Unbiased Risk Estimate Algorithms for Image Deconvolution

XUE, Feng

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Electronic Engineering

The Chinese University of Hong Kong
June 2013

題獻/Dedication

獻給我的父母

To my parents

致謝

首先，我對我的論文導師Thierry Blu教授致以深深的謝意。他為我確定了這個有趣的課題，並且從始至終都給予我悉心的指導。他是一名出色的學術導師，也是一個研究與教學方面傑出的典範。他專業的指導給予了我最初的研究經驗，使我深切的體會到了什是真正的科學研究。感謝他在我們無數有趣的討論中分享他對于信號與圖像處理領域極有見地的觀點和獨特的視角。如若沒有他對研究課題全局性的把握和對每一個技術細節的密切關注，這項論文工作是不可能完成的。他對科研的熱情也激發了我對工程和科學的興趣；他敏捷的思辨使我了解到如何做真正的研究。

我也對Florian Luisier博士表達我衷心的感謝：感謝他在我們共同的研究工作中辛苦的努力和巨大的幫助。我真誠的感謝潘漢傑：在與他的討論和交流中我得到了很多有益的建議。

我非常感謝香港中文大學計算機科學與工程學系的賈佳亞教授和香港科技大學計算機科學與工程學系的權龍教授對論文工作的全面而又細心的評審。

感謝圖像與視頻處理實驗室的顏慶義教授，湛偉權教授和王曉剛教授對我研究工作提出的豐富的建議。同時我也對所有實驗室成員表達深深的謝意：李曉峰，王萌，趙瑞，李蔚，馬家廉，韋理英，趙叢，歐陽萬裏，張任奇，韓慶龍，曾星宇，劉强，馬林，盛律，李松南，董潔。我很珍惜和你們一起共度的愉快時光，這幾年的博士生涯是我生命中最為難忘的一段歲月。

我十分感謝香港中文大學生物學系的姜裏文教授為我提供顯微鏡儀器。我很感謝同王浩博士有益的討論和交流，感謝他提供的生物樣本圖片。

我對我的父母表達我最深沈的謝意，尤其是我的母親。在我整個的生命中，他們給予我無私的愛跟無條件的支持。感謝他們在我博士生涯中所給予的始終如一的堅定的無條件的支持和鼓勵。我的母親承受著比我更大的精神壓力，並且總是在我最艱難的時刻鼓勵著我堅持下去。他們的愛是我生活前進和追尋夢想的動力。沒有他們的鼓勵和支持，我不會有足夠的勇氣選擇攻讀博士學位。

謝謝我的好朋友：張梓軒博士和王進輝。

最後，感謝我的母校 — 香港中文大學。她那靜雅的山間校園，濃厚的學術氛圍和

卓越的研究環境給予了我豐富多彩且極富意義的生活。這段難忘的博士生涯不僅極大地提高了我的學術素養，也無比地豐富了我的人生。

Acknowledgements

I would like to express my deepest gratitude to my thesis advisor, Prof. Thierry Blu, for proposing this project to me and for his guidance throughout my work. He is an excellent mentor and a fantastic model for research and teaching. His expert guidance gave me my first exposure to real science and my first research experience. I thank him for sharing his great and special vision of signal and image processing during our countless interesting discussions. The thesis work would not have been possible without his global view on problems being considered and close inspection of any technical details. His passion for research stimulated my interest for engineering and science; his critical thinking shed light on how to do "real" research.

I would also like to express my gratitude to Dr. Florian Luisier for his great effort and kind help on the joint work. I sincerely thank Hanjie Pan for many fruitful exchanges during his discussions and communications.

I am obliged to Prof. Jiaya Jia and Prof. Long Quan for accepting to evaluate my thesis.

I would also like to thank Prof. King Ngi Ngan, Prof. Wai Kuen Cham and Prof. Xiao Gang Wang, who are faculty members of Image and Video Processing Lab, for their useful suggestions on my research work. I send my personal thanks to all the members of this laboratory: Hiu Fung Lee, Jing Shao, Meng Wang, Shuai Yi, Rui Zhao, Wei Li, Ka Lim Ma, Liying Wei, Cong Zhao, Wanli Ouyang, Renqi Zhang, Qinglong Han, Xingyu Zeng, Qiang Liu, Lin Ma, Lü Sheng, Songnan Li, Jie Dong. It was a pleasure and an honour to work with all of you!

Many thanks go to Prof. Liwen Jiang of Department of Biology for providing the microscopy facilities at CUHK. I thankfully acknowledge the helpful discussions and communications with Dr. Hao Wang, who also provided me with many biological samples.

I have a very special thought for my parents, for their unlimited love and unconditional support though my whole life, especially my mother. I would like to thank my parents for their She was suffering from significantly more emotional pressure than me, and always encouraged me to be persistent to survive the past periods of slowdowns. Their love motivates me to pursue my life and my dreams. I would not have enough courage to pursue Ph.D. degree without their support. Special thanks also to my dear friend, Zixuan Zhang and Jinhui Wang.

Finally, I would like to thank the postgraduate studentship of CUHK for financial support throughout my thesis.

摘要

本論文工作的主題是圖像反卷積問題。在很多實際應用，例如生物醫學成像，地震學，天文學，遙感和光學成像中，觀測數據經常會出現令人不愉快的退化現象，這種退化一般由模糊效應（例如光學衍射限條件）和噪聲汙染（比如光子計數噪聲和讀出噪聲）造成的，這兩者都是物理儀器自身的條件限制造成的。作為一個標准的線性反問題，圖像反卷積經常被用作恢復觀測到的模糊的有噪點的圖像。我們旨在基于無偏差風險估計准則研究新的反卷積算法。本論文工作主要分為以下兩大部分。

首先，我們考慮在加性高斯白噪聲條件下的圖像非盲反卷積問題，即准確的點擴散函數已知。我們的研究准則是最小化均方誤差的無偏差估計，即SURE. SURE-LET方法最初被應用于圖像降噪問題。本論文工作擴展該方法至討論圖像反卷積問題：我們提出了一個新的SURE-LET算法，用于快速有效地實現圖像複原功能。具體而言，我們將反卷積過程參數化表示為有限個基本函數的線性組合，稱作LET方法。反卷積問題最終簡化為求解該線性組合的最優線性系數。由于SURE的二次項本質和線性參數化表示，求解線性系數可由求解線性方程組而得。實驗結果顯示該論文提出的方法在信噪比，圖像的視覺質量和運算時間等方面均優于其他迄今最優秀的算法。

論文的第二部分討論圖像盲複原中的點擴散函數估計問題。我們提出了blur-SURE — 一個均方誤差修正版的無偏差估計 — 作為點擴散函數估計的最新准則，即點擴散函數由最小化這個新的目標函數獲得。然後我們利用這個估計的點擴散函數，用第一部分所提出的SURE-LET算法進行圖像的非盲複原。我們以一些典型的點擴散函數形式（高斯函數最為典型）為例詳細闡述該blur-SURE理論框架。實驗結果顯示最小化blur-SURE能夠更准確的估計點擴散函數，從而獲得更加優越的反卷積性能。相比于圖像非盲複原，盲複原所得的圖片的視覺質量損失可忽略不計。

本論文所提出的基于無偏差估計的算法可擴展至其他噪聲模型。由于本論文以SURE基礎的方法在理論上並不僅限于卷積問題，該方法可用于解決數據的其他線

性失真問題。

關鍵詞： 圖像，反卷積，均方誤差，SURE，維納濾波，冗余Haar小波

Abstract

The subject of this thesis is image deconvolution. In many real applications, e.g. biomedical imaging, seismology, astronomy, remote sensing and optical imaging, undesirable degradations by blurring effect (e.g. optical diffraction-limited condition) and noise corruption (e.g. photon-counting noise and readout noise) are inherent to any physical acquisition device. Image deconvolution, as a standard linear inverse problem, is often applied to recover the images from their blurred and noisy observations. Our interest lies in novel deconvolution algorithms based on unbiased risk estimate. This thesis is organized in two main parts as briefly summarized below.

We first consider non-blind image deconvolution with the corruption of additive white Gaussian noise (AWGN), where the point spread function (PSF) is exactly known. Our driving principle is the minimization of an unbiased estimate of mean squared error (MSE) between observed and clean data, known as "Stein's unbiased risk estimate" (SURE). The SURE-LET approach, which was originally developed for denoising, is extended to the deconvolution problem: a new SURE-LET deconvolution algorithm for fast and efficient implementation is proposed. More specifically, we parametrize the deconvolution process as a linear combination of a small number of known basic processings, which we call the linear expansion of thresholds (LET), and then minimize the SURE over the unknown linear coefficients. Due to the quadratic nature of SURE and the linear parametrization, the optimal linear weights of the combination is finally achieved by solving a linear system of equations. Experiments show that the proposed approach outperforms other state-of-the-art methods in terms of PSNR, SSIM, visual quality, as well as computation time.

The second part of this thesis is concerned with PSF estimation for blind deconvolution. We propose a "blur-SURE" — an unbiased estimate of a filtered version of MSE — as a novel criterion for estimating the PSF, from the observed image only, i.e. the PSF is identified by minimizing this new objective functional, whose validity has been theoretically verified. The blur-SURE framework is exemplified with a number of parametric forms of the PSF, most typically, the Gaussian kernel. Experiments show that the blur-SURE minimization yields highly accurate estimate of PSF parameters. We then perform non-blind deconvolution using the SURE-LET algorithm proposed in Part I, with the estimated PSF. Experiments show that the estimated PSF results in superior deconvolution performance, with a negligible quality loss, compared to the deconvolution with the exact PSF.

One may extend the algorithms based on unbiased risk estimate to other noise model. Since the SURE-based approaches does not restrict themselves to convolution operation, it is possible to extend them to other distortion scenarios.

Keywords: image, deconvolution, mean square error (MSE), Stein's unbiased risk estimate (SURE), linear expansion of thresholds (LET), blur-MSE, blur-SURE, Wiener filtering, undecimated Haar wavelet.

List of Notations

Symbols & Conventions

$\mathbf{v}$	Vector of N components v_n
v	Any of the components v_n
$\mathbf{M}$	Matrix of $L \times N$ components $m_{l,n}$
I	Identity matrix
$\mathbf{F}(\cdot)$	Vector function of N components $f_n(\cdot)$
$f(\cdot)$	Any of the scalar functions $f_n(\cdot)$

Operators

$*$	Convolution
$\mathscr{E}$	Mathematical expectation
∇	Gradient of a scalar function
$\mathrm{div}\{\cdot\}$	Divergence of a vector function
$\mathrm{trace}\{\cdot\}$	Trace of a square matrix
$\mathrm{diag}\{\cdot\}$	Diagonal of a square matrix

Acronyms

AWGN	Additive Whit Gaussian Noise
BLS	Bayesian Lease-Squares
DFT	Discrete Fourier Transform
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
GCV	Generalized Cross-Validation
GSM	Gaussian Scale Mixture
LET	Linear Expansion of Thresholds
MAD	Median of the Absolute Deviation
MAP	Maximum A Posteriori
MMSE	Minimum Mean-Squared error

MSE	Mean-Squared error
PDF	Probability Density Function
PSNR	Peak Signal-to-Noise Ratio
SNR	Signal-to-Noise Ratio
SSIM	Structural Similarity
SURE	Stein's Unbiased Risk Estimate
TV	Total Variation
UWT	Undecimated Wavelet Transform

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations and objectives

In many real applications, e.g. medical imaging [14], seismology [15], astronomical imaging [16], remote sensing [17] and optical imaging [18], the observed images are often degraded during image acquisition. There are two main sources of degradation, described as follows.

The first degradation is *blurring effect* caused by the limitations of image acquisition devices, e.g., the diffraction phenomenon of optical imaging system [19]. Blurring is often mathematically modelled by convolution, and within this model the blur is characterized by the *point spread function* (PSF) or *kernel* [20].

Another type of degradation is *noise corruption* introduced by the measurements, e.g., photon-counting noise arising from the fluctuation of the number of incoming photons [21], the thermal and readout noise of the electronic acquisition devices and the analog-to-digital conversion [22]. Although the amount of noise actually depends on the signal intensity, it is often modelled as an additive independent (typically Gaussian) random variable, especially when the magnitude of the measured signal is sufficiently high.

Due to the poor visual quality of the obtained images, image deconvolution is often applied to remove the degrading phenomena and recover the images from their blurred and noisy observations [20]. This problem can be divided into two categories: non-blind deconvolution, which assumes that the PSF is exactly known in advance, and blind deconvolution, where the PSF is (partially) unknown.

1.2 Mathematical formulation for problem statement

The degradation is formulated as the following linear model [20]:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{b} \tag{1.1}$$

where $\mathbf{y} \in \mathbb{R}^N$ is the observations of the original data $\mathbf{x} \in \mathbb{R}^N$, linearly distorted by convolution matrix $\mathbf{H} \in \mathbb{R}^{N\times N}$ and corrupted by the additive white Gaussian noise $\mathbf{b} \in \mathbb{R}^N$ with variance σ^2. Non-blind deconvolution is to find a good estimate of $\mathbf{x}$ from the measurements $\mathbf{y}$, given exact $\mathbf{H}$, whereas blind deconvolution attempts to estimate both $\mathbf{x}$ and $\mathbf{H}$, from $\mathbf{y}$ only.

Image deconvolution is an ill-conditioned or singular problem due to the nature of the PSF [20]. In particular, ill-conditioned convolution operator tends to yield highly noise-sensitive solutions, which makes the deconvolution problem particularly challenging [20].

1.3 Survey of non-blind deconvolution approaches

Since there is no unique solution to deconvolution problem in view of ill-conditioning and noise, the selection of a specific solution must be guided by some criterion or a set of criteria [20]. A huge amount of literature is dedicated to this selection. In this section, we propose a formal classification of various existing deconvolution approaches, based on their underlying criteria. For each class, we further discuss some of its most popular representatives.

1.3.1 Regularization

Regularization is a standard technique to cope with the ill-posed nature of the deconvolution problem, by imposing certain regularity conditions on the original image [23]. The principle of regularization is to find an estimate $\widehat{\mathbf{x}} \in \mathbb{R}^N$ of an original $\mathbf{x} \in \mathbb{R}^N$ by minimizing the regularized cost functional $J(\mathbf{x}, \mathbf{y})$, i.e. $\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} J(\mathbf{x}, \mathbf{y})$, which is usually of the following generic form [24]:

$$J(\mathbf{x}, \mathbf{y}) = \|\mathbf{H}\mathbf{x} - \mathbf{y}\|^2 + \lambda\Phi(\mathbf{x}) \tag{1.2}$$

where $\|\mathbf{H}\mathbf{x} - \mathbf{y}\|^2$ is the data-fidelity term, λ is the regularization parameter, $\Phi(\mathbf{x})$ is the regularization term. The most commonly used regularizers are listed below.

Wiener filtering/Tikhonov regularization [25; 26]

Tikhonov regularization uses ℓ_2-norm for regularization term $\Phi(\mathbf{x}) = \|\mathbf{Sx}\|^2$ for some suitably chosen Tikhonov matrix $\mathbf{S}$, which is usually chosen as the identity matrix to give preference to smaller norms, or high-pass operators to enforce smoothness. Tikhonov regularization yields the closed-form solution, given by [27]:

$$\widehat{\mathbf{x}} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda\mathbf{S}^{\mathrm{T}}\mathbf{S}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{y} \tag{1.3}$$

While straightforward to implement, this type of regularization often produces over-smoothed edges [28].

Transform-domain sparsity constraints [29; 30]

The coefficients of a frame-based representation of the original image are estimated under a sparsity-inducing regularizer such as the ℓ_1-norm [29], formulated as [30]:

$$J(\mathbf{x}, \mathbf{y}) = \|\mathbf{Ac} - \mathbf{y}\|^2 + \lambda\|\mathbf{c}\|_1 \tag{1.4}$$

where $\mathbf{c}$ is the transform coefficients of the unknown image $\mathbf{x}$. In particular, in wavelet-based deconvolution methods, $\mathbf{A}$ is often chosen as $\mathbf{A} = \mathbf{HW}$, where $\mathbf{W}$ denotes inverse wavelet transform. The underlying philosophy in dealing with the ℓ_1-norm regularization criterion is that most images have a sparse representation in the wavelet domain [31]. Another advantage of the ℓ_1-based regularization over the ℓ_2-based is that as opposed to the latter, ℓ_1-regularization is less sensitive to and better to preserve sharp edges.

One is the most popular methods for solving problem (1.4) is in the class of *iterative shrinkage-thresholding* (IST) algorithms[1], where each iteration involves matrix-vector multiplication involving $\mathbf{A}$ and $\mathbf{A}^{\mathrm{T}}$ followed by a shrinkage/soft-threshold step [30]. This algorithm can be reinterpreted as the proximal forward-backward iterative scheme [24], expectation-minimization algorithm [34] and majorization-minimization algorithm [35]. More recently, fast algorithms of IST have been proposed, e.g., fast thresholded Landweber algorithm [36], FISTA [37], SALSA [38] and *i*-LET [39].

Total-Variation (TV) regularization [28; 40]

The underlying principle of TV regularization is that signals with excessive and possibly

[1] Other names in the signal processing literature include, for example, thresholded Landweber method [32] and split Bregman method [33]

spurious detail have high total variation, that is, the integral of the absolute gradient of the signal is high. It can be formulated as [41]:

$$J(\mathbf{x}, \mathbf{y}) = \|\mathbf{H}\mathbf{x} - \mathbf{y}\|^2 + \lambda \|\mathbf{x}\|_{\mathrm{TV}} \tag{1.5}$$

where $\| \cdot \|_{\mathrm{TV}}$ denotes TV norm. TV regularization is remarkably effective at simultaneously preserving edges whilst smoothing away noise in flat regions, even at low signal-to-noise ratios [28]. Several recent algorithms that solves (1.5) have been proposed, see [41–44] for example.

1.3.2 Regularized inversion followed by denoising

For further improvements of deconvolution quality, the most state-of-the-art methods are usually decomposed as a two-step procedure that consists of regularized inverse followed by noise reduction. Some representatives of this popular class of approaches are described below.

ForWaRD [45]

This two-stage shrinkage procedure successively operates in Fourier and wavelet domains with an optimal balance (in terms of an approximate MSE metric) between the amount of Fourier and wavelet regularization. ForWaRD also considers different Fourier shrinkage parameters at different DWT scales.

More sophisticated denoising techniques [46–49]

The recent deconvolution algorithms, e.g., the SV-GSM [46], GSM [47], SA-DCT [48] and BM3D [49], apply the regularized inverse followed by a more sophisticated denoising technique (BLS-GSM [50], SA-DCT [51] and BM3D [52], respectively). This strategy achieves the most state-of-the-art results, usually at the expense of a high computational complexity.

1.3.3 Bayesian approach

Another class of deconvolution algorithms are derived in a Bayesian framework, where some *a priori* statistical knowledge of the original image are assumed [5; 34; 47; 53–59] . The regularization approaches can be re-interpreted in this framework as maximum a posteriori (MAP) or penalized likelihood. We refer the interested reader to [5; 60] for a more detailed description of the Bayesian approach.

1.3.4 Remark

Either regularization techniques or Bayesian methods need to find a reasonable value of the regularization parameter, to keep a good balance between data-fidelity and regularization terms [23; 61–63], see generalized cross validation (GCV) method [63] for example.

1.4 Survey of blind deconvolution approaches

Blind deconvolution is a highly ill-posed problem, especially when no parametric form of the underlying PSF is provided or assumed, due to not only the ill-conditioned degradation operator, but also the underdetermined nature of the problem itself: the number of unknowns ($\mathbf{x}$ and $\mathbf{H}$) is significantly more than that of knowns ($\mathbf{y}$), and hence, there may exist multiple solution-pairs (original image + PSF) that produce the same observed image [64].

1.4.1 Non-parametric blind deconvolution

To deal with the ill-posedness of the problem, most existing methods formulate the deconvolution process as minimization of an objective functional, by taking into account a certain prior knowledge or assumptions of the original image and the PSF, within a regularization framework or a Bayesian approach [65]. For a comprehensive review of the non-parametric blind deconvolution methods, refer to [66–68] and the references therein.

Iterative blind deconvolution algorithms

With much initial success, an iterative blind deconvolution algorithm [69] spurred a great deal of interest in this subject. This pioneering work essentially generalized the Fienup phase retrieval algorithm [70], and *a priori* knowledge is limited to the non-negativity of images. In each iteration one obtains estimates of the object and the PSF by simple inverse filtering.

Subsequent improvements and variants of this basis scheme [69] further impelled research in this field. [10; 71] proposed iterative algorithms based on Wiener-type filtering [25]; [11; 72; 73] developed Bayesian-based blind deconvolution algorithms based on Richardson-Lucy algorithm [74; 75], which has proved to be robust in the presence of noise; [76–79] proposed expectation-maximization-based [80] maximum likelihood blind deconvolution algorithms; [9; 12] proposed multiplicative iterative algorithms to maximize the likelihood probability.

Regularization technique

It imposes certain regularity conditions (e.g. smoothness) on the original image and the PSF, and formulates the blind deconvolution as the following minimization problem:

$$\min_{\mathbf{x},\mathbf{h}} \|\mathbf{Hx}-\mathbf{y}\|^2 + \lambda_1\Phi_1(\mathbf{x}) + \lambda_2\Phi_2(\mathbf{h}) \tag{1.6}$$

where Φ_1 and Φ_2 are the regularization terms of original image $\mathbf{x}$ and PSF $\mathbf{h}$, respectively; λ_1 and λ_2 are two positive regularization parameters which measure the trade off between a good fit and the regularity of the solutions $\mathbf{x}$ and $\mathbf{h}$. The commonly used choices of regularizer are:

- Tikhonov regularization [81]:

  $$\min_{\mathbf{x},\mathbf{h}} \|\mathbf{Hx}-\mathbf{y}\|^2 + \lambda_1\|\mathbf{Dx}\|^2 + \lambda_2\|\mathbf{Dh}\|^2 \tag{1.7}$$

  where $\mathbf{D}$ is the first-order differencing matrix.

- TV regularization [82]:

  $$\min_{\mathbf{x},\mathbf{h}} \|\mathbf{Hx}-\mathbf{y}\|^2 + \lambda_1\|\mathbf{x}\|_{\mathrm{TV}} + \lambda_2\|\mathbf{h}\|_{\mathrm{TV}} \tag{1.8}$$

- Hybrid regularization [83]:

  $$\min_{\mathbf{x},\mathbf{h}} \|\mathbf{Hx}-\mathbf{y}\|^2 + \lambda_1\|\mathbf{x}\|_{\mathrm{TV}} + \lambda_2\|\mathbf{Sh}\|^2 \tag{1.9}$$

  where $\mathbf{S}$ is the discrete Laplacian matrix.

- Lipschitz regularization [84], anisotropic regularizations [85], and others.

This technique can be re-interpreted as the maximum a posteriori (MAP) from Bayesian perspective.

Bayesian approach

It assumes underlying prior models on the image and the PSF, and then via Bayes' rule, formulates the logarithm of the posterior distribution as a cost functional to be optimized. It is known that the exact posterior distribution is very hard to compute. Variational Bayesian methods are employed to approximate the posterior via minimizing the Kullback-Leibler (KL) divergence related to the posterior distribution [86].

Standard prior models for the image and blur include simultaneous autoregressive (SAR) [5; 87; 88], conditional autoregressive, Gaussian prior [87] and TV prior [88]. Notice that in [88], using a TV function as the image prior and a SAR model as the blur prior under the Bayesian setting is equivalent to the formulation of [83].

Recently, more sophisticated prior models have been proposed, e.g., Student's-t distribution to model the sparse kernels [89], a mixture of Gaussians for the gradient of the image and a mixture of exponentials for the PSF [90–92], a sparse prior to model the image edges [93] and a local prior of the image [91]. Moreover, the non-convex regularization terms have also been introduced [93; 94]. Refer to [95] for a thorough description of the Bayesian approach to blind deconvolution.

Remark

Both the original image and the PSF are often jointly estimated by minimizing an uniform objective functional [5; 82; 87; 88], whereas a few works separate the blur identification and image restoration: the original image is estimated by non-blind deconvolution using the PSF estimated by the blur identification [90–92].

In all of these existing approaches, the regularization parameters must be determined properly so that the algorithm can provide good recovered images and blurs. The parameters can be estimated by either variational Bayesian method [88] or generalized cross validation (GCV) method [83; 96]. However, all the proposed methods require expensive computational cost.

1.4.2 Parametric blind deconvolution

Necessity of parametric blind deconvolution

In several particular applications, e.g., linear motion blur [97; 98], atmospheric turbulence [99], out-of-focus blur [100], astronomical imaging [101] and fluorescence microscopy [102], the parametric forms of the PSF are either theoretically available or practically assumed [3; 103]. In this context, the PSF is completely characterized by only a few parameters [102] or at least belongs to a limited space of admissible functions [65]. The parametrized representation of the PSF can be considered as another type of regularization [65], provided that the number of the PSF parameters is not too large. Unlike the non-parametric blind deconvolution, parametric one has only a small number of PSF parameters to estimate, which dramatically reduces the degrees of freedom in the problem.

Examples in practical applications

Examples of the parametric approach can be found in the applications of deconvolution microscopy [102; 104; 105] and motion deblurring [97; 98; 106], where the analytical models of PSF are provided in the specific contexts. Besides, [3] considered the PSF in a form of low-exponent Lévy function with two parameters; [103] performed a soft combination of several typical parametric blur structures (e.g. Gaussian, uniform and linear blurs). Still, both the estimates of the original image and the PSF parameters are obtained either simultaneously [102; 105] or separately [3; 65; 97].

1.5 Objective assessment of the deconvolution quality

Evaluating the deconvolution quality is essential to compare various deconvolution algorithms or to validate a deconvolution procedure applied on real data. In this section, we discuss the most popular approaches to image quality assessment. By objective quality assessment, we mean a mathematical measure that quantifies the similarity between the deconvolved image $\widehat{\mathbf{x}} \in \mathbb{R}^N$ and the original image $\mathbf{x} \in \mathbb{R}^N$.

1.5.1 Peak Signal-to-Noise Ratio (PSNR)

The mean-squared error (MSE)

$$\mathrm{MSE} = \frac{1}{N}\|\widehat{\mathbf{x}} - \mathbf{x}\|^2 = \frac{1}{N}\sum_{n=1}^{N}(\widehat{x}_n - x_n)^2 \tag{1.10}$$

is the most widely used measure of quality. It is often normalized by the square of the maximum value of the signal and represented in a logarithmic scale yielding the popular peak signal-to-noise ratio (PSNR):

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\|\mathbf{x}\|_\infty^2}{\mathrm{MSE}}\right) \tag{1.11}$$

1.5.2 Structural Similarity Index (SSIM)

However, the MSE is considered as a poor visual quality metric, mainly due to its non-adaptivity to local signal specificities (intensity, correlation, ...) [107]. This has led some researchers to design new measures of quality that better correlate with human visual perception. The most popular representative of these quality metrics is probably the structural similarity index (SSIM) introduced in [108]. Taking into account three types of similarities: the luminance similarity, the contrast similarity and the structural similarity, the SSIM between two images $\mathbf{x}$ and $\mathbf{y}$ is defined as[2]:

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_{\mathbf{x}}\mu_{\mathbf{y}} + C_1)(2\sigma_{\mathbf{xy}} + C_2)}{(\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2 + C_1)(\mu_{\mathbf{x}}^2 + \mu_{\mathbf{y}}^2 + C_2)} \tag{1.12}$$

where all the notations are described in [108].

1.6 Thesis contributions

The general goal of this thesis is to design non-blind/blind deconvolution algorithms based on unbiased risk estimates that satisfy the following requirements:

- Efficiency: the proposed solutions have to be competitive with the state-of-the-art deconvolution methods, with respect to an objective measure of quality (e.g. PSNR and SSIM).
- Computational cost: the proposed algorithms should be faster than (or at least as fast as) the most efficient deconvolution algorithms available.

Along the lines leading to the fulfilment of the above requirements, this thesis brings the following contributions.

1.6.1 Theoretical contributions

Non-blind deconvolution

To come up with efficient deconvolution algorithms, we need to achieve optimal performances with respect to an objective measure of quality — an estimate of the MSE. For AWGN, such an estimate has been established by Charles Stein and is nowadays known as Stein's unbiased risk estimate (SURE) [109]. Since it involves only the measurements, we can minimize SURE instead of MSE in practice. In this work, we extended SURE theory to non-blind deconvolution. We derived several expressions of SURE bound to linear or non-linear type of processings. We applied the linear combination of multiple Wiener filterings with different regularization parameters: automatically finding a

[2] A Matlab code that computes the SSIM map and SSIM index is available at: http://www.ece.uwaterloo.ca/~z70wang/research/ssim/.

good combination by SURE minimization enabled us to avoid to manually adjust single regularization parameter.

PSF estimation for blind deconvolution

To achieve superior performance to other state-of-the-art blind deconvolution methods, we need to find an accurate estimate of PSF, which is essential to blind deconvolution. In this work, we proposed a novel criterion for PSF estimation — blur-SURE, which is an unbiased estimate of a filtered version of standard MSE. We theoretically and experimentally proved that the PSF can be accurately estimated from the measurements only, by minimizing the blur-SURE. We exemplified the blur-SURE framework with several function forms of PSF in different applications.

1.6.2 Algorithmic contributions

Non-blind deconvolution

Most of the existing deconvolution algorithms require the optimization of several nonlinear parameters and/or involve sophisticated redundant transforms, see [49; 61–63] for example. Consequently, their computational burden is usually quite heavy.

To achieve a high-quality non-blind deconvolution in a low computation time, we apply the *linear expansion of thresholds* (LET), which was first proposed in [110], to the deconvolution problem. The optimal linear parameters of this expansion are then the ones that minimize an unbiased estimate of the MSE. Thanks to the quadratic form of such MSE estimates, the whole parameters optimization boils down to solving a linear system of equations, which is very fast and exact. In this LET strategy, fast deconvolution algorithms can then be devised.

PSF estimation for blind deconvolution

Most of the existing non-deconvolution methods applied alternating minimization (AM) algorithm between the original image and the PSF, see [5; 82; 83; 88; 91] for example. Each iteration requires expensive computational cost, and the AM algorithm often has slow convergence.

To achieve a high-quality blind deconvolution in a low computation time, we devised several fast algorithms for the blur-SURE minimization in different applications. Then, we apply the proposed non-blind deconvolution algorithm with the estimated PSF.

1.7 Organization

The Part I of this thesis is dedicated to non-blind deconvolution problem. Although numerous works have been already undertaken in this (yet still active) field of research, we present in Chapter 2 a SURE-LET framework for general linear distortion model, not limited to deconvolution.

In Chapter 3, we apply the SURE-LET strategy to linear or non-linear processings, and arbitrary (including orthonormal and redundant) transform. We derive several SURE expressions in each context.

In Part II of this thesis, we consider the blind deconvolution problem. We present in Chapter 4 a generic blur-SURE framework for PSF estimation.

Chapters 5 and 6 are dedicated to several applications of parametric blind deconvolution. Experiments show that the estimated PSF parameter by the blur-SURE minimization is very close to the true one, and the deconvolution performance is already competitive with most state-of-the-art approaches.

Our concluding remarks, as well as some perspectives, are finally reported in Chapter 7.

Part I

The SURE-LET Approach to Non-blind Deconvolution

Chapter 2

The SURE-LET Framework for Deconvolution

2.1 Motivations

Given some blurred and noisy measurements, the goal of deconvolution task is to find the best estimate of the underlying original signal. The key point is then to quantify how close to the original signal a given estimate is. We have already discussed in Chapter 1 several objective measures of quality assessment. From a practical point of view, the mean-squared error (MSE) clearly emerges as the best candidate, due to its appealing mathematical properties (quadratic, symmetric, differentiable, invariant to unitary transforms). In this chapter, we thus propose to consider the minimization of the MSE as the driving principle of our deconvolution strategy.

2.2 Related work

This framework that will be introduced in this chapter is essentially based on Stein's unbiased risk estimate (SURE). In this section, we thus briefly review the SURE-based approaches designed for denoising and deconvolution.

SURE — an unbiased estimate of the MSE under additive Gaussian noise assumption — was first proposed by C. Stein in [109], and has been recently revitalized for denoising purpose. The most straightforward way of distinguishing information from noise in the wavelet domain consists in thresholding the wavelet coefficients. Donoho and Johnstone proposed *SUREshrink*, which chooses the optimal threshold value T by minimizing SURE [13]. Chang *et al.* proposed *BayesShrink* to derive their threshold in a Bayesian framework, assuming a generalized Gaussian distribution for the wavelet coefficients [111]. For computational efficiency, *Linear expansion of thresholds* (LET), a new parametrized form of thresholding function in wavelet domain, was proposed in [1; 110]. Aimed at the SURE minimization, the SURE-LET approach reduces the denoising problem to solving a simple linear system of equations.

More recently, SURE has been extended to any linear restoration problem [112–115]. Since SURE depends on the observed data only, it can be practically used for the following purposes:

1. To optimize the parameters of the processing, e.g., the Tikhonov regularization parameter [115] or the optimal parameters involved in the non-local means (NLM) denoising technique [116];

2. To monitor the PSNR improvement during the iterations of IST algorithms, without referring to the original unknown data, e.g., automatic determination of the number of iterations [112], optimal update of the parameters at each iteration [115];

3. As a minimization criterion for designing denoising/deconvolution algorithms expressed as LET structure, the so-called SURE-LET approach [1; 113; 117]. It requires the basic structure or parametric form of the estimator to be determined in advance [1; 113; 115].

Our work follows the last point. Similar to the work of [113], we further extend the SURE-LET approach to regularized inversion followed by denoising. Note that [113] empirically chooses the regularization parameter of the regularized inverse filter. The main contribution of our work, also the major difference from [113], is that besides applying a linear combination of non-linear thresholding functions for the denoising step as in [113], we also linearly parametrize the regularized inversion into a number of basic Wiener filters with different (but fixed) regularization parameters. Instead of optimizing or empirically choosing non-linear regularization parameters [61; 62; 113; 115], the proposed multi-Wiener SURE-LET approach finds the best linear combination of multiple Wiener deconvolutions. The linear parametrization reduces the deconvolution problem to solving a linear system of equations. Its solution, i.e., the weights in the linear combination, automatically constitutes the minimum MSE/SURE combination as final estimate. As a result, the proposed deconvolution algorithm is fast and of low computational complexity[1].

[1] A demo software is available at www.ee.cuhk.edu.hk/~tblu/demos/.

2.3 Problem statement

Consider the linear model

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{b}, \tag{2.1}$$

where $\mathbf{y} = [y_n]_{n=1\ldots N} \in \mathbb{R}^{N\times 1}$ is a distorted observation of the original (unknown) data $\mathbf{x} = [x_n]_{n=1\ldots N} \in \mathbb{R}^{N\times 1}$. The matrix $\mathbf{H} \in \mathbb{R}^{N\times N}$ implements a linear distortion[2], while the vector $\mathbf{b} = [b_n]_{n=1\ldots N} \in \mathbb{R}^{N\times 1}$ is a zero-mean additive Gaussian noise corruption with covariance matrix $\mathbf{C} \succ 0$. Our objective is to find an estimate $\widehat{\mathbf{x}} = \mathbf{f}(\mathbf{y}) = [f_n(\mathbf{y})]_{n=1\ldots N}$ that minimizes the mean squared error (MSE) defined by:

$$\text{MSE} = \frac{1}{N}\|\widehat{\mathbf{x}} - \mathbf{x}\|^2 = \frac{1}{N}\sum_{n=1}^{N}(\widehat{x}_n - x_n)^2 \tag{2.2}$$

Here, we would like to insist that the estimate $\widehat{\mathbf{x}} = \mathbf{f}(\mathbf{y})$ is only the outcome of the processing $\mathbf{f}$. A key feature of our approach is to estimate the function (or processing) $\mathbf{f} : \mathbb{R}^N \to \mathbb{R}^N$ that transforms $\mathbf{y}$ into $\widehat{\mathbf{x}}$, rather than the estimate $\widehat{\mathbf{x}}$ itself.

2.4 Stein's Unbiased Risk Estimate (SURE) for deconvolution

Since we do not have access to the original signal $\mathbf{x}$, we cannot compute the above *Oracle* MSE as Eq.(2.2). However, without any assumptions on the original data $\mathbf{x}$, we will see that it is possible to replace this quantity by an unbiased estimate which is a function of $\mathbf{y}$ only. This has an important consequence: contrary to what is frequently done in the deconvolution literature (Bayesian approaches), the original signal is not modelled as a random process in our framework (we do not even require $\mathbf{x}$ to belong to a specific class of signals). Thus, the observed randomness of the observed data $\mathbf{y}$ only originates from the Gaussian noise $\mathbf{b}$.

2.4.1 Original SURE

Based on the linear model shown as Eq.(2.1) and the additive Gaussian noise assumption, the MSE can be replaced by a statistical estimate, SURE, involving the measurements $\mathbf{y}$ only. This is summarized in the following theorem [113; 115; 118].

[2] The theory that follows can be applied to rectangular matrices, but for the sake of simplicity, we restrict ourselves to square matrices.

Theorem 2.1 Given the linear model shown as Eq.(2.1) with any invertible matrix $\mathbf{H} \in \mathbb{R}^{N\times N}$, the following random variable

$$\epsilon = \frac{1}{N}\left(\|\mathbf{f}(\mathbf{y})\|^2 - 2\mathbf{y}^{\mathrm{T}}\mathbf{H}^{-\mathrm{T}}\mathbf{f}(\mathbf{y}) + 2\mathrm{div}_{\mathbf{y}}\left\{\mathbf{C}\mathbf{H}^{-\mathrm{T}}\mathbf{f}(\mathbf{y})\right\}\right) + \frac{1}{N}\|\mathbf{x}\|^2$$

is an unbiased estimator of the MSE defined in Eq.(2.2); i.e.,

$$\mathscr{E}\{\epsilon\} = \frac{1}{N}\mathscr{E}\left\{\|\mathbf{f}(\mathbf{y}) - \mathbf{x}\|^2\right\}.$$

where $\mathscr{E}\{\cdot\}$ denotes the mathematical expectation operator.

See the proof in Appendix A.1. In particular, if the Gaussian noise $\mathbf{b}$ is independent and identically distributed (i.i.d.) with variance σ^2, SURE becomes

$$\epsilon = \frac{1}{N}\left(\|\mathbf{f}(\mathbf{y})\|^2 - 2\mathbf{y}^{\mathrm{T}}\mathbf{H}^{-\mathrm{T}}\mathbf{f}(\mathbf{y}) + 2\sigma^2\mathrm{div}_{\mathbf{y}}\left\{\mathbf{H}^{-\mathrm{T}}\mathbf{f}(\mathbf{y})\right\}\right) + \frac{1}{N}\|\mathbf{x}\|^2, \tag{2.3}$$

which results from specifying the covariance matrix $\mathbf{C} = \sigma^2\mathbf{I}$.

Corollary 2.2 In the particular case of an i.i.d. Gaussian noise with variance σ^2, if the processing $\mathbf{f}(\cdot)$ can be expressed as $\mathbf{f}(\mathbf{y}) = \mathbf{H}^{\mathrm{T}}\tilde{\mathbf{f}}(\mathbf{y})$, then SURE becomes

$$\epsilon = \frac{1}{N}\left(\|\mathbf{H}^{\mathrm{T}}\tilde{\mathbf{f}}(\mathbf{y})\|^2 - 2\mathbf{y}^{\mathrm{T}}\tilde{\mathbf{f}}(\mathbf{y}) + 2\sigma^2\mathrm{div}_{\mathbf{y}}\left\{\tilde{\mathbf{f}}(\mathbf{y})\right\}\right) + \frac{1}{N}\|\mathbf{x}\|^2. \tag{2.4}$$

2.4.2 Regularized approximation of SURE

If the matrix $\mathbf{H}$ is ill-conditioned or singular, ϵ in Eq.(2.3) fails to be a reliable estimate of the MSE. To keep the stability of ϵ, we may intuitively replace the unstable inverse $\mathbf{H}^{-1}$ by a Tikhonov regularized inverse [26]

$$\mathbf{H}_{\beta}^{-1} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \beta\mathbf{S}^{\mathrm{T}}\mathbf{S}\right)^{-1}\mathbf{H}^{\mathrm{T}}, \tag{2.5}$$

for some parameter $\beta > 0$ and matrix $\mathbf{S} \in \mathbb{R}^{N\times N}$. We can then approximate the original SURE as

$$\epsilon_{\beta} = \frac{1}{N}\left(\|\mathbf{f}(\mathbf{y})\|^2 - 2\mathbf{y}^{\mathrm{T}}\mathbf{H}_{\beta}^{-\mathrm{T}}\mathbf{f}(\mathbf{y}) + 2\sigma^2\mathrm{div}\left\{\mathbf{H}_{\beta}^{-\mathrm{T}}\mathbf{f}(\mathrm{y})\right\}\right) + \frac{1}{N}\|\mathbf{H}_{\beta}^{-1}\mathbf{H}\mathbf{x}\|^2. \tag{2.6}$$

This approximated estimator of the MSE turns out to be an unbiased estimate of $\frac{1}{N}\|\mathbf{f}(\mathbf{y}) - \mathbf{H}_{\beta}^{-1}\mathbf{H}\mathbf{x}\|^2$ — a modified MSE (see the proof in Appendix A.2). Hence, if

$\mathbf{H}_\beta^{-1}\mathbf{Hx}$ is close enough to $\mathbf{x}$ (hypothesis on $\mathbf{x}$), then, ϵ_β, the approximated SURE, is a good estimate of the MSE.

In imaging application, a good choice for $\mathbf{S}$ is a high-pass operator. A reasonable value for β should achieve a good balance between the approximation accuracy and the stability of ϵ_β; i.e., β should be neither too large, to avoid significant loss of high-frequency features, nor too small, to keep the stability of ϵ_β. We also define a regularized inverse for $\mathbf{S}$ as $\mathbf{S}_\beta^{-1} = \beta\left(\mathbf{H}^\mathrm{T}\mathbf{H} + \beta\mathbf{S}^\mathrm{T}\mathbf{S}\right)^{-1}\mathbf{S}^\mathrm{T}$, which leads to the following decomposition of the identity matrix: $\mathbf{H}_\beta^{-1}\mathbf{H} + \mathbf{S}_\beta^{-1}\mathbf{S} = \mathbf{I}$.

2.5 The SURE-LET approach

In practice, we will minimize the approximated SURE shown as Eq.(2.6) instead of the actual MSE in Eq.(2.2). The next question that naturally arises is: how to choose the function $\mathbf{f}$, such that $\mathbf{f}(\mathbf{y})$ is sufficiently close to $\mathbf{x}$? Here, we adopt the *linear expansion of thresholds* (LET) method [1; 110; 117], which parametrizes $\mathbf{f}(\mathbf{y})$ as a linear combination of a small number of pre-defined basic functions (or processing) $\mathbf{f}_k$ for $k = 1, 2, ..., K$; i.e.,

$$\mathbf{f}(\mathbf{y}) = \sum_{k=1}^{K} a_k \mathbf{f}_k(\mathbf{y}) = \underbrace{[\mathbf{f}_1(\mathbf{y})\;\mathbf{f}_2(\mathbf{y})\;\ldots\;\mathbf{f}_K(\mathbf{y})]}_{\mathbf{f}(\mathbf{y})} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_K \end{bmatrix}}_{\mathbf{a}} = \mathbf{Fa}, \tag{2.7}$$

where $K \ll N$ is the number of linear coefficients a_k. Here, $\mathbf{f}(\mathbf{y}) = \mathbf{Fa}$ is a shorthand matrix notation to outline the linearity of the representation. The deconvolution problem then amounts to find the linear coefficients a_k that minimize ϵ_β. This parametrization dramatically reduces the size of the estimation problem from N number of pixels to K number of basis functions. Note that the LET decomposition shown as Eq.(2.7) does not imply any hypothesis on $\mathbf{x}$ itself. Yet, a "bad" basis of processing $\mathbf{f}_k$ will obviously lead to poorer deconvolution performances compared to a "good" basis.

Substituting Eq.(2.7) into Eq.(2.6), and performing the differentiation of ϵ_β with respect to a_k for all $k = 1, 2, ..., K$, we obtain the linear system of equations

$$\sum_{k'=1}^{K} \underbrace{\frac{1}{N}\mathbf{f}_k^{\mathrm{T}}(\mathbf{y})\mathbf{f}_{k'}(\mathbf{y})}_{\mathbf{M}_{k,k'}} a_{k'} - \underbrace{\frac{1}{N}\left(\mathbf{y}^{\mathrm{T}}\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{f}_k(\mathbf{y}) - \sigma^2 \mathrm{div}\left\{\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{f}_k(\mathbf{y})\right\}\right)}_{c_k} = 0, \tag{2.8}$$

for $k = 1, 2, ..., K$. These equations can be summarized in matrix form as $\mathbf{Ma} = \mathbf{c}$, where $\mathbf{M} = [\mathbf{M}_{k,k'}]_{k,k'=1,2,...,K}$ and $\mathbf{c} = [c_1\ c_2\ ...\ c_K]^{\mathrm{T}}$. Due to the possible singularity of $\mathbf{M}$, we solve the following ℓ_2-regularized linear system of equations

$$(\mathbf{M} + \mu\mathbf{I})\mathbf{a} = \mathbf{c} \quad \Leftrightarrow \quad \mathbf{a} = (\mathbf{M} + \mu\mathbf{I})^{-1}\mathbf{c} \tag{2.9}$$

instead of Eq.(2.8), where $\mu > 0$ is a regularization parameter.

It is also worth noting that the corresponding MSE minimization leads to solving $\mathbf{Ma}_{\mathrm{MSE}} = \mathbf{F}^{\mathrm{T}}\mathbf{x}$, with the solution, namely *MSE-LET*, serving as a counterpart to SURE-LET. The accuracy of using the SURE-LET estimate can then be verified by comparing it with *MSE-LET*.

Notice that this SURE-LET framework can be used for any linear model as Eq.(2.1), not limited to convolution. What this approach suggests is that the practitioner may choose at will a set of K different algorithms (ideally with complementary processing behaviors of $\mathbf{f}_k(\mathbf{y})$) and optimize a weighting of these algorithms to get the best of them at once.

In the following chapter, we will further discuss the application of the SURE-LET paradigm to different basic processings $\mathbf{f}_k(\mathbf{y})$, in the context of deconvolution.

2.6 Summary

In this chapter, we have devised a general procedure (SURE-LET) for the linear model shown as Eq.(2.1), not limited to convolution. The general procedure of the SURE-LET framework is summarized as Algorithm 2.1.

There are three key ingredients of the proposed approach listed below:

1. An unbiased estimate of the MSE, known as SURE. In particular, this statistical quantity only depends on the observed measurements and can thus be computed in practice. In contrast to the popular Bayesian approach, no prior on the unknown original data is required to derive SURE.

Algorithm 2.1: General SURE-LET Framework

```
Input: distortion matrix $\mathbf{H}$ and observed data $\mathbf{y}$;
Output: optimal estimate $\widehat{\mathbf{x}}$ in MSE sense
begin
    1. construct elementary functions $\mathbf{f}_k$ for $k = 1, 2, ..., K$;
    2. compute $\mathbf{M}$ and $\mathbf{c}$ according to Eq.(2.8);
    3. solve the linear system of equation $\mathbf{Ma} = \mathbf{c}$ according to Eq.(2.9);
    4. perform linear combination $\mathbf{f}(\mathbf{y}) = \sum_{k=1}^{K} a_k \mathbf{f}_k(\mathbf{y})$ as Eq.(2.7), to obtain
    the final estimate $\widehat{\mathbf{x}}$.
end
```

2. An unbiased estimate of a modified MSE, known as regularized approximation of the SURE. There is only a weak hypothesis on the original data for this approximation.

3. A *linear expansion of thresholds* (LET): in image deconvolution, SURE was usually used for optimizing one or several non-linear parameters. LET was first proposed for image denoising. In this chapter, we have extended the LET strategy to the general linear model, by building a linearly parametrized estimator which offers more flexibility than the standard processing functions. Thanks to the quadratic form of SURE, we have shown that the optimal (in the minimum SURE sense) parameters are simply the solution of a linear system of equations. From a computational point of view, this makes the SURE-LET strategy particularly attractive.

Chapter 3

Multi-Wiener SURE-LET Approach

3.1 Problem statement

Chapter 2 has described the SURE-LET framework for inverse problems with general linear distortion model. Figure 3.1 shows the image degradation of Eq.(2.1) in the context of image deconvolution, where the distortion matrix $\mathbf{H}$ denotes the convolution operation, which results in blurred observations.

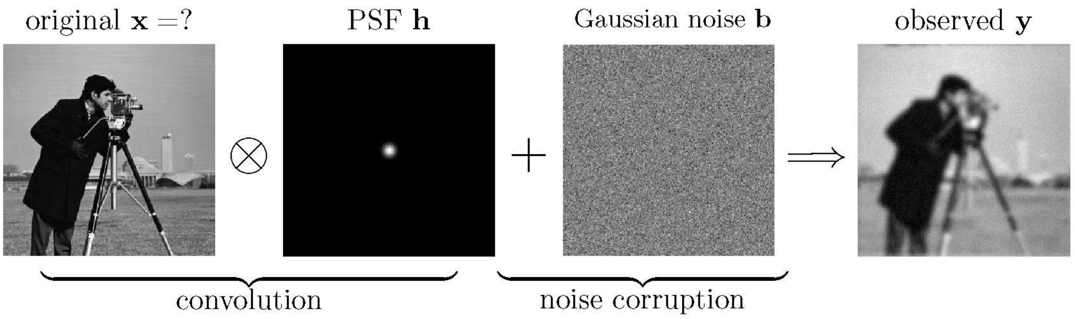


Figure 3.1: *The model of image degradation shown as Eq.(2.1): convolution (blurring effect) followed by noise corruption.*

This chapter exemplifies the SURE-LET framework in the scenario of Figure 3.1, by showing how to construct the elementary processing $\mathbf{f}_k$ in Eq.(2.7).

3.2 Linear deconvolution: multi-Wiener filtering

Beginning with the simplest case, we choose each $\mathbf{f}_k(\mathbf{y})$ in Eq.(2.7) to be a Wiener filter with a given regularization parameter λ_k:

$$\mathbf{f}_k(\mathbf{y}) = \underbrace{\left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda_k \mathbf{S}^{\mathrm{T}}\mathbf{S}\right)^{-1}\mathbf{H}^{\mathrm{T}}}_{\mathbf{H}_{\lambda_k}^{-1}}\mathbf{y}, \text{ for } k = 1, 2, ..., K. \tag{3.1}$$

In the standard case of a shift-invariant convolution operator, the matrix products with $\mathbf{H}$ or $\mathbf{H}^{\mathrm{T}}$ are commutative. Each elementary processing $\mathbf{f}_k(\mathbf{y})$ can then be written as $\mathbf{f}_k(\mathbf{y}) = \mathbf{H}^{\mathrm{T}}\tilde{\mathbf{f}}_k(\mathbf{y})$ where $\tilde{\mathbf{f}}_k(\mathbf{y}) = \left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda_k\mathbf{S}^{\mathrm{T}}\mathbf{S}\right)^{-1}\mathbf{y}$ and Corollary 2.2 applies. Consequently, we do not need to use the regularized SURE ϵ_β defined in Eq.(2.6). The SURE-LET method consists in finding the minimum SURE/MSE combination of these Wiener filters, which is achieved by solving Eq.(2.8). Finally, the SURE of Eq.(2.4) is simplified as:

$$\begin{aligned}\epsilon = {} & \frac{1}{N}\left\|\sum_{k=1}^{K} a_k \mathbf{H}_{\lambda_k}^{-1}\mathbf{y}\right\|^2 - \frac{2}{N}\sum_{k=1}^{K} a_k \mathbf{y}^{\mathrm{T}}\left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda_k\mathbf{S}^{\mathrm{T}}\mathbf{S}\right)^{-1}\mathbf{y} \\ & + \frac{2}{N}\sigma^2\sum_{k=1}^{K} a_k \mathrm{Tr}\left\{\left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda_k\mathbf{S}^{\mathrm{T}}\mathbf{S}\right)^{-1}\right\} + \frac{1}{N}\|\mathbf{x}\|^2. \end{aligned} \tag{3.2}$$

Note that $\mathrm{div}_{\mathbf{y}}\{\tilde{\mathbf{f}}_k(\mathbf{y})\} = \mathrm{Tr}\left\{\left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda_k\mathbf{S}^{\mathrm{T}}\mathbf{S}\right)^{-1}\right\}$. Thus, the SURE minimization over linear coefficients a_k reduces to solving the linear system of equations $\mathbf{Ma} = \mathbf{c}$ shown as Eq.(2.8), with k-th element of $\mathbf{c}$ given as:

$$c_k = \frac{1}{N}\left[\mathbf{y}^{\mathrm{T}}\left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda_k\mathbf{S}^{\mathrm{T}}\mathbf{S}\right)^{-1}\mathbf{y} - \sigma^2\mathrm{Tr}\left\{\left(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda_k\mathbf{S}^{\mathrm{T}}\mathbf{S}\right)^{-1}\right\}\right]. \tag{3.3}$$

where the trace can be efficiently computed in Fourier domain, provided both $\mathbf{H}$ and $\mathbf{S}$ are convolution matrices.

The flowchart of the SURE-optimized multi-Wiener deconvolution is shown in Fig. 3.2. We observe that different values of λ_k capture different details and features of the image, and the optimal linear coefficients a_k produce the combined estimate with the best balance between noise reduction and edge preservation. The key advantage of the multi-Wiener SURE optimization is that it avoids the empirical adjustment of a unique non-linear regularization parameter λ, contrary to [61; 62; 115].

3.3 SURE-LET in orthonormal wavelet representation

3.3.1 Mathematical formulation

We now present the construction of elementary functions $\mathbf{f}_k(\mathbf{y})$ for the non-linear deconvolution approach which consists of multi-Wiener filtering followed by pointwise thresholding in orthonormal wavelet domain[1]. For LET strategy shown as Eq.(2.7),

[1] By thresholding, we denote any non-linear operation (smooth or non-smooth).

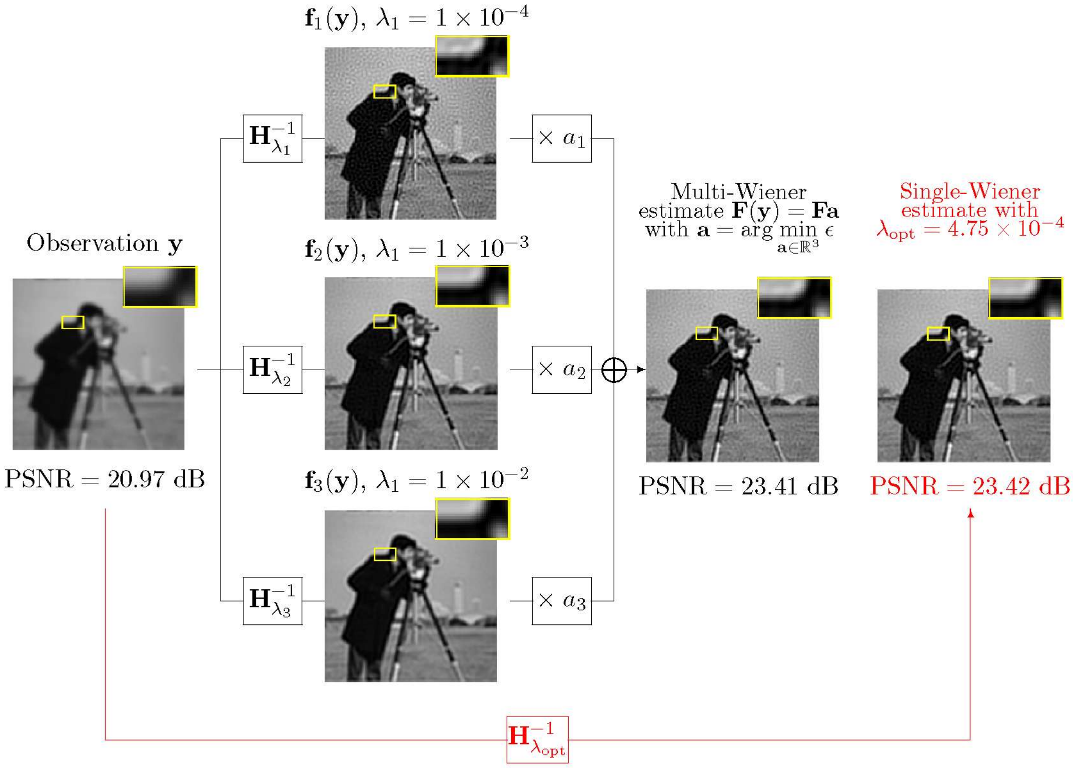

$\mathbf{f}_1(\mathbf{y})$, $\lambda_1 = 1 \times 10^{-4}$
$\mathbf{f}_2(\mathbf{y})$, $\lambda_1 = 1 \times 10^{-3}$
$\mathbf{f}_3(\mathbf{y})$, $\lambda_1 = 1 \times 10^{-2}$
Observation $\mathbf{y}$
PSNR = 20.97 dB
$\mathbf{H}^{-1}_{\lambda_1}$
$\mathbf{H}^{-1}_{\lambda_2}$
$\mathbf{H}^{-1}_{\lambda_3}$
$\times\, a_1$
$\times\, a_2$
$\times\, a_3$
Multi-Wiener estimate $\mathbf{F}(\mathbf{y}) = \mathbf{Fa}$ with $\mathbf{a} = \arg\min_{\mathbf{a}\in\mathbb{R}^3} \epsilon$
PSNR = 23.41 dB
Single-Wiener estimate with $\lambda_{\text{opt}} = 4.75 \times 10^{-4}$
PSNR = 23.42 dB
$\mathbf{H}^{-1}_{\lambda_{\text{opt}}}$

Figure 3.2: *Example of linearly combining three Wiener filters with regularization parameters λ_1, λ_2 and λ_3, balanced by their weights $a_1 = 0.43$, $a_2 = 0.61$ and $a_3 = -0.04$ given by Eq.(2.8). The linearly-combined estimate is equivalent to a single optimal Wiener filtering in terms of PSNR.*

besides from the multiple Wiener filtering, we also linearly parametrize the pointwise thresholding functions: $\theta(\cdot) = \sum_{l=1}^{L} a_l \theta_l(\cdot)$, where $(\cdot)$ denotes any wavelet coefficient to be thresholded.

For orthonormal wavelet transform, we use $\mathbf{W}$ and $\mathbf{W}^{-1}$ to denote wavelet decomposition and reconstruction. Here, $\mathbf{W}$ is an orthonormal matrix, such that $\mathbf{W}^{-1} = \mathbf{W}^{\mathrm{T}}$. Our processing $\mathbf{f}(\mathrm{y})$ can be mathematically expressed as:

$$\mathbf{f}(\mathbf{y}) = \sum_{m=1}^{M} \sum_{l=1}^{L} a_{m,l} \underbrace{\mathbf{W}^{-1}\theta_l \Big(\overbrace{\mathbf{W}\mathbf{H}^{-1}_{\lambda_m}\mathrm{y}}^{\mathbf{w}_m}\Big)}_{\mathbf{f}_{m,l}(\mathbf{y})} \tag{3.4}$$

where $\mathbf{H}^{-1}_{\lambda_m}$ is defined in Eq.(3.1), $\theta_l(\cdot)$ denotes elementary pointwise thresholding function, which is applied to high-pass subbands only ($\theta_l(\cdot)$ is identity function for low-pass subbands), $\mathbf{w}_m$ is the wavelet coefficients to be thresholded, for m-th Wiener filtering. Since Eq.(3.4) cannot be rewritten in the form of $\mathbf{f}(\mathbf{y}) = \mathbf{H}^{\mathrm{T}}\tilde{\mathbf{f}}(\mathbf{y})$, we need to use the regularized SURE given by Eq.(2.6), which is further simplified as:

$$\begin{aligned}\epsilon_\beta =& \frac{1}{N}\Big\|\sum_{m,l} a_{m,l}\theta_l(\mathbf{w}_m)\Big\|^2 - \frac{2}{N}\sum_{m,l} a_{m,l}\underbrace{\big(\mathbf{W}\mathbf{H}_\beta^{-1}\mathbf{y}\big)}_{\mathbf{w}_\beta}^{\mathrm{T}}\theta_l(\mathbf{w}_m)\\ &+ \frac{2}{N}\sigma^2\sum_{m,l} a_{m,l}\boldsymbol{\alpha}_m^{\mathrm{T}}\theta'_l(\mathbf{w}_m) + \frac{1}{N}\|\underbrace{\mathbf{W}\mathbf{H}_\beta^{-1}\mathbf{H}\mathbf{x}}_{\mathbf{w}_\mathbf{x}}\|^2, \end{aligned} \tag{3.5}$$

where the vector $\boldsymbol{\alpha}_m$ consists of diagonal element of matrix $\mathbf{P}_m = \mathbf{W}\mathbf{H}_{\lambda_m}^{-1}\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{W}^{-1}$, denoted as $\boldsymbol{\alpha}_m = \mathrm{diag}\{\mathbf{P}_m\}$, $\theta'_l(\cdot)$ denotes the derivative of the thresholding function. The derivation of this expression is given in Appendix A.3.

3.3.2 SURE minimization in orthonormal wavelet domain

During the derivation of Eq.(3.5), the Euclidian norm is preserved in the transform domain, thanks to the orthonormal transformation. Thus, the regularized SURE given by Eq.(3.5) can be completely expressed in orthonormal wavelet domain. Using the notation of wavelet subbands, this expression becomes:

$$\begin{aligned}\epsilon_\beta =& \frac{1}{N}\Big\|\sum_{j=1}^{J}\sum_{m,l} a_{m,l}^j\theta_l(\mathbf{w}_m^j)\Big\|^2 + \frac{1}{N}\Big\|\sum_{m} a_m^{J+1}\mathbf{w}_m^{J+1}\Big\|^2\\ &- \frac{2}{N}\sum_{j=1}^{J}\sum_{m,l} a_{m,l}^j\underbrace{\big(\mathbf{W}^j\mathbf{H}_\beta^{-1}\mathbf{y}\big)}_{\mathbf{w}_\beta^j}^{\mathrm{T}}\theta_l(\mathbf{w}_m^j) - \frac{2}{N}\sum_{m} a_m^{J+1}\underbrace{\big(\mathbf{W}^{J+1}\mathbf{H}_\beta^{-1}\mathbf{y}\big)}_{\mathbf{w}_\beta^{J+1}}^{\mathrm{T}}\mathbf{w}_m^{J+1}\\ &+ \frac{2}{N}\sigma^2\sum_{j=1}^{J}\sum_{m,l} a_{m,l}^j\boldsymbol{\alpha}_m^{j\,\mathrm{T}}\theta'_l(\mathbf{w}_m^j) + \frac{2}{N}\sigma^2\sum_{m} a_m^{J+1}\boldsymbol{\alpha}_m^{J+1\,\mathrm{T}}\mathbf{1}\\ &+ \frac{1}{N}\Big\|\sum_{j=1}^{J}\underbrace{\mathbf{W}^j\mathbf{H}_\beta^{-1}\mathbf{H}\mathbf{x}}_{\mathbf{w}_\mathbf{x}^j}\Big\|^2 + \frac{1}{N}\Big\|\underbrace{\mathbf{W}^{J+1}\mathbf{H}_\beta^{-1}\mathbf{H}\mathbf{x}}_{\mathbf{w}_\mathbf{x}^{J+1}}\Big\|^2, \end{aligned} \tag{3.6}$$

where $\mathbf{W}^j$ denotes j-th wavelet high-pass subband for $j = 1, 2, ..., J$, $\mathbf{w}^j$ denotes the wavelet coefficients in j-th subband, $\boldsymbol{\alpha}_m^j = \mathrm{diag}\{\mathbf{W}^j\mathbf{H}_{\lambda_m}^{-1}\mathbf{H}_\beta^{-1}\mathbf{W}^{j\,\mathrm{T}}\}$ in j-th subband for $j = 1, 2, ..., J, J+1$, the index $J+1$ denotes the low-pass subband. Thus, the regularized SURE ϵ_β can be decomposed as the sum of each wavelet subband ϵ_β^j for

$j = 1, 2, ..., J, J+1$:

$$\epsilon_\beta = \sum_{j=1}^{J} \underbrace{\left\{ \frac{1}{N} \left\| \sum_{m,l} a_{m,l}^j \theta_l(\mathbf{w}_m^j) \right\|^2 - \frac{2}{N} \sum_{m,l} a_{m,l}^j \left({\mathbf{w}_\beta^j}^{\mathrm{T}} \theta_l(\mathbf{w}_m^j) - \sigma^2 {\boldsymbol{\alpha}_m^j}^{\mathrm{T}} \theta_l'(\mathbf{w}_m^j) \right) + \frac{1}{N} \left\| \mathbf{w}_{\mathbf{x}}^j \right\|^2 \right\}}_{\epsilon_\beta^j}$$

$$+ \underbrace{\frac{1}{N} \left\| \sum_m a_m^{J+1} \mathbf{w}_m^{J+1} \right\|^2 - \frac{2}{N} \sum_m a_m^{J+1} \left({\mathbf{w}_\beta^{J+1}}^{\mathrm{T}} \mathbf{w}_m^{J+1} - \sigma^2 {\boldsymbol{\alpha}_m^{J+1}}^{\mathrm{T}} \mathbf{1} \right) + \frac{1}{N} \left\| \mathbf{w}_{\mathbf{x}}^{J+1} \right\|^2}_{\epsilon_\beta^{J+1}} \tag{3.7}$$

Finally, the SURE minimization can be performed in each wavelet subband, i.e. equivalent to solving $J+1$ linear systems of equations in each subband, separately:

$$\min_{a_{m,l}} \epsilon_\beta \Longleftrightarrow \min_{a_{m,l}^j} \epsilon_\beta^j \Longrightarrow \mathbf{M}^j \mathbf{a}^j = \mathbf{c}^j, \quad \text{for } j = 1, 2, ..., J, J+1 \tag{3.8}$$

- for $j = 1, 2, ..., J$: the matrix $\mathbf{M}^j = {\mathbf{F}^j}^{\mathrm{T}} \mathbf{F}^j \in \mathbb{R}^{ML \times ML}$ with

$$\mathbf{F}^j = \left[\theta_1(\mathbf{w}_1^j), ..., \theta_L(\mathbf{w}_1^j), ..., \theta_1(\mathbf{w}_M^j), ..., \theta_L(\mathbf{w}_M^j) \right] \in \mathbb{R}^{N^j \times ML}$$

  where N^j is the dimension of j-th wavelet subband. $\mathbf{a}^j = \left[a_{1,1}^j, ..., a_{l,m}^j, ..., a_{L,M}^j \right]^{\mathrm{T}} \in \mathbb{R}^{ML \times 1}$, $\mathbf{c}^j = \left[c_{1,1}^j, ..., c_{l,m}^j, ..., c_{L,M}^j \right]^{\mathrm{T}} \in \mathbb{R}^{ML \times 1}$ with the element $c_{l,m}^j$ given by:

$$c_{l,m}^j = {\mathbf{w}_\beta^j}^{\mathrm{T}} \theta_l(\mathbf{w}_m^j) - \sigma^2 {\boldsymbol{\alpha}_m^j}^{\mathrm{T}} \theta_l'(\mathbf{w}_m^j)$$

- for $j = J+1$: the matrix $\mathbf{M}^{J+1} = {\mathbf{F}^{J+1}}^{\mathrm{T}} \mathbf{F}^{J+1} \in \mathbb{R}^{M \times M}$ with

$$\mathbf{F}^{J+1} = \left[\mathbf{w}_1^{J+1}, \mathbf{w}_2^{J+1}, ..., \mathbf{w}_M^{J+1} \right] \in \mathbb{R}^{N^{J+1} \times M}$$

  where N^{J+1} is the dimension of $(J+1)$-th low-pass subband. $\mathbf{a}^{J+1} = \left[a_m^{J+1} \right]_{m \in [1;M]}$, $\mathbf{c}^{J+1} = \left[c_m^{J+1} \right]_{m \in [1;M]}$ with the element c_m^{J+1} given by:

$$c_m^j = {\mathbf{w}_\beta^{J+1}}^{\mathrm{T}} \mathbf{w}_m^{J+1} - \sigma^2 {\boldsymbol{\alpha}_m^{J+1}}^{\mathrm{T}} \mathbf{1}$$

To summarize, Figure 3.3 describes the whole procedure of SURE-LET approach in orthonormal wavelet representation.

3.3.3 Computational issues

Computation of noise variance in particular wavelet subband

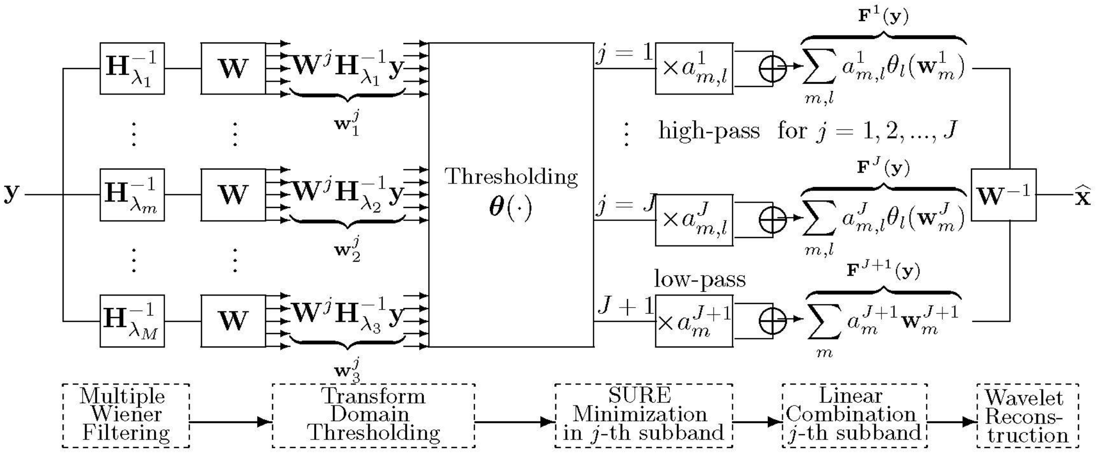

$\mathbf{F}^1(\mathbf{y})$
$\mathbf{y}$
$\mathbf{H}_{\lambda_1}^{-1}$
$\mathbf{H}_{\lambda_m}^{-1}$
$\mathbf{H}_{\lambda_M}^{-1}$
$\mathbf{W}$
$\mathbf{W}^j\mathbf{H}_{\lambda_1}^{-1}\mathbf{y}$
$\mathbf{w}_1^j$
$\mathbf{W}^j\mathbf{H}_{\lambda_2}^{-1}\mathbf{y}$
$\mathbf{w}_2^j$
$\mathbf{W}^j\mathbf{H}_{\lambda_3}^{-1}\mathbf{y}$
$\mathbf{w}_3^j$
Thresholding
$\boldsymbol{\theta}(\cdot)$
$j=1$
$\times a_{m,l}^1$
$\sum_{m,l} a_{m,l}^1\theta_l(\mathbf{w}_m^1)$
high-pass for $j = 1, 2, ..., J$
$\mathbf{F}^J(\mathbf{y})$
$j=J$
$\times a_{m,l}^J$
$\sum_{m,l} a_{m,l}^J\theta_l(\mathbf{w}_m^J)$
low-pass
$\mathbf{F}^{J+1}(\mathbf{y})$
$J+1$
$\times a_m^{J+1}$
$\sum_m a_m^{J+1}\mathbf{w}_m^{J+1}$
$\mathbf{W}^{-1}$
$\widehat{\mathbf{x}}$
Multiple Wiener Filtering
Transform Domain Thresholding
SURE Minimization in j-th subband
Linear Combination j-th subband
Wavelet Recons-truction

Figure 3.3: *Flowchart of the SURE-LET approach in orthonormal wavelet representation: SURE minimization in wavelet domain.*

Generally, the thresholding function $\theta_l(\cdot)$ has a threshold T as its parameter, and the parameter is often adaptive to the noise level of particular wavelet subband for better performance. For example, one may choose the following thresholding functions used in [1]:

$$\begin{cases} \theta_1(w^j) = w^j \\ \theta_2(w^j) = w^j\Big[1 - \exp\big(-\frac{w^j}{T^j}\big)^2\Big] \end{cases} \tag{3.9}$$

where $T^j = 3\sigma_j$, σ_j^2 is the noise variance of j-th wavelet subband. More specifically, for the m-th Wiener deconvolver and j-th wavelet subband, the noise becomes $\mathbf{b}_m^j = \mathbf{W}^j\mathbf{H}_{\lambda_m}^{-1}\mathbf{b}$. The variance denoted by $\sigma_{m,j}^2$ is theoretically given by:

$$\sigma_{m,j}^2 = \frac{1}{N^j}\mathscr{E}\Big\{{\mathbf{b}_m^j}^{\mathrm{T}}\mathbf{b}_m^j\Big\} = \frac{1}{N^j}\mathscr{E}\Big\{\mathbf{b}^{\mathrm{T}}\mathbf{H}_{\lambda_m}^{-\mathrm{T}}{\mathbf{W}^j}^{\mathrm{T}}\mathbf{W}^j\mathbf{H}_{\lambda_m}^{-1}\mathbf{b}\Big\} = \mathrm{Tr}\Big(\mathbf{W}^j\mathbf{H}_{\lambda_m}^{-1}\mathbf{H}_{\lambda_m}^{-\mathrm{T}}{\mathbf{W}^j}^{\mathrm{T}}\Big) \tag{3.10}$$

Because of the huge size of the matrices, Eq. (3.10) cannot be computed in practice. Due to the easily accessible actions by these matrices on any vector, Monte-Carlo technique provides a good approximation of $\sigma_{m,j}^2$, which is described below as Algorithm

3.1.

Algorithm 3.1: Monte-Carlo algorithm to evaluate $\sigma_{m,j}^2$ in Eq.(3.10)

```
Input: noise variance $\sigma^2$, Wiener filtering $\mathbf{H}_{\lambda_m}^{-1}$ and orthonormal wavelet
       transform $\mathbf{W}^j$;
Output: noise variance $\sigma_{m,j}^2$
for $i \leftarrow 1$ to $I$ do
    1. Generate a Gaussian white noise $\mathbf{b}^i \in \mathbb{R}^N$ with variance $\sigma^2$;
    2. Apply the operator $\mathbf{W}^j\mathbf{H}_{\lambda_m}^{-1}$ to $\mathbf{b}^i$ to get the vector
       $\mathbf{b}_{j,m}^i = \mathbf{W}^j\mathbf{H}_{\lambda_m}^{-1}\mathbf{b}^i \in \mathbb{R}^{N_j}$;
    3. Compute the noise variance according to $\sigma_{m,j}^{i2} = \frac{1}{N_j}\mathbf{b}_{m,j}^{i\mathrm{T}}\mathbf{b}_{m,j}^i$.
end
```

Finally, $\sigma_{m,j}^2$ is obtained by averaging the I randomizations:

$$\sigma_{m,j}^2 = \frac{1}{I}\sum_{i=1}^{I}\sigma_{m,j}^{i2} \tag{3.11}$$

Computation of diagonal element $\boldsymbol{\alpha}_m^j$

M-C simulation is still applied to evaluate diagonal element $\boldsymbol{\alpha}_m^j = \mathrm{diag}\{\mathbf{W}^j\mathbf{H}_{\lambda_m}^{-1}\mathbf{H}_{\beta}^{-1}\mathbf{W}^{j\mathrm{T}}\}$, summarized below as Algorithm 3.2.

Algorithm 3.2: Monte-Carlo algorithm to evaluate $\boldsymbol{\alpha}_m^j$ in Eq.(3.6)

```
Input: wavelet transformations $\mathbf{W}^j$ and $\mathbf{W}^{j\mathrm{T}}$, Wiener filterings $\mathbf{H}_{\lambda_m}^{-1}$ and $\mathbf{H}_{\beta}^{-1}$;
Output: diagonal element $\boldsymbol{\alpha}_m^j = \mathrm{diag}\left\{\mathbf{W}^j\mathbf{H}_{\lambda_m}^{-1}\mathbf{H}_{\beta}^{-1}\mathbf{W}^{j\mathrm{T}}\right\}$
for $i \leftarrow 1$ to $I$ do
    1. Generate a normalized Gaussian white noise $\mathbf{b}^i \in \mathbb{R}^{N^j}$;
    2. Apply the operator $\mathbf{P}_m^j = \mathbf{W}^j\mathbf{H}_{\lambda_m}^{-1}\mathbf{H}_{\beta}^{-\mathrm{T}}\mathbf{W}^{j\mathrm{T}}$ to $\mathbf{b}^i$ to get the vector
       $\mathbf{b}_{m,j}^i = \mathbf{P}_m^j\mathbf{b}^i \in \mathbb{R}^{N^j}$;
    3. Compute the component-wise product of $\mathbf{b}_{m,j}^i$ with $\mathbf{b}^i$ to get a vector $\mathbf{u}^i$,
       which is also $\mathrm{diag}\{\mathbf{b}_{m,j}^i\mathbf{b}^{i\mathrm{T}}\}$.
end
```

Finally, α_m^j is obtained by averaging the I randomizations:

$$\alpha_m^j = \frac{1}{I}\sum_{i=1}^{I}\mathbf{u}^i \tag{3.12}$$

See [1] for the proof.

3.4 SURE-LET approach for redundant wavelet representation

We now present the construction of elementary functions $\mathbf{f}_k(\mathbf{y})$ for the non-linear deconvolution approach which consists of multi-Wiener filtering followed by redundant transform-domain thresholding. Both steps are linearly parametrized using the LET strategy shown as Eq.(2.7).

An illustrative description of the proposed non-linear deconvolution approach with redundant wavelet transform is shown in Figures 3.4 and 3.5. In Figure 3.4, the matrices $\mathbf{D}$ and $\mathbf{R}$ represent a pair of linear decomposition/reconstruction that satisfies the perfect reconstruction condition $\mathbf{RD} = \mathbf{I}$. Typically, $\mathbf{D}$ and $\mathbf{R}$ implement a $(J+1)$-band filterbank of undecimated filters. They are structured as $\mathbf{D} = \left[\mathbf{D}_1^{\mathrm{T}}\, \mathbf{D}_2^{\mathrm{T}} \ldots \mathbf{D}_J^{\mathrm{T}}\, \mathbf{D}_{J+1}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{(J+1)N\times N}$ and $\mathbf{R} = [\mathbf{R}_1\, \mathbf{R}_2 \ldots \mathbf{R}_J\, \mathbf{R}_{J+1}] \in \mathbb{R}^{N\times(J+1)N}$, where $\mathbf{D}_j \in \mathbb{R}^{N\times N}$ and $\mathbf{R}_j \in \mathbb{R}^{N\times N}$ for $j = 1, 2, \ldots, J, J+1$. We consider $(J+1)$ bands for the convenience of the future discussion. Mathematically, the flowchart of Figure 3.4 can be described by the following function:

$$\mathbf{f}(\mathbf{y}) = \mathbf{R}\boldsymbol{\theta}\left(\mathbf{D}\mathbf{H}_\lambda^{-1}\mathbf{y}\right). \tag{3.13}$$

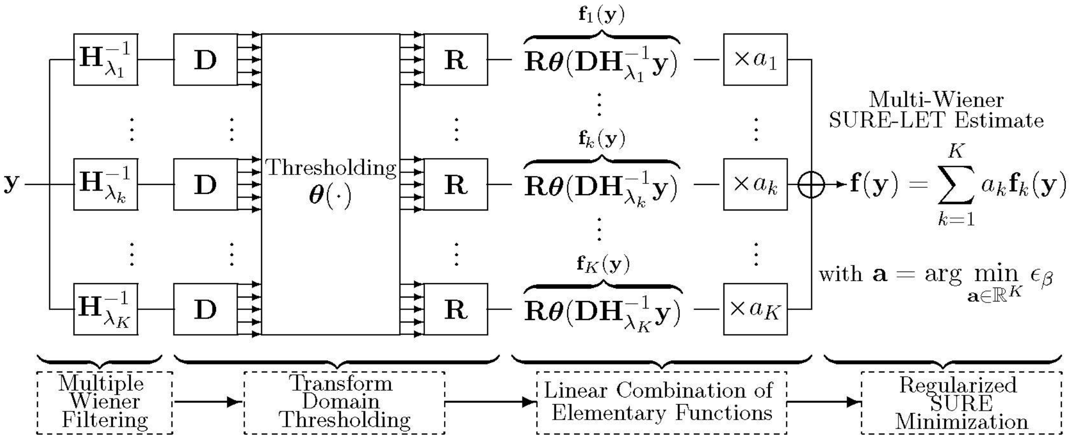

$\mathbf{f}_1(\mathbf{y})$
$\mathbf{H}^{-1}_{\lambda_1}$
D
R
$\mathbf{R}\boldsymbol{\theta}(\mathbf{D}\mathbf{H}^{-1}_{\lambda_1}\mathbf{y})$
$\times a_1$
Multi-Wiener
SURE-LET Estimate
$\mathbf{f}_k(\mathbf{y})$
y
$\mathbf{H}^{-1}_{\lambda_k}$
D
Thresholding
$\boldsymbol{\theta}(\cdot)$
R
$\mathbf{R}\boldsymbol{\theta}(\mathbf{D}\mathbf{H}^{-1}_{\lambda_k}\mathbf{y})$
$\times a_k$
$\mathbf{f}(\mathbf{y}) = \sum_{k=1}^{K} a_k\mathbf{f}_k(\mathbf{y})$
$\mathbf{f}_K(\mathbf{y})$
$\mathbf{H}^{-1}_{\lambda_K}$
D
R
$\mathbf{R}\boldsymbol{\theta}(\mathbf{D}\mathbf{H}^{-1}_{\lambda_K}\mathbf{y})$
$\times a_K$
with $\mathbf{a} = \arg\min_{\mathbf{a}\in\mathbb{R}^K} \epsilon_\beta$
Multiple Wiener Filtering
Transform Domain Thresholding
Linear Combination of Elementary Functions
Regularized SURE Minimization

Figure 3.4: *Flowchart of the proposed multi-Wiener SURE-LET approach with redundant wavelet transform.*

Corollary 3.1 Given the processing $\mathbf{f}(\cdot)$ defined by Eq.(3.13) and considering a pointwise function $\boldsymbol{\theta}(\cdot)$, the regularized approximation of SURE ϵ_β introduced in Eq.(2.6) can be further derived as

$$\epsilon_\beta = \frac{1}{N}\left(\|\mathbf{f}(\mathbf{y})\|^2 - 2\mathbf{y}^{\mathrm{T}}\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{f}(\mathbf{y}) + 2\sigma^2\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\theta}'(\mathbf{w})\right) + \frac{1}{N}\left\|\mathbf{H}_\beta^{-1}\mathbf{H}\mathbf{x}\right\|^2. \tag{3.14}$$

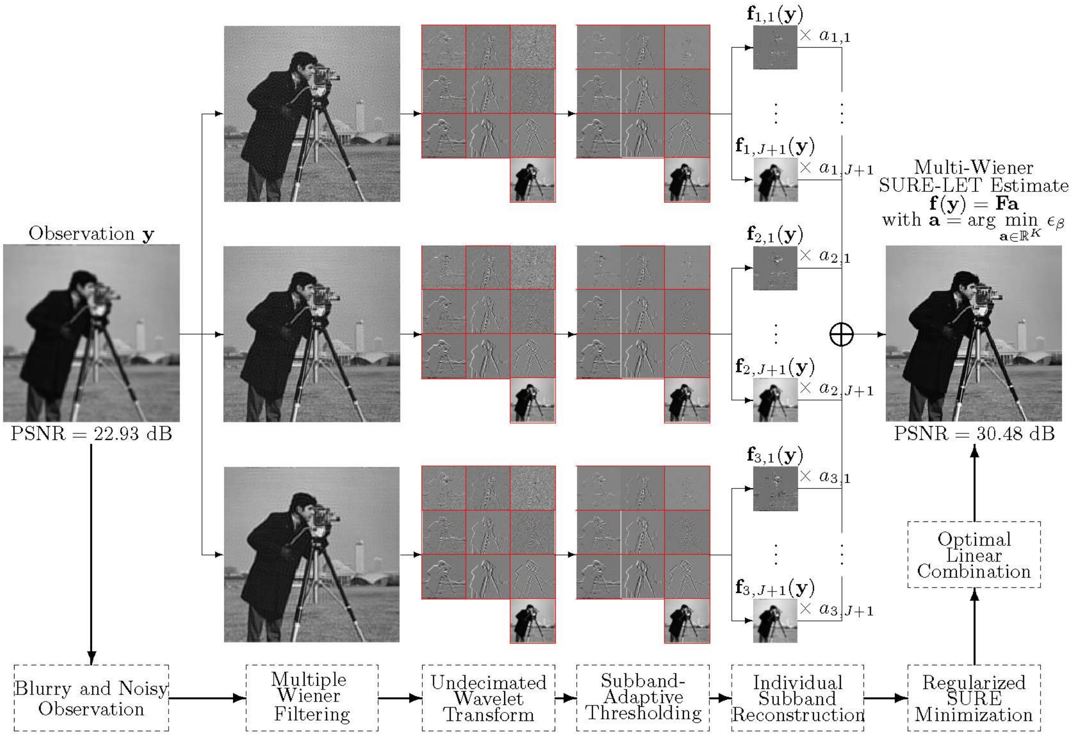

$\mathbf{f}_{1,1}(\mathbf{y})$
$\times\, a_{1,1}$
$\mathbf{f}_{1,J+1}(\mathbf{y})$
$\times\, a_{1,J+1}$
Multi-Wiener
SURE-LET Estimate
$\mathbf{f}(\mathbf{y}) = \mathbf{F}\mathbf{a}$
with $\mathbf{a} = \arg\min_{\mathbf{a}\in\mathbb{R}^K} \epsilon_\beta$
Observation $\mathbf{y}$
$\mathbf{f}_{2,1}(\mathbf{y})$
$\times\, a_{2,1}$
$\mathbf{f}_{2,J+1}(\mathbf{y})$
$\times\, a_{2,J+1}$
PSNR = 22.93 dB
PSNR = 30.48 dB
$\mathbf{f}_{3,1}(\mathbf{y})$
$\times\, a_{3,1}$
$\mathbf{f}_{3,J+1}(\mathbf{y})$
$\times\, a_{3,J+1}$
Optimal Linear Combination
Blurry and Noisy Observation
Multiple Wiener Filtering
Undecimated Wavelet Transform
Subband-Adaptive Thresholding
Individual Subband Reconstruction
Regularized SURE Minimization

Figure 3.5: *Illustration of the proposed multi-Wiener SURE-LET approach with redundant wavelet transform. Note that (1) The undecimated wavelet subbands and their processed reconstructions are displayed in reduced size for convenience; (2) The reconstruction $\mathbf{R}_j$ is performed to the specific j-th subband only, by setting all the other subbands to zero [1]; (3) The thresholding function $\boldsymbol{\theta}(\cdot)$ can also be linearly parametrized as $\boldsymbol{\theta}(\cdot) = \sum_{l=1}^{L} a_l \boldsymbol{\theta}_l(\cdot)$.*

where the vector $\mathbf{w} = \mathbf{D}\mathbf{H}_\lambda^{-1}\mathbf{y} = \left[\mathbf{w}_1^{\mathrm{T}}\ \mathbf{w}_2^{\mathrm{T}}\ ...\ \mathbf{w}_J^{\mathrm{T}}\ \mathbf{w}_{J+1}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{R}^{(J+1)N}$, with $\mathbf{w}_j = \mathbf{D}_j\mathbf{H}_\lambda^{-1}\mathbf{y} \in \mathbb{R}^N$ for $j = 1, 2, ..., J+1$, denotes the transform coefficients. The vector $\boldsymbol{\theta}'(\mathbf{w}) = [\theta_n'(w_n)]_{n\in[1;JN]}$ represents the first derivative of the pointwise function $\boldsymbol{\theta}$, which is applied to high-pass subbands only. The vector $\boldsymbol{\alpha} \in \mathbb{R}^{(J+1)N}$ is defined by

$$\boldsymbol{\alpha} = \mathrm{diag}\left\{\mathbf{D}\mathbf{H}_\lambda^{-1}\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{R}\right\} = [\mathbf{P}_{n,n}]_{n\in[1;(J+1)N]}$$

where $\mathbf{P} = \mathbf{D}\mathbf{H}_\lambda^{-1}\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{R} \in \mathbb{R}^{(J+1)N\times(J+1)N}$. Consequently, the vector $\boldsymbol{\alpha}$ is structured as $\boldsymbol{\alpha} = \left[\boldsymbol{\alpha}_1^{\mathrm{T}}\ \boldsymbol{\alpha}_2^{\mathrm{T}}\ ...\ \boldsymbol{\alpha}_J^{\mathrm{T}}\ \boldsymbol{\alpha}_{J+1}^{\mathrm{T}}\right]^{\mathrm{T}}$, where

$$\boldsymbol{\alpha}_j = \mathrm{diag}\left\{\mathbf{D}_j\mathbf{H}_\lambda^{-1}\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{R}_j\right\} \in \mathbb{R}^N \tag{3.15}$$

for $j = 1, 2, ..., J+1$.

The proof of Corollary 3.1 is very similar to that of Appendix A.3 and Corollary 1 in [1]. We thus omit it here.

Note that for the final deconvolution results of this thesis, we confine ourselves to redundant wavelet transforms, due to their superior denoising abilities. Since the applied redundant wavelet transform is not orthogonal, the minimization of ϵ_β has to be performed in the image domain to ensure a global minimum MSE optimality.

Suppose that the decomposition $\mathbf{D}$ produces J highpass subbands and one lowpass subband (indexed as the $(J+1)$-th subband) that is not thresholded, then $\mathbf{f}(\mathbf{y})$ of Eq.(3.13) can be linearly parametrized as

$$\mathbf{f}(\mathbf{y}) = \underbrace{\sum_{m=1}^{M}\sum_{l=1}^{L}\sum_{j=1}^{J} a_{m,l,j}\mathbf{R}_j\boldsymbol{\theta}_l\left(\mathbf{w}_{m,j}\right)}_{\text{highpass subbands}} + \underbrace{\sum_{m=1}^{M} a_{m,J+1}\mathbf{R}_{J+1}\mathbf{w}_{m,J+1}}_{\text{lowpass subband}}, \tag{3.16}$$

where M denotes the number of Wiener filters, L the number of elementary pointwise thresholding functions, and J the number of highpass wavelet subbands (typically $J = 9$ for three decomposition levels).

As Eq.(3.16) shows, $K = M \cdot J \cdot L + M$ weights $a_{m,l,j}$ and L thresholding function need to be determined. The weights $a_{m,l,j}$ are obtained by minimizing the regularized approximation of SURE ϵ_β, which boils down to solving a linear system of K equations shown as Eq.(2.9).

3.5 Computational aspects

Similar to Section 3.3, the proposed method of Section 3.4 also requires to compute several Wiener filters (3.1), the subband-dependent noise variances σ_j^2 and the subband-dependent vectors $\boldsymbol{\alpha}_j$ (3.15), listed here below:

- Wiener filter: $\mathbf{H}_\lambda^{-1} = \left(\mathbf{H}^\mathrm{T}\mathbf{H} + \lambda\mathbf{S}^\mathrm{T}\mathbf{S}\right)^{-1}\mathbf{H}^\mathrm{T}$;
- Variance σ_j^2 of the colored noise $\mathbf{b}_j = \mathbf{D}_j\mathbf{H}_\lambda^{-1}\mathbf{b}$, theoretically given as

$$\sigma_j^2 = \frac{1}{N_j}\mathscr{E}\{\mathbf{b}_j^\mathrm{T}\mathbf{b}_j\} = \frac{1}{N_j}\mathscr{E}\{\mathbf{b}^\mathrm{T}\mathbf{H}_\lambda^{-\mathrm{T}}\mathbf{D}_j^\mathrm{T}\mathbf{D}_j\mathbf{H}_\lambda^{-1}\mathbf{b}\} = \frac{\sigma^2}{N_j}\mathrm{Tr}\left(\mathbf{H}_\lambda^{-\mathrm{T}}\mathbf{D}_j^\mathrm{T}\mathbf{D}_j\mathbf{H}_\lambda^{-1}\right) \tag{3.17}$$

- Vector $\boldsymbol{\alpha}_j = \mathrm{diag}\left\{\mathbf{D}_j\mathbf{H}_\lambda^{-1}\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{R}_j\right\}$.

In this section, we show how to perform these computations analytically in the Fourier domain, for redundant wavelet transforms. For the sake of brevity of the discussion and formulation, we stick to one-dimensional case. The 2-D computations can then be straightforwardly deduced. We define the coefficients of a 1-D filter as $g(n)$, the coefficients of the input of the 1-D filter as $x(n)$ and the coefficients of the output of the 1-D filter by $y(n)$, for $n = 0, 1, ..., N-1$. All these coefficients are assumed to be zero outside the domain $[0, N-1]$. Note that the implementation of the proposed approach needs the specification of boundary conditions. The most commonly used are periodic extensions. Yet, periodic extensions may generate unwanted artifacts (discontinuity at the boundaries), so symmetric boundary extensions are often preferred in practice [44; 119; 120]. In the next two sections, we discuss the computations of the above mentioned terms for these two particular choices of boundary conditions.

3.5.1 Periodic boundary extensions

Circular convolution

Under periodic boundary conditions, i.e. $x(k) = x(k + n_0N)$ for any $n_0 \in \mathbb{Z}$ and $k = 0, 1, ..., N-1$, the convolution is circular and reads as:

$$\begin{aligned} y(n) &= \sum_{k\in\mathbb{Z}} g(n-k)x(k) \\ &= \sum_{n_0\in\mathbb{Z}} \sum_{k=0}^{N-1} g(n-k+n_0N)\, \underbrace{x(k+n_0N)}_{=x(k)} \\ &= \sum_{k=0}^{N-1} \underbrace{\sum_{n_0\in\mathbb{Z}} g(n-k+n_0N)}_{g_N(n-k)}\, x(k) \end{aligned} \tag{3.18}$$

where

$$g_N(n) = \sum_{n_0\in\mathbb{Z}} g(n+n_0N) \tag{3.19}$$

Consequently, the matrices $\mathbf{H}_\lambda^{-1}$, $\mathbf{D}_j$, $\mathbf{R}_j$, and $\mathbf{H}_\beta^{-\mathrm{T}}$ are all circulant. They can thus be efficiently computed by discrete Fourier transform (DFT) with period N [121], where the Fourier coefficients of $x(n)$ and $g(n)$ are given as:

$$\begin{aligned}
X\big(e^{j\frac{2\pi kn}{N}}\big) &\overset{\text{def}}{=\!=} X(k) = \sum_{n\in\mathbb{Z}} x(n)e^{-j\frac{2\pi kn}{N}} \\
&= \sum_{n_0\in\mathbb{Z}}\sum_{n=0}^{N-1} x(n+n_0N)e^{-j\frac{2\pi k(n+n_0N)}{N}} \\
&= \sum_{n=0}^{N-1}\underbrace{\sum_{n_0\in\mathbb{Z}} x(n+n_0N)}_{x_N(n)}e^{-j\frac{2\pi kn}{N}} = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{N}}
\end{aligned} \tag{3.20}$$

$$\begin{aligned}
G\big(e^{j\frac{2\pi kn}{N}}\big) &\overset{\text{def}}{=\!=} G(k) = \sum_{n\in\mathbb{Z}} g(n)e^{-j\frac{2\pi kn}{N}} \\
&= \sum_{n_0\in\mathbb{Z}}\sum_{n=0}^{N-1} g(n+n_0N)e^{-j\frac{2\pi k(n+n_0N)}{N}} \\
&= \sum_{n=0}^{N-1}\underbrace{\sum_{n_0\in\mathbb{Z}} g(n+n_0N)}_{g_N(n)}e^{-j\frac{2\pi kn}{N}}
\end{aligned} \tag{3.21}$$

Note that $x_N(n) = x(n)$ for $n = 0, 1, ..., N-1$ due to the periodicity of $x(n)$ and the interval $[0, N-1]$.

Analytical computation of σ_j^2 given in Eq.(3.17) (see also [113])

Theorem 3.2 Under periodic boundary conditions, given a zero-mean white Gaussian noise $b(n)$ with variance σ^2 and a filter $g(n)$, the variance of $p(n) = \sum_{k\in\mathbb{Z}} g(n-k)b(k)$ is given by

$$\mathscr{E}\left\{p(n)^2\right\} = \sigma^2\sum_{n=0}^{N-1} g_N(n)^2 = \frac{\sigma^2}{N}\sum_{k=0}^{N-1}\left|G(e^{j\frac{2\pi k}{N}})\right|^2$$

for $n = 0, 1, ..., N-1$, where $g_N(n)$ and $G\big(e^{j\frac{2\pi k}{N}}\big)$ are given by Eq.(3.19) and Eq. (3.21)).

Proof The second equality comes from Parseval's theorem [121]. The i.i.d. condition of the zero-mean white Gaussian noise $b(n)$ implies that

$$\mathscr{E}\left\{b(n)b(n')\right\} = \sigma^2\delta(n-n') = \begin{cases} \sigma^2 & \text{, for } n = n' \\ 0 & \text{, otherwise} \end{cases}. \tag{3.22}$$

Thus, we obtain that the variance of the filtered noise $p(n)$ is:

$$\begin{aligned}
\mathscr{E}\left\{p(n)^2\right\} &= \sum_{k\in\mathbb{Z}}\sum_{k'\in\mathbb{Z}} g_N(n-k)g_N(n-k')\mathscr{E}\left\{b(k)b(k')\right\} \\
&\overset{(3.22)}{=} \sigma^2 \sum_{k\in\mathbb{Z}} g(n-k)^2 = \sigma^2 \sum_{n_0\in\mathbb{Z}}\sum_{k=0}^{N-1} g(n-k+n_0N)^2 \\
&\overset{(3.19)}{=} \sigma^2 \sum_{k=0}^{N-1} \underbrace{\sum_{n_0\in\mathbb{Z}} g(n-k+n_0N)^2}_{g_N(n-k)} \overset{n'=n-k}{=\!=\!=} \sigma^2 \sum_{n'=n}^{n-(N-1)} g_N(n')^2 \\
&= \sigma^2 \sum_{n=0}^{N-1} g_N(n)^2 \quad \text{(by periodicity } g_N(n) = g_N(n+n_0N))
\end{aligned}$$

■

Note that Theorem 3.2 is a straightforward application of the result obtained for filtered wide-sense stationary process (see for instance Theorem 3.14 of [122]). It is restated and proved here for later comparison with that obtained under symmetric boundary conditions.

Analytical computation of $\boldsymbol{\alpha_j}$ given by Eq.(3.15)

Theorem 3.3 Under periodic boundary conditions, the vector formed by the diagonal elements of the convolution matrix $\mathbf{G} \in \mathbb{R}^{N\times N}$ is given by

$$\mathrm{diag}\{\mathbf{G}\} = g_N(0)\cdot\mathbf{1} = \left(\frac{1}{N}\sum_{k=0}^{N-1} G\big(e^{j\frac{2\pi k}{N}}\big)\right)\cdot\mathbf{1}$$

where $g_N(n)$ given by Eq.(3.19) is the filter implemented by $\mathbf{G}$, $G\big(e^{j\frac{2\pi k}{N}}\big)$ is given by Eq.(3.21).

Proof The second equality is essentially the inverse DFT. In matrix form, each element of the diagonal of $\mathbf{G}$ can be obtained as $\mathbf{e}_{k'}^{\mathrm{T}}\mathbf{G}\mathbf{e}_{k'}$, where the indicator vector $\mathbf{e}_{k'}$ is defined as $\mathbf{e}_{k'} = [0\ ...\ 0\ 1\ 0\ ...\ 0]^{\mathrm{T}} \in \mathbb{R}^N$ with k'-th element 1. In convolution form, it reads as:

$$\mathbf{e}_{k'}^{\mathrm{T}}\mathbf{G}\mathbf{e}_{k'} = \sum_{k=1}^{N}\delta(k-k')\left(\sum_{n=1}^{N} g_N(k-n)\delta(n-k')\right) = g_N(0)$$

where $\delta(k-k')$ is defined as in Eq.(3.22). ■

We conclude the proof by using the fact that, under periodic boundary conditions, the convolution matrix $\mathbf{G}$ is circulant and thus its diagonal elements are all equal.

3.5.2 Symmetric convolution

The term "symmetric convolution" was first introduced by S. Martucci in [123], to describe the convolution under symmetric boundary conditions. In [123], he discussed 64 possible types of symmetric convolutions, depending on half-point or whole-point symmetry. The symmetric convolution can be performed by DCT or Types I-VIII DST, according to specific types of symmetry. Tables 3.1 and 3.2 summarize the filtering under various symmetric situations.

The following subsections will describe how to perform these computations in half-point and whole-point symmetric boundary extensions.

3.5.3 Half-point symmetric boundary extensions

Formulation for half-point symmetric convolution

Lemma 3.4 Given a signal $x(n)$ and filter $g(n)$, under half-point symmetric boundary condition of $x(n)$, convolution is defined as

$$y(n) = \sum_{k=0}^{N-1} \big(g_{2N}(n-k) + g_{2N}(n+k+1)\big)x(k) \tag{3.23}$$

for $n = 0, 1, ..., N-1$, where $g_{2N}(n) = \sum_{n_0 \in \mathbb{Z}} g(n + 2n_0 N)$

Proof Since $x(n) = x(n + n_0 \cdot 2N)$ for $\forall n_0 \in \mathbb{Z}$ and $x(n) = x(2N-1-n)$ by half-point symmetry, we have:

$$\begin{aligned}
y(n) &= \sum_{k\in\mathbb{Z}} g(n-k)x(k) = \sum_{k=0}^{2N-1} \Bigg(\underbrace{\sum_{n_0\in\mathbb{Z}} g\big(n-k+n_0\cdot 2N\big)}_{g_{2N}(n-k)}\Bigg)x(k) \\
&= \sum_{k=0}^{N-1} g_{2N}(n-k)x(k) + \sum_{k=N}^{2N-1} g_{2N}(n-k)x(k) \\
&= \sum_{k=0}^{N-1} g_{2N}(n-k)x(k) + \sum_{k'=0}^{N-1} \underbrace{g_{2N}\big(n-(2N-1-k')\big)}_{g_{2N}(n+k'+1)\ (\text{period}=2N)}\ \underbrace{x(2N-1-k')}_{x(k')\ (\text{by symmetry})} \\
&= \sum_{k=0}^{N-1} g_{2N}(n-k)x(k) + \sum_{k=0}^{N-1} g_{2N}(n+k+1)x(k)
\end{aligned}$$

input signal $x(n)$			filter $g(n)$		output signal $y(n)$	
Symmetry	Transform	Period	Symmetry	Transform	Symmetry	Transform
• even around $n = 0$ • even around $n = N - 1$	N-dim DCT-I	$2N - 2$	$g(n) = g(-n)$	N-dim DCT-I	• even around $n = 0$ • even around $n = N - 1$	N-dim DCT-I
			$g(n) = -g(-n)$	$(N-2)$-dim DST-I	• odd around $n = 0$ • odd around $n = N - 1$	$(N-2)$-dim DST-I
			$g(n) = g(-1-n)$	$(N-1)$-dim DCT-II	• even around $n = -\frac{1}{2}$ • even around $n = N - \frac{3}{2}$	$(N-1)$-dim DCT-II
			$g(n) = -g(-1-n)$	$(N-1)$-dim DST-II	• odd around $n = -\frac{1}{2}$ • odd around $n = N - \frac{3}{2}$	$(N-1)$-dim DST-II
• even around $n = -\frac{1}{2}$ • even around $n = N - \frac{1}{2}$	N-dim DCT-II	$2N$	$g(n) = g(-n)$	$(N+1)$-dim DCT-I	• even around $n = -\frac{1}{2}$ • even around $n = N - \frac{1}{2}$	N-dim DCT-II
			$g(n) = -g(-n)$	$(N-1)$-dim DST-I	• odd around $n = -\frac{1}{2}$ • odd around $n = N - \frac{1}{2}$	N-dim DST-II
			$g(n) = g(1-n)$	N-dim DCT-II	• even around $n = 0$ • even around $n = N$	$(N+1)$-dim DCT-I
			$g(n) = -g(-1-n)$	N-dim DST-II	• odd around $n = -1$ • odd around $n = N - 1$	$(N-1)$-dim DST-I
• even around $n = 0$ • odd around $n = N$	N-dim DCT-III	$4N$	$g(n) = g(-n)$	N-dim DCT-III	• even around $n = 0$ • odd around $n = N$	N-dim DCT-III
			$g(n) = -g(-n)$	N-dim DST-III	• odd around $n = 0$ • even around $n = N$	N-dim DST-III
			$g(n) = g(-1-n)$	N-dim DCT-IV	• even around $n = -\frac{1}{2}$ • odd around $n = N - \frac{1}{2}$	N-dim DCT-IV
			$g(n) = -g(-1-n)$	N-dim DST-IV	• odd around $n = -\frac{1}{2}$ • even around $n = N - \frac{1}{2}$	N-dim DST-IV
• even around $n = -\frac{1}{2}$ • odd around $n = N - \frac{1}{2}$	N-dim DCT-IV	$4N$	$g(n) = g(-n)$	N-dim DCT-III	• even around $n = -\frac{1}{2}$ • odd around $n = N - \frac{1}{2}$	N-dim DCT-IV
			$g(n) = -g(-n)$	N-dim DST-III	• odd around $n = -\frac{1}{2}$ • even around $n = N - \frac{1}{2}$	N-dim DST-IV
			$g(n) = g(1-n)$	N-dim DCT-IV	• even around $n = 0$ • odd around $n = N$	N-dim DCT-III
			$g(n) = -g(-1-n)$	N-dim DST-IV	• odd around $n = -1$ • even around $n = N - 1$	N-dim DST-III

Table 3.1: *Filtering* $y(n) = \sum_{k \in \mathbb{Z}} g(n-k)y(k)$ *under symmetric extension (I)*

input signal $x(n)$			filter $g(n)$		output signal $y(n)$	
Symmetry	Transform	Period	Symmetry	Transform	Symmetry	Transform
• odd around $n=-1$ • odd around $n=N$	N-dim DST-I	$2N+2$	$g(n)=g(-n)$	$(N+1)$-dim DCT-I	• odd around $n=-1$ • odd around $n=N$	N-dim DST-I
			$g(n)=-g(-n)$	N-dim DST-I	• even around $n=-1$ • even around $n=N$	$(N+1)$-dim DCT-I
			$g(n)=g(1-n)$	$(N+1)$-dim DCT-II	• odd around $n=-\frac{1}{2}$ • odd around $n=N+\frac{1}{2}$	$(N+1)$-dim DST-II
			$g(n)=-g(1-n)$	$(N+1)$-dim DST-II	• even around $n=-\frac{1}{2}$ • even around $n=N+\frac{1}{2}$	$(N+1)$-dim DCT-II
• odd around $n=-\frac{1}{2}$ • odd around $n=N-\frac{1}{2}$	N-dim DST-II	$2N$	$g(n)=g(-n)$	$(N+1)$-dim DCT-I	• odd around $n=-\frac{1}{2}$ • odd around $n=N-\frac{1}{2}$	N-dim DST-II
			$g(n)=-g(-n)$	$(N-1)$-dim DST-I	• even around $n=-\frac{1}{2}$ • even around $n=N-\frac{1}{2}$	N-dim DCT-II
			$g(n)=g(1-n)$	N-dim DCT-II	• odd around $n=-1$ • odd around $n=N-1$	$(N-1)$-dim DST-I
			$g(n)=-g(1-n)$	N-dim DST-II	• even around $n=0$ • even around $n=N$	$(N+1)$-dim DCT-I
• odd around $n=-1$ • even around $n=N-1$	N-dim DST-III	$4N$	$g(n)=g(-n)$	N-dim DCT-III	• odd around $n=-1$ • even around $n=N-1$	N-dim DST-III
			$g(n)=-g(-n)$	N-dim DST-III	• even around $n=-1$ • odd around $n=N-1$	N-dim DCT-III
			$g(n)=g(1-n)$	N-dim DCT-IV	• odd around $n=-\frac{1}{2}$ • even around $n=N-\frac{1}{2}$	N-dim DST-IV
			$g(n)=-g(1-n)$	N-dim DST-IV	• even around $n=-\frac{1}{2}$ • odd around $n=N-\frac{1}{2}$	N-dim DCT-IV
• odd around $n=-\frac{1}{2}$ • even around $n=N-\frac{1}{2}$	N-dim DST-IV	$4N$	$g(n)=g(-n)$	N-dim DCT-III	• odd around $n=-\frac{1}{2}$ • even around $n=N-\frac{1}{2}$	N-dim DST-IV
			$g(n)=-g(-n)$	N-dim DST-III	• even around $n=-\frac{1}{2}$ • odd around $n=N-\frac{1}{2}$	N-dim DCT-IV
			$g(n)=g(1-n)$	N-dim DCT-IV	• odd around $n=-1$ • even around $n=N-1$	N-dim DST-III
			$g(n)=-g(1-n)$	N-dim DST-IV	• even around $n=0$ • odd around $n=N$	N-dim DCT-III

Table 3.2: *Filtering $y(n)=\sum_{k\in\mathbb{Z}} g(n-k)y(k)$ under symmetric extension (II)*

Merging the two terms completes the proof. ■

Computation of filtering via transform domain

The general computation of filtering is summarized as three steps:

1. According to the boundary conditions of $x(n)$ and the symmetric type of $g(n)$, perform the corresponding forward transform (DCT or DST) to obtain the coefficients $X(k)$ and $G(k)$;

2. Perform point-wise multiplication in transform domain to obtain the coefficients $Y(k) = G(k)X(k)$;

3. Perform corresponding inverse transform to obtain $y(n)$.

Tables 3.1 and 3.1 list various types of transform. Taking half-symmetric boundary extension of $x(n)$ and whole-point symmetric filter $g(n)$ for example, the following theorem gives the detailed computation in transform domain.

Theorem 3.5 Given $x(n)$ with half-symmetric boundary extension and whole-point symmetric filter $g(n)$, then, the DFT of $x(n)$ and $g(n)$ become N-dim DCT-II and $(N+1)$-dim DCT-I, respectively. The output signal $y(n)$ is half-point symmetric, its DFT becomes N-dim DCT-II.

Proof Under this boundary condition, DFT of $x(n)$ is:

$$\begin{aligned}
X(k) &= \sum_{n=0}^{2N-1} x(n)e^{-j\frac{2\pi kn}{2N}} = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{2N}} + \sum_{n=N}^{2N-1} x(n)e^{-j\frac{2\pi kn}{2N}} \\
&= \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{2N}} + \sum_{n=0}^{N-1} x(2N-1-n)e^{-j\frac{2\pi k(2N-1-n)}{2N}} \\
&= \sum_{n=0}^{N-1} x(n)\left(e^{-j\frac{2\pi kn}{2N}} + e^{j\frac{2\pi k(n+1)}{2N}}\right) \\
&= 2e^{j\frac{\pi k}{2N}} \underbrace{\sum_{n=0}^{N-1} x(n)\cos\left[\frac{\pi k}{N}\left(n+\frac{1}{2}\right)\right]}_{N\text{-dim DCT-II standard}}
\end{aligned}$$

The whole-point symmetric filter satisfies $g(n) = g(-n)$, then, the DFT of $g(n)$ is:

$$\begin{aligned} G(k) &= \sum_{n\in\mathbb{Z}} g(n)e^{-j\frac{2\pi kn}{2N}} = \sum_{n=-(N-1)}^{N} \underbrace{\Big(\sum_{n_0\in\mathbb{Z}} g(n+n_0\cdot 2N)\Big)}_{g_{2N}(n)} e^{-j\frac{2\pi kn}{2N}} \\ &= \sum_{n=-(N-1)}^{-1} g_{2N}(n)e^{-j\frac{2\pi kn}{2N}} + \sum_{n=1}^{N-1} g_{2N}(n)e^{-j\frac{2\pi kn}{2N}} + g_{2N}(0) + (-1)^k g_{2N}(N) \\ &= 2\times \underbrace{\left[\frac{1}{2}g_{2N}(0) + \frac{(-1)^k}{2}g_{2N}(N) + \sum_{n=1}^{N-1} g_{2N}(n)\cos\left(\frac{\pi kn}{N}\right)\right]}_{(N+1)\text{-dim DCT-I standard}} \end{aligned}$$

From Eq.(3.23), we have:

$$\begin{aligned} y(2N-1-n) &= \sum_{k=0}^{N-1}\Big[g_{2N}(2N-1-n-k) + g_{2N}(2N-1-n+k+1)\Big]x(k) \\ &= \sum_{k=0}^{N-1}\Big[g_{2N}(-1-n-k) + g_{2N}(-n+k)\Big]x(k) \quad (\text{period} = 2N) \\ &= \sum_{k=0}^{N-1}\Big[g_{2N}(n+k+1) + g_{2N}(n-k)\Big]x(k) \quad (\text{by } g_{2N}(n) = g_{2N}(-n)) \\ &= y(n) \end{aligned}$$

which illustrates that $y(n)$ is half-point symmetric. Hence, the DFT of $y(n)$ has the same form with that of $x(n)$. We omit here. ■

Analytical computation of σ_j^2 given in Eq.(3.17)

Theorem 3.6 Under half-point symmetric boundary conditions, given a zero-mean white Gaussian noise $b(n)$ with variance σ^2 and a filter $g(n)$, the variance of $p(n) = \sum_{k\in\mathbb{Z}} g(n-k)b(k)$ is given by:

$$\mathscr{E}\left\{p(n)^2\right\} = \sigma^2 \sum_{k=0}^{N-1}\Big[g_{2N}(n-k) + g_{2N}(n+k+1)\Big]^2$$

in spatial domain, where $g_{2N}(n)$ is given in Eq.(3.23), and is also expressed as:

$$\mathscr{E}\{p(n)^2\} = \frac{\sigma^2}{2N}\sum_{k=0}^{2N-1}\left\{\Big|G(e^{j\frac{\pi k}{N}})\Big|^2 + Re\Big\{\big(G(e^{j\frac{\pi k}{N}})\big)^2 e^{j\frac{\pi k}{N}(2n+1)}\Big\}\right\}$$

in Fourier domain, where $G\big(e^{j\frac{\pi k}{N}}\big) = \sum_{n=0}^{2N-1} g_{2N}(n)e^{-j\frac{\pi k}{N}n}$.

See Appendix A.4 for the proof. Theorem 3.6 indicates that the variance of the filtered noise $p(n)$ depends on n, which implies that filtering under periodic extension breaks the stationary property of $b(n)$. Furthermore, it can be easily generalized to two-dimensional case, where the given Gaussian white noise $b(m,n)$ for $m = 0, 1, ..., M-1$ and $n = 0, 1, ..., N-1$ with variance σ^2 is two-dimensional data. According to Theorem 3.6, applying any possible filter $g(m,n)$ to $b(m,n)$ will produce the filtered noise $p(m,n)$ that can be expressed as:

$$p(m,n) = \sum_{k=0}^{2M-1}\sum_{l=0}^{2N-1} g_{2M,2N}(m-k,n-l)b(k,l) = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} u(k,l)b(k,l) \quad (3.24)$$

where

$$\begin{aligned} u(k,l) &= g_{2M,2N}(m+k+1,n-l) + g_{2M,2N}(m+k+1,n+l+1) \\ &+ g_{2M,2N}(m-k,n-l) + g_{2M,2N}(m-k,n+l+1) \end{aligned} \quad (3.25)$$

and

$$g_{2M,2N}(m,n) = \sum_{m_0\in\mathbb{Z}}\sum_{n_0\in\mathbb{Z}} g(m+2m_0M, n+2n_0N) \quad (3.26)$$

The variance of the filtered noise $p(m,n)$ is given in the following theorem.

Corollary 3.7 If the filtered noise $p(m,n)$ is given by Eqs.(3.24), (3.25) and (3.26), its variance is written as:

$$\mathscr{E}\big(p(m,n)\big)^2 = \sigma^2 \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} \big(u(k,l)\big)^2 \quad (3.27)$$

in time domain, and is also given by

$$\mathscr{E}\big(p(m,n)\big)^2 = \frac{\sigma^2}{16MN}\sum_{k=0}^{2M-1}\sum_{l=0}^{2N-1} \big|U(k,l)\big|^2 \quad (3.28)$$

in Fourier domain, where

$$\begin{aligned} \Big|U(k,l)\Big|^2 &= 2Re\Big\{G(z_1,z_2)G(z_1,z_2)z_1^{2m+1}z_2^{2n+1}\Big\} + 4Re\Big\{G(z_1,z_2)G(z_1^{-1},z_2)z_2^{2n+1}\Big\} \\ &+ 2Re\Big\{G(z_1^{-1},z_2)G(z_1^{-1},z_2)z_1^{-(2m+1)}z_2^{2n+1}\Big\} + 4Re\Big\{G(z_1,z_2)G(z_1,z_2^{-1})z_1^{2m+1}\Big\} \\ &+ 2\Big(\big|G(z_1,z_2)\big|^2 + \big|G(z_1^{-1},z_2)\big|^2\Big) \end{aligned} \quad (3.29)$$

where G is given as:

$$\begin{cases} G(z_1, z_2) = \sum_{m=0}^{2M-1} \sum_{n=0}^{2N-1} g_{2M,2N}(m,n) z_1^{-m} z_2^{-n} \\ G(z_1^{-1}, z_2) = \sum_{m=0}^{2M-1} \sum_{n=0}^{2N-1} g_{2M,2N}(m,n) z_1^{m} z_2^{-n} \\ G(z_1, z_2^{-1}) = \sum_{m=0}^{2M-1} \sum_{n=0}^{2N-1} g_{2M,2N}(m,n) z_1^{-m} z_2^{n} \end{cases} \tag{3.30}$$

and $z_1 = e^{j\frac{\pi k}{M}}$, $z_2 = e^{j\frac{\pi l}{N}}$.

The proof is similar to Appendix A.4 of one-dimensional case. We omit it here.

Analytical computation of $\boldsymbol{\alpha}_j$ given in Eq.(3.15)

Theorem 3.8 Under half-point symmetric boundary conditions, the vector formed by the diagonal elements of the convolution matrix $\mathbf{G} \in \mathbb{R}^{N\times N}$ is given by

$$\left[\text{diag}\left\{\mathbf{G}\right\}\right]_n = g_{2N}(0) + g_{2N}(2n+1)$$

for $n = 0, 1, ..., N-1$, where $g_{2N}(n)$ is given by Eq.(3.23). It can also be expressed in the Fourier domain as

$$\left[\text{diag}\left\{\mathbf{G}\right\}\right]_n = \frac{1}{2N} \sum_{k=0}^{2N-1} G\big(e^{j\frac{\pi k}{N}}\big)\Big(1 + e^{j\frac{\pi k}{N}(2n+1)}\Big)$$

where $G(e^{j\frac{\pi k}{N}})$ is given in Theorem 3.6.

Proof The second equality in Fourier domain is essentially the forward DFT with period $2N$ of the first one. In matrix form, each element of the diagonal of $\mathbf{G}$ can be obtained as $\mathbf{e}_n^{\mathrm{T}}\mathbf{G}\mathbf{e}_n$, where the indicator vector $\mathbf{e}_n$ is defined as $\mathbf{e}_n = [0\ ...\ 0\ 1\ 0\ ...\ 0]^{\mathrm{T}} \in \mathbb{R}^N$ with n-th element 1. In the convolution form of Eq.(3.23), it reads as:

$$\begin{aligned} \mathbf{e}_n^{\mathrm{T}}\mathbf{G}\mathbf{e}_n &= \sum_{k=1}^{N} \delta(k-n) \left(\sum_{k'=1}^{N} \Big[g_{2N}(k'+k+1) + g_{2N}(k'-k)\Big] \delta(k'-n) \right) \\ &= \sum_{k=1}^{N} \delta(k-n) \Big[g_{2N}(n+k+1) + g_{2N}(n-k)\Big] \\ &= g_{2N}(n+n+1) + g_{2N}(n-n) \\ &= g_{2N}(2n+1) + g_{2N}(0) \end{aligned}$$

where $\delta(k-k')$ is defined as in Eq.(3.22). ■

From this theorem, we can see that contrary to the periodic boundary conditions, the diagonal elements of $\mathbf{G}$ are not equal, as $\mathbf{G}$ is not a circulant matrix under symmetric boundary conditions. However, it is not in contradiction to the fact that if we consider the problem as periodic extension with period $2N$, the matrix $\mathbf{G}^{2N\times 2N}$ is still circulant.

Furthermore, the results of Theorem 3.8 can be easily generalized to two-dimensional case, stated in the following corollary.

Corollary 3.9 Under half-point boundary condition, the diagonal element of convolution matrix $\mathbf{G}$, which denotes any possible 2-D filter $g(m,n)$, is given by:

$$\begin{aligned}\operatorname{diag}(\mathbf{G}) &= g_{2M,2N}(0,0) + g_{2M,2N}(2m+1,2n+1)\\ &+ g_{2M,2N}(2m+1,0) + g_{2M,2N}(0,2n+1)\end{aligned} \tag{3.31}$$

where $g_{2M,2N}(n)$ is defined by Eq.(3.26). It can also be expressed in Fourier domain:

$$\operatorname{diag}(\mathbf{G}) = \frac{1}{4MN}\sum_{k=0}^{2M-1}\sum_{l=0}^{2N-1} G(z_1,z_2)\Big(1 + z_1^{2m+1} + z_2^{2n+1} + z_1^{2m+1}z_2^{2n+1}\Big) \tag{3.32}$$

where $G(z_1,z_2)$ is given in Eq.(3.30).

3.5.4 Whole-point symmetric boundary extensions

Formulation for whole-point symmetric convolution

Lemma 3.10 Given a signal $x(n)$ and filter $g(n)$, under whole-point symmetric boundary condition of $x(n)$, convolution can be written as anyone of the following two expressions:

$$\begin{cases} y(n) = \sum_{k=0}^{N-1} g_{2N-2}(n-k)x(k) + \sum_{k=1}^{N-2} g_{2N-2}(n+k)x(k)\\ y(n) = \sum_{k=1}^{N-2} g_{2N-2}(n-k)x(k) + \sum_{k=0}^{N-1} g_{2N-2}(n+k)x(k)\end{cases} \tag{3.33}$$

for $n = 0,1,...,N-1$, where $g_{2N-2}(n) = \sum_{n_0\in\mathbb{Z}} g\big(n + n_0\cdot(2N-2)\big)$.

Proof Under whole-point symmetric extension, the signal $x(n)$ is with $(2N-2)$ period, and satisfies $x(n) = x(2N-2-n)$ for $n = 0, 1, ..., N-1$. Hence, the convolution is:

$$\begin{aligned} y(n) &= \sum_{k\in\mathbb{Z}} g(n-k)x(k) = \sum_{k=0}^{2N-3} \Big(\underbrace{\sum_{n_0\in\mathbb{Z}} g\Big(n-k+n_0\cdot(2N-2)\Big)}_{g_{2N-2}(n-k)} \Big) x(k) \\ &= g_{2N-2}(n)x(0) + g_{2N-2}\big(n-N+1\big)x(N-1) + \sum_{k=1}^{N-2} g_{2N-2}(n-k)x(k) \\ &+ \underbrace{\sum_{k=N}^{2N-3} g_{2N-2}(n-k)x(k)}_{=\sum_{k=1}^{N-2} g_{2N-2}\left(n+k\right)x(k)} \quad \text{(by changing variable } k' = 2N-2-k\text{)} \end{aligned} \tag{3.34}$$

Hence, we obtain:

$$\begin{aligned} y(n) &= g_{2N-2}(n)x(0) + g_{2N-2}\big(n-N+1\big)x(N-1) \\ &+ \sum_{k=1}^{N-2} \Big(g_{2N-2}(n-k) + g_{2N-2}(n+k) \Big) x(k) \end{aligned}$$

Merging the terms yields Eq.(3.33), and completes the proof. ■

Computation of filtering via transform domain

Again, refer to Tables 3.1 and 3.1 for various types of transform. Taking whole-symmetric boundary extension of $x(n)$ and whole-point symmetric filter $g(n)$ for example, the following theorem gives the detailed computation in transform domain.

Theorem 3.11 Given $x(n)$ with whole-point symmetric boundary extension and whole-point symmetric filter $g(n)$, then, the DFT of $x(n)$ and $g(n)$ become N-dim DCT-I and $(N+1)$-dim DCT-I, respectively. The output signal $y(n)$ is whole-point symmetric, its DFT is exactly the same as that of $x(n)$.

Proof Lemma 3.10 stated that $x(n)$ is with period $(2N-2)$ under whole-point symmetric extension. By changing variable $n' = 2N-2-n$, the frequency coefficients

$X(k)$ is given as:

$$\begin{aligned}
X(k) &= \sum_{n=0}^{2N-3} x(n)e^{-j\frac{2\pi kn}{2N-2}} \\
&= x(0) + x(N-1)\underbrace{e^{-j\frac{2\pi k(N-1)}{2N-2}}}_{e^{j\pi k}} + \sum_{n=1}^{N-2} x(n)e^{-j\frac{2\pi kn}{2N-2}} + \underbrace{\sum_{n=N}^{2N-3} x(n)e^{-j\frac{2\pi kn}{2N-2}}}_{\sum_{n=1}^{N-2} x(n)e^{j\frac{2\pi kn}{2N-2}}} \\
&= x(0) + (-1)^k x(N-1) + \sum_{n=1}^{N-2} x(n)\left(e^{-j\frac{2\pi kn}{2N-2}} + e^{j\frac{2\pi kn}{2N-2}}\right) \\
&= 2\underbrace{\left[\frac{1}{2}x(0) + \frac{(-1)^k}{2}x(N-1) + \sum_{n=1}^{N-2} x(n)\cos\left(\frac{\pi kn}{N-1}\right)\right]}_{N\text{-dim DCT-I standard}}
\end{aligned}$$

In this case, $G(k)$ is given by:

$$\begin{aligned}
G(k) &= \sum_{n=-(N-2)}^{N-1} g_{2N-2}(n)e^{-j\frac{2\pi kn}{2N-2}} \\
&= g_{2N-2}(0) + (-1)^k g_{2N-2}(N-1) + \left(\sum_{n=-(N-2)}^{-1} + \sum_{n=1}^{N-2}\right) g_{2N-2}(n)e^{-j\frac{2\pi kn}{2N-2}} \\
&= 2\underbrace{\left[\frac{1}{2}g_{2N-2}(0) + \frac{(-1)^k}{2}g_{2N-2}(N-1) + \sum_{n=1}^{N-2} g_{2N-2}(n)\cos\left(\frac{\pi kn}{N-1}\right)\right]}_{N\text{-dim DCT-I standard}}
\end{aligned}$$

The output $y(n)$ keeps the same symmetry as $x(n)$. ■

Analytical computation of σ_j^2 given in Eq.(3.17)

Theorem 3.12 Under whole-point symmetric boundary conditions, given a zero-mean white Gaussian noise $b(n)$ with variance σ^2 and a filter $g(n)$, the filtered noise $p(n)$ is given by:

$$p(n) = \sum_{k=1}^{N-2}\Big[\underbrace{\hat{g}(n-k) + \hat{g}(n+k)}_{u(k)}\Big]b(k) + \hat{g}(n)b(0) + \hat{g}(n-N+1)b(N-1) \qquad (3.35)$$

where we denote $\hat{g}(n) = g_{2N-2}(n)$ for brevity. The variance of $p(n)$ is given by:

$$\mathscr{E}\left\{\big(p(n)\big)^2\right\} = \frac{\sigma^2}{2}\sum_{k\in\mathbb{P}} u(k)^2 - \frac{\sigma^2}{4}u(0)^2 - \frac{\sigma^2}{4}u(N-1)^2 \qquad (3.36)$$

in spatial domain, where $\mathbb{P}$ denotes any interval with period length $2N-2$, and is also expressed as:

$$\mathscr{E}\Big\{\big(p(n)\big)^2\Big\} = \frac{\sigma^2}{2N-2}\sum_{k=0}^{2N-3}\Big\{\big|G(z)\big|^2 + Re\Big\{\big(G(z)\big)^2 z^{2n}\Big\}\Big\} \tag{3.37}$$

$$- \frac{\sigma^2}{2N-2}\Big\{\Big(\sum_{k=0}^{2N-3} G(z)z^n\Big)^2 + \Big(\sum_{k=0}^{2N-3}(-1)^k G(z)z^n\Big)^2\Big\} \tag{3.38}$$

in frequency domain, where $G(z) = \sum_{k'} \hat{g}(k')z^{-k'}$ with $z = e^{j\frac{2\pi k}{2N-2}}$.

See the proof in Appendix A.5. The computation of 2-D case can be naturally extended, refer to Appendix A.6.

Analytical computation of $\boldsymbol{\alpha}_j$ given in Eq.(3.15)

Theorem 3.13 Under half-point symmetric boundary conditions, the vector formed by the diagonal elements of the convolution matrix $\mathbf{G} \in \mathbb{R}^{N\times N}$ is given by

$$\big[\text{diag}\,\{\mathbf{G}\}\,\big]_n = \begin{cases} g_{2N-2}(0) & \text{for } n = 0 \text{ or } N-1 \\ g_{2N-2}(0) + g_{2N-2}(2n) & \text{for } n = 1, 2, ..., N-2 \end{cases}$$

where $g_{2N-2}(n)$ is given by Eq.(3.33).

Proof The second equality in Fourier domain is essentially the forward DFT with period $2N$ of the first one. In matrix form, each element of the diagonal of $\mathbf{G}$ can be obtained as $\mathbf{e}_n^{\mathrm{T}}\mathbf{G}\mathbf{e}_n$, where the indicator vector $\mathbf{e}_n$ is defined as $\mathbf{e}_n = [0 \,...\, 0\; 1\; 0 \,...\, 0]^{\mathrm{T}} \in \mathbb{R}^N$ with n-th element 1. In the convolution form of Eq.(3.23), it reads as:

$$\begin{aligned}\mathbf{e}_n^{\mathrm{T}}\mathbf{G}\mathbf{e}_n &= \sum_{k=1}^{N}\delta(k-n)\Bigg(\Bigg(\sum_{k'=1}^{N} g_{2N-2}(n-k') + \sum_{k'=2}^{N-1} g_{2N-2}(n+k')\Bigg)\delta(k'-n)\Bigg) \\ &= \begin{cases} g_{2N-2}(0) & \text{for } n = 0 \text{ or } N-1 \\ g_{2N-2}(0) + g_{2N-2}(2n) & \text{for } n = 1, 2, ..., N-2 \end{cases}\end{aligned}$$

where $\delta(k-k')$ is defined as in Eq.(3.22). ■

3.6 Results and discussions

3.6.1 Experimental setting

We consider the following benchmark convolution kernels commonly used in [34; 44; 49; 88]:

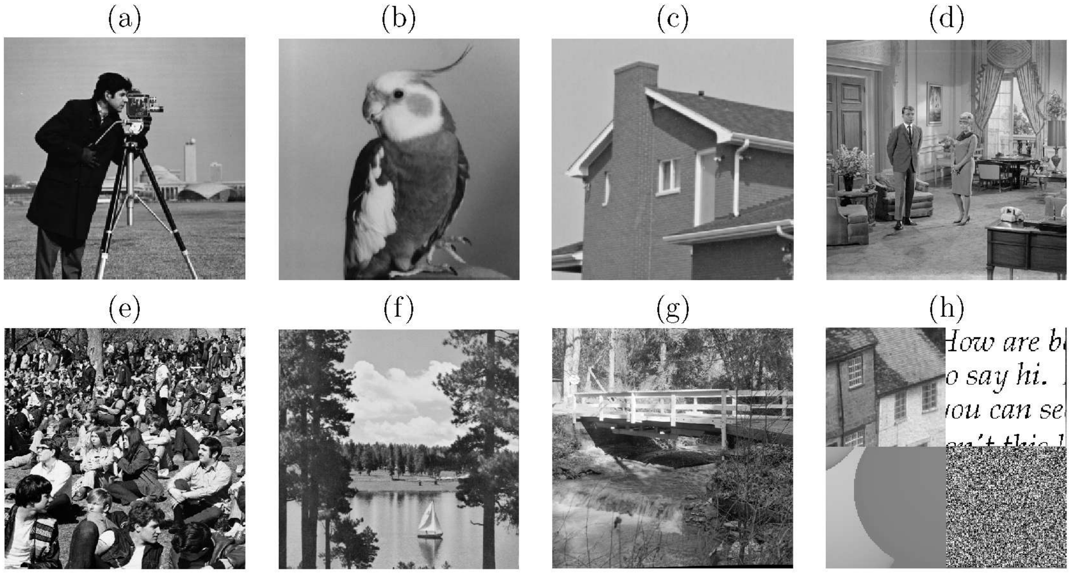


Figure 3.6: *Original images. (a) Cameraman* 256 × 256; *(b) Coco* 256 × 256; *(c) House* 256 × 256; *(d) Couple* 512 × 512; *(e) Crowd* 512 × 512; *(f) Lake* 512 × 512; *(g) Bridge* 512 × 512; *(h) Mixture* 512 × 512.

- **Rational filter:** $h(i, j) = (1 + i^2 + j^2)^{-1}$ for $i, j = -7, ..., 7$;
- **Separable filter:** 5×5 filter with weights $[1, 4, 6, 4, 1]/16$ along both horizontal and vertical directions;
- 5×5 **uniform blur**;
- 9×9 **uniform blur**;
- **Gaussian blur** with standard deviation (std) 3.

The blurred images are subsequently contaminated by i.i.d Gaussian noise with various variance σ^2. The test dataset contains eight 8-bit images of size 256×256 or 512×512 displayed in Figure 3.6[2], covering a wide range of natural images. The experimental performance is measured by PSNR defined as Eq.(1.11) and SSIM defined as Eq.(1.12). Note that all the PSNR results (in dB) and the SSIM results reported in this section have been averaged over 10 noise realizations.

3.6.2 Influence of the number of Wiener filters

Our multi-Wiener SURE-LET approach involves several Wiener filters as elementary processing in the linear combination. In this section, we evaluate the influence of

[2] All 512 × 512 images are available at http://decsai.ugr.es/cvg/CG/base.htm

Case	*House*, **Separable** $\sigma = 1$ Input: 30.92dB BM3D: 35.80dB		*Cameraman*, **Gaussian** $\sigma = 10$ Input: 20.22dB BM3D: 22.60dB		*Coco*, **Gaussian** $\sigma = 1$ Input: 26.45dB BM3D: 31.27dB	
Number of deconvolutions	SURE-LET	*MSE-LET*	SURE-LET	*MSE-LET*	SURE-LET	*MSE-LET*
one [1]	34.33	*34.59*	22.30	*22.47*	31.02	*31.20*
two [2]	35.78	*36.04*	22.37	*22.65*	31.40	*31.68*
three [3]	**36.31**	*36.47*	**22.52**	*22.70*	**31.70**	*32.00*
four [4]	**36.27**	*36.51*	**22.54**	*22.74*	**31.70**	*32.03*

[1] one deconvolution: $\lambda = 10^{-4}\sigma^2$;

[2] two deconvolutions: $\lambda_1 = 10^{-4}\sigma^2$ and $\lambda_2 = 10^{-2}\sigma^2$;

[3] three deconvolutions: $\lambda_1 = 10^{-4}\sigma^2$, $\lambda_2 = 10^{-3}\sigma^2$ and $\lambda_3 = 10^{-2}\sigma^2$;

[4] four deconvolutions: $\lambda_1 = 10^{-5}\sigma^2$, $\lambda_2 = 10^{-4}\sigma^2$, $\lambda_3 = 10^{-3}\sigma^2$ and $\lambda_4 = 10^{-2}\sigma^2$.

Table 3.3: *PSNR results of our multi-Wiener SURE-LET for various number of Wiener filters.*

the retained number of Wiener filters on the deconvolution performance. Based on the observation that a reasonable value of the regularization parameter λ should be proportional to the noise variance σ^2 [43; 49], we focus on the ratio λ/σ^2. Table 3.3 shows the results obtained when considering one to four Wiener filters for various images, blur kernels and noise levels. We observe that one or two Wiener filters may not always produce the best performance, whereas three or four are able to cope with various scenarios (different images and different blur kernels). Moreover, using four Wiener filterings does not bring any significant performance improvement, compared to using three. Besides, we also found that for one Wiener filter, the optimal value of λ/σ^2 typically varies in the range 2×10^{-4} to 2×10^{-3}. For these reasons, in the following sections, we use three Wiener filters (i.e. $M = 3$) with $\lambda_1 = 1 \times 10^{-4}\sigma^2$, $\lambda_2 = 1 \times 10^{-3}\sigma^2$ and $\lambda_3 = 1 \times 10^{-2}\sigma^2$. Extensive tests show that the deconvolution performance is largely insensitive to the actual choice of λ's.

3.6.3 Influence of the parameters on the deconvolution performance

Apart from the case of λ_1, λ_2 and λ_3 already discussed above (Section 3.6.2), we report the following experimental observations, based on extensive tests with various images, blur types and noise levels. We consider the following two experimental scenarios for typical examples.

- **Scenario A**: *House*, 9×9 **uniform blur**, noise variance $\sigma^2 = 1.0$;

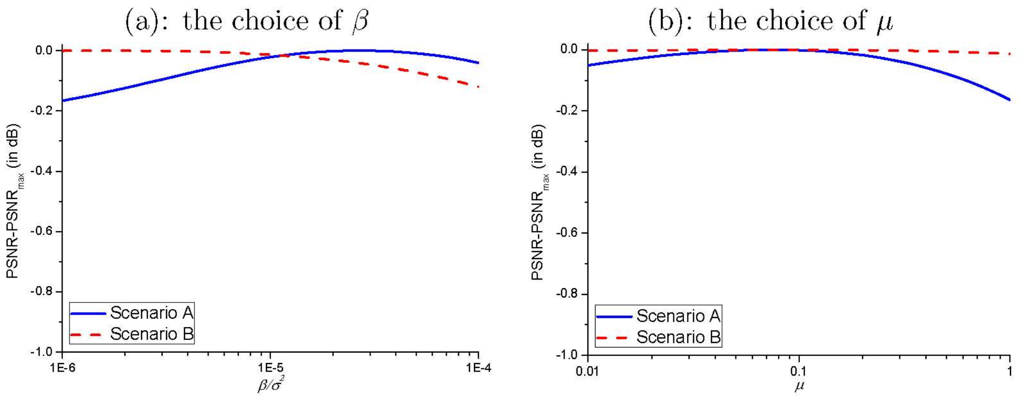


Figure 3.7: *The relationship between β, μ and PSNR performance under two particular degradation scenarios. (a): β vs. PSNR loss, compared to maximum PSNR with optimal β; (b): μ vs. PSNR loss, compared to maximum PSNR with optimal μ.*

- **Scenario B**: *Crowd*, **Gaussian kernel** with variance 9.0, noise variance $\sigma^2 = 10.0$

Influence of β in Eq.(3.14) and μ in Eq.(2.9) Figure 3.7 shows the relationship between β, μ and PSNR performance under two particular degradation scenarios. From it, we can see that choosing β in $[5\times10^{-6}\sigma^2, 5\times10^{-5}\sigma^2]$ and μ in $[0.01, 0.1]$ yields very similar deconvolution results with PSNR loss within 0.05dB, compared to the maximum PSNR with optimal values.

Regarding the choice of the thresholding functions θ_1 and θ_2 To be efficient, the pointwise thresholding functions $\boldsymbol{\theta}_l(\cdot)$ have to satisfy some desirable properties that are discussed in [1; 110]. We empirically found that a linear combination of the following two functions (see Figure 3.8)

$$\left\{\begin{array}{lcl}\theta_1(w) & = & w\left(1-\exp\left(-\left(\dfrac{w}{T_1}\right)^4\right)\right)\\[2ex] \theta_2(w) & = & w\left(1-\exp\left(-\left(\dfrac{w}{T_2}\right)^4\right)\right)\end{array}\right. \tag{3.39}$$

with $T_1 = 4\sigma_j$ and $T_2 = 9\sigma_j$ yields satisfactory results. We compare our choice shown as Eq.(3.39) with other two alternatives, reported in [1; 113]:

- **Choice A** (in [1]): $\theta_1(w) = w$, $\theta_2(w) = w\Big(1-\exp\big(-(\frac{w}{3\sigma_j})^8\big)\Big)$
- **Choice B** (in [113]): $\theta_1(w) = w$, $\theta_2(w) = w\Big(\tanh\big(\frac{w+3.5\sigma_j}{2.25\sigma_j}\big) - \tanh\big(\frac{w-3.5\sigma_j}{2.25\sigma_j}\big)\Big)$

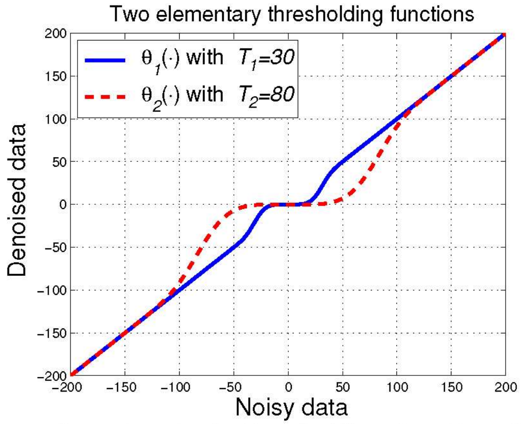


Figure 3.8: *Shape of our two basic thresholding functions $\theta_1(\cdot)$ and $\theta_2(\cdot)$, given by* (3.39).

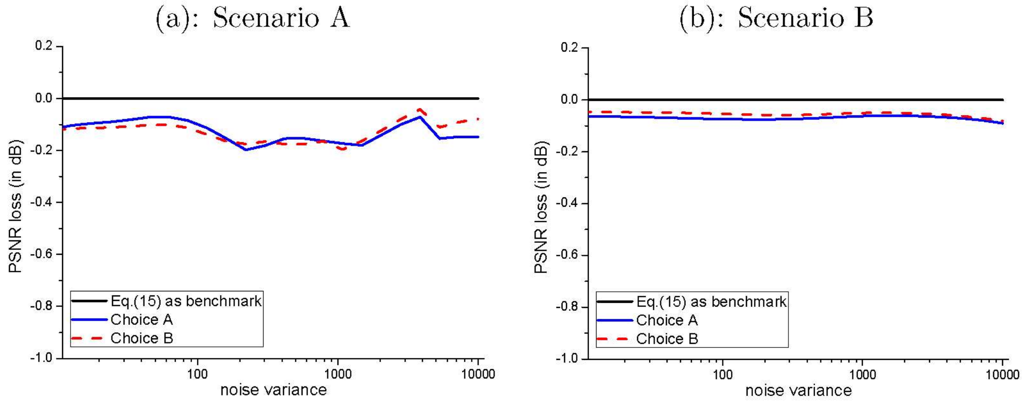


Figure 3.9: *Choice of thresholding functions θ_1 and θ_2 vs. PSNR loss, compared to PSNR obtained using Eq.*(3.39).

Figure 3.9 shows the comparison curves between the three options with Eq.(3.39) as the benchmark: the comparison is evaluated as the PSNR loss between Choice A, B and Eq.(3.39), respectively. From the curves in Figure 3.9, we can see that the the two alternatives yield the PSNR loss within 0.2dB.

Regarding the choice of the thresholds T_1 and T_2 In the preceding paragraph, we choose $T_1 = 4\sigma_j$ and $T_2 = 9\sigma_j$. We may vary the values of T_1 and T_2 to see the PSNR performance: Figure 3.10 shows the relationship between T_1, T_2 and PSNR loss, compared to maximum PSNR with optimal T_1 and T_2.

We found that any combinations of T_1 in $[3\sigma, 6\sigma]$ and T_2 in $[7\sigma, 10\sigma]$ achieve very similar PSNR performance (the PSNR variations are generally within 0.1dB). It implies that We empirically found that the proposed algorithm is able to keep similar deconvolution quality for a wide range of value T_1 and T_2.

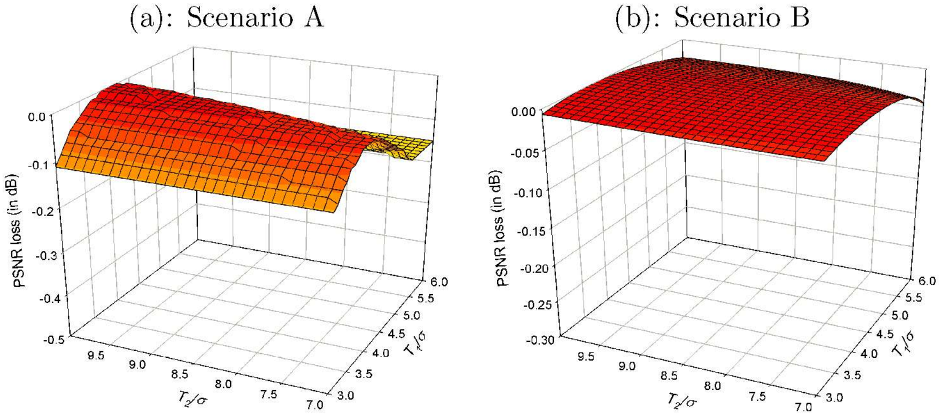


Figure 3.10: *The relationship between T_1, T_2 and PSNR performance under two particular degradation scenarios: T_1 and T_2 vs. PSNR loss, compared to maximum PSNR with optimal T_1 and T_2.*

Choice of wavelet basis In this thesis, we apply an undecimated Haar wavelet to perform wavelet-domain thresholding, since the redundant Haar substantially outperforms other types of redundant wavelets for image denoising [1]. This observation is also true for image deconvolution.

Concluding remarks Based on these observations, all the parameters involved in the proposed multi-Wiener SURE-LET algorithm are specified in Table 3.4. According to this table, we have $K = MJL + M = 6J + 3$ elementary functions, which yields K weights to be optimized using Eq. (2.9). Typically, $J = 9$ or 12, for three or four levels of an undecimated Haar wavelet decomposition, and hence, we have $K = 6 \cdot 9 + 3 = 57$ linear coefficients to be optimized for $J = 9$. Table 3.4 shows that only 7 parameters involved in our algorithm need to be manually tuned, whereas we have typically up to $K = 75$ (for 4 wavelet iterations) parameters that are automatically adjusted to the image. By comparison, BM3D [49; 52] has more than 60 non-linear parameters to be manually determined for an efficient block matching and collaborative filtering.

$\mathbf{S}$ in Eqs.(3.1) and (2.5)	Discrete Laplacian operator
β in Eq.(2.5)	$\beta = 1 \times 10^{-5}\sigma^2$
λ_k in Eq.(3.1)	$\lambda_1 = 1 \times 10^{-4}\sigma^2$; $\lambda_2 = 1 \times 10^{-3}\sigma^2$; $\lambda_3 = 1 \times 10^{-2}\sigma^2$
T_l in Eq.(3.39)	$T_1 = 4\sigma_j$; $T_2 = 9\sigma_j$
Transforms $\mathbf{D}$ and $\mathbf{R}$	Undecimated Haar wavelet transform
μ in Eq.(2.9)	$\mu = 5 \times 10^{-2}$

Table 3.4: *Parameters setting of the proposed multi-Wiener SURE-LET algorithm.*

Finally, we would like to note that the preceding observations strongly implies that the proposed algorithm is very robust and largely insensitive to the parameters. No optimization is needed here. The main reason for the robustness is that any changes on the parameters or functions (β, μ, θ_1, θ_2, T_1 and T_2) are eventually counterbalanced by an (optimal) adaptation of the (at least 57) LET coefficients a_k (final PSNR variations of $0.1 \sim 0.2$dB). For this reason, the proposed algorithm becomes highly adaptive to various scenarios, even if the parameters in Table 3.4 are not well-tuned.

3.6.4 Influence of the boundary conditions: periodic vs symmetric

Table 3.5 displays the PSNR results obtained for different boundary conditions (periodic, half-point symmetric or whole-point symmetric). As observed, the type of boundary conditions has an impact on the deconvolution performance. From the results, we can conclude that symmetric boundary conditions perform uniformly better than periodic. This is mainly because symmetric boundary conditions ensure slowly-varying changes at the boundaries, unlike periodic conditions which may introduce discontinuities. The deconvolution gain obtained by symmetric boundary conditions over periodic ones depends on the image and blur scenario. Table 3.5 reports the PSNR results obtained on *House* and *Cameraman* as two extreme cases: for *Cameraman*, the boundary conditions have almost no influence on the deconvolution performance, whereas significant differences are noticed for *House*.

3.6.5 Comparison with the state-of-the-art

In Tables 3.6–3.11, we present several comparisons between the proposed multi-Wiener SURE-LET method and some state-of-the-art deconvolution techniques whose softwares are available online, in terms of PSNR and SSIM. Various degradation scenarios have been considered. In order to compare with other methods in exactly the same experimental situations, we have considered only periodic boundary condition. The

Boundary conditions	periodic	symmetric		periodic	symmetric	
		half-point	whole-point		half-point	whole-point
Image	*House* 256 × 256					
Blur kernel	**Rational filter**			**Separable filter**		
Input	25.64	26.30	26.29	30.92	31.83	31.82
SURE-LET	35.33	**35.50**	35.46	36.40	**36.51**	36.47
MSE-LET	*35.48*	*35.61*	*35.58*	*36.58*	*36.82*	*36.77*
Blur kernel	9 × 9 **uniform**			**Gaussian blur** with std 3		
Input	24.09	24.75	24.75	24.22	24.86	24.86
SURE-LET	32.97	**32.98**	32.97	29.37	**30.03**	30.01
MSE-LET	*33.19*	*33.26*	*33.20*	*29.74*	*30.32*	*30.32*
Image	*Cameraman* 256 × 256					
Blur kernel	**Rational filter**			**Separable filter**		
Input	22.24	22.37	22.36	25.67	25.79	25.79
SURE-LET	30.91	**30.97**	30.96	30.83	**30.93**	30.90
MSE-LET	*31.12*	*31.17*	*31.16*	*31.10*	*31.17*	*31.16*
Blur kernel	9 × 9 **uniform**			**Gaussian blur** with std 3		
Input	20.76	20.89	20.89	20.97	21.10	21.10
SURE-LET	27.40	**27.47**	27.46	23.97	**24.14**	24.14
MSE-LET	*27.75*	*27.77*	*27.76*	*24.25*	*24.41*	*24.40*

Table 3.5: *PSNR results of our multi-Wiener SURE-LET for different boundary conditions ($\sigma^2 = 1$).*

state-of-the-art methods include ForWaRD[3] [45], SA-DCT[4] [48], BM3D[5] [49], TV-MM[6] [43], C-SALSA[7] [38] and Pesquet's *et al.* work[8] [113]. We have run all the source codes by default throughout all the experiments performed. For TV-MM [43] and C-SALSA [38], we used the default stopping criteria suggested by their respective authors. For [113], we used symlet-8 translation invariant wavelet, as it gives slightly better performance than using Haar undecimated wavelet (generally by 0.2∼0.3dB).

The "*MSE-LET*" rows of Tables Tables 3.6–3.11 are the results of minimizing the actual MSE in lieu of the SURE, demonstrating that the latter is a good substitute to the MSE minimization.

It can be seen from the reported PSNR and SSIM scores that the proposed SURE-LET algorithm uniformly achieves the highest performance. Also note that the higher PSNR usually yields higher SSIM. Although TV-MM is well-known for its outstanding

[3] The source code of ForWaRD is available at http://dsp.rice.edu/software/forward

[4] The source code of SA-DCT is available at http://www.cs.tut.fi/~foi/SA-DCT/#ref_software

[5] The source code of BM3D is available at http://www.cs.tut.fi/~foi/GCF-BM3D/

[6] The source code of TV-MM is available at http://www.lx.it.pt/~bioucas/code.htm

[7] The source code of C-SALSA is available at http://cascais.lx.it.pt/~mafonso/salsa.html

[8] The source code of [113] is available at http://www-syscom.univ-mlv.fr/~chaux/toolbox/TOOLBOX_Surelet_deconvolution_v1.0.zip

σ	1	5	10	30	50	100	1	5	10	30	50	100
Image	*Cameraman* 256 × 256						*Coco* 256 × 256					
Input	20.97	20.78	20.22	16.61	13.33	7.91	26.45	25.78	24.19	17.92	13.89	8.06
ForWaRD	23.76	22.88	22.40	21.18	20.35	18.79	31.18	29.48	28.51	25.91	24.22	20.80
SA-DCT	23.73	**23.10**	22.50	20.80	19.39	16.86	31.27	**29.82**	28.58	24.76	22.16	18.09
BM3D	**24.05**	**23.10**	**22.61**	**21.46**	20.78	**19.74**	31.27	29.70	**28.72**	26.61	25.48	23.80
TV-MM	23.93	22.80	22.06	21.10	20.24	18.65	**31.65**	29.59	28.56	26.30	24.75	21.81
C-SALSA	23.75	22.81	22.26	21.27	20.47	18.95	31.35	29.07	28.47	26.13	24.31	21.50
[113]	23.32	22.44	21.83	20.78	20.25	18.95	30.39	28.58	27.70	25.81	24.82	16.88
SURE-LET	**23.97**	**23.02**	**22.52**	**21.50**	**20.91**	**19.80**	**31.57**	**29.80**	**28.80**	**26.95**	**25.78**	**24.02**
MSE-LET	*24.25*	*23.30*	*22.70*	*21.72*	*21.11*	*20.08*	*31.88*	*30.10*	*29.12*	*27.28*	*26.10*	*24.46*
Image	*House* 256 × 256						*Couple* 512 × 512					
Input	24.22	23.81	22.73	17.52	13.73	8.02	23.55	23.20	22.26	17.39	13.68	8.01
ForWaRD	28.87	27.43	26.63	24.27	22.87	20.23	26.40	25.25	24.62	23.12	22.25	19.64
SA-DCT	28.94	27.86	26.75	23.51	21.32	17.69	26.43	**25.35**	24.62	22.75	21.20	18.04
BM3D	29.19	**27.90**	**27.10**	**25.17**	23.98	**22.32**	**26.60**	**25.39**	**24.76**	23.42	22.72	21.37
TV-MM	**29.35**	27.50	26.72	24.60	23.08	20.63	26.12	24.92	24.20	23.02	21.79	19.53
C-SALSA	29.25	27.46	26.55	24.51	22.97	20.50	26.36	25.20	24.59	23.30	22.04	19.91
[113]	28.19	26.64	25.90	23.93	22.72	15.57	26.29	25.06	24.49	23.20	22.44	21.48
SURE-LET	**29.27**	**27.81**	**27.00**	**25.20**	**24.10**	**22.35**	**26.56**	**25.40**	**24.80**	**23.59**	**22.90**	**21.88**
MSE-LET	*29.62*	*28.01*	*27.24*	*25.54*	*24.33*	*22.62*	*26.64*	*25.47*	*24.88*	*23.68*	*22.99*	*21.96*
Image	*Crowd* 512 × 512						*Lake* 512 × 512					
Input	16.56	16.49	16.27	14.45	12.19	7.55	22.98	22.67	21.83	17.24	13.61	7.99
ForWaRD	**19.97**	18.83	18.22	17.21	16.57	15.25	26.80	25.39	24.65	22.94	21.72	18.84
SA-DCT	19.87	**18.97**	**18.33**	17.02	16.30	14.97	**26.82**	**25.49**	24.57	22.44	20.97	18.05
BM3D	**20.01**	18.80	18.15	16.83	16.15	14.93	**26.90**	**25.48**	24.68	22.97	22.10	20.42
TV-MM	19.43	18.77	18.17	17.10	16.18	14.87	26.58	25.25	24.38	22.83	21.46	19.12
C-SALSA	19.92	**18.88**	**18.31**	**17.33**	**16.73**	15.65	26.74	**25.40**	24.67	23.05	21.73	19.50
[113]	19.80	18.61	18.04	16.91	16.36	15.55	26.67	25.20	24.44	22.89	22.04	20.74
SURE-LET	**20.05**	**18.90**	**18.30**	**17.25**	**16.67**	**15.79**	**26.90**	**25.48**	**24.80**	**23.33**	**22.48**	**21.18**
MSE-LET	*20.21*	*19.06*	*18.48*	*17.40*	*16.81*	*15.89*	*26.97*	*25.62*	*24.92*	*23.49*	*22.64*	*21.32*
Image	*Bridge* 512 × 512						*Mixture* 512 × 512					
Input	21.39	21.18	20.57	16.77	13.41	7.94	14.84	14.79	14.64	13.31	11.46	7.29
ForWaRD	23.73	22.79	22.28	21.32	20.53	18.41	16.63	15.94	**15.72**	**15.15**	**14.77**	14.08
SA-DCT	23.75	22.84	22.32	21.27	20.50	18.63	16.64	**16.13**	**15.80**	**15.19**	**14.75**	13.64
BM3D	**23.85**	**22.88**	**22.38**	21.38	20.78	19.61	**16.74**	15.95	15.59	14.91	14.60	14.02
TV-MM	23.56	22.69	21.99	20.51	19.80	18.12	16.37	15.69	15.20	14.72	14.46	13.52
C-SALSA	23.78	**22.87**	22.09	21.38	20.52	18.89	16.52	15.96	15.65	15.07	14.70	13.92
[113]	23.70	22.73	22.24	21.27	20.71	19.87	16.56	15.88	15.57	14.98	14.67	14.27
SURE-LET	**23.90**	**22.95**	**22.44**	**21.54**	**20.95**	**20.01**	**16.70**	**16.07**	**15.74**	**15.15**	**14.80**	**14.43**
MSE-LET	*23.97*	*23.01*	*22.53*	*21.61*	*21.05*	*20.14*	*16.78*	*16.12*	*15.81*	*15.28*	*14.93*	*14.53*

*Best PSNR results within a 0.1dB margin are highlighted. The results have been averaged over 10 noise realizations.

Table 3.6: *PSNR Comparison of some state-of-the-art deconvolution methods under Gaussian blur with variance 9*.*

performance on regularly-structured images such as *House*, *Coco*, and the left-bottom of *Mixture*, it is substantially outperformed by the proposed algorithm. For more complicated images like *Crowd* with lots of irregular edges and disordered features, the proposed method also shows better PSNR results than the other techniques. ForWaRD is more effective for these less structured images than for cartoon-like images. BM3D, which achieves the best performance among the other techniques on average, is not as efficient on these irregular structures. Note that the proposed multi-Wiener SURE-LET algorithm is very robust to a wide range of noise levels from $\sigma^2 = 1$ to $\sigma^2 = 1 \times 10^4$. In

σ	1	5	10	30	50	100	1	5	10	30	50	100
Image	*Cameraman* 256 × 256						*Coco* 256 × 256					
Input	0.64	0.51	0.34	0.11	0.06	0.02	0.83	0.65	0.42	0.11	0.05	0.02
ForWaRD	0.73	0.66	0.60	0.56	0.43	0.37	0.85	0.78	0.79	0.66	0.60	0.48
SA-DCT	0.72	0.69	0.61	0.51	0.35	0.30	0.85	**0.87**	0.83	0.53	0.48	0.41
BM3D	0.73	0.69	0.66	0.59	0.56	0.50	0.86	0.82	0.80	0.75	0.72	0.69
TV-MM	0.74	0.67	0.68	0.55	0.42	0.36	**0.90**	0.80	0.83	0.73	0.65	0.53
C-SALSA	0.73	0.69	0.66	0.56	0.48	0.38	0.89	0.86	0.83	0.72	0.63	0.50
[113]	0.68	0.63	0.56	0.51	0.43	0.38	0.84	0.75	0.74	0.65	0.63	0.22
SURE-LET	**0.76**	**0.71**	**0.67**	**0.62**	**0.58**	**0.53**	**0.90**	**0.87**	**0.85**	**0.81**	**0.78**	**0.73**
MSE-LET	*0.77*	*0.73*	*0.70*	*0.65*	*0.62*	*0.57*	*0.91*	*0.88*	*0.86*	*0.82*	*0.80*	*0.75*
Image	*House* 256 × 256						*Couple* 512 × 512					
Input	0.71	0.57	0.38	0.12	0.06	0.02	0.70	0.67	0.62	0.38	0.24	0.10
ForWaRD	0.75	0.76	0.71	0.60	0.55	0.41	**0.88**	0.80	0.75	0.62	0.60	0.38
SA-DCT	0.76	0.78	0.73	0.54	0.48	0.36	**0.88**	**0.81**	0.75	0.58	0.45	0.38
BM3D	0.78	0.76	0.73	0.67	0.64	0.59	**0.88**	**0.81**	**0.77**	0.66	0.60	0.50
TV-MM	**0.83**	0.77	0.72	0.63	0.60	0.44	0.84	0.73	0.72	0.62	0.49	0.38
C-SALSA	0.81	0.77	0.74	0.63	0.55	0.44	0.87	0.80	0.75	0.65	0.52	0.38
[113]	0.73	0.71	0.67	0.57	0.54	0.31	0.86	0.77	0.75	0.62	0.58	0.51
SURE-LET	0.81	**0.78**	**0.76**	**0.72**	**0.68**	**0.63**	**0.88**	**0.81**	**0.77**	**0.67**	**0.61**	**0.53**
MSE-LET	*0.83*	*0.80*	*0.78*	*0.73*	*0.70*	*0.65*	*0.88*	*0.81*	*0.77*	*0.68*	*0.62*	*0.54*
Image	*Crowd* 512 × 512						*Lake* 512 × 512					
Input	0.61	0.60	0.59	0.50	0.41	0.26	0.78	0.75	0.68	0.42	0.28	0.13
ForWaRD	0.86	0.80	0.76	0.68	0.59	0.45	0.91	0.84	0.81	0.73	0.56	0.52
SA-DCT	0.86	0.81	**0.77**	0.64	0.57	0.40	0.91	**0.87**	0.80	0.67	0.48	0.52
BM3D	**0.88**	0.81	0.76	0.63	0.55	0.40	**0.92**	**0.87**	0.83	0.73	0.67	0.58
TV-MM	0.85	0.81	0.76	0.65	0.55	0.40	0.88	0.84	0.79	0.68	0.53	0.39
C-SALSA	**0.88**	**0.82**	**0.77**	**0.68**	**0.61**	**0.51**	0.91	0.86	0.81	0.67	0.56	0.42
[113]	**0.88**	0.78	0.75	0.63	0.57	**0.51**	0.87	0.83	0.81	0.66	0.67	0.59
SURE-LET	**0.88**	0.81	**0.77**	0.67	**0.61**	0.50	**0.92**	**0.87**	**0.84**	**0.75**	**0.70**	**0.61**
MSE-LET	*0.89*	*0.82*	*0.78*	*0.68*	*0.62*	*0.52*	*0.92*	*0.87*	*0.84*	*0.76*	*0.70*	*0.61*
Image	*Bridge* 512 × 512						*Mixture* 512 × 512					
Input	0.61	0.60	0.57	0.40	0.28	0.12	0.56	0.53	0.46	0.26	0.17	0.09
ForWaRD	**0.84**	0.74	0.68	**0.60**	0.53	0.38	0.71	0.68	**0.64**	**0.54**	**0.49**	0.35
SA-DCT	**0.84**	0.75	0.68	**0.60**	0.53	0.38	0.71	0.68	**0.64**	**0.54**	**0.49**	0.37
BM3D	**0.84**	0.75	**0.70**	0.59	0.52	0.40	**0.74**	0.67	0.62	0.53	0.48	0.43
TV-MM	0.80	0.70	0.64	0.53	0.42	0.35	0.67	0.59	0.54	0.47	0.33	0.22
C-SALSA	**0.84**	**0.76**	0.67	**0.60**	0.53	0.40	0.69	0.62	0.57	0.44	0.35	0.25
[113]	0.82	0.72	0.68	0.59	0.53	0.42	0.69	0.62	0.57	0.53	0.39	0.40
SURE-LET	**0.84**	0.75	**0.70**	**0.60**	**0.54**	**0.44**	**0.74**	**0.68**	**0.64**	**0.54**	**0.49**	**0.44**
MSE-LET	*0.84*	*0.76*	*0.71*	*0.60*	*0.54*	*0.45*	*0.75*	*0.68*	*0.64*	*0.54*	*0.49*	*0.44*

*The results have been averaged over 10 noise realizations.

Table 3.7: *SSIM Comparison of some state-of-the-art deconvolution methods under Gaussian blur with variance 9**.

particular, significant improvements over other deconvolution algorithms are frequently observed for large noise variance.

Regarding the subjective visual quality (see Figures. 3.11–3.13), the proposed deconvolution algorithm preserves various image details, while introducing very few artifacts. For *House*, it achieves good preservation of uniform areas and regularly-sharp edges,

σ	1	5	10	30	50	100	1	5	10	30	50	100
Blur	**Rational filter**						**Separable filter**					
Input	22.60	22.33	21.55	17.15	13.58	7.98	25.78	25.22	23.82	17.85	13.88	8.06
ForWaRD	28.32	24.65	23.37	21.48	20.20	18.60	29.09	26.47	25.04	22.77	21.74	19.75
SA-DCT	**28.94**	25.00	**23.74**	21.99	20.99	19.02	29.23	**26.97**	**25.68**	**23.43**	22.40	20.77
BM3D	**28.95**	**25.12**	23.70	21.87	21.05	19.67	**29.40**	**27.00**	**25.72**	23.37	22.36	20.43
TV-MM	28.64	**25.10**	23.42	20.16	18.71	17.32	28.95	**26.93**	**25.70**	22.98	20.87	19.28
C-SALSA	28.63	24.90	23.49	20.45	18.82	18.31	28.96	26.57	25.34	22.44	20.46	16.46
[113]	28.44	24.87	23.53	21.71	20.91	19.92	29.03	26.70	25.48	23.25	22.16	20.61
SURE-LET	**28.87**	**25.10**	**23.81**	**22.12**	**21.32**	**20.19**	**29.37**	**27.00**	**25.70**	**23.50**	**22.52**	**21.04**
MSE-LET	*28.94*	*25.23*	*23.95*	*22.22*	*21.44*	*20.32*	*29.48*	*27.07*	*25.80*	*23.58*	*22.59*	*21.16*
Blur	5 × 5 **uniform blur**						9 × 9 **uniform blur**					
Input	23.24	22.92	22.04	17.33	13.66	8.01	21.16	20.96	20.38	16.69	13.37	7.93
ForWaRD	28.31	25.15	23.89	22.06	20.11	18.82	25.77	23.48	22.65	21.21	20.08	18.63
SA-DCT	27.96	24.76	23.82	22.32	21.43	19.86	25.83	23.31	22.48	20.82	19.71	18.23
BM3D	**28.57**	**25.50**	**24.46**	22.65	21.88	20.21	**26.12**	**23.75**	22.82	21.50	20.90	19.70
TV-MM	28.31	**25.52**	**24.41**	21.74	19.97	18.24	25.85	23.57	22.43	19.95	18.39	16.44
C-SALSA	28.36	25.16	24.15	21.78	18.59	17.48	25.74	23.42	22.58	20.67	19.52	18.00
[113]	28.25	25.20	24.17	22.52	21.74	20.41	25.82	23.39	22.47	21.32	20.83	19.96
SURE-LET	**28.65**	**25.55**	**24.43**	**22.78**	**22.00**	**20.78**	**26.10**	**23.79**	**22.95**	**21.71**	**21.10**	**20.13**
MSE-LET	*28.69*	*25.64*	*24.56*	*22.90*	*22.11*	*20.88*	*26.17*	*23.88*	*23.05*	*21.81*	*21.19*	*20.24*

*Best PSNR results within a 0.1dB margin are highlighted. The results have been averaged over 10 noise realizations.

Table 3.8: *PSNR performance of Bridge image for various blurs and noise levels**.

σ	1	5	10	30	50	100	1	5	10	30	50	100
Blur	**Rational filter**						**Separable filter**					
Input	0.73	0.72	0.69	0.49	0.34	0.15	0.93	0.91	0.88	0.65	0.46	0.22
ForWaRD	**0.97**	0.86	0.78	0.60	0.50	0.41	0.97	0.93	0.90	0.74	0.64	0.51
SA-DCT	**0.97**	0.86	0.79	0.62	0.53	0.40	**0.99**	**0.95**	**0.91**	**0.76**	0.66	0.50
BM3D	**0.97**	0.87	0.79	0.62	0.54	0.40	**0.99**	**0.95**	**0.91**	**0.76**	0.66	0.49
TV-MM	**0.97**	0.87	0.78	0.58	0.46	0.42	**0.99**	**0.95**	**0.91**	**0.76**	0.65	0.48
C-SALSA	**0.97**	**0.88**	0.80	0.58	0.46	0.43	**0.99**	0.94	0.90	0.74	0.65	0.47
[113]	0.96	**0.88**	0.78	0.60	0.52	0.43	**0.99**	0.94	0.90	0.75	0.66	0.51
SURE-LET	**0.97**	**0.88**	**0.80**	**0.64**	**0.57**	**0.46**	**0.99**	**0.95**	**0.91**	**0.76**	**0.68**	**0.54**
MSE-LET	*0.97*	*0.88*	*0.80*	*0.65*	*0.58*	*0.47*	*0.99*	*0.95*	*0.91*	*0.77*	*0.68*	*0.56*
Blur	5 × 5 **uniform blur**						9 × 9 **uniform blur**					
Input	0.81	0.80	0.77	0.56	0.40	0.18	0.81	0.72	0.67	0.51	0.40	0.18
ForWaRD	**0.96**	0.88	0.83	0.70	0.61	0.44	**0.90**	0.78	**0.72**	0.58	0.51	0.39
SA-DCT	0.95	0.86	0.83	0.71	0.60	0.45	**0.90**	0.77	**0.72**	0.58	0.49	0.39
BM3D	**0.96**	**0.90**	**0.85**	0.71	0.62	0.46	**0.90**	0.78	**0.72**	0.60	0.53	0.41
TV-MM	**0.96**	**0.90**	**0.85**	0.69	0.58	0.44	**0.90**	0.78	**0.72**	0.56	0.47	0.33
C-SALSA	**0.96**	0.88	**0.85**	0.69	0.53	0.49	**0.90**	0.78	**0.72**	0.58	0.49	0.38
[113]	**0.96**	0.88	**0.85**	0.71	0.62	0.50	**0.90**	0.78	**0.72**	0.60	0.52	0.43
SURE-LET	**0.96**	**0.90**	**0.85**	**0.72**	**0.64**	**0.52**	**0.90**	**0.79**	**0.72**	**0.61**	**0.55**	**0.45**
MSE-LET	*0.96*	*0.90*	*0.85*	*0.72*	*0.65*	*0.53*	*0.91*	*0.79*	*0.73*	*0.62*	*0.56*	*0.46*

*The results have been averaged over 10 noise realizations.

Table 3.9: *SSIM performance of Bridge image for various blurs and noise levels**.

while for *Crowd*, it preserves the finer details of the irregularities. In Figure 3.13, the differences between the various algorithms are clearly visible: the proposed algorithm introduces fewer artifacts than the other techniques. In particular, the text part of

σ	1	5	10	30	50	100	1	5	10	30	50	100
Blur	**Rational filter**						**Separable filter**					
Input	15.96	15.89	15.70	14.07	11.95	7.47	18.38	18.27	17.94	15.47	12.76	7.74
ForWaRD	27.67	19.95	17.77	15.76	15.01	13.99	26.01	22.03	20.42	17.50	16.30	14.96
SA-DCT	28.54	20.64	18.20	15.99	15.32	14.40	24.35	22.25	20.95	17.93	16.70	15.46
BM3D	28.53	19.24	17.25	15.45	14.85	14.10	26.54	22.26	20.04	17.14	16.15	14.86
TVMM	28.28	20.40	17.30	14.41	13.78	13.18	27.17	22.80	20.64	17.15	15.25	13.80
C-SALSA	27.26	20.14	18.04	15.80	15.02	13.99	26.58	21.75	20.16	17.51	16.19	13.80
[113]	25.92	19.52	17.74	15.77	15.12	14.42	24.67	21.40	20.01	17.67	16.67	15.41
SURE-LET	**29.90**	**21.10**	**18.53**	**16.12**	**15.50**	**14.71**	**27.80**	**23.13**	**21.08**	**18.05**	**16.94**	**15.70**
MSE-LET	*30.28*	*21.34*	*18.70*	*16.27*	*15.58*	*14.78*	*28.11*	*23.40*	*21.43*	*18.21*	*17.06*	*15.79*
Blur	5×5 **uniform blur**						9×9 **uniform blur**					
Input	15.92	15.86	15.67	14.04	11.94	7.46	14.58	14.54	14.40	13.13	11.35	7.24
ForWaRD	24.76	19.80	18.26	16.29	15.58	14.42	20.42	17.21	16.16	15.20	14.68	13.79
SA-DCT	25.35	19.40	17.57	16.03	15.48	14.74	21.29	17.10	16.13	15.06	14.60	13.93
BM3D	25.53	19.78	17.90	16.18	15.56	14.58	20.66	17.07	16.01	15.00	14.68	14.09
TVMM	25.70	20.39	18.31	15.66	14.52	13.64	20.70	17.02	15.65	14.23	13.66	12.87
C-SALSA	24.29	19.67	18.15	15.77	15.37	14.00	20.24	17.23	16.30	14.95	14.29	13.30
[113]	23.19	19.14	17.83	16.37	15.78	15.00	19.39	16.67	15.92	14.99	14.70	14.36
SURE-LET	**26.34**	**20.75**	**18.90**	**16.74**	**16.04**	**15.25**	**21.60**	**17.75**	**16.65**	**15.32**	**15.01**	**14.52**
MSE-LET	*26.60*	*21.00*	*19.06*	*16.85*	*16.13*	*15.35*	*21.90*	*17.89*	*16.76*	*15.42*	*15.07*	*14.60*

*Best PSNR results within a 0.1dB margin are highlighted. The results have been averaged over 10 noise realizations.

Table 3.10: *PSNR performance of Mixture image for various blurs and noise levels**.

σ	1	5	10	30	50	100	1	5	10	30	50	100
Blur	**Rational filter**						**Separable filter**					
Input	0.66	0.63	0.57	0.35	0.26	0.15	0.84	0.81	0.74	0.51	0.40	0.26
ForWaRD	0.94	0.86	0.75	0.54	0.48	0.40	0.98	0.94	0.88	0.67	0.64	0.48
SA-DCT	**0.98**	0.87	0.77	0.53	0.50	0.43	0.98	0.94	0.90	0.69	0.64	0.50
BM3D	**0.98**	0.86	0.75	0.59	0.52	**0.45**	**0.99**	**0.95**	0.90	0.73	0.64	0.51
TV-MM	**0.98**	0.80	0.75	0.45	0.36	0.36	0.98	**0.95**	0.90	0.66	0.58	0.45
C-SALSA	0.92	0.75	0.63	0.49	0.40	0.28	0.98	0.90	0.86	0.66	0.54	0.37
[113]	0.89	0.86	0.75	0.49	0.45	0.42	0.98	0.92	0.88	0.66	0.65	0.50
SURE-LET	**0.98**	**0.89**	**0.80**	**0.61**	**0.53**	**0.45**	**0.99**	**0.95**	**0.91**	**0.77**	**0.67**	**0.53**
MSE-LET	*0.98*	*0.89*	*0.79*	*0.59*	*0.51*	*0.44*	*0.99*	*0.95*	*0.91*	*0.77*	*0.66*	*0.52*
Blur	5×5 **uniform blur**						9×9 **uniform blur**					
Input	0.68	0.65	0.58	0.36	0.27	0.16	0.53	0.51	0.44	0.24	0.16	0.08
ForWaRD	0.95	0.88	0.78	0.67	0.59	0.48	0.88	0.74	0.67	**0.55**	0.48	0.43
SA-DCT	0.95	0.88	0.80	0.67	0.59	0.48	0.88	0.74	0.67	**0.55**	0.48	0.43
BM3D	**0.97**	0.88	0.80	0.67	0.59	0.49	**0.89**	0.74	0.66	**0.55**	**0.50**	**0.44**
TV-MM	0.95	0.88	0.73	0.67	0.45	0.44	0.88	0.74	0.65	0.45	0.44	0.38
C-SALSA	0.95	0.81	0.70	0.57	0.49	0.34	0.80	0.65	0.56	0.38	0.30	0.22
[113]	0.95	0.86	0.77	0.67	0.60	0.48	0.86	0.67	0.67	0.47	0.48	0.43
SURE-LET	**0.97**	**0.89**	**0.83**	**0.69**	**0.61**	**0.50**	**0.89**	**0.76**	**0.68**	**0.55**	**0.50**	**0.44**
MSE-LET	*0.97*	*0.89*	*0.82*	*0.68*	*0.60*	*0.49*	*0.89*	*0.75*	*0.68*	*0.54*	*0.49*	*0.44*

*The results have been averaged over 10 noise realizations.

Table 3.11: *SSIM performance of Mixture image for various blurs and noise levels**.

Mixture is easier to read in the image restored by the proposed multi-Wiener SURE-LET.

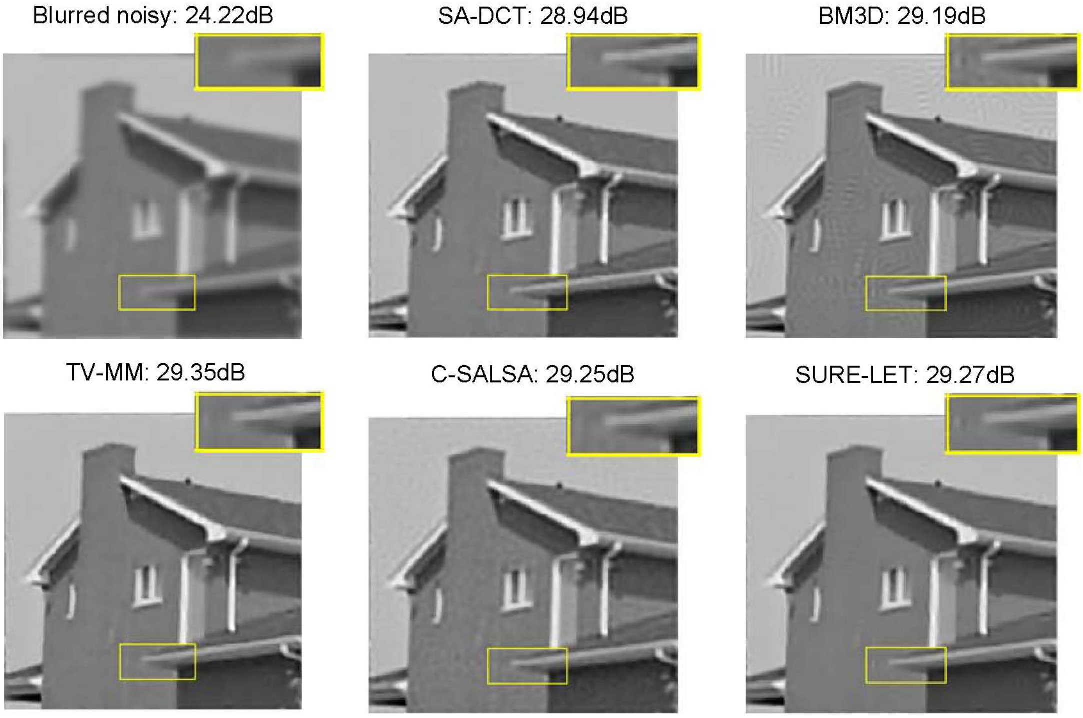


Figure 3.11: *Restoration of House degraded by Gaussian blur of variance 9 with noise std $\sigma = 1$.*

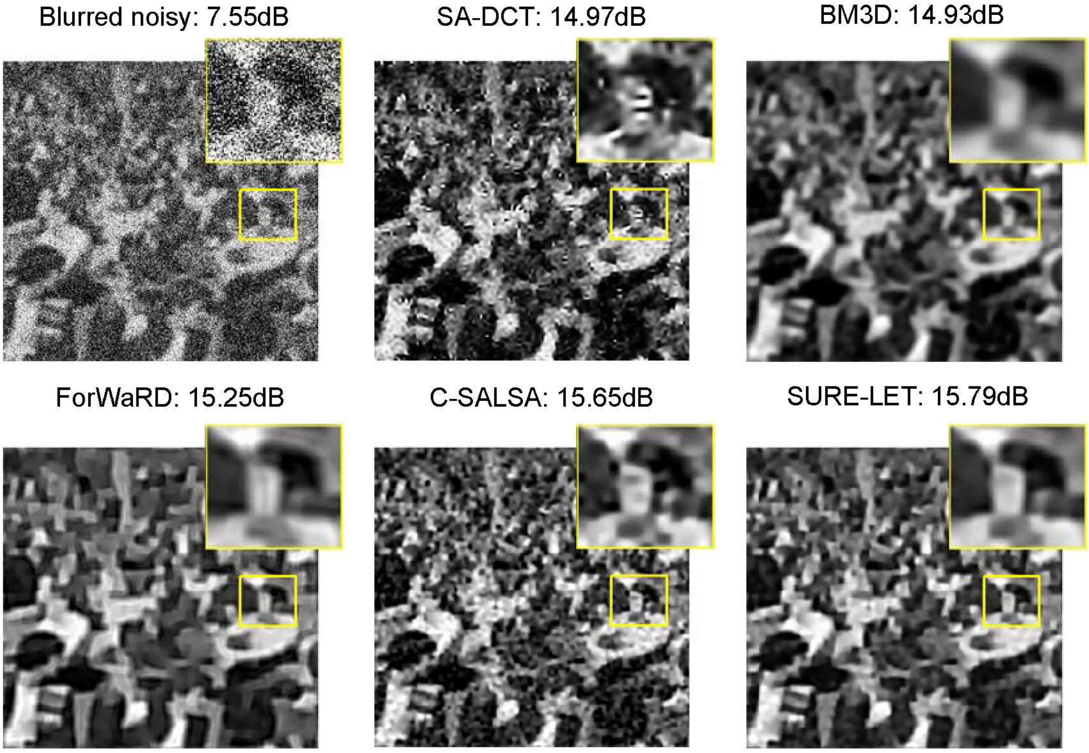


Figure 3.12: *Restoration of Crowd degraded by Gaussian blur of variance 9 with noise std $\sigma = 100$.*

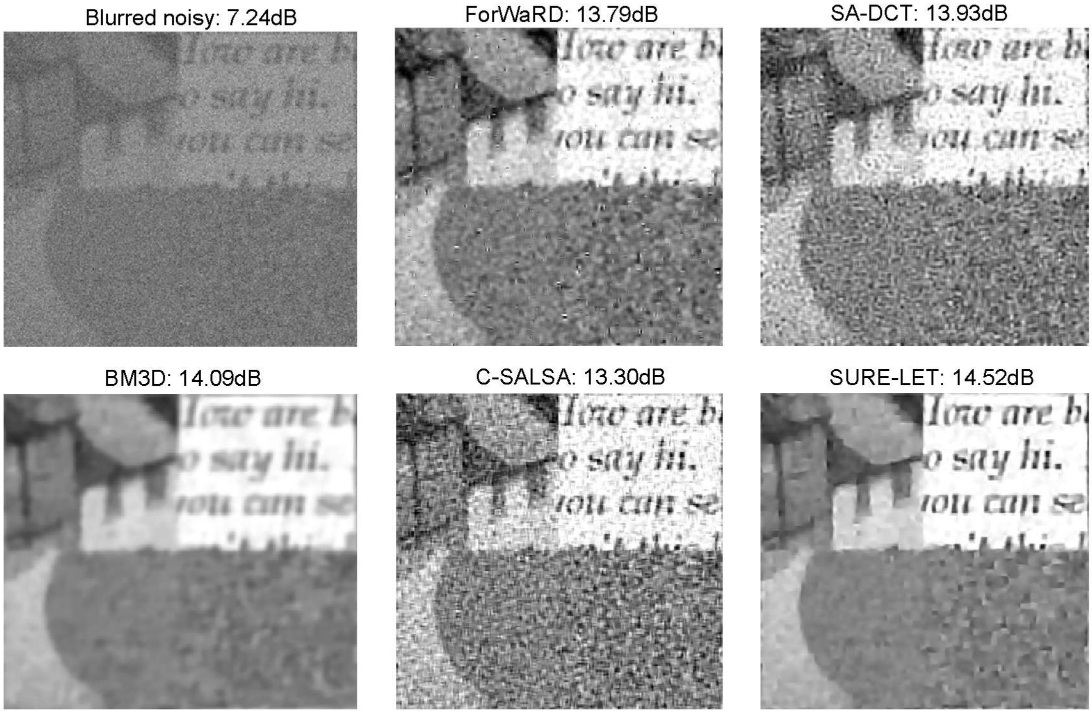


Figure 3.13: *Restoration of Mixture degraded by* 9×9 *uniform blur with noise std* $\sigma = 100$

3.6.6 Analysis of computational complexity

The most computationally-intensive part of the proposed algorithm is the construction of the elementary functions $\mathbf{f}_k(\mathbf{y})$, due to the computation of the thresholding $\boldsymbol{\theta}(\cdot)$ and of its first-order derivative, as well as the independent reconstructions of all the processed subbands. Compared to these steps, the construction and resolution of the linear system of equations shown as Eq.(2.9) of low order K has a negligible computational cost.

Based on these observations, the computational complexity of the proposed multi-Wiener SURE-LET is roughly evaluated as KN, which is linearly proportional to both the number of basis functions K and pixel number N. Since the processing time of the proposed deconvolution algorithm is independent of the content of the input data, it grows linearly with the data size. Tables 3.12 and 3.13 report the execution time of the various algorithms, measured on two hardware environments. As expected, the processing time of a 512×512 image is roughly four times that of a 256×256 image for the proposed algorithm.

Finally, we would like to emphasize the low complexity of the proposed algorithm.

Since our deconvolution approach merely boils down to solving a linear system of equations, it is substantially faster than other state-of-the-art techniques, as confirmed in Tables 3.12 and 3.13. We would also like to stress that our implementation uses standard Matlab code only, without any compiled routines, and can be easily parallelized for even faster processing.

Degradation scenario	ForWaRD [45]	SA-DCT [48]	BM3D [49]	TV-MM [43]	C-SALSA [38]	Pesquet [113]	SURE-LET (our work)
Cameraman **Gaussian blur** $\sigma = 1$	4.73	3.70	2.96	156.13	55.04	177.32	**1.36**
Mixture **Rational filter** $\sigma = 10$	41.03	16.18	13.65	289.14	55.72	540.22	**5.34**
Crowd 9×9 **uniform** $\sigma = 50$	59.95	16.13	12.37	202.89	42.03	513.68	**5.34**

[*] Hardware environment used: Intel(R) Core(TM)2 Duo CPU E7400 @2.80GHz, memory size: 3GB.

Table 3.12: *Comparison of the computational time of various deconvolution techniques (units: sec.)*.*

Degradation scenario	ForWaRD [45]	SA-DCT [48]	BM3D [49]	TV-MM [43]	C-SALSA [38]	Pesquet [113]	SURE-LET (our work)
Cameraman **Gaussian blur** $\sigma = 1$	3.62	2.81	1.90	103.91	33.52	114.12	**0.69**
Mixture **Rational filter** $\sigma = 10$	31.11	12.28	8.00	93.50	34.32	375.74	**2.69**
Crowd 9×9 **uniform** $\sigma = 50$	45.53	12.33	7.80	59.08	25.18	353.52	**2.69**

[*] Hardware environment used: Intel(R) Core(TM) i3-2100 CPU @3.10GHz, memory size: 2GB.

Table 3.13: *Comparison of the computational time of various deconvolution techniques (units: sec.)*.*

3.7 Conclusion

In this chapter, we have presented a new image deconvolution method based on the SURE-LET approach initially developed for image denoising applications. The main originality of the proposed deconvolution approach is the use of multiple Wiener filters with different but fixed regularization parameters, to avoid the empirical and fastidious adjustment of the non-linear regularization parameter.

Although the proposed multi-Wiener SURE-LET deconvolution algorithm has low complexity, its performances are already quite competitive with the state-of-the-art deconvolution techniques, both quantitatively and visually. The great flexibility and the limited computational cost of the proposed approach suggest that it is possible to develop more sophisticated forms of basic processings; e.g., performing a multivariate thresholding by taking into account inter-scale and/or intra-scale dependencies between the wavelet coefficients as in [110]. Electing more directional deconvolutions instead of using classic Wiener filters might also lead to improved performances. There is thus a substantial margin of improvement for SURE-LET type deconvolution algorithms.

Part II

The SURE-based Approach to Blind Deconvolution

Chapter 4

The Blur-SURE Framework to PSF Estimation

4.1 Introduction

As mentioned in Chapter 1, there are two categories of approaches to tackle blind image deconvolution. The first class performs joint estimation of original image and the point spread function (PSF), by minimizing an uniform objective functional [5; 82; 87; 88; 91; 124–126], whereas another category of approaches separate the blur identification and image restoration: the original image is estimated by non-blind deconvolution using the PSF estimated by the blur identification [90; 92; 127–130].

Recently, the separate estimation receives more attention, due to its several advantages over the joint estimation. As reported in [64], simultaneous MAP estimation of both latent image and blur kernel is guaranteed to fail even with infinitely large images sampled from the prior. On the other hand, a MAP estimation of the kernel alone is well constrained and is guaranteed to succeed to recover the true blur, because the number of parameters to estimate (i.e. kernel size) is small relative to the number of the image pixels measured. Another merit of separate estimation is that it allows to apply any developed high-quality deconvolution algorithm to estimate the original image, once the PSF is estimated. For these reasons, we choose to separately estimate the kernel and the image. Note that the PSF estimation is extremely significant for the deconvolution performance, as an inaccurately estimated kernel definitely results in great quality loss of the restored image, and many existing deconvolution algorithms are very sensitive to the accuracy of PSF estimation.

For the kernel estimation, it is often to formulate an objective functional, by incorporating prior knowledge of the natural image statistics and the blur kernel characteristics [64], and then perform alternating minimization. Regarding the image prior, due to the expected sparsity of the derivatives of the latent images [90; 91; 126], [90; 92] model the image gradient by a mixture of Gaussian distributions, [91] uses both global prior to model gradient density and local prior to suppress ringing artifacts, [126] utilizes a non-convex ℓ_p quasi norm with $0 < p < 1$ to better model the heavy tail of image gradient [131]. Since the motion kernel tends to be sparse, it is often characterized by a sparsity prior and enforced to be non-negative: [90; 91] assume the blur kernel to follow a mixture of exponential distributions, [82; 125; 126] use the totoal-variation prior for the blur kernel. In addition, kernel can also be estimated using the sharp edge prediction [129; 130].

4.2 Problem statement

Consider the linear observation model

$$\mathbf{y} = \mathbf{H}_0\mathbf{x} + \mathbf{b} \tag{4.1}$$

where $\mathbf{y} \in \mathbb{R}^N$ is the observed data of the original (unknown) $\mathbf{x} \in \mathbb{R}^N$, the latent true (unknown) matrix $\mathbf{H}_0$ denotes a linear distortion, the vector $\mathbf{b} \in \mathbb{R}^N$ is a zero-mean additive Gaussian noise corruption with covariance matrix $\mathbf{C} \succ 0$. The objective is to find a good estimate of $\mathbf{x}$. As mentioned in Section 4.1, our basic procedure consists of first estimating the distortion matrix $\mathbf{H}_0$, and then finding the estimate of $\mathbf{x}$ with the estimated $\mathbf{H}$.

In this chapter, our focus is to estimate the matrix $\mathbf{H}_0$, such that the estimated $\mathbf{H}$ is as close to the true $\mathbf{H}_0$ as possible, from the observed data $\mathbf{y}$ only. Note that our proposed theoretical framework is not limited to convolution operation: the approach is applicable for any distortion matrix $\mathbf{H}_0$.

4.3 The blur-SURE framework for general linear model

4.3.1 Blur-MSE: a modified version of MSE

Denoting a function (or processing) by $\mathbf{f} : \mathbb{R}^N \to \mathbb{R}^N$, applied to the observed data $\mathbf{y}$, the standard MSE is defined as Eq.(2.2) [1; 110; 112; 115]. To estimate the matrix $\mathbf{H}_0$, instead of the standard MSE, we consider the following blurred (filtered or distorted) version:

$$\text{blur-MSE} = \frac{1}{N}\mathscr{E}\left\{\left\|\mathbf{H}\mathbf{f}(\mathbf{y}) - \mathbf{H}_0\mathbf{x}\right\|^2\right\} \tag{4.2}$$

as the objective functional, which measures the difference between two distorted (blurred) data: $\mathbf{Hf}(\mathbf{y})$ and $\mathbf{H}_0\mathbf{x}$. Our approach, to be described below, consists in restricting $\mathbf{f}$ to a subclass of processings — typically, Wiener filterings — that depend on $\mathbf{H}$, and then minimizing an estimate of this blur-MSE over $\mathbf{H}$.

4.3.2 Blur-MSE minimization

We are now going to restrict ourselves to the linear processing denoted by $\mathbf{f}(\mathbf{y}) = \mathbf{W_H y}$, where the notation $\mathbf{W_H}$ emphasizes that the matrix $\mathbf{W}$ is related to the matrix $\mathbf{H}$. It is well-known that for the linear model denoted by (2.1) with the known matrix $\mathbf{H}$, the ideal linear processing $\mathbf{W_H}$ that minimizes the standard MSE given as $\frac{1}{N}\left\|\mathbf{W_H y}-\mathbf{x}\right\|^2$ is Wiener filtering, expressed as [25]:

$$\mathbf{W_H} = \mathbf{SH}^{\mathrm{T}}\big(\mathbf{HSH}^{\mathrm{T}} + \mathbf{C}\big)^{-1} = \big(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \mathbf{H}^{\mathrm{T}}\mathbf{CH}^{-\mathrm{T}}\mathbf{S}^{-1}\big)^{-1}\mathbf{H}^{\mathrm{T}} \tag{4.3}$$

in matrix notation, where the covariance matrices $\mathbf{S} = \mathscr{E}\{\mathbf{xx}^{\mathrm{T}}\}$, $\mathbf{C} = \mathscr{E}\{\mathbf{bb}^{\mathrm{T}}\}$. Now, if we base our processing on Wiener filtering, then the following theorem shows that the matrix $\mathbf{H}$ that minimizes the blur-MSE is related in a simple way to the original matrix $\mathbf{H}_0$.

Theorem 4.1 Consider only linear processings of the form $\mathbf{f}(\mathbf{y}) = \mathbf{W_H y}$, where $\mathbf{W_H}$ is defined as Eq.(4.3). The minimization of the blur-MSE defined by (4.2) over $\mathbf{H}$:

$$\min_{\mathbf{H}} \frac{1}{N}\mathscr{E}\Big\{\left\|\mathbf{HW_H y} - \mathbf{H}_0\mathbf{x}\right\|^2\Big\}, \tag{4.4}$$

yields the solution $\mathbf{H}$ to Eq.(4.4), which satisfies $\mathbf{HSH}^{\mathrm{T}} = \mathbf{H}_0\mathbf{SH}_0^{\mathrm{T}}$.

See Appendix A.7 for the proof, which is similar to Wiener theory for performing denoising/deconvolution [25]. Note that Theorem 4.1 is valid for any linear distortion $\mathbf{H}$, not limited to convolution operation. This statement is also valid for rectangular matrices $\mathbf{H} \in \mathbb{R}^{N\times M}$. For the sake of brevity, the following discussions are restricted to square matrix.

4.3.3 Blur-SURE: an unbiased estimate of the blur-MSE

Notice that we cannot directly minimize the blur-MSE, as $\mathbf{H}_0\mathbf{x}$ is unknown in practice. However, based on the linear model given by Eq.(4.1), the quantity of blur-MSE can be replaced by a statistical estimate — blur-SURE, involving only the measurements $\mathbf{y}$, as summarized in the following theorem.

Theorem 4.2 Given the linear model as Eq.(4.1), the following random variable:

$$\tilde{\epsilon} = \frac{1}{N}\Big\{\big\|\mathbf{Hf}(\mathbf{y}) - \mathbf{y}\big\|^2 + 2\mathrm{div}_{\mathbf{y}}\big(\mathbf{CHf}(\mathbf{y})\big) - \mathrm{Tr}\big(\mathbf{C}\big)\Big\} \tag{4.5}$$

is an unbiased estimator of the blur-MSE denoted by Eq.(4.2), i.e. $\mathscr{E}\{\epsilon\} = \frac{1}{N}\mathscr{E}\Big\{\big\|\mathbf{Hf}(\mathbf{y}) - \mathbf{H}_0\mathbf{x}\big\|^2\Big\}$, where Tr denotes matrix trace, $\mathbf{C}$ is covariance matrix of noise $\mathbf{b}$, the divergence $\mathrm{div}_{\mathbf{y}}\mathbf{v} = \sum_{n=1}^{N}\frac{\partial v_n}{\partial y_n}$ for $\forall \mathbf{v} \in \mathbb{R}^N$.

See Appendix A.8 for the proof, which is similar to the proofs of Stein's lemma [109] and the standard SURE for the linear model [115]. Note that as same as Theorem 4.1, this theorem is also valid for any linear distortion $\mathbf{H}$ (rectangular or square matrix) and any processing $\mathbf{f}$, under the assumption of additive Gaussian noise $\mathbf{b}$. In particular, if the Gaussian noise $\mathbf{b}$ is independent and identically distributed (i.i.d.) with variance σ^2, the blur-SURE becomes:

$$\tilde{\epsilon} = \frac{1}{N}\big\|\mathbf{Hf}(\mathbf{y}) - \mathbf{y}\big\|^2 + \frac{2\sigma^2}{N}\mathrm{div}_{\mathbf{y}}\big(\mathbf{Hf}(\mathbf{y})\big) - \sigma^2 \tag{4.6}$$

which results from specifying the covariance matrix $\mathbf{C} = \sigma^2\mathbf{I}$ in Eq.(4.5).

As blur-SURE involves only the known measurements $\mathbf{y}$, we can directly minimize blur-SURE instead of minimizing blur-MSE in practice.

4.4 Application of blur-SURE framework for PSF estimation

4.4.1 Problem statement in the context of convolution

In the context of image deconvolution, Figure 4.1 shows an example of image degradation shown as Eq.(4.1), where the distortion matrix $\mathbf{H}_0$ denotes the convolution operation performed by point spread function (PSF) or kernel, which results in blurred observations. It also shows our basic procedure of blind deconvolution: PSF estimation followed by non-blind deconvolution. Now, we can apply the blur-SURE framework to estimate the unknown PSF.

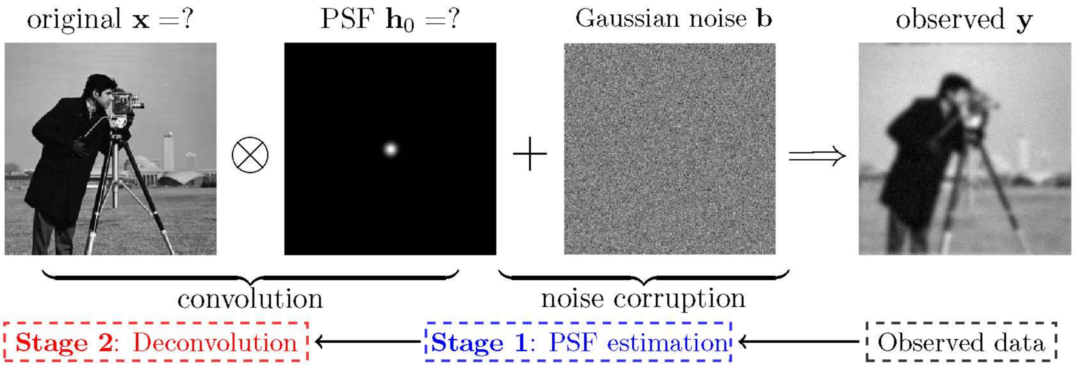


Figure 4.1: *The convolution example of the linear distortion model shown as Eq.(4.1) and the basic procedure of blind deconvolution: PSF estimation followed by non-blind deconvolution.*

4.4.2 Blur-MSE minimization for PSF estimation

Section 4.3.2 has presented the concept of blur-MSE, and Theorem 4.1 has shown that the blur-MSE minimization with Wiener filtering yields a very simple relation $\mathbf{HSH} = \mathbf{H}_0\mathbf{SH}_0$. In the case of convolution matrix $\mathbf{H}_0$, Theorem 4.1 can be further simplified as the following corollary — a Fourier description of Theorem 4.1.

Corollary 4.3 If both the signal $\mathbf{x}$ and the noise $\mathbf{b}$ are stationary, the square matrix $\mathbf{H}$ denotes convolution, the Wiener filtering $\mathbf{W_H}$ defined as Eq.(4.3) can be rewritten as:

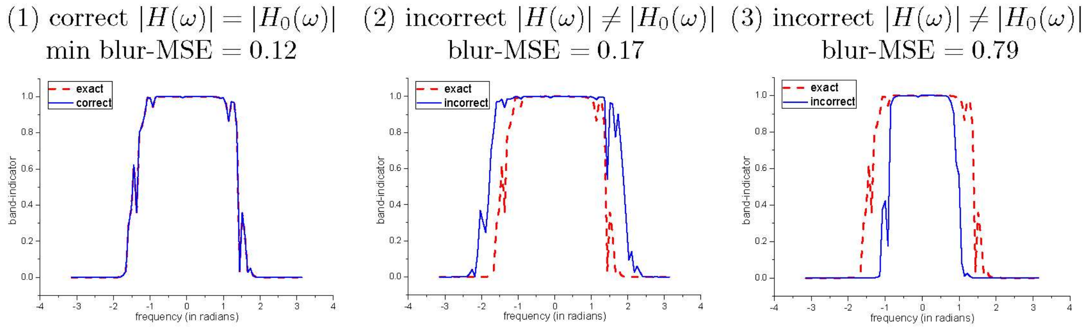
$$W_H(\omega) = \frac{H^*(\omega)}{|H(\omega)|^2 + C(\omega)/S(\omega)} \tag{4.7}$$

in Fourier domain, where $H(\omega)$ is the Fourier representation of $\mathbf{H}$, $S(\omega)$ and $C(\omega)$ are the mean power spectral densities of signal $\mathbf{x}$ and noise $\mathbf{b}$, respectively. Thus, the minimization of the blur-MSE denoted by Eq.(4.4) yields the solution $H(\omega)$, which satisfies:

$$\underbrace{H(\omega) \cdot \overbrace{\frac{H^*(\omega)}{\big|H(\omega)\big|^2 + C(\omega)/S(\omega)}}^{W_H(\omega)}}_{U(\omega)} = \underbrace{H_0(\omega) \cdot \overbrace{\frac{H_0^*(\omega)}{\big|H_0(\omega)\big|^2 + C(\omega)/S(\omega)}}^{W_{H_0}(\omega)}}_{U_0(\omega)} \quad \text{for } \forall\omega \tag{4.8}$$

which implies that $\big|H(\omega)\big| = \big|H_0(\omega)\big|$ for $\forall\omega$.

This corollary is easily proved by the fact that under convolution operation and the assumption of stationary process [25], all the matrices involved in the blur-MSE



(1) correct $|H(\omega)| = |H_0(\omega)|$ min blur-MSE $= 0.12$ (2) incorrect $|H(\omega)| \neq |H_0(\omega)|$ blur-MSE $= 0.17$ (3) incorrect $|H(\omega)| \neq |H_0(\omega)|$ blur-MSE $= 0.79$

Figure 4.2: *The behaviours of band-indicator $U(\omega)$ in (4.8): correct $|H(\omega)| = |H_0(\omega)|$ yields the minimum blur-MSE, shown in (1).*

become circulant, thus can be diagonalized by Fourier transform.

Let us call $U(\omega) = H(\omega)W_H(\omega)$ in Eq.(4.8) as *frequency-band indicator* or *band indicator* for short[1], as it marks a certain frequency band as 0 or 1 with a narrow transition between the two values (see Figure 4.2 for example). Corollary 4.3 states that: (1) the blur-MSE minimization is essentially equivalent to matching the band indicator $U(\omega)$ to the true $U_0(\omega)$ in Fourier domain; (2) the blur-MSE minimization can only derives the equality of magnitude frequency response of the PSF: $|H(\omega)| = |H_0(\omega)|$, which is irrelevant to the phase response. Hence, if the frequency response of the PSF has positive value with zero phase shift at any frequency ω, the blur-MSE minimization succeeds in estimating accurate PSF, by seeking its frequency response.

To exemplify this corollary, Figure 4.2 shows the band-indicator matching as Eq.(4.8). We can see that the minimum blur-MSE is reached as the two band-indicators match perfectly $U(\omega) = U_0(\omega)$, and the derived equality $|H(\omega)| = |H_0(\omega)|$ holds, as shown in Figure 4.2-(1). If $|H(\omega)| \neq |H_0(\omega)|$, the corresponding band-indicator $U(\omega)$ does not match the exact $U_0(\omega)$ well, and the blur-MSE does not reach its minimum, as shown in Figure 4.2-(2,3).

4.4.3 Approximation of exact Wiener filtering

The blur-MSE, incorporating the exact Wiener filtering as the processing $\mathbf{f}$, has been justified as a reliable objective functional both theoretically and experimentally. However, note that the exact Wiener filtering cannot be used in practice, since the signal-to-noise ratio (SNR) $S(\omega)/C(\omega)$ is unknown. Consider the basic observation that for

[1] Regarding the terminology of $U(\omega)$, refer to [132], where the authors named $U(\omega)$ as *combined response*.

natural images with strong low frequencies and weak high frequencies, $C(\omega)/S(\omega)$ increases as the frequency ω goes up, since the noise content is often relatively flat with frequency. We replace $C(\omega)/S(\omega)$ by $\lambda\|\omega\|^2$, where the parameter λ needs to be optimized, such that $C(\omega)/S(\omega) \approx \lambda\|\omega\|^2$. Thus, we summarize the notations of exact and approximated Wiener filterings as Table 4.1.

Notations	Exact	Approximation
Matrix	$\mathbf{W_H} = \big(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \mathbf{H}^{\mathrm{T}}\mathbf{C}\mathbf{H}^{-\mathrm{T}}\mathbf{S}^{-1}\big)^{-1}\mathbf{H}^{\mathrm{T}}$	$\mathbf{W}_{\mathbf{H},\lambda} = \big(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda\mathbf{R}\big)^{-1}\mathbf{H}^{\mathrm{T}}$
Frequency	$W_H(\omega) = \frac{H^*(\omega)}{\|H(\omega)\|^2+C(\omega)/S(\omega)}$	$W_{H,\lambda}(\omega) = \frac{H^*(\omega)}{\|H(\omega)\|^2+\lambda\|\omega\|^2}$

Table 4.1: *Notations of exact and approximated Wiener filterings: $\lambda\mathbf{R}$ is a circulant matrix with eigenvalues $\lambda\|\omega\|^2$, to approximate the matrix $\mathbf{H}^{\mathrm{T}}\mathbf{C}\mathbf{H}^{-\mathrm{T}}\mathbf{S}^{-1}$ in exact Wiener filtering Eqs.(4.3) and (4.7).*

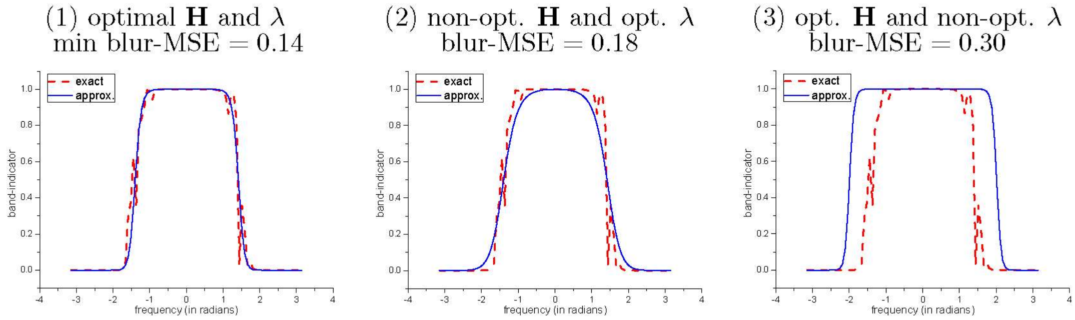
Due to the regularization parameter λ introduced, we formulate the PSF estimation as joint minimization of the blur-MSE over both $\mathbf{H}$ and λ, i.e. $\min_{\mathbf{H},\lambda} \frac{1}{N}\mathscr{E}\Big\{\|\mathbf{H}\mathbf{W}_{\mathbf{H},\lambda}\mathbf{y} - \mathbf{H}_0\mathbf{x}\|^2\Big\}$. By Corollary 4.3, the solution $H(\omega)$ satisfies $\big|H(\omega)\big| \approx \big|H_0(\omega)\big|$, and also, with the obtained λ, constitutes the best approximation of the exact band indicator $U_0(\omega)$, shown as Eq.(4.9). Table 4.2 lists all the notations of band indicators.

$$\underbrace{U_{H,\lambda}(\omega) = \frac{|H(\omega)|^2}{|H(\omega)|^2 + \lambda\|\omega\|^2}}_{\text{approximated band-indicator}} \approx \underbrace{\frac{\big|H_0(\omega)\big|^2}{\big|H_0(\omega)\big|^2 + C(\omega)/S(\omega)} = U_0(\omega)}_{\text{exact band-indicator}} \tag{4.9}$$

Notations	Exact	Approximation
Matrix	$\mathbf{U_H} = \mathbf{H}\big(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \mathbf{H}^{\mathrm{T}}\mathbf{C}\mathbf{H}^{-\mathrm{T}}\mathbf{S}^{-1}\big)^{-1}\mathbf{H}^{\mathrm{T}}$	$\mathbf{U}_{\mathbf{H},\lambda} = \mathbf{H}\big(\mathbf{H}^{\mathrm{T}}\mathbf{H} + \lambda\mathbf{R}\big)^{-1}\mathbf{H}^{\mathrm{T}}$
Frequency	$U_H(\omega) = \frac{\|H(\omega)\|^2}{\|H(\omega)\|^2+C(\omega)/S(\omega)}$	$U_{H,\lambda}(\omega) = \frac{\|H(\omega)\|^2}{\|H(\omega)\|^2+\lambda\|\omega\|^2}$

Table 4.2: *Notations of exact and approximated band indicators.*

Figure 4.3 shows the different approximations of the band-indicator. We can see that only both the $\mathbf{H}$ and λ obtained by minimizing the blur-MSE yield the best approximation.



(1) optimal $\mathbf{H}$ and λ min blur-MSE $= 0.14$

(2) non-opt. $\mathbf{H}$ and opt. λ blur-MSE $= 0.18$

(3) opt. $\mathbf{H}$ and non-opt. λ blur-MSE $= 0.30$

Figure 4.3: *The approximation of band-indicator as Eq.(4.9): the blur-MSE minimization yields the best approximation of band-indicator.*

4.4.4 Blur-SURE minimization for PSF estimation

Recalling the blur-SURE formulation given by Eq.(4.6), if the processing $\mathbf{f}$ is the approximated Wiener filtering specified in Table 4.1, we obtain the objective functional:

$$\tilde{\epsilon} = \frac{1}{N}\left\|\mathbf{H}\mathbf{W}_{\mathbf{H},\lambda}\mathbf{y} - \mathbf{y}\right\|^2 + \frac{2\sigma^2}{N}\mathrm{Tr}\big(\mathbf{H}\mathbf{W}_{\mathbf{H},\lambda}\big) - \sigma^2 \tag{4.10}$$

as the unbiased estimate of the blur-MSE, given as $\frac{1}{N}\left\|\mathbf{H}\mathbf{W}_{\mathbf{H},\lambda}\mathbf{y} - \mathbf{H}_0\mathbf{x}\right\|^2$. Note that $\mathrm{div}_{\mathbf{y}}(\mathbf{H}\mathbf{W}_{\mathbf{H},\lambda}\mathbf{y}) = \mathrm{Tr}(\mathbf{H}\mathbf{W}_{\mathbf{H},\lambda})$. Taking three filters for example:

- **finite rational filter** $h_0(i,j) = (1+i^2+j^2)^{-1}$ for $i,j = -7,...,0,...,7$;
- **separable filter** with filter $[1,4,6,4,1]/16$ along both horizontal and vertical directions;
- 9×9 **uniform filter.**

Figure 4.4 shows the curves of the blur-MSE and the blur-SURE under different noise variance. It demonstrates that the blur-SURE is very close to the blur-MSE in various situations.

4.5 Concluding remarks

Since the blur-SURE depends on the observed data $\mathbf{y}$ only, it is a good substitute for the blur-MSE in practice. Also, it has been justified that minimization of the blur-MSE yields the accurate estimate of PSF. Therefore, we formulate the PSF estimation as joint minimization of the blur-SURE over both $\mathbf{H}$ and λ. The flowchart shown in Figure 4.5 summarizes the blur-SURE framework. Also note that the blur-MSE minimization serves as an *oracle* counterpart of the blur-SURE minimization.

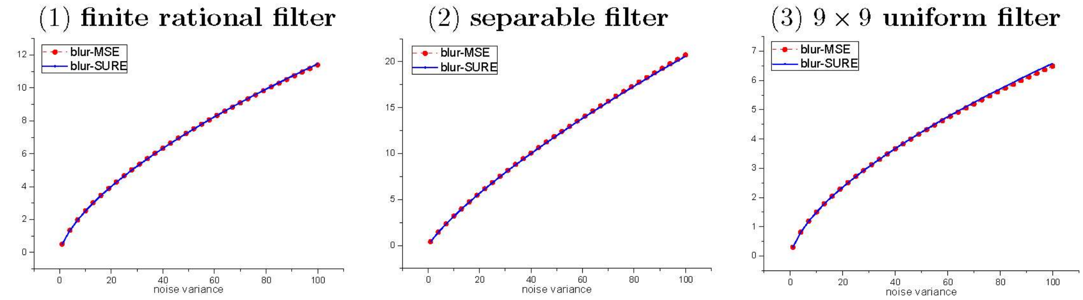


Figure 4.4: *The blur-MSE and blur-SURE denoted by Eq.(4.10) with the regularization parameter* $\lambda = 1 \times 10^{-4}\sigma^2$*: example of* Cameraman *blurred by three filters, under various noise variances* σ^2*.*

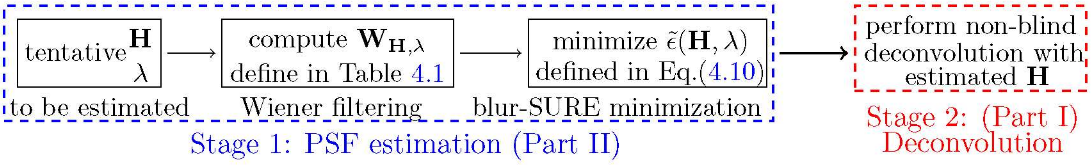

tentative $\mathbf{H}$ λ
to be estimated
compute $\mathbf{W}_{\mathbf{H},\lambda}$ define in Table 4.1
Wiener filtering
minimize $\tilde{\epsilon}(\mathbf{H}, \lambda)$ defined in Eq.(4.10)
blur-SURE minimization
Stage 1: PSF estimation (Part II)
perform non-blind deconvolution with estimated $\mathbf{H}$
Stage 2: (Part I) Deconvolution

Figure 4.5: *The procedure of PSF estimation: joint minimization of the blur-SURE over* $\mathbf{H}$ *and* λ*, as shown in Eq.(4.10).*

In this chapter, we proposed a new approach to estimate the distortion matrix, more specifically, the PSF, based on a new criterion — blur-SURE: a statistical estimate of blur-MSE. The simple Wiener filtering, as the linear solution of the blur-MSE minimization, can be well approximated using an optimal regularization parameter λ, which is jointly estimated by minimizing blur-SURE, along with the PSF.

Chapter 5

The Blur-SURE Approach to Parametric PSF Estimation

5.1 Introduction

5.1.1 Overview of parametric PSF estimation

Chapter 1 has described the necessity and advantage of parametric estimation of PSF in real applications, compared to non-parametric one. A great number of applications have also been listed, ranging from astronomical imaging to biomedical imaging, where the reliable parametric form of PSF is available: experimentally measured [133], practically assumed [17] or mathematically derived [134].

Given the parametric form of PSF, the PSF estimation problem boils down to estimate the small number of involved parameters, instead of estimate the pixels in the finite support of PSF. The most commonly used approach to PSF parameter estimation is to establish a certain objective functional, and perform alternating minimization between original image and PSF parameters, see [102; 105] for example. More recently, learning-based approach has been proposed to this parametric estimation [65].

5.1.2 Gaussian PSF as a typical example

Among all the parametric forms of the PSF shown in [3; 97; 102; 104], of particular interest is the Gaussian function, which can be used for modelling many degradation scenarios in real applications: [135; 136] studied the possibility of Gaussian approximation to microscopy PSF; [17; 22] used the Gaussian function to approximate the PSF of electro-optical systems under various imaging conditions; [99; 101; 137] used a Gaussian function to approximate the PSF caused by the atmospheric turbulence in astronomical imaging; [138–141] used the Gaussian function to model the defocus blur in photography. Thus, the estimation of Gaussian blur variance becomes particularly interesting.

Note that Gaussian blur size is difficult to estimate using traditional blur identification approach [93; 103], especially for those approaches relying on edge recognition, e.g. [93] fails to recover the image, since the frequency response of Gaussian blur decays so fast that all the high frequency content of the original image is eliminated. Despite of the difficulty, [138; 140; 141] tried to estimate the blur variance based on edge-based parametric representation [142]. [138] obtained the second derivative of a blurred edge by applying a steerable second derivative of Gaussian operator to the edge, thus, the distance between its extrema provides a measure of blur scale. Based on the work of [138], [140] improved the estimation results, by removing the estimate outliers and performing the blur propagation. Furthermore, [141] derived a closed-form solution of the edge blurriness based on [140]. However, these methods are very sensitive to the noise corruption: the edge localization and measurement become less accurate under higher noise level. In addition, the blur variance can also be estimated in regularization framework, e.g. [105; 143] uses Richardson-Lucy algorithm [74; 75] with TV regularization to perform joint estimation of the object and Gaussian blur variance; Wiener-based [10] or Richardson-Lucy-based [11] iterative methods project the updated PSF into the Gaussian family during each iterative step. Moreover, [3] detects the blur size by appropriate 1-D Fourier analysis of the observed image.

5.1.3 Outline of this chapter

Taking several typical parametric forms of PSF (including Gaussian kernel) for example, this chapter will exemplify and demonstrate the blur-SURE framework proposed in Chapter 4 in the following lines:

- to show the closeness of the blur-SURE to blur-MSE (as its *oracle* version);
- to demonstrate the high accuracy of the parametric estimation of PSF, under various blur types and noise levels;
- to develop a fast algorithm of the blur-SURE minimization.

5.2 Parametric estimation: problem formulation

Suppose the PSF is of known parametric form, such that the PSF is completely represented by only a few parameters $\mathbf{s} = [s_1, s_2, ..., s_P]^{\mathrm{T}}$ involved in the function form [65; 102]. Denote the latent true parameter by $\mathbf{s}_0$. Therefore, the blur-SURE minimization is performed over the PSF parameters $\mathbf{s}_0$ and λ:

$$\min_{\mathbf{s},\lambda} \underbrace{\frac{1}{N}\left\|\mathbf{H}_{\mathbf{s}}\mathbf{W}_{\mathbf{s},\lambda}\mathbf{y} - \mathbf{y}\right\|^2 + \frac{2\sigma^2}{N}\mathrm{Tr}(\mathbf{H}_{\mathbf{s}}\mathbf{W}_{\mathbf{s},\lambda}) - \sigma^2}_{\text{blur-SURE: } \tilde{\epsilon}(\mathbf{s},\lambda)} \tag{5.1}$$

where the notations of the convolution matrix $\mathbf{H}_{\mathbf{s}}$ and the Wiener filtering $\mathbf{W}_{\mathbf{s},\lambda}$ indicate their dependences on the parameters $\mathbf{s}$. Again, the solution pair $(\mathbf{s}, \lambda)$ constitutes the best approximation of band-indicator, shown as Eq.(4.9).

5.3 Examples of PSF parameter estimation

We consider the following typical types of PSF.

5.3.1 Gaussian kernel

The most typical PSF form denoted by:

$$\mathbf{h}_s(i,j) = C \cdot \exp\left(- \frac{i^2 + j^2}{2s^2} \right) \tag{5.2}$$

with variance s^2, where (i, j) denotes the 2-D coordinates, C is normalization coefficient such that $\sum_{i,j} \mathbf{h}_s(i,j) = 1$. s is the unknown PSF parameter to be estimated.

To experimentally justify Corollary 4.3, we use the exact Wiener filtering (Eqs.(4.3) and (4.7)) in the blur-MSE and blur-SURE minimizations, Figure 5.1 demonstrates that the estimated $s = s_0$ with zero error. If we apply the approximated Wiener filtering, Figure 5.2 shows that: (1) the blur-SURE is a reliable estimator of blur-MSE, and the estimated $s \approx s_0$, as shown in Figure 5.2-(a); (2) the band-indicator is well approximated with the optimal λ and s obtained by minimizing the blur-SURE, as shown in Figure 5.2-(b).

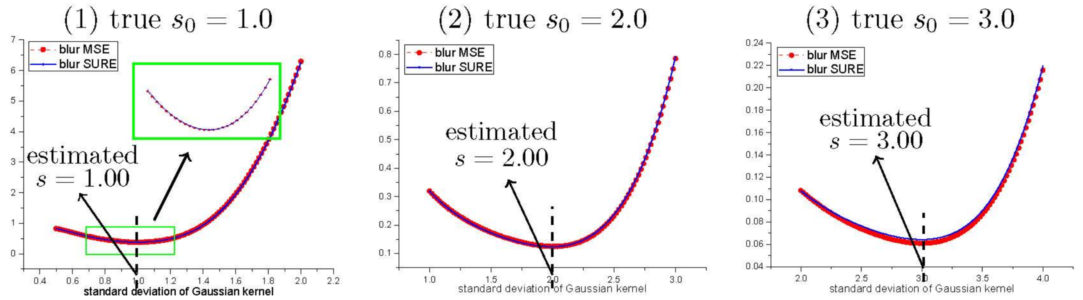


Figure 5.1: *The relationship between s and blur-SURE with exact Wiener filtering (Eqs.(4.3) and (4.7)) (also the blur-MSE as the oracle counterpart of $\tilde{\epsilon}(s)$): example of Cameraman blurred by true Gaussian kernel with standard deviations $s_0 = 1.0, 2.0, 3.0$ and noise variance $\sigma^2 = 1.0$.*

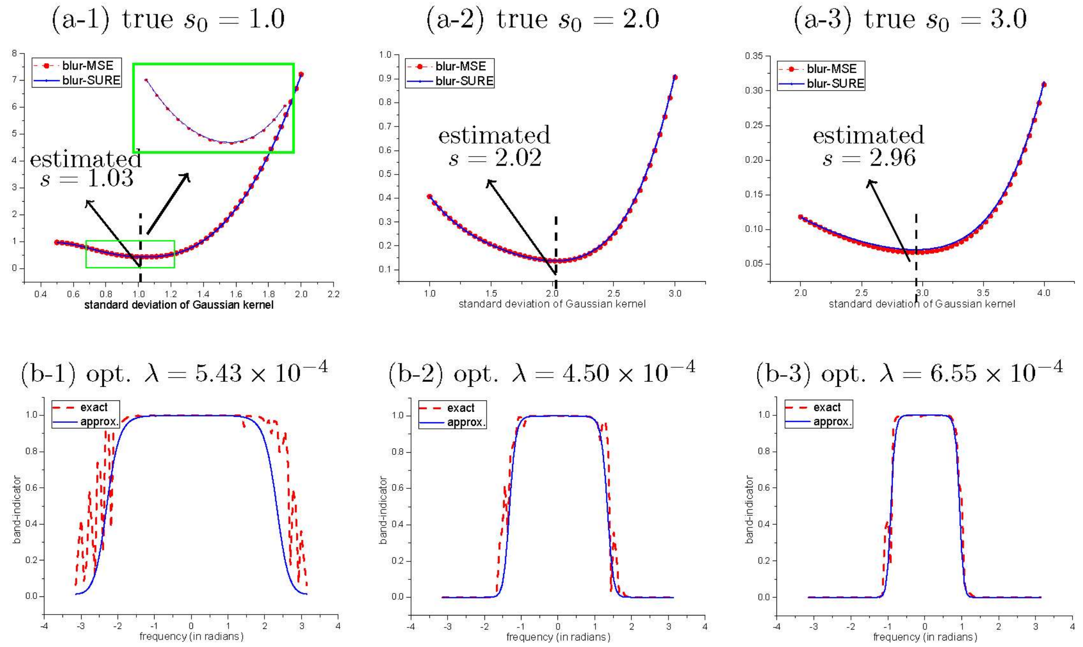


Figure 5.2: *Example of Cameraman blurred by true Gaussian kernel with standard deviation $s_0 = 1.0, 2.0, 3.0$ and noise variance $\sigma^2 = 1.0$. Fig.(a) show the relationship between s and blur-SURE (also the blur-MSE as the oracle counterpart of $\tilde{\epsilon}(s, \lambda)$). Fig.(b) show the corresponding approximations of the band-indicator Eq.(4.9), with the estimated s and λ, where the red dashed curve represents the exact $U_0(\omega)$ and the blue curve is the approximated $U_{\mathbf{H},\lambda}(\omega)$.*

5.3.2 Non-Gaussian PSF with scaling factor s

It is also possible to estimate the scaling factor of any PSF-type function within the blur-SURE framework, where the scaling factor controls the blur size. Here, we consider the following three typical non-Gaussian functions with scaling factor s, which are also circularly symmetric functions of radius $r = \sqrt{i^2 + j^2}$, where (i, j) is the 2-D coordinates:

- **exponential function** given as:

$$\mathbf{h}_s(r) = C \cdot \exp\left[-\left(\frac{r}{s}\right)^3 \right] \tag{5.3}$$

- ***jinc* function**[1], which is frequently used for describing the optical diffraction [19], given as:

$$\mathbf{h}_s(r) = C \cdot \left[\frac{2J_1(r/s)}{r/s}\right]^2 \tag{5.4}$$

  where $J_1(\cdot)$ is first-kind Bessel function of first-order [19].

- **rational function** given as:

$$\mathbf{h}_s(r) = C \cdot \left[1 + \left(\frac{r}{s}\right)^4\right]^{-1} \tag{5.5}$$

where C is the normalization coefficient, such that $\sum_{i,j} \mathbf{h}_s(i,j) = 1$. Figure 5.3 shows that for the non-Gaussian kernel case, the blur-SURE minimization also yields highly accurate estimate of the scaling factor $s \approx s_0$.

5.4 Minimization via the approximated function $\lambda = \lambda(s)$

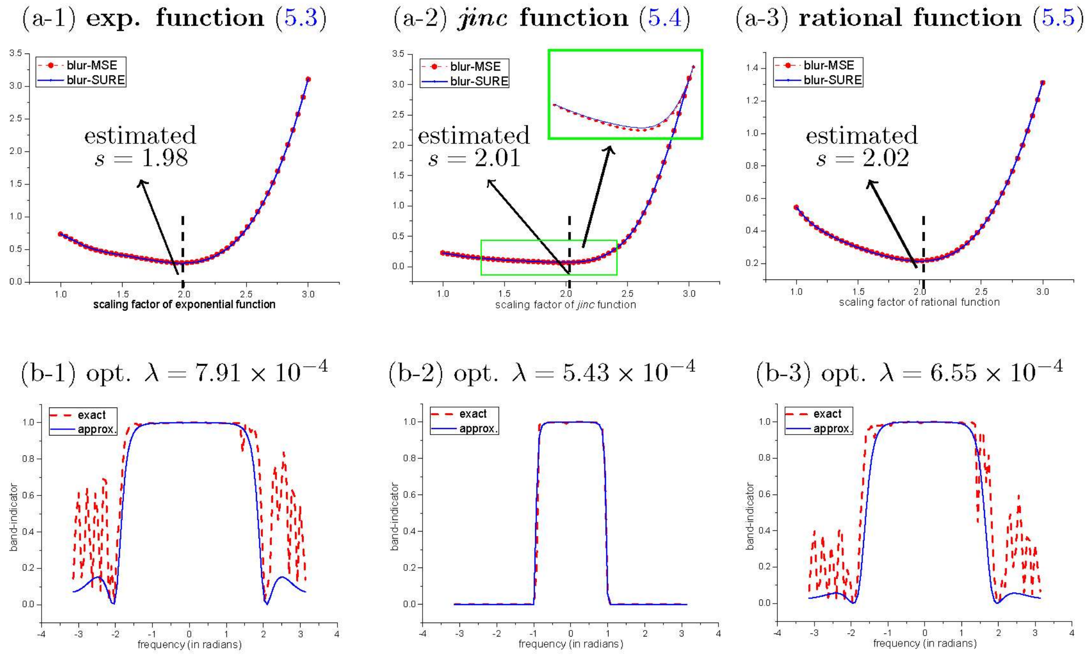
Considering the blur-SURE minimization over two scalar variables — PSF parameter s and regularization parameter λ, the most straitforward way to perform the minimization is to do exhaustive search over all the possible values of s and λ in a certain range. If we take 50 sampling points for s and λ to process the image of size 256×256, it costs 22.55sec for $50 \times 50 = 2500$ times of computing blur-SURE[2]. Another possibility is to use alternating minimization of the blur-SURE between s and λ. Sometimes, however, the convergence might be quite slow.

Here, we introduce a more efficient method: establishing the relationship between s and λ, and inserting the function $\lambda = \lambda(s)$ into the blur-SURE denoted by Eq.(5.1). Thus, the simplified minimization is performed over only one variable s without alternation.

Experimentally, we found that the function $\lambda = \lambda(s)$ can be well approximated

[1] The terminology *jinc* comes from optics, due to the structural similarity to sinc function, e.g. see [144].

[2] Hardware environment: Inter(R) Core(TM)2 Duo CPU E7400@2.80GHz, memory size: 3.25GB.



(a-1) **exp. function** (5.3) (a-2) ***jinc* function** (5.4) (a-3) **rational function** (5.5)

(b-1) opt. $\lambda = 7.91 \times 10^{-4}$ (b-2) opt. $\lambda = 5.43 \times 10^{-4}$ (b-3) opt. $\lambda = 6.55 \times 10^{-4}$

Figure 5.3: *Example of Cameraman blurred by three blur functions with true scaling factor $s_0 = 2.0$ and noise variance $\sigma^2 = 1.0$. Fig.(a) show the relationship between the scaling factor s and blur-SURE (also the blur-MSE as the oracle counterpart of $\tilde{\epsilon}(s, \lambda)$). Fig.(b) show the corresponding approximations of the band-indicator* (4.9).

using the following simple polynomial of order 2:

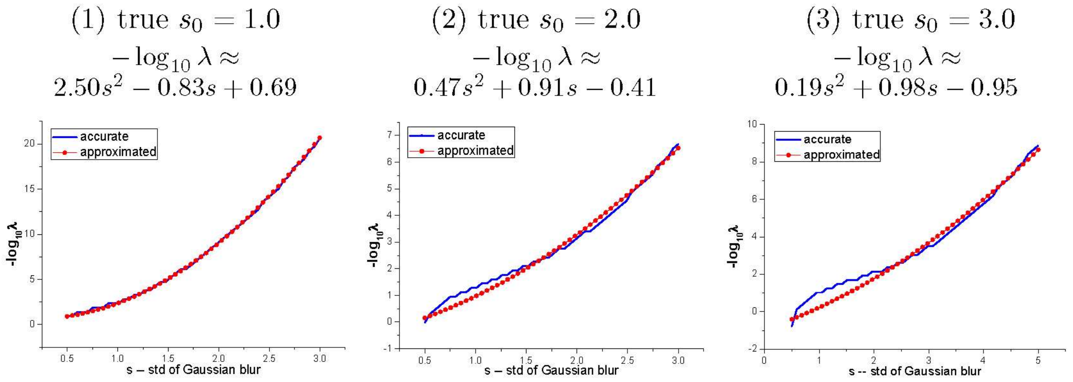
$$-\log_{10}\lambda \approx a_1 s^2 + a_2 s + a_3 \tag{5.6}$$

where the linear coefficients a_1, a_2 and a_3 can be obtained as follows. Given a sequence of tentative s_k for $k = 1, 2, ..., K$, there always exists an optimal λ_k corresponding to the minimum blur-SURE for each s_k, that are fitted by Eq.(5.6).

$$\underbrace{\begin{bmatrix} -\log_{10}\lambda_1 \\ -\log_{10}\lambda_2 \\ \vdots \\ -\log_{10}\lambda_K \end{bmatrix}}_{\Lambda} \approx a_1 \underbrace{\begin{bmatrix} s_1^2 \\ s_2^2 \\ \vdots \\ s_K^2 \end{bmatrix}}_{\mathbf{s}_1} + a_2 \underbrace{\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_K \end{bmatrix}}_{\mathbf{s}_2} + a_3 \underbrace{\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}}_{\mathbf{1}} = \underbrace{\begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{1} \end{bmatrix}}_{\mathbf{S}} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}}_{\mathbf{a}} \tag{5.7}$$

with the coefficients $\mathbf{a} \in \mathbb{R}^3$ to be determined. The optimal coefficients $\widehat{\mathbf{a}}$ can be found by minimizing the least square error:

$$\min_{\mathbf{a}} \|\Lambda - \mathbf{S}\mathbf{a}\|^2 \qquad \Longrightarrow \qquad \widehat{\mathbf{a}} = (\mathbf{S}^{\mathrm{T}}\mathbf{S})^{-1}\mathbf{S}^{\mathrm{T}}\Lambda \tag{5.8}$$


(1) true $s_0 = 1.0$ $-\log_{10}\lambda \approx 2.50s^2 - 0.83s + 0.69$

(2) true $s_0 = 2.0$ $-\log_{10}\lambda \approx 0.47s^2 + 0.91s - 0.41$

(3) true $s_0 = 3.0$ $-\log_{10}\lambda \approx 0.19s^2 + 0.98s - 0.95$

Figure 5.4: *The function $\lambda = \lambda(s)$ (blue curve) and its approximation (red curve): example of Cameraman blurred by Gaussian functions with $s_0 = 1.0,\ 2.0,\ 3.0$ and noise level $\sigma = 1.0$.*

Experimentally, we found that six sample-points (s_k, λ_k) are sufficient to obtain the optimal $\widehat{\mathbf{a}}$ that produces the good approximation of the function $\lambda = \lambda(s)$. Figure 5.4 shows the approximation of $\lambda = \lambda(s)$ using Eq.(5.7), by taking six sample values of $s_k = 0.5,\ 1.0,\ 1.5,\ 2.0,\ 2.5,\ 3.0$ and the corresponding λ_k, to obtain the polynomial coefficients $\mathbf{a}$ using Eq.(5.8).

The algorithm is summarized as Algorithm 5.1. Figure 5.5 shows line search for s of the blur-SURE $\tilde{\epsilon}(s, \lambda(s))$, where the approximated $\lambda = \lambda(s)$ is given in Figure 5.4.

Algorithm 5.1: Algorithm by approximated function $\lambda = \lambda(s)$

```
Input: $\tilde{\epsilon}(s, \lambda)$: objective function given as Eq.(5.1);
Output: optimal $\lambda$ and $s$
begin
    1. take six sample-values $s_k = 0.5,\ 1.0,\ 1.5,\ 2.0,\ 2.5,\ 3.0$ for $k = 1, 2, ..., 6$;
    2. for each $s_k$, find the optimal $\lambda_k$ corresponding to the minimum $\epsilon$: $\min_{\lambda_k} \tilde{\epsilon}(s_k, \lambda_k)$;
    3. use polynomial (5.7) to approximate $\lambda = \lambda(s)$, with the coefficients $\mathbf{a}$ obtained by (5.8);
    4. incorporate Eq.(5.6) into $\tilde{\epsilon}(s, \lambda)$, thus, $\tilde{\epsilon}(s, \lambda(s))$ becomes a function of single variable $s$;
    5. minimize $\tilde{\epsilon}(s, \lambda(s))$ over $s$.
end
```

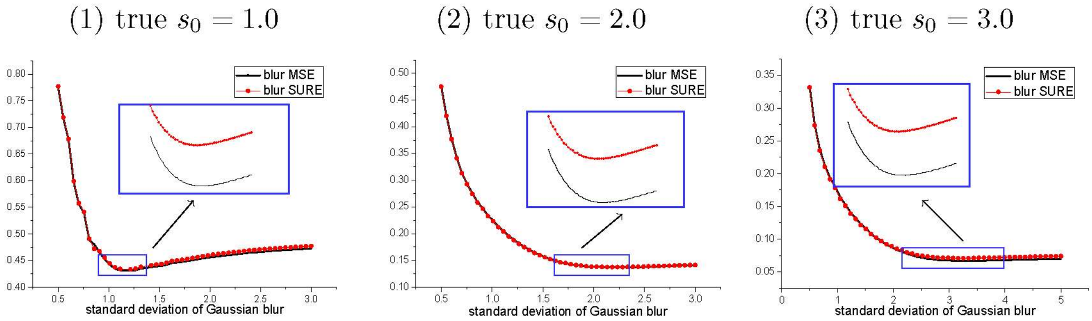

(1) true $s_0 = 1.0$ (2) true $s_0 = 2.0$ (3) true $s_0 = 3.0$

Figure 5.5: *Minimization of* $\tilde{\epsilon}(s, \lambda(s))$ *over* s*, where the function* $\lambda = \lambda(s)$ *is approximated using Figure 5.4: example of Cameraman blurred by Gaussian functions with* $s_0 = 1.0, 2.0, 3.0$ *and noise level* $\sigma^2 = 1.0$.

5.5 Results and discussions

5.5.1 Experimental setting

We consider the following convolution kernels described above, with scaling factor s:

- **Gaussian function** by Eq.(5.2);
- Non-Gaussian functions: **exponential function** by Eq.(5.3), ***jinc* function** by Eq.(5.4), and **rational function** by Eq.(5.5).

The blurred images are subsequently contaminated by i.i.d Gaussian noise with various variance σ^2, which is measured by blur signal-to-noise ratio (BSNR), defined as $\mathrm{BSNR} = 10 \log_{10} \left(\frac{\|\mathbf{H}_0\mathbf{x} - \mathrm{mean}(\mathbf{H}_0\mathbf{x})\|^2}{N\sigma^2} \right)$. The test dataset contains six 8-bit images of size 256×256 or 512×512 displayed in Figure 5.6[3], covering a wide range of natural images (people, animal, building, remote sensing, etc.). As emphasized above, our blind deconvolution approach consists of two steps: we firstly estimate the PSF parameter from the observed image by the blur-SURE minimization, and then use our recently proposed non-blind deconvolution algorithm [6] to perform deconvolution, with the estimated PSF. For the first step, we present our estimated PSF parameter, compared to the latent true parameter s_0; for the second step, the deconvolution performance is measured by PSNR defined as Eq.(1.11). Note that all the results reported in this section have been averaged over 10 noise realizations.

[3] All 512×512 images are available at http://decsai.ugr.es/cvg/CG/base.htm

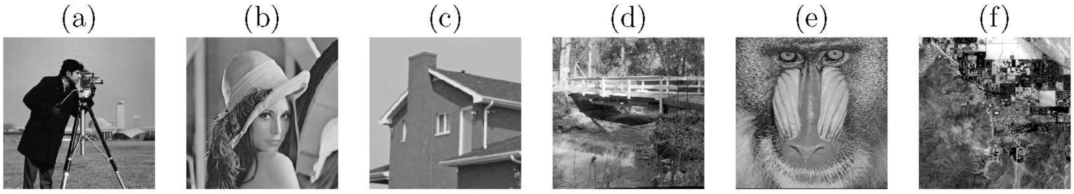


Figure 5.6: *Original images. (a) Cameraman* 256 × 256; *(b) Lena* 256 × 256; *(c) House* 256 × 256; *(d) Bridge* 512 × 512; *(e) Mandrill* 512 × 512; *(f) California* 512 × 512.

5.5.2 Non-Gaussian functions: estimation of scaling factor s

Table 5.1 shows the estimation of scaling factor of the non-Gaussian blur functions by minimizing blur-SURE, where the italic numbers stand for the *oracle* results of minimizing blur-MSE. We can see that: (1) the estimated s by the blur-SURE minimization is very close to the latent true s_0 for a wide range of images and various noise levels; (2) the blur-SURE minimization yields very close estimated s to that by minimizing the blur-MSE, which demonstrates that blur-SURE is an accurate estimate of blur-MSE.

true s_0		$s_0 = 1.0$				$s_0 = 2.0$				$s_0 = 3.0$			
BSNR (in dB)		40	30	20	10	40	30	20	10	40	30	20	10
blur type		**exponential function** (5.3)											
Cameraman	blur-SURE	1.01	1.02	1.03	1.05	1.97	2.00	2.03	2.04	2.99	2.95	2.96	3.03
	blur-MSE	*1.00*	*1.02*	*1.03*	*1.06*	*1.97*	*2.00*	*2.03*	*2.07*	*3.00*	*3.00*	*3.05*	*3.10*
Mandrill	blur-SURE	0.99	1.00	1.02	0.98	2.00	1.99	1.98	1.95	3.00	3.00	2.97	2.94
	blur-MSE	*0.99*	*1.00*	*1.02*	*0.98*	*2.00*	*1.99*	*1.98*	*1.96*	*3.00*	*3.00*	*2.97*	*2.91*
California	blur-SURE	1.00	1.02	1.05	1.08	1.99	2.01	2.07	2.07	3.00	3.00	3.02	3.03
	blur-MSE	*0.99*	*1.02*	*1.05*	*1.13*	*1.99*	*2.00*	*2.06*	*2.13*	*3.00*	*2.98*	*3.02*	*3.00*
blur type		***jinc* function** (5.4)											
Cameraman	blur-SURE	1.00	1.01	1.04	1.02	1.98	2.02	2.05	2.00	3.01	3.04	3.06	3.10
	blur-MSE	*1.00*	*1.00*	*1.04*	*1.10*	*2.00*	*2.00*	*2.05*	*2.06*	*2.98*	*3.02*	*3.04*	*3.12*
Mandrill	blur-SURE	1.00	1.00	0.99	0.94	2.00	2.00	1.96	1.96	3.00	3.01	2.99	3.01
	blur-MSE	*1.00*	*1.00*	*0.99*	*0.92*	*1.99*	*2.00*	*1.98*	*1.95*	*3.00*	*3.01*	*3.01*	*2.98*
California	blur-SURE	1.00	1.01	1.03	1.03	2.00	2.00	2.00	2.00	3.00	3.00	2.98	3.04
	blur-MSE	*1.00*	*1.01*	*1.03*	*1.07*	*2.00*	*2.01*	*2.00*	*2.00*	*3.00*	*3.00*	*3.01*	*3.02*
blur type		**rational function** (5.5)											
Cameraman	blur-SURE	1.00	1.03	1.06	1.11	1.98	2.06	2.04	2.02	2.98	2.98	2.95	3.12
	blur-MSE	*1.00*	*1.03*	*1.06*	*1.13*	*1.98*	*2.03*	*2.07*	*2.05*	*2.98*	*3.04*	*3.01*	*3.15*
Mandrill	blur-SURE	0.99	1.00	0.99	0.94	1.99	1.99	1.94	1.92	3.01	3.00	2.93	2.91
	blur-MSE	*0.99*	*1.00*	*0.99*	*0.94*	*1.99*	*1.98*	*1.90*	*1.91*	*3.01*	*3.00*	*2.93*	*2.91*
California	blur-SURE	1.00	1.04	1.07	1.11	1.97	2.06	2.10	2.12	2.99	2.98	3.02	3.02
	blur-MSE	*1.00*	*1.04*	*1.10*	*1.15*	*1.97*	*2.02*	*2.10*	*2.10*	*2.99*	*2.98*	*3.00*	*3.00*

Table 5.1: *Estimation of scaling factor s of non-Gaussian functions under different BSNR*

5.5.3 Gaussian function: estimation of standard deviation s

Tables 5.2, 5.3 and 5.4 report the estimated s of Gaussian blur under three true $s_0 = 1.0,\ 2.0,\ 3.0$. We also compare our estimation results with other state-of-the-art methods, including APEX [3], Wiener iterative method [10] and Richardson-Lucy method [11]. Note that [138; 140; 141] try to estimate the local blur variance by detecting and measuring the edges, and obtain the focus map, where each pixel is assigned with quite different estimated blur variance. It is difficult to obtain a number to represent the blurriness of the whole image from the focus map. Hence, we do not include their results here. From Tables 5.2, 5.3 and 5.4, we can see that: (1) our approach generally yields more accurate estimation of blur size s than other methods, in terms of error defined as $e = |s - s_0|$; (2) minimizations of blur-SURE and blur-MSE produce very similar estimation results.

Note that PSF estimation is used for non-blind deconvolution step. It is also important to study the influence of the PSF accuracy upon the deconvolution performance.

BSNR (in dB)	40	30	20	10	40	30	20	10	40	30	20	10
Image	*Cameraman* 256×256				*Lena* 256×256				*House* 256×256			
APEX [3]	0.91	0.91	0.91	0.91	1.23	1.21	1.17	1.16	1.28	1.28	1.28	1.23
Wiener iterative [10]	1.08	1.09	0.75	0.68	1.17	1.10	0.85	0.66	1.39	1.27	0.93	0.66
Richardson-Lucy [11]	0.87	0.66	—	—	0.93	0.73	—	—	0.95	0.77	—	—
blur-SURE	**1.00**	**1.06**	**1.07**	**1.05**	**1.03**	**1.02**	**1.06**	**1.10**	**0.98**	**0.99**	**1.02**	**1.17**
blur-MSE	*1.00*	*1.06*	*1.04*	*1.13*	*1.04*	*1.01*	*1.07*	*1.12*	*0.94*	*0.99*	*1.03*	*1.26*
Image	*Bridge* 512×512				*Mandrill* 512×512				*California* 512×512			
APEX [3]	1.14	1.09	**0.98**	0.77	1.14	1.14	1.03	0.85	1.57	**0.98**	0.91	0.77
Wiener iterative [10]	0.89	0.81	0.64	0.50	0.85	0.81	0.69	0.52	0.85	0.79	0.62	0.51
Richardson-Lucy [11]	0.89	0.78	0.68	—	0.85	0.78	—	—	0.78	0.77	0.63	—
blur-SURE	**1.01**	**1.02**	1.05	**1.10**	**1.00**	**1.01**	**1.00**	**0.95**	**1.00**	1.07	**1.07**	**1.11**
blur-MSE	*1.01*	*1.03*	*1.05*	*1.10*	*1.00*	*1.01*	*1.00*	*0.94*	*1.00*	*1.08*	*1.07*	*1.11*

Table 5.2: *Estimation of std s of Gaussian function under true $s_0 = 1.0$ and BSNR*

5.5.4 Comparison of deconvolution performance with the state-of-the-art

We use the SURE-LET algorithm [6] to perform deconvolution with the blur-SURE estimated PSF. We compare our results with other state-of-the-art methods. APEX [3] use SECB method [2; 3] to perform deconvolution. SeDDaRA has two versions: with constant parameter [7] and with frequency-dependent parameter [4]. For the first version, we use an approach proposed in [8] to optimize the result, denoted by [7]+ [8]. For the second version, we use the unknown original image $\mathbf{x}$ to obtain the involved

BSNR (in dB)	40	30	20	10	40	30	20	10	40	30	20	10
Image	*Cameraman* 256 × 256				*Lena* 256 × 256				*House* 256 × 256			
APEX [3]	1.94	1.97	1.91	1.65	2.12	2.08	2.02	1.76	2.15	2.15	**2.05**	1.76
Wiener iterative [10]	**2.02**	1.97	1.59	0.90	2.16	2.17	1.86	0.83	2.29	2.29	1.92	0.87
Richardson-Lucy [11]	1.87	1.50	0.83	—	1.94	1.81	0.70	—	1.91	1.85	0.93	—
blur-SURE	2.08	**1.99**	**1.97**	**2.09**	**2.07**	**2.05**	**2.04**	**2.20**	**2.06**	**2.02**	2.12	**2.18**
blur-MSE	*2.04*	*2.00*	*2.09*	*2.15*	*2.11*	*2.13*	*2.11*	*2.29*	*2.13*	*2.10*	*2.18*	*2.27*
Image	*Bridge* 512 × 512				*Mandrill* 512 × 512				*California* 512 × 512			
APEX [3]	1.84	1.80	1.72	1.61	1.98	**1.98**	1.84	1.51	2.60	2.52	2.35	2.12
Wiener iterative [10]	1.90	1.78	1.06	0.92	1.90	1.90	1.35	0.94	1.90	1.66	1.02	0.84
Richardson-Lucy [11]	1.81	1.79	0.62	—	**2.00**	1.92	1.16	—	1.86	1.65	1.44	—
blur-SURE	**2.02**	**2.02**	**2.06**	**2.09**	1.97	1.97	**1.93**	**1.90**	**2.05**	**2.06**	**2.06**	**2.01**
blur-MSE	*2.02*	*2.02*	*2.06*	*2.07*	*1.97*	*1.96*	*1.93*	*1.91*	*2.05*	*2.05*	*2.05*	*2.00*

Table 5.3: *Estimation of std s of Gaussian function under true $s_0 = 2.0$ and BSNR*

BSNR (in dB)	40	30	20	10	40	30	20	10	40	30	20	10
Image	*Cameraman* 256 × 256				*Lena* 256 × 256				*House* 256 × 256			
APEX [3]	2.82	2.64	3.27	**3.16**	2.91	2.72	2.37	2.26	2.91	2.72	2.40	2.26
Wiener iterative [10]	3.21	3.21	3.08	1.77	3.21	3.31	3.33	2.44	3.21	3.40	3.35	2.50
Richardson-Lucy [11]	**2.97**	2.84	—	—	**2.99**	2.84	1.78	—	**2.97**	**2.84**	1.96	—
blur-SURE	2.95	**3.09**	**3.04**	3.23	2.93	**3.15**	**3.28**	**3.25**	**3.03**	3.22	**3.16**	**3.09**
blur-MSE	*3.02*	*3.11*	*3.20*	*3.29*	*3.07*	*3.12*	*3.31*	*3.29*	*3.17*	*3.12*	*3.15*	*3.14*
Image	*Bridge* 512 × 512				*Mandrill* 512 × 512				*California* 512 × 512			
APEX [3]	2.79	2.61	3.14	**3.07**	2.86	2.69	2.45	3.10	3.36	3.30	3.17	**3.05**
Wiener iterative [10]	3.19	3.19	2.69	1.83	3.19	3.19	**3.04**	1.68	3.19	3.19	2.58	1.75
Richardson-Lucy [11]	2.94	2.56	1.48	—	**2.99**	2.89	2.21	—	2.94	2.87	2.14	—
blur-SURE	**3.04**	**3.01**	**3.07**	3.14	2.92	**2.93**	2.94	**2.96**	**3.04**	**3.05**	**3.01**	2.93
blur-MSE	*3.04*	*3.04*	*3.12*	*3.22*	*2.92*	*2.92*	*2.93*	*2.95*	*3.03*	*3.04*	*2.99*	*3.01*

Table 5.4: *Estimation of std s of Gaussian function under true $s_0 = 3.0$ and BSNR*

frequency-dependent parameter to optimize [4] in synthetic experiments, denoted by *oracle* SeDDaRA [4]. Molina *et al.* work [5] has BR and BD modes, for which we use the parameters as recommended in this paper. Besides, we also compare our results with two multiplicative iterative algorithms [9; 12]. The deconvolution performance is measured by PSNR. We experimentally found that the PSNR performances of SeDDaRA [4; 7] and multiplicative iterative algorithms [9; 12] are poor, but the visual quality is reasonably good. For this reason, we linearly rescale the restored image $\widehat{\mathbf{x}}$ to $a\widehat{\mathbf{x}} + b$, where a and b optimize the PSNR, i.e. $\min_{a,b} \|a\widehat{\mathbf{x}} + b - \mathbf{x}\|^2$.

From Tables 5.5 and 5.6, we can observe that our proposed method outperforms the other methods on average. By comparing our proposed method with non-blind SURE-LET algorithm [6], we would also like to note that in most cases, our blind Gaussian deconvolution could achieve the PSNR loss within 0.1dB, compared to the results of non-blind algorithm [6] (reported in italic number in Tables 5.5 and 5.6, where the PSF is exactly known. Figures 5.7 and 5.8 show two visual comparisons between the existing methods. We can see that: (1) our resulting restorations are shaper and most of the blur is removed; (2) the proposed method achieves similar visual quality with non-blind algorithm [6], shown in Figure 5.7.

BSNR (in dB)		40	30	20	10	40	30	20	10
Image		*Cameraman* 256 × 256				*Lena* 256 × 256			
APEX with SECB [2; 3]		30.13	28.06	25.68	22.77	29.44	30.08	28.97	26.25
Wiener iterative [10]		28.29	26.99	24.11	22.13	28.51	29.38	26.58	23.61
R-L iterative [11]		28.90	25.04	21.14	12.24	30.62	25.75	23.75	17.49
SeDDaRA	[7]+ [8]	26.84	26.61	25.30	21.15	31.33	30.91	28.91	23.92
	oracle [4]	30.99	28.06	25.63	23.17	36.03	32.78	29.78	26.86
[5]	BR mode	30.17	27.68	25.84	23.10	35.33	32.77	30.15	26.86
	BD mode	30.42	27.79	25.65	17.87	35.07	32.53	29.84	22.65
sparse prior [92]+ [94]		25.64	15.72	12.44	12.27	29.77	16.58	14.99	14.65
MIA [12]		28.33	27.74	25.38	21.06	32.65	31.56	28.96	23.82
AMIA [9]		29.55	28.85	25.44	21.06	34.70	33.41	29.01	23.82
Xu [145]		26.70	26.52	22.55	—	27.67	27.73	25.23	—
blur-SURE	with SURE-LET [6]	**33.10**	**30.40**	**27.87**	**25.35**	**37.70**	**34.97**	**31.90**	**28.66**
	non-blind [6]	*33.10*	*30.49*	*27.91*	*25.35*	*37.79*	*34.97*	*31.95*	*28.66*
Image		*House* 256 × 256				*Bridge* 512 × 512			
APEX with SECB [2; 3]		28.91	30.24	29.49	26.89	28.63	28.34	26.15	22.36
Wiener iterative [10]		24.63	28.95	27.66	23.91	27.79	25.75	23.84	21.25
R-L iterative [11]		35.87	25.50	24.13	17.65	18.48	13.60	13.47	14.48
SeDDaRA	[7]+ [8]	32.39	32.00	29.92	24.40	26.86	26.68	25.61	21.97
	oracle [4]	38.20	34.50	31.36	28.51	30.64	27.80	25.50	23.31
[5]	BR mode	32.86	29.64	28.20	27.98	29.70	27.91	25.94	23.15
	BD mode	33.96	31.45	30.76	21.39	29.16	27.71	25.74	18.33
sparse prior [92]+ [94]		27.90	19.24	15.94	14.88	27.78	16.19	13.49	13.40
MIA [12]		33.93	32.76	29.97	24.29	27.73	27.43	25.75	21.95
AMIA [9]		35.91	34.36	29.99	24.29	28.20	28.00	26.17	21.95
Xu [145]		27.46	27.43	25.06	—	26.76	26.55	24.04	—
blur-SURE	with SURE-LET [6]	**38.57**	**36.17**	**33.33**	**30.62**	**31.68**	**29.35**	**27.20**	**24.83**
	non-blind [6]	*38.70*	*36.21*	*33.33*	*30.74*	*31.70*	*29.38*	*27.22*	*24.84*
Image		*Mandrill* 512 × 512				*California* 512 × 512			
APEX with SECB [2; 3]		26.79	26.17	25.08	22.42	18.96	28.10	25.34	21.81
Wiener iterative [10]		26.67	24.90	23.04	21.18	27.11	25.13	23.13	20.57
R-L iterative [11]		15.62	23.70	22.47	18.00	13.01	12.96	13.10	12.95
SeDDaRA	[7]+ [8]	24.66	24.54	23.91	21.97	25.71	25.52	24.44	21.13
	oracle [4]	30.28	26.87	24.17	21.94	30.39	27.26	24.71	22.23
[5]	BR mode	27.32	25.77	24.10	21.92	29.05	27.05	24.82	21.57
	BD mode	26.22	24.26	23.98	19.50	28.33	26.73	24.61	16.76
sparse prior [92]+ [94]		20.15	21.81	16.85	15.62	21.43	19.56	13.63	12.97
MIA [12]		25.14	25.05	24.18	21.97	26.55	26.29	24.66	21.12
AMIA [9]		25.39	25.34	24.40	21.97	27.11	26.94	25.18	21.12
Xu [145]		24.52	24.43	23.97	21.95	26.07	25.88	23.49	—
blur-SURE	with SURE-LET [6]	**30.68**	**27.86**	**25.49**	**23.20**	**31.43**	**28.63**	**26.16**	**23.60**
	non-blind [6]	*30.68*	*27.87*	*25.49*	*23.22*	*31.43*	*28.73*	*26.25*	*23.61*

Table 5.5: *PSNR (in dB) of Gaussian deconvolution under true* $s_0 = 1.0$ *and BSNR*

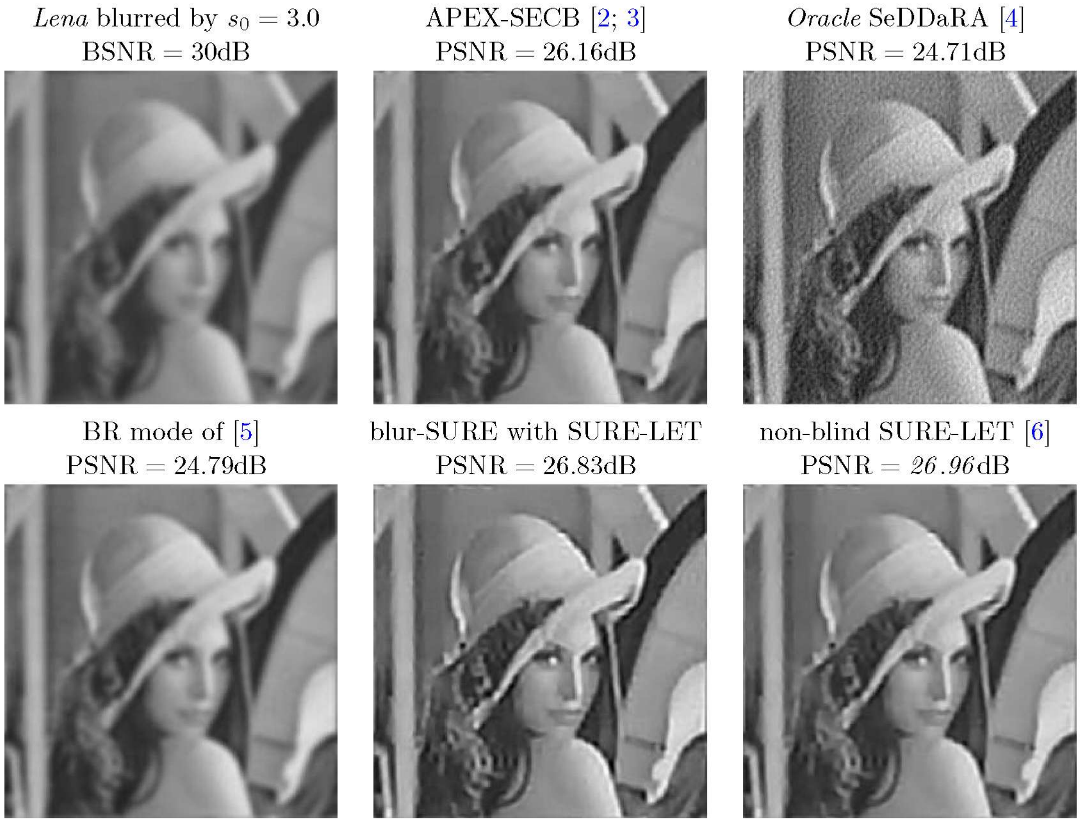
BSNR (in dB)		40	30	20	10	40	30	20	10
Image		*Cameraman* 256 × 256				*Lena* 256 × 256			
APEX with SECB [2; 3]		25.60	24.72	23.78	22.41	29.54	28.65	27.35	25.56
Wiener iterative [10]		23.21	22.61	20.77	16.73	24.94	24.74	23.45	17.94
R-L iterative [11]		24.96	21.92	12.26	12.30	29.76	27.52	19.89	17.18
SeDDaRA	[7]+ [8]	23.15	23.00	22.26	19.79	26.69	26.37	25.25	22.43
	oracle [4]	24.47	23.00	21.86	20.72	28.59	26.97	25.54	24.01
[5]	BR mode	22.54	23.74	22.96	21.71	25.52	27.15	26.29	24.77
	BD mode	23.86	23.39	22.38	17.85	27.04	26.76	25.77	21.87
sparse prior [92]+ [94]		23.90	16.60	12.61	12.30	16.72	19.53	15.19	14.68
MIA [12]		22.79	22.68	22.15	19.72	25.86	25.73	25.08	22.36
AMIA [9]		23.30	23.08	22.15	19.72	26.53	26.26	25.09	22.36
Xu [145]		23.71	23.78	19.82	—	25.44	24.52	23.23	—
blur-SURE	with SURE-LET [6]	**26.19**	**25.44**	**24.44**	**23.20**	**30.40**	**29.41**	**28.01**	**26.33**
	non-blind [6]	*26.28*	*25.45*	*24.47*	*23.22*	*30.59*	*29.47*	*28.01*	*26.40*
Image		*House* 256 × 256				*Bridge* 512 × 512			
APEX with SECB [2; 3]		30.70	29.97	28.84	26.72	25.59	24.84	23.95	22.61
Wiener iterative [10]		24.87	25.02	24.02	18.23	22.75	21.95	18.58	17.11
R-L iterative [11]		30.72	28.11	19.61	17.42	25.18	24.84	19.77	15.90
SeDDaRA	[7]+ [8]	27.84	27.57	26.37	23.10	23.39	23.27	22.67	20.53
	oracle [4]	30.23	28.63	27.23	25.66	24.52	23.17	22.10	21.05
[5]	BR mode	26.78	29.30	27.99	25.84	22.86	23.67	23.02	21.74
	BD mode	26.78	28.60	27.04	21.23	22.86	23.37	22.64	18.08
sparse prior [92]+ [94]		20.15	19.24	15.24	14.89	23.91	16.60	13.56	13.42
MIA [12]		27.16	27.01	26.21	23.01	23.10	23.02	22.57	20.51
AMIA [9]		27.89	27.54	26.22	23.01	23.47	23.41	22.65	20.51
Xu [145]		25.29	25.01	22.48	—	23.78	23.63	21.08	—
blur-SURE	with SURE-LET [6]	**31.85**	**31.06**	29.75	**28.10**	**26.09**	**25.29**	**24.31**	**23.12**
	non-blind [6]	*31.97*	*31.08*	*29.89*	*28.21*	*26.10*	*25.29*	*24.31*	*23.12*
Image		*Mandrill* 512 × 512				*California* 512 × 512			
APEX with SECB [2; 3]		23.27	22.64	22.00	21.17	20.37	21.65	21.84	21.03
Wiener iterative [10]		21.57	21.26	19.19	17.43	22.12	20.53	17.61	16.43
R-L iterative [11]		23.39	22.71	19.17	17.40	23.59	22.94	13.03	13.10
SeDDaRA	[7]+ [8]	21.56	21.48	21.17	20.22	21.72	21.58	20.98	19.25
	oracle [4]	22.65	21.26	20.21	19.42	23.38	21.85	20.63	19.46
[5]	BR mode	21.14	21.59	21.23	20.56	20.99	21.92	21.26	19.95
	BD mode	21.14	21.46	21.07	19.26	20.99	21.60	20.89	17.22
sparse prior [92]+ [94]		21.78	19.42	16.02	15.63	22.63	16.98	13.51	12.99
MIA [12]		21.29	21.26	21.07	20.22	21.24	21.18	20.81	19.24
AMIA [9]		21.52	21.50	21.15	20.22	21.65	21.60	20.94	19.24
Xu [145]		21.80	21.77	21.29	16.70	22.42	22.33	20.14	—
blur-SURE	with SURE-LET [6]	**23.56**	**22.87**	**22.12**	**21.30**	**24.72**	**23.83**	**22.76**	**21.42**
	non-blind [6]	*23.59*	*22.88*	*22.14*	*21.31*	*24.81*	*23.88*	*22.76*	*21.42*

Table 5.6: *PSNR (in dB) of Gaussian deconvolution under true* $s_0 = 2.0$ *and BSNR*

5.5.5 Application to real images

In our last set of experiments, the method is applied to real images:

- **Astronomical image:** *Jupiter*, shown in Figure 5.9;
- **Text image:** *Text*, shown in Figure 5.10.



Lena blurred by $s_0 = 3.0$
BSNR = 30dB

APEX-SECB [2; 3]
PSNR = 26.16dB

Oracle SeDDaRA [4]
PSNR = 24.71dB

BR mode of [5]
PSNR = 24.79dB

blur-SURE with SURE-LET
PSNR = 26.83dB

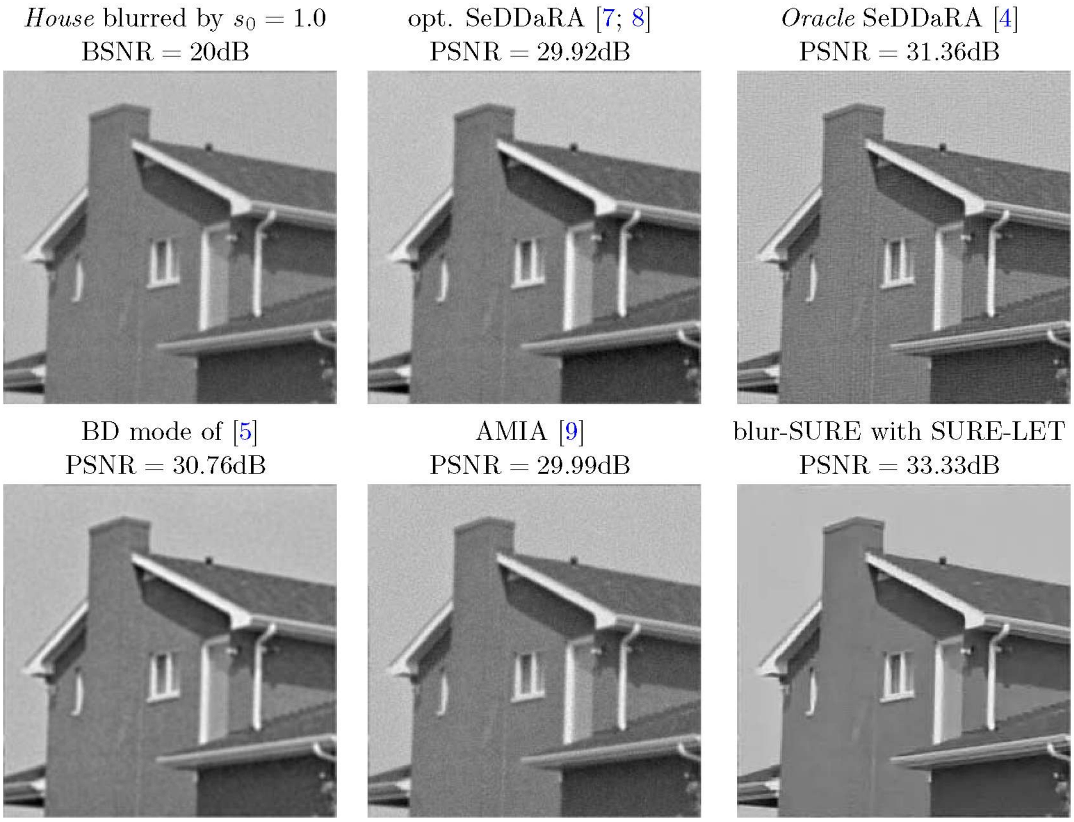
non-blind SURE-LET [6]
PSNR = *26.96*dB

Figure 5.7: *Restoration of Lena: visual comparison between APEX-SECB [2; 3], oracle SeDDaRA [4], Molina et al. work [5], our proposed blur-SURE method, and non-blind SURE-LET method [6]. (Note that the blurred* PSNR = 23.20*dB.)*

- **Biosample image:** *Biosample*, shown in Figure 5.11.

There are no exact expressions of the PSFs for the images. However, as suggested in [5; 17; 88; 101], the PSF of astronomical imaging can be well approximated by Gaussian function. Figure 5.9 shows the restored images by various state-of-the-art methods. Note that for Wiener iterative method [10] and MIA [12], we optimized the initial guess according to the visual quality. We can easily see that our proposed approach have restored a great amount of clarity, especially, revealed the clear stripes on *Jupiter*.

The *Text* images with different blurriness were captured by digital camera, due to out-of-focus. The out-of-focus blur can also be approximated by Gaussian function. Figure 5.10 shows the visual comparison of the restored images by various deblurring algorithms. We can see our results are better than the works of [5] and [145]: especially for the second and third cases, it is difficult to recognize most characters from the



House blurred by $s_0 = 1.0$
BSNR = 20dB

opt. SeDDaRA [7; 8]
PSNR = 29.92dB

Oracle SeDDaRA [4]
PSNR = 31.36dB

BD mode of [5]
PSNR = 30.76dB

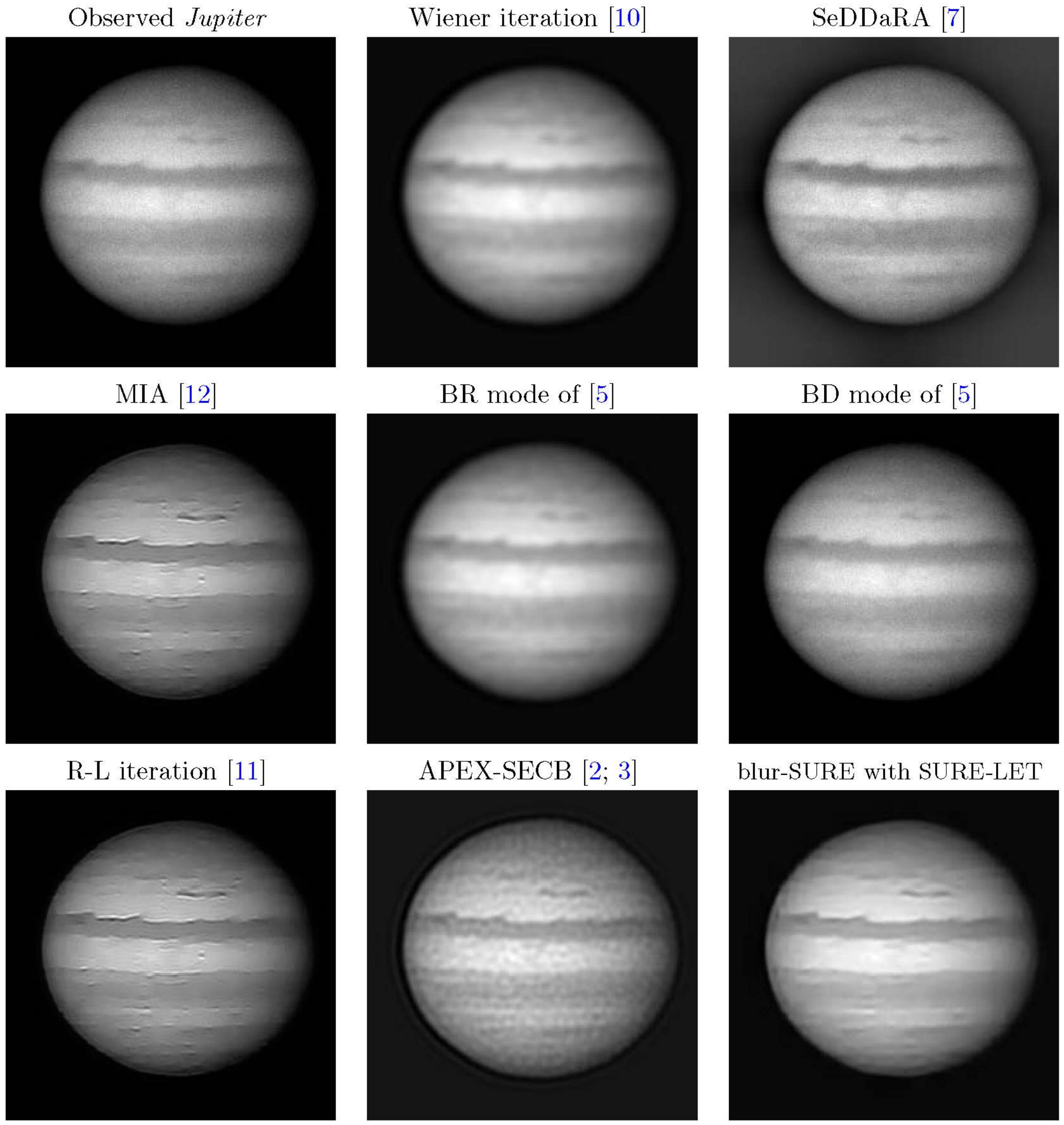
AMIA [9]
PSNR = 29.99dB

blur-SURE with SURE-LET
PSNR = 33.33dB

Figure 5.8: *Restoration of House: visual comparison between SeDDaRA [7; 8], oracle SeDDaRA [4], Molina et al. work [5], AMIA [9] and our proposed blur-SURE method. (Note that the blurred* PSNR = 29.81*dB.)*

observed images, whereas most characters become recognizable in our restored images.

The *Biosample* images were recorded by confocal microscopy. As suggested in [105; 136], the PSF of microscopy imaging can be well approximated by Gaussian function. Figure 5.11 shows the restored images. For the biologists, the first and third restorations have the big visual improvements.



Observed *Jupiter*

Wiener iteration [10]

SeDDaRA [7]

MIA [12]

BR mode of [5]

BD mode of [5]

R-L iteration [11]

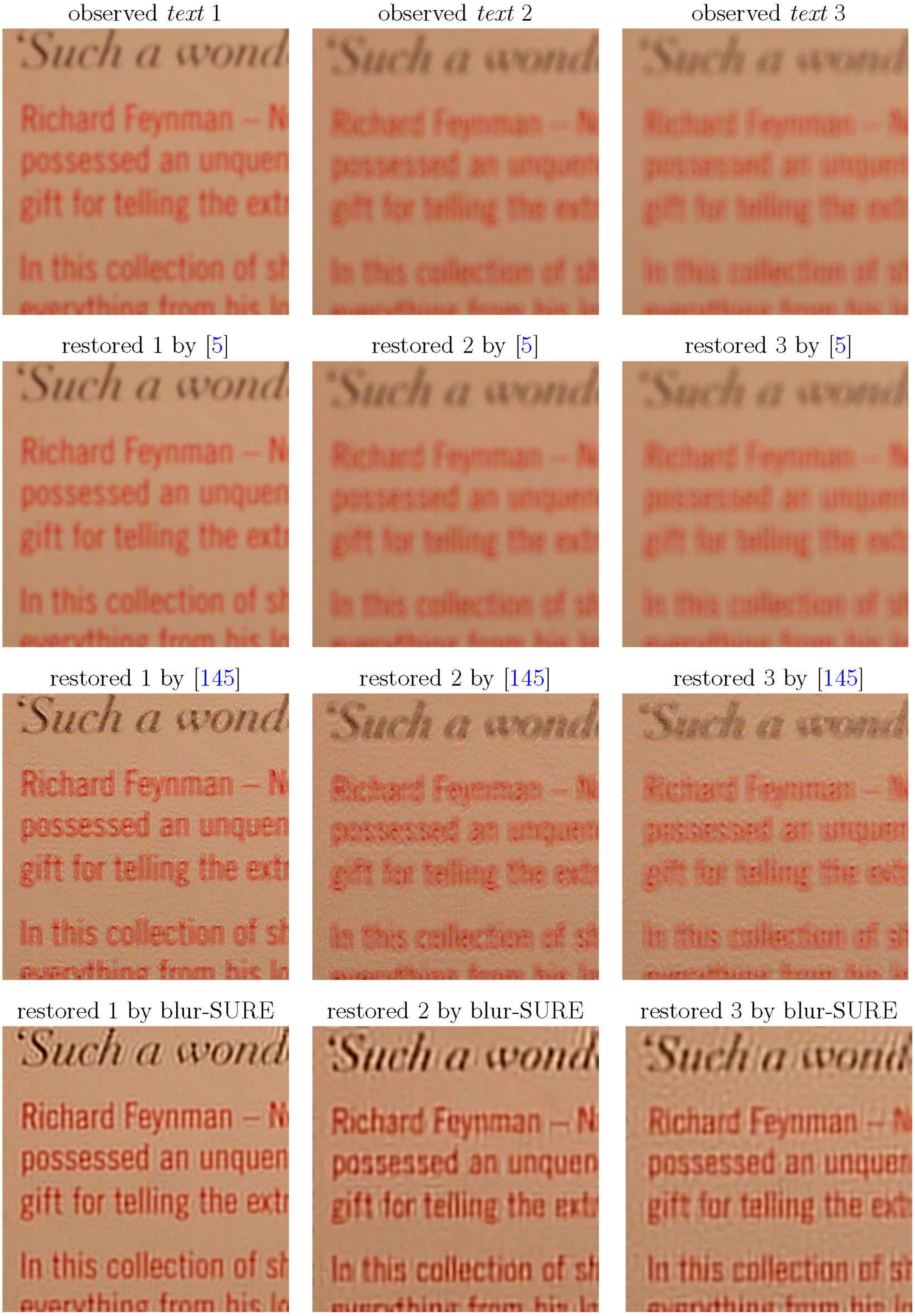
APEX-SECB [2; 3]

blur-SURE with SURE-LET

Figure 5.9: *Restoration of Jupiter: visual comparison between Wiener iterative method [10], R-L iterative method [11], SeDDaRA [7], Molina et al. work [5], MIA [12], APEX-SECB [2; 3] and our proposed blur-SURE method. The estimated noise standard deviation is $\sigma = 4.68$ by using MAD (median absolute deviation) [13], the estimated standard deviation of Gaussian blur is $s = 2.30$ and $s = 2.41$, by APEX and blur-SURE, respectively.*

5.6 Conclusion

In this chapter, we exemplified the blur-SURE framework with estimating scaling factor involved in the function forms of PSF, most typically, the variance of Gaussian blur.



observed *text* 1 observed *text* 2 observed *text* 3

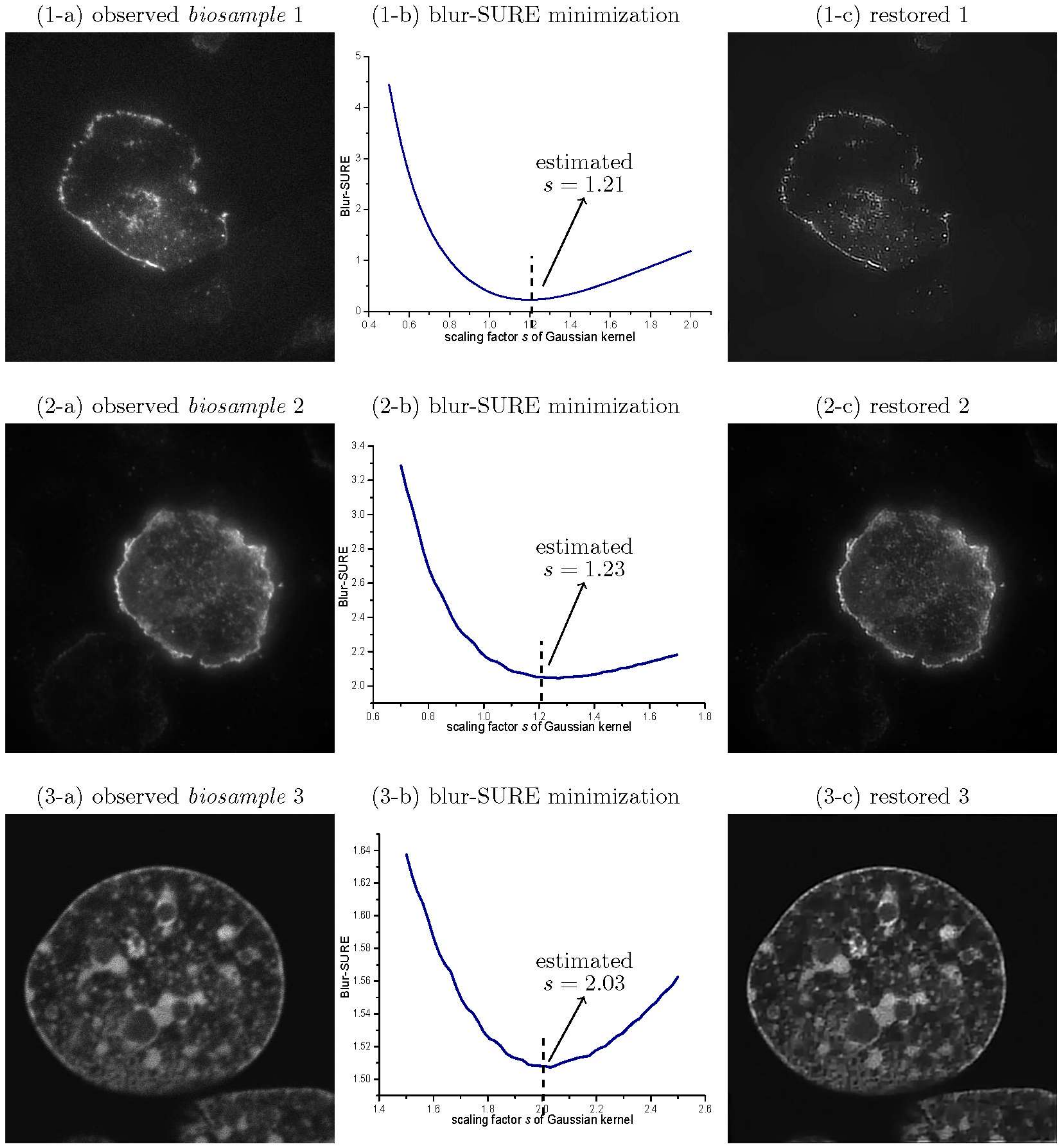
restored 1 by [5] restored 2 by [5] restored 3 by [5]

restored 1 by [145] restored 2 by [145] restored 3 by [145]

restored 1 by blur-SURE restored 2 by blur-SURE restored 3 by blur-SURE

Figure 5.10: *Restorations of Text.*

We also proposed an efficient minimization method to perform the blur-SURE minimization, by expressing the regularization paramter λ in terms of the PSF paramter.



(1-a) observed *biosample* 1

(1-b) blur-SURE minimization

(1-c) restored 1

(2-a) observed *biosample* 2

(2-b) blur-SURE minimization

(2-c) restored 2

(3-a) observed *biosample* 3

(3-b) blur-SURE minimization

(3-c) restored 3

Figure 5.11: *Restorations of Biosample.*

Results obtained show that the proposed method has significant improvement of quality both numerically and visually. The examples of the blur kernel function listed in this chapter are but the exemplifications of blur-SURE framework for PSF estimation problem. It is worth noting that SURE-type minimization itself does not specify any particular parametric form of PSF. There is huge potential to develop specific algorithms for various application, e.g. fluorescence microscopy [102; 105; 136] and motion deblurring [90; 124; 129; 130], based on the SURE-type principle.

Chapter 6

The Blur-SURE Approach to Motion Deblurring

6.1 Introduction

6.1.1 Background of motion deblurring

As one of the most common blur types, motion blur frequently occurs in real applications, e.g. camera shake [90; 91] and sports photography [64], when there is relative motion between the camera and the object being captured. Hence, motion deblurring, which aims at recovering a visually sharp image from a single, motion-blurred photograph, has attracted considerable attention in recent years. If one assumes that the underlying unknown blur kernel, describing the motion blur, is shift-invariant, the problem reduces to that of image blind deconvolution.

As same as the discussion in Chapter 4, we tackle the motion deblurring problem by first estimating the kernel, and then performing non-blind deconvolution with the estimated blur kernel [90; 92; 127–130], where the kernel estimation is essential to blind deconvolution.

6.1.2 Related work: parametric estimation of motion blur

For simplicity, motion blur is usually considered as one-dimensional, i.e. unidirectional with constant speed. Thus, motion blur is completely characterized by blur orientation and blur length. In this context, the blur identification reduces to estimate the two blur parameters. Refer to [97] for a systematic review of parametric estimation of motion blur.

[98; 106] uses Radon transform taken along different angles to detect the motion direction. [97] applies the steerable filter [146] to the power spectrum of the blurred image, and then, detect the maximum response value. [97; 106; 147] compute the 2-D cepstrum of the blurred image to find the two pronounced negative peaks in the cepstrum. The cepstral analysis has been further studied by [148–150], based on image gradients. Recently, more models and criteria for kernel estimation have been proposed, e.g. α-motion blur constraint model [151] and generalized cross-validation (GCV) criterion [152]. [153; 154] employ a high-pass filter (e.g. derivative operation) to the blurred image, and estimate the blur parameters based on the statistics of the image gradients.

6.1.3 Outline of this chapter

Taking linear motion blur for example, this chapter will exemplify and demonstrate the blur-SURE framework proposed in Chapter 4, to estimate the blur direction and blur length, in the following lines:

- to show the closeness of the blur-SURE to blur-MSE (as its *oracle* version), in the context of motion deblurring;
- to demonstrate the high accuracy of the estimation of blur direction and blur length, under various noise levels;
- to develop a fast algorithm of the blur-SURE minimization, in the context of motion deblurring.

6.2 Parametric estimation of motion blur: problem formulation

6.2.1 Parametrized form of linear motion blur

If the motion blur is linearly constant, the blur kernel $h(u,v)$ can be expressed as [97; 106]:

$$h(u,v) = \begin{cases} \frac{1}{L} & \text{if} \quad \sqrt{u^2+v^2} \leq \frac{L}{2} \quad \text{and} \quad \frac{u}{v} = -\tan\theta \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

with two key parameters: (1) blur length L measured by the pixel number; (2) blur orientation θ evaluated by the angle w.r.t. the horizontal direction in degrees. Thus, the blur identification problem boils down to estimating the two parameters: L and θ, from the observation $\mathbf{y}$ [97; 98; 106; 149].

6.2.2 The blur-SURE framework to motion blur estimation

Recalling Eq.(6.1), we denote the motion-blur kernel with blur length L and angle θ by $h_{L,\theta}$, and the corresponding convolution matrix by $\mathbf{H}_{L,\theta}$. Thus, we formulate the

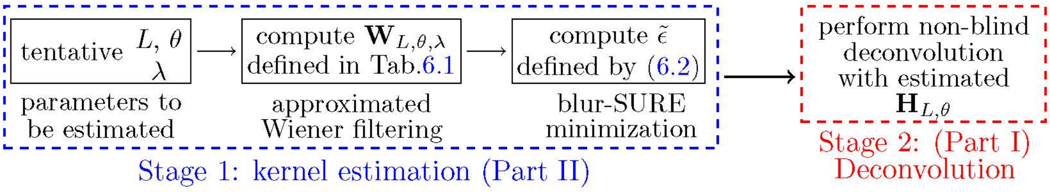

tentative L, θ λ
parameters to be estimated
compute $\mathbf{W}_{L,\theta,\lambda}$ defined in Tab.6.1
approximated Wiener filtering
compute $\tilde{\epsilon}$ defined by (6.2)
blur-SURE minimization
Stage 1: kernel estimation (Part II)
perform non-blind deconvolution with estimated $\mathbf{H}_{L,\theta}$
Stage 2: (Part I) Deconvolution

Figure 6.1: *The procedure of motion blur estimation: joint minimization of the blur-SURE over L, θ and λ, as shown in (6.2).*

motion blur estimation as the following joint minimization problem:

$$\min_{L,\theta,\lambda} \underbrace{\frac{1}{N}\left\|\mathbf{H}_{L,\theta}\mathbf{W}_{L,\theta,\lambda}\mathbf{y}-\mathbf{y}\right\|^2+\frac{2\sigma^2}{N}\mathrm{Tr}(\mathbf{H}_{L,\theta}\mathbf{W}_{L,\theta,\lambda})-\sigma^2}_{\text{blur-SURE: }\tilde{\epsilon}(L,\theta,\lambda)} \tag{6.2}$$

over three decision variables: L, θ and λ, where the notations are explained in Table 6.1. Figure 6.1 summarizes the blur-SURE approach to parametric estimation of motion blur.

Expressions	Exact	Approximation
Notations	Wiener filtering	
Matrix	$\mathbf{W}_{L,\theta}=\left(\mathbf{H}_{L,\theta}^{\mathrm{T}}\mathbf{H}_{L,\theta}+\mathbf{H}_{L,\theta}^{\mathrm{T}}\mathbf{C}\mathbf{H}_{L,\theta}^{-\mathrm{T}}\mathbf{S}^{-1}\right)^{-1}\mathbf{H}_{L,\theta}^{\mathrm{T}}$	$\mathbf{W}_{L,\theta,\lambda}=\left(\mathbf{H}_{L,\theta}^{\mathrm{T}}\mathbf{H}_{L,\theta}+\lambda\mathbf{R}\right)^{-1}\mathbf{H}_{L,\theta}^{\mathrm{T}}$
Frequency	$W_{L,\theta}(\omega)=\frac{H_{L,\theta}^*(\omega)}{\lvert H_{L,\theta}(\omega)\rvert^2+C(\omega)/S(\omega)}$	$W_{L,\theta,\lambda}(\omega)=\frac{H_{L,\theta}^*(\omega)}{\lvert H_{L,\theta}(\omega)\rvert^2+\lambda\lVert\omega\rVert^2}$
Notations	Frequency-band indicator	
Matrix	$\mathbf{U}_{L,\theta}=\mathbf{H}_{L,\theta}\left(\mathbf{H}_{L,\theta}^{\mathrm{T}}\mathbf{H}_{L,\theta}+\mathbf{H}_{L,\theta}^{\mathrm{T}}\mathbf{C}\mathbf{H}_{L,\theta}^{-\mathrm{T}}\mathbf{S}^{-1}\right)^{-1}\mathbf{H}_{L,\theta}^{\mathrm{T}}$	$\mathbf{U}_{L,\theta,\lambda}=\mathbf{H}_{L,\theta}\left(\mathbf{H}_{L,\theta}^{\mathrm{T}}\mathbf{H}_{L,\theta}+\lambda\mathbf{R}\right)^{-1}\mathbf{H}_{L,\theta}^{\mathrm{T}}$
Frequency	$U_{L,\theta}(\omega)=\frac{\lvert H_{L,\theta}(\omega)\rvert^2}{\lvert H_{L,\theta}(\omega)\rvert^2+C(\omega)/S(\omega)}$	$U_{L,\theta,\lambda}(\omega)=\frac{\lvert H_{L,\theta}(\omega)\rvert^2}{\lvert H_{L,\theta}(\omega)\rvert^2+\lambda\lVert\omega\rVert^2}$

Table 6.1: *Notations of exact and approximated Wiener filterings and frequency-band indicators, in the context of motion deblurring.*

6.3 An example of the blur-SURE approach to motion blur estimation

Figure 6.2 shows an example of minimizing blur-SURE to estimate the blur direction and length. Figure 6.2-(3-4) show that

- the very accurate estimates of θ and L by minimizing the blur-SURE, as shown in Corollary 4.3;
- the closeness of blur-SURE to blur-MSE, which indicates that blur-SURE is a reliable estimator of blur-MSE, as stated in Theorem 4.2.

We can see that the blur-SURE minimization shown in Eq.(6.2) may have multiple local minima of blur length L. However, it does not affect the validity of the conclusion

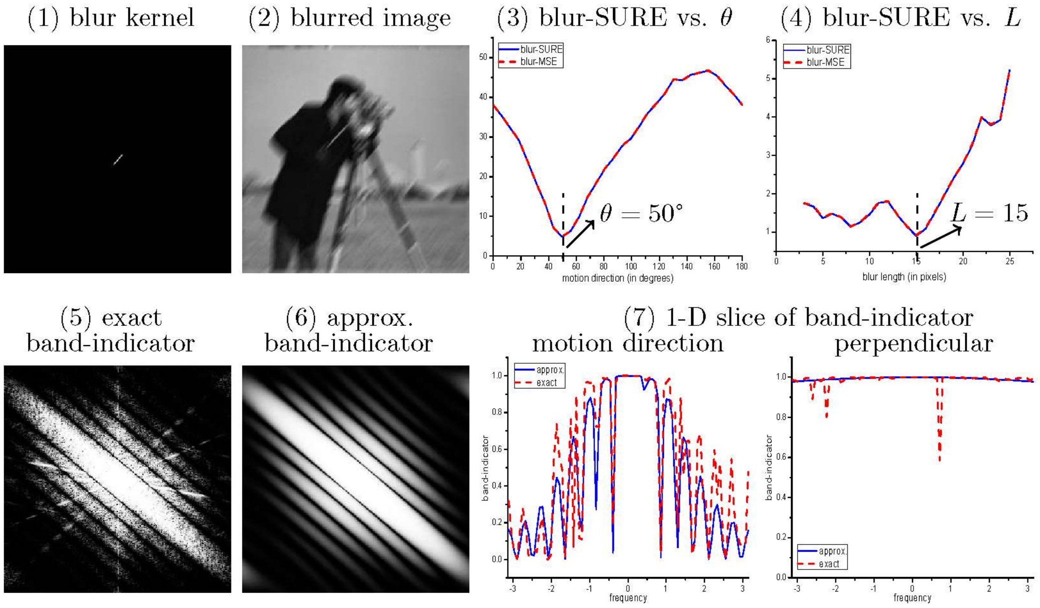


Figure 6.2: *Example of Cameraman, blurred by linear motion kernel with* $\theta_0 = 50°$ *and* $L_0 = 15$*pixels, corrupted by noise with BSNR=30dB.*

that the global minimum is always consistent with the latent true value. To show the validity of Corollary 4.3, Figure 6.2-(5-6) demonstrate that the band-indicator computed by the estimated L, θ and λ is a good approximation to exact one. Figure 6.2-(7) clearly shows the 1-D curve comparisons, by selecting two representative directions from the 2-D images (5-6).

Additional remark From the discussions mentioned above, we found many similarities between the proposed blur-SURE and GCV criterion introduced in [152]: (1) both blur-SURE and GCV deals only with the magnitude of Fourier coefficients, whereas the phase cannot be identified; (2) both of them may have many local minima. However, the proposed blur-SURE is easier to minimize than GCV, as it has a smaller number of parameters need to be optimized than GCV. Also, it has been justified that the global minimum of blur-SURE corresponds to the accurate estimates of the parameters. Therefore, we need to investigate and develop a safe and efficient minimization algorithm, which guarantees that the global minimum is reached at a faster speed.

6.4 Implementation issues

For motion blur estimation, we experimentally obtained the following observations, which may dramatically simplify the minimization procedure of Eq.(6.2).

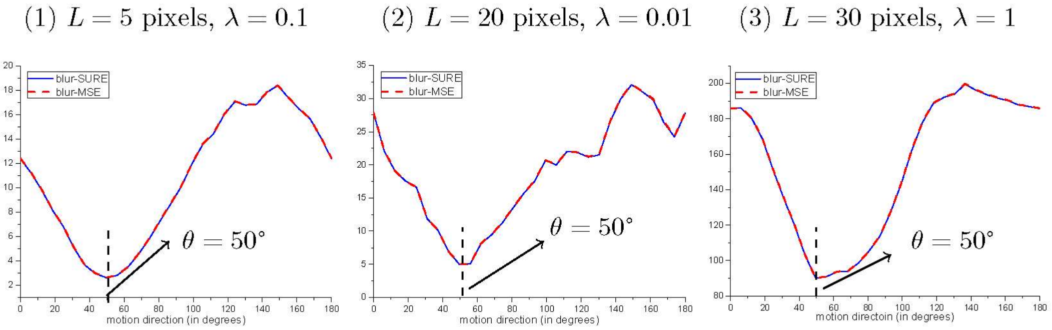

(1) $L = 5$ pixels, $\lambda = 0.1$ (2) $L = 20$ pixels, $\lambda = 0.01$ (3) $L = 30$ pixels, $\lambda = 1$

Figure 6.3: *Example of Cameraman blurred by true motion kernel with* $\theta_0 = 50$°*and* $L_0 = 15$ *pixels, corrupted by noise with BSNR=30dB.*

6.4.1 Estimation of motion direction

Firstly, we found that the accuracy of direction estimation does not strongly depend on the tentative values of L and λ. In other words, the blur-SURE minimization is significantly more sensitive to θ than to L and λ, and therefore, this criterion is not demanding in finding "optimal" L and λ, at the first stage of estimating the angle θ. It is probably because that the band-indicator of Eq.(4.8) is prominently directional in Fourier domain, as shown in Figure 6.2-(5-6). Thus, θ, as the parameter describing the directionality, is the most dominant factor of the band-indicator. Consequently, the blur-SURE minimization behaves more like finding θ to match the two directional band-indicators (i.e. both sides of Eq.(4.8)), rather than optimizing L and λ.

Figure 6.3 shows three examples to empirically verify the first claim: minimizing blur-SURE could always accurately detect the blur angle, even with arbitrary (non-optimal) values of L and λ.

6.4.2 Estimation of blur length

It is highly preferred to estimate the blur direction first, and then, use the estimated θ to estimate blur length. At the second stage, we have to perform joint minimization of both L and λ, since unlike θ, the tentative value of λ affects the length estimation. The exhaustive search for L can be performed as follows: considering a certain range of L, for each tentative value of L, we can find an optimal λ by minimizing the blur-SURE. Thus, the optimal function $\lambda = \lambda(L)$ is established, and inserted into the blur-SURE. Finally, we minimize the blur-SURE over L only. Figure 6.4 shows the three examples of minimizing the blur-SURE over L (line search), incorporating the function $\lambda = \lambda(L)$.

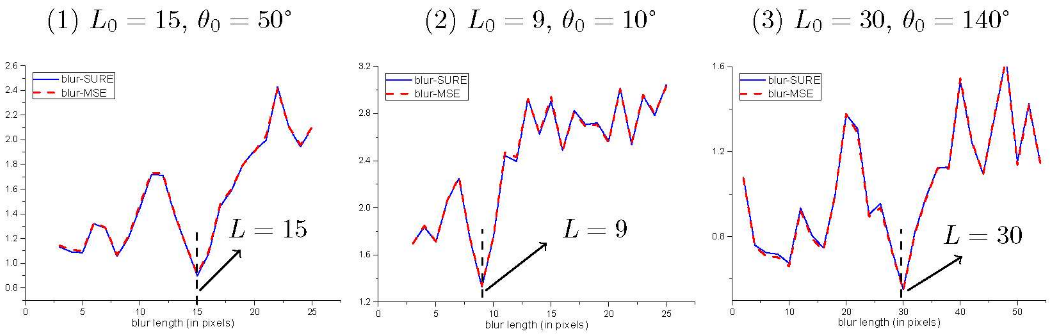

(1) $L_0 = 15$, $\theta_0 = 50°$ (2) $L_0 = 9$, $\theta_0 = 10°$ (3) $L_0 = 30$, $\theta_0 = 140°$

Figure 6.4: *Example of Cameraman blurred by motion kernel, corrupted by noise with BSNR=30dB.*

6.4.3 Short summary

Based on all the discussions mentioned above, the blur-SURE minimization is summarized as the following Algorithm 6.1.

```
Algorithm 6.1: Blur-SURE Minimization Algorithm For Motion Kernel Estimation
Input:  $\tilde{\epsilon}(L, \theta, \lambda)$ — objective function given as Eq.(6.2);
Output:  optimal $\theta$ and $L$
begin
    1. line search for optimal $\theta = \arg\min_\theta \epsilon(L, \lambda)$, with tentative $L = 20$ and $\lambda = 0.1$;
    2. exhaustive search for optimal $L$ and $\lambda$, with estimated $\theta$:
    begin
        1. take sample-values $L_k$ within a certain range of $L$, for $k = 1, 2, ..., K$;
        2. for each $L_k$, find the optimal $\lambda_k = \arg\min_\lambda \epsilon(\theta, L_k)$, and establish $\lambda = \lambda(L)$;
        3. incorporate $\lambda = \lambda(L)$ into $\epsilon(L, \lambda)$;
        4. line search for optimal $L = \arg\min_L \epsilon(L, \lambda(L))$.
    end
end
```

6.5 Results and discussions

6.5.1 Experimental setting

We consider the following two motion blur kernels:

- $\theta_0 = 40°$, $L_0 = 15$ pixels;
- $\theta_0 = 140°$, $L_0 = 35$ pixels;

The blurred images are subsequently contaminated by i.i.d Gaussian noise with various variance σ^2, which is measured by blur signal-to-noise ratio (BSNR), defined as $\text{BSNR} = 10\log_{10}\left(\frac{\|\mathbf{H}_0\mathbf{x}-\text{mean}(\mathbf{H}_0\mathbf{x})\|^2}{N\sigma^2}\right)$. We use *Cameraman*, *House* and *Mandrill* shown in Figure 5.6 as the test images.

6.5.2 Estimations of blur direction and length

Now, we compare our estimated kernel parameters to that by other state-of-the-art methods, including Cepstral method [150], Radon method [97] and refined Radon method [98]. The estimation performance is evaluated by the error: $e_\theta = \theta - \theta_0$ and $e_L = L - L_0$.

Tables 6.2 and 6.3 show the comparisons. We can see that: (1) the proposed blur-SURE method consistently outperforms the other methods, and is quite robust to high noise levels; (2) the results by the blur-MSE minimization is exactly the same with that by minimizing blur-SURE.

BSNR (in dB)	40	30	20	10	40	30	20	10	40	30	20	10
Image	*Cameraman*				*House*				*Mandrill*			
blur type	$\theta_0 = 40°,\quad L_0 = 15$ pixels											
Cepstral [150]	-1	-1	-5	-9	5	5	0	-5	-4	-5	3	-10
Radon [97]	3	5	9	11	3	-4	-10	-15	2	3	-8	-12
refined Radon [98]	**1**	**1**	5	6	**-1**	**-1**	-2	-13	**1**	**0**	-2	-4
blur-SURE	**1**	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**-1**	-1	**-1**	**-1**
blur-MSE	*1*	*1*	*1*	*0*	*1*	*1*	*0*	*0*	*-1*	*-1*	*-1*	*-1*
blur type	$\theta_0 = 140°,\quad L_0 = 35$ pixels											
Cepstral [150]	1°	1	5	-5	3	4	3	-11	2	2	2	-4
Radon [97]	-4	-5	-5	-6	-3	-5	-6	-6	0	-1	-3	-4
refined Radon [98]	**0**	**0**	**0**	1	**0**	**0**	1	1	**0**	**0**	**0**	**0**
blur-SURE	**0**	**0**	**0**	**0**	1	1	**1**	**0**	**0**	**0**	**0**	**0**
blur-MSE	*0*	*0*	*0*	*0*	*1*	*1*	*1*	*0*	*0*	*0*	*0*	*0*

Table 6.2: *Estimation error of blur orientation θ under different BSNR*

6.5.3 Motion deblurring: the synthetic experiments

In this subsection, we present the motion deblurring results and compare them with the following state-of-the-art methods: J.Oliveira, *et al*'s work [98], Taeg Cho, *et al*'s work [155], Sungh Cho, *et al*'s work [129], Q.Shan, *et al*'s work [91] and L.Xu, *et al*'s work [145]. For [91; 129; 145; 155], we use the parameters by default or by the authors'

BSNR (in dB)	40	30	20	10	40	30	20	10	40	30	20	10
Image	*Cameraman*				*House*				*Mandrill*			
blur type	$\theta_0 = 40°, \quad L_0 = 15$ pixels											
Cepstral [150]	1	1	1	7	2	2	1	4	1	0	1	23
[98]	-2	-2	-2	-2	-2	-2	-2	**0**	-2	-2	-2	-1
blur-SURE	**0**	**0**	**0**	**0**	**0**	**0**	**0**	-1	**0**	**0**	**0**	**0**
blur-MSE	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*-1*	*0*	*0*	*0*	*0*
blur type	$\theta_0 = 140°, \quad L_0 = 35$ pixels											
Cepstral [150]	-7	-7	-8	-9	-6	-6	-16	-12	-2	-2	-2	-6
[98]	-2	-2	-2	-2	-2	-2	-2	-2	-5	-5	-5	-5
blur-SURE	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
blur-MSE	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*

Table 6.3: *Estimation error of blur length L under different BSNR*

recommendations. For [98] and our work (denoted by 'blur-SURE' in Table 6.4), we perform motion deblurring using our recently proposed SURE-LET algorithm [6], with the estimated kernel by [98] and blur-SURE, respectively.

The deconvolution performance is measured by PSNR defined as Eq.(1.11). Table 6.4 shows PSNR results of the existing methods, Figure 6.5 shows an example of visual comparison. From the results, we can see that our deblurring results outperform the others significantly, both numerically and visually.

BSNR (in dB)	40	30	20	10	40	30	20	10	40	30	20	10
Image	*Cameraman*				*House*				*Mandrill*			
blur type	$\theta_0 = 40°, \quad L_0 = 15$ pixels											
input	20.53	20.51	20.31	18.68	23.17	23.15	22.95	21.38	20.41	20.40	20.33	19.68
[98]	22.31	22.01	21.79	21.34	25.76	25.46	25.39	24.41	21.92	21.52	21.18	20.69
[155]	22.45	22.10	21.86	20.73	23.45	23.30	23.27	22.93	21.67	21.55	21.33	20.81
[129]	22.31	22.25	22.76	19.74	25.17	25.73	24.16	23.31	23.33	22.87	22.31	20.58
[91]	24.51	23.69	22.35	—	22.61	23.03	23.04	20.01	21.51	21.52	21.06	18.74
[145]	25.28	25.02	21.39	—	23.92	24.24	23.01	—	21.74	21.89	20.71	18.85
blur-SURE	**27.36**	**26.59**	**24.62**	**22.43**	**31.53**	**31.07**	**29.33**	**26.67**	**26.01**	**24.83**	**22.74**	**21.13**
blur type	$\theta_0 = 140°, \quad L_0 = 35$ pixels											
input	17.87	17.86	17.76	16.94	20.54	20.53	20.44	19.64	19.18	19.18	19.13	18.72
[98]	21.18	21.05	20.89	20.31	24.27	24.16	24.12	23.65	20.02	19.82	19.79	19.78
[155]	20.56	20.52	20.34	20.04	25.56	25.40	25.32	24.34	20.51	20.44	20.34	20.00
[129]	18.33	17.89	17.40	18.78	21.02	21.06	24.73	18.39	20.79	19.60	18.96	**20.72**
[91]	21.48	21.37	19.14	—	20.91	20.94	20.79	14.86	20.35	20.30	19.57	17.15
[145]	22.54	22.40	19.26	—	23.31	23.31	21.44	—	19.94	19.57	19.30	—
blur-SURE	**29.74**	**26.22**	**23.27**	**21.08**	**34.15**	**30.84**	**27.97**	**24.78**	**27.76**	**24.26**	**21.91**	20.45

Table 6.4: *PSNR (in dB) of motion deblurring by several state-of-the-art methods*

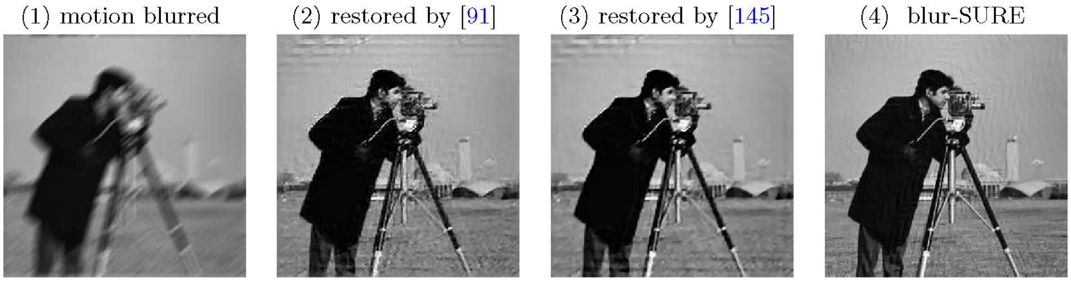

(1) motion blurred (2) restored by [91] (3) restored by [145] (4) blur-SURE

Figure 6.5: *An example of visual comparison.*

6.5.4 Motion deblurring: the real experiment

In our last set of experiments, we test the proposed algorithms on the benchmark data in a real-world database, established by [156]: *Church* shown in Figure 6.6, *Clock* shown in Figure 6.7 and *Backyard* shown in Figure 6.8. The authors of [156] recorded and analysed real camera motion on a robot platform, which allows them to record a sequence of sharp images and the unconstrained camera motion with full 6-D degree of freedom at micrometer accuracy. Refer to [156] for the detailed procedure of the robotic setup. The dataset contains both uniform and non-uniform blurs: the underlying blurs in Figure 6.6 and Figure 6.8 are approximately uniform, whereas the blur of Figure 6.7 is non-uniform across the image [156]. We compare our results with the following recently proposed works:

- **Fergus [90]:** pioneering work of motion deblurring, which combines variational approach [157] with natural image statistics [158].

- **Shan [91]:** MAP formulation, where the kernel is assumed to follow exponentially distribution. The image prior combines natural image statistics and constraints of image gradients.

- **Xu [145]:** assuming uniform blur, refined the work of [91], by introducing new inference strategies and fast optimization technique [129].

- **Cho [129]:** assuming uniform blur, predicted strong edges from an estimated latent image. This work accelarated the computation by introducing a number of techniques.

- **Hirsch [159]:** assuming the underlying blur is non-uniform, deduced a fast forward model for camera shake.

- **Whyte [128]:** assuming the underlying blur is non-uniform, derived the geometric properties of camera shake.

For deblurring color images, all the works mentioned above except for [159] dealt with three color channels independently. The work of [159] has special handling which enforces constraints in YUV space. For our algorithm, we transform the color image into the YUV space, and then estimate the PSF parameters from the Y channel, perform deconvolution on Y, U and V channels with the estimated kernel from the Y channel. Finally, we transform YUV back to RGB representation. We also applied our deblurring algorithm to RGB channels independently, using the estimated kernel from Y channel. We found that working in the YUV space yields very similar visual results than in RGB.

From the three figures, we can see that our restored image is comparable to other results, in terms of visual quality. Notice that our result in Figure 6.7 is not very satisfactory. There are two main reasons: (1) the underlying blur is spatially varying across the image [156], however, our approach assumed uniform blur; (2) the underlying blur is not simple linear motion blur, which cannot be accurately expressed by our two parameters L and θ.

In addition to the benchmark test data, we also apply the method to a real motion-blurred image *Pavilion* captured by a digital camera, shown in Figure 6.9-(1). The estimated parameters of the motion blur from Figure 6.9-(1) are $\theta = 0°$and $L = 15$ pixels, obtained by our approach. We use our developed SURE-LET approach to perform the non-blind deconvolution with the estimated blur kernel [6]. Figure 6.9-(2-6) show the restored images by various methods. Figure 6.10 shows two zoom-in parts for better comparison. Regarding the computational time, the work of [145], as the most efficient algorithm to our best knowledge, takes around 37 seconds. For our method, notice that the blur-SURE can be completely computed in Fourier domain only: there is no need to perform inverse Fourier transform. By this technique, the running time of our algorithm is around 33 seconds, where finding PSF parameters takes 15 seconds, and SURE-LET deconvolution takes another 18 seconds. Furthermore, we experimentally found that it is not necessary to optimize the value of λ in Eq.(6.2) when estimating the PSF parameters, since any value of λ between 10^{-4} and 10^{-1} could yield accurate estimate of motion length. Based on this empirical observation,

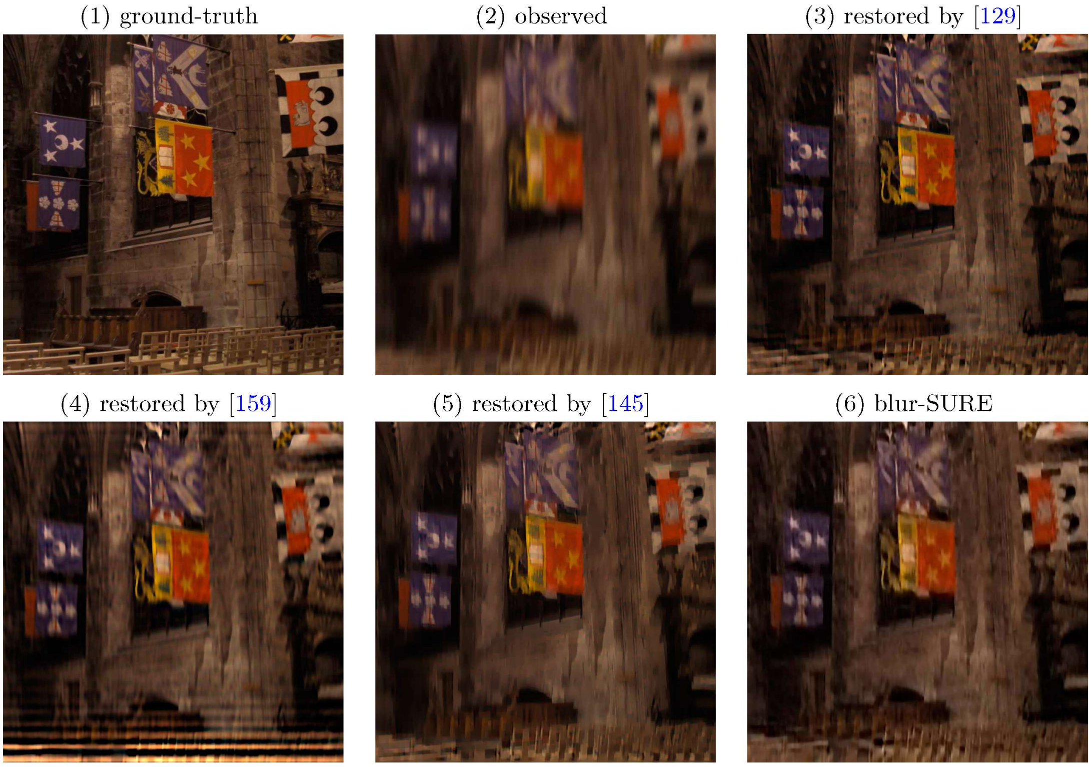


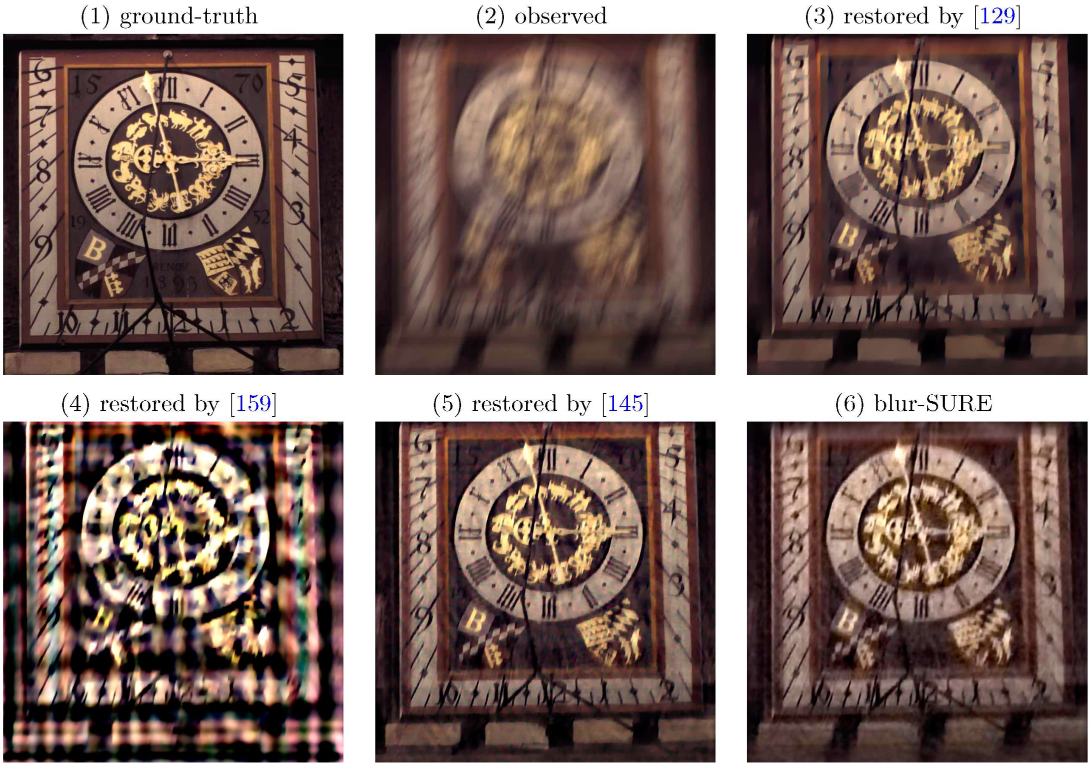
Figure 6.6: *Restoration of Church: visual comparison between various motion deblurring approaches.*

the computation time of our algorithm is further reduced to 24 seconds, where finding PSF parameters takes around 6 seconds[1].

6.6 Conclusion

Results obtained show that the proposed method has significant improvement of quality both numerically and visually. Compared to the other methods, the main advantage of our approach is that 1) the estimation of both blur angle and length is performed in a unified framework, instead of individually applying Radon and cepstral methods; 2) it does not need to analyze 2-D cepstrum and measure the prominent peaks [97; 98; 106; 149], which is easily affected by the severe noise corruption.

[1] All the computation time was recorded in the same hardware environment: Pentium(R) 4 CPU 3.00GHz, memory size 3.50GB.



(1) ground-truth

(2) observed

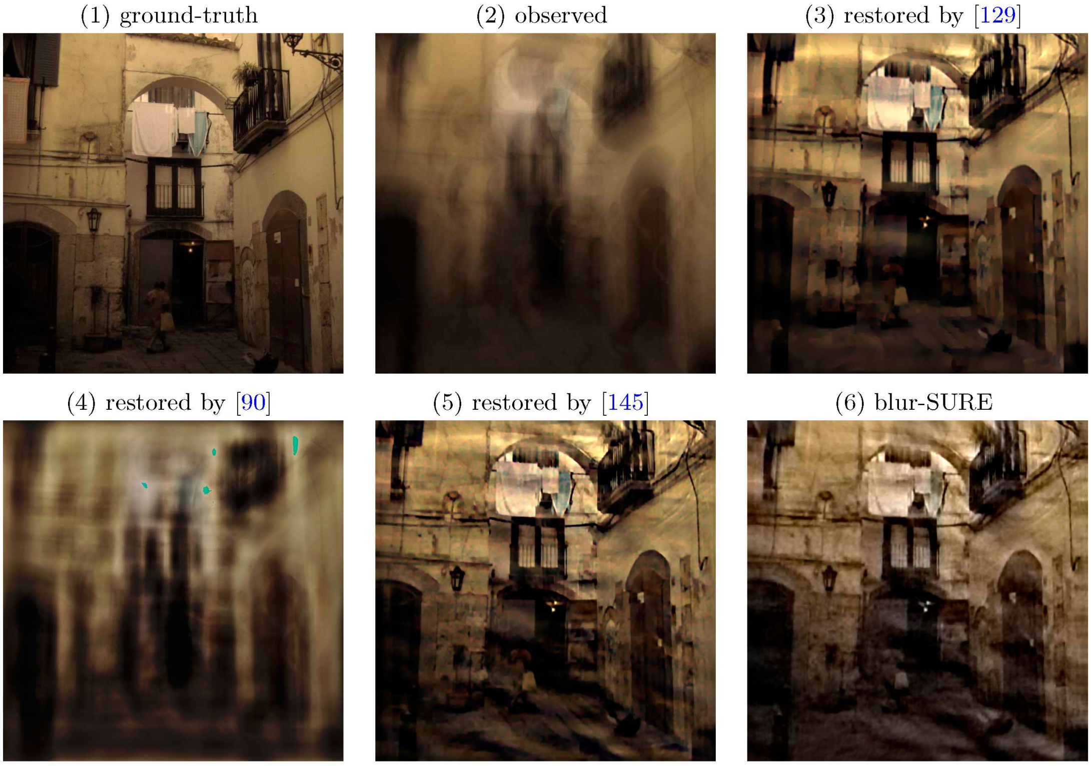
(3) restored by [129]

(4) restored by [159]

(5) restored by [145]

(6) blur-SURE

Figure 6.7: *Restoration of Clock: visual comparison between various motion deblurring approaches.*



(1) ground-truth

(2) observed

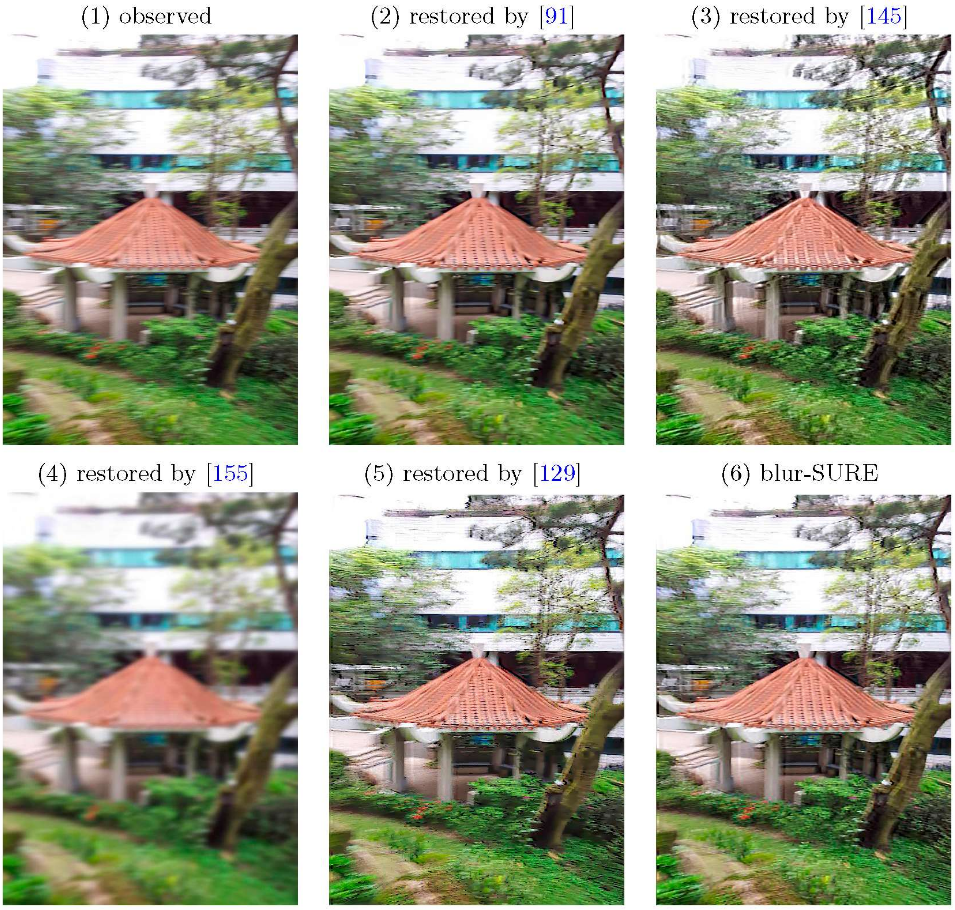
(3) restored by [129]

(4) restored by [90]

(5) restored by [145]

(6) blur-SURE

Figure 6.8: *Restoration of Backyard: visual comparison between various motion deblurring approaches.*



(1) observed

(2) restored by [91]

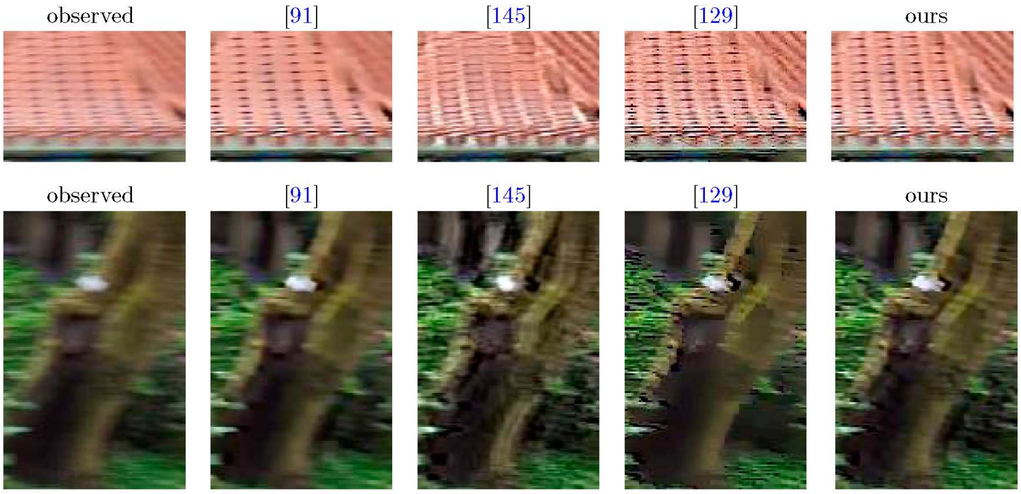
(3) restored by [145]

(4) restored by [155]

(5) restored by [129]

(6) blur-SURE

Figure 6.9: *Restoration of image Pavilion: visual comparison between various motion deblurring approaches.*



observed

[91]

[145]

[129]

ours

observed

[91]

[145]

[129]

ours

Figure 6.10: *Zoom-in parts of the restorations of Pavilion.*

Chapter 7

Epilogue

In this chapter, we will conclude our work in Section 7.1. And the future work will be discussed in Section 7.2.

7.1 Summary

This thesis mainly discusses image deconvolution, which has arisen in a great many applications, for example, photography, biomedical imaging, astronomical imaging and remote sensing. A successful deconvolution algorithm restores the observed images of poor visual quality to clearer image details and lesser noise corruption, such that the practitioners are able to easily identify the contents and objects involved in the observations.

In Part I, we tackled the classical non-blind image deconvolution problem, and proposed a new objective functional: Stein's unbiased risk estimate (SURE) — an unbiased estimate of mean squared error (MSE). Instead of the estimate itself, we parametrized the whole deconvolution processing as a linear combination of several given elementary functions. In this way, the deconvolution problem becomes to find the optimal linear coefficients, which constitute the best combination of the basic processings. The linear parametrization leads to a direct (i.e. non-iterative) parameters optimization. We worked out several SURE-based estimators based on multiple Wiener filterings, applicable in arbitrary linear transformed domains. The SURE optimization automatically finds the best combination of several Wiener filterings with different but fixed regularization parameters, which is nearly equivalent to single Wiener filtering with optimal regularization parameter, in terms of deconvolution performance.

In Part II, we considered blind image deconvolution problem. We first estimate the point spread function (PSF) or blur kernel, and then, with the estimated PSF, apply our proposed SURE-LET algorithm to perform non-blind deconvolution. For the PSF estimation, we proposed a new criterion: a modified version of standard SURE, called blur-SURE. We showed that by incorporating a simple Wiener filtering, the blur-SURE minimization yields very accurate estimate of the PSF. We exemplified the blur-SURE framework with two typical applications: Gaussian kernel and motion blur.

7.2 Perspectives

We believe that the SURE-based approaches presented in this thesis could be extended along the following directions.

Other noise models We only derived the unbiased estimates of the MSE for Gaussian noise. In fact, it is possible to find unbiased estimates of the MSE for a much broader class of noise statistics.

Other inverse problems We restricted our investigations to deconvolution problem. However, the SURE theory and the blur-SURE framework are valid for any distortion matrix, which may arise in other real applications.

Other applications of blind deconvolution In this thesis, we discussed the two important applications only: Gaussian kernel and linear motion blur. Note that the blur-SURE framework itself does not specify any particular type of blur kernel, for example, one may apply the blur-SURE approach to fluorescence microscopy, where the function form of PSF has been thoroughly investigated [102; 134; 136]. It is also possible to incorporate the blur-SURE technique into learning-based deconvolution approach [65; 160].

Appendix A

Proof

A.1 Proof of Theorem 2.1

Proof Expanding the MSE (2.2) and using $\mathbf{x} = \mathbf{H}^{-1}(\mathbf{y} - \mathbf{b})$, we obtain:

$$\begin{aligned}\frac{1}{N}\mathscr{E}\left\{\left\|\mathbf{f}(\mathbf{y}) - \mathbf{x}\right\|^2\right\} &= \frac{1}{N}\left(\mathscr{E}\left\{\left\|\mathbf{f}(\mathbf{y})\right\|^2\right\} - 2\mathscr{E}\left\{\mathbf{x}^{\mathrm{T}}\mathbf{f}(\mathbf{y})\right\} + \mathscr{E}\left\{\left\|\mathbf{x}\right\|^2\right\}\right) \\ &= \frac{1}{N}\mathscr{E}\left\{\left\|\mathbf{f}(\mathbf{y})\right\|^2\right\} - \frac{2}{N}\mathscr{E}\left\{\mathbf{y}^{\mathrm{T}}\mathbf{H}^{-\mathrm{T}}\mathbf{f}(\mathbf{y})\right\} \\ &+ \frac{2}{N}\mathscr{E}\left\{\mathbf{b}^{\mathrm{T}}\mathbf{H}^{-\mathrm{T}}\mathbf{f}(\mathbf{y})\right\} + \frac{1}{N}\mathscr{E}\left\{\left\|\mathbf{x}\right\|^2\right\}\end{aligned} \tag{A.1}$$

Consider the multivariate Gaussian probability density function $q(\mathbf{b}) \propto \exp(-\frac{\mathbf{b}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{b}}{2})$. It satisfies $q(\mathbf{b})\mathbf{b} = -\mathbf{C}\nabla_{\mathbf{b}}q(\mathbf{b})$ where $\nabla_{\mathbf{b}}$ is the gradient operator w.r.t. $\mathbf{b}$. Hence,

$$\begin{aligned}\mathscr{E}\left\{\mathbf{b}^{\mathrm{T}}\mathbf{H}^{-\mathrm{T}}\mathbf{f}(\mathbf{y})\right\} &= \int \mathbf{b}^{\mathrm{T}}\mathbf{H}^{-\mathrm{T}}\mathbf{f}(\mathbf{y})q(\mathbf{b})\mathrm{d}\mathbf{b} \\ &= -\int (\nabla_{\mathbf{b}}q(\mathbf{b}))^{\mathrm{T}}\underbrace{\mathbf{C}\mathbf{H}^{-\mathrm{T}}\mathbf{f}(\mathbf{y})}_{\mathbf{u}(\mathbf{y})}\mathrm{d}\mathbf{b} \\ &= -\sum_{n=1}^{N}\int \frac{\partial q(\mathbf{b})}{\partial b_n}u_n(\mathbf{y})\mathrm{d}\mathbf{b}\end{aligned} \tag{A.2}$$

Noting that $\int_{-\infty}^{\infty}\frac{\partial q(\mathbf{b})}{\partial b_n}u_n\mathrm{d}b_n = -\int_{-\infty}^{\infty}\frac{\partial u_n}{\partial b_n}q(\mathbf{b})\mathrm{d}b_n$, which follows from integration by parts, and the fact that $\left|u_nq(\mathbf{b})\right| \to 0$ as $|b_n| \to \infty$, we have

$$\begin{aligned}\mathscr{E}\left\{\mathbf{b}^{\mathrm{T}}\mathbf{H}^{-\mathrm{T}}\mathbf{f}(\mathbf{y})\right\} &= \sum_{n=1}^{N}\int_{-\infty}^{\infty}\frac{\partial u_n}{\partial b_n}q(\mathbf{b})\partial\mathbf{b} = \mathscr{E}\left\{\sum_{n=1}^{N}\frac{\partial u_n}{\partial b_n}\right\} \\ &= \mathscr{E}\left\{\sum_{n=1}^{N}\frac{\partial u_n}{\partial y_n}\right\} = \mathscr{E}\left\{\mathrm{div}_{\mathbf{y}}\mathbf{u}\right\}\end{aligned} \tag{A.3}$$

Substituting (A.22) into (A.21) completes the proof. ∎

A.2 Proof of Eq.(2.6) in Section 2.4.2

Proof Similar to Appendix A.1, using $\mathbf{Hx} = \mathbf{y} - \mathbf{b}$, we expand the modified MSE as

$$\begin{aligned}\frac{1}{N}\mathscr{E}\left\{\left\|\mathbf{f}(\mathbf{y}) - \mathbf{H}_\beta^{-1}\mathbf{Hx}\right\|^2\right\} &= \frac{1}{N}\left(\mathscr{E}\left\{\left\|\mathbf{f}(\mathbf{y})\right\|^2\right\} - 2\mathscr{E}\left\{\mathbf{x}^\mathrm{T}\mathbf{H}^\mathrm{T}\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{f}(\mathbf{y})\right\} + \mathscr{E}\left\{\left\|\mathbf{H}_\beta^{-1}\mathbf{Hx}\right\|^2\right\}\right)\\ &= \frac{1}{N}\left(\mathscr{E}\left\{\left\|\mathbf{f}(\mathbf{y})\right\|^2\right\} - 2\mathscr{E}\left\{\mathbf{y}^\mathrm{T}\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{f}(\mathbf{y})\right\} + 2\mathscr{E}\left\{\mathbf{b}^\mathrm{T}\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{f}(\mathbf{y})\right\}\right)\\ &\quad + \frac{1}{N}\mathscr{E}\left\{\left\|\mathbf{H}_\beta^{-1}\mathbf{Hx}\right\|^2\right\} \qquad (A.4)\end{aligned}$$

According to Appendix A.1, we have

$$\mathscr{E}\left\{\mathbf{b}^\mathrm{T}\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{f}(\mathbf{y})\right\} = \mathscr{E}\left\{\mathrm{div}_\mathbf{y}\left\{\mathbf{CH}_\beta^{-\mathrm{T}}\mathbf{f}(\mathbf{y})\right\}\right\} \qquad (A.5)$$

Substituting (A.5) into (A.4) completes the proof. ■

A.3 Proof of Eq.(3.5) in Section 3.3.1

Proof Consider the regularized SURE, given as Eq.(2.6), and apply the orthogonality of $\mathbf{W}$. Ignoring the constant coefficients $\frac{1}{N}$, $\frac{2}{N}$ and $\frac{2}{N}\sigma^2$, the first term is:

$$\begin{aligned}\left\|\mathbf{f}(\mathbf{y})\right\|^2 &= \left\|\sum_{m,l} a_{m,l}\mathbf{W}^{-1}\theta_l(\mathbf{w}_m)\right\|^2 = \left\|\mathbf{W}^{-1}\underbrace{\sum_{m,l} a_{m,l}\theta_l(\mathbf{w}_m)}_{\mathbf{u}}\right\|^2\\ &= \mathbf{u}^\mathrm{T}\mathbf{W}^{-\mathrm{T}}\mathbf{W}^{-1}\mathbf{u} = \mathbf{u}^\mathrm{T}\underbrace{\mathbf{WW}^{-1}}_{\mathbf{I}}\mathbf{u} = \left\|\sum_{m,l} a_{m,l}\theta_l(\mathbf{w}_m)\right\|^2, \qquad (A.6)\end{aligned}$$

the second term is:

$$\begin{aligned}\mathbf{y}^\mathrm{T}\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{f}(\mathbf{y}) &= \sum_{m,l} a_{m,l}\mathbf{y}^\mathrm{T}\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{W}^{-1}\theta_l(\mathbf{w}_m) = \sum_{m,l} a_{m,l}\mathbf{y}^\mathrm{T}\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{W}^\mathrm{T}\theta_l(\mathbf{w}_m)\\ &= \sum_{m,l} a_{m,l}\left(\mathbf{WH}_\beta^{-1}\mathbf{y}\right)^\mathrm{T}\theta_l(\mathbf{w}_m) \qquad (A.7)\end{aligned}$$

The third term is

$$\mathrm{div}_\mathbf{y}\left\{\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{f}(\mathrm{y})\right\} = \sum_{m,l} a_{m,l}\mathrm{div}_\mathbf{y}\left\{\underbrace{\overbrace{\mathbf{H}_\beta^{-\mathrm{T}}\mathbf{W}^{-1}}^{\mathbf{G}}\theta_l(\mathbf{w}_m)}_{\mathbf{g}(\mathbf{y})}\right\} \qquad (A.8)$$

where $\mathrm{div}_{\mathbf{y}}\{\mathbf{g}(\mathbf{y})\}$ can be further developed as:

$$\begin{aligned}
\mathrm{div}_{\mathbf{y}}\Big\{\mathbf{g}(\mathbf{y})\Big\} &= \sum_{n=1}^{N} \frac{\partial g_n(\mathbf{y})}{\partial y_n} = \sum_{n=1}^{N}\sum_{r=1}^{R} \mathbf{G}_{n,r} \sum_{k=1}^{R} \frac{\partial[\theta_l(\mathbf{w}_m)]_r}{\partial w_k} \frac{\partial w_k}{\partial y_n} \\
&= \sum_{n=1}^{N}\sum_{r=1}^{R} \mathbf{G}_{n,r} \sum_{k=1}^{R} \frac{\partial[\theta_l(\mathbf{w}_m)]_r}{\partial w_k} [\mathbf{W}\mathbf{H}_{\lambda_m}^{-1}]_{k,n} \\
&= \sum_{k=1}^{R}\sum_{r=1}^{R} \frac{\partial[\theta_l(\mathbf{w}_m)]_r}{\partial w_k} \underbrace{\sum_{n=1}^{N} [\mathbf{W}\mathbf{H}_{\lambda_m}^{-1}]_{k,n} \mathbf{G}_{n,r}}_{\left[\mathbf{W}\mathbf{H}_{\lambda_m}^{-1}\mathbf{G}\right]_{k,r}} \\
&= \sum_{k=1}^{R}\sum_{r=1}^{R} \underbrace{\frac{\partial[\theta_l(\mathbf{w}_m)]_r}{\partial w_k}}_{\left[\mathbf{J}_{\theta_l}(\mathbf{w}_m)\right]_{r,k}} \Big[\mathbf{W}\mathbf{H}_{\lambda_m}^{-1}\mathbf{H}_{\beta}^{-\mathrm{T}}\mathbf{W}^{-1}\Big]_{k,r} \\
&= \sum_{k=1}^{R}\sum_{r=1}^{R} \Big[\mathbf{W}\mathbf{H}_{\lambda_m}^{-1}\mathbf{H}_{\beta}^{-\mathrm{T}}\mathbf{W}^{-1}\Big]_{k,r} \Big[\mathbf{J}_{\theta_l}(\mathbf{w}_m)\Big]_{r,k} \\
&= \mathrm{Tr}\Big[\underbrace{\mathbf{W}\mathbf{H}_{\lambda_m}^{-1}\mathbf{H}_{\beta}^{-\mathrm{T}}\mathbf{W}^{-1}}_{\mathbf{P}} \mathbf{J}_{\theta_l}(\mathbf{w}_m)\Big] \qquad \text{(A.9)}
\end{aligned}$$

where Tr denotes the matrix trace.

In particular, ignoring the subscript l and m, if the thresholding function $\theta(\mathbf{w})$ is pointwise processing, i.e. $[\theta(\mathbf{w})]_r = \theta(w_r)$. Thus,

$$\frac{\partial[\theta(\mathbf{w})]_r}{\partial w_k} = \frac{\partial\theta(w_r)}{\partial w_k} = \begin{cases} \theta'(w_k) & \text{if } k = r \\ 0 & \text{otherwise} \end{cases}$$

Consequently, the matrix $\mathbf{J}_{\theta}(\mathbf{w})$ becomes diagonal matrix in this case. Thus, Eq.(A.8) is simplified as:

$$\mathrm{div}_{\mathbf{y}}\Big\{\mathbf{g}(\mathbf{y})\Big\} = \sum_{k=1}^{R} \big[\mathbf{P}\big]_{k,k} \big[\mathbf{J}_{\theta}(\mathbf{w})\big]_{k,k} = \boldsymbol{\alpha}^{\mathrm{T}}\theta'(\mathbf{w}) \qquad \text{(A.10)}$$

where $\boldsymbol{\alpha} = \mathrm{diag}\{\mathbf{P}\} = [\mathbf{P}_{k,k}]_{k\in[1;N]}$, $\theta'(\mathbf{w}) = [\theta'(w_k)]_{k\in[1;N]}$.

The last term is:

$$\big\|\mathbf{H}_{\beta}^{-1}\mathbf{H}\mathbf{x}\big\|^2 = \big\|\mathbf{W}^{-1}\underbrace{\mathbf{W}\mathbf{H}_{\beta}^{-1}\mathbf{H}\mathbf{x}}_{\mathbf{u}}\big\|^2 = \mathbf{u}^{\mathrm{T}}\underbrace{\mathbf{W}\mathbf{W}^{-1}}_{\mathbf{I}}\mathbf{u} = \big\|\mathbf{W}\mathbf{H}_{\beta}^{-1}\mathbf{H}\mathbf{x}\big\|^2 \qquad \text{(A.11)}$$

Combining Eq.(A.6)–Eq.(A.11) obtains Eq.(3.5) and thus, completes the proof. ■

A.4 Proof of Theorem 3.6

Proof From Lemma 3.4, $p(n)$ can be expressed as:

$$p(n) = \sum_{k=0}^{N-1} \big[\underbrace{g_{2N}(n-k) + g_{2N}(n+k+1)}_{u(k)}\big] b(k)$$

then, using the i.i.d. condition of $b(n)$ as in (3.22), the variance of $p(n)$ is

$$\mathscr{E}\{p(n)^2\} = \mathscr{E}\left\{\left(\sum_{k=0}^{N-1} u(k)b(k)\right)^2\right\} = \sigma^2 \sum_{k=0}^{N-1} u(k)^2 \tag{A.12}$$

Due to the fact that $u(-k) = u(k-1)$ and $u(k+2N) = u(k)$, by change of variable and Parseval's theorem, we have

$$\sum_{k=0}^{N-1} u(k)^2 = \frac{1}{2}\sum_{k=0}^{2N-1} u(k)^2 = \frac{1}{4N}\sum_{k=0}^{2N-1} \big|U(k)\big|^2 \tag{A.13}$$

where $U(k)$ is the DFT of $u(k')$:

$$\begin{aligned} U(k) &= \sum_{k'=0}^{2N-1} u(k')e^{-j\frac{2\pi kk'}{2N}} \\ &= \sum_{k'=0}^{2N-1} g_{2N}(n-k')e^{-j\frac{\pi kk'}{N}} + \sum_{k'=0}^{2N-1} g_{2N}(n+k'+1)e^{-j\frac{\pi kk'}{N}} \\ &= e^{-j\frac{\pi kn}{N}} \sum_{k'=0}^{2N-1} g_{2N}(k')e^{j\frac{\pi kk'}{N}} + e^{j\frac{\pi k}{N}(n+1)} \sum_{k'=0}^{2N-1} g_{2N}(k')e^{-j\frac{\pi kk'}{N}} \end{aligned} \tag{A.14}$$

Recalling $G\big(e^{j\frac{\pi k}{N}}\big) = \sum_{n=0}^{2N-1} g_{2N}(n)e^{-j\frac{\pi kn}{N}}$, and combining with (A.14), $U(k)$ is related to $G\big(e^{j\frac{\pi k}{N}}\big)$ through:

$$U(k) = e^{-j\frac{\pi kn}{N}} G\big(e^{-j\frac{\pi k}{N}}\big) + e^{j\frac{\pi k}{N}(n+1)} G\big(e^{j\frac{\pi k}{N}}\big) \tag{A.15}$$

Substituting (A.15) into (A.13), and combining with (A.12) complete the proof. ■

A.5 Proof of Theorem 3.12

Proof Under whole-point symmetric boundary extension, Eq.(3.35) can be directly obtained by rewriting (3.33). Note that $p(n)$ is also periodic with period $(2N-2)$. However, $p(n)$ is not symmetric, if the filter $g(n)$ is not symmetric. From Eq.(3.35), for $n = 0, 1, ..., 2N-3$, the variance of $p(n)$ is:

$$\mathscr{E}\Big\{\big(p(n)\big)^2\Big\} = \mathscr{E}\Bigg\{\Bigg(\sum_{k=1}^{N-2}\Big[\underbrace{\hat{g}(n-k)+\hat{g}(n+k)}_{u(k)}\Big]b(k) + \underbrace{\hat{g}(n)}_{\frac{1}{2}u(0)}b(0) + \underbrace{\hat{g}(n-N+1)}_{\frac{1}{2}u(N-1)}b(N-1)\Bigg)^2\Bigg\}$$

Due to the whiteness of the noise $b(n)$, we obtain:

$$\mathscr{E}\Big\{\big(p(n)\big)^2\Big\} = \sigma^2\sum_{k=1}^{N-2}u(k)^2 + \frac{\sigma^2}{4}u(0)^2 + \frac{\sigma^2}{4}u(N-1)^2 \tag{A.16}$$

Now, observe that $u(k)$ satisfies $u(k) = u(-k)$ and $u(k) = u(k+2N-2)$. We consider the following formula over a particular interval of period $[-(N-2), N-1]$:

$$\begin{aligned}\sum_{k\in\mathbb{P}}u(k)^2 &= \sum_{k=-(N-2)}^{-1}u(k)^2 + \sum_{k=1}^{N-2}u(k)^2 + u(0)^2 + u(N-1)^2\\ &= 2\sum_{k=1}^{N-2}u(k)^2 + u(0)^2 + u(N-1)^2\end{aligned}$$

Hence,

$$\sum_{k=1}^{N-2}u(k)^2 = \frac{1}{2}\sum_{k\in\mathbb{P}}u(k)^2 - \frac{1}{2}u(0)^2 - \frac{1}{2}u(N-1)^2 \tag{A.17}$$

Thus, combining (A.16) with (A.17) obtains Eq.(3.36), where the first term is:

$$\sum_{k\in\mathbb{P}}u(k)^2 = \sum_{k=0}^{2N-3}u(k)^2 = \frac{1}{2N-2}\sum_{k=0}^{2N-3}\big|U(k)\big|^2$$

by Parseval's theorem, where

$$\begin{aligned}U(k) &= \sum_{k'=0}^{2N-3}\Big(\hat{g}(n-k')+\hat{g}(n+k')\Big)e^{-j\frac{2\pi kk'}{2N-2}}\\ &= \sum_{k'=0}^{2N-3}\hat{g}(n-k')e^{-j\frac{2\pi kk'}{2N-2}} + \sum_{k'=0}^{2N-3}\hat{g}(n+k')e^{-j\frac{2\pi kk'}{2N-2}}\\ &= e^{-j\frac{2\pi kn}{2N-2}}\underbrace{\sum_{k'=0}^{2N-3}\hat{g}(k')e^{j\frac{2\pi kk'}{2N-2}}}_{G\left(z^{-1}\right)} + e^{j\frac{2\pi kn}{2N-2}}\underbrace{\sum_{k'=0}^{2N-3}\hat{g}(k')e^{-j\frac{2\pi kk'}{2N-2}}}_{G\left(z\right)}\end{aligned}$$

by denoting $z = e^{j\frac{2\pi k}{2N-2}}$ and $G(z) = \sum_{k'}\hat{g}(k')z^{-k'}$. Thus, using $G(z^{-1}) = \big(G(z)\big)^*$, we have:

$$\big|U(k)\big|^2 = 2\big|G(z)\big|^2 + 2Re\Big\{\big(G(z)\big)^2 z^{2n}\Big\}$$

also, the last two terms in (3.36) are given by:

$$\begin{cases} \hat{g}(n) & = \frac{1}{2N-2}\sum_{k=0}^{2N-3} G(z)z^{n} \\ \hat{g}(n-N+1) & = \frac{1}{2N-2}\sum_{k=0}^{2N-3} G(z)z^{n-N+1} = \frac{1}{2N-2}\sum_{k=0}^{2N-3}(-1)^{k}G(z)z^{n} \end{cases}$$

Finally, Eq.(3.36) becomes (3.37) in frequency domain. ■

A.6 Derivation of noise variance in 2-D case (Section 3.5.4)

Under whole-point symmetric boundary condition of $x(m,n)$, denoting $\hat{g}(m,n) = g_{2M-2,2N-2}(m,n)$, the convolution formula is given as:

$$\begin{aligned} p(m,n) = \; & \underbrace{\hat{g}(m,n)}_{\frac{1}{4}u(0,0)} b(0,0) + \underbrace{\hat{g}(m,n-N+1)}_{\frac{1}{4}u(0,N-1)} b(0,N-1) \quad \text{(four corners)} \\ + \; & \underbrace{\hat{g}(m-M+1,n)}_{\frac{1}{4}u(M-1,0)} b(M-1,0) + \underbrace{\hat{g}(m-M+1,n-N+1)}_{\frac{1}{4}u(M-1,N-1)} b(M-1,N-1) \\ + \; & \sum_{k=1}^{M-2} \Big[\underbrace{\hat{g}(m-k,n) + \hat{g}(m+k,n)}_{\frac{1}{2}u(k,0)}\Big] x(k,0) \quad \text{(sum over first column } l=0\text{)} \\ + \; & \sum_{k=1}^{M-2} \Big[\underbrace{\hat{g}(m-k,n-N+1) + \hat{g}(m+k,n-N+1)}_{\frac{1}{2}u(k,N-1)}\Big] x(k,N-1) \text{ (last col. } l=N-1\text{)} \\ + \; & \sum_{l=1}^{N-2} \Big[\underbrace{\hat{g}(m,n-l) + \hat{g}(m,n+l)}_{\frac{1}{2}u(0,l)}\Big] x(0,l) \quad \text{(sum over first row } k=0\text{)} \\ + \; & \sum_{l=1}^{N-2} \Big[\underbrace{\hat{g}(m-M+1,n-l) + \hat{g}(m-M+1,n+l)}_{\frac{1}{2}u(M-1,l)}\Big] x(M-1,l) \text{ (last row } k=M-1\text{)} \\ + \; & \sum_{k=1}^{M-2}\sum_{l=1}^{N-2} \Big(\underbrace{\hat{g}(m-k,n-l) + \hat{g}(m-k,n+l) + \hat{g}(m+k,n-l) + \hat{g}(m+k,n+l)}_{u(k,l)}\Big) b(k,l) \end{aligned}$$

Thus, due to the whiteness of the noise $b(m,n)$, the variance of $p(m,n)$ is:

$$\begin{aligned} \mathscr{E}\Big\{p(m,n)^2\Big\} = \; & \frac{\sigma^2}{16}\Big(\underbrace{u(0,0)^2 + u(0,N-1)^2 + u(M-1,0)^2 + u(M-1,N-1)^2}_{C=\frac{1}{2}[A(0)+A(M-1)+B(0)+B(N-1)]}\Big) \\ & + \frac{\sigma^2}{4}\sum_{k=1}^{M-2}\Big(\underbrace{u(k,0)^2 + u(k,N-1)^2}_{A(k)}\Big) + \frac{\sigma^2}{4}\sum_{l=1}^{N-2}\Big(\underbrace{u(0,l)^2 + u(M-1,l)^2}_{B(l)}\Big) \\ & + \sigma^2\sum_{k=1}^{M-2}\sum_{l=1}^{N-2} u(k,l)^2 \end{aligned}$$

Considering the summation of $u(k,l)^2$ over a whole interval of period $(2M-2, 2N-2)$, we have:

$$\begin{aligned}\sum_{k\in\mathbb{P}}\sum_{l\in\mathbb{P}} u(k,l)^2 &= \sum_{k\in\mathbb{P}}\left(\sum_{l=-(N-2)}^{-1} u(k,l)^2 + \sum_{l=1}^{N-2} u(k,l)^2 + u(k,0)^2 + u(k,N-1)^2\right)\\ &= \underbrace{\sum_{k\in\mathbb{P}}\left(u(k,0)^2 + u(k,N-1)^2\right)}_{\text{fist term}} + \underbrace{2\sum_{k\in\mathbb{P}}\sum_{l=1}^{N-2} u(k,l)^2}_{\text{second term}}\end{aligned}$$

where

$$\begin{aligned}\text{first term} &= u(0,0)^2 + u(0,N-1)^2 + u(M-1,0)^2 + u(M-1,N-1)^2\\ &+ 2\sum_{k=1}^{M-2}\left(u(k,0)^2 + u(k,N-1)^2\right)\end{aligned}$$

and

$$\begin{aligned}\text{second term} &= 2\sum_{l=1}^{N-2}\left(\sum_{k=-(M-2)}^{-1} u(k,l)^2 + \sum_{k=1}^{M-2} u(k,l)^2 + u(0,l)^2 + u(M-1,l)^2\right)\\ &= 2\sum_{l=1}^{N-2}\left(u(0,l)^2 + u(M-1,l)^2\right) + 4\sum_{l=1}^{N-2}\sum_{k=1}^{M-2} u(k,l)^2\end{aligned}$$

and hence,

$$\begin{aligned}\sum_{k\in\mathbb{P}}\sum_{l\in\mathbb{P}} u(k,l)^2 &= u(0,0)^2 + u(0,N-1)^2 + u(M-1,0)^2 + u(M-1,N-1)^2\\ &+ 2\sum_{k=1}^{M-2}\left(\underbrace{u(k,0)^2 + u(k,N-1)^2}_{A(k)}\right) + 2\sum_{l=1}^{N-2}\left(\underbrace{u(0,l)^2 + u(M-1,l)^2}_{B(l)}\right)\\ &+ 4\sum_{l=1}^{N-2}\sum_{k=1}^{M-2} u(k,l)^2\end{aligned}$$

Thus, using $\sum_{(k,l)\in\mathbb{P}} u(k,l)^2$, the variance of $p(n)$ becomes:

$$\begin{aligned}\mathscr{E}\left\{p(m,n)^2\right\} &= \frac{\sigma^2}{16}C + \frac{\sigma^2}{4}\sum_{k=1}^{M-2} A(k) + \frac{\sigma^2}{4}\sum_{l=1}^{N-2} B(l)\\ &+ \frac{\sigma^2}{4}\left(\sum_{k\in\mathbb{P}}\sum_{l\in\mathbb{P}} u(k,l)^2 - C - 2\sum_{k=1}^{M-2} A(k) - 2\sum_{l=1}^{N-2} B(l)\right)\\ &= \frac{\sigma^2}{16}\left(4\sum_{k\in\mathbb{P}}\sum_{l\in\mathbb{P}} u(k,l)^2 - 3C - 4\sum_{k=1}^{M-2} A(k) - 4\sum_{l=1}^{N-2} B(l)\right)\\ &= \frac{\sigma^2}{16}\left(\frac{1}{(M-1)(N-1)}\sum_{k\in\mathbb{P}}\sum_{l\in\mathbb{P}} \left|U(k,l)\right|^2 - 3C - 4\sum_{k=1}^{M-2} A(k) - 4\sum_{l=1}^{N-2} B(l)\right)\end{aligned}$$

where $U(k,l)$, C, $A(k)$ and $B(l)$ can be directly computed from $G(z_1,z_2)$. ■

A.7 Proof of Theorem 4.1

Proof In Eq.(4.4), denoting $\mathbf{H}\mathbf{W}_\mathbf{H}$ by $\mathbf{U}$, and replacing $\mathbf{y}$ by $\mathbf{H}_0\mathbf{x}+\mathbf{b}$, the blur-MSE becomes:

$$\begin{aligned}\text{blur-MSE} &= \frac{1}{N}\mathscr{E}\Big\{\|\mathbf{U}\mathbf{y}-\mathbf{H}_0\mathbf{x}\|^2\Big\} = \frac{1}{N}\mathscr{E}\Big\{\|\mathbf{U}(\mathbf{H}_0\mathbf{x}+\mathbf{b})-\mathbf{H}_0\mathbf{x}\|^2\Big\}\\ &= \frac{1}{N}\mathscr{E}\Big\{\|(\mathbf{U}-\mathbf{I})\mathbf{H}_0\mathbf{x}+\mathbf{U}\mathbf{b}\|^2\Big\} = \frac{1}{N}\mathscr{E}\Big\{\|(\mathbf{U}-\mathbf{I})\mathbf{H}_0\mathbf{x}\|^2\Big\} + \frac{1}{N}\mathscr{E}\Big\{\|\mathbf{U}\mathbf{b}\|^2\Big\}\\ &= \frac{1}{N}\mathrm{Tr}\Big((\mathbf{U}-\mathbf{I})\mathbf{H}_0\mathbf{S}\mathbf{H}_0^{\mathrm{T}}(\mathbf{U}-\mathbf{I})^{\mathrm{T}}\Big) + \frac{1}{N}\mathrm{Tr}\big(\mathbf{U}\mathbf{C}\mathbf{U}^{\mathrm{T}}\big)\end{aligned} \tag{A.18}$$

where Tr denotes matrix trace, $\mathbf{I}$ is identity matrix, $\mathbf{S}=\mathscr{E}\{\mathbf{x}\mathbf{x}^{\mathrm{T}}\}$, the covariance matrix $\mathbf{C}=\mathscr{E}\{\mathbf{b}\mathbf{b}^{\mathrm{T}}\}$. Thus, the minimization of the blur-MSE over $\mathbf{U}$ yields that $(\mathbf{U}-\mathbf{I})\mathbf{H}_0\mathbf{S}\mathbf{H}_0^{\mathrm{T}}+\mathbf{U}\mathbf{C}=\mathbf{0}$, which implies that:

$$\mathbf{U}=\mathbf{H}_0\mathbf{S}\mathbf{H}_0^{\mathrm{T}}\big(\mathbf{H}_0\mathbf{S}\mathbf{H}_0^{\mathrm{T}}+\mathbf{C}\big)^{-1} \tag{A.19}$$

On the other hand, from (4.4) and (4.3), we obtain:

$$\mathbf{U}=\mathbf{H}\mathbf{W}_\mathbf{H}=\mathbf{H}\mathbf{S}\mathbf{H}^{\mathrm{T}}\big(\mathbf{H}\mathbf{S}\mathbf{H}^{\mathrm{T}}+\mathbf{C}\big)^{-1} \tag{A.20}$$

Combining Eq.(A.19) with Eq.(A.20) obtains $\mathbf{H}\mathbf{S}\mathbf{H}^{\mathrm{T}}=\mathbf{H}_0\mathbf{S}\mathbf{H}_0^{\mathrm{T}}$, which completes the proof. ■

A.8 Proof of Theorem 4.2

Proof In Eq.(4.2), substituting $\mathbf{y}-\mathbf{b}$ for $\mathbf{H}_0\mathbf{x}$, and expanding the blur-MSE, we obtain:

$$\begin{aligned}\text{blur-MSE} &= \frac{1}{N}\mathscr{E}\Big\{\|\mathbf{H}\mathbf{f}(\mathbf{y})-\mathbf{H}_0\mathbf{x}\|^2\Big\} = \frac{1}{N}\mathscr{E}\Big\{\|\mathbf{H}\mathbf{f}(\mathbf{y})-\mathbf{y}+\mathbf{b}\|^2\Big\}\\ &= \frac{1}{N}\left(\mathscr{E}\Big\{\|\mathbf{H}\mathbf{f}(\mathbf{y})-\mathbf{y}\|^2\Big\}+2\mathscr{E}\Big\{\mathbf{b}^{\mathrm{T}}\mathbf{H}\mathbf{f}(\mathbf{y})\Big\}-2\mathscr{E}\Big\{\mathbf{b}^{\mathrm{T}}\mathbf{y}\Big\}+\mathscr{E}\Big\{\|\mathbf{b}\|^2\Big\}\right)\\ &= \frac{1}{N}\left(\mathscr{E}\Big\{\|\mathbf{H}\mathbf{f}(\mathbf{y})-\mathbf{y}\|^2\Big\}+2\mathscr{E}\Big\{\mathbf{b}^{\mathrm{T}}\mathbf{H}\mathbf{f}(\mathbf{y})\Big\}-\mathscr{E}\Big\{\|\mathbf{b}\|^2\Big\}\right)\end{aligned} \tag{A.21}$$

Consider the multivariate Gaussian probability density function $q(\mathbf{b})\propto\exp\big(-\frac{\mathbf{b}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{b}}{2}\big)$. It satisfies $q(\mathbf{b})\mathbf{b}=-\mathbf{C}\nabla_\mathbf{b}q(\mathbf{b})$, where $\nabla_\mathbf{b}$ is the gradient operator w.r.t. $\mathbf{b}$. Hence,

$$\mathscr{E}\Big\{\mathbf{b}^{\mathrm{T}}\mathbf{H}\mathbf{f}(\mathbf{y})\Big\}=\int\mathbf{b}^{\mathrm{T}}\mathbf{H}\mathbf{f}(\mathbf{y})q(\mathbf{b})\mathrm{d}\mathbf{b}=-\int\big(\nabla_\mathbf{b}q(\mathbf{b})\big)^{\mathrm{T}}\underbrace{\mathbf{C}\mathbf{H}\mathbf{f}(\mathbf{y})}_{\mathbf{v}(\mathbf{y})}\mathrm{d}\mathbf{b}=-\sum_{n=1}^{N}\int\frac{\partial q(\mathbf{b})}{\partial b_n}v_n(\mathbf{y})\mathrm{d}\mathbf{b}$$

Noting that $\int_{-\infty}^{\infty} \frac{\partial q(\mathbf{b})}{\partial b_n} v_n \mathrm{d}b_n = -\int_{-\infty}^{\infty} \frac{\partial v_n}{\partial b_n} q(\mathbf{b}) \mathrm{d}b_n$, which follows from integration by parts, and the fact that $|v_n q(\mathbf{b})| \to 0$ as $|b_n| \to \infty$, we have:

$$\mathscr{E}\Big\{\mathbf{b}^{\mathrm{T}}\mathbf{Hf}(\mathbf{y})\Big\} = \sum_{n=1}^{N} \int_{-\infty}^{\infty} \frac{\partial v_n}{\partial b_n} q(\mathbf{b}) \partial \mathbf{b} = \mathscr{E}\left\{\sum_{n=1}^{N} \frac{\partial v_n}{\partial b_n}\right\} = \mathscr{E}\left\{\sum_{n=1}^{N} \frac{\partial v_n}{\partial y_n}\right\} = \mathscr{E}\Big\{\mathrm{div}_{\mathbf{y}}\mathbf{v}\Big\} \tag{A.22}$$

The last term in (A.21) is:

$$\mathscr{E}\Big\{\left\|\mathbf{b}\right\|^2\Big\} = \mathscr{E}\Big\{\mathbf{b}^{\mathrm{T}}\mathbf{b}\Big\} = \mathscr{E}\Big\{\mathrm{Tr}\big(\mathbf{b}\mathbf{b}^{\mathrm{T}}\big)\Big\} = \mathrm{Tr}\Big(\mathscr{E}\big\{\mathbf{b}\mathbf{b}^{\mathrm{T}}\big\}\Big) = \mathrm{Tr}\big(\mathbf{C}\big) \tag{A.23}$$

Substituting (A.22) and (A.23) into (A.21) completes the proof. ■

Bibliography

[1] T. Blu and F. Luisier, “The SURE-LET approach to image denoising,” *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2778–2786, 2007.

[2] A. Carasso, “Direct blind deconvolution,” *SIAM Journal on Applied Mathematics*, pp. 1980–2007, 2001.

[3] ——, “The APEX method in image sharpening and the use of low exponent lévy stable laws,” *SIAM Journal on Applied Mathematics*, pp. 593–618, 2002.

[4] J. Caron, N. Namazi, and C. Rollins, “Noniterative blind data restoration by use of an extracted filter function,” *Applied Optics*, vol. 41, no. 32, pp. 6884–6889, 2002.

[5] R. Molina, J. Núñez, F. Cortijo, and J. Mateos, “Image restoration in astronomy: a Bayesian perspective,” *IEEE Signal Processing Magazine*, vol. 18, no. 2, pp. 11–29, 2001.

[6] F. Xue, F. Luisier, and T. Blu, “SURE-LET image deconvolution using multiple Wiener filters,” in *Proc. of IEEE International Conference on Image Processing,* Orlando, Florida, USA, Oct. 2012, pp. 3037–3040.

[7] J. Caron, N. Namazi, R. Lucke, C. Rollins, P. Lynn Jr *et al.*, “Blind data restoration with an extracted filter function,” *Optics Letters*, vol. 26, no. 15, pp. 1164–1166, 2001.

[8] Y. Chuang, “Study on blind image separation and restoration,” Master Thesis, *National Cheng Kung University*, 2002.

[9] J. Zhang, Q. Zhang, and G. He, “Blind image deconvolution by means of asymmetric multiplicative iterative algorithm,” *J. Opt. Soc. Am. A*, vol. 25, no. 3, pp. 710–717, 2008.

[10] B. Davey, R. Lane, and R. Bates, “Blind deconvolution of noisy complex-valued image,” *Optics Communications*, vol. 69, no. 5-6, pp. 353–356, 1989.

[11] D. Fish, A. Brinicombe, E. Pike, and J. Walker, “Blind deconvolution by means of the Richardson–Lucy algorithm,” *J. Opt. Soc. Am. A*, vol. 12, no. 1, pp. 58–65, 1995.

[12] J. Zhang, Q. Zhang, and G. He, “Blind deconvolution: multiplicative iterative algorithm,” *Optics letters*, vol. 33, no. 1, pp. 25–27, 2008.

[13] D. Donoho and I. Johnstone, “Adapting to Unknown Smoothness Via Wavelet Shrinkage.” *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[14] S. Wan, B. Raju, and M. Srinivasan, "Robust deconvolution of high-frequency ultrasound images using higher-order spectral analysis and wavelets," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 50, no. 10, pp. 1286–1295, 2003.

[15] T. Sjoeberg, L. Gelius, and I. Lecomte, "2-D deconvolution of seismic image blur," *SEG Annual Meeting, Dallas, Texas*, 2003.

[16] J. Starck, E. Pantin, and F. Murtagh, "Deconvolution in astronomy: A review," *Publications of the Astronomical Society of the Pacific*, vol. 114, no. 800, pp. 1051–1069, 2002.

[17] G. Poropat, "Effect of system point spread function, apparent size, and detector instantaneous field of view on the infrared image contrast of small objects," *Optical Engineering*, vol. 32, no. 10, pp. 2598–2607, 1993.

[18] P. Sarder and A. Nehorai, "Deconvolution methods for 3-D fluorescence microscopy images," *IEEE Signal Processing Magazine*, vol. 23, no. 3, pp. 32–45, 2006.

[19] M. Born and E. Wolf, *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light.* Cambridge University Press, 1999.

[20] H. Andrews and B. Hunt, *Digital image restoration.* Prentice-Hall, Englewood Cliffs, N.J., 1977.

[21] J. Goodman, *Introduction to Fourier optics.* McGraw-hill, New York, 1968.

[22] G. Holst, *Electro-optical imaging system performance.* International Society for Optical Engineering, 2008.

[23] A. Neumaier, "Solving ill-conditioned and singular linear systems: A tutorial on regularization," *SIAM Review*, vol. 40, no. 3, pp. 636–666, 1998.

[24] P. Combettes and V. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling and Simulation*, vol. 4, no. 4, pp. 1168–1200, 2006.

[25] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series.* Wiley, New York, 1964.

[26] A. Tikhonov and V. Arsenin, *Solutions of ill-posed problems.* Winston and Sons, Washington, 1977.

[27] A. Tikhonov, "Solution of incorrectly formulated problems and the regularization method," *Soviet Math. Dokl.*, vol. 4, pp. 1035–1038, 1963.

[28] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[29] M. Zibulevsky and M. Elad, "L1–L2 optimization in signal and image processing," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 76–88, 2010.

[30] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on pure and applied mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[31] M. Elad, M. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 972–982, 2010.

[32] L. Landweber, "An iteration formula for Fredholm integral equations of the first kind," *Amer. J. Math*, vol. 73, no. 3, pp. 615–624, 1951.

[33] E. Esser, "Applications of Lagrangian-based alternating direction methods and connections to split Bregman," *UCLA CAM Report*, vol. 9, p. 31, 2009.

[34] M. Figueiredo and R. Nowak, "An em algorithm for wavelet-based image restoration," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 906–916, 2003.

[35] M. Figueiredo, J. Bioucas-Dias, and R. Nowak, "Majorization–minimization algorithms for wavelet-based image restoration," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2980–2991, 2007.

[36] C. Vonesch and M. Unser, "A fast thresholded Landweber algorithm for wavelet-regularized multidimensional deconvolution," *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 539–549, 2008.

[37] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[38] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 681–695, 2011.

[39] H. Pan and T. Blu, "Sparse image restoration using iterated linear expansion of thresholds," in *Proc. of IEEE International Conference on Image Processing*, Brussels, Belguim, Sep. 2011, pp. 1905–1908.

[40] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 460–489, 2006.

[41] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1, pp. 89–97, 2004.

[42] P. Combettes and J. Pesquet, "Image restoration subject to a total variation constraint," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1213–1222, 2004.

[43] J. Bioucas-Dias, M. Figueiredo, and J. Oliveira, "Total variation-based image deconvolution: a majorization-minimization approach," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 861-864, 2006.

[44] O. Michailovich, “An iterative shrinkage approach to total-variation image restoration,” *IEEE Transactions on Image Processing*, vol. 20, no. 5, pp. 1281–1299, 2011.

[45] R. Neelamani, H. Choi, and R. Baraniuk, “ForWaRD: Fourier-wavelet regularized deconvolution for ill-conditioned systems,” *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 418–433, 2004.

[46] J. Guerrero-Colón, L. Mancera, and J. Portilla, “Image restoration using space-variant Gaussian scale mixtures in overcomplete pyramids,” *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 27–41, 2008.

[47] J. Bioucas-Dias, “Bayesian wavelet-based image deconvolution: a GEM algorithm exploiting a class of heavy-tailed priors,” *IEEE Transactions on Image Processing*, vol. 15, no. 4, pp. 937–951, 2006.

[48] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Inverse halftoning by point-wise shape-adaptive DCT regularized deconvolution,” in *Proc. of International Workshop Spectral Meth. Multirate Signal Processing,* 2006.

[49] ——, “Image restoration by sparse 3-D transform-domain collaborative filtering,” in *Proc. of SPIE Electronic Imaging*, vol. 6812, San Jose, California, USA, Jan. 2008.

[50] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, “Image denoising using scale mixtures of Gaussians in the wavelet domain,” *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, 2003.

[51] A. Foi, V. Katkovnik, and K. Egiazarian, “Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images,” *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1395–1411, 2007.

[52] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[53] A. Jain, *Fundamentals of digital image processing.* Prentice Hall information and system sciences series, 1989.

[54] A. Jalobeanu, L. Blanc-Feraud, and J. Zerubia, “An adaptive Gaussian model for satellite image deblurring,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 613–621, 2004.

[55] F. Jeng and J. Woods, “Compound Gauss-Markov random fields for image estimation,” *IEEE Transactions on Signal Processing*, vol. 39, no. 3, pp. 683–697, 1991.

[56] G. Chantas, N. Galatsanos, A. Likas, and M. Saunders, “Variational Bayesian image restoration based on a product of t-distributions image prior,” *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1795–1805, 2008.

[57] M. Mignotte, “A non-local regularization strategy for image deconvolution,” *Pattern Recognition Letters*, vol. 29, no. 16, pp. 2206–2212, 2008.

[58] L. Mancera, S. Babacan, R. Molina, and A. Katsaggelos, “Image restoration by mixture modelling of an overcomplete linear representation,” in *Proc. of IEEE International Conference on Image Processing,* Cairo, Egypt, Nov. 2009, pp. 3949–3952.

[59] S. Babacan, R. Molina, and A. Katsaggelos, “Sparse Bayesian image restoration,” in *Proc. of IEEE International Conference on Image Processing,* Hong Kong, China, Sep. 2010, pp. 26–29.

[60] A. Mohammad-Djafari, “A full bayesian approach for inverse problems,” in *Maximum Entropy and Bayesian Methods.* Kluwer Academic Publishers, 1996, pp. 135–143.

[61] N. Galatsanos and A. Katsaggelos, “Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation,” *IEEE Transactions on Image Processing*, vol. 1, no. 3, pp. 322–336, 2002.

[62] A. Thompson, J. Brown, J. Kay, and D. Titterington, “A study of methods of choosing the smoothing parameter in image restoration by regularization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 326–339, 1991.

[63] M. Lukas, “Asymptotic optimality of generalized cross-validation for choosing the regularization parameter,” *Numerische Mathematik*, vol. 66, no. 1, pp. 41–66, 1993.

[64] A. Levin, Y. Weiss, F. Durand, and W. Freeman, “Understanding and evaluating blind deconvolution algorithms,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1964–1971.

[65] T. Kenig, Z. Kam, and A. Feuer, “Blind image deconvolution using machine learning for three-dimensional microscopy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2191–2204, 2010.

[66] D. Kundur and D. Hatzinakos, “Blind image deconvolution,” *IEEE Signal Processing Magazine*, vol. 13, no. 3, pp. 43–64, 1996.

[67] A. Carasso, “Linear and nonlinear image deblurring: A documented study,” *SIAM journal on numerical analysis*, pp. 1659–1689, 1999.

[68] T. Bishop, S. Babacan, B. Amizic, A. Katsaggelos, T. Chan, and R. Molina, “Blind image deconvolution: problem formulation and existing approaches,” in *Blind image deconvolution: theory and applications*, P. Campisi and K. Egiazarian, Eds. CRC press, 2007.

[69] G. Ayers and J. Dainty, “Iterative blind deconvolution method and its applications,” *Optics letters*, vol. 13, no. 7, pp. 547–549, 1988.

[70] J. R. Fienup, “Phase retrieval algorithms: a comparison,” *Applied optics*, vol. 21, no. 15, pp. 2758–2769, 1982.

[71] R. G. Lane, “Blind deconvolution of speckle images,” *J. Opt. Soc. Am. A*, vol. 9, no. 9, pp. 1508–1514, 1992.

[72] H. Lanteri, C. Aime, H. Beaumont, and P. Gaucherel, “Blind deconvolution using the richardson-lucy algorithm,” in *Proceedings of SPIE*, vol. 2312, 1994, pp. 182–192.

[73] F. Tsumuraya, N. Miura, and N. Baba, “Iterative blind deconvolution method using Lucy’s algorithm,” *Astronomy and Astrophysics*, vol. 282, pp. 699–708, 1994.

[74] W. Richardson, “Bayesian-based iterative method of image restoration,” *J. Opt. Soc. Am. A*, vol. 62, no. 1, pp. 55–59, 1972.

[75] L. Lucy, “An iterative technique for the rectification of observed distributions,” *Astronomical Journal*, vol. 79, pp. 745–765, 1974.

[76] T. J. Holmes, “Blind deconvolution of quantum-limited incoherent imagery: maximum-likelihood approach,” *J. Opt. Soc. Am. A*, vol. 9, no. 7, pp. 1052–1061, 1992.

[77] E. Thiébaut and J.-M. Conan, “Strict a priori constraints for maximum-likelihood blind deconvolution,” *J. Opt. Soc. Am. A*, vol. 12, no. 3, pp. 485–492, 1995.

[78] T. Zaccheo and R. Gonsalves, “Iterative maximum-likelihood estimators for positively constrained objects,” *J. Opt. Soc. Am. A*, vol. 13, no. 2, pp. 236–242, 1996.

[79] R. Lane, “Methods for maximum-likelihood deconvolution,” *J. Opt. Soc. Am. A*, vol. 13, no. 10, pp. 1992–1998, 1996.

[80] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society. Series B*, pp. 1–38, 1977.

[81] Y. You and M. Kaveh, “A regularization approach to joint blur identification and image restoration,” *IEEE Transactions on Image Processing*, vol. 5, no. 3, pp. 416–428, 1996.

[82] T. Chan and C. Wong, “Total variation blind deconvolution,” *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 370–375, 1998.

[83] H. Liao and M. K. Ng, “Blind deconvolution using generalized cross-validation approach to regularization parameter estimation,” *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 3005–3019, 2011.

[84] Y. Huang and M. Ng, “Lipschitz and total-variational regularization for blind deconvolution,” *Communications in Computational Physics*, vol. 4, pp. 195–206, 2008.

[85] Y. You and M. Kaveh, “Blind image restoration by anisotropic regularization,” *IEEE Transactions on Image Processing*, vol. 8, no. 3, pp. 396–407, 1999.

[86] S. Kullback and R. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[87] A. Likas and N. Galatsanos, “A variational approach for bayesian blind image deconvolution,” *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2222–2233, 2004.

[88] S. Babacan, R. Molina, and A. Katsaggelos, “Variational Bayesian blind deconvolution using a total variation prior,” *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 12–26, 2009.

[89] D. Tzikas, A. Likas, and N. Galatsanos, “Variational bayesian sparse kernel-based blind image deconvolution with student’s-t priors,” *IEEE Transactions on Image Processing*, vol. 18, no. 4, pp. 753–764, 2009.

[90] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman, “Removing camera shake from a single photograph,” in *ACM Transactions on Graphics*, vol. 25, no. 3, 2006, pp. 787–794.

[91] Q. Shan, J. Jia, and A. Agarwala, “High-quality motion deblurring from a single image,” in *ACM Transactions on Graphics*, vol. 27, no. 3, 2008, p. 73.

[92] A. Levin, Y. Weiss, F. Durand, and W. Freeman, “Efficient marginal likelihood optimization in blind deconvolution,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* Colorado Springs, USA, June, 2011, pp. 2657–2664.

[93] M. Almeida and L. Almeida, “Blind and semi-blind deblurring of natural images,” *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 36–52, 2010.

[94] A. Levin, R. Fergus, F. Durand, and W. Freeman, “Image and depth from a conventional camera with a coded aperture,” in *ACM Transactions on Graphics*, vol. 26, no. 3, 2007, p. 70.

[95] R. Molina, J. Núñez, F. Cortijo, and J. Mateos, “Image restoration in astronomy: a Bayesian perspective,” *IEEE Signal Processing Magazine*, vol. 18, no. 2, pp. 11–29, 2001.

[96] H. Liao, F. Li, and M. Ng, “Selection of regularization parameter in total variation image restoration,” *J. Opt. Soc. Am. A*, vol. 26, no. 11, pp. 2311–2320, 2009.

[97] F. Krahmer, Y. Lin, B. McAdoo, K. Ott, J. Wang, D. Widemann, and B. Wohlberg, “Blind image deconvolution: Motion blur estimation,” Technical Report, University of Minnesota, 2006.

[98] J. Oliveira, M. Figueiredo, and J. Bioucas-Dias, “Blind estimation of motion blur parameters for image deconvolution,” *Pattern Recognition and Image Analysis*, pp. 604–611, 2007.

[99] B. Cooke, B. Laubscher, C. Borel, T. Lomheim, and C. Klein, “Methodology for rapid infrared multi-spectral, electro-optical imaging system performance analysis and synthesis,” in *Proc. of SPIE*, vol. 2743, 1996, pp. 52–86.

[100] R. Fabian and D. Malah, “Robust identification of motion and out-of-focus blur parameters from blurred and noisy images,” *Graphical Models and Image Processing*, vol. 53, no. 5, pp. 403–412, 1991.

[101] A. Moffat, “A theoretical investigation of focal stellar images in the photographic emulsion and application to photographic photometry,” *Astronomy and Astrophysics*, vol. 3, pp. 455–461, 1969.

[102] J. Markham and J. Conchello, “Parametric blind deconvolution: a robust method for the simultaneous estimation of image and blur,” *J. Opt. Soc. Am. A*, vol. 16, no. 10, pp. 2377–2391, 1999.

[103] L. Chen and K. Yap, “A soft double regularization approach to parametric blind image deconvolution,” *IEEE Transactions on Image Processing*, vol. 14, no. 5, pp. 624–633, 2005.

[104] P. Sarder and A. Nehorai, “Deconvolution methods for 3-D fluorescence microscopy images,” *IEEE Signal Processing Magazine*, vol. 23, no. 3, pp. 32–45, 2006.

[105] P. Pankajakshan, L. Blanc-Féraud, B. Zhang, Z. Kam, J.-C. Olivo-Marin, and J. Zerubia, “Parametric Blind Deconvolution for Confocal Laser Scanning Microscopy – Proof of Concept,” INRIA, Rapport de recherche RR-6493, April, 2008.

[106] M. Ebrahimi Moghaddam and M. Jamzad, “Motion blur identification in noisy images using mathematical models and statistical measures,” *Pattern recognition*, vol. 40, no. 7, pp. 1946–1957, 2007.

[107] Z. Wang and A. C. Bovik, “Mean squared error: love it or leave it? a new look at signal fidelity measures,” *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009.

[108] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[109] C. Stein, “Estimation of the mean of a multivariate normal distribution,” *The Annals of Statistics*, vol. 9, no. 6, pp. 1135–1151, 1981.

[110] F. Luisier, T. Blu, and M. Unser, “A new SURE approach to image denoising: Interscale orthonormal wavelet thresholding,” *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 593–606, 2007.

[111] S. G. Chang, B. Yu, and M. Vetterli, “Adaptive wavelet thresholding for image denoising and compression,” *Image Processing, IEEE Transactions on*, vol. 9, no. 9, pp. 1532–1546, 2000.

[112] C. Vonesch, S. Ramani, and M. Unser, “Recursive risk estimation for non-linear image deconvolution with a wavelet-domain sparsity constraint,” in *Proc. of IEEE International Conference on Image Processing,* San Diego, California, USA, Oct. 2008, pp. 665–668.

[113] J. Pesquet, A. Benazza-Benyahia, and C. Chaux, “A SURE approach for digital signal/image deconvolution problems,” *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4616–4632, 2009.

[114] R. Giryes, M. Elad, and Y. Eldar, "The projected GSURE for automatic parameter tuning in iterative shrinkage methods," *Applied and Computational Harmonic Analysis*, vol. 30, no. 3, pp. 407–422, 2010.

[115] Y. Eldar, "Generalized SURE for exponential families: Applications to regularization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 471–481, 2009.

[116] D. Van De Ville and M. Kocher, "SURE-based non-local means," *IEEE Signal Processing Letters*, vol. 16, no. 11, pp. 973–976, 2009.

[117] M. Raphan and E. Simoncelli, "Optimal denoising in redundant representations," *IEEE Transactions on Image Processing*, vol. 17, no. 8, pp. 1342–1352, 2008.

[118] T. Blu, "The SURE-LET methodology — A prior-free approach to signal and image denoising," in *Plenary Presentation at the Eighth International Workshop on Sampling Theory and Applications (SampTA'09),* Marseille, May 18-22, 2009.

[119] G. Strang, "The discrete cosine transform," *SIAM Review*, vol. 41, no. 1, pp. 135–147, 1999.

[120] G. Strang and T. Nguyen, *Wavelets and filter banks.* Wellesley Cambridge Press, 1996.

[121] A. Oppenheim, A. Willsky, and S. Nawab, *Signals and systems.* Prentice-Hall Englewood Cliffs, NJ, 1983, vol. 2.

[122] T. Ha, *Theory and design of digital communication systems.* Cambridge University Press, 2010.

[123] S. Martucci, "Symmetric convolution and the discrete sine and cosine transforms," *IEEE Transactions on Signal Processing*, vol. 42, no. 5, pp. 1038–1051, 1994.

[124] J. Jia, "Single image motion deblurring using transparency," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* Minneapolis, Minnesota, USA, June, 2007, pp. 1–8.

[125] B. Amizic, S. Babacan, R. Molina, and A. Katsaggelos, "Sparse bayesian blind image deconvolution with parameter estimation," in *Proc. of European Conference on Signal Processing*, 2010, pp. 626–630.

[126] B. Amizic, R. Molina, and A. K. Katsaggelos, "Sparse bayesian blind image deconvolution with parameter estimation," *EURASIP Journal of Image and Video Processing*, vol. 2012, no. 20, 2012.

[127] L. Yuan, J. Sun, L. Quan, and H. Shum, "Image deblurring with blurred/noisy image pairs," in *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, 2007, p. 1.

[128] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* San Francisco, USA, June, 2010, pp. 491–498.

[129] S. Cho and S. Lee, “Fast motion deblurring,” in *ACM Transactions on Graphics (TOG)*, vol. 28, no. 5, 2009, p. 145.

[130] N. Joshi, R. Szeliski, and D. Kriegman, “PSF estimation using sharp edge prediction,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* Anchorage, Alaska, USA, June, 2008, pp. 1–8.

[131] D. Krishnan and R. Fergus, “Fast image deconvolution using hyper-laplacian priors,” *Advances in Neural Information Processing Systems*, vol. 22, pp. 1–9, 2009.

[132] O. Michailovich and A. Tannenbaum, “Blind deconvolution of medical ultrasound images: a parametric inverse filtering approach,” *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 3005–3019, 2007.

[133] R. W. Cole, T. Jinadasa, and C. M. Brown, “Measuring and interpreting point spread functions to determine confocal microscope resolution and ensure quality control,” *Nature protocols*, vol. 6, no. 12, pp. 1929–1941, 2011.

[134] S. Frisken Gibson and F. Lanni, “Experimental test of an analytical model of aberration in an oil-immersion objective lens used in three-dimensional light microscopy,” *J. Opt. Soc. Am. A*, vol. 8, no. 10, pp. 1601–1613, 1991.

[135] A. Santos and I. Young, “Model-based resolution: applying the theory in quantitative microscopy,” *Applied Optics*, vol. 39, no. 17, pp. 2948–2958, 2000.

[136] B. Zhang, J. Zerubia, and J. Olivo-Marin, “Gaussian approximations of fluorescence microscope point-spread function models,” *Applied Optics*, vol. 46, no. 10, pp. 1819–1829, 2007.

[137] M. Shimizu, S. Yoshimura, M. Tanaka, and M. Okutomi, “Super-resolution from image sequence under influence of hot-air optical turbulence,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* Anchorage, Alaska, USA, June, 2008, pp. 1–8.

[138] J. Elder and S. Zucker, “Local scale control for edge detection and blur estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, pp. 699–716, 1998.

[139] P. Favaro and S. Soatto, “A geometric approach to shape from defocus,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 406–417, 2005.

[140] S. Bae and F. Durand, “Defocus magnification,” in *Computer Graphics Forum*, vol. 26, no. 3, 2007, pp. 571–579.

[141] W. Zhang and W. Cham, “Single image refocusing and defocusing,” *IEEE Transactions on Image Processing*, no. 99, pp. 1–1, 2012.

[142] P. van Beek, “Edge-based image representation and coding,” Ph.D. Thesis, *Delft University of Technology*, 1995.

[143] P. Pankajakshan, B. Zhang, L. Blanc-Féraud, Z. Kam, J. Olivo-Marin, and J. Zerubia, "Blind deconvolution for thin-layered confocal imaging," *Applied optics*, vol. 48, no. 22, pp. 4437–4448, 2009.

[144] Q. Cao, "Generalized jinc functions and their application to focusing and diffraction of circular apertures," *J. Opt. Soc. Am. A*, vol. 20, no. 4, pp. 661–667, 2003.

[145] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. of European Conference on Computer Vision,* Crete, Greece, Sep, 2010, pp. 157–170.

[146] W. Freeman, E. Adelson, M. I. of Technology. Media Laboratory. Vision, and M. Group, "The design and use of steerable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[147] J. Biemond, R. Lagendijk, and R. Mersereau, "Iterative methods for image deblurring," *Proceedings of the IEEE*, vol. 78, no. 5, pp. 856–883, 1990.

[148] H. Ji and C. Liu, "Motion blur identification from image gradients," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* Anchorage, Alaska, USA, June, 2008 , pp. 1–8.

[149] H. Sun, M. Desvignes, and Y. Yunhui, "Motion blur adaptive identification from natural image model," in *Proc. of IEEE International Conference on Image Processing,* Cairo, Egypt, Nov. 2009, pp. 137–140.

[150] H. Sun, M. Desvignes, Y. Yan, and W. Liu, "Motion blur parameters identification from radon transform image gradients," in *IEEE Annual Conference on Industrial Electronics*, 2009, pp. 2098–2103.

[151] S. Dai and Y. Wu, "Motion from blur," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* Anchorage, Alaska, USA, June, 2008, pp. 1–8.

[152] S. Reeves and R. Mersereau, "Blur identification by the method of generalized cross-validation," *IEEE Transactions on Image Processing*, vol. 1, no. 3, pp. 301–311, 1992.

[153] Y. Yitzhaky, I. Mor, A. Lantzman, and N. Kopeika, "Direct method for restoration of motion-blurred images," *J. Opt. Soc. Am. A*, vol. 15, no. 6, pp. 1512–1519, 1998.

[154] A. Levin, "Blind motion deblurring using image statistics," *Advances in Neural Information Processing Systems*, vol. 19, p. 841, 2007.

[155] T. Cho, S. Paris, B. Horn, and W. Freeman, "Blur kernel estimation using the radon transform," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition,* Colorado Springs, USA, June, 2011, pp. 241–248.

[156] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling, "Recording and playback of camera shake: benchmarking blind deconvolution with a real-world database," in *ECCV*. Springer, 2012, pp. 27–40.

[157] J. W. Miskin, "Ensemble learning for independent component analysis," in *in Advances in Independent Component Analysis.* Citeseer, 2000.

[158] D. J. Field, "What is the goal of sensory coding?" *Neural computation*, vol. 6, no. 4, pp. 559–601, 1994.

[159] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Scholkopf, "Fast removal of non-uniform camera shake," in *2011 IEEE International Conference on Computer Vision*, pp. 463–470.

[160] H. Zhang, J. Yang, Y. Zhang, N. M. Nasrabadi, and T. S. Huang, "Close the loop: Joint blind image restoration and recognition with sparse representation prior," in *Proc. of IEEE International Conference on Computer Vision,* Barcelona, Spain, Nov. 2011, pp. 770–777.