergo some sort of degradation in the form of noise/blur depending on the camera hardware and environmental conditions in which the photos are taken. This leads to an ever-increasing demand for effective and efficient image enhancement algorithms to achieve high quality output images in digital photography systems. In this dissertation, a new graph-based framework is introduced for different image restoration applications. This framework is based on exploiting the existing self-similarity in images. We introduce a new definition of normalized graph Laplacian matrix for image processing. We use this new definition to develop effective enhancement algorithms for image deblurring, image denoising, and image sharpening.

First, we develop a regularization framework for image deblurring by constructing a new graph-based cost function. Minimizing the corresponding cost function yields effective outputs for different blur types including out-of-focus and motion blurs. Our proposed deblurring algorithm based on the new definition of normalized graph Laplacian provides performance and analysis advantages over previous methods. We have shown its effectiveness for several synthetic and real deblurring examples.

Second, we develop a new graph-based framework for image denoising. The proposed denoising method exploits the similarity information in images by constructing the similarity matrix which in turn is used to derive the corresponding graph Laplacian. A graph-based objective function with new data fidelity and smoothness terms is constructed and minimized. We also establish the relationship between our proposed regularized framework and two well-known iterative methods for improving the performance of kernel-based denoising methods; namely, diffusion and boosting iterations. We compare the performance of the proposed denoising method with that of NLM algorithm [11] and demonstrate that our proposed algorithm is able to enhance over NLM. Furthermore, we present a graph-based analysis framework for multi-layer image decomposition using diffusion and boosting iterations.

Third, we propose a new data-adaptive sharpening algorithm based on the notion of difference of smoothing operators. We provide an interpretation of our proposed sharpening method as the image-derived version of difference of Gaussians (DoG) operator broadly used in numerous image processing and computer graphics tasks [66, 121, 122].

Finally, we provide a theoretical study on the reported range of the eigenvalues of various definitions of normalized graph Laplacian for different graph structures. This sheds light on the existing ambiguity on the spectral range of such matrices in different applications.

To my family,

Acknowledgments

Above all, I would like to express my deepest gratitude to my advisor, Professor Peyman Milanfar, for his guidance, patience, encouragement, and providing me and all his students with excellent atmosphere for doing research. Without his guidance and persistent help this dissertation would not have been possible. I also want to thank my other dissertation committee members, Professor Alex Pang and Professor Claire Gu, for their invaluable advice.

I am also thankful for the former and current members of MDSP lab, Xiang Zhu, Hossein Talebi, Sujoy Biswas, Rob Sumner, and M. Hossein Daraei. They are excellent researchers and great friends.

I would like to thank my family for all their love, encouragement, and endurance. I am grateful for my parents and my lovely grandmother, who supported me in all aspects of my life and provided me with a great atmosphere to grow. I am thankful for my sister, Elham, who always encouraged me towards my goals through her positive and supportive words.

Chapter 1

Introduction

Abstract - In this chapter, we discuss various sources of distortion in imaging systems. We describe the mathematical model for image formation process. We also review some of the existing methods for removing distortions in images. Finally, we briefly discuss the contributions of this dissertation for different image restoration applications.

1.1 Imaging Pipeline and Different Sources of Distortion

With ever-increasing demand for high quality images in applications ranging from medical imaging, astronomical imagery, microscopy, aerial and satellite imagery, and photography using cell-phone cameras, there is a tremendous need for efficient and effective image enhancement methods. In order to develop such methods, having a good understanding of different sources of image degradation is essential. Capturing scene radiance and converting it to digital images in current digital cameras takes place

in different stages as shown in Fig. 1.1. During the image capturing process, several distortions are introduced to the final digital pictures. Noise and blur are two main factors which hinder producing pleasing output images. Noise usually comes from the imaging circuitry and lack of enough light captured by the imaging device. However, blur has more complicated sources, e.g., relative motion between camera and the objects in the scene. The incident light from the scene is focused by the camera lens on camera sensor (CCD or CMOS). In this step, defocus blur is produced due to imperfect lens settings or limited depth of field. The camera sensor contains color filter arrays which are sensitive to different wavelengths in the incoming light. The arriving photons are accumulated in the sensors and are converted to electrical voltage which in turn is stored in digital formats using Analog to Digital converters (A/D). In this step, quantization noise is introduced which is an important source of distortion, especially when the image bit-depth is limited. Dark current noise and thermal noise are introduced due to the camera pipeline circuitry which increase in proportion to the exposure time of the camera. Also, fixed pattern noise is noticeable for longer exposure shots where particular pixels are vulnerable to producing brighter intensities above the background noise. There is another type of noise called shot noise which originates from the particle nature of photons. The number of photons hitting the camera sensor is signal dependent and can be described by a Poisson distribution [30]. When the exposure time is large enough such that the number of photons reaching camera sensor is sufficient, the Poisson distribution will be similar to the Gaussian distribution and the noise can be modeled as zero mean independent and identically distributed Gaussian noise [30]. A demosaicking

process is used to interpolate the color data collected by color filter arrays. Gamma correction, color correction, and white balancing are among other steps that are used in the camera pipeline for generating the final output image.

As regards various types of blur, diffraction-limited blur is related to resolution limits due to diffraction [8]. Atmospheric turbulence blur is another type of blur which is prevalent in long-distance imaging systems [130]. Motion blur is caused because of the motion of the camera itself or the motion of the objects in the scene. This situation happens especially in cases where the camera can not be held fixed. For instance, for pictures taken with existing cell-phones, camera shaking is inevitable. Motion blur is more severe in situations where the camera exposure time is long or there is motion in the objects within a scene. In poorly lit scenes, we need longer exposure times to reduce the noise which in turn results in more complex motion blurs in the final image. Therefore, there is always this tradeoff between reducing the amount of noise versus having a sharper image. In this dissertation, we propose solutions for cases where the input image is noisy and/or blurry.

1.2 Image formation model

The process of image formation can be mathematically described using the following model:

$$\mathbf{y} = A\mathbf{z} + \mathbf{n}, \tag{1.1}$$

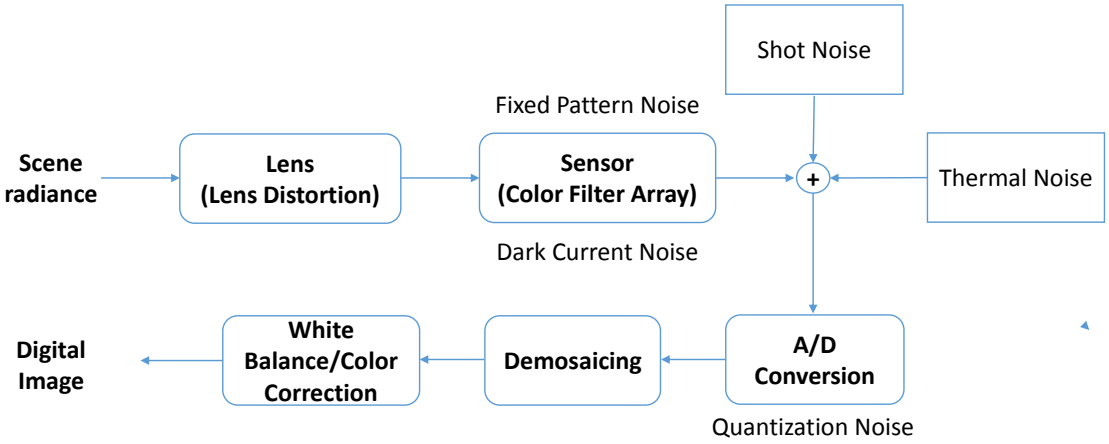


Fig. 1.1: Block diagram of camera imaging pipeline.

in which $\mathbf{z}$ is an $n \times 1$ lexicographically ordered vector representation of the original (distortion free) $\sqrt{n} \times \sqrt{n}$ image Z. $\mathbf{n}$ is a noise vector consisting of independent and identically distributed zero mean noise with standard deviation σ, and A is an $n \times n$ blurring matrix. Also, $\mathbf{y}$ is an $n \times 1$ vector representation of the input degraded image. In this dissertation, we use the model in equation (1.1) to develop our restoration algorithms for image deblurring and sharpening. For image denoising, $A = I$ with I identity matrix, for which the model in (1.1) reduces to

$$\mathbf{y} = \mathbf{z} + \mathbf{n}. \tag{1.2}$$

1.3 Contributions

In this dissertation, we propose a general data-adaptive, graph-based framework for image restoration. We develop new methods for different restoration tasks including image deblurring, image denoising, and image sharpening. In what follows,

we briefly discuss our contributions in different chapters of the dissertation.

- **Chapter 2 - A Graph-based Regularized Framework for Image Deblurring**

We develop a regularization framework using the existing interrelationship among different parts of images. Self-similarity in images provides us with an effective tool for regularizing the ill-posed deblurring problem. We introduce a new definition of the normalized graph Laplacian with advantages for image processing applications. Using this new definition of graph Laplacian, we construct a cost function with new data fidelity and regularization terms whose minimization gives the final estimate for the deblurring problem. The proposed deblurring algorithm is successfully applied to synthetic and real out-of-focus and motion deblurring examples.

- **Chapter 3 - Graph-based Denoising And Smoothing: Interpretation Based on Diffusion and Boosting Iterations**

We use our definition of the normalized graph Laplacian and the corresponding regularization term in Chapter 2 along with a new graph-based data fidelity term to develop an iterative algorithm for image denoising. We provide a framework to describe the underlying mechanism in kernel similarity-based methods and offer a path for their further improvement. We specifically study the relationship between our proposed denoising algorithm and two well-known iterative methods widely used for improving the performance of similarity-based denoising algorithms; namely, diffusion and boosting

iterations [69]. Moreover, we introduce an analysis framework for multi-layer image decomposition using our smoothing and Laplacian matrices.

- **Chapter 4 - Data-adaptive Image Sharpening Based on The Difference of Smoothing Operators**

There are many applications in image processing, computer vision, and computer graphics in which difference of Gaussians (DoG) operators are used for different purposes. In Chapter 4, we exploit the data-adaptive filtering and Laplacian matrices to develop a simple yet effective sharpening algorithm. We establish its connection with the DoG operator in [66, 121, 122]. We use synthetic and real examples to demonstrate efficacy of our algorithm for image sharpening.

- **Chapter 5 - A Discussion on The Range of The Eigenvalues of Different Normalized Graph Laplacians**

Eigenvalues of graph Laplacians are important factors in revealing the structural properties of the underlying graphs. We provide a theoretical study on the range of the eigenvalues of different normalized graph Laplacians for different graph structures. This sheds light on the existing ambiguity in the literature regarding the range of spectra of such matrices.

1.4 Existing Methods for Image Deblurring, Denoising, And Sharpening

In image deblurring, the goal is to invert the effect of blur in the input image while avoiding noise amplification and ringing artifacts in the final solution. For this purpose, any deblurring algorithm needs to incorporate appropriate mechanisms to regularize the ill-posed inverse problem. Several regularization approaches have been used in existing deblurring algorithms in the literature. Some non-blind deblurring papers use TV regularization terms [75, 115]. Nonlocal TV-based regularization terms are used in [127]. In [48], a hardware attachment is used to deblur images. Also, a progressive inter-scale intra-scale approach has been exploited for non-blind image deconvolution in [124]. Shan et al. have proposed a cost function for motion deblurring with different derivative terms in the data fidelity term [97]. Cho et al. propose a blur model for handling outliers in deblurring problem [16]. In [126, 131], multiple images are used to improve the performance of deblurring. Some other papers connect the idea of graph signal representation and associated Laplacian matrix in graph theory with nonlocal similarity in image restoration [29, 69, 99].

There is a very rich literature on different methods to come up with a desirable estimate of the latent image for image denoising. In general, regularization methods for denoising can be divided into two categories. The first group are those classical methods that take advantage of some prior knowledge about images. Algorithms that exploit either image smoothness priors [93] or sparsity of image spectral coefficients in

some specific basis (e.g., Wavelet or DCT) [26] fall into this group. The second class of methods exploits the existing self-similarity in images. Kernel-based denoising methods like bilateral, nonlocal means (NLM), and LARK [11,69,105,112], fall into this category. In a wide angle view, all the above mentioned algorithms perform denoising based on some type of subset selection or shrinkage operation in a fixed or adaptive basis. Note that state-of-the- art image denoising methods like BM3D try to find the optimal type of shrinkage operation by combining self-similarity information with sparsity property of image spectrum coefficients in some appropriate domain [21].

Similarly, there is a vast literature on different techniques for image sharpening and contrast enhancement. There are many methods based on improving the linear unsharp masking technique. [87] is based on the idea of adaptive tuning of the sharpening parameter based on local characteristics of the input image. [55] tries to learn the strength parameter of unsharp mask from an external set of training images. In [5], sigma filtering is combined with unsharp masking and a clipping process is added to control the noise and overshoot effects. Another variant of unsharp masking using an exploratory data model has been presented in [23]. Also, a nonlinear unsharp mask algorithm is used for mammogram enhancement in [77]. Methods based on the histogram of input images constitute another category of algorithms for contrast enhancement [2,86,123]. While the aforementioned methods are simple to implement, they still produce noise amplification and overshoot artifacts which deteriorates the quality of the final results. Anisotropic diffusion [80, 110, 118] and shock filters [76] are nonlinear scale-space PDE-based approaches widely used for edge enhancement in different image

processing applications. [1,34,37] are among variants of those PDE-based seminal works. PDE-based methods provide a powerful mechanism for edge enhancement. However, they can lack texture and fine detail preservation in the final output images. Bilateral filter [112] has been used widely as an edge preserving smoothing filter for different image processing and computer vision tasks [57]. In [125], the adaptive bilateral filter is proposed by introducing an offset to the range kernel definition of the original bilateral filter. This enables the filter to switch its behavior from smoothing to sharpening based on the pixel-wise adjustment of the offset parameter using a training procedure [125]. Guided filtering (GF) [43] is another edge preserving filter widely used for different image processing tasks [96]. In [83], a variant of GF based on the same shifting idea in [125] is used for sharpness enhancement and noise reduction. Also, a weighted version of GF is presented in [62] based on a similar pixel-wise regularization parameter tuning in [32,71] for detail manipulation. In [128], a restoration algorithm for noisy and weakly blurred images based on kernel regression [105] is introduced. In [19, 108], nonlocal multi-scale approaches have been exploited effectively for image editing and automatic sharpness enhancement, respectively. Diffusion maps for edge aware image editing are introduced in [31]. An image enhancement method based on piecewise linear directional smoothing and sharpening has been proposed in [94]. Local Laplacian filters in [78] reduce the halo artifacts prevalent in methods based on Laplacian pyramid. Also, a fast realization of this technique has been proposed in [3]. While these methods produce good results, there still exists room for performance improvement, especially when the input image is noisy and the goal is to simultaneously remove the noise and boost the sharpness. Also,

there is a need for a more general framework to provide a better understanding of the underlying lowpass and highpass filters involved in the sharpening operation.

Chapter 2

A Graph-based Regularized Framework for Image Deblurring

Abstract - Any image can be represented as samples of a function defined on a weighted graph, in which the underlying structure of the image is encoded in kernel similarity and associated Laplacian matrices. In this chapter, we develop an iterative graph-based framework for image deblurring based on a new definition of the *normalized* graph Laplacian. We propose a cost function which consists of a new data fidelity term and a regularization term derived from the specific definition of the normalized graph Laplacian. The normalizing coefficients used in the definition of the Laplacian and the associated regularization term are obtained using fast symmetry preserving matrix balancing. This results in some desired spectral properties for the normalized Laplacian such as being symmetric, positive semi-definite, and returning the zero vector when applied to a constant image. Our algorithm comprises of outer and inner iterations, where

in each outer iteration, the similarity weights are recomputed using the previous estimate and the updated objective function is minimized using inner Conjugate Gradient (CG) iterations. This procedure improves the performance of the algorithm for image deblurring, where we do not have access to a good initial estimate of the underlying image. Moreover, the specific form of the cost function allows us to render the spectral analysis for the solutions of the corresponding linear equations. Experimental results verify the effectiveness of the proposed algorithm on both synthetic and real examples.

2.1 Introduction

Most real pictures exhibit some amount of degradation depending on the camera and settings used to capture the scene, environmental conditions, and the amount of relative motion between camera and subject, among other factors. Restoration algorithms aim to undo undesired distortions like blur and/or noise from the degraded image. In this chapter, we concentrate on problems where the main distortion of the image comes from blurring. We assume linear shift invariant point spread functions (PSFs), such that the blurring process is described through the following linear model

$$\mathbf{y} = A\mathbf{z} + \mathbf{n}. \tag{2.1}$$

In this model, the blurring matrix A is constructed from the corresponding PSF and usually has a special structure depending on the type of boundary condition assumptions [24, 42].

Most existing deblurring methods rely on optimizing a cost function of the

form

$$E(\mathbf{z}) = \|\mathbf{y} - A\mathbf{z}\|^2 + \eta\ R(\mathbf{z}) \tag{2.2}$$

with respect to the unknown image vector $\mathbf{z}$. The first term in the above is the data fidelity term and the second term implies a prior term which regularizes the inherently ill-posed problem. In such algorithms, the parameter η controls the amount of regularization to keep the final estimate from being too smooth or exhibiting unpleasant noise amplification and ringing artifacts. Deblurring algorithms can be classified based on the type of blurs they deal with, and also different choices of the regularization term they exploit to solve the deblurring problem [27, 132]. A large class of deblurring algorithms take advantage of a total variation (TV)-type regularization term [14, 75, 115]. They mostly differ in the specific definition of the TV term and the optimization method for solving the resulting cost function. Other methods use a nonlocal differential operator as the regularization term with different norms [82, 106, 127]. Sparsity-based methods are also motivated by sparse representation of images in some appropriate domain [25, 73]. In [59], a Hessian norm regularization is used for deblurring, with biomedical applications. Example-based manifold priors are used in [74] to regularize the deblurring problem. In [17], a prior term is added to encourage the estimate to have a gradient distribution similar to a reference. Furthermore, some recent algorithms are based on the idea of decoupling deblurring and denoising and exploiting the powerful BM3D algorithm [21] in their denoising phase [22, 61]. In [22], BM3D frames are defined explicitly and based on a generalized Nash equilibrium approach, the two objective functions for denoising and deblurring parts are balanced. This algorithm is one of the best existing

deblurring methods for symmetric blurs (e.g., Gaussian and out-of-focus blurs). For motion deblurring applications [58, 60], a hyper-Laplacian prior based on the statistics of natural images is used. Shan et al. [97] have proposed a cost function in which the data fidelity term involves different derivative terms for motion deblurring of natural images.

In this chapter, we propose a new approach for kernel similarity-based image deblurring by introducing a novel data-adaptive objective function. We also show that a special case of the proposed approach can be used for image sharpening. Figure 2.1 depicts a block diagram of our iterative deblurring method. As shown in Fig. 2.1 and Algorithm 1, the proposed method consists of a number of steps (outer iterations), such that at each step k, an updated objective function is minimized using Conjugate Gradient (CG) inner iterations to obtain the corresponding estimate $\hat{\mathbf{z}}^{(k)}$. To clarify the differences and contributions of this work as compared to some other nonlocal regularization works [9, 10, 35, 36, 68, 82, 104, 109, 127], it is worthwhile listing them here.

1. We propose a new cost function (2.15) for image restoration based on a new definition of the *normalized* graph Laplacian. The proposed cost function (2.15) includes a normalized regularization term derived from this new definition of the graph Laplacian as well as a new data fidelity term. The normalizing coefficients are obtained from a fast symmetry preserving matrix balancing algorithm [56]. This results in some desired spectral properties for the graph Laplacian. Namely, the proposed Laplacian is symmetric, positive semi-definite, and when applied to a constant vector, it returns the zero vector. In this chapter, we will discuss the

spectral properties of the proposed graph Laplacian and compare its attributes and performance with the existing graph Laplacians.

2. Taking advantage of the quadratic form of the proposed cost function as well as spectral properties of the proposed Laplacian matrix, we present a filtering interpretation for different terms in our objective function as a tool for spectral analysis of the resulting restoration algorithms. Moreover, the symmetric and sparse nature of the resulting filtering and Laplacian matrices equips us with the required tools for efficient implementation of the algorithm using CG and fast sparse matrix-vector products.

3. The way we initially compute the kernel similarity values is different in the sense that we start with a once denoised version of the input noisy and blurry image, and hence avoid the contribution of the noise and ringing artifacts of other deblurring algorithm to the computation of the similarity weights as in e.g., [127]. This denoised initial image is also exploited as a plug-in estimator of $A\mathbf{z}$ (blurred clean image) used in the Predicted Mean Squared Error (PMSE) measure for stopping the inner CG iterations. We allow the weights to be updated during the outer iterations to improve the performance of the algorithm by computing the similarity weights from the enhanced versions of the input image through the outer iterations.

4. Our approach is quite general in the sense that it is able to handle a variety of different PSFs including symmetric PSFs and more challenging motion blur PSFs.

As compared to [70], our approach is different in the following respects:

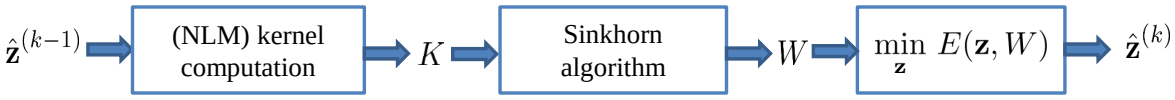


Fig. 2.1: Block diagram of our proposed iterative deblurring method. K is the kernel similarity matrix, and W is the doubly stochastic filtering matrix. $\hat{\mathbf{z}}^{(k)}$ is the estimate corresponding to optimization of the objective function at the kth outer iteration of the algorithm.

5. Although the general symmetrizing idea is similar, [70] focuses on symmetrizing smoothing filters, in which it starts from a non-symmetric smoothing filter and returns its symmetrized version using the original Sinkhorn matrix balancing algorithm in [100] with performance and analysis advantages described in [70]. In this chapter, we start from the symmetric and non-negative similarity matrix K and use a different fast matrix balancing algorithm, designed for scaling symmetric and non-negative matrices, with fast convergence and symmetry-preserving properties even when the matrix scaling algorithm is stopped early [56].

6. We use the symmetric and doubly stochastic output of [56] to define the normalized Laplacian and we use it in a variational graph-based formulation for the underlying restoration problems.

2.2 Related Graph-based Regularization Approaches

In this section, we summarize some of the existing methods based on the idea of nonlocal regularization in a graph-based framework. We first clarify our notation

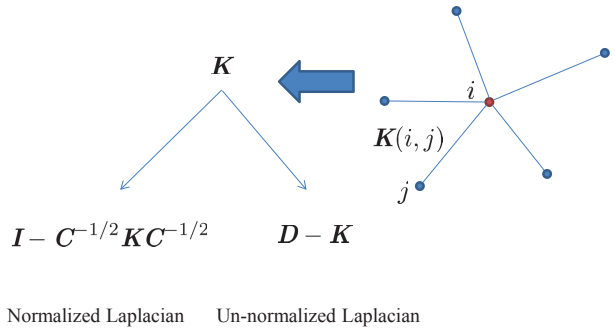


Fig. 2.2: Graph representation of images and construction of kernel similarity matrix K, un-normalized Laplacian $D - K$ and normalized Laplacian $I - C^{-1/2}KC^{-1/2}$.

and summarize some of the definitions commonly used in the nonlocal regularization approaches in the literature. As depicted in Fig. 2.2, any image can be defined as an intensity function on the vertices V of a weighted graph $G = (V, E, K)$ consisting of a finite set V of vertices (image pixels) and a finite set $E \subset V \times V$ of edges (i, j) with the corresponding weights $K(i, j)$ which measure similarity between vertices (pixels) i and j in the graph (e.g., Eq. 2.10). The function (intensity) values of the image can be denoted as a vector $\mathbf{z} = [z(1), ..., z(n)]^T$. The similarity weights are represented as an $n \times n$ matrix K, which is symmetric, positive definite, and non-negative valued.

As shown in Fig. 2.2, graph Laplacian matrix is derived from K and plays an important role in describing the underlying structure of the graph signal. There are three different definitions of the graph Laplacian commonly used in the literature in the context of graph signal and image processing, each having different spectral properties [20, 98, 99]. In this chapter, we present a fourth one, a new normalized graph Laplacian for image processing purposes. In Table 2.1, we have summarized the

properties of different types of normalized Laplacian used in the literature along with those of our proposed definition.

In [9, 29], the difference of a function $z : V \to \Re$ on an edge $(i, j) \in E$ of the graph G is defined as:

$$(dz)(i, j) = \sqrt{K(i, j)}(z(j) - z(i)). \tag{2.3}$$

Also, the gradient vector of a function z at a vertex $i \in V$ can be expressed as:

$$\nabla z(i) = [dz(i, j_1), ..., dz(i, j_m)]^T, \quad \forall (i, j) \in E. \tag{2.4}$$

Accordingly, the Laplace operator of z at a vertex i is derived as:

$$\Delta z(i) = \sum_{j, j \sim i} K(i, j)(z(i) - z(j)). \tag{2.5}$$

where $j \sim i$ stands for the vertices j in the graph such that j is connected to i; i.e., $(i, j) \in E$.

The authors in [9, 29] propose a nonlocal regularization approach using the Dirichlet energy function:

$$R(\mathbf{z}) = \frac{1}{2} \sum_{i=1}^{n} \|\nabla z(i)\|^2 = \frac{1}{2} \sum_{i=1}^{n} \sum_{j, j \sim i} K(i, j)(z(i) - z(j))^2. \tag{2.6}$$

The regularization functional R essentially enforces the similar pixels of the image -as measured by the function $K(., .)$- to remain similar in the final estimate. By minimizing the above cost function with respect to the unknown $\mathbf{z}$, they recover the desired image. Note that the regularization term (2.6) can be expressed based on the un-normalized graph Laplacian $D - K$ as [9, 99, 114]:

$$R(\mathbf{z}) = \mathbf{z}^T (D - K)\mathbf{z}, \tag{2.7}$$

TABLE 2.1: Properties of different normalized graph Laplacians. Last row is our definition.

Reference	Graph Laplacian	Symmetric	DC eigenvector	Stochastic property
[68]	$I - D^{-1/2}KD^{-1/2}$	Yes	No	No
[104, 109]	$I - D^{-1}K$	No	Yes	$D^{-1}K$ is row-stochastic
ours	$\mathbf{I - C^{-1/2}KC^{-1/2}}$	**Yes**	**Yes**	**W is doubly-stochastic**

where $D = diag\{K\mathbf{1}_n\}$ is a diagonal matrix whose ith diagonal element is the sum of the elements of the ith row of K, and $\mathbf{1}_n$ is the n-dimensional vector of all ones. In [9, 29], the authors also introduce the Laplace operator associated to the normalized graph Laplacian $I - D^{-1/2}KD^{-1/2}$. However, they do not use this definition of the Laplacian in their formulation because of the fact that the output of this operator is not null when the input is constant.

In [36], the authors introduce the gradient-based and difference-based regularizing functionals, respectively as (we consider here their discrete versions):

$$J(\mathbf{z}) = \sum_{i=1}^{n} \phi(\|\nabla z(i)\|^2) = \sum_{i=1}^{n} \phi(\sum_{j,j\sim i} K(i,j)(z(j) - z(i))^2), \tag{2.8}$$

$$J_a(\mathbf{z}) = \sum_{i=1}^{n} \sum_{j,j\sim i} \phi(K(i,j)(z(j) - z(i))^2), \tag{2.9}$$

in which $\phi(s)$ is a positive function, convex in $\sqrt{s}$, with $\phi(0) = 0$. They consider the quadratic case $\phi(s) = s$, where the above functionals coincide. They also investigate the case $\phi(s) = \sqrt{s}$, for which nonlocal TV and anisotropic nonlocal TV functionals are derived from the gradient-based and difference-based approaches, respectively. They have applied their framework to inpainting and detecting and removing irregularities from textures.

In [104], Szlam, Maggioni, and Coifman propose function-adapted diffusion processes (using the random walk Laplacian $I - D^{-1}K$). They also propose a filtering procedure using a type of thresholding of the expansion coefficients of the input function on the linearly independent bases of the operator $D^{-1}K$. Reference [109] is also based on the same idea (expansion of the input data on the space spanned by the right eigenvectors of random walk Laplacian) for surface smoothing with weights derived *locally* in a non data-adaptive manner. In [10], a patch-based functional is considered for denoising 3-D image sequences acquired via fluorescence microscopy. This functional is based on minimizing a difference penalty term which is defined using the weighted difference between its patches rather than the weighted difference between its pixels. The minimizer of such a cost function can be equivalently expressed as a nonlocal filtering process; i.e., $\hat{\mathbf{z}} = D^{-1}K\mathbf{y}$. In [127], Zhang *et al.* propose two efficient algorithms for solving nonlocal TV-based image deconvolution[1]. They also provide a weight updating strategy within these iterative methods which was found to be *ineffective* in improving the performance of their algorithms. Therefore, they chose to compute the similarity weights only once from the simple Tikhonov regularization based deblurring estimate. Also, [82] proposes a regularization technique using total variation on nonlocal graphs for inverse problems, when the input data has undergone linear degradation as well as additive noise. Note that our deblurring algorithm uses a different nonlocal approach, in which the corresponding regularization term is defined using the normalizing coefficients derived from Sinkhorn's algorithm in [56]. Moreover, based in our experiments,

[1] As mentioned, the corresponding regularization term is derived using $\phi(s) = \sqrt{s}$ in Eq. (2.8).

the weight updating strategy is indeed effective in improving the performance of the proposed algorithm within the same quadratic framework. Furthermore, according to the analysis provided in [97], using the data fidelity term involving different derivatives of the residual is better able to model the underlying process for deblurring problems (especially for real motion-blurred images).

2.3 Derivation of Building Block Matrices of The Proposed Algorithm

In this section, we introduce the kernel similarity matrix K and a closely related doubly stochastic symmetric matrix W as the main filtering building blocks of our iterative algorithm from a graph point of view. Having these matrices at hand, we can define the normalized Laplacian matrix whose spectral properties are crucial for analyzing the behavior of the algorithm.

2.3.1 Kernel Similarity Matrix K And Filtering Matrix W

While our approach is general enough to include any valid kernel similarity function [44,69], the (i,j)th element of the kernel similarity matrix K is computed here using the nonlocal means (NLM) definition as [11]

$$K(i,j) = \exp(-\frac{\|\tilde{\mathbf{z}}_i - \tilde{\mathbf{z}}_j\|^2}{h^2}), \tag{2.10}$$

in which $\tilde{\mathbf{z}}_i$ and $\tilde{\mathbf{z}}_j$ are patches around the pixels i and j of the image $\tilde{\mathbf{z}}$, and h is a smoothing parameter. Note that at each outer iteration, the kernel similarity weights are re-computed from the estimate at the previous iteration. As a result of the above definition for the kernel similarity weights, the matrix K would be a symmetric non-negative matrix. Furthermore, we only compute the similarity between each patch and a small neighborhood of patches around it (e.g., a search window of size 11×11 of patches around each patch). Therefore, the matrix K is sparse. This sparse structure is appealing from a computational point of view.

Applying Sinkhorn matrix balancing procedure [100] to the matrix K yields the doubly stochastic filtering matrix[2] W. We use a recent fast version of the original algorithm for symmetric non-negative matrices [56]. This balancing algorithm returns a diagonal scaling matrix $C^{-1/2}$, such that the resulting matrix $W = C^{-1/2}KC^{-1/2}$ is a symmetric non-negative doubly stochastic matrix. Since W is symmetric, it can be decomposed as $W = VSV^T$, where V is an orthonormal matrix whose columns are the eigenvectors of W, and $S = diag\{\lambda_1, \lambda_2, ..., \lambda_n\}$ is a diagonal matrix consisting of eigenvalues of W as its diagonal elements. Moreover, since W is doubly stochastic, it has unity spectral radius [69]. The largest eigenvalue is exactly equal to 1 with the corresponding DC eigenvector $\mathbf{v}_1 = (1/\sqrt{n})[1, 1, ..., 1]^T = (1/\sqrt{n})\mathbf{1}_n$ [69]. Intuitively, this means that applying W to a signal preserves the DC component of the signal. This is a desirable property for filtering purposes. Note that the spectral analysis of the matrix W reveals its inherent low-pass nature (the largest eigenvalue corresponds to

[2] A matrix with non-negative entries is doubly stochastic if each of its rows and each of its columns sum to 1.

the DC component) [70].

2.3.2 Normalized Graph Laplacian Matrix

At this point, we define our normalized graph Laplacian matrix as

$$I - W = I - C^{-1/2}KC^{-1/2}. \tag{2.11}$$

This is the proper definition of the normalized graph Laplacian matrix for image filtering purposes, as opposed to its earlier definition in graph theory literature $I-D^{-1/2}KD^{-1/2}$ [99,114]. It is worthwhile comparing this traditional definition of the normalized Laplacian with our proposed definition which is based on a very different scaling of the similarity matrix K using matrix balancing [56]. Our definition of the normalized Laplacian ($I - W = I - C^{-1/2}KC^{-1/2}$) is symmetric, positive semi-definite, with the zero eigenvalue associated to the constant eigenvector $\frac{1}{\sqrt{n}}\mathbf{1}_n$. Hence, when applied to a constant function, it returns a zero vector. The traditional definition of the normalized graph Laplacian lacks the desired filtering property of having DC eigenvector as one of the basis eigen functions [99]. As a result, the definition of normalized graph Laplacian in (2.11) is proposed and used in this dissertation. This definition has the desired spectral properties for our specific applications as well as a nice filtering interpretation. In fact, the set of eigenvectors of $I - W$ can be considered as the basis functions of the underlying graph, and its eigenvalues can be thought of as the corresponding graph frequencies. Also, note that the Laplacian $I - W$ has a high-pass filtering nature (with null eigenvalue corresponding to the DC eigenvector). This property is consistent with

the expected behavior of the Laplacian filter in image processing. Consequently, when applied to an image, $I - W$ can be directly interpreted as a data-adaptive Laplacian filter. Therefore, it enables us to incorporate different types of filters in the data term coupled to the regularization term based on the application at hand. In the "random walk" Laplacian $I - D^{-1}K$, (from the theory of Markov chains), the (i, j)th element of $D^{-1}K$ represents the probability of moving from node i to node j of the graph in one step, given that we are in node i [67]. A similar random walk interpretation can be provided by our symmetric doubly stochastic filtering matrix $W = C^{-1/2}KC^{-1/2}$, with analysis and performance advantages over $D^{-1}K$ for image filtering, as discussed in [70] [3]. Furthermore, for image deblurring applications, another advantage is that our resulting linear equations are symmetric and positive definite, providing us with fast methods for solving large linear systems of equations with optimization methods like CG.

In order to better demonstrate the different expressions of the difference and Laplacian operators as well as the regularization term corresponding to our normalized Laplacian, we state them here. We can define the difference operator corresponding to the proposed normalized graph Laplacian as:

$$dz(i,j) = \sqrt{K(i,j)}\left(\frac{z(j)}{\sqrt{C(j,j)}} - \frac{z(i)}{\sqrt{C(i,i)}}\right), \tag{2.12}$$

where $C(j, j)$ and $C(i, i)$ are the corresponding j and ith diagonal elements of the diagonal matrix C derived from the matrix balancing algorithm [56, 70]. From the

[3] In fact, W can be thought of as the transition probability matrix of the Markov chain defined on the graph.

above equation along with the definition of the divergence operator [9], the Laplace operator corresponding to the normalized Laplacian $I - C^{-1/2}KC^{-1/2}$ is:

$$\triangle z(i) = \frac{1}{\sqrt{C(i,i)}} \sum_{j,j\sim i} K(i,j)(\frac{z(i)}{\sqrt{C(i,i)}} - \frac{z(j)}{\sqrt{C(j,j)}}), \tag{2.13}$$

As a result, our proposed regularization term can be written as:

$$R(\mathbf{z}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j,j\sim i} K(i,j)(\frac{z(i)}{\sqrt{C(i,i)}} - \frac{z(j)}{\sqrt{C(j,j)}})^2 = \mathbf{z}^T(I-W)\mathbf{z}. \tag{2.14}$$

Note that the Laplace operator in (2.13) describes the effect of our normalized Laplacian at each pixel i, when applied to an input vector $\mathbf{z}$. As the Laplace operator is a second order derivative operator, the name Laplacian for the corresponding matrix operator is appropriate, and common in graph theory. In the next section, we will describe how to use the proposed graph Laplacian to develop a new restoration algorithm.

2.4 Proposed Deblurring Method

As depicted in Fig. 2.1, the proposed algorithm consists of inner and outer iterations. The reason is that for computing the data-adaptive matrix K, a good rough estimate of the underlying unknown image is needed. This estimate is gradually improved as we proceed through iterations. In each outer iteration, the matrix W is computed once and used to define the following objective function to be minimized with respect to the unknown image $\mathbf{z}$

$$E(\mathbf{z}) = (\mathbf{y} - A\mathbf{z})^T\{I + \beta(I-W)\}(\mathbf{y} - A\mathbf{z}) + \eta\ \mathbf{z}^T(I-W)\mathbf{z}, \tag{2.15}$$

where $\beta \geq 0$ and $\eta > 0$ are the parameters to be tuned based on the amount of noise and blur. Note that in the above objective function, data and prior terms are coupled via the matrix W. This coupling is controlled by means of the parameter β. The first term favors a solution $\mathbf{z}$ such that its blurred and then filtered version is as close as possible to the filtered version of the input $\mathbf{y}$. Frequency selectivity of this common filter is determined by the parameter β according to the amount of the noise and blur. The second term is essentially a data-adaptive difference term favoring certain smoother solutions based on the structure of the underlying data encoded in the normalized Laplacian matrix $I - W$, defined in the previous section.

Let us take a look at the cost function in (2.15) from a filtering point of view. This filtering interpretation provides a more intuitive perspective on the objective function. For this purpose, Eq. (2.15) is rewritten in the following form

$$E(\mathbf{z}) = \|\{I + \beta(I - W)\}^{1/2}(\mathbf{y} - A\mathbf{z})\|^2 + \eta\|(I - W)^{1/2}\mathbf{z}\|^2. \tag{2.16}$$

Note that $I + \beta(I - W) = V\Lambda V^T$ is a symmetric and positive semi-definite matrix. Therefore, the matrix $\{I + \beta(I - W)\}^{1/2} = V\Lambda^{1/2}V^T$ has a filtering behavior similar to that of $I + \beta(I - W)$. Once we have the eigendecomposition of the filtering matrix W, the ith diagonal element of the matrix Λ can be written in terms of the associated ith diagonal element of S (that is λ_i) as $1 + \beta(1 - \lambda_i)$. Since the matrix $I - W$ is a high-pass filter, with $\beta > 0$, $I + \beta(I - W)$ behaves like a sharpening filter on the residuals $\mathbf{y} - A\mathbf{z}$, and so does $\{I + \beta(I - W)\}^{1/2}$. According to the analysis provided in [97], using the data fidelity term involving different derivatives of the residual is better

able to model the underlying phenomenon for deblurring problems (especially for real images). The same analysis applies to the second term in (2.16), where both Laplacian $I-W$, and its square root $(I-W)^{1/2}$, are adaptive high-pass filters. Consequently, the resulting regularization expression in (2.16) adaptively penalizes high frequencies in the final solution to avoid unpleasant artifacts due to the noise amplifications and ringing artifacts while maintaining fine details in the restored image.

In order to minimize the cost function in (2.15) at each step, the corresponding gradient is set equal to zero as

$$\nabla E(\mathbf{z}) = -2A^T\{I+\beta(I-W)\}(\mathbf{y}-A\mathbf{z}) + 2\eta(I-W)\mathbf{z} = \mathbf{0}, \tag{2.17}$$

which results in the following symmetric positive definite system of liner equations

$$(A^T\{I+\beta(I-W)\}A + \eta(I-W))\mathbf{z} = A^T\{I+\beta(I-W)\}\mathbf{y}. \tag{2.18}$$

Conjugate Gradient is then used to solve the above system. Also, note that A and A^T are interpreted as blurring with the PSF or its flipped version, respectively. Our experiments show that three outer iterations suffice to get the desired deblurred output in most cases. Also, note that the only restriction on the parameter β is that it should be selected such that the corresponding system of linear equations in (2.18) remains positive definite. The matrix $I+\beta(I-W)$ is also required to be positive semi-definite for the existence of its square root in the data fit term in (2.16). A sufficient condition is $\beta \geq 0$.

Algorithm 1 iterative restoration algorithm

Inputs: blurred noisy image $\mathbf{y}$, blurring matrix A

Output: deblurred estimate $\mathbf{z}^\star$

Initializations:

1. Estimate the noise standard deviation $\hat{\sigma}$ in $\mathbf{y}$ using algorithm in [47].

2. Denoise $\mathbf{y}$ using denoising algorithm in [21] to derive $\hat{\mathbf{z}}^{(0)}$.

3. Set $k = 0$.

while not converged **do**

- Compute K from $\hat{\mathbf{z}}^{(k)}$ using Eq. (2.10).

- Apply Sinkhorn algorithm in [56] to K to get the diagonal matrix $C^{-1/2}$.

- Compute the filtering matrix as $W = C^{-1/2}KC^{-1/2}$.

- Solve objective function in (2.15) using CG to compute $\hat{\mathbf{z}}^{(k+1)}$.

- Set $\mathbf{z}^\star = \hat{\mathbf{z}}^{(k+1)}$, and $k = k + 1$.

end while

return $\mathbf{z}^\star$

2.4.1 Spectral Analysis of The Overall Deblurring Algorithm

For analysis purposes, we are able to provide a filtering interpretation of the final estimate at each outer step of the algorithm. Note that the minimizer of the cost function in (2.16) can be expressed as:

$$\hat{\mathbf{z}} = F(A, W)A^T(I + \beta(I - W))\mathbf{y}, \tag{2.19}$$

where

$$F(A, W) = \{A^T(I + \beta(I - W))A + \eta(I - W)\}^{-1}. \tag{2.20}$$

Eq. (2.19) can be interpreted as (1) filtering $\mathbf{y}$ by $I + \beta(I - W)$, (2) back projection through multiplication by the transpose of the blurring matrix A, and (3) applying the symmetric matrix $F(A, W)$. In other words, if we consider the spectral decomposition of this symmetric matrix as $F(A, W) = \Theta\Upsilon\Theta^T$, the columns of the matrix Θ serve as an orthonormal basis for filtering the vector $A^T(I + \beta(I - W))\mathbf{y}$, thereby providing a spectral filtering interpretation for the corresponding deblurring solution at each outer step of the algorithm. Since an inverse operation is involved in (2.19), we consider a simple experiment investigating the condition number of the matrix $A^T(I+\beta(I-W))A+\eta(I-W)$. For this purpose, we use the MATLAB code in [41] to explicitly construct the blurring matrix A related to out-of-focus blur with radius 7. Table 2.2 illustrates the condition number of $A^T(I+\beta(I-W))A+\eta(I-W)$ for different values of the parameters η and β. The condition numbers of $A^T(I + \beta(I - W))A + \eta(I - W)$ for different values of η in comparison to the condition number of A^TA show the effectiveness of our procedure for regularizing the ill-posed deblurring problem and the corresponding linear

TABLE 2.2: Condition number of $(A^T(I+\beta(I-W))A+\eta(I-W))$ for different values of η and β and blurring matrix A corresponding to out-of-focus blur with radius 7. The condition number of A^TA is 5.74×10^{20}.

	$\eta=0.05$	$\eta=0.1$	$\eta=0.2$	$\eta=0.3$
$\beta=0.1$	2.02×10^3	1.17×10^3	705	459
$\beta=0.5$	1.97×10^3	1.15×10^3	700	402
$\beta=0.7$	1.95×10^3	1.14×10^3	699	543

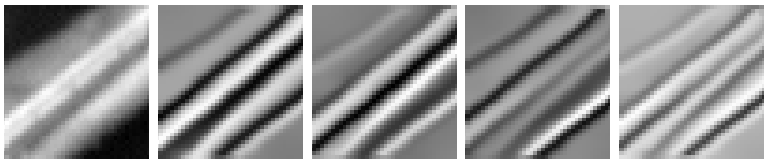


Fig. 2.3: (a) From left to right: original 41×41 image, and the eigenvectors of $F(A,W)$ corresponding to the four largest eigenvalues for $\beta=0.7$ and $\eta=0.2$.

equations. Also, the basis eigenvectors in Θ corresponding to the four largest eigenvalues of $F(A,W)$ are depicted in Fig. 2.3. As can be seen in Fig. 2.3, the eigenvectors associated with the largest eigenvalues of $F(A,W)$ indicate the data-adaptive nature of the corresponding filter.

2.5 Image Sharpening As a Special Case of The Proposed Objective Function

It is interesting to consider a special case of the above objective function in (2.15) for image sharpening when the input image contains a moderate blur, but no

information about the blurring process is available. In such cases, one can resort to the following cost function

$$E(\mathbf{z}) = (\mathbf{y} - \mathbf{z})^T \{I + \beta(I - W)\}(\mathbf{y} - \mathbf{z}), \tag{2.21}$$

which comes from Eq. (2.15), by setting $A = I$ and $\eta = 0$. Optimizing the above objective function using gradient descent (GD), yields[4]:

$$\hat{\mathbf{z}}_\ell^{(1)} = \hat{\mathbf{z}}_{\ell-1}^{(1)} + \mu\{I + \beta(I - W)\}(\mathbf{y} - \hat{\mathbf{z}}_{\ell-1}^{(1)}). \tag{2.22}$$

By selecting the step size parameter $\mu = 1$, and with zero initialization of the GD iterations in (2.22); i.e., $\hat{\mathbf{z}}^{(0)} = \mathbf{0}$, the first iteration takes the form

$$\hat{\mathbf{z}}_1^{(1)} = \{I + \beta(I - W)\}\mathbf{y}. \tag{2.23}$$

For $\beta > 0$, Eq. (2.23) can be interpreted as data-adaptively adding to the input image some amount of its high-pass filtered version. This procedure results in a sharper image. Although there is no access to the exact PSF, since the matrix W is computed from the input blurred image, it contains some information about the original image as well as the blurring process. Therefore, Eq. (2.23) provides us with a data-adaptive sharpening (or to say rough deblurring) technique.

2.6 Implementation Details

The first step of the iterative algorithm is to compute the kernel similarity matrix[5] K. At each outer iteration k, we compute this matrix from the final estimate of

[4]We consider the first outer iteration here.

[5]This is computed from the denoised version of the input image at the beginning of the algorithm.

Fig. 2.4: Set of color images used for evaluation of our method: (a)Building image (480×640), (b)Bikes image (494×494) , (c)Girl image (496×700), (d)Street image (480×640), (e)Boat image (420×520), and (f)Book shelf image (580×520).

the previous step, i.e., from $\hat{\mathbf{z}}^{(k-1)}$, as shown in Fig. 2.1. The values of the regularization parameters η and β are selected based on the noise variance and blurring scenario, and are kept fixed at each step of the algorithm, for all the test images. For deblurring examples, for instance, the parameter β lies in the range $(0, 1)$, and the parameter η is empirically selected in the range $(0, 0.4)$. The closer is β to 1, the larger is the effect of the data-adaptive high-pass filter $I + \beta(I - W)$ in the data term of the cost function in (2.15), which results in encouraging higher frequencies of $A\mathbf{z}$ to be close to those of $\mathbf{y}$. Similarly, the larger is the value of η, the more penalty is put on the norm of the

TABLE 2.3: SSIM and PSNR performance of the kernel similarity-based algorithm and IDDBM3D [22] for Gaussian blur kernel of size 25×25 with standard deviation 1.6 and out-of-focus blur generated using disk function of radius 7. In each cell, the first number denotes SSIM value, and the second number represents PSNR value in dB.

Blur	Building		Bikes		Girl		Street		Boat		Book shelf	
	ours	[22]	ours	[22]	ours	[22]	ours	[22]	ours	[22]	ours	[22]
Gaussian	0.972	0.979	0.976	0.979	0.969	0.975	0.973	0.977	0.964	0.965	0.982	0.984
($\sigma^2 = 0.2$)	28.57	29.30	27.56	28.36	33.18	33.84	28.80	30.29	29.11	29.49	27.25	28.15
out-of-focus	0.951	0.963	0.954	0.960	0.937	0.948	0.952	0.961	0.928	0.931	0.971	0.970
($\sigma^2 = 0.2$)	28.44	29.64	26.92	28.08	32.28	33.00	29.71	31.35	28.79	28.75	27.60	28.19
Gaussian	0.957	0.968	0.962	0.967	0.952	0.963	0.962	0.967	0.944	0.950	0.971	0.978
($\sigma^2 = 1$)	27.59	28.62	26.24	27.22	32.34	32.97	28.34	29.33	28.00	28.62	26.27	27.31
out-of-focus	0.917	0.936	0.897	0.917	0.905	0.920	0.905	0.934	0.865	0.883	0.940	0.953
($\sigma^2 = 1$)	26.90	27.80	25.04	25.75	30.85	31.42	27.19	29.00	26.77	27.35	25.63	26.64

high-pass filtered version of the desired solution $\mathbf{z}$. For instance, larger values of η and smaller values of β are used when the amount of noise is high in the input image, and the image is moderately blurred. Similarly, when the amount of noise is low while the image is severely blurred, larger values of β and smaller values of η are used. These tunings are done for each scenario of blur and noise for a set of test images to have visually pleasant results, and are kept fixed for all other input images with the same degree of degradation, as shown in the next section.

In experiments, in order to avoid noise amplification and ringing artifacts, the maximum number of inner and outer iterations are set beforehand based on the amount of degradation[6], and then the iterations are stopped using a rough estimate of

[6]For more blurry images, we need more iterations for convergence of CG iterations. Also, as we initialize the CG iterations with more enhanced images as we proceed through the outer iterations, the number of inner iterations is decreased by a step nDec, as the number of outer iterations increases.

Predicted-MSE (PMSE) measure as[7]

$$\widehat{PMSE}(q,k) = \frac{1}{n}\|\widehat{A\mathbf{z}} - A\hat{\mathbf{z}}_k^{(q)}\|^2, \tag{2.24}$$

where $\widehat{A\mathbf{z}}$ is an estimate of the blurred clean image ($A\mathbf{z}$), which is derived by denoising input noisy blurry image, and $\hat{\mathbf{z}}_k^{(q)}$ is the corresponding estimate of the desired image at the kth CG iteration of the qth outer iteration. That is, we stop CG iterations whenever $\widehat{PMSE}(q,k+1) > \widehat{PMSE}(q,k)$.

There are two main computational burdens for the algorithm. First is the computation of the kernel similarity coefficients, where its special form allows us to take advantage of the idea of integral images [46]. This technique is very effective to reduce the computational complexity of the algorithm. Second is the matrix-vector products required at each iteration of CG method for optimizing the objective function in (2.15). However, because of the special structure of the matrices involved, it is possible to implement the algorithm using Fast Fourier Transform (FFT) and fast sparse matrix-vector products. It is also possible to exploit the symmetric structure of the kernel similarity matrix K (and of course that of W) to reduce memory requirements.

2.7 Numerical Experiments

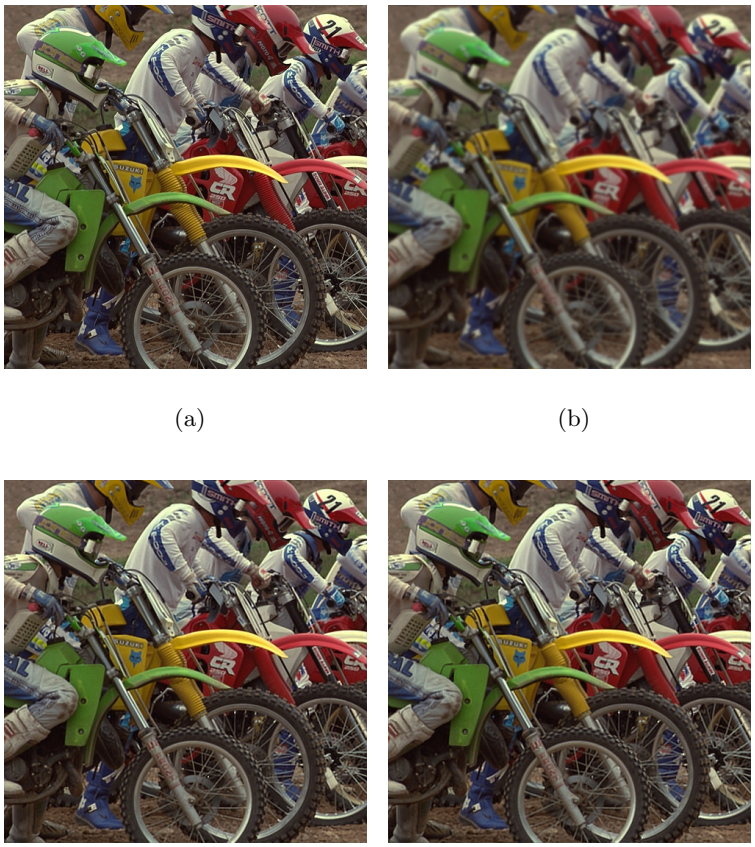
In this section, the effectiveness of our iterative approach is verified through a number of synthetic and real experiments. Throughout the deblurring experiments, our focus is on more practical cases with severe blur and small amount of noise in

[7]Predicted-MSE is defined as $PMSE(q,k) = \frac{1}{n}\|A(\mathbf{z} - \hat{\mathbf{z}}_k^{(q)})\|^2$ [111].

TABLE 2.4: Set of parameters in different synthetic color image deblurring examples in this chapter. nInner is the maximum number of the inner CG iterations.

Experiment	η	β	h	nInner
Symmetric ($\sigma^2 = 0.2$)	0.003	0.2	5.5	100
Symmetric ($\sigma^2 = 1$)	0.008	0.001	7.5	100
motion blur ($\sigma^2 = 0.2$)	0.006	0.4	6	100
motion blur ($\sigma^2 = 1$)	0.01	0.01	6.5	80

the captured images. We report experiments for Gaussian, out-of-focus, and nonlinear camera motion blur. For all cases, we have compared the performance of our algorithm with some of the best existing non-blind deblurring algorithms. Also, for both motion and out-of-focus blurs, the iterative algorithm is applied to real images to evaluate its performance for such more complicated cases. Since the proposed method is a non-blind deblurring algorithm, for real deblurring examples, we use PSFs derived from other existing blur kernel estimation methods. For this purpose, in case of real out-of-focus blur, the PSF is estimated using 'deconvblind' MATLAB function. In case of real motion deblurring, the estimated PSFs from [15, 97] are used. For color images, the proposed deblurring algorithm is applied independently to R, G, and B channels of the input color image to get the final estimate. In all the experiments, object oriented MATLAB functions in [72] are used for performing matrix-vector products of the form $A\mathbf{z}$ and $A^T\mathbf{z}$. PSNR in dB and the SSIM index are used for comparison purposes [117]. SSIM index is shown to be a more reliable metric for comparison of deblurring algorithms than the widely used PSNR measure [117].



(a) (b)

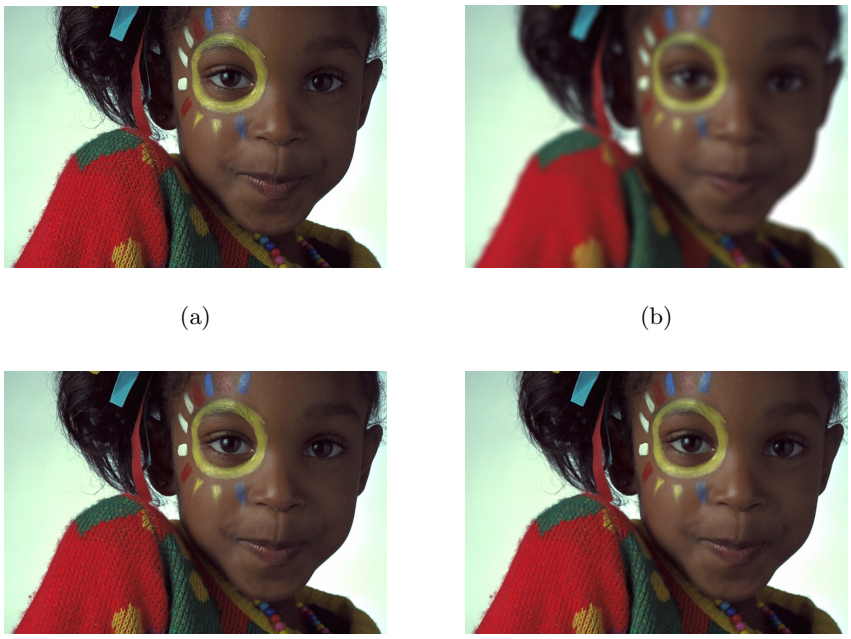
(c) (d)

Fig. 2.5: Deblurring example with Gaussian blur: (a)clean Bikes image, (b)blurred noisy image, (c)output of [22], and (d)output of our algorithm.

2.7.1 Symmetric Blurs

Two kinds of symmetric blur are considered for these examples: Gaussian blur and out-of-focus blur. A 25×25 Gaussian blur with standard deviation 1.6 is convolved with a set of color images[8] shown in Fig. 2.4. Also, out-of-focus blur is produced using a

[8]Test images are from Kodak Lossless True Color Image Suite (http://r0k.us/graphics/kodak/) and the web page for [40].

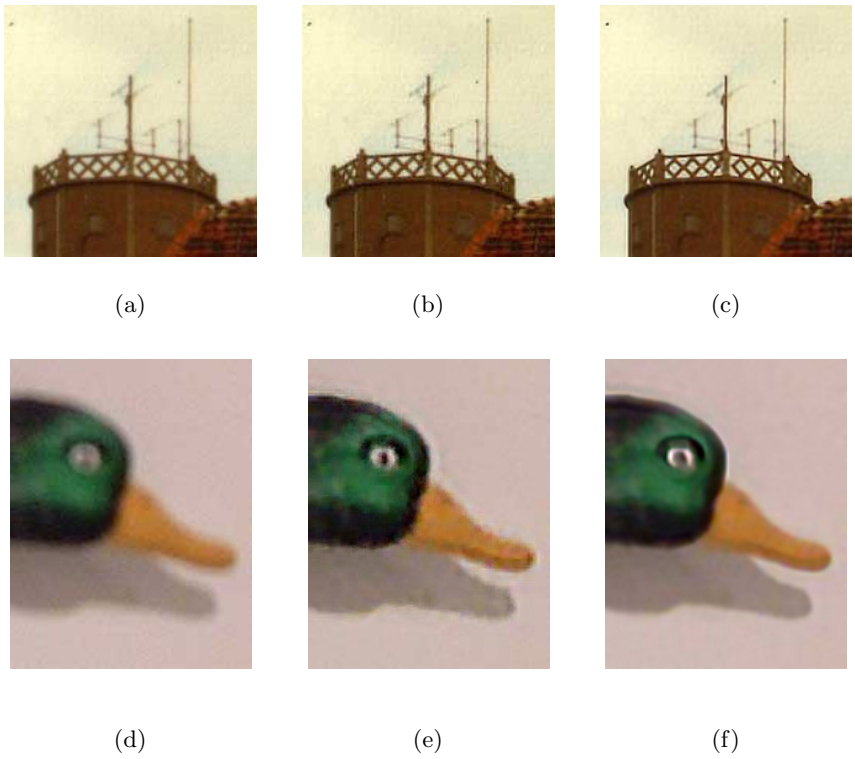


(a) (b)

(c) (d)

Fig. 2.6: Deblurring example with out-of-focus blur: (a)clean Girl image, (b)blurred noisy image, (c)output of [22], and (d)output of our algorithm.

disk function with radius 7 and is used to generate the corresponding blurred examples. Then, additive white Gaussian noise with variances equal to 0.2 and 1 is added to the blurred images. We compare the performance of our algorithm with that of IDDBM3D algorithm in [22]. Periodic boundary conditions are used in these examples [42]. Also, we use patch size of 5×5, search neighborhood size of 11×11, number of outer iterations equal to 3, and the step decrement of the number of inner iterations equal to 30 in these experiments. The values of the parameters η, β, h, and maximum number of inner CG iterations have been summarized in Table 2.4. Note that, the parameters of both



(a) (b) (c)

(d) (e) (f)

Fig. 2.7: Real sharpening examples: left column: input blurred noisy image, middle column: output of Focus Magic software at http://www.focusmagic.com, and right column: output of our algorithm ($\eta = 0.1$ and $\beta = 0.06$ for both images).

algorithms are set for best performance in each case for this set of images, and for the given blurs and noise variances.

Figures 2.5 and 2.6 depict deblurring outputs of our algorithm compared to those of IDDBM3D for noise variance of 0.2 and synthetic Gaussian and out-of-focus blurs, respectively. Also, Table 2.3 summarizes the numerical deblurring results. As can be seen in Table 2.3, our kernel similarity-based algorithm shows very close performance to the state-of-the-art IDDBM3D algorithm in [22] in the case of Gaussian blur. Also,

our iterative algorithm performs acceptably in the case of out-of-focus blur. In some cases, our algorithm exhibits slightly better visual quality as can be seen, e.g., in smooth parts of the face of the Girl image in Fig. 2.6. There is one key difference between our proposed algorithm and IDDBM3D. IDDBM3D is a two step algorithm, in which denoising and deblurring are decoupled. Each step of IDDBM3D essentially involves solving two different objective functions, one for deblurring and the other for denoising. Regarding the computational complexity, even though our algorithm has been written entirely in MATLAB (except the initial denoising step which we use the code provided by the authors in [21]) [9], our algorithm runs faster, making it more appropriate for practical image deblurring applications. To be more specific, for a 480×640 color image, the MATLAB implementation of our kernel similarity-based method runs 4 times faster than the code for IDDBM3D run on a 2.8 GHz Intel Core i7 processor. Furthermore, our method just relies on an initial denoising, whereas IDDBM3D depends on an appropriate estimate from another deblurring algorithm in its grouping phase. In addition, as we demonstrate in the remaining experiments, the proposed method has the flexibility to be applied to a wide variety of blurs including both symmetric and non-symmetric blurs, while IDDBM3D has been designed and tested specifically for symmetric blurs.

Figure 2.7 shows the output of our algorithm when applied to real noisy and out-of-focus blurred images compared to the outputs of the Focus Magic deblurring software. As can be seen, our algorithm is better able to handle noise amplification related issues. In the following subsections, we consider the effect of different factors on

[9] In contrast, the computational demanding parts of IDDBM3D have been implemented in C++ using MATLAB mex files.

the performance of the proposed deblurring method.

2.7.1.1 Effect of The Patch Size on The Performance of The Proposed Algorithm

In this subsection, we add an experiment investigating the relationship between the patch size and blur kernel width in the deblurring algorithm for the case of out-of-focus blur. Synthetic examples are produced by applying out-of-focus blur kernels with radii 5, 7, 9, and 11 to the Girl image. White Gaussian noise with variance $\sigma^2 = 0.4$ is also added to the blurred images. The corresponding SSIM values are plotted in Fig. 2.8 versus the patch size for different radii of the out-of-focus blur[10]. It can be seen that for out-of-focus blur, the best performance is not strongly dependent on the patch size regardless of the out-of-focus blur kernel radius. It shows that for such blur kernels, the structure around each pixel is described well just by considering a small 5×5 patch around it. In other words, there is no specific relation between the patch size and the size of the blur kernel. In fact, one can fix this parameter and change other parameters like the regularization parameter η to control the quality of the output image.

2.7.1.2 Effect of The Smoothing Parameter h on The Performance of The Proposed Algorithm

In order to investigate the effect of the parameter h, we apply out-of-focus blur with radius 7 to the clean Girl image and then add noise with two different variances (0.2

[10]We consider patch sizes 3, 5, and 7.

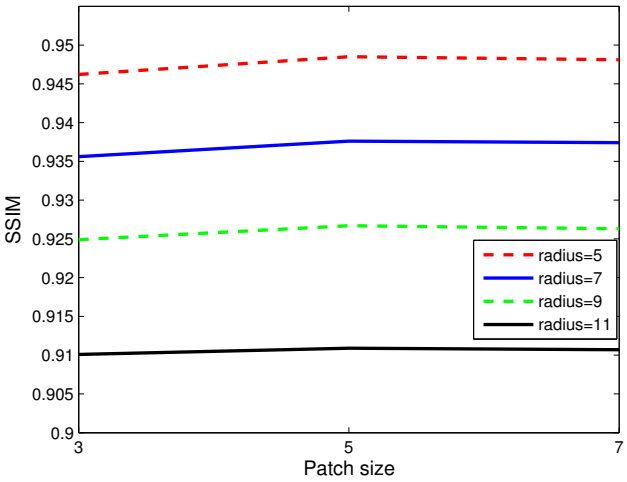


Fig. 2.8: The relationship between the patch size and blur kernel width for Girl image and out-of-focus blur with different radii.

and 1) to it. The results are shown in Fig. 2.9. Note that the same set of parameters are used for both cases of noise variance. As can be seen, with the same set of parameters, the optimal smoothing parameter h is greater for the higher noise level. Also, it is evident from Fig. 2.9 that the algorithm is not very sensitive to the selection of this parameter.

2.7.1.3 Effect of The Initialization of The CG Iterations on The Performance of The Algorithm

In order to investigate the effect of the specific initialization for CG iterations at first step of the deblurring algorithm, we consider a simple experiment with Girl image for Gaussian blur with standard deviation 1.6 and noise variance $\sigma^2 = 1$. We consider two different initializations, of CG iterations with (1) the denoised version of the input noisy and blurred image versus (2) initializing it with zero image, i.e., $\hat{\mathbf{z}}_0 = \mathbf{0}$.

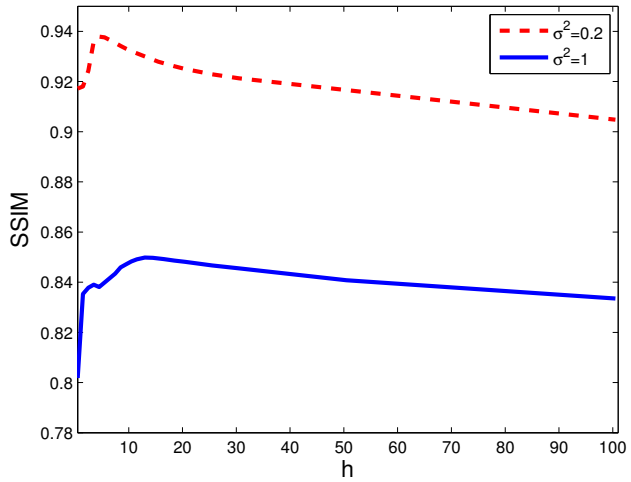


Fig. 2.9: The effect of the smoothing parameter h in the kernel similarity function for Girl image and out-of-focus blur with radius 7 for two different noise variances.

As can be seen in Fig. 2.10, in case of initializing with the denoised image, the algorithm converges faster. However, with simple zero initialization, we obtain the same result, but after more CG iterations. The proposed algorithm is not sensitive to the initialization.

2.7.1.4 Effect of Oracle Scenario on The Performance of The Proposed Algorithm

It is instructive to show reconstructions starting from the oracle scenario where the true images are used for the weights and applied to noisy data. We consider a deblurring scenario in which test images are synthetically blurred with 25×25 Gaussian blur kernel with standard deviation 1.6 and the additive white Gaussian noise with variance 0.4 is added. Then, we compare the performance of the proposed algorithm, when the similarity weights are computed from the oracle images compared to the cases when these weights are derived from the given input (blurred, noisy) images. The results

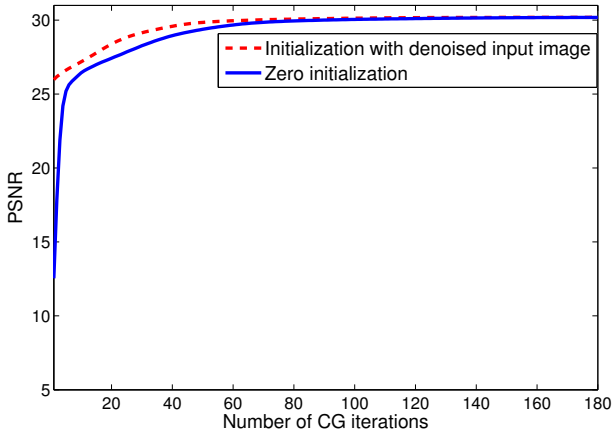


Fig. 2.10: Convergence plots of the CG iterations for different initializations.

TABLE 2.5: PSNR oracle performance of the proposed algorithm vs. the low-noise case.

Image	Girl	Bikes	Street	Building
Oracle	33.06	27.27	29.07	28.43
Algorithm 1	32.42	26.29	28.11	27.95

are summarized in Table 2.5. As expected, using the oracle image for computing the similarity weights improves the performance of the algorithm. Also, it can be seen that for practical cases when the amount of noise is low, the results of the algorithm are not much different from their oracle counterparts.

2.7.1.5 Effect of The Noise Level in The Computation of The Similarity Weights

We add an experiment considering the effect of noise in computing the similarity weights. The Girl image is blurred by out-of-focus blur (of radius 7), and different

TABLE 2.6: ISNR values for the Girl image and out-of-focus blur with different noise standard deviation when the weights are computed from the input noisy blurred image.

	$\sigma = 1$	$\sigma = 3$	$\sigma = 6$	$\sigma = 9$	$\sigma = 20$	$\sigma = 30$
ISNR	4.20	3.63	3.73	4.25	6.54	8.87

amounts of noise (up to standard deviation equal to 30) are added to the blurred image. We fix all the parameters for different noise levels except the smoothing parameter h which is changed proportional to the noise standard deviation as $h = 0.05\sigma$. In all the experiments, the similarity weights are computed from the noisy blurred input image in the first step of the algorithm. Table 2.6 shows the ISNR values for different noise levels[11]. As can be seen even for very high noise levels and with computation of the weights from the input image, the algorithm is able to provide improvement with respect to the input degraded image. Although one can assume that the algorithm will fail at some noise level, we can conclude that it is robust enough to the errors in the input image for computing the similarity weights.

2.7.2 Synthetic Motion Blur

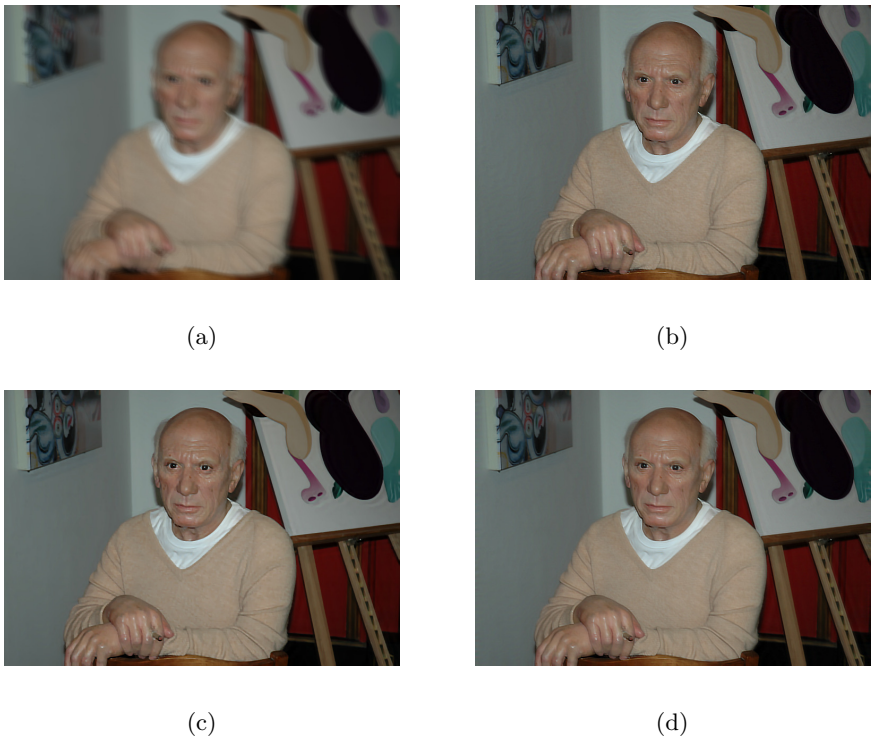
For assessing the performance of the algorithm in case of motion deblurring, we use the complex camera motion blur kernel provided by Shan et al. [97]. Again, noise with variances equal to 0.2 and 1 is added to the blurred images. In this case, we compare the proposed algorithm with two of the best available non-blind motion

[11]ISNR is defined as the difference between the PSNRs of the output image and the input noisy blurred image.

TABLE 2.7: SSIM and PSNR performance of the kernel similarity-based algorithm in comparison with those of motion deblurring methods in [58] and [97] for synthetic camera motion blur. In each cell, the first number denotes SSIM value, and the second number represents PSNR value.

Image	$\sigma^2 = 0.2$			$\sigma^2 = 1$		
	ours	[58]	[97]	ours	[58]	[97]
Building	0.9743 29.47	0.9734 29.74	0.9741 29.51	0.9525 28.45	0.9507 28.48	0.9572 28.45
Bikes	0.9756 28.04	0.9731 28.26	0.9692 26.95	0.9589 26.89	0.9470 26.45	0.9496 26.51
Girl	0.9636 33.54	0.9624 33.53	0.9586 33.15	0.9354 32.15	0.9351 31.99	0.9379 32.08
Street	0.9791 31.11	0.9766 31.50	0.9741 31.00	0.9584 29.75	0.9507 29.32	0.9558 29.35
Boat	0.9634 29.91	0.9687 29.81	0.9532 28.37	0.9342 28.57	0.9311 28.38	0.9344 28.27
Book shelf	0.9857 28.81	0.9839 29.24	0.9821 28.42	0.9711 27.31	0.9680 27.61	0.9693 27.21

deblurring works. Periodic boundary conditions are used in these examples [42]. Also, we use patch size of 5×5, search neighborhood size of 11×11, number of outer iterations equal to 3, and the step decrement of the number of inner iterations equal to 30 in these experiments. The values of the parameters η, β, h, and maximum number of inner CG iterations have been summarized in Table 2.4. The parameters of all algorithms are set for best performance. Table 2.7 illustrates the quantitative results in this case. From the numerical results in Table 2.7, it is evident that our proposed kernel similarity-based algorithm shows very good performance in the case of non-blind nonlinear motion deblurring.



(a) (b)

(c) (d)

Fig. 2.11: Real motion deblurring example: (a)input blurred noisy image, (b)output of hyper-Laplacian algorithm [58], (c)output of [97], and (d)output of our algorithm ($\eta = 0.031, \beta = 0.6$).

2.7.3 Real Motion Deblurring

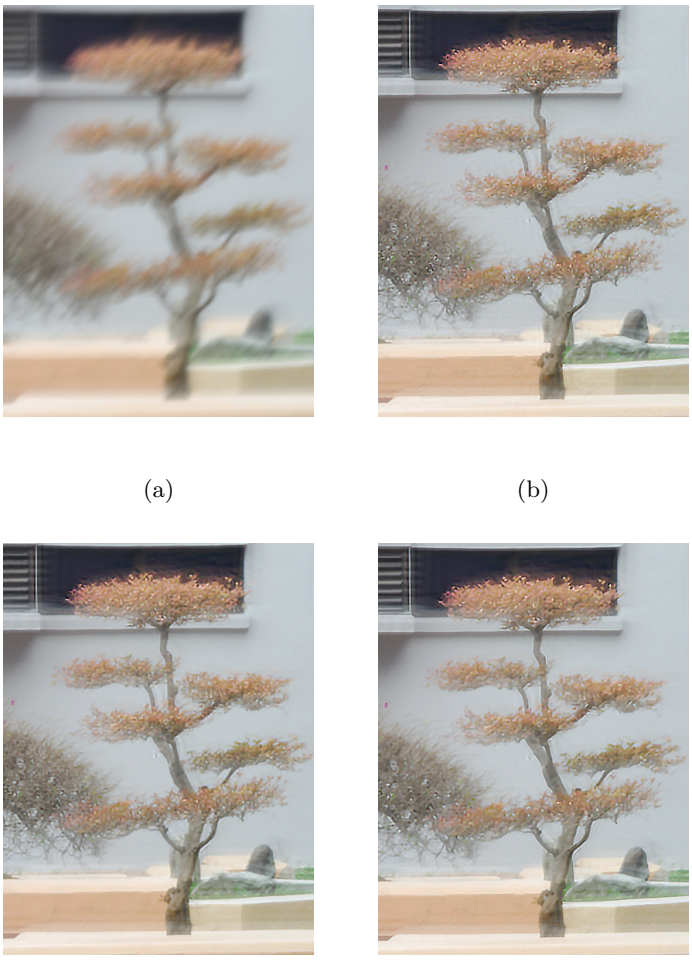
Now, we deal with more challenging motion blur situations where the blur kernel is estimated using two of the existing blur kernel estimation methods [15,97] from real motion blurred test images. The estimated blur kernels are used independently to derive the final deblurred images. The performance of the proposed method is compared with those of [97] and [58], which are among the best non-blind motion deblurring methods.

(a) (b)

(c) (d)

Fig. 2.12: Real motion deblurring example: (a)input blurred noisy image, (b)output of hyper-Laplacian algorithm [58], (c)output of [97], and (d)output of our algorithm ($\eta = 0.25, \beta = 2.5$).

For all examples, the patch size and search window size are selected to be 5×5 and 11×11, respectively. Also, we use reflective boundary conditions for these experiments [42,72]. Figures 2.11, 2.12, and 2.13 show the outputs of different methods, when the blur kernels are estimated using the algorithm in [97]. Also, Figures 2.14 and

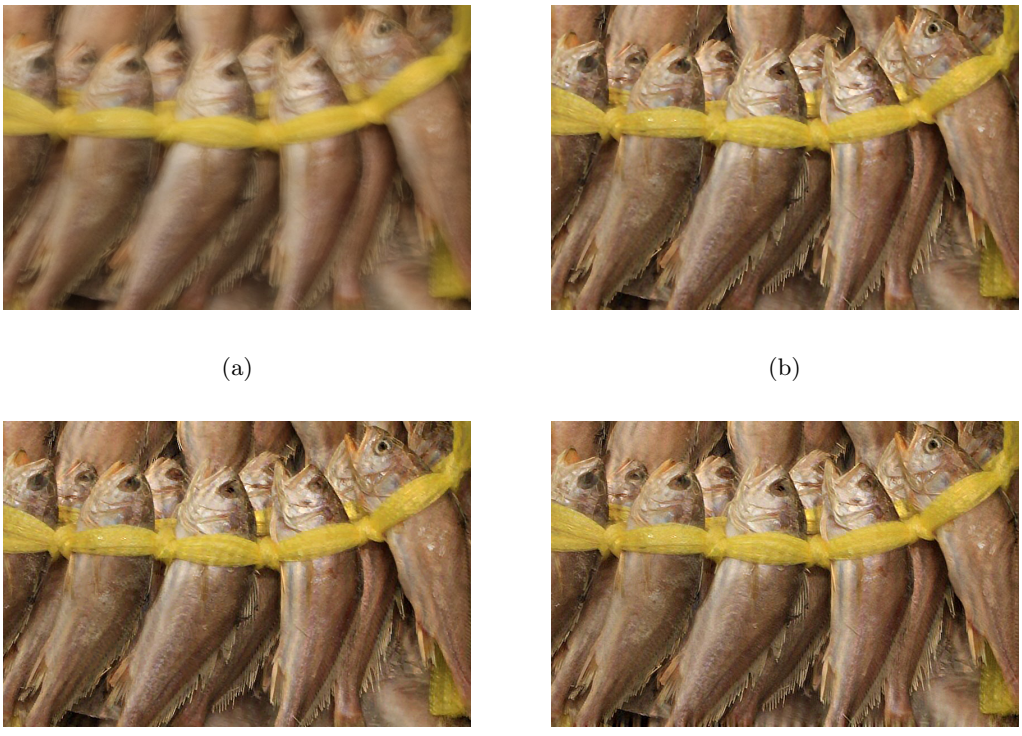


(a) (b)

(c) (d)

Fig. 2.13: Real motion deblurring example: (a)input blurred noisy image, (b)output of hyper-Laplacian algorithm [58], (c)output of [97], and (d)output of our algorithm ($\eta = 0.018, \beta = 0.9$).

2.15 illustrate the results of different algorithms when applied to real blurred images using corresponding estimated blur kernels from [15]. As can be seen, our iterative deblurring algorithm produces high quality outputs as good as the state-of-the art.



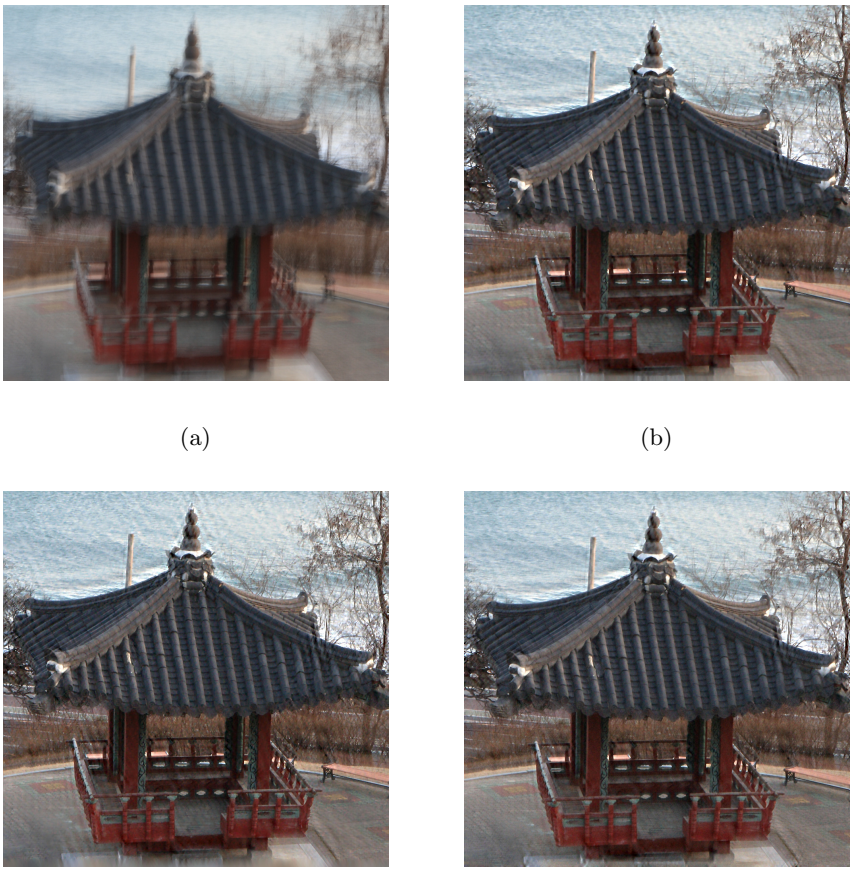
(a) (b)

(c) (d)

Fig. 2.14: Real motion deblurring example: (a)input blurred noisy image, (b)output of hyper-Laplacian algorithm [58], (c)output of [97], and (d)output of our algorithm ($\eta = 0.031, \beta = 0.6$).

2.7.4 Comparison With Traditional Normalized Graph Laplacian

In this section, we set up an experiment to investigate the performance of the proposed graph Laplacian with respect to that of the traditional normalized graph Laplacian. If we denote $W_D = D^{-1/2}KD^{-1/2}$, then the traditional normalized graph Laplacian would be $I - W_D$. We define the following cost function by replacing W with



(a) (b)

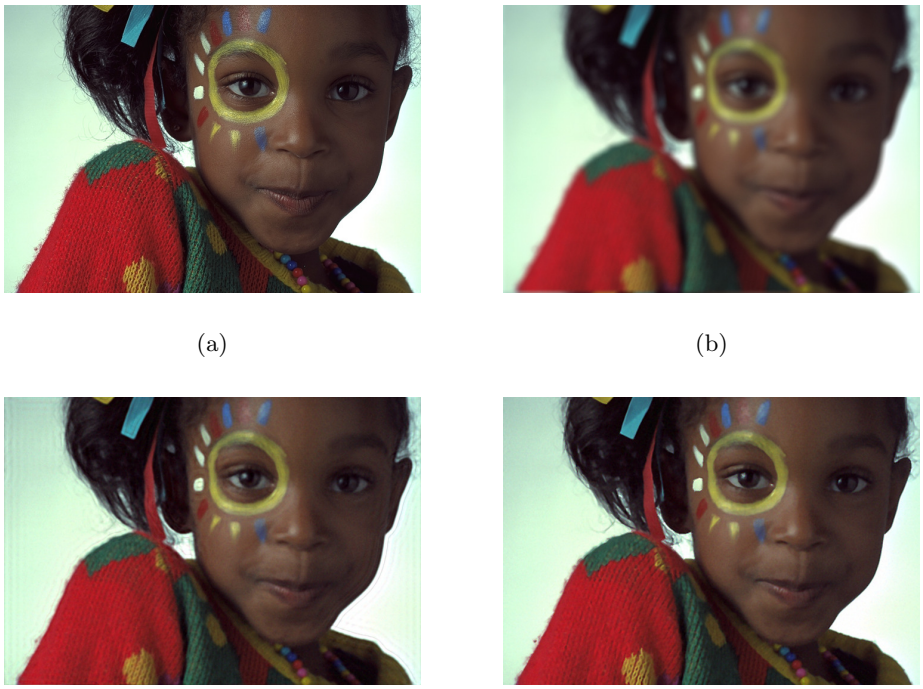
(c) (d)

Fig. 2.15: Real motion deblurring example: (a)input blurred noisy image, (b)output of hyper-Laplacian algorithm [58], (c)output of [97], and (d)output of our algorithm ($\eta = 0.032, \beta = 0.6$).

W_D in (2.15) as

$$E_D(\mathbf{z}) = (\mathbf{y} - A\mathbf{z})^T\{I + \beta(I - W_D)\}(\mathbf{y} - A\mathbf{z}) + \eta\ \mathbf{z}^T(I - W_D)\mathbf{z}. \tag{2.25}$$

The out-of-focus blurred Girl image in Figure 2.16(b) and the motion blurred Street image in Figure 2.17(b) are used for the experiments. The deblurring results from



(a) (b)

(c) (d)

Fig. 2.16: Deblurring examples with blurred noisy Girl image by out-of-focus blur with radius 7 and additive white Gaussian noise with standard deviation $\sigma = 1$: (a)clean image, (b)blurred noisy image, (c)output of the deblurring algorithm with the corresponding traditional normalized Laplacian (PSNR = 29.40dB, SSIM = 0.8734), and (d)output of our proposed deblurring algorithm (PSNR = 30.58dB, SSIM = 0.9058).

minimizing (2.25) related to traditional normalized graph Laplacian along with those of our proposed method are shown in Figures 2.16 and 2.17. As can be seen, our proposed normalized Laplacian produces deblurring outputs with higher quality within the same framework. Also, these experiments show that other definitions of the graph Laplacian can be used within our proposed framework and still produce acceptable results.

(a) (b)

(c) (d)

Fig. 2.17: Motion deblurring examples with blurred noisy Street image by synthetic motion blur kernel and additive white Gaussian noise with standard deviation $\sigma = 1$: (a)clean image, (b)blurred noisy image, (c)output of the deblurring algorithm with the corresponding traditional normalized Laplacian (PSNR = 27.93dB), and (d)output of our proposed deblurring algorithm (PSNR = 29.75dB).

Summary - In this chapter, we proposed a general framework for kernel similarity-based image restoration. We have introduced a new objective function for image deblurring by coupling the data and prior terms via structurally encoded filtering and Laplacian matrices. Also, we have presented a graph-based filtering interpretation of the proposed method, providing better intuition for data-adaptive approaches as well

as a path for further improvement of such approaches. Through experiments, the effectiveness of the kernel similarity-based method has been verified for a range of blurring scenarios via comparison with some of the existing state-of-the-art algorithms. Also, a special case within the proposed framework was highlighted for image sharpening. This kernel similarity-based approach is general enough to be exploited for many different restoration tasks, as long as there is a reasonable way to estimate the kernel similarity matrix K.

Chapter 3

Graph-based Denoising And Smoothing: Interpretation Based on Diffusion and Boosting Iterations

Abstract - As shown in the previous chapter, any image can be represented as a function defined on a discrete weighted graph whose vertices are image pixels. Each pixel can be linked to other pixels via graph edges with corresponding weights derived from similarities between image pixels (graph vertices) measured in some appropriate fashion. Image structure is encoded in the Laplacian matrix derived from these similarity weights. Taking advantage of this graph-based point of view, we present a general regularization framework for image denoising. A number of well-known existing denoising methods like bilateral, NLM, and LARK can be described within this formulation. Moreover, we present an analysis for the filtering behavior of the proposed method based

on the spectral properties of Laplacian matrix. Some of the well established iterative approaches for improving kernel-based denoising like diffusion and boosting iterations are special cases of our general framework. The proposed approach provides a better understanding of enhancement mechanisms in self similarity-based methods, which can be used for their further improvement. Experimental results verify the effectiveness of this approach for the task of image denoising.

3.1 Introduction

In the past decade, algorithms that exploit the existing self-similarity in images have shown promising denoising results. Denoising algorithms like bilateral, non-local means (NLM), and LARK [11,69,105,112] are among such kernel similarity-based methods. All the above mentioned algorithms perform denoising based on some sort of shrinkage operation in an adaptive basis. There are other approaches that attempt to use the non-local similarity idea in a variational formulation [29,69,81,85,99]. Inspired by these works, in this chapter, we present a general graph-based framework for image denoising. An objective function is formulated and iteratively optimized in which data fidelity and smoothness terms are coupled via Laplacian and similarity matrices of the underlying image. Two widely used iterative denoising methods are discussed as special cases of the proposed framework. Moreover, taking advantage of filtering and Laplacian matrices, we propose a graph-based analysis framework for multi-layer image decomposition.

3.2 Kernel Formulation for Denoising

We propose the following unified cost function for kernel-based image denoising:

$$E(\mathbf{z}) = (\mathbf{y} - \mathbf{z})^T F(K)(\mathbf{y} - \mathbf{z}) + \eta\, \mathbf{z}^T G(L)\, \mathbf{z}, \tag{3.1}$$

in which K is a data-dependent kernel similarity matrix whose (i, j)th element is the kernel similarity coefficient between pairs of pixels i and j. L is the corresponding Laplacian matrix computed from K, and η is a positive regularization parameter which balances the first term (data fidelity term) and the second term (smoothness term). Also, $F(.)$ and $G(.)$ are functions of K and L, to be specified shortly. While the core discussion is applicable to any valid choice of kernels [69], to keep focus here, we use the NLM kernel as a canonical example in the remainder of the chapter. Kernel weight coefficients are computed from a pre-filtered version of the observed image. In what follows, we discuss two instances of the above energy function.

3.2.1 Case 1: Un-normalized Laplacian

By defining $G(L) = L = D - K$ and $F(K) = K$, we have

$$E(\mathbf{z}) = (\mathbf{y} - \mathbf{z})^T K(\mathbf{y} - \mathbf{z}) + \eta\, \mathbf{z}^T (D - K)\, \mathbf{z}. \tag{3.2}$$

Note that, as discussed in the previous chapter, $D - K$ is the un-normalized Laplacian matrix widely used in graph theory [20, 114]. The first term is essentially a weighted data fidelity term and the second term is a difference term that adaptively penalizes large derivatives based on the structure of data encoded in the Laplacian matrix. The

gradient descent (GD) iterations can be used to minimize (3.2) with respect to $\mathbf{z}$:

$$\begin{aligned}\hat{\mathbf{z}}_k &= \hat{\mathbf{z}}_{k-1} - \mu \nabla E(\mathbf{z})_{|_{\mathbf{z}=\hat{\mathbf{z}}_{k-1}}} \\ &= \hat{\mathbf{z}}_{k-1} + \mu K(\mathbf{y} - \hat{\mathbf{z}}_{k-1}) - \mu\eta(D-K)\hat{\mathbf{z}}_{k-1} \end{aligned} \tag{3.3}$$

Here, μ is the step size for GD iterations. At convergence, the corresponding estimate would be

$$\hat{\mathbf{z}} = (K + \eta(D-K))^{-1} K\mathbf{y}. \tag{3.4}$$

Interestingly, with K as e.g., the NLM kernel, and for the specific choice of $\eta = 1$ (both terms contribute equally strongly), we have precisely the NLM denoising algorithm:

$$\hat{\mathbf{z}}_{NLM} = D^{-1}K\mathbf{y}. \tag{3.5}$$

Moreover, as shown in Section 3.5, with a fixed smoothing parameter h, tuning the regularization parameter η, yields an estimation with lower mean squared error (MSE) compared to standard NLM solution. We can use SURE-based MSE estimation approach for adjusting the regularization parameter η [91].

3.2.2 Case 2: Normalized Laplacian

The second approach is to apply Sinkhorn-Knopp matrix scaling algorithm [56,100] to the symmetric non-negative matrix K to construct the filtering matrix $W = C^{-1/2}KC^{-1/2}$, as shown in the previous chapter. The resulting matrix W is a symmetric non-negative doubly stochastic matrix. Hence, based on Perron-Frobenius theory, W has unity spectral radius with largest eigenvalue $\lambda_1 = 1$ whose corresponding eigenvector

is $\mathbf{v}_1 = (\frac{1}{\sqrt{n}})\mathbf{1}_n$. Moreover, it can be decomposed as $W = VSV^T$, in which V is an orthonormal matrix whose ith column $\mathbf{v}_i$ is the ith eigenvector of W. The corresponding ith eigenvalue λ_i is the ith diagonal element of S. Then, the normalized Laplacian matrix is defined as $I - W$. In this case, our objective function becomes

$$E(\mathbf{z}) = (\mathbf{y} - \mathbf{z})^T W(\mathbf{y} - \mathbf{z}) + \eta\, \mathbf{z}^T(I - W)\, \mathbf{z}, \tag{3.6}$$

in which, $F = W$ and $G = I - W$. Note that, for positive semi-definite matrix W, the above cost function can be rewritten as

$$E(\mathbf{z}) = \|W^{1/2}(\mathbf{y} - \mathbf{z})\|^2 + \eta\, \mathbf{z}^T(I - W)\, \mathbf{z}, \tag{3.7}$$

The corresponding data fidelity term is essentially derived by measuring the squared norm of the filtered version of the residual $\mathbf{y} - \mathbf{z}$ in the space spanned by data-adaptive orthonormal eigenvectors of the filtering matrix W. Also, note that the data-fidelity term in (3.7) can be thought of as the data-adaptive variant of the data term in the TV-Hilbert model in [4, 12]. Also, the second term is a difference term that data-adaptively penalizes high frequency components in the solution using our specific definition of the normalized Laplacian operator. By computing the gradient of (3.6) with respect to $\mathbf{z}$, one obtains an iterative GD update equation as

$$\hat{\mathbf{z}}_k = \hat{\mathbf{z}}_{k-1} + \mu W(\mathbf{y} - \hat{\mathbf{z}}_{k-1}) - \mu\eta(I - W)\hat{\mathbf{z}}_{k-1}. \tag{3.8}$$

By appropriate selection of the step size parameter[1] μ, (3.8) converges to

$$\hat{\mathbf{z}} = (W + \eta(I - W))^{-1} W\mathbf{y}. \tag{3.9}$$

[1]It can be shown that $\mu = 1$ is a sufficient condition for the convergence of the iterative algorithm in (3.8).

Again, for the case $\eta = 1$, optimization of (3.6) leads to $\hat{\mathbf{z}} = W\mathbf{y}$ which has been shown to outperform its non-symmetric counterpart $\hat{\mathbf{z}}_{NLM} = D^{-1}K\mathbf{y}$ [70]. Also, note that for the case $\eta = 2$, the equation (3.9) reduces to the same steady state solution as the SOS boosting algorithm in [92].

3.2.3 Spectral Analysis Based on Eigenvectors of Laplacian

Since the filtering matrix W and the normalized Laplacian matrix $I - W$ have the same eigenvectors, (3.9) can be interpreted as filtering the observed image in a space spanned by the eigenvectors of the Laplacian. Note that (3.9) can be expressed as

$$\hat{\mathbf{z}} = W'\mathbf{y} = VS'V^T\mathbf{y}, \tag{3.10}$$

where S' is a diagonal matrix whose ith diagonal element λ'_i is a function of the corresponding eigenvalue λ_i of W as

$$\lambda'_i = p(\lambda_i) = \frac{\lambda_i}{(1-\eta)\lambda_i + \eta}, \tag{3.11}$$

Optimizing the value of η in (3.11) or (3.8) with respect to an appropriate measure (e.g., using SURE [91]) gives the final estimate. As an illustration, an edge patch of size 31×31 is considered in Fig. 3.1(a) and white Gaussian noise of standard deviation 25 is added to it (Fig. 3.1(b)). The resulting output of iterative algorithm (3.8) is shown in Fig. 3.1(c). Figure 3.1(d) shows the spectrum of W (λ_i) and the corresponding spectrum of W' (λ'_i). As will be shown in experiments, normalized Laplacian formulation results in denoising outputs with slightly better visual quality. Moreover,

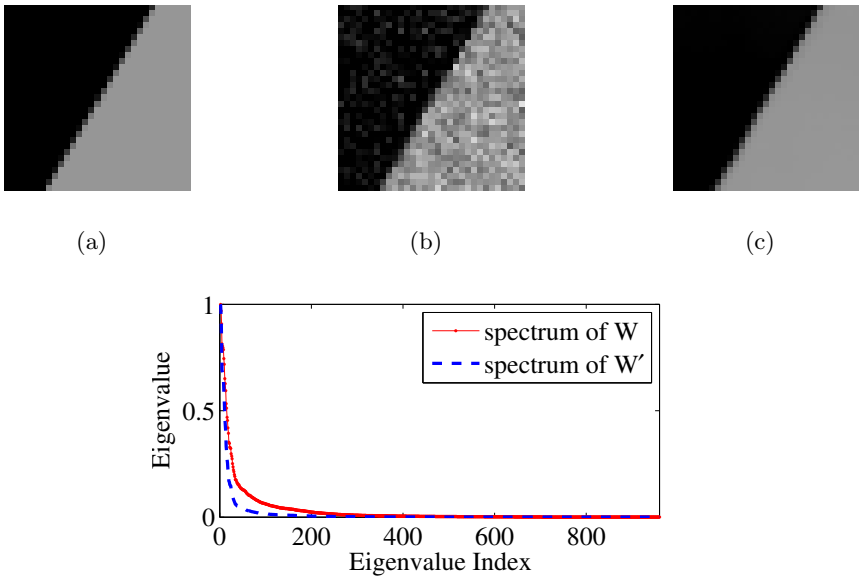


Fig. 3.1: Denoising experiment on 31×31 piece-wise constant synthetic patch (a)clean patch, (b)noisy patch by adding white Gaussian noise with $\sigma = 25$ (PSNR = 20.31dB) , (c)output of iterative algorithm (3.8) (PSNR= 35.96dB, $\eta = 19$), and (d)spectrum of filter matrices W (λ_i's) and W' (λ_i''s).

filtering analysis of the normalized case is straightforward in the space spanned by V.

3.3 Diffusion and Boosting As Special Cases of the Proposed Algorithm

Two widely used iterative methods, namely diffusion and boosting, have been effectively used for improving the performance of kernel-based denoising algorithms [69]. Two extreme cases of our more general iterative approach (Eq. 3.8) lead to diffusion

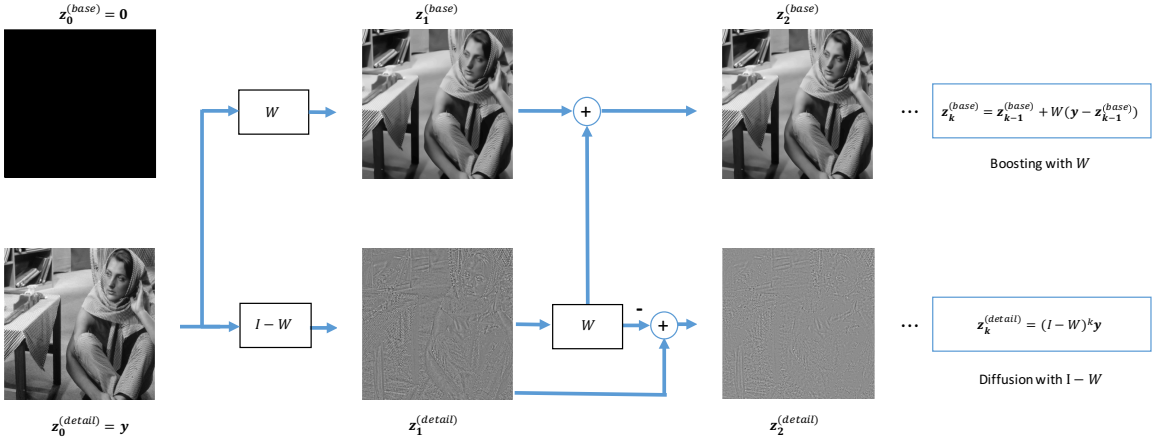


Fig. 3.2: Block diagram of the proposed graph-based approach for multi-layer image decomposition.

and boosting iterations, as we illustrate below.

By setting $\eta = \infty$ in (3.6), the corresponding GD update equation becomes diffusion iterations as

$$\hat{\mathbf{z}}_k = W\hat{\mathbf{z}}_{k-1}. \tag{3.12}$$

Initializing (3.12) with $\hat{\mathbf{z}}_0 = \mathbf{y}$ leads to $\hat{\mathbf{z}}_\infty = \frac{1}{n}\mathbf{1}_n\mathbf{y}^T\mathbf{1}_n$, which corresponds to a final constant estimate (an estimation without variance) [69].

On the other hand, by choosing η to be equal to 0, the effect of the smoothness term in (3.6) is omitted, for which, boosting iterations is derived as

$$\hat{\mathbf{z}}_k = \hat{\mathbf{z}}_{k-1} + W(\mathbf{y} - \hat{\mathbf{z}}_{k-1}). \tag{3.13}$$

In this case, initializing the boosting algorithm in (3.13) with $\hat{\mathbf{z}}_0 = \mathbf{0}$, results in $\hat{\mathbf{z}}_\infty = \mathbf{y}$. This corresponds to an un-biased estimate of the original image with variance equal to

(a) Lena (b) Barbara (c) Cameraman (d) House

Fig. 3.3: Set of images used for evaluating the performance of our denoising algorithm with respect to standard NLM.

(a) (b) (c) (d) (e)

Fig. 3.4: Denoising experiment on 256×256 House image, (a)clean image, (b)noisy image ($\sigma = 20$), (c)standard NLM output image (PSNR = 30.80dB), (d)output of iterative algorithm (3.3) (PSNR= 32.16dB, $\eta = 0.67$), and (e)output of iterative algorithm (3.8) (PSNR= 32.37dB, $\eta = 0.82$).

the noise variance [69].

TABLE 3.1: Denoising PSNR and SSIM performance of the iterative algorithms (3.3) and (3.8).

Images	NLM	Un-normalized		Normalized	
	PSNR (SSIM)	PSNR (SSIM)	η	PSNR (SSIM)	η
Lena	28.68 (0.761)	29.66 (0.807)	0.53	29.73 (0.814)	0.58
Barbara	28.02 (0.794)	29.36 (0.855)	0.53	29.35 (0.860)	0.58
Cameraman	29.18 (0.781)	29.69 (0.833)	0.53	29.48 (0.843)	0.82
House	30.80 (0.768)	32.16 (0.845)	0.67	32.37 (0.849)	0.82

3.4 Graph-based multi-layer image decomposition and its interpretation based on diffusion and boosting iterations

Our proposed graph-based framework provides a general formulation for different kernel similarity criteria and the corresponding edge preserving filters. These edge-aware filters are widely used for multi-layer image decomposition, smoothing, and detail manipulation [32,50,103,108]. In this section, we provide a unified framework for such multi-layer decomposition methods. We will also show that the base and detail layers in different steps of the decomposition process can be derived and described via diffusion and boosting iterations. More specifically, in the first step, the base layer is derived by applying the smoothing filter W to the input image $\mathbf{y}$ as $W\mathbf{y}$ and the detail layer is computed by subtracting the base layer from the input image as $(I - W)\mathbf{y}$.

Similarly, in the second step, smoothing filter W is again applied to the detail layer from the previous step to get $W(I-W)\mathbf{y} = (W-W^2)\mathbf{y}$, which in turn is added to the base layer in the first step to get the base layer in the second step as

$$(W-W^2)\mathbf{y} + W\mathbf{y} = (2W-W^2)\mathbf{y}. \tag{3.14}$$

Also, the detail layer in the second step of the decomposition would be simply

$$(I-W)\mathbf{y} - (W-W^2)\mathbf{y} = (I-2W+W^2)\mathbf{y} = (I-W)^2\mathbf{y}. \tag{3.15}$$

Following a similar approach, the base layer at the k-th stage of the decomposition can be expressed as boosting with the smoothing filter W, when initialized with $\mathbf{z}_0^{(base)} = \mathbf{0}$ as

$$\mathbf{z}_k^{(base)} = \mathbf{z}_{k-1}^{(base)} + W(\mathbf{y} - \mathbf{z}_{k-1}^{(base)}). \tag{3.16}$$

On the other hand, if we start from $\mathbf{z}_0^{(detail)} = \mathbf{y}$, the detail layer at the k-th stage of the decomposition can be derived as diffusion iterations with the normalized graph Laplacian $I-W$ after k steps

$$\mathbf{z}_k^{(detail)} = (I-W)\mathbf{z}_{k-1}^{(detail)} = (I-W)^k\mathbf{y}. \tag{3.17}$$

This interpretation is intuitive in the sense that the properties of base and detail layers can be described via the low-pass and high-pass characteristics of the smoothing and Laplacian matrices, respectively. Figure 3.2 illustrates the underlying idea.

(a) (b) (c) (d)

Fig. 3.5: Denoising experiment on 256×256 color parrot image, (a)original image, (b)noisy image ($\sigma = 20$, PSNR= 17.06dB), (c)standard NLM output image (PSNR= 27.68dB), and (d)output of iterative algorithm (3.8) (PSNR= 28.54dB).

3.5 Experimental Results

In this section, we compare the performance of the proposed denoising algorithm with that of the standard NLM [11] and global image denoising [107].

3.5.1 Comparisons With NLM Denoising Algorithm

In order to evaluate the effectiveness of the proposed kernel-based denoising approach, we apply this algorithm for restoration of 256×256 benchmark images. For denoising experiments, Gaussian noise with standard deviation 20 is added to images shown in Fig. 3.3 and performance of iterative algorithms (3.3) and (3.8) are compared against the output of standard NLM. Peak signal to noise ratio (PSNR) in dB, and SSIM [116] are used as quantitative measures for comparison. As can be seen in Table 3.1, in all cases we get better results in terms of PSNR and SSIM with respect to standard

(a) House (b) Cameraman (c) Peppers (d) Aerial

Fig. 3.6: Set of images used for evaluating the performance of our denoising algorithm with respect to global denoising algorithm.

TABLE 3.2: Denoising performance and run-time in seconds of the iterative algorithm (3.8) and global denoising [107].

Images	G-NLM [107]		Proposed	
	PSNR (SSIM)	run-time	PSNR (SSIM)	run-time
House	32.67 (0.856)	92	32.37 (0.849)	60
Cameraman	29.53 (0.855)	98	29.48 (0.843)	61
Peppers	30.26 (0.865)	93	30.41 (0.867)	56
Aerial	26.67 (0.836)	93	26.89 (0.853)	60

NLM. Also, note that SSIM values in Table 3.1 reflect slightly better visual quality of the results of normalized iterative algorithm (3.8) compared to un-normalized algorithm (3.3). Figure 3.4 illustrates House image denoised using the general iterative kernel-based approaches (3.3) and (3.8) compared to standard NLM denoising. Additionally, the result of applying normalized iterative algorithm (3.8) to a noisy color image with the same experimental settings as for the previous examples is shown in Fig. 3.5.

(a) (b) (c) (d)

Fig. 3.7: Denoising experiment on 256×256 Aerial image, (a)original image, (b)noisy image ($\sigma = 20$, PSNR= 22.08dB), (c)output of global denoising algorithm in [107] (PSNR= 26.67dB), and (d)output of iterative algorithm (3.8) (PSNR= 26.89dB).

Experimental results show that our algorithm produces superior results to those of standard NLM denoising algorithm.

3.5.2 Comparison With Global Denoising Algorithm

The idea of global image denoising using similarity information among *all* pixels in the image has been recently introduced in [107]. In this part, we compare the results of the denoising algorithm in (3.8) with that of the global image denoising [107]. In this case, Gaussian noise with standard deviation 20 is added to images shown in Fig. 3.6 and performance of iterative algorithm (3.8) is compared against the output of the denoising algorithm in [107]. Simulation results indicate that our proposed method is able to produce comparable outputs with less running time when run on the same system. Figure 3.7 shows an example comparing the results from these two algorithms. Also, in Table 3.2, the performance and run-time of the proposed denoising algorithm

is compared with those of [107].

Summary - We have developed a new graph-based framework for image denoising and smoothing in this chapter. This general approach encompasses some well-known existing denoising methods, and provides a path for further improvements. Also, we have presented a general kernel similarity-based analysis formulation for multi-layer image decomposition and provided its interpretation based on diffusion and boosting iterations with data-adaptive operators.

Chapter 4

Data-adaptive Image Sharpening Based on The Difference of Smoothing Operators

Abstract - In this chapter, we propose an effective data-adaptive filtering mechanism for sharpening of noisy and moderately blurred images. We establish the connection of our proposed data-adaptive filtering procedure with the classic difference of Gaussians (DoG) operator widely used in image processing and computer graphics. Our proposed filter renders a data-adaptive and noise robust version of the classical DoG filter. We also discuss interesting special cases of our general sharpening method. Experimental results verify the effectiveness of the proposed technique for sharpening real images.

4.1 Introduction

Any effective image enhancement algorithm should be equipped with appropriate mechanisms for dealing with visual degradations resulting from noise and blur. Specifically, image sharpening is a challenging task when the input image is noisy, even when the image is not severely blurred. The reason is that image sharpening is essentially a high pass filtering operation aiming at amplifying high frequency details in the input image. On the other hand, noise components in the input image exhibit high frequency attributes and any attempt to magnify fine details in the input image will naturally result in amplifying noise as well. For instance, widely used classical linear unsharp mask filters are very sensitive to the noise in the input image [87]. Another source of visual degradation is the effect of over-sharpening or the so-called overshoot/undershoot effects [5]. It occurs when the sharpening algorithm tries to sharpen parts of the image which are already in focus, especially edges, which causes unpleasant artifacts in such high contrast areas of the image.

In this chapter, we propose a new sharpening filter based on the data-adaptive smoothing and Laplacian matrices for noisy and mildly blurred images. This approach is general enough to include any valid construction of smoothing and Laplacian matrices with appropriate definition of the similarity measure [69]. We establish the connection between our proposed sharpness enhancement filter and the classical DoG filters widely used in different image manipulation tasks in the literature, which provides a better understanding of the mechanism and the functionality of the underlying parameters in

our proposed data-adaptive filter. Ours can be considered as a data derived and noise robust variant of the DoG filter. As such, it can be used in various tasks in which DoG operators are exploited [66,121,122]. We also discuss interesting simpler special cases of our proposed filter. Experimental results demonstrate the effectiveness of our approach in dealing with real examples. Namely, the underlying structure-aware mechanism of our proposed filter enables it to enhance the sharpness in the input image while reducing noise amplification and other artifacts like halo and false color artifacts.

4.2 Problem Formulation and Proposed Sharpening Filter

In this section, we describe the underlying model and review the idea of classical sharpening filters based on the notion of difference of Gaussians operators. Then, we elaborate on our structure-aware sharpening filter based on *data-adaptive* smoothing and Laplacian matrices.

4.2.1 Underlying model

As shown before, the degradation process in image formation is usually mathematically modeled as

$$\mathbf{y} = A\mathbf{z} + \mathbf{n}. \tag{4.1}$$

In this chapter, we deal with input images which are noisy but not severely blurred. Without deblurring, the goal is to find an appropriate operator F such that $\hat{\mathbf{z}} = F\mathbf{y}$ is a sharpness enhanced version of the input $\mathbf{y}$ without amplifying noise and other edge

related artifacts; and without explicitly inverting the blur operator. In fact, we do not assume knowledge of A.

4.2.2 Classical DoG Operator

The idea of classical DoG filter is that the difference of two Gaussian kernels is able to produce a range of different kernels with various desired frequency responses. In [66], the standard DoG operator is introduced as an approximation to Laplacian of Gaussian operator for edge detection[1]

$$DoG_{\varepsilon,k}(x,y) = G_{\varepsilon}(x,y) - G_{k\varepsilon}(x,y), \tag{4.2}$$

where ε is the standard deviation of the Gaussian function defined as $G_{\varepsilon}(x,y) = \frac{1}{2\pi\varepsilon^2}\exp\{-\frac{x^2+y^2}{2\varepsilon^2}\}$. $k > 1$ is a positive factor. Also, x and y are spatial coordinates. An extension of the standard DoG filter in [66] is defined in [122] as

$$DoG_{\varepsilon,k,\tau}(x,y) = G_{\varepsilon}(x,y) - \tau G_{k\varepsilon}(x,y), \tag{4.3}$$

in which the parameter $\tau \in (0,1)$ determines the sensitivity of the edge detector [122]. For instance, for small values of τ, the DoG operator in (4.3) is less sensitive to noise at the expense of losing some edges in the input image. Eq. (4.3) can be rewritten in terms of the standard DoG operator in (4.2) as

$$DoG_{\varepsilon,k,\tau}(x,y) = (1-\tau)G_{\varepsilon}(x,y) + \tau DoG_{\varepsilon,k}(x,y). \tag{4.4}$$

Note that the Gaussian operator preserves the average intensity of the input image while the average response of the standard DoG operator $DoG_{\varepsilon,k}(x,y)$ is zero. Therefore,

[1]The relationship between DoG operator and the Laplacian of Gaussian has been discussed in the Appendix 4.A.

changing the parameter τ in (4.4) for achieving the desired effects, inadvertently alters the average brightness of the input image[2]. Using a simple reparameterization , the authors in [121] describe the following family of sharpening filters

$$S_{\varepsilon,k}(x,y) = \frac{DoG_{\varepsilon,k,\tau}(x,y)}{1-\tau} = (1+\beta)G_{\varepsilon}(x,y) - \beta G_{k\varepsilon}(x,y), \tag{4.5}$$

in which $\beta = \frac{\tau}{1-\tau}$. Compared to the formulation in (4.3), while the average brightness in the input image is retained, the level of sharpening can also be controlled by the parameter β in (4.5). This average intensity-preserving sharpening operator has been derived mostly through a heuristic approach. It is intrinsically linear and not data-dependent. In the following subsections, we start from a filter design framework based on the spectral properties of the data-driven Laplacian operators. Then, in Section 4.3, we elaborate on its relationship with classical DoG-based sharpening operators described here in this section. This helps explain the properties of the DoG-based operators from a filtering point of view. It also provides a powerful structure-aware mechanism for dealing with more complicated real-world images where there is a need to incorporate nonlinear filters for better performance.

4.2.3 Structure-aware Sharpening Filter

Since any sharpening operator is inherently a highpass filter, it inadvertently leads to amplifying high frequency noise components in the input image and causes artifacts related to over-sharpening high contrast regions in the input image. Therefore,

[2]Preserving the average values of the input image is a desired property for filtering tasks in image processing [69].

there is a need to incorporate an effective smoothing mechanism in the sharpening operator to alleviate these shortcomings while preserving image structures. Exploiting the existing self-similarity in natural images tends to be an effective way to take into account the underlying structure of images when constructing such operators [11,69], and thereby reducing the related artifacts. In the following, we use data-adaptive smoothing and Laplacian matrices introduced in previous chapters as the main building blocks for creating the final sharpening filter.

As discussed previously, the smoothing filter W is a symmetric and doubly stochastic matrix. Since W is stochastic, its largest eigenvalue would be $\lambda_1 = 1$ corresponding to the constant eigenvector which highlights its lowpass filtering properties [51]. It also preserves the average brightness when applied to an image [69]. On the other hand, the Laplacian matrix is defined as $I - W$ with zero eigenvalue corresponding to the constant eigenvector. As such, it returns zero vector when applied to a constant signal. Therefore, it can be considered as a data-adaptive high pass operator. Note that these matrices are evaluated based on the similarity information among different parts of an image and encode the underlying structures of images in an effective manner. Note that we are able to define the data-adaptive unsharp mask filter using this definition of Laplacian as $I + \beta(I - W)$ which essentially adds a weighted highpass filtered version of the input image to itself to highlight the high frequency details. The problem with direct application of this filter is that it suffers from noise amplification and edge artifacts. We discuss how to improve its performance next.

4.2.3.1 Proposed Data-adaptive Sharpening Filter

For any effective sharpness enhancement operator, two requirements need to be satisfied: first, noise amplification should be avoided. Second, the level of contrast in the image should be increased without introducing overshoot and gradient reversal (halo) artifacts along edges. To this end, and inspired by DoG filtering idea, we propose a three stage filtering approach as shown in Fig. 4.1. We first apply a nonlinear smoothing operator to the input noisy image. This filter is aimed to data adaptively reduce the effect of the noise in the input image while avoiding over-smoothing. The smoothing operation is followed by a data derived unsharp masking operation controlled by the parameter $\beta > 0$. Finally, the smoothing filter is applied again in order to further control the effect of amplified noise and overshoot artifacts due to the unsharp mask filter. More formally, having the lowpass smoothing and highpass Laplacian operators at our disposal, we propose the following data-adaptive sharpening filter

$$F = W_1\{I + \beta(I - W_2)\}W_1, \tag{4.6}$$

where W_1 and W_2 are constructed from similarity matrices K_1 and K_2 with smoothing parameters h_1 and h_2, respectively. This provides us with the flexibility to better control the smoothing and sharpening operations. The parameters of the filter need to be tuned for the desired effects. Selection of these parameters will be discussed later. Furthermore, as will be shown in subsequent sections, our proposed filter based on the spectral properties of smoothing and Laplacian matrices exhibits nice connections to the classical DoG operator. Algorithm 2 summarizes different steps of the proposed

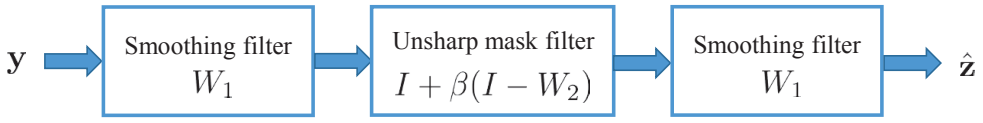


Fig. 4.1: Block diagram of the proposed enhancement algorithm. $\mathbf{y}$ is the input image and $\hat{\mathbf{z}}$ is the enhanced output image.

sharpening algorithm.

4.3 Interpretation of The Proposed Filter As Data-adaptive Difference of Smoothing Operator

In this section, we demonstrate the relationship between our proposed noise and edge aware sharpening filter and classical difference of Gaussians filter. This analysis sheds light on the properties of the proposed filter and gives a better understanding of the underlying parameters. Specifically, we provide a data-adaptive noise and edge aware version of the classical DoG-based operators for use in different applications. Note that the proposed filtering matrix F in (4.6) can be rewritten as

$$F = (1 + \beta)W_1^2 - \beta W_1 W_2 W_1. \tag{4.7}$$

If we define $G(W_1) = W_1^2$ and $H(W_1, W_2) = W_1 W_2 W_1$, then G can be considered as double application (diffusion operator) of the filter generated from data-adaptive Gaussian kernel in (2.10). On the other hand, H can be thought of as a cascade of data dependent Gaussian filters with a larger bandwidth compared to that of G. The difference in bandwidth of the filters G and H is determined by the scaling parameters

Algorithm 2 sharpness enhancement algorithm

Input: noisy blurry image $\mathbf{y}$

Output: sharpness enhanced image $\hat{\mathbf{z}}$

- Convert the RGB color image $\mathbf{y}$ to YCbCr color space.

- Compute similarity matrices K_1 and K_2 from the Y channel.

- Apply matrix scaling algorithm in [56] to K_1 and K_2 to get the diagonal matrices $C_1^{-1/2}$ and $C_2^{-1/2}$, and compute the filtering matrices $W_1 = C_1^{-1/2} K_1 C_1^{-1/2}$ and $W_2 = C_2^{-1/2} K_2 C_2^{-1/2}$.

- Construct the sharpening filter $F = (1+\beta) W_1^2 - \beta W_1 W_2 W_1$.

- Compute luminance channel estimate $\hat{Y}$ by applying F to Y.

- Compute chroma channel estimates $\hat{C}_b$ and $\hat{C}_r$ by applying F to C_b and C_r with smaller value of β.

- Convert the resulting estimated luminance and chroma channels back into the RGB color space to obtain the final sharpened image $\hat{\mathbf{z}}$.

h_1 and h_2 in the NLM kernel used for defining the smoothing filters W_1 and W_2, respectively. In fact, our proposed sharpening filter in (4.7) can be viewed as the data derived version of the DoG-based filter in (4.5). Here, we use the NLM kernel definition for producing matrices W_1 and W_2. However, any other edge aware kernel definition in the literature [11, 43, 105, 112, 125] can also be exploited within the proposed filtering scheme.

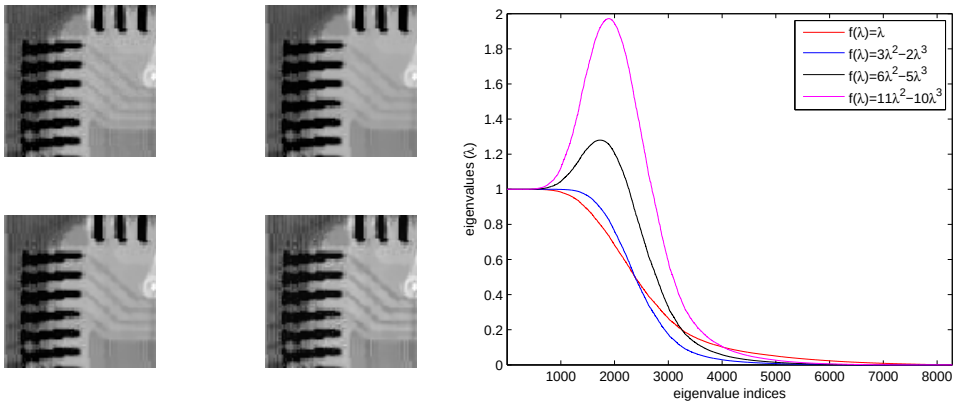
4.3.1 Special Cases of The Proposed Sharpening Framework

There is a nice connection with other image filter design paradigms when we consider the simple case where $W_1 = W_2 = W$. In this case, the filter F in (4.7) takes the following form

$$F = (1 + \beta)W^2 - \beta W^3. \tag{4.8}$$

Specifically, with $\beta = 2$, (4.8) boils down to $F = 3W^2 - 2W^3$. This filter is the data-adaptive version of the classical sharpened (linear) smoothing filer of Kaiser and Hamming in [49]. This polynomial function of the symmetric smoothing filter $W = VSV^T$ provides a way for improving the spectral properties of the initial filter (W) by manipulating its spectrum λ with a polynomial[3] function $f(\lambda) = 3\lambda^2 - 2\lambda^3$. This polynomial function is also well-known as the *smoothstep* function broadly used in computer graphics [7]. Also, note that for different values of the parameter β in (4.8) different band pass filters are generated. As an illustration, if we start from the spectrum of the low-

[3] λ's are the eigenvalues of W as the diagonal elements of the diagonal matrix S. Also, the columns of the orthonormal matrix V are the eigenvectors of W which serve as a data-adaptive basis for filtering purposes [108].



(a) (b)

Fig. 4.2: (a)top left: original 91×91 image, (a)top right: output with $f(\lambda) = 3\lambda^2 - 2\lambda^3$, (a)bottom left: output with $f(\lambda) = 6\lambda^2 - 5\lambda^3$, (a)bottom left: output with $f(\lambda) = 11\lambda^2 - 10\lambda^3$, and (b)the eigenvalues of the smoothing matrix W constructed from (a)top left along with the eigenvalues corresponding to sharpened smoothing filter $3W^2 - 2W^3$ and band pass filters $6W^2 - 5W^3$ and $11W^2 - 10W^3$.

pass symmetric filter $W = VSV^T$ constructed from a 91×91 image in Fig. 4.2(a), the spectral modifications due to the polynomial function $f(\lambda) = (1+\beta)\lambda^2 - \beta\lambda^3$ are shown in Fig. 4.2(b) for different values of the parameter β.

4.4 Reducing Color Artifacts

Any sharpness enhancement operator tends to increase the contrast and enhance the details in the input image. Besides high frequency noise components present in the input image, there are other high contrast artifacts in color images usually called

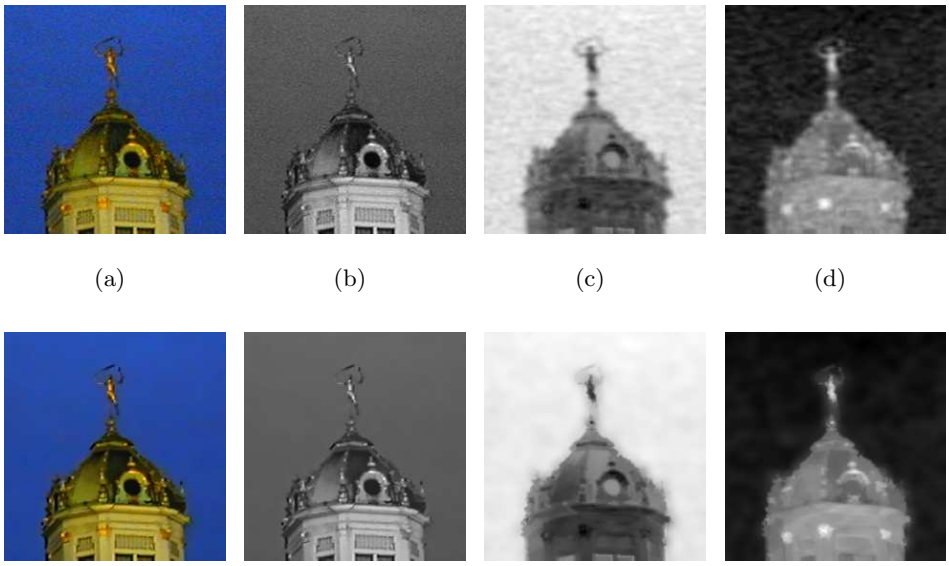


Fig. 4.3: First row: (a)input JPEG image and its different channels: (b)luminance Y channel, (c)chrominance Cb channel, and (d)chrominance Cr channel. Second row: (e)output of our sharpening filter and its different enhanced channels: (f)enhanced luminance Y channel, (g)enhanced chrominance Cb channel, and (h)enhanced chrominance Cr channel.

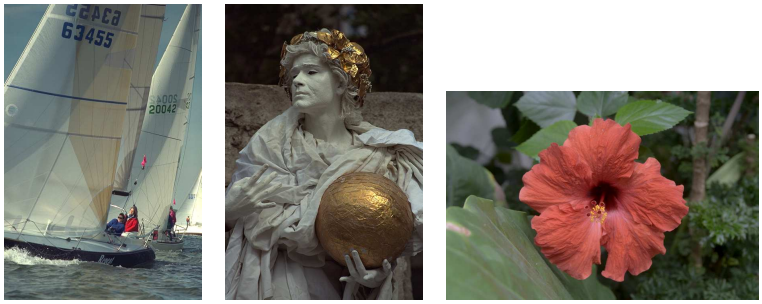
false color artifacts [13, 79]. These sorts of color artifacts are created during the image formation process and are amplified when using a compression scheme like JPEG. Due to their high frequency, and structured nature, they might get amplified by the sharpening operation. This phenomenon causes unpleasant color artifacts in the final image which degrades the visual quality. There exist some recent works trying to alleviate such distortions mostly as post processing algorithms [13, 18, 88]. We do not delve into the details of these methods. Instead, we opt for a simple mechanism within our filtering

framework to avoid amplification of color artifacts. It is well understood that human visual system is less sensitive to fine changes in chroma channels than those details in luminance channel [13]. In order to reduce these artifacts in the final output, we use a similar strategy to the one in [128]. Strictly speaking, we convert the input RGB color image to a suitable luminance-chrominance color space; i.e., YCbCr. Then, the kernel similarity coefficients computed for the luminance channel Y can also be used for filtering the chroma channels. This prevents the false color artifacts in color channels from contributing to the filtering coefficients and is effective in reducing such distortions in the final output. However, a less aggressive sharpening is applied to color channels by choosing a smaller value of the sharpening parameter β for chroma channels. This will further avoid amplification of false color artifacts in the sharpened image. Figure 4.3 illustrates the input JPEG image along with its different channels in YCbCr color space and their enhanced versions using our proposed algorithm[4]. As can be seen, our proposed sharpening filter is able to reduce the color artifacts while avoiding noise amplification in the final output.

4.5 Experimental Results

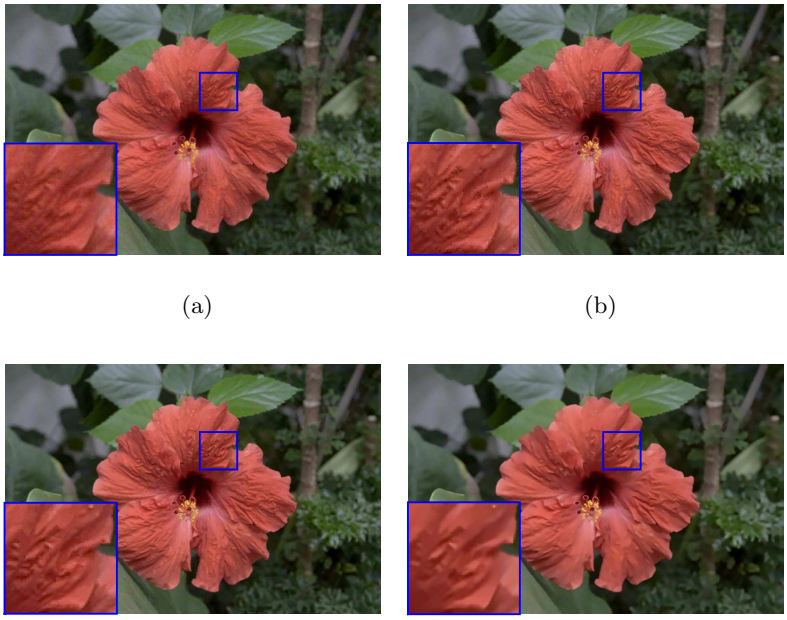
In this section, we demonstrate the performance of the proposed sharpening filter via a number of examples. Also, using synthetic examples the effects of different parameters are investigated. The performance of the proposed method is compared with those of [5, 43, 128] for real images.

[4]The input image is from the test examples in http://www.neatimage.com/examples.html.



(a) (b) (c)

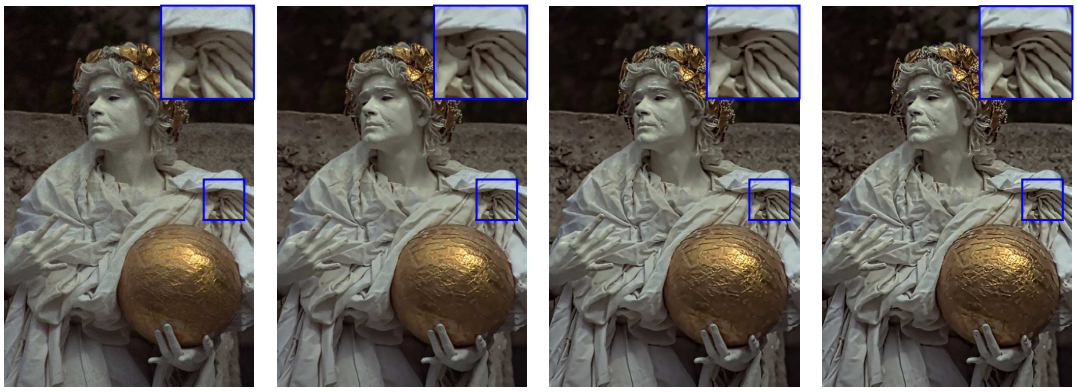
Fig. 4.4: Set of color images used in synthetic experiments.



(a) (b)

(c) (d)

Fig. 4.5: Effect of the parameter h_1 via synthetic examples with $\beta = 1.5, k = 3$: (a)input noisy and blurry image, (b)output of our sharpening algorithm with $h_1 = 0.8$, (c)output image with $h_1 = 1.2$, and (d)output image with $h_1 = 2$.



(a) (b) (c) (d)

Fig. 4.6: Effect of the parameter β via synthetic examples with $h_1 = 1.4, k = 3$: (a)input noisy and blurry image, (b)output of our sharpening algorithm with $\beta = 1.5$, (c)output image with $\beta = 2.5$, and (d)output image with $\beta = 3.5$.

4.5.1 Investigating The Effect of Different Parameters via Synthetic Examples

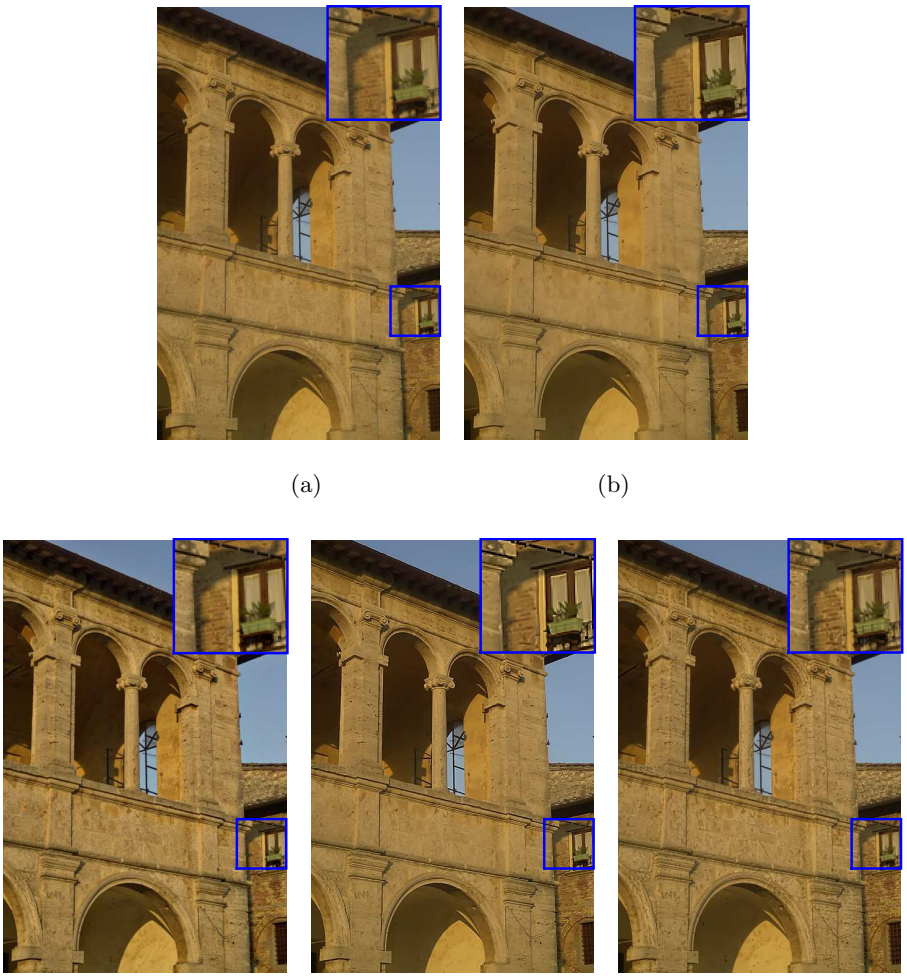
In this subsection, we use the test images in Fig. 4.4 to show the effect of different parameters. Slight out-of-focus blur and additive white Gaussian noise are added to test images in Fig. 4.4. For this purpose, each image is convolved with a 3×3 disk function. Then, additive white Gaussian noise with standard deviation equal to 5 is added to generate synthetic noisy and blurry examples.

The smoothing parameters h_1 and h_2 in the NLM kernel play an important role in the lowpass characteristics of the corresponding filters W_1 and W_2, respectively. These parameters control the level of sharpening and smoothing achieved by the proposed DoG-based filter in (4.7). We adopt $h_2 = kh_1$ with a positive factor $k > 1$ which

(a) (b) (c) (d)

Fig. 4.7: Effect of the parameter k via synthetic examples with $h_1 = 1.4, \beta = 1.5$: (a)input noisy and blurry image, (b)output of our sharpening algorithm with $k = 2$, (c)output image with $k = 3.5$, and (d)output image with $k = 6$.

controls the level of contrast enhancement in the output image. Also, note that the larger the parameter h_1 the smoother is the resultant image. In fact, we control the level of noise reduction by the parameter h_1. It also helps avoid the unpleasing halo artifacts in the output image. On the other hand, the parameter β controls the amount of sharpening in the image as it appears in the unsharp mask part of the filter F in (4.6). In Fig. 4.5, we fix the parameters k and β to be equal to 3 and 1.5, respectively. Then, we change the value of the scaling parameter h_1. As can be seen, by increasing this parameter the level of smoothing is increased resulting in more noise reduction. On the other hand, increasing the parameter β increases the detail enhancement level of the filter, as depicted in Fig. 4.6. The parameter k provides the algorithm with finer control over the degree of contrast enhancement, as illustrated in Fig. 4.7 with fixed



(a) (b)

(c) (d) (e)

Fig. 4.8: (a)input image, (b)output of [5], (c)output of [43], (d)output of [128], and (e)output of our sharpening algorithm with $h_1 = 0.7, k = 4, \beta = 1.6$.

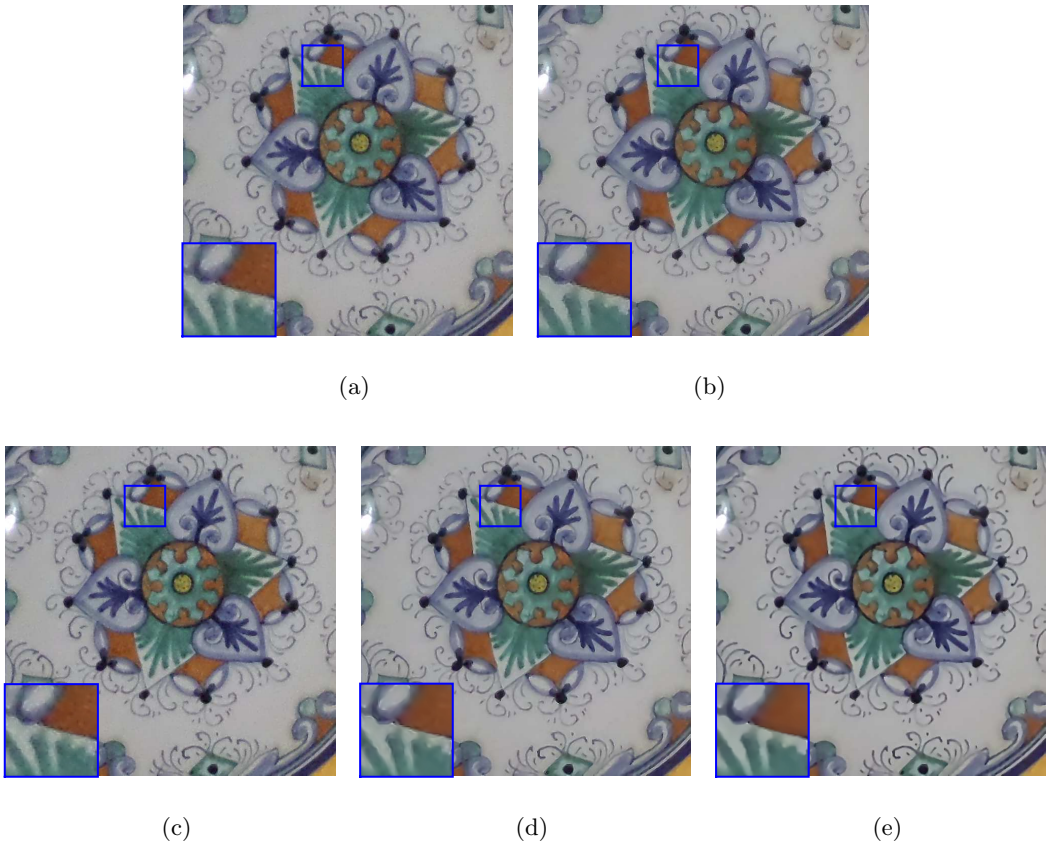
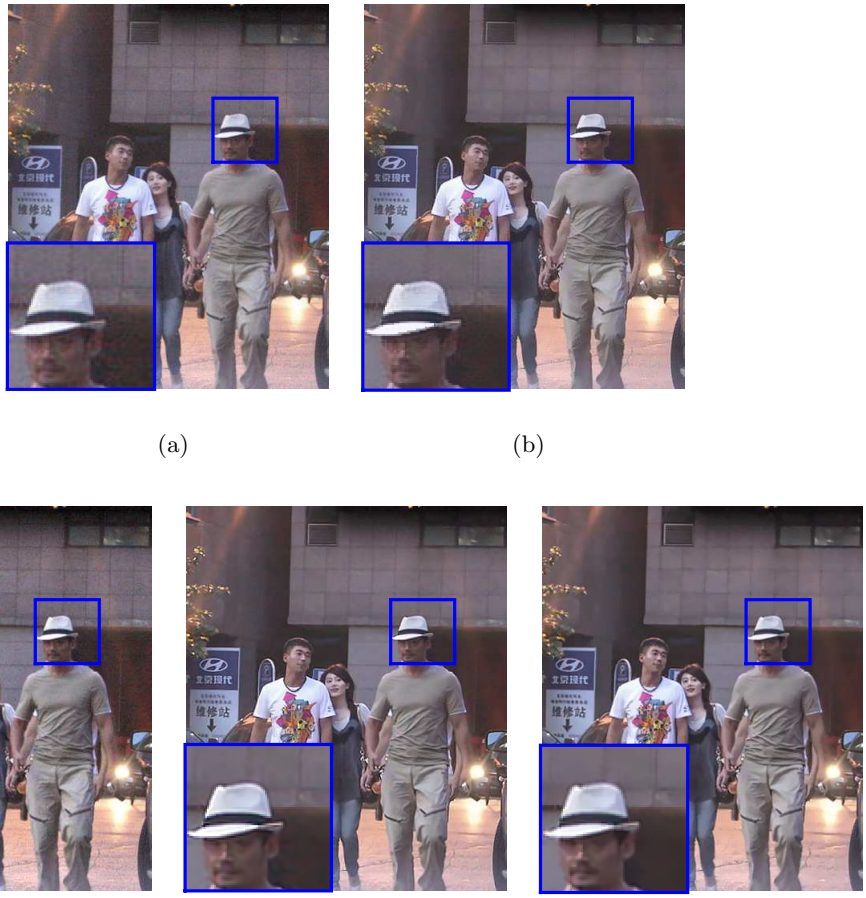


Fig. 4.9: (a)input image, (b)output of [5], (c)output of [43], (d)output of [128], and (e)output of our sharpening algorithm with $h_1 = 2.7, k = 4, \beta = 1.7$.

parameters β and h_1.

Here, we rely on subjective evaluation to tune these parameters according to the level of noise and blur in the input image. However, any appropriate metric can be exploited to come up with automatic parameter tuning for the proposed framework. For instance, sharpness metrics in [33,129] could be used for automatic parameter selection.



(a) (b)

(c) (d) (e)

Fig. 4.10: (a)input image, (b)output of [5], (c)output of [43], (d)output of [128], and (e)output of our sharpening algorithm with $h_1 = 1.4, k = 4.5, \beta = 1.2$.

4.5.2 Real Sharpening Examples

Next, we verify the performance of our algorithm for dealing with different real scenarios. we compare the quality of images produced by our method with those of other existing algorithms for sharpness and contrast enhancement in [5,43,128]. The parameters of all algorithms are set for best subjective performance. Figure 4.8(a) depicts the input image with moderate noise and blur. Figure 4.8(e) shows the output of our sharpening algorithm compared to those of algorithms [5, 43, 128] in Figure 4.8(b)-(d). In all experiments the sharpening parameter β is selected to be equal to 0.2 for chroma channels in our proposed algorithm. As can be seen, our algorithm is better able to restore the fine details while controlling the effect of noise compared to the enhancement algorithms in [5, 43]. Compared to the result of [128], our method avoids noise amplification artifacts while reducing the unpleasant halo effects along edges. Figure 4.9 illustrates another real example with stronger noise and color artifacts. As is evident from the results, the sharpening algorithm in [5] produces noise amplification artifacts when the amount of noise is high in the input image. Also, the enhancement method in [43] does not effectively perform noise reduction and sharpening operations simultaneously in this case. The sharpening method in [128] produces edge artifacts as can be seen in Fig. 4.9(d). However, our proposed algorithm is able to effectively sharpen the edges while avoiding noise and edge artifacts and reducing the color distortions, as can be seen in Fig. 4.9(e). The example in Fig. 4.10 is even more challenging as it contains more severe noise and moderate blur along with color artifacts, especially across edges.

The algorithm in [5] does not carefully control the effect of noise in the input image and produces noise related unpleasant artifacts. Similarly, the enhancement code in [43] does not effectively deal with artifacts due to the the noise in the input image. The algorithm in [128], however, is able to provide a good level of noise reduction and contrast enhancement (Fig. 4.10(d)). As regards visual quality, our result in Fig. 4.10(e) is more effective in reducing the color artifacts.

Summary - We have proposed a new sharpening filter based on the spectral properties of data-adaptive smoothing and Laplacian matrices. We have established an analysis framework that explains the relationship between our proposed sharpening procedure and classical DoG filter. This analysis further sheds light on the underlying parameters in our proposed formulation and introduces a data dependent and noise robust version of the DoG operator for use in relevant applications. We have also shown special cases of our more general approach. We verified the effectiveness of the proposed method for sharpening real noisy and blurry images.

4.A Relationship Between Laplacian of Gaussian and Difference of Gaussians Operators

DoG operator in (4.2) can be viewed as an approximation of the Laplacian of Gaussian operator [63, 64]. More formally, we start from the diffusion equation

$$\frac{1}{2}\nabla^2(G_\varepsilon * Z) = \partial_{\varepsilon^2}(G_\varepsilon * Z), \tag{4.9}$$

in which Z is the input image, ε^2 is the scale parameter (variance of the Gaussian kernel

G_ε), and $\nabla^2(G_\varepsilon * Z) = \frac{\partial^2(G_\varepsilon * Z)}{\partial x^2} + \frac{\partial^2(G_\varepsilon * Z)}{\partial y^2}$. Also, we have [64]

$$\partial_{\varepsilon^2}(G_\varepsilon * Z) \approx \frac{(G_{k\varepsilon} * Z) - (G_\varepsilon * Z)}{(k^2 - 1)\varepsilon^2}. \tag{4.10}$$

One can combine equations (4.9) and (4.10) as

$$-\frac{1}{2}\nabla^2(G_\varepsilon * Z) \approx \frac{(DoG_{\varepsilon,k} * Z)}{(k^2 - 1)\varepsilon^2}, \tag{4.11}$$

or equivalently,

$$-\frac{1}{2}(k^2 - 1)\varepsilon^2\nabla^2(G_\varepsilon * Z) \approx (DoG_{\varepsilon,k} * Z), \tag{4.12}$$

where $\varepsilon^2\nabla^2(G_\varepsilon * Z)$ is the output of the scale normalized Laplacian of Gaussian operator [64]. Equation (4.12) implies that the output of the DoG operator is an approximation of the corresponding response of the scale normalized Laplacian up to a constant factor.

Chapter 5

A Discussion on The Range of The Eigenvalues of Different Normalized Graph Laplacians

Abstract - In this chapter, we try to find the range of the eigenvalues for different normalized graph Laplacians; random walk Laplacian ($I - D^{-1}K$), traditional normalized graph Laplacian[1] ($I - D^{-1/2}KD^{-1/2}$), and the newly introduced normalized Laplacian ($I - C^{-1/2}KC^{-1/2}$).

5.1 Definitions and Theorems Used in Analyses

As discussed earlier, a graph $G = (V, E, K)$ is defined as a finite set of vertices V which are connected via a finite set $E \subset V \times V$ of edges (i, j) with the corresponding

[1]The diagonal element d_i in D is usually referred to as the degree of the vertex i in the graph.

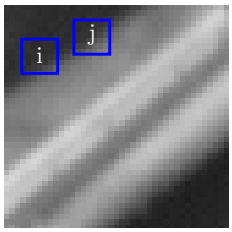


Fig. 5.1: 41×41 image used in the experiments and patches around pixels i and j of the image.

weights $K(i,j)$ which measure the similarity between vertices i and j in the graph. The similarity weights $K(i,j)$ are derived using an appropriate non-negative kernel function $k(\mathbf{f}_i, \mathbf{f}_j)$ where $\mathbf{f}_i$ and $\mathbf{f}_j$ are the feature vectors associated with vertices i and j, respectively. Putting all these weights together, the similarity (affinity) matrix K of the graph G is derived. The graph Laplacian matrix is then defined whose spectral properties characterize the underlying structure of the graph with data points as its vertices. There are at least two different definitions of the *normalized* graph Laplacian in the literature [20,114]. Also, a new definition of normalized graph Laplacian has been presented in this dissertation with some desired spectral properties [51–54, 70].

The eigenvalues of graphs (eigenvalues of graph Laplacians) are intimately related to almost all parameters of a graph and play an important part in understanding graph structures [20]. In this chapter, we focus on *normalized* graph Laplacians, and we present a discussion on the range of the eigenvalues of these matrices for different choices of kernel similarity function k and different types of graph construction. Note that we illustrate our results in the context of graph representation of images. However,

the conclusions are valid for a general graph in other applications as well. In the context of graph representation for image processing, the vertices of the graph are image pixels. As depicted in Fig. 5.1, the feature vectors associated with each vertex (pixel) are usually considered to be the column stacked version of the intensity values in a small patch around each pixel in the image [11,69]. A popular choice for the kernel definition is [11,36]

$$K(i,j) = \exp(-\frac{\|\mathbf{f}_i - \mathbf{f}_j\|^2}{h^2}). \tag{5.1}$$

A path in a graph is a sequence of edges connecting a sequence of distinct vertices in the graph. A graph G is connected if any two vertices in G can be joined by a path [6]. A subset of vertices $B \subset V$ of G (a subgraph of G) is called a (connected) component if any two vertices in B are connected by a path and if there are no connections between vertices in B and the rest of the graph. A graph in which each vertex is connected to all other vertices, or equivalently the kernel similarity matrix is full, is called a complete graph. Next, we clarify some definitions used in this chapter and explain the theorems and facts that support our conclusions and derivations.

5.1.1 Spectral Radius of Stochastic Matrices

The spectral radius $\rho(A)$ of a matrix A is defined as the largest absolute value of its eigenvalues. If we define the sum of the elements of the ith row of A as r_i and the sum of the elements of its jth column as c_j, the spectral radius satisfies the following inequalities [84]

$$\min_i r_i \leq \rho(A) \leq \max_i r_i \tag{5.2}$$

$$\min_j c_j \leq \rho(A) \leq \max_j c_j. \tag{5.3}$$

Note that for row-stochastic matrices, $\min_i r_i = \max_i r_i = 1$. Also, for doubly-stochastic matrices, we have both $\min_i r_i = \max_i r_i = 1$ and $\min_j c_j = \max_i c_j = 1$, such that for these matrices the spectral radius is unity.

5.1.2 Irreducible matrices and connected graphs

A matrix A is irreducible if there does not exist a permutation matrix P such that [65, 84]

$$PAP^T = \begin{pmatrix} B & 0 \\ C & D \end{pmatrix}$$

in which B and D are square matrices. It implies that for an irreducible matrix A and for any i and j, there exists some s such that $A^s(i,j) > 0$ [65, 84]. There is a nice connection between irreducible matrices defined on a graph and the connectedness of the corresponding graph. In fact, $D^{-1}K$ can be interpreted as a transition probability of a Markov chain defined on the vertices of a graph [20, 67]. It turns out that the following three statements are equivalent [20]:

1. Graph G is connected[2].

2. $D^{-1}K$ is irreducible.

3. The second smallest eigenvalue of the normalized graph Laplacian $I - D^{-1}K$ is strictly positive.

[2]Note that in most applications of graph theory for image enhancement, the corresponding graph is connected as the four local nearest neighbors of each pixel in the image are taken into account when building the similarity matrix K [35].

5.1.3 Perron-Frobenius theorem

For any irreducible matrix A with nonnegative elements, we have [45, 84, 95]

1. $\rho(A) > 0$ and $\rho(A)$ is a simple (unrepeated) eigenvalue of A.

2. The left and right eigenvectors of A corresponding to the largest eigenvalue $\rho(A)$ are both positive vectors (with positive elements).

Moreover, based on the discussion on the spectral radius of stochastic matrices in subsection 5.1.1, for any nonnegative, irreducible, and row-stochastic matrix A, the unique largest eigenvalue is unity with the corresponding constant eigenvector[3] $\mathbf{1}_n$, as by definition, we have $A\mathbf{1}_n = \mathbf{1}_n$ for stochastic matrices [69].

5.1.4 Gershgorin's theorem

Gershgorin's theorem says that the eigenvalues of a matrix lie in the union of some certain disks. Specifically, every eigenvalue λ of an $n \times n$ matrix $A = (a_{ij})$ is at least in one of the disks [45, 84]

$$|\lambda - a_{ii}| \leq R_i = \sum_{j \neq i}^{n} \mid a_{ij} \mid,\ i = 1, 2, ..., n, \tag{5.4}$$

in which R_i is the ith deleted row sum of A. In case of the row-stochastic matrix $A = D^{-1}K$ and for the Gaussian kernel in (5.1), we have $a_{ii} = \frac{1}{d_i}$ where d_i is the ith diagonal element of the matrix D. Also, we have $R_i = 1 - \frac{1}{d_i}$. Therefore, in such cases, the Gershgorin's disks take the following form

$$|\lambda - \frac{1}{d_i}| \leq 1 - \frac{1}{d_i},\ i = 1, 2, ..., n, \tag{5.5}$$

[3] or $(1/\sqrt{n})\mathbf{1}_n$ in its normalized form for an $n \times n$ symmetric matrix A.

or equivalently

$$\frac{2}{d_i} - 1 \leq \lambda \leq 1,\ i = 1, 2, ..., n. \tag{5.6}$$

Note that since generally $d_i > 1$ for image processing applications, we would have $-1 < \frac{2}{d_i} - 1 < 1$. It is evident that according to Gershgorin's theorem, the diagonal elements of D play an important role in determining the spectral range of $D^{-1}K$.

5.1.5 A Theorem on Characterizing The Graphs Preserving Positive Definiteness Upon Thresholding

Any graph (or to say, any graph structure) induces a hard thresholding operation on the elements of every symmetric $n \times n$ matrix $A = (a_{ij})$, mapping it to a new matrix A_G defined by [39]

$$(A_G)_{ij} = \begin{cases} a_{ij} & \text{if } (i,j) \in E \text{ or } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{5.7}$$

Then, the matrix A_G is said to be derived from A by thresholding A with regard to the graph G [39]. The following theorem from [38, 39] fully describes the condition for preserving positive definiteness when a graph-based thresholding is applied to a positive definite matrix A.

Theorem 3.1 in [39]: Let A be an arbitrary $n \times n$ symmetric positive definite matrix. Threshold A with respect to a graph $G = (V, E)$ with the resulting thresholded matrix A_G. Then

$$A_G \textit{ is positive definite for all positive definite } A \iff G = \cup_{i=1}^{\tau} G_i \textit{ for some } \tau \in \mathbf{N}, \tag{5.8}$$

where G_i, $i = 1, ..., \tau$, denote disjoint, *complete* components of G. It essentially certifies that the only thresholding operation that guarantees positive definiteness of the new thresholded matrix A_G is the trivial one which results in a block diagonal matrix A_G where inside each block there is no thresholding [38]. The above theorem holds true for positive semi-definite matrices as well [39].

Now that we have the required mathematical tool at our disposal, in the next section, we shall discuss on the range of the spectrum of different normalized graph Laplacians one by one. We will also investigate the effect of different graph constructions in Section 5.2.4.

5.2 Investigation of The Spectral Range of Normalized Graph Laplacian Matrices

As described in [114], there are various ways one can construct a graph using the input data points (feature vectors) associated with different vertices of the graph. Different approaches for constructing the graph result in different structures for the kernel similarity and Laplacian matrices of the graph. These graph structures can be split into two general categories: the methods resulting in *full* and the ones producing *sparse* kernel similarity matrices. When the similarity weights are computed for all pairs of vertices in the weighted graph, the resulting kernel similarity and Laplacian matrices of the graph would be *full* matrices. While for instance, for ε-neighborhood and k-nearest neighbor graph constructions, the resulting kernel similarity matrices of

the corresponding weighted graphs are sparse [114]. We investigate the spectrum of different normalized Laplacian matrices when the structure of the matrix transitions from a full matrix to a sparse one.

5.2.1 Random Walk Laplacian $I - D^{-1}K$

In general (without any assumption on the kernel function other than non-negativity of its elements), since $D^{-1}K$ is non-negative, irreducible, and row-stochastic, based on Perron-Frobenius theorem, it has spectral radius equal to 1. Therefore, in general the eigenvalues of $D^{-1}K$ are in the range $[-1, 1]$, leading to the conclusion that generally the graph Laplacian $I - D^{-1}K$ has eigenvalues in the range $[0, 2]$ as presented in e.g., [67,109]. Also, its unique smallest eigenvalue is 0 with the corresponding constant positive eigenvector. However, if K is computed from a positive definite kernel like the Gaussian kernel in (5.1), K and $D^{-1}K$ would be positive semi-definite matrices[4] [44,69]. Consequently, in such cases, the eigenvalues of $D^{-1}K$ and $I - D^{-1}K$ lie in the interval $[0, 1]$, as can be seen in [44, 69, 119].

5.2.2 Traditional Normalized Laplacian $I - D^{-1/2}KD^{-1/2}$

The spectral properties of $I - D^{-1}K$ and those of $I - D^{-1/2}KD^{-1/2}$ are closely related. Specifically, $I - D^{-1/2}KD^{-1/2}$ and $I - D^{-1}K$ have the same set of eigenvalues with different set of eigenvectors, such that if $\mathbf{v}$ is an eigenvector of $I - D^{-1}K$ with the eigenvalue λ, then $D^{1/2}\mathbf{v}$ is an eigenvector of $I - D^{-1/2}KD^{-1/2}$ with the same eigenvalue

[4] D^{-1} is a diagonal positive definite matrix and since K is positive semi-definite, $D^{-1}K$ would be a positive semi-definite matrix.

λ [20, 69] [5]. Consequently, the eigenvalues of $I - D^{-1/2}KD^{-1/2}$ are in general in the range $[0, 2]$ as presented in e.g., [99, 101]. On the other hand, when we have a positive definite kernel as in (5.1), the spectrum of $I - D^{-1/2}KD^{-1/2}$ would be in the range $[0, 1]$, as can be found in [44, 69, 119].

5.2.3 Sinkhorn Normalized Laplacian

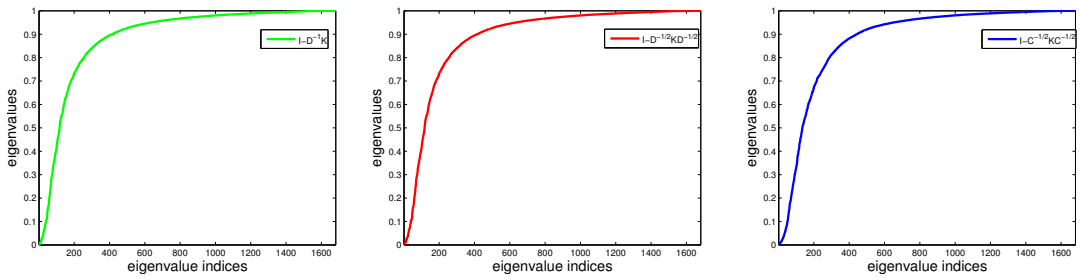
We have introduced a new type of normalized Laplacian in [51] based on the symmetrizing ideas in [70]. In [51], a fast symmetry preserving Sinkhorn-based matrix scaling algorithm is used to define the normalized graph Laplacian $I - C^{-1/2}KC^{-1/2}$. The positive normalizing coefficients in the diagonal matrix $C^{-1/2}$ are derived using the matrix balancing algorithm in [56] which yield a doubly stochastic matrix $C^{-1/2}KC^{-1/2}$. Since $C^{-1/2}KC^{-1/2}$ is nonnegative, irreducible, and doubly stochastic, the same analysis as for the random walk Laplacian applies here as well[6].

5.2.4 Spectrum of Different Graph Structures: Full Versus Sparse Similarity Matrices

In this subsection, we present an observation concerning the dependency of the graph spectrum on the sparsity of its affinity matrix. Specifically, as will be shown, some of the results with positive definite kernel functions differ when one chooses to just compute the similarity weights between each vertex (feature vector) and a *subset* of

[5] Note that the eigenvector of $I - D^{-1/2}KD^{-1/2}$ corresponding to the 0 eigenvalue is $D^{1/2}\mathbf{1}_n$, and the traditional normalized Laplacian lacks the desired property of having the constant eigenvector as one of its basis vectors which is not desirable from a filtering point of view [51, 69, 99].

[6] Note that in contrast to $D^{-1}K$, $C^{-1/2}KC^{-1/2}$ is symmetric which makes the corresponding spectral analyses easier. Also, for filtering purposes, in general $C^{-1/2}KC^{-1/2}$ outperforms $D^{-1}K$ [51, 70].

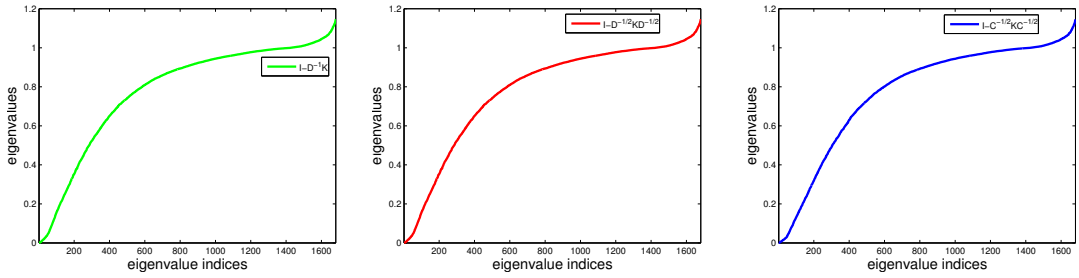


(a) (b) (c)

Fig. 5.2: Spectrum of different full normalized Laplacians: (a)spectrum of $I - D^{-1}K$, (b)spectrum of $I - D^{-1/2}KD^{-1/2}$, and (c)spectrum of $I - C^{-1/2}KC^{-1/2}$.

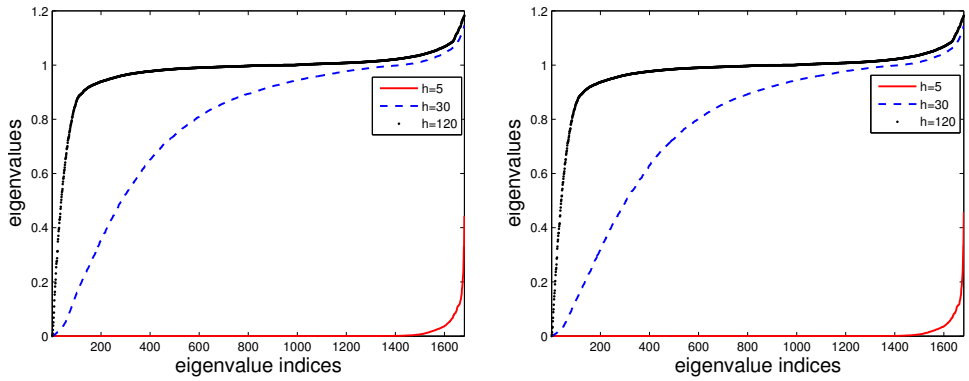
vertices rather than *all* other vertices in the graph. This decision is based on practical considerations for each specific application at hand. It essentially results in setting to zero most of the similarity coefficients in the kernel similarity matrix K and equivalently making it sparse. This changes the positive definiteness of K, even though the kernel similarity function is positive definite. For illustration purposes, we consider a part of Barbara image of size 41×41 in Fig. 5.1, from which we compute the *full* similarity matrix K based on the kernel definition in (5.1) to construct different Laplacians. In this case, Fig. 5.2 shows the spectrum of different Laplacians in increasing order. As can be seen, in this case where we use an admissible positive definite kernel and all the similarities between different pixels (vertices) are taken into account (i.e., the similarity matrix is full), the eigenvalues of all Laplacians reside in the range $[0, 1]$.

In the next experiment, we consider the same setting but with the similarity coefficients computed from a small neighborhood (11×11) around each pixel. This results in a sparse matrix K, or to say roughly an approximation of the full matrix by



(a) (b) (c)

Fig. 5.3: Spectrum of different sparse normalized Laplacians: (a)spectrum of $I - D^{-1}K$, (b)spectrum of $I - D^{-1/2}KD^{-1/2}$, and (c)spectrum of $I - C^{-1/2}KC^{-1/2}$.



(a) (b)

Fig. 5.4: Spectrum of different sparse normalized Laplacians in terms of scaling parameter h in (5.1): (a)spectrum of $I - D^{-1}K$ for different values of h, and (b)spectrum of $I - C^{-1/2}KC^{-1/2}$ for different values of h.

essentially zeroing some of its elements. The resulting spectrum of all the Laplacians in this case are plotted in Fig. 5.3. As can be seen, in this case, even though we are using the same valid positive definite kernel, the eigenvalues are not exactly in the range $[0, 1]$.

This phenomenon can be described using the theorem in Section 5.1.5. In particular, if we start from the full positive definite kernel similarity matrix K, then the sparse kernel similarity matrix derived by taking into account a small neighborhood around each pixel can be thought of as a graph-induced thresholded matrix obtained from full K. Therefore, based on the theorem in Section 5.1.5, in general there is no guarantee that the sparse kernel similarity matrix remains positive definite, even though it has been constructed using the same positive definite kernel in (5.1). As a result, in general, the normalized matrices $D^{-1}K$ and $C^{-1/2}KC^{-1/2}$ won't remain positive definite any longer. Furthermore, note that a bound on the range of the eigenvalues of $I - D^{-1}K$ can be suggested in this case. Specifically, based on the discussion in Section 5.1.4, the eigenvalues of $D^{-1}K$ lie in at least one of the intervals

$$\frac{2}{d_i} - 1 \leq \lambda \leq 1,\ i = 1, 2, ..., n. \tag{5.9}$$

or equivalently, eigenvalues of the normalized Laplacian $I - D^{-1}K$ live in the union of the intervals

$$0 \leq \lambda' \leq 2 - \frac{2}{d_i},\ i = 1, 2, ..., n, \tag{5.10}$$

in which λ' denotes a generic eigenvalue of $I - D^{-1}K$. It can be seen that changing the degree of the vertices in the graph leads to a change in the upper bound for the eigenvalues of the corresponding Laplacian matrix. Specifically, we are able to describe the effect of the scaling parameter h in (5.1) on the spectrum of the resulting Laplacian matrix. In particular, increasing h boosts the values d_i of the diagonal elements of D. It increases the upper bounds in (5.10). On the other hand, if we decrease the value of

h in (5.1), then at some point we would have a *diagonally dominant* matrix $A = D^{-1}K$, i.e.,

$$a_{ii} > R_i,\ i = 1, 2, ..., n, \tag{5.11}$$

or equivalently,

$$\frac{2}{d_i} > 1,\ i = 1, 2, ..., n, \tag{5.12}$$

which enforces the lower bounds in (5.9) to be positive. In such cases, based on Gershgorin's theorem, the sparse matrix $D^{-1}K$ would be necessarily a positive definite matrix. A similar analysis can be applied to the doubly stochastic matrix $C^{-1/2}KC^{-1/2}$ using the diagonal elements c_i of the matrix C. Figure 5.4 depicts the spectrum of $I - D^{-1}K$ and that of $I - C^{-1/2}KC^{-1/2}$ for different values of the scaling parameter h. As can be seen in Fig. 5.4, for small values of h, the corresponding spectrum of the normalized graph Laplacian would be in the range $[0, 1]$. It means that $D^{-1}K$ and $C^{-1/2}KC^{-1/2}$ are positive definite for small values of the scaling parameter h.

Another point we would like to emphasize here is that regardless of the choice of the graph structure (sparse vs. full), the resulting graph Laplacian always remains positive semi-definite, but with different range of eigenvalues for different graph constructions. For image enhancement applications, there is no clear answer yet as to which definition is most appropriate and should yield the best results in general. While such an investigation is beyond the scope of this dissertation, it can be considered as a future research work.

Summary - The proposed analysis in this chapter alleviates the ambiguity

on the reported range of the eigenvalues of different normalized graph Laplacians in the literature, as some of the existing papers report this range to be in its most general form $[0, 2]$, while others consider this range to be $[0, 1]$. Also, the proposed analysis unifies the mathematical background behind each case.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this dissertation, we introduced a general framework for image restoration using self-similarity information in images. We proposed a new definition of normalized graph Laplacian for image processing applications using a specific matrix scaling procedure applied to the similarity matrix. We used the resulting filtering and Laplacian operators to develop algorithms for different restoration tasks including image deblurring, image denoising, and sharpening.

Taking advantage of the graph Laplacian operator, a new similarity-based prior was constructed for regularizing the ill-posed deblurring problem in Chapter 2. We introduced a novel objective function for deblurring with a new data fidelity term and the corresponding regularization term derived from our definition of the Laplacian matrix. We showed that minimizing the corresponding objective function yields state-

of-the-art results for different types of blurs including out-of-focus and motion blurs. Also, our graph-based formulation provides a framework for spectral analysis of the underlying restoration method. It is also general in the sense that any other definition of the graph Laplacian can be incorporated within the same framework. Moreover, we showed that our specific definition of the normalized graph Laplacian gives performance and analysis advantages over the previous definitions.

We developed a new graph-based regularization framework for image denoising in Chapter 3 with appropriate selection of the data fidelity term. We showed that our denoising algorithm is able to describe the underlying mechanism for some well-known similarity-based denoising methods like NLM [11] and outperform such methods. Also, we compared the performance of our proposed denoising algorithm with that of global similarity-based denoising method in [107], and demonstrated its capability for producing comparable results with analysis and run-time advantages. Furthermore, we explored the connection between our denoising method and two widely used iterations for improving the performance of denoising methods, namely, boosting and diffusion iterations.

In Chapter 4, we developed an efficient yet effective sharpening algorithm based on the notion of difference of smoothing operators. We took advantage of the smoothing and sharpening operators introduced in previous chapters to develop a new algorithm for image sharpening. We showed via synthetic and real examples that our sharpening algorithm generates very good results even in difficult situations where the input image is both noisy and moderately blurred. We established the connection between our

proposed detail enhancement algorithm and the widely used difference of Gaussians (DoG) operators. Specifically, we showed that our algorithm provides a data-dependent variant of the DoG operators. Therefore, it can be used in different image processing, and computer vision applications where there is a need for better performance.

In Chapter 5, we provided a discussion on the range of the eigenvalues of different normalized graph Laplacians. The eigenvalues of graph Laplacians are good indicators of graph structures. Chapter 5 alleviates the existing ambiguity in reported range of the spectrum of normalized graph Laplacians in the literature.

6.2 Future Work and Extensions

As discussed, our graph-based framework provides a general path that can be applied to different restoration problems. In this section, we briefly discuss future research directions within our proposed framework.

6.2.1 Optimal Eigenbasis for Representing Any Given Image

There is an interesting question that is worthwhile answering within our graph-based formulation. Given any clean (distortion-free) image vector $\mathbf{z} = [z_1, ..., z_n]^T$, what is the best (optimal) $(m+1)$-dimensional subspace $V_m = [\mathbf{v}_0, \mathbf{v}_1, ..., \mathbf{v}_m]$ for representing the signal $\mathbf{z}$? Finding an appropriate answer to this question will be useful for constructing the corresponding orthogonal filtering bases for different image analysis problems. For this purpose, algorithms similar to Orthogonal Matching Pursuit (OMP) can be used [28]. The following algorithm provides an example for this purpose.

Algorithm 3 Algorithm for finding optimal subspace for representing any given signal

Initializations:

$\mathbf{v}_0 = \frac{1}{\sqrt{n}}\mathbf{1}_n$, $\mathbf{r}_0 = \mathbf{z} - \mathbf{v}_0\mathbf{v}_0^T\mathbf{z}$, and $V_0 = [\mathbf{v}_0]$.

for $k = 1 : m$ **do**

- Minimize $\|\mathbf{r}_{k-1} - \mathbf{v}_k\mathbf{v}_k^T\mathbf{r}_{k-1}\|^2$ over $\mathbf{v}_k$, and subject to $\mathbf{v}_k^T V_{k-1} = \mathbf{0}$ and $\mathbf{v}_k^T\mathbf{v}_k = 1$ (Orthogonality to the existing eigenvectors and orthonormality condition).

- Update: $\mathbf{r}_k = \mathbf{r}_{k-1} - \mathbf{v}_k\mathbf{v}_k^T\mathbf{r}_{k-1}$ and

- $V_k = [\mathbf{v}_0, \mathbf{v}_1, ..., \mathbf{v}_k]$.

end for

return V_m

In the above algorithm, one starts with the constant eigenvector $\mathbf{v}_0 = \frac{1}{\sqrt{n}}\mathbf{1}_n$. Then, the element corresponding to this eigenvector will be peeled off from the signal to get $\mathbf{r}_0 = \mathbf{z} - \mathbf{v}_0\mathbf{v}_0^T\mathbf{z}$. Next, the best eigen basis $\mathbf{v}_1$ that optimally represents $\mathbf{r}_0$ and is orthogonal to $\mathbf{v}_0$ is found. This process is repeated to derive the set of optimal $m+1$ orthonormal eigenvectors.

6.2.2 A Unified Framework for Estimating The Filtering Matrix and The Latent Image

In our current formulation, we first construct the filtering matrix W and the corresponding graph Laplacian $I - W$ using predefined kernel definitions. These filters are built from a pre-filtered version of the input degraded image. Then, we use these matrices in a regularization framework to address different restoration problems. What

if we use a unified framework to alternately estimate the matrix W (or equivalently, its eigenvalues and eigenvectors) and the latent image $\mathbf{z}$? For instance, if we start from the following cost function for image denoising

$$E(\mathbf{z}, W) = \|W(\mathbf{y} - \mathbf{z})\|^2 + \eta\|(I - W)\mathbf{z}\|^2, \tag{6.1}$$

and noting that $W = VSV^T$, (6.1) can be rewritten as

$$E(\mathbf{z}, S, V) = \|SV^T(\mathbf{y} - \mathbf{z})\|^2 + \eta\|(I - S)V^T\mathbf{z}\|^2. \tag{6.2}$$

In this case, we can start from an appropriate initialization for the latent image $\mathbf{z}$ as well as the orthonormal matrix V and the diagonal matrix S and alternate between minimizing the above cost function for $\mathbf{z}$ on one hand, and for S and V on the other hand. The quadratic nature of the above cost function guarantees the convexity of the cost function in each case when the other two variables are kept fixed and the objective function is solved for the third one. Note that this procedure can also be extended to the more general deblurring problem.

6.2.3 Patch Ordering Idea As Sorting Nodes In A Graph

Recently, the idea of image processing using reordering of image patches has been presented in [89]. This idea comprises of two steps: first finding an appropriate ordering of all patches in an image, and second processing the reordered patches. The effectiveness of this approach for different image restoration tasks has been shown in [89,90,113]. However, there is still a need for a unified framework to explain and further improve this approach. As discussed in previous chapters, patches in an image can be

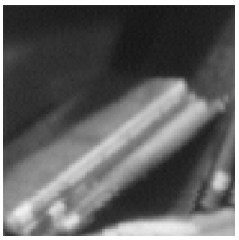
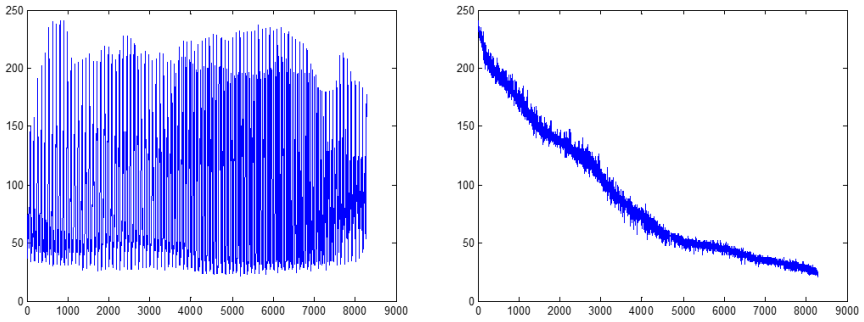


Fig. 6.1: Original image used for computing the similarity matrix K.



(a) (b)

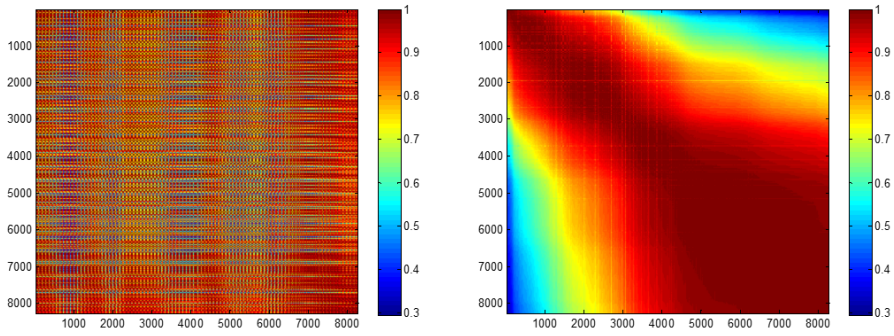
Fig. 6.2: (a)Pixel intensity values (pixels in raster-scan order), and (b)pixel intensity values when pixels are ordered using the permutation from sorting the second eigenvector of the graph Laplacian matrix.

considered as nodes in the corresponding graph. It is feasible to put the patch ordering idea in a graph-based framework using our proposed formulation. This will give a better understanding for the above mentioned steps in related algorithms and provides a path for their further improvement. In what follows, we briefly discuss the underlying ideas.

6.2.3.1 Patch Ordering Using a Graph-based Formulation

It can be shown that the required permutation for sorting the nodes of a graph can be acquired through sorting the elements of the eigenvector corresponding to the second smallest eigenvalue of the corresponding graph Laplacian matrix [102]. This idea can be traced back to as spectral embedding.

For illustration purposes, we consider an image of size 91×91 in Fig. 6.1. We use NLM kernel similarity definition to build the similarity matrix K and the corresponding un-normalized graph Laplacian $D - K$. Then, we use the Nyström algorithm [107, 120] to efficiently compute the second eigenvector of the corresponding graph Laplacian. Figure 6.2(a) shows the pixel values after sorting the pixels as a column stacked vector (i.e., image vector $\mathbf{z}$). Also, Fig. 6.2(b) depicts the pixel values in the permuted vector $P\mathbf{z}$. Note that the permutation matrix P is derived by sorting the elements of the second eigenvector of the associated un-normalized graph Laplacian. As can be seen, this graph-based technique provides a reasonable sorting of image pixels in terms of their intensities. Figures 6.3(a) and 6.3(b) depict the kernel similarity matrices before and after sorting, respectively. As can be seen in Fig. 6.3(b), pixel index sorting provides a rough clustering of pixels in the resulting kernel similarity matrix, with similar pixels concentrated on specific parts of this matrix. In fact, using patch sorting idea in a graph-based framework is advantageous as it brings similar pixels together. This actually facilitates processing similar pixels in a local manner after sorting.



(a) (b)

Fig. 6.3: (a)Kernel similarity matrix from pixels indexed by raster-scan ordered version of the input image $\mathbf{z}$, and (b)kernel similarity matrix indexed by the permutation matrix P.

6.2.3.2 Regularized Restoration Framework Using Filtering And Laplacian Matrices After Sorting The Graph Nodes (Image Pixels)

After sorting the pixels via a graph-based formulation, one is able to bring similar pixels together and compute the kernel similarity for each pixel with its neighboring *similar* pixels rather than just using the pixels in a spatial neighborhood around it. Thereby, a new kernel similarity matrix is constructed which is better able to describe the similarity relationship among different pixels. Consequently, the resulting regularization term $(P\mathbf{z})^T(I - W_P)(P\mathbf{z})$ leads to a better smoothness prior for regularizing restoration problems[1].

[1]Note that, in this respect, W_P is a sparse matrix computed *locally* after rearranging the pixels in $\mathbf{z}$ in an ordered manner $P\mathbf{z}$. Therefore, the sparse matrices W and W_P are not simply a permuted version of each other; i.e., in general, $W_P \neq P^T W P$.

Bibliography

[1] Luis Alvarez and Luis Mazorra. Signal and image restoration using shock filters and anisotropic diffusion. *SIAM Journal on Numerical Analysis*, 31(2):590–605, 1994.

[2] T. Arici, S. Dikbas, and Y. Altunbasak. A histogram modification framework and its application for image contrast enhancement. *IEEE Transactions on Image Processing*, 18(9):1921–1935, Sept 2009.

[3] Mathieu Aubry, Sylvain Paris, Samuel W Hasinoff, Jan Kautz, and Frédo Durand. Fast local Laplacian filters: Theory and applications. *ACM Transactions on Graphics (TOG)*, 33(5):167, 2014.

[4] Jean-François Aujol, Guy Gilboa, Tony Chan, and Stanley Osher. Structure-texture image decomposition - modeling, algorithms, and parameter selection. *International Journal of Computer Vision*, 67(1):111–136, 2006.

[5] Radu Ciprian Bilcu and Markku Vehvilainen. Constrained unsharp masking for image enhancement. In *Image and Signal Processing*, pages 10–19. Springer, 2008.

[6] Béla Bollobás. *Modern graph theory*, volume 184. Springer, 1998.

[7] Alexey Boreskov and Evgeniy Shikin. *Computer Graphics: From Pixels to Programmable Graphics Hardware*. CRC Press, 2013.

[8] Max Born and Emil Wolf. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. CUP Archive, 2000.

[9] Sébastien Bougleux, Abderrahim Elmoataz, and Mahmoud Melkemi. Local and nonlocal discrete regularization on weighted graphs for image and mesh processing. *International Journal of Computer Vision*, 84(2):220–236, 2009.

[10] Jérôme Boulanger, Charles Kervrann, Patrick Bouthemy, Peter Elbau, J-B Sibarita, and Jean Salamero. Patch-based nonlocal functional for denoising fluorescence microscopy image sequences. *IEEE Transactions on Medical Imaging*, 29(2):442–454, 2010.

[11] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling and Simulation*, 4:490–530, 2005.

[12] Antoni Buades, Triet M Le, Jean-Michel Morel, and Luminita A Vese. Fast cartoon+ texture image filters. *IEEE Transactions on Image Processing*, 19(8):1978–1986, 2010.

[13] Joonyoung Chang, Hee Kang, and Moon Gi Kang. Correction of axial and lateral chromatic aberration with false color filtering. *IEEE Transactions on Image Processing*, 22(3):1186–1198, March 2013.

[14] G. Chantas, N.P. Galatsanos, R. Molina, and A.K. Katsaggelos. Variational Bayesian image restoration with a product of spatially weighted total variation image priors. *IEEE Transactions on Image Processing*, 19(2):351 –362, Feb. 2010.

[15] Sunghyun Cho and Seungyong Lee. Fast motion deblurring. In *ACM Transactions on Graphics (SIGGRAPH ASIA)*, volume 28, page 145, 2009.

[16] Sunghyun Cho, Jue Wang, and Seungyong Lee. Handling outliers in non-blind image deconvolution. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 495–502. IEEE, 2011.

[17] Taeg Sang Cho, C.L. Zitnick, N. Joshi, Sing Bing Kang, R. Szeliski, and W.T. Freeman. Image restoration by matching gradient distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):683 –694, April 2012.

[18] A. Choudhury and G. Medioni. Perceptually motivated automatic color contrast enhancement. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1893–1900, Sept 2009.

[19] A. Choudhury and G. Medioni. Perceptually motivated automatic sharpness enhancement using hierarchy of non-local means. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 730–737, Nov 2011.

[20] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[21] K.. Dabov, A.. Foi, V.. Katkovnik, and K.. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080 –2095, Aug. 2007.

[22] A. Danielyan, V. Katkovnik, and K. Egiazarian. BM3D frames and variational image deblurring. *IEEE Transactions on Image Processing*, 21(4):1715 –1728, April 2012.

[23] Guang Deng. A generalized unsharp masking algorithm. *IEEE Transactions on Image Processing*, 20(5):1249–1261, May 2011.

[24] M Donatelli, C Estatico, A Martinelli, and S Serra-Capizzano. Improved image deblurring with anti-reflective boundary conditions and re-blurring. *Inverse Problems*, 22(6):2035, 2006.

[25] W. Dong, L. Zhang, G. Shi, and X. Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620 –1630, April 2013.

[26] David L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

[27] M. Elad, P. Milanfar, and R. Rubinstein. Analysis versus synthesis in signal priors. *Inverse Problems*, 23(3):947–968, June 2007.

[28] Michael Elad and Michal Aharon. Image denoising via sparse and redundant

representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

[29] Abderrahim Elmoataz, Olivier Lezoray, and Sébastien Bougleux. Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing. *IEEE Transactions on Image Processing*, 17(7):1047–1060, 2008.

[30] Hilda Faraji and W James MacLean. CCD noise removal in digital images. *IEEE Transactions on Image Processing*, 15(9):2676–2685, 2006.

[31] Zeev Farbman, Raanan Fattal, and Dani Lischinski. Diffusion maps for edge-aware image editing. In *ACM Transactions on Graphics (TOG)*, volume 29, page 145. ACM, 2010.

[32] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. In *ACM Transactions on Graphics (TOG)*, volume 27, page 67. ACM, 2008.

[33] R. Ferzli and L.J. Karam. A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB). *IEEE Transactions on Image Processing*, 18(4):717–728, April 2009.

[34] G. Gilboa, N. Sochen, and Y.Y. Zeevi. Image enhancement and denoising by complex diffusion processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1020–1036, Aug 2004.

[35] Guy Gilboa and Stanley Osher. Nonlocal linear image regularization and supervised segmentation. *Multiscale Modeling & Simulation*, 6(2):595–630, 2007.

[36] Guy Gilboa and Stanley Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028, 2008.

[37] Guy Gilboa, Nir Sochen, and Yehoshua Y Zeevi. Forward-and-backward diffusion processes for adaptive image enhancement and denoising. *IEEE Transactions on Image Processing*, 11(7):689–703, 2002.

[38] D. Guillot and B. Rajaratnam. Functions preserving positive definiteness for sparse matrices. *Trans. Amer. Math. Soc.*, 367:627–649, 2015.

[39] Dominique Guillot and Bala Rajaratnam. Retaining positive definiteness in thresholded matrices. *Linear Algebra and its Applications*, 436(11):4143–4160, 2012.

[40] Ankit Gupta, Neel Joshi, Larry Zitnick, Michael Cohen, and Brian Curless. Single image deblurring using motion density functions. In *ECCV '10: Proceedings of the 10th European Conference on Computer Vision*, 2010.

[41] Per Christian Hansen. Regularization tools version 4.0 for Matlab 7.3. *Numerical Algorithms*, 46(2):189–194, 2007.

[42] Per Christian Hansen, James G. Nagy, and Dianne P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering*. SIAM, 1 edition, 2006.

[43] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, June 2013.

[44] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220, 2008.

[45] Johnson C.R. Horn, R.A. *Matrix Analysis*. U.K.: Cambridge Univ. Press, 1991.

[46] Mohamed Hussein, Fatih Porikli, and Larry Davis. Kernel integral images: A framework for fast non-uniform filtering. In *Computer Vision and Pattern Recognition, 2008.*

[47] John Immerkaer. Fast noise variance estimation. *Computer Vision and Image Understanding*, 64(2):300–302, 1996.

[48] Neel Joshi, Sing Bing Kang, C Lawrence Zitnick, and Richard Szeliski. Image deblurring using inertial measurement sensors. In *ACM Transactions on Graphics (TOG)*, volume 29, page 30. ACM, 2010.

[49] J. Kaiser and R. Hamming. Sharpening the response of a symmetric nonrecursive filter by multiple use of the same filter. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 25(5):415–422, Oct 1977.

[50] Michael Kass and Justin Solomon. Smoothed local histogram filters. In *ACM Transactions on Graphics (TOG)*, volume 29, page 100. ACM, 2010.

[51] A. Kheradmand and P. Milanfar. A general framework for regularized, similarity-

based image restoration. *IEEE Transactions on Image Processing*, 23(12):5136–5151, Dec 2014.

[52] Amin Kheradmand and Peyman Milanfar. A general framework for kernel similarity-based image denoising. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pages 415–418. IEEE, 2013.

[53] Amin Kheradmand and Peyman Milanfar. Motion deblurring with graph Laplacian regularization. In *SPIE/IS&T Electronic Imaging*, pages 94040C–94040C. International Society for Optics and Photonics, 2015.

[54] Amin Kheradmand and Peyman Milanfar. Non-linear structure-aware image sharpening with difference of smoothing operators. *Frontiers in ICT*, 2:22, 2015.

[55] Sang Ho Kim and Jan P. Allebach. Optimal unsharp mask for image sharpening and noise removal. *Journal of Electronic Imaging*, 14(2):023005–023005–13, 2005.

[56] Philip A Knight and Daniel Ruiz. A fast algorithm for matrix balancing. *IMA Journal of Numerical Analysis*, 33:1029–1047, 2013.

[57] Pierre Kornprobst and Jack Tumblin. *Bilateral filtering: Theory and applications*. Now Publishers Inc, 2009.

[58] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. *NIPS*, pages 1033–1041, 2009.

[59] S. Lefkimmiatis, A. Bourquard, and M. Unser. Hessian-based norm regularization

for image restoration with biomedical applications. *IEEE Transactions on Image Processing*, 21(3):983 –995, March 2012.

[60] Anat Levin, Rob Fergus, Frédo Durand, and William T Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics (SIGGRAPH)*, 26(3):70, 2007.

[61] Xin Li. Fine-granularity and spatially-adaptive regularization for projection-based image deblurring. *IEEE Transactions on Image Processing*, 20(4):971 –983, April 2011.

[62] Zhengguo Li, Jinghong Zheng, Zijian Zhu, Wei Yao, and Shiqian Wu. Weighted guided image filtering. *IEEE Transactions on Image Processing*, 24(1):120–129, Jan 2015.

[63] Tony Lindeberg. *Scale-Space Theory in Computer Vision*. Springer Science & Business Media, 1993.

[64] Tony Lindeberg. Image matching using generalized scale-space interest points. In *Journal of Mathematical Imaging and Vision*, volume 52, pages 3–36. Springer Berlin/Heidelberg, 2015.

[65] Marvin Marcus and Henryk Minc. *A survey of matrix theory and matrix inequalities*, volume 14. Courier Dover Publications, 1992.

[66] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217, 1980.

[67] Marina Meila and Jianbo Shi. A random walks view of spectral segmentation. In *8th International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2001.

[68] François G Meyer and Xilin Shen. Perturbation of the eigenvectors of the graph Laplacian: Application to image denoising. *Applied and Computational Harmonic Analysis*, 36(2):326–334, 2014.

[69] P. Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *Signal Processing Magazine, IEEE*, 30(1):106–128, 2013.

[70] Peyman Milanfar. Symmetrizing smoothing filters. *SIAM Journal on Imaging Sciences*, 6(1):263–284, 2013.

[71] Dongbo Min, Sunghwan Choi, Jiangbo Lu, Bumsub Ham, Kwanghoon Sohn, and M.N. Do. Fast global image smoothing based on weighted least squares. *IEEE Transactions on Image Processing*, 23(12):5638–5653, Dec 2014.

[72] James G Nagy, Katrina Palmer, and Lisa Perrone. Iterative methods for image deblurring: a MATLAB object-oriented approach. *Numerical Algorithms*, 36(1):73–93, 2004.

[73] R. Neelamani, Hyeokho Choi, and R. Baraniuk. Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems. *IEEE Transactions on Signal Processing*, 52(2):418 – 433, Feb. 2004.

[74] Jie Ni, Pavan Turaga, Vishal M. Patel, and Rama Chellappa. Example-driven

manifold priors for image deconvolution. *IEEE Transactions on Image Processing*, 20(11):3086–3096, Nov. 2011.

[75] João P Oliveira, José M Bioucas-Dias, and Mário AT Figueiredo. Adaptive total variation image deblurring: A majorization–minimization approach. *Signal Processing*, 89(9):1683–1693, 2009.

[76] Stanley Osher and Leonid I Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis*, 27(4):919–940, 1990.

[77] K. Panetta, Yicong Zhou, S. Agaian, and Hongwei Jia. Nonlinear unsharp masking for mammogram enhancement. *IEEE Transactions on Information Technology in Biomedicine*, 15(6):918–928, Nov 2011.

[78] Sylvain Paris, Samuel W. Hasinoff, and Jan Kautz. Local Laplacian filters: Edge-aware image processing with a Laplacian pyramid. pages 68:1–68:12. ACM, 2011.

[79] Sung Hee Park, Hyung Suk Kim, S. Lansel, M. Parmar, and B.A. Wandell. A case for denoising before demosaicking color filter array data. In *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*, pages 860–864, Nov 2009.

[80] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, Jul 1990.

[81] Gabriel Peyré. Image processing with nonlocal spectral bases. *Multiscale Modeling and Simulation*, 7(2):703–730, 2008.

[82] Gabriel Peyré, Sébastien Bougleux, and Laurent Cohen. Non-local regularization of inverse problems. In *Proc. 10th European Conference on Computer Vision (ECCV08)*, pages 57–68, Oct. 2008.

[83] Cuong Cao Pham, Synh Viet Uyen Ha, and Jae Wook Jeon. Adaptive guided image filtering for sharpness enhancement and noise reduction. In *Advances in Image and Video Technology*, pages 323–334. Springer, 2012.

[84] S.U. Pillai, Torsten Suel, and Seunghun Cha. The Perron-Frobenius theorem: some of its applications. *Signal Processing Magazine, IEEE*, 22(2):62–75, March 2005.

[85] Luis Pizarro, Pavel Mrázek, Stephan Didas, Sven Grewenig, and Joachim Weickert. Generalised nonlocal image smoothing. *International Journal of Computer Vision*, 90(1):62–87, 2010.

[86] Stephen M Pizer, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987.

[87] A. Polesel, G. Ramponi, and V.John Mathews. Image enhancement via adaptive

unsharp masking. *IEEE Transactions on Image Processing*, 9(3):505–510, Mar 2000.

[88] J. Rabin, J. Delon, and Y. Gousseau. Removing artefacts from color and contrast modifications. *IEEE Transactions on Image Processing*, 20(11):3073–3085, Nov 2011.

[89] I. Ram, M. Elad, and I. Cohen. Image processing using smooth ordering of its patches. *IEEE Transactions on Image Processing*, 22(7):2764–2774, July 2013.

[90] Idan Ram, Israel Cohen, and Michael Elad. Patch-ordering-based wavelet frame and its use in inverse problems. *IEEE Transactions on Image Processing*, 23(7):2779–2792, 2014.

[91] S. Ramani, T. Blu, and M. Unser. Monte-Carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms. *IEEE Transactions on Image Processing*, 17(9):1540–1554, September 2008.

[92] Yaniv Romano and Michael Elad. Boosting of image denoising algorithms. *SIAM Journal on Imaging Sciences*, 8(2):1187–1219, 2015.

[93] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[94] F. Russo. An image-enhancement system based on noise estimation. *IEEE Transactions on Instrumentation and Measurement*, 56(4):1435–1442, Aug 2007.

[95] E Seneta. *Non-negative matrices and Markov chains*. Springer, 1981.

[96] Hae-Jong Seo and Peyman Milanfar. Robust flash denoising/deblurring by iterative guided filtering. *EURASIP Journal on Advances in Signal Processing*, 2012(1):1–19, 2012.

[97] Qi Shan, Jiaya Jia, and Aseem Agarwala. High-quality motion deblurring from a single image. *ACM Transactions on Graphics (SIGGRAPH)*, 27(3):73:1–73:10, August 2008.

[98] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[99] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE, Signal Processing Magazine*, 30(3):83–98, 2013.

[100] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific J. Math*, 21(2):343–348, 1967.

[101] Alexander J Smola and Risi Kondor. Kernels and regularization on graphs. In *Learning theory and kernel machines*, pages 144–158. Springer, 2003.

[102] Justin Solomon. *Numerical Algorithms*. AK Peters/CRC Press, 2015.

[103] Kartic Subr, Cyril Soler, and Frédo Durand. Edge-preserving multiscale image decomposition based on local extrema. In *ACM Transactions on Graphics (TOG)*, volume 28, page 147. ACM, 2009.

[104] Arthur D Szlam, Mauro Maggioni, and Ronald R Coifman. Regularization on graphs with function-adapted diffusion processes. *The Journal of Machine Learning Research*, 9:1711–1739, 2008.

[105] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Transactions on Image Processing*, 16(2):349 –366, Feb. 2007.

[106] H. Takeda, S. Farsiu, and P. Milanfar. Deblurring using regularized locally adaptive kernel regression. *IEEE Transactions on Image Processing*, 17(4):550 –563, April 2008.

[107] H. Talebi and P. Milanfar. Global image denoising. *IEEE Transactions on Image Processing*, 23(2):755–768, Feb 2014.

[108] H. Talebi and P. Milanfar. Nonlocal image editing. *IEEE Transactions on Image Processing*, 23(10):4460–4473, Oct 2014.

[109] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358. ACM, 1995.

[110] Bart M ter Haar Romeny. *Geometry-Driven Diffusion in Computer Vision*. Kluwer Academic Publishers, 1994.

[111] Alan M Thompson, John C Brown, Jim W Kay, and D Michael Titterington. A study of methods of choosing the smoothing parameter in image restoration by

regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):326–339, 1991.

[112] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839 –846, Jan. 1998.

[113] Gregory Vaksman, Michael Zibulevsky, and Michael Elad. Patch ordering as a regularization for inverse problems in image processing. *SIAM Journal on Imaging Sciences*, 9(1):287–319, 2016.

[114] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[115] Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM J. Img. Sci.*, 1(3):248–272, August 2008.

[116] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600 –612, April 2004.

[117] Zhou Wang and Alan C Bovik. Mean squared error: love it or leave it? a new look at signal fidelity measures. *IEEE, Signal Processing Magazine*, 26(1):98–117, 2009.

[118] Joachim Weickert. *Anisotropic diffusion in image processing*. Teubner-Verlag, Stuttgart, 1998.

[119] Yair Weiss. Segmentation using eigenvectors: a unifying view. In *Proc. Seventh Intl. Conf. Computer Vision*, pages 975–982, 1999.

[120] Christopher Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Proceedings of the 14th Annual Conference on Neural Information Processing Systems*, number EPFL-CONF-161322, pages 682–688, 2001.

[121] Holger Winnemöller, Jan Eric Kyprianidis, and Sven C Olsen. XDoG: an extended difference-of-Gaussians compendium including advanced image stylization. *Computers & Graphics*, 36(6):740–753, 2012.

[122] Holger Winnemöller, Sven C Olsen, and Bruce Gooch. Real-time video abstraction. In *ACM Transactions On Graphics (TOG)*, volume 25, pages 1221–1226. ACM, 2006.

[123] Xiaolin Wu. A linear programming approach for optimal contrast-tone mapping. *IEEE Transactions on Image Processing*, 20(5):1262–1272, May 2011.

[124] Lu Yuan, Jian Sun, Long Quan, and Heung-Yeung Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. In *ACM Transactions on Graphics (SIGGRAPH)*, volume 27, page 74, 2008.

[125] Buyue Zhang and J.P. Allebach. Adaptive bilateral filter for sharpness enhancement and noise removal. *IEEE Transactions on Image Processing*, 17(5):664–678, May 2008.

[126] Haichao Zhang, David Wipf, and Yanning Zhang. Multi-image blind deblurring using a coupled adaptive sparse prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1051–1058, 2013.

[127] Xiaoqun Zhang, Martin Burger, Xavier Bresson, and Stanley Osher. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM Journal on Imaging Sciences*, 3(3):253–276, 2010.

[128] Xiang Zhu and P. Milanfar. Restoration for weakly blurred and strongly noisy images. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 103–109, Jan 2011.

[129] Xiang Zhu and Peyman Milanfar. A no-reference sharpness metric sensitive to blur and noise. In *Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on*, pages 64–69. IEEE, 2009.

[130] Xiang Zhu and Peyman Milanfar. Removing atmospheric turbulence via space-invariant deconvolution. *IEEE Transactions on pattern analysis and machine intelligence*, 35(1):157–170, 2013.

[131] Xiang Zhu, Filip Šroubek, and Peyman Milanfar. Deconvolving PSFs for a better

motion deblurring using multiple images. In *Computer Vision–ECCV 2012*, pages 636–647. Springer, 2012.

[132] M. Zibulevsky and M. Elad. L1-L2 optimization in signal and image processing. *IEEE, Signal Processing Magazine*, 27(3):76 –88, May 2010.