

DEBLURRING TEXT IMAGES AFFECTED BY MULTIPLE KERNELS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

By
Tolga Dizdärer
May 2018

Deblurring Text Images Affected by Multiple Kernels

By Tolga Dizdärer

May 2018

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Mustafa elebi Pınar(Advisor)

Gözde Bozdağı Akar

Selim Aksoy

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

DEBLURRING TEXT IMAGES AFFECTED BY MULTIPLE KERNELS

Tolga Dizdärer

M.S. in Industrial Engineering

Advisor: Mustafa Çelebi Pınar

May 2018

Image deblurring is one of the widely studied and challenging problems in image recovery. It is an estimation problem dealing with restoration of a linearly transformed image that is additionally disturbed with noise. In our research, we propose a new method to solve deblurring problems on text images affected by multiple kernels. In our approach we focus specifically on almost binary images that have specific intensity structures. First, we propose a non-convex non-blind deblurring model and provide an efficient algorithm that can restore a text-like image when the blurring kernel is known. Then we provide our alternate setting, the semi-blind problem, where a kernel is determined as a linear combination of multiple kernels. We propose how one can attack the deblurring problem by using dictionaries that are constructed using any prior information about the kernel. We propose a semi-blind deblurring model that can estimate optimal kernel using the elements of the dictionary. We consider a unique algorithm structure that favors regularizing the iterations through scaled parameter values and argue the advantages of this approach. Lastly, we consider some specific problems that are commonly used in the literature where one can utilize our alternate problem setting. We argue how one can construct a dictionary that can maximize the utility gained by the prior information regarding the blurring process and present the performance of our model in such cases.

Keywords: Deblurring, image restoration, inverse problems, non-convex optimization.

ÖZET

BİRDEN ÇOK BULANIKLAŞMA UNSURUNDAN ETKİLENMİŞ METİN GÖRÜNTÜLERİNİN NETLEŞTİRİLMESİ

Tolga Dizdärer

Endüstri Mühendisliđi, Yüksek Lisans

Tez Danışmanı: Mustafa Çelebi Pınar

Mayıs 2018

Bulanık görüntülerin netleştirilmesi literatürde en geniş çaplı çalışılan araştırma alanları arasında yer almakta olup son yıllarda bu alanda bir çok yeni yaklaşım sunulmuştur. Görüntü netleştirme, doğrusal deđişen ve rastgele gürültüye maruz kalmış bir resim vektörünün tahminlenmesi üzerine çalışmaktadır. Araştırmamızda, birden fazla bulanıklaşma unsurundan etkilenmiş metin resimleri için bulanıklık giderme modeli sunmaktayız. Yaklaşımımızda öncelikle yaklaşık olarak ikitonlu olan resim gruplarına odaklanmaktayız. İlk olarak, bulanıklaşma unsurunun bilindiđi duruma yaklaşım olarak konveks olmayan bir model geliştireyoruz ve bu modelin çözümüne yönelik metin resimlerini tahmin edecek bir algoritma sunuyoruz. Ardından bulanıklaşma unsuru hakkında kısmi bir bilginin var olduđu durumları düşünerek bulanıklaşmayı birden fazla unsurun doğrusal kombinasyonunun oluşturduđu bir yaklaşımı varsayıyoruz. Bu yöntemde netleştirme operasyonu için gerekli unsunları kapsayacak ve doğru resmi bulmayı sağlayacak bir sözlük oluşturma işlemine yaklaşımı gösteriyoruz. Bu yeni yaklaşım doğrultusunda kısmi bilgi altında doğru bulanıklaşma unsurunu bulabilecek bir model sunuyoruz. Algoritmalarımızda özgün bir yaklaşım olarak kendi içerisinde düzenleme sağlayacak bir yapıyı ortaya koyuyoruz. Son olarak, literatürde çoklukla kaşılaşılan ve yaklaşımımızın fayda sağlayacağı belirli problemleri inceliyoruz. Bu örnekler üzerinden bulanıklaşma üzerine kısmi bilginin var olduđu durumlarda yaklaşımımızın fayda sağlayacağı bir sözlüğün nasıl yaratılacağını gösterip modelimizin bu koşullar altındaki performansını sunuyoruz.

Anahtar sözcükler: Görüntü netleştirme, resim restorasyonu, ters problemler, konveks olmayan optimizasyon.

Acknowledgement

Firstly, I would like to express my sincere gratitude to my advisor Prof. Mustafa Çelebi Pınar for his support and his belief in me throughout my graduate studies at Bilkent. It has been a great pleasure to work under his guidance.

I also would like to thank Prof. Gözde Bozdağı Akar and Prof. Selim Aksoy for accepting to read and review my thesis and for providing invaluable comments and suggestions.

I would like to thank all of the professors and staff in the Department of Industrial Engineering for their help and support. Along with them, I would like to extend my thanks to my fellow colleagues and friends for their friendship through hard times.

I would like to thank my family for their endless love and support. It is most wonderful to have their unwavering support.

Finally, I would like to thank Zeynep Tekin, whom I am indebted to for her endless care and encouragement.

Contents

1	Introduction	1
2	Problem Definition	4
2.1	Problem Setting	7
2.2	Our Motivation	8
3	Literature Review	11
4	Non-Blind Deblurring Model for Almost Binary Images	17
4.1	The Non-Blind Model	19
4.2	Solution Methodology	20
5	Semi-Blind Deblurring Model for Almost Binary Images	25
5.1	The Semi-Blind Model	26
5.2	Extended Model for Large Dictionaries	27
6	Computational Study	31

6.1	Model Properties	32
6.2	Performance Metrics	34
6.3	Performance Analysis	36
6.4	Sensitivity Analysis	39
7	Specially Structured Blur Kernels	45
7.1	Motion Blurring Problem	46
7.2	Out-of-Focus Blurring Problem	48
7.3	Gaussian Blurring Problem	50
8	Conclusion	53
A	Variables and Derivative Operators	60
B	Testing Images and Kernels	62
C	Additional Results	66
D	Effects of Regularization Terms	70

List of Figures

2.1	Example naive solution	5
4.1	Example histogram for an image that is almost binary	18
6.1	(a)Our method. (b)Lucy-Richardson Algorithm [1, 2]. (c)Total Variation [3]. (d) l_0 regularized model [4]. (e) Hyper-laplacian priors [5]. (f) [6].	36
6.2	(a)Our method. (b)Lucy-Richardson Algorithm [1, 2]. (c)Total Variation [3]. (d) l_0 regularized model [4]. (e) Hyper-laplacian priors [5]. (f) [6].	37
6.9	PSNR value of the restored image through iterations.	38
6.10	Total l_2 change in the estimated when we scale parameters	40
6.3	(a)Our method. (b)Lucy-Richardson Algorithm [1, 2]. (c)Total Variation [3]. (d) l_0 regularized model [4]. (e) Hyper-laplacian priors [5]. (f) [6].	41

6.4	(a)Images recovered using our proposed method. (b)Image recovered using Lucy-Richardson Algorithm [1, 2]. (c)Image recovered using Total Variation Method [3]. (d)Image recovered using the l0 regularized model [4]. (e) Image recovered using hyper-laplacian priors using model [5]. (f)Image recovered by the method in [6].	42
6.5	(a)True Image. (a)Observed Image. (a)Restored Image.	43
6.6	(a)True Image. (a)Observed Image. (a)Restored Image.	43
6.7	(a)True Image. (a)Observed Image. (a)Restored Image.	43
6.8	Image recovery by our model using a large dictionary. Top to bottom: true image and kernel, two intermediate solutions, the final solution.	44
7.1	Sample dictionary to solve semi-blind motion deblurring problem	47
7.2	Images affected by semi-blind motion blurring kernels recovered using a specially designed dictionary. (a)True Image. (b)Observed Image. (c)Restored Image.	48
7.3	Sample dictionary to solve semi-blind out-of-focus deblurring problem	49
7.4	Images affected by semi-blind out-of-focus blurring kernels recovered using a specially designed dictionary. (a)True Image. (b)Observed Image. (c)Restored Image.	50
7.5	Images affected by semi-blind gaussian blurring kernels recovered using a specially designed dictionary. (a)True Image. (b)True Kernel. (c)Observed Image. (d)Restored Image. (e)Restored Kernel.	51
B.1	True images, observed images and blurring kernels for cartoon image	62

B.2	True images, observed images and blurring kernels for vector image	63
B.3	True images, observed images and blurring kernels for handwriting image	64
B.4	True images, observed images and blurring kernels for cube image	65
C.1	(a)Our method. (b)Lucy-Richardson Algorithm [1, 2]. (c)Total Variation [3]. (d)l0 regularized model [4]. (e) Hyper-laplacian priors [5]. (f) [6].	66
C.2	(a)Our method. (b)Lucy-Richardson Algorithm [1, 2]. (c)Total Variation [3]. (d)l0 regularized model [4]. (e) Hyper-laplacian priors [5]. (f) [6].	67
C.3	(a)Our method. (b)Lucy-Richardson Algorithm [1, 2]. (c)Total Variation [3]. (d)l0 regularized model [4]. (e) Hyper-laplacian priors [5]. (f) [6].	68
C.4	(a)Our method. (b)Lucy-Richardson Algorithm [1, 2]. (c)Total Variation [3]. (d)l0 regularized model [4]. (e) Hyper-laplacian priors [5]. (f) [6].	69
D.1	Estimated solution (a)Without ϕ (b)With ϕ	70
D.2	Estimated solution (a)Without ν (b)With ν	71
D.3	Estimated solution (a)Without μ (b)With μ	71
D.4	Estimated solution (a)Without γ (b)With γ	72
D.5	Estimated solution (a)Without ζ (b)With ζ	72
D.6	Estimated solution (a)Without η (b)With η	73

List of Tables

6.1	The comparison of average computation time for a single kernel identification step for selected NNLS algorithms	31
6.2	The comparison of PSNR values for estimated images	36
6.3	The comparison of ISNR values for estimated images	37
6.4	The comparison of SSIM values for estimated images	37
A.1	Definition of Variables	60
A.2	Formulations for Derivative Operators	61

Chapter 1

Introduction

Image recovery deals with restoration of images that cannot be observed without some form of perturbation. If the image is corrupted by random noise, the recovery process is addressed as a denoising problem. In denoising problems the noise usually follows a specific distribution and one needs to estimate a clean solution by utilizing this knowledge. If the image has gone through a convolution operation with a blurring matrix, which is equivalent to the linear transformation of the image vector, the setting becomes a deblurring problem. Denoising and deblurring problems are generally very interconnected as one needs to consider the additive noise in deblurring problems.

The deblurring problem is a widely studied area, drawing attention from multiple disciplines. Whereas from an electrical engineering perspective one may focus on the restoration of an observed signal which values precision, computer engineering and operations research practitioners generally focus more attention on the qualitative performance of the deblurring approach valuing the favorable perception of estimated image. In this thesis, we tackle a deblurring problem where we restore a digital image that is both linearly transformed by a blurring kernel and disturbed by random gaussian noise and propose an algorithm that provides very effective solutions both quantitatively and qualitatively.

First, we introduce the blurring operation. Then, we lay out the problem setting in chapter 2 and describe the relevant Literature in chapter 3. In chapter 4, we present our non-blind deblurring problem for text images. After proposing an efficient algorithm for the non-blind model, we move to the model proposed for semi-blind setting in chapter 5. Lastly, we present our findings in chapter 6 and describe some special settings where our approach is most effective at attacking the semi-blind problem in chapter 7.

For simplicity, assume that we are working on grayscale images. An image is a collection of pixels with varying intensity levels. The color of each pixel depends on the value of the intensity where the white pixels have the highest intensity value of 255, and the black pixels have the lowest intensity of 0. All remaining shades are represented by intensity values ranging between 0 and 255. Assuming that the image has a size of $m \times n$ pixels, the image can be represented with a matrix $x \in \mathbb{R}^{m \times n}$ with each element $x_{i,j} \in [0, 255]$.

Blur occurs due to the presence of a perturbation in the imaging process. In our approach, we focus on shift-invariant blur where every pixel in the image is affected by the same blurring kernel, $h \in \mathbb{R}^{s \times t}$. Mathematically, this operation can be represented as discrete convolution of an image with a blurring matrix. We denote the convolution operator with \otimes and compute the blurred image, y as:

$$y = x \otimes h$$

$$\implies y(k_1, k_2) = \sum_{l_1} \sum_{l_2} x(l_1, l_2) h(k_1 - l_1, k_2 - l_2). \quad (1.1)$$

Note that the convolution operation attains the intensity of a pixel using linear combination of all pixels with weights assigned by their corresponding blurring kernel elements. However, the computation for the pixels near to edges may refer to kernel elements that do not correspond to any pixel of the original image. To compute the blurred intensities for these pixels, one needs to define a boundary condition to handle the outliers. In the Literature there are different boundary assumptions to overcome this problem such as flipping the image in each direction or repeating the same image in every direction. Beyond handling outliers,

these assumptions determine the fast methods one uses to compute the blurring operation. In our research, we assume Zero Boundary Condition which assumes that the pixels beyond the boundaries have intensities of 0.

While there are various blurring kernels representing different blurring operations, there are some kernel types commonly encountered in the Literature. Some of them that we deal with in our research are motion blur, out-of-focus blur and gaussian blur. These kernels are used to represent unique blurring operations observed in practice. For example, motion blur kernel is used when the objects of interest is not stationary and the gaussian blur represents the blurring caused by statistical light scatter and sampling by receptive fields with Gaussian profiles [7]. In practical settings, a single type of kernel is usually not sufficient to represent the blurring operation as more than one type of kernel affects the image, so one also needs to consider combinations of multiple kernels.

Chapter 2

Problem Definition

In a deblurring problem, the observed image, y , is represented by:

$$y = x \circledast h + \epsilon, \tag{2.1}$$

where x is the true image, h is the blurring kernel and ϵ is unknown (typically Gaussian) noise. As this operation is linear, it is alternately represented using the linear formulation:

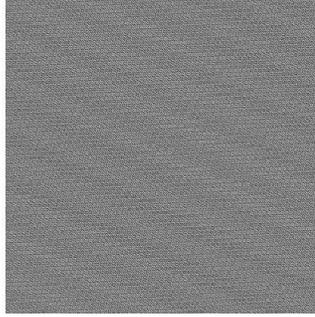
$$y = Hx + \epsilon, \tag{2.2}$$

where H is the large blurring matrix. The presence of this representation is important, as it allows one to have a convex data fidelity term in a mathematical model.

The objective of a deblurring problem is to find the true image x using a well-structured approach. Due to this practice, most image deblurring models are a form of inverse problem trying to minimize the distance between y and Hx . Despite the simple problem definition, the additive random noise leads one to require sophisticated techniques to be able to find a good solution. In fact, in many settings recovering the exact original image is deemed as impossible. One of the simplest approaches to this problem is to consider an image, x , that satisfies:

$$y = Hx. \tag{2.3}$$

If one pursues such solution, the resulting image generally becomes similar to fig. 2.1. In the Literature this approach, which completely disregards the noise, is called a "naive solution".



(a)

Figure 2.1: Example naive solution

In order to understand why this approach fails, one needs to consider how the solution is formulated. When one disregards the distance between the blurred and observed images, one simply reverses the blurring process. This action is generally represented using inverse filters. Assuming that the blurring matrix, H , is positive definite, we can represent this operation using the inverse of H as follows:

$$x^{naive} = H^{-1}y = H^{-1}(Hx + \epsilon) = x + H^{-1}\epsilon. \quad (2.4)$$

The second term, $H^{-1}\epsilon$, is called the "inverted noise" and it is the reason one finds a noisy naive solution. One can use singular value decomposition to further analyze the effect of the inverted noise. It is known that the SVD of the blurring matrix, H , is unique and can be defined as:

$$\begin{aligned}
 H = U\Sigma V^T &= \begin{bmatrix} u_1 & \dots & u_N \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_N \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_N^T \end{bmatrix} \\
 &= \sum_{i=1}^N \sigma_i u_i v_i^T,
 \end{aligned} \quad (2.5)$$

where $N = mn$.

Then the inverse of the large blurring matrix is computed as:

$$H^{-1} = \sum_{i=1}^N \frac{1}{\sigma_i} u_i v_i^T, \quad (2.6)$$

and the inverted noise is:

$$H^{-1} \epsilon = \sum_{i=1}^N \frac{u_i^T \epsilon}{\sigma_i} v_i. \quad (2.7)$$

This term is the noise added to the original image when pursuing the naive solution. The noise level in the resulting image depends on positive $\sigma_1, \dots, \sigma_N$ values, which can be very small. Especially since the elements of diagonal matrix, Σ , is non-negative and non-increasing, when the condition number of the blurring kernel, $\frac{\sigma_1}{\sigma_N}$, is very large, σ_i can become very small at indexes close to N . Hence much of the high-frequency information of the image become unobservable due to error in the naive solution. This makes the problem very hard to solve using simple arithmetics.

When faced with such problem a widely accepted approach is to use regularization terms that help one find solutions that intuitively have the same characteristics as the expected solution. These terms are used alongside a function that minimizes the distance between y and Hx . The regularization terms bring two advantages to the model. First, these terms restrict the types of images the model may return depending on the type of regularization term. For example one can use terms that exploit the sparsity of edges to find natural images, or make use of the repetition of textures to find image with such structures. Second advantage of this approach is that using additional terms in the objective function pushes the solution away from naive solution by synthetically creating a gap between estimated and observed images. This can limit the effect of the inverted noise and help the resulting image imitate the true blurring process. The biggest disadvantage of regularization terms is that most of the effective regularization terms are very complex and they are hard or impossible to represent using convex functions. Hence, most of the state-of-the-art methods in the Literature make use of non-convex terms and try to take advantage of the unique properties of their formulations to come up with an iterative algorithm. In this respect, many

methods do not enforce optimality conditions on the model, but rather focus on finding a good solution.

2.1 Problem Setting

In general, the Literature dealing with image deblurring is divided into two interconnecting settings. The first type deals with the non-blind problem, where the corrupted image is restored using a known blurring kernel. When the blurring kernel, h is known, the remaining problem is to determine the true image x without knowing the noise, ϵ . Although this problem may appear similar to a denoising problem, which happens when H is the identity matrix, the blurring problems are generally much harder. This is because unlike a noised image, a blurred image may share very little similarities with the the original image. The information present on the image generally hides from plain observation due to blurring operation. This pushes one to guess regularization terms that should apply to a variety of different image types. As the regularization terms are based on ideas that are too broad to represent every type of image, it is almost necessary to restrict the problem setting to come up with an effective method. Furthermore, as many problems require multiple regularization terms, one needs to narrow the problem even more in order not to require precise balancing between different terms.

The second type of deblurring problem is the blind case, which does not assume any knowledge of the blurring kernel and tries to estimate the blurring kernel and/or the true image. As the only information one has in this setting is the observed image, one needs to consider an even larger search space accommodating both potential kernels and the potential images. The most common approach for solving this problem is finding the image and kernel alternatively, however this also leads some complications. Since one needs to estimate interconnecting variables, the problem is very ill-defined. This directs one to solve the problem iteratively, usually requiring many iterations and not guaranteeing any

convergence. Although there are many fast approaches proposed in the Literature, as many images contain upwards of a million variables, it is very hard to create a very complex algorithm with tractable solution time. Hence, most of the fast algorithms rely on algorithms that can be processed with simple operations and take large iteration steps, trading the efficiency off for speed. Due to this, blind problems generally lead to inferior solutions compared to a non-blind one.

2.2 Our Motivation

Limitations regarding to both settings has led us to utilize a hybrid setting to represent the deblurring problem, which we believe is more useful for practical applications and more effective at preserving the performance advantages of the non-blind problem. Instead of restricting the problem to the definitions of blind or non-blind problem, we assume a limited knowledge regarding the blurring kernel. In our model, we assume the knowledge of a dictionary of kernels, D , which can consist of different types of blurring kernels $h_i \in D$ where $i = 1, \dots, |D| = n$. We assume that the true kernel has a mixed distribution, which is constructed using linear combinations of kernels in D with respective weights w_i where $i = 1, \dots, n$. Lee et al. [8] use this idea to estimate blur kernels. They assume that the blur kernel is separable and basic patterns of the dictionary are constructed from Gaussian functions and they provide a method to compute some blur kernels with specific structures. Our approach differs from theirs due to how we construct the dictionary. Rather than assuming a mathematical property for the kernel, we provide an effective way to construct a dictionary for different kinds of kernels that only uses prior information that can be inferred from the blurring settings. That allows us to provide methods for a range of kernel types such as motion blur and out-of-focus blur, and allows us to solve the kernel estimation problem using dictionaries of minimum size. In this sense, our methods relies heavily on constructing an efficient dictionary. To the best of our knowledge, ours is the first approach to focus on how to construct a minimal dictionary to estimate a problem specific kernel.

As the model uses a limited prior knowledge, our problem lies at the intersection of blind and non-blind deblurring problems, which may be called a semi-blind deblurring problem. This assumption does not necessarily eliminate all problems associated with blind problems. The setting still requires one work with a non-convex model and requires one to determine the image and kernel alternatively. However, it allows one to simplify the problems both in size and complexity, leading to major potential performance improvements in kernel identification. Moreover, we find that the new setting allows one to find kernels more effectively in many settings.

Beside our performance improvements, we find that our approach is in compliance with the theory surrounding the image restoration problem. It is known from practical applications that in many deblurring problems, the estimated kernel is a combination of different types of generally known blurring kernels. One can note that this assumption restricts the applicability of our methods compared to a blind problem. Especially when the kernel is very complex in shape and is unpredictable, it may not be most effective to design a dictionary that contains all of the elements required for the true kernel. However, we find that in many applications, such problems appear to be negligible, and the model provides good solutions. Furthermore, we find that due to favorable structure of many kernels, one can exploit the distribution of the kernels, allowing one to estimate certain complex blurring kernels using very simple steps. We also aim to identify the conditions under which our assumptions lead to acceptable solutions.

Using these assumptions in its basis, our method is directed towards deblurring text images, with a special focus on restoring almost binary images. Besides providing a novel formulation for text deblurring, we make another contribution to the Literature by using the idea of regularization beyond modeling since we incorporate it into our algorithm design. Currently, most Literature aims to provide a single formulation, upon which the model will improve the solution towards that direction in an iterative scheme. However, we believe that a better approach can be achieved when one utilizes different formulations by scaling their effects at each iteration. We find that using such an approach not only increases the solution quality substantially, but allows one to lift the strict requirement

for a termination condition by allowing the model to regularize itself through iterations.

Chapter 3

Literature Review

Various methods have been proposed to solve the image deblurring problem. Especially, with the recent advances in computing power and the deblurring problems finding applications in many recognition tasks, one can observe an increasing rate of interest towards this problem in the last few decades. Many break-through advances has been made on this problem since the first applications on the blind deblurring such as [9, 10]. Due to its complexity, many models proposed to this date revolve around these break-through ideas proposed over the decades. Despite this similarity, many different approach have been used to tackle this problem. Deblurring methods generally exploit specific characteristics of a blurring kernel or the desired solution to estimate the true image. One approach towards the deblurring problem focuses on spectral properties of the blurring kernel, trying to minimize the loss of information included by random noise. Another approach focuses on the probabilistic structure of the image, trying to find an image that maximizes the probability of observing the noisy image using problem-specific priors. Building on these ideas, another type of Literature focuses on variational methods. These methods focus on applying statistically sensible regularization terms to estimate the true image. This approach generally focuses on modeling the blurring process and finding a solution that satisfies required properties of the deconvolved image.

Apart from the methodology, the Literature is generally divided according to problem settings. These may consist of the assumptions regarding the distribution of noise or the set of kernels one considers. Furthermore, many papers specialize towards some specific problem settings. These may consist of natural images, text images, random signals or applications regarding specific optical observations. As we consider the deblurring of text images as our primary goal, we also focus on the Literature that deal with this setting.

One of the most novel methods to approach deblurring problems is the Wiener Filtering [11]. This method restores the image using Wiener filter, which is designed to minimize the mean squared error between the estimated image and desired image using a linear time-invariant filtering. It provides a solution that balances between inverse filtering and noise elimination by applying a new filter g to the true image.

It computes the frequency domain of g as:

$$G(k_1, k_2) = \frac{H(k_1, k_2)W(k_1, k_2)}{|H(k_1, k_2)|^2W(k_1, k_2) + \varepsilon(k_1, k_2)} \quad (3.1)$$

where $G(k_1, k_2)$ is the Fourier transform of g , $H(k_1, k_2)$ is the Fourier transform of the blurring kernel, h , $W(k_1, k_2)$ is the mean power spectral density of the true image, x , and $\varepsilon(k_1, k_2)$ is the mean power spectral density of the noise, ϵ . Then it estimates the true image as:

$$X(k_1, k_2) = G(k_1, k_2) \otimes Y(k_1, k_2) \quad (3.2)$$

where $X(k_1, k_2)$ is the Fourier transform of estimated image, \hat{x} , and $Y(k_1, k_2)$ is the Fourier transform of the observed image. There are some literature that extends this idea. Zheng [12] propose to replace the FFT transform by Hartley transform to reduce computational time and memory cost. Tekalp et al. [13] define an extended Wiener set based on the Wiener solution in Fourier domain and propose to solve the image restoration problem using projections onto convex sets.

Another main approach used to tackle the image deblurring problem is the spectral filtering methods. These techniques are used to take advantage of the

information present in an image based on its wavelength. Some of the common methods using this idea are Truncated SVD (TSVD) method or Tikhonov's method [14]. These methods utilize SVD decomposition of the naive solution, which is equivalent to:

$$x = H^{-1}y = \sum_{i=1}^N \frac{u_i^T y}{\sigma_i} v_i. \quad (3.3)$$

As the resulting image is affected by the noise resulting from low σ_i values present at high indexes, they propose a scaling term ξ_i , that can restore the image by eliminating the loss of information caused by high error terms. Their alternate formulation is:

$$x = \sum_{i=1}^N \xi_i \frac{u_i^T y}{\sigma_i} v_i. \quad (3.4)$$

Using this idea, TSVD method puts a hard threshold on the image elements so that

$$\xi_i = \begin{cases} 0 & , i \in 1, \dots, v \\ 1 & , i \in v + 1, \dots, N \end{cases} \quad (3.5)$$

On the other hand, Tikhonov method scales the element values using the term:

$$\xi_i = \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2}, \quad (3.6)$$

where α is a constant parameter.

Another approach for the deblurring problem is Bayesian Inference Framework, which determines the probability distribution of observing the blurred image under the given information. This method is used to compute the optimal image and/or kernels under specific distribution assumptions regarding to image distributions. In Bayesian Inference Framework, one estimates the image using Maximum a posteriori (MAP) estimation, which attempts to find the image and blurring kernel that has the highest probability of being observed according to the following formula:

$$p(x, h|y) = \frac{p(y|x, h)p(x)p(h)}{p(y)} \quad (3.7)$$

The Richardson-Lucy method [1, 2] is one of the novel methods proposed in this sense which assumes a uniform prior distribution, $p(x)$. Shepp and Vardi [15] extend this idea, utilizing Poisson distributed noise assumption. Temerinac-Ott et al [16] combine a regularized version of Richardson-Lucy method with TV constraints and provide a method that can deal with spatially variant kernels. They handle spatial variance by splitting the image into several blocks that are individually treated using spatially invariant kernels. They also propose a method to optimize parameter settings. Keuper et al [17] propose a blind deconvolution model by proposing additional restrictions on the blurring kernel. They observe that the Optical Transform Function of the kernel, which is equivalent to its Fourier Transform, is smooth and impose additional TV constraint on this term. Krishnan et al. [5] point out that using priors that enforce heavy-tailed gradient distribution has been very effective at many image restoration problem. They propose a hyper-Laplacian prior and show that a very fast analytical solution can be found for specific exponents.

Various variational methods have also been proposed to attack the image deblurring problem. The methods generally rely on regularization terms to overcome the ill-posed nature of the problem. One of the most commonly used term is the Tikhonov-Miller prior [18], which leads to solutions with small l_2 norms. This term is used beyond image deblurring finding applications in statistics and other inverse problems. Although it is used separately as a variational term, it is equivalent to Tikhonov method used in spectral filtering methods. In fact, Tikhonov spectral filter with weight α yields the solution to the problem:

$$\min_x \|y - Hx\|^2 + \alpha^2 \|x\|^2. \quad (3.8)$$

Another very commonly used prior is the edge emphasizing Total Variation term [3], which is formulated as:

$$TV_x = \|\sqrt{|\nabla_v x|^2 + |\nabla_h x|^2}\|_1. \quad (3.9)$$

This term minimizes the l_1 distance of image gradients, providing solutions with sparse edges. As sparsity of the edges is ideally represented using l_0 minimization, many methods propose alternate ideas to penalize the edge sparsity. Xu

et al [19] propose an unnatural l_0 regularization term, a piecewise function that behaves similar to l_0 norm. They find their function to be closer to l_0 function compared to previously proposed methods. Krishnan et al. [20] note the computational complexity of estimating l_0 prior and proposes a normalized l_1 prior. The prior scales the l_1 length of the edges with l_2 norm. Due to this term being scale invariant, it allows them to restore image without destroying the magnitude of the gradients. Lee et al. [21] note that there are not many commonly used priors for blurring kernels. They propose to restrict the blurring kernel to a specific dictionaries using basic blur kernels. They find that their approach is more effective at restoring the blurring kernel.

As the deblurring algorithms are very problem specific, there are methods in the Literature that deals specifically with text images. Cho et al. [22] notice that many priors used for natural images are too weak to constraint text images. They identify three main characteristic for text images and propose a prior that can satisfy such requirements. Pan et al [4] use priors that enforce sparsity of the edges with another term that enforces sparsity of pixel intensities. As the number of white pixels are generally dominant in text images, they find this approach to be very effective in restoring text images. Cao et al. [23] propose a method that restores text images in natural scenes using a text specific dictionary. Anwar et al. [24] propose a new image prior that represents the frequency bands of an image as a sparse linear combination filter responses of some sharp images. They analyze the images in Fourier domain and deblur using a training set that belongs to the same image class as the observed image.

Some Literature deal specifically with two-tone text images. Li et al. [26] propose a method that can jointly determine the blurring kernel and the true image. Their idea relies on adjusting the blurring kernel to provide a two-tone image. Jiang et al. [27] propose a two-phase method for text deblurring. First, they employ a two-tone prior to find an initial kernel estimate. In the second stage, they utilize the estimated kernel with a softer constraint on pixel intensities to estimate images that can extend beyond two-tone. Köhler et al. [25] propose a binarization driven text deblurring method for text images. They use a probability map that separates the background and text using a binary variable.

There are also more sophisticated methods that utilize machine learning tools to attack text deblurring problem. Hradiš et al. [28] use convolutional neural network to estimate blurring kernels and restore text images. Joshi et al. [29] use eigenfaces of multiple sharp face images to constrain the search space to successfully restore blurry face images.

As the deblurring problems are generally very large, most of the proposed methods incorporate some sort of an iterative scheme to solve the problem in reasonable time. Some commonly used iterative methods are Split Bregman iterations [30], alternating minimization algorithm [31], and half-quadratic splitting minimization [32]. There are also some novel methods proposed in the Literature. Tofghi et al. [33] propose an iterative method using projections onto epigraph sets of the TV function based on the optimization technique based on POCS framework proposed by Cetin et al. [34]. Tekalp et al. [35] propose a fast recursive identification algorithm for two-dimensional autoregressive image model.

Chapter 4

Non-Blind Deblurring Model for Almost Binary Images

Focusing on restoration of text images, our aim is to build a method that can successfully restore image details and provide a text-like solution. Although many methods have been proposed to solve deblurring problem for natural images, we find that many of them fail to provide a text-like solution due to the unique properties associated with text images. In our experiments we identified that these methods very often return images that are noisy and deficient in details. As such details may be crucial in text images, we place a great emphasis on the fact that our proposed model is robust to noise and capable of providing detailed solutions. Hence, we propose a non-blind model which provides a text-like solution without compromising image quality.

As our model aims to provide superior detail preservation, we restrict our efforts to a specific type of image. Although our model can be adapted to work on different kinds of text images, we consider it to be more effective for almost binary images. Almost binary, in this context, means that the pixel intensities of the image are accumulated around the extreme intensity values. Images typically have their intensities in the interval of $[0, 255]$. For our definition, we scale these values and use an intensity interval of $[0, 1]$ throughout approach. Hence, we work

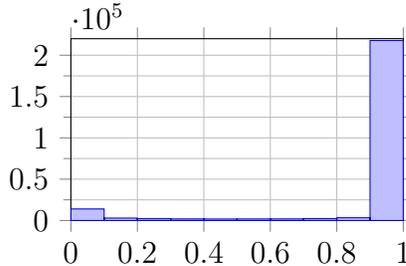


Figure 4.1: Example histogram for an image that is almost binary

on images that have a histogram of intensities similar to fig. 4.1. Luckily, such characteristic is present in many text images, allowing us to effectively use this model in a variety of problems.

In our formulation of the non-blind image deblurring problem, we assume that blurring kernel is a linear combination of a set of kernels. Hence, we formulate the observed image as:

$$y = x \circledast \left(\sum_{i=1}^n w_i h_i \right) + \epsilon, \quad (4.1)$$

where w is the weight of each convolution matrix in the real blurring operation. For the non-blind problem, where the blurring kernel is known, we simplify the expression to:

$$y = x \circledast h + \epsilon. \quad (4.2)$$

In our solution approach, we focus on the variational methods. We utilize regularization terms based on the statistical properties of images to overcome the ill-posedness problem. Apart from minimizing the estimation error, we resort to a commonly used principle of an image, that the edges of an image are very sparse. In the Literature, there are various regularization terms that exploit this characteristic of an image. Some popular applications are isotropic and anisotropic total variation [3], hyper-laplacian priors [5] and l1/l2 sparsity measures as used in [20]. In our case, we follow the l_0 minimization of the image intensities and gradients, which adds an element of non-convexity into our model. Moreover, for each pixel of the image, we push the image to get values at its upper and lower bounds. This allows the image to transform from a noisy image to a binary-like text image. We also add noise eliminating regularization terms, utilizing the first

and second derivatives.

4.1 The Non-Blind Model

Let us first define the variables of our model. We denote the true image by x , blurry/noisy image by y , and the known blurring kernel by h . The extended list of our variables and operators are defined in appendix A. Our model for solving the deblurring problem is as follows:

$$\min_x \|x \otimes h - y\|^2 + \phi\Phi_0 + \lambda\Phi_1 + \gamma\Phi_2 + \eta\Phi_3, \quad (4.3)$$

where Φ_0 denotes the convex regularization terms and $\Phi_i, i = 1, 2, 3$ denote separate non-convex regularization terms. These terms are explicitly:

$$\Phi_0 = \|\nabla_v x \otimes h - \nabla_v y\|^2 + \|\nabla_h x \otimes h - \nabla_h y\|^2 + \frac{\nu}{\phi} (\|\nabla_{hh} x\|^2 + \|\nabla_{vv} x\|^2), \quad (4.4)$$

$$\Phi_1 = \|\nabla x\|_0, \quad (4.5)$$

$$\Phi_2 = \|x\|_0 + \frac{\zeta}{\gamma} \|1 - x\|_0, \quad (4.6)$$

$$\Phi_3 = -(\|x\|^2 + \|1 - x\|^2). \quad (4.7)$$

The main term in the objective function minimizes the l_2 difference between the observed image and blurred image. One can observe that this is equivalent to minimizing $\|e\|^2$, the magnitude of the additive noise. It is known that minimizing l_2 length of the noise is especially effective at treating Gaussian noise, which is used in many application as expressed in [36].

The convex regularization term, Φ_0 , utilizes two different ideas. First part of the term uses first derivative in horizontal and vertical directions to minimize the difference between blurred and observed edges. The same idea is also used in [37] and [24]. It is generally used to suppress the ringing effect in resulting images, but is also used as a data fidelity term for kernel identification in some papers such as [25]. Second part of the term minimizes the second derivative of

the image. This term pushes the model to provide images with derivatives that are persistent in horizontal and/or vertical directions. We find this idea to be very powerful for finding the basic structures of the text images.

The first non-convex regularization term, Φ_1 , minimizes the number of non-zero edges. As the almost binary text images generally have pixels that have neighboring pixels with similar intensities, the resulting image is expected to have a small number of non-zero edges. Although this element is non-convex, [4] proposes a method to estimate a local solution for an optimization problem with this regularizer.

The second non-convex regularization term, Φ_2 , minimizes the number of pixels that do not have intensities of 0 or 1. This regularization term is based on the almost binary assumption. As such images have their intensities lie at extremes, one expects to find a very sparse set of pixels which have intensities in $x_i \in (0, 1)$. There are also other papers in the Literature that make use of the sparsity of pixel intensities in some frequency domain. Some of these applications are [4] where the model reduces number of non-zero pixels and [38] where the model reduces the dark channel of the recovered image.

The last regularization term, Φ_3 , is a concave term. As the model is a minimization problem, the resulting effect expected from this term is to increase the value of $\|x\|^2 + \|1 - x\|^2$. As this expression is smooth and increasing towards both edges, it favors solutions closer to extreme pixels, 0 and 1. So, we use this expression to shift the non-extreme values to edges, which iterates a current solution towards an almost binary image.

4.2 Solution Methodology

Having defined the essential regularization terms that can represent an almost binary text image, we focus on the solution methods. One can observe that solving this problem to optimality is not easily attainable. This is because the model

has three non-convex regularization terms. In fact, Φ_3 is concave, which can lead to an ill-defined problem structure. As a result, we restrict our attention to finding a good solution for this problem. First, we use the alternating minimization method proposed in [4] to overcome the non-convexity induced by the first regularization term, Φ_1 . This method depends on creating an auxiliary variable u to represent ∇x and adding the additional objective term that minimizes $\|\nabla x - u\|^2$. This allows the model to decompose into two sub-problems, which we will solve alternatively. The two sub-problems are:

$$\min_x \|x \otimes h - y\|^2 + \mu \|\nabla x - u\|^2 + \phi \Phi_0 + \gamma \Phi_2 + \eta \Phi_3, \quad (4.8)$$

$$\min_u \lambda \|u\|_0 + \mu \|\nabla x - u\|^2. \quad (4.9)$$

We can solve eq. (4.9) by a simple algebra. Its solution is as follows:

$$u_i^* = \begin{cases} \nabla x_i & (\nabla x_i)^2 \geq \frac{\lambda}{\mu} \\ 0 & \text{otherwise.} \end{cases} \quad (4.10)$$

In order to solve eq. (4.8), we need a method that can reliably find local solutions for large-scale, non-convex problems. For this reason, we turn to proximal gradient methods, specifically the methods in the Literature that can handle non-convex and non-smooth models. In this respect, we use the Accelerated Inexact Proximal Gradient proposed in [39]. To solve our model with Proximal Gradient, we split our objective function terms into convex and non-convex parts. We denote our convex elements by function f and non-convex elements using function g :

$$\begin{aligned} f(x) = & \|x \otimes h - y\|^2 + \mu \|\nabla x - u\|^2 + \phi (\|\nabla_v x \otimes h - \nabla_v y\|^2 + \|\nabla_h x \otimes h - \nabla_h y\|^2) \\ & + \nu (\|\nabla_{hh} x\|^2 + \|\nabla_{vv} x\|^2), \end{aligned} \quad (4.11)$$

and

$$g(x) = \gamma \|x\|_0 + \zeta \|1 - x\|_0 - \eta (\|x\|^2 + \|1 - x\|^2). \quad (4.12)$$

Using these functions, we can compute x iteratively, where each iterate x_s is found by using proximal gradient:

$$x_s = \text{prox}_{t_s}(g)(x_{s-1} - t_s \nabla f(x_{s-1})). \quad (4.13)$$

As f is smooth and convex, we can find its gradient. Moreover, we can compute its value very efficiently using the Fast Fourier Transform. This is because the Fourier transform of convolution of two functions is equivalent to pairwise product of the Fourier transforms of the two functions [40]. Explicitly, we use the following formula to compute the derivative of f at x_s :

$$\begin{aligned} \nabla f(x_s) = & \mathcal{F}^{-1}([\overline{\mathcal{F}(h)}\mathcal{F}(h) + \mu\overline{\mathcal{F}(\nabla)}\mathcal{F}(\nabla) + \phi(\overline{\mathcal{F}(h)}\overline{\mathcal{F}(\nabla_v)})\mathcal{F}(h)\mathcal{F}(\nabla_v) \\ & + \overline{\mathcal{F}(h)}\overline{\mathcal{F}(\nabla_h)}\mathcal{F}(h)\mathcal{F}(\nabla_h)])\mathcal{F}(x_s) - [\overline{\mathcal{F}(h)}\mathcal{F}(y) + \mu\overline{\mathcal{F}(\nabla_h)}\mathcal{F}(u_h) \\ & + \overline{\mathcal{F}(\nabla_v)}\mathcal{F}(u_v)] + \phi(\overline{\mathcal{F}(\nabla_v)}\overline{\mathcal{F}(h)})\mathcal{F}(\nabla_v)\mathcal{F}(y) \\ & + \overline{\mathcal{F}(\nabla_h)}\overline{\mathcal{F}(h)}\mathcal{F}(\nabla_h)\mathcal{F}(y)]) \end{aligned} \quad (4.14)$$

Then, assigning $z_s = x_{s-1} - t_s \nabla f(x_{s-1})$, we compute $x_s = \text{prox}_{t_s}(g)(z_s)$ through the following minimization problem:

$$\text{prox}_{t_s}(g)(z_s) = \underset{u}{\operatorname{argmin}} \{ \gamma \|u\|_0 + \zeta \|1 - u\|_0 - \eta (\|u\|^2 + \|1 - u\|^2) + \frac{1}{t_s} \|u - z_s\|^2 \}. \quad (4.15)$$

Notice that this minimization problem is equivalent to:

$$\begin{aligned} \text{prox}_{t_s}(g)(z_s) = & \underset{u}{\operatorname{argmin}} \{ \gamma \sum_{\forall i} \|u_i\|_0 + \zeta \sum_{\forall i} \|1 - u_i\|_0 - \eta (\sum_{\forall i} (u_i)^2 + \\ & \sum_{\forall i} (1 - u_i)^2) + \frac{1}{t_s} \sum_{\forall i} (u_i - z_{si})^2 \} \quad (4.16) \\ = & \sum_{\forall i} \underset{u}{\operatorname{argmin}} \{ \gamma \|u_i\|_0 + \zeta \|1 - u_i\|_0 - \eta ((u_i)^2 + (1 - u_i)^2) + \frac{1}{t_s} (u_i - z_{si})^2 \} \end{aligned}$$

Then this minimization problem can be separated into $|u|$ subproblems, allowing one to find the optimal solution of each element, u_i , of vector u separately. We further note that, under some specific condition, the optimal solution for the subproblem can be computed with simple algebra. We notice that non-convex

elements in the subproblems have very predictable values. The only cases when l_0 minimized terms are non-zero is when $u_i = 0$ or $u_i = 1$. Furthermore we can calculate the objective value under the two aforementioned conditions. Then we can search for the optimal solution by considering three cases: when $u_i = 0$, $u_i = 1$ and $u_i \neq \{0, 1\}$. We will find that beside knowing the objective value under the first two conditions, we can transform the third case into a convex problem under a very simple assumption. Let us consider the three cases:

Case 1: If $u_i = 0$:

$$\begin{aligned} c_i^{lb} &= \gamma \|u_i\|_0 + \zeta \|1 - u_i\|_0 - \eta(u_i^2 + (1 - u_i)^2) + \frac{1}{t_s}(u_i - z_{s_i})^2 \\ &= 0 + \zeta - \eta(0 + 1^2) + \frac{1}{t_s}z_{s_i}^2 \\ \implies c_i^{lb} &= \zeta + \frac{1}{t_s}z_{s_i}^2 - 1^2\eta. \end{aligned}$$

Case 2: If $u_i = 1$:

$$\begin{aligned} c_i^{ub} &= \gamma \|u_i\|_0 + \zeta \|1 - u_i\|_0 - \eta(u_i^2 + (1 - u_i)^2) + \frac{1}{t_s}(u_i - z_{s_i})^2 \\ &= \gamma + 0 - \eta(1^2 + 0) + \frac{1}{t_s}(1 - z_{s_i})^2 \\ \implies c_i^{ub} &= \gamma + \frac{1}{t_s}(1 - z_{s_i})^2 - 1^2\eta. \end{aligned}$$

Case 3: If $u_i \neq \{0, 1\}$:

$$\begin{aligned} c_i^{mid} &= \gamma \|u_i\|_0 + \zeta \|1 - u_i\|_0 - \eta(u_i^2 + (1 - u_i)^2) + \frac{1}{t_s}(u_i - z_{s_i})^2 \\ &= \zeta + \gamma + \min_{u_i} \{-\eta u_i^2 - \eta(u_i^2 - 2u_i + 1^2) + \frac{1}{t_s}(u_i^2 - 2u_i z_{s_i} + z_{s_i}^2)\}. \end{aligned}$$

The inner minimization problem is a polynomial of degree two, with the form:

$$\min_{u_i} \left\{ \left(\frac{1}{t_s} - 2\eta \right) u_i^2 + \left(-\frac{2z_{s_i}}{t_s} + 2\eta \right) u_i + C \right\},$$

where C is some constant. Assuming that our choice of parameter η satisfies $\frac{1}{t_s} - 2\eta > 0$, this problem becomes a convex minimization problem. It attains its minimum at:

$$u_i = \frac{\frac{z_{s_i}}{t_s} - \eta}{\frac{1}{t_s} - 2\eta}.$$

We can substitute this solution back into the inner minimization problem to find the optimal c_i^{mid} as well, which we later use to compare the objective values under the three stated conditions to determine the optimal u .

Having computed the optimal objective values for each state of u_i and their corresponding optimal values, we can compare the objective values to determine the optimal value for each element in eq. (4.15) as:

$$u_i^* = \begin{cases} 0 & , c_i^{lb} < \min\{c_i^{ub}, c_i^{mid}\} \\ 1 & , c_i^{ub} < \min\{c_i^{lb}, c_i^{mid}\} \\ \frac{z_{s_i}}{t_s} - \eta & , \text{otherwise.} \\ \frac{1}{t_s} - 2\eta & \end{cases} \quad (4.17)$$

Having determined the methods required to solve our model, we summarize our algorithm in algorithm 1 below.

Algorithm 1 Almost Binary Deblurring Algorithm

- 1: **procedure** NON-BLIND DEBLUR(h, y)
 - 2: **Assign** $s_0, \phi, \lambda, \gamma, \zeta, \eta, \mu$
 - 3: **Set** $x \leftarrow y, s \leftarrow 1$
 - 4: **while** $s < s_{max}$ **do**
 - 5: Solve for u using eq. (4.10)
 - 6: Compute z_s by using the derivative of eq. (4.11)
 - 7: Find the proximal mapping x_s by utilizing eq. (4.17)
 - 8: $\mu \leftarrow 2\mu, s \leftarrow s + 1$
 - 9: **end-while**
-

Chapter 5

Semi-Blind Deblurring Model for Almost Binary Images

Our proposed non-blind deblurring model in section 4.1 was:

$$\min_x \|x \otimes h - y\|^2 + \mu \|\nabla x - u\|^2 + \phi \Phi_0 + \gamma \Phi_2 + \eta \Phi_3, \quad (5.1)$$

where u is calculated using eq. (4.10). Whereas we could utilize our prior knowledge of h in that problem, in the semi-blind case, we do not have the full knowledge of h . Instead our aim is to determine the true image and blurring kernel simultaneously by utilizing the dictionary D . As in our formulation of the problem we assumed that the true blurring kernel is a linear combination of a set of kernels in the dictionary, our true image is represented as: $y = x \otimes (\sum_{i=1}^n w_i h_i) + \epsilon$, where h is the set of kernels in the dictionary and w is the weight of each convolution matrix in the real blurring operation. As the sum of the elements of a kernel is equal to 1 for the types of kernels we work on, we further impose the following restrictions on the model:

$$\mathbb{1}^T w = 1, w \geq 0. \quad (5.2)$$

First, we note that the convolution operator has the distributive property [41].

So, we can alternatively represent the observed image as:

$$y = (x \circledast (\sum_{i=1}^n w_i h_i)) + \epsilon = \sum_{i=1}^n (x \circledast (w_i h_i)) + \epsilon. \quad (5.3)$$

Moreover, we note that convolution operator is associative with scalar multiplication. As w_i is a scalar value, we can also re-write this formulation as:

$$y = \sum_{i=1}^n (x \circledast (w_i h_i)) + \epsilon = \sum_{i=1}^n (w_i (x \circledast h_i)) + \epsilon. \quad (5.4)$$

As one can use the above three formulations interchangeably, we use the following formulation in our model:

$$y = \sum_{i=1}^n x \circledast w_i h_i + \epsilon. \quad (5.5)$$

5.1 The Semi-Blind Model

Our formulation for the semi-blind deblurring problem is:

$$\begin{aligned} \min_{x,w} & \left\| \sum_{i \in D} x \circledast w_i h_i - y \right\|^2 + \mu \|\nabla x - u\|^2 + \phi \Phi_0 + \gamma \Phi_2 + \eta \Phi_3 \\ \text{s.t.} & \mathbb{1}^T w = 1 \\ & w \geq 0. \end{aligned} \quad (5.6)$$

Similar to the approach most commonly employed in the Literature, we can look for a solution by solving for x and w alternatively. Although this approach may not guarantee converge to a local solutions, in practical experiments we find that the solution converges to a good solution. In this respect, we decompose the model over x and w and consider the following two problems:

$$\min_x \|x \circledast H^* - y\|^2 + \mu \|\nabla x - u\|^2 + \phi \Phi_0 + \gamma \Phi_2 + \eta \Phi_3, \quad (5.7)$$

where $H^* = \sum_{i \in D} w_i^* h_i$ by eq. (5.3) and

$$\begin{aligned} \min_w & \left\| \sum_{i \in D} w_i U_i^* - y \right\|^2 \\ \text{s.t.} & \mathbb{1}^T w = 1 \\ & w \geq 0, \end{aligned} \tag{5.8}$$

where $U_i^* = x \otimes h_i^*$ by eq. (5.4).

We have already proposed an algorithm to solve the initial subproblem in algorithm 1. Given x , we can also attack eq. (5.8) to find a solution w^* . Note that, for fixed U^* values, this model can be expressed as a QP with equality and inequality constraints. Therefore, we can use many methods available to solve this problem efficiently. In practice, one of the fastest methods to solve such problems is the Fast NNLS method, which is an extension of the Active Set Method. We find in our computational studies that solving each QP problem using Fast NNLS method takes only fraction of a second, enabling us to handle kernel approximation processes very quickly.

5.2 Extended Model for Large Dictionaries

In the current structure of our model in eq. (5.6), we assume that our blurring kernel is a linear combination of a subset of kernels and find weights accordingly. However, we should note one possible issue. As we are working on a dictionary of kernels, which can be large in size, we may have to deal with many pre-defined kernels that are not effective in a given image. Even though our method is capable of being computationally efficient, there is another issue that can harm the effectiveness of our model. It is that solving eq. (5.8) may result in a kernel that over-fits the corresponding blurred image y . In other words, the model may disregard the effect of noise in the image by attaching positive weights to kernels that have zero weight in the true kernel, due to its motivation to close the gap between $w^T U^*$ and y . As a result, we find it beneficial to push the

model to use less kernels, in the hope of overcoming this problem. Similar to the approach proposed in [21], we put a restriction on the sparsity of kernel elements. Assuming that we want the model to use at most Γ kernels, we can reformulate model eq. (5.8) as:

$$\begin{aligned}
& \min_w \left\| \sum_{i \in D} w_i U_i^* - y \right\|^2 + \tau \|w\|^2 \\
& \text{s.t. } \mathbb{1}^T w = 1 \\
& \quad \|w\|_0 \leq \Gamma \\
& \quad w \geq 0.
\end{aligned} \tag{5.9}$$

We approximate this problem by adding $\|w\|_0$ as a regularization term to the objective. Then the approximate problem becomes:

$$\begin{aligned}
& \min_w \left\| \sum_{i \in D} w_i U_i^* - y \right\|^2 + \tau \|w\|^2 + \theta \|w\|_0 \\
& \text{s.t. } \mathbb{1}^T w = 1 \\
& \quad w \geq 0.
\end{aligned} \tag{5.10}$$

One issue here is how one should set θ so that the original restriction on w will hold. Although one may always use a trial-and-error approach, a theoretically proper choice of θ will be clear later.

Notice that the structure of this problem is very similar to eq. (4.3) in terms of its first non-convex regularization term. Again, we can solve this problem by using the alternating minimization method and considering the two sub-problems:

$$\begin{aligned}
& \min_w \left\| \sum_{i \in D} w_i U_i^* - y \right\|^2 + \tau \|w\|^2 + \alpha \|w - v\|^2 \\
& \text{s.t. } \mathbb{1}^T w = 1 \\
& \quad w \geq 0,
\end{aligned} \tag{5.11}$$

and

$$\min_v \alpha \|w - v\|^2 + \theta \|v\|_0. \tag{5.12}$$

We solve eq. (5.11) using a Non-Negative Least Squares algorithm. Similar to

eq. (4.9), the optimal solution of eq. (5.12) is:

$$v_i^* = \begin{cases} w_i & (w_i)^2 \geq \frac{\theta}{\alpha} \\ 0 & \text{otherwise.} \end{cases} \quad (5.13)$$

Going back to the optimal choice of θ , one can infer that the choice of θ depends on the design parameter α . Supposing that we want to make sure the model uses at most Γ kernels, setting $\theta = \frac{\alpha}{\Gamma^2}$ will push the model to use no more than Γ kernels. That is because setting as such would result in all positive weights, w_i , satisfying:

$$\begin{aligned} w_i^2 &\geq \frac{\frac{\alpha}{\Gamma^2}}{\alpha} = \frac{1}{\Gamma^2} \\ \implies w_i &\geq \frac{1}{\Gamma}. \end{aligned} \quad (5.14)$$

As we know that each resulting weight, w_i is larger than $\frac{1}{\Gamma}$ and the sum of all w_i 's is equal to 1, then we cannot have more than Γ non-zero weights.

Building upon our findings for the sub-problems, we summarize our algorithm in algorithm 2 below.

Algorithm 2 Almost Binary Deblurring Algorithm

```
1: procedure SEMI-BLIND DEBLUR( $y$ )
2:   Assign  $s_0, \phi, \lambda, \gamma, \zeta, \eta, \mu$ 
3:   Set  $x \leftarrow y, w \leftarrow \frac{1}{n} [1 \ 1 \ \dots \ 1]^T, s \leftarrow 1, t \leftarrow 1$ 
4:   while  $t < t_{max}$  do
5:     Set  $s \leftarrow 1$ 
6:     while  $s < s_{max}$  do
7:       Solve for  $u$  using eq. (4.10)
8:       Compute  $z_s$  by using the derivative of eq. (4.11)
9:       Find the proximal mapping  $x_s$  by utilizing eq. (4.17)
10:       $\mu \leftarrow 2\mu, s \leftarrow s + 1$ 
11:    end-while
12:    if  $s=1$  then
13:      Set  $v \leftarrow 0$ 
14:    else
15:      Solve for  $v$  using eq. (5.13)
16:    end-if
17:    Solve for  $w$  using NNLS Solver on eq. (5.11)
18:    Set  $\theta \leftarrow 2\theta, t \leftarrow t + 1$ 
19:  end-while
```

Chapter 6

Computational Study

Before discussing the performance of the proposed model, it is beneficial to reaffirm which NNLS solver to use in our constrained QP problem in eq. (5.11). For this purpose, we compare the average computational time for different commonly used solution methods. The solvers used in our tests are Active Set, Fast NNLS Method [42], Anti-Lopsided Algorithm [43], Nesterov’s Accelerated Method [44] and ADDM [45, 46]. Our for solution times for a single test image are given in table 6.1.

Table 6.1: The comparison of average computation time for a single kernel identification step for selected NNLS algorithms

Image	Solution Method	Solution Time(s)
Cartoon	Fast NNLS	0.0246
Cartoon	ADDM	0.0759
Cartoon	Nesterov	0.1183
Cartoon	Anti-Lopsided	0.2616
Cartoon	Active Set	0.4471

Based on its superior performance, we employ Fast NNLS method in our kernel identification step and use it to test the performance of our semi-blind algorithm.

6.1 Model Properties

Our experiments in the computational study were conducted over images that have varying sizes, ranging from 500x500 to 1000x1000 pixels. Although our methods were tested exclusively on grayscale images, we would expect a similar performance under color images due to favorable properties of text images. For blurring kernels, we used randomly generated kernels which are linear combinations of Motion-blur and/or Gaussian kernels. All of the images were disturbed by Gaussian noise with Signal-to-Noise Ratio of 40.

During our initial experiments, we have assigned our parameters some fixed values. They are $\hat{\phi} = 0.003$, $\hat{\lambda} = 0.0007$, $\hat{\gamma} = 0.14$, $\hat{\zeta} = 0.14$, $\hat{\eta} = 0.014$, $\hat{\mu} = 0.4$, $\hat{\nu} = 0.0002$. However, our later experiments with scaling these values according to iteration number showed that such an alteration improves solution quality significantly. In this respect, we decided to scale the parameters using two different weights. One of these weights are used to increase the effect of the parameter through iterations, and the other one is used to limit the non-convex terms to specific iterations. Their formulations are:

$$\gamma_s = k_3 \frac{s-1}{s} \hat{\gamma}, \zeta_s = k_3 \frac{s-1}{s} \hat{\zeta}, \eta_s = k_3 \frac{s-1}{s} \hat{\eta}, \nu_s = k_2 \frac{s-1}{s} \hat{\nu}, \quad (6.1)$$

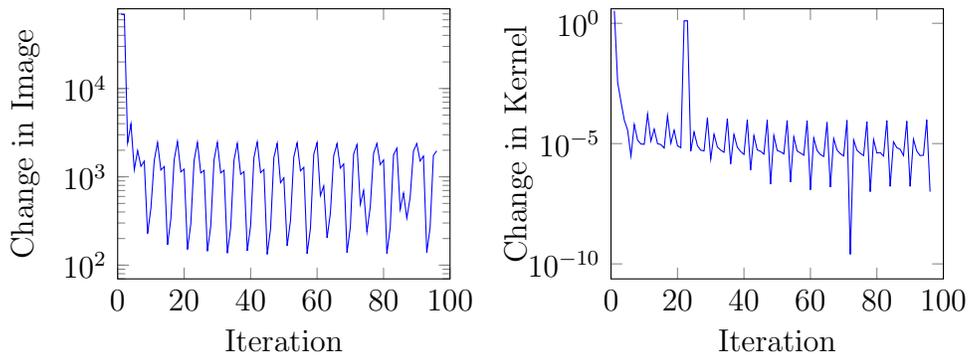
where s is the iteration number and k_2 and k_3 are calculated as:

$$k_i = \begin{cases} 1 & \text{mod}(s, i) = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (6.2)$$

This approach enhances our model in two ways. First, by scaling the parameters by $\frac{s-1}{s}$ at each iteration, we allow the model to start out the computation without the non-convex regularization terms, allowing the initial solutions to have minimal loss of information. Secondly, by using non-zero regularization weights only at some specific intervals using k , we do not force the model to pursue a specific path for solution, but rather allow various regularization terms to occasionally course-correct the model to direct the current solution towards desired characteristics of a text image. Compared to using regularization terms with

constant weights where changes to the solution are usually sudden and unpredictable, this change allows the model to correct itself, which in turn allows the model to not depend on a very strict termination rule. As this characteristic is not very common among image deblurring algorithms, we believe that this is an important addition to our approach.

As our models are heavily non-convex, we find it not possible to reach optimal solutions. However, we find it important to analyze how our images change throughout iterations. As many deblurring models require very precise adjustments and strict termination conditions, the detailed behavior of our solutions would help us determine the final solution more easily. For this, we plot the l_2 distance of the change in the image and kernel intensity throughout iterations. For each iteration we plot the distance between the current solution and the solution found in previous iteration. Our findings are as follows:



One notes that the images change rapidly for the first few iterations in a decreasing manner. After some point, the change fluctuates between 100 and 1000. We find that the reason for this fluctuation is due to our practice of scaling the regularization parameters through iterations. When the non-convex terms get positive weights in some specific iterations, model changes some pixels according to its weight. However we should note that when observing the interim images, this fluctuation does not affect the image quality, but rather makes the image go through some visually insignificant changes. In observation, the images go through a short series transformations in a circular manner and does not lose the required detail in the process. These fluctuations generally have a maximum

value of 2000, which translate to an average change of only 0.008 in pixel intensity for a 500x500 image.

6.2 Performance Metrics

Deblurring is a complex task that requires a multifaceted analysis for evaluation. Although there are some simple methods in the Literature that may handle the evaluation tasks for general performance of the system, an important portion of the performance evaluation relies on qualitative analysis.

One of the most commonly used metric used many different estimation problems is the Mean Squared Error (MSE). This metric determines the distance between the recovered and the original image, evaluating how effective method is in restoring an image close to the original. Assuming that the original image is denoted with x and recovered image is denoted with \hat{x} , the mean squared error is calculated as:

$$MSE = \frac{\sum_{\forall i, \forall j} (x_{ij} - \hat{x}_{ij})^2}{\sum_{\forall i, \forall j} 1}. \quad (6.3)$$

Based on MSE, another statistic that is more commonly encountered in deblurring literature is the Peak Signal-to-Noise Ratio (PSNR). It is a method frequently used to find compression quality and recovery performance. This statistic finds the ratio between the maximum intensity of the image and the effect of the noise in the recovered system. This ratio is assumed to reflect as an approximation for the human perception of quality. Therefore it allows one to find how good the restoration is in a practical manner. It's formulation is directly related to MSE and is defined as follows:

$$PSNR = 10 \log_{10} \frac{(x_{max})^2}{MSE}. \quad (6.4)$$

As we assume that the maximum intensity of an image element is 1, we directly use the following formula:

$$PSNR = 10 \log_{10} \left(\frac{1}{MSE} \right). \quad (6.5)$$

Whereas a good solution is expected to have a small MSE, the opposite holds for PSNR. As this ratio computes the maximum signal as a proportion of the noise error, a higher PSNR value translates to better solution.

Another similar metric we consider is Improvement in Signal-to-Noise ratio (ISNR). This metrics allows one to calculate the total improvement the model provides in terms of PSNR. It is calculated as:

$$ISNR = 10 \log_{10} \left(\frac{\sum_{\forall i, \forall j} (x_{ij} - y_{ij})^2}{\sum_{\forall i, \forall j} (x_{ij} - \hat{x}_{ij})^2} \right). \quad (6.6)$$

Another common used metric for deblurring is Structural Similarity index (SSIM) [47]. In image compressing, this metric is used to evaluate the perceived quality of an image. In deblurring, it is used to measure the similarity between two images. For a true image, x , and a restored image, \hat{x} , SSIM is calculated as follows:

$$SSIM = \frac{(2\mu_{\hat{x}}\mu_x + C_1)(2\sigma_{\hat{x}x} + C_2)}{(\mu_{\hat{x}}^2 + \mu_x^2 + C_1)(\sigma_{\hat{x}}^2 + \sigma_x^2 + C_2)}, \quad (6.7)$$

where $\mu_x, \mu_{\hat{x}}$ are the averages of x and \hat{x} , $\sigma_x, \sigma_{\hat{x}}$ are the variances of x and \hat{x} , $\sigma_{\hat{x}x}$ is the covariance matrix of x and \hat{x} , and C_1 and C_2 are the terms used to stabilize the division.

One should acknowledge that these performance metrics are not sufficient to compare the performance of different deblurring models. It is observed in Literature that having higher PSNR values is not necessarily equivalent to handling a more effective restoration. Although a good solution is expected to have a high PSNR value, comparing this value across applications may lead to a false observation. In that sense, one must also make qualitative assessments on the images to determine the best solution. Although this does not lead to very specific assessment of a solution, especially for text images, having a better readability and observing similar patterns in the original and estimated image is generally a reliable source of comparability. In that respect, we draw attention to evaluating the performance of different methods based on how close the estimated solution is to the original, and how the models behave in providing a text-like solution.

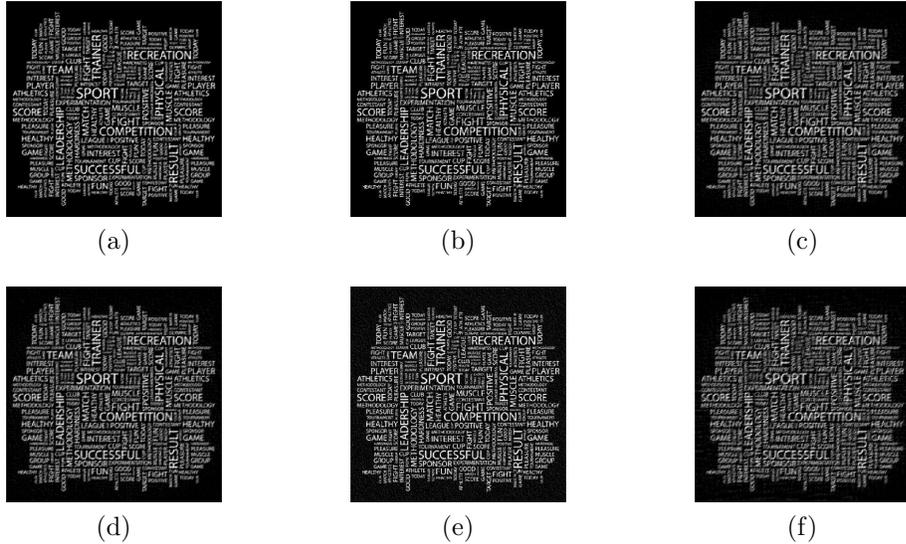


Figure 6.1: (a)Our method. (b)Lucy-Richardson Algorithm [1, 2]. (c)Total Variation [3]. (d)l0 regularized model [4]. (e) Hyper-laplacian priors [5]. (f) [6].

6.3 Performance Analysis

To verify the effectiveness of our proposed model, we present four images constructed with different blurring kernels and random noises. The test images and kernels we use are available in appendix B. First, we test the model under the assumption that the blurring kernel is known and compare our findings with other applications in the Literature. We get the results reported in figs. 6.1 to 6.4. We also compare the PSNR, ISNR and SSIM values of the estimated images. They are given in tables 6.2 to 6.4. We also present the results for some blurred images commonly encountered in the Literature in appendix C.

Table 6.2: The comparison of PSNR values for estimated images

Image	Our Method	Lucy Richardson [1, 2]	Total Variation [3]	L0 Regularization [4]	Hyper Laplacian [5]	Outliers [6]
Vector	15.526	18.222	14.603	15.426	19.949	14.598
Handwriting	18.740	15.010	17.875	15.973	17.688	15.973
Cube	14.545	12.447	13.264	11.850	12.959	11.998
Cartoon	28.457	14.876	19.757	22.038	18.750	17.645

Table 6.3: The comparison of ISNR values for estimated images

Image	Our Method	Lucy Richardson [1, 2]	Total Variation [3]	L0 Regularization [4]	Hyper Laplacian [5]	Outliers [6]
Vector	4.3026	7.8776	2.6301	3.7220	9.6061	2.5771
Handwriting	8.8649	3.6871	6.9979	4.9201	6.8094	5.1233
Cube	5.3054	1.9786	2.8840	1.1956	2.4897	1.3812
Cartoon	14.6084	0.7244	3.6800	6.6201	3.4216	-0.6100

Table 6.4: The comparison of SSIM values for estimated images

Image	Our Method	Lucy Richardson [1, 2]	Total Variation [3]	L0 Regularization [4]	Hyper Laplacian [5]	Outliers [6]
Vector	0.8776	0.9333	0.7337	0.7451	0.7574	0.6623
Handwriting	0.8199	0.5449	0.6790	0.5474	0.6688	0.4807
Cube	0.8438	0.7103	0.7493	0.6015	0.7204	0.5626
Cartoon	0.9926	0.4307	0.6324	0.7264	0.5309	0.7414

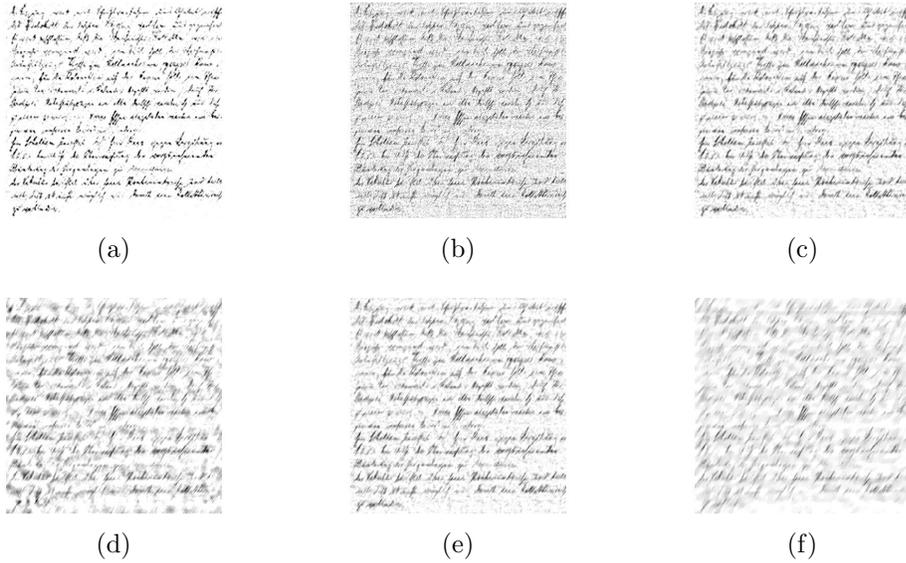


Figure 6.2: (a)Our method. (b)Lucy-Richardson Algorithm [1, 2]. (c)Total Variation [3]. (d)l0 regularized model [4]. (e) Hyper-laplacian priors [5]. (f) [6].

We find that, compared to some existing Literature, our model performs quite

favorably in terms of delivering a text-like solution. Whereas many methods are qualified to find a solution that can recover text structures, our model specializes in providing a solution that looks like a text image without noise.

After reaffirming the effectiveness of our non-blind model in restoring the main structure of the image without losing much detail, we now test the semi-blind model using a dictionary that only consists of true kernels. This can show how well the model behaves in restoring the blurring kernel. Our findings are given in figs. 6.5 to 6.7.

Finally, we experiment with a dictionary which is made up of a large number of kernels which or may not be effective in our image. This feature adds two elements to our problem. First, when the number of true kernels is small but the dictionary is large, the large number of kernels become a decoy to our model, which helps us evaluate the efficiency of our model when we are not certain about which kernels affect the image. Secondly, using a sufficiently large number of kernel allows us to recover an image whose blurring kernels are completely unknown. Hence, theoretically, this approach can be used to solve blind deblurring models. We test our "cartoon" image under this condition. The estimated solution is as in fig. 6.8. We also plot the change in PSNR values through iterations. It is presented in fig. 6.9.

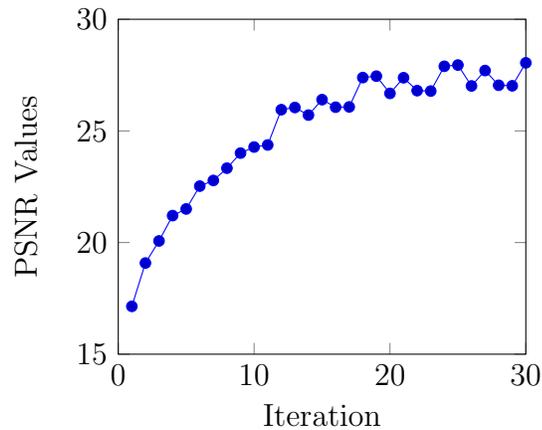


Figure 6.9: PSNR value of the restored image through iterations.

6.4 Sensitivity Analysis

Generally, in deblurring models, the solutions are very sensitive to change in regularization parameters. For this reason we also consider how our solution changes when we change our assigned parameter values. As our model is highly non-convex we analyze the sensitivity of our model to parameter weights by computing the solution for different values of $\phi, \nu, \mu, \gamma, \zeta$ and η for the Handwriting image. For each weight, we fix the other parameters and scale its value by 15 different weights ranging from 0 to 50. Then, we compute the l_2 change in the solution compared to the solution without the scale. This allows us to analyze how precisely one needs to set the parameters without losing the effectiveness of our algorithm. Our findings are in fig. 6.10.

We find that for a small change in parameters values, the estimated solution does not show a significant change. Increasing or decreasing the parameter values by less than 15% results in a change of at most 1% at the newly estimated image. We also find that the maximum change in any pixel of the estimated images is 25 for this case. On the other hand, when the parameters are changed significantly, the changes in estimated solution also become more extreme. We find that that increasing the parameter values by more than 100% leads to a change of more than 150 for some pixel, hence showing the sensitivity of the parameters to large changes.

We also analyze how each regularization term affects the image by comparing the solution where the regularizer has zero weight to the solution with increased weight. This allows us to see the effect each term has in deblurring the image. We present our findings in appendix D.

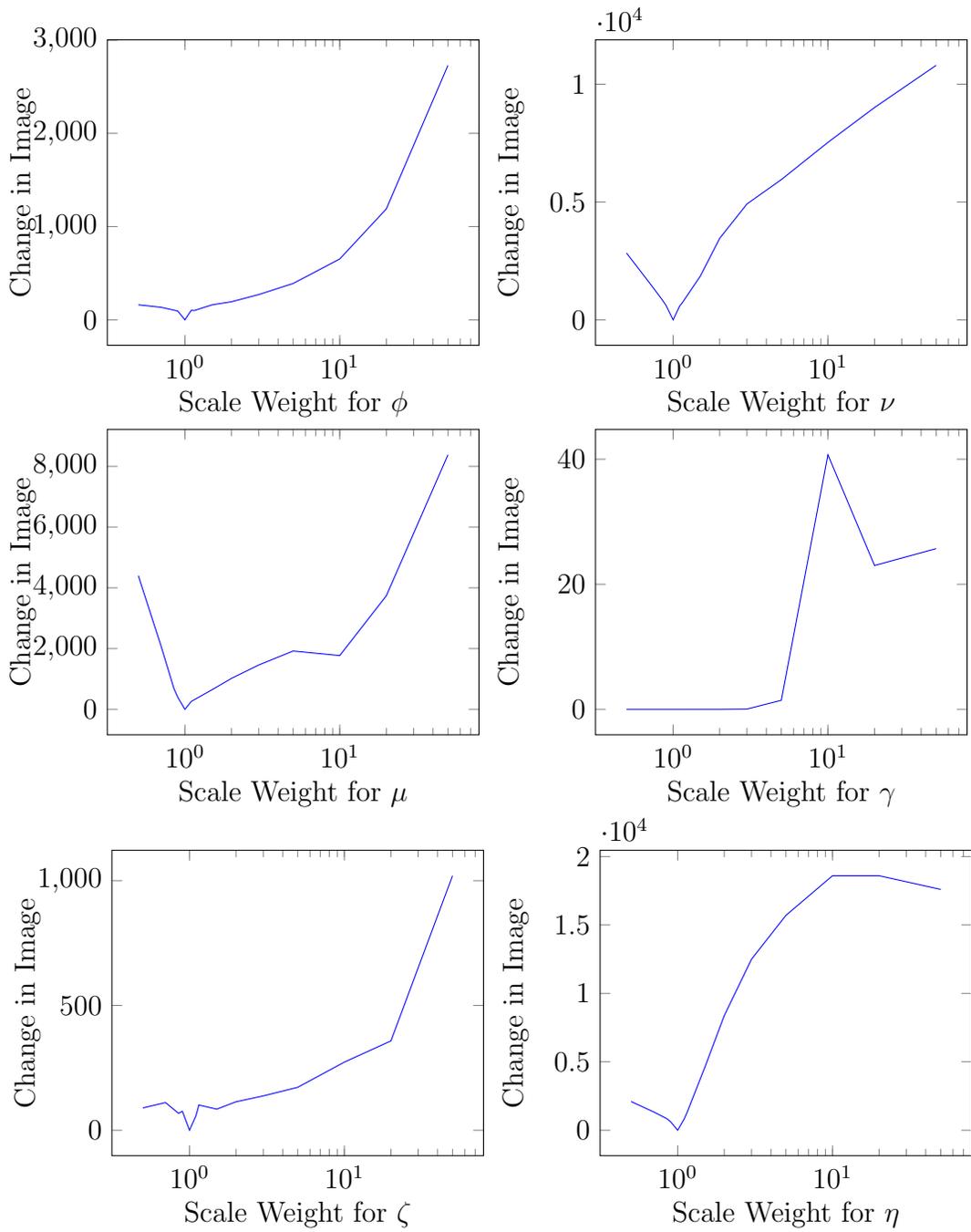
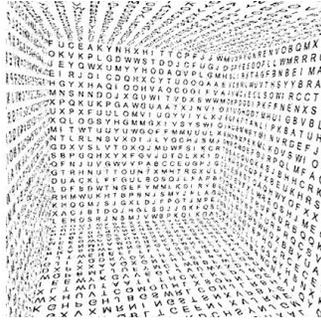
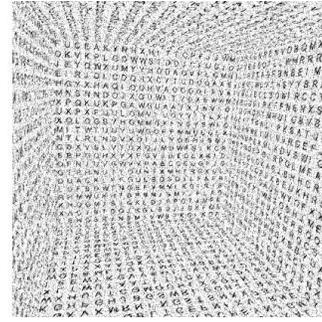


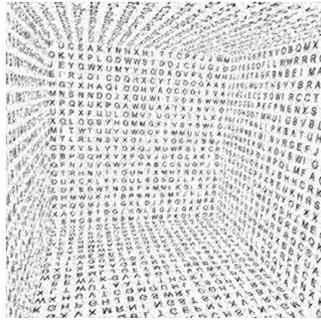
Figure 6.10: Total l_2 change in the estimated when we scale parameters



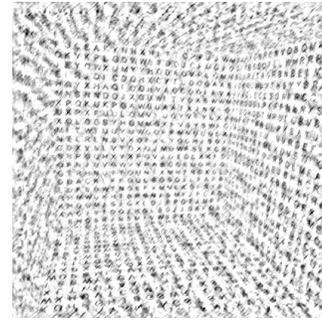
(a)



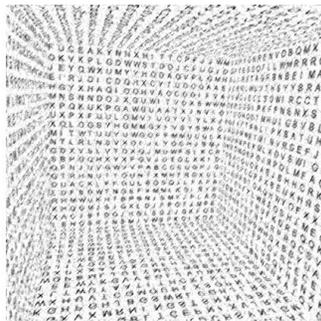
(b)



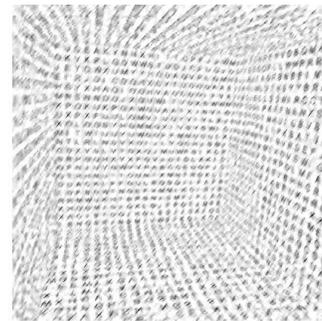
(c)



(d)



(e)

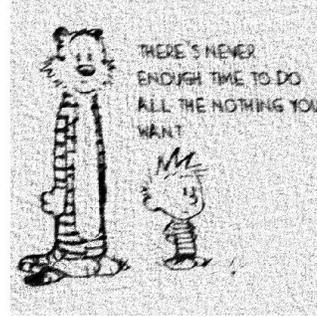


(f)

Figure 6.3: (a)Our method. (b)Lucy-Richardson Algorithm [1,2]. (c)Total Variation [3]. (d)10 regularized model [4]. (e) Hyper-laplacian priors [5]. (f) [6].



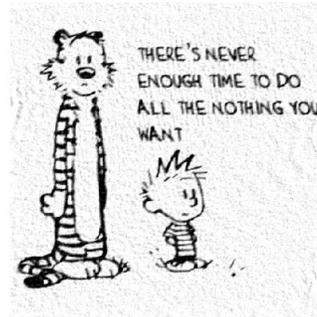
(a)



(b)



(c)



(d)



(e)



(f)

Figure 6.4: (a) Images recovered using our proposed method. (b) Image recovered using Lucy-Richardson Algorithm [1,2]. (c) Image recovered using Total Variation Method [3]. (d) Image recovered using the l_0 regularized model [4]. (e) Image recovered using hyper-laplacian priors using model [5]. (f) Image recovered by the method in [6].



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)

Figure 6.8: Image recovery by our model using a large dictionary. Top to bottom: true image and kernel, two intermediate solutions, the final solution.

Chapter 7

Specially Structured Blur Kernels

We have stated in section 2.2 that our representation of a blurring kernel as linear combination of simpler kernels allows us to efficiently pursue the true kernels for some special settings. In this section we will design a few instances where this proposition holds true.

Let us assume again that we are working on a deblurring problem with some limited knowledge of the blurring kernel. This prior knowledge can consist of kernel's type, size, length or angle depending on the problem setting. In most of the currently available methods in the Literature, it is not possible to incorporate such information directly into the model. Due to non-existence of this feature, the performance of these methods do not generally improve in the light of additional information. As in many practical cases some unutilized knowledge is already present in the problem definition, it is important to present an extensible approach to make better use of the prior information in kernel identification process. With the incorporation of kernel dictionaries, our model can accomplish such a task.

In the model we use the dictionary elements to act as a building block for the kernel. As dictionary design is an independent process that must be handled

before using the deblurring algorithm, one can make use of this step by designing it according to the problems' specific requirements. On one side, one can pursue any kernel structure similar to a blind problem by creating a very large dictionary, on the other side, one can use a very limited dictionary that forces the model to find only specific kernel structures by restricting the search space. This approach adds a valuable flexibility to our model by allowing it to adapt to many different settings. We find that this feature not only reduces the problem size substantially, but also allows one to estimate kernels more accurately.

First, consider a completely blind deblurring problem with no prior knowledge. When dealing with this problem, one can create a very large dictionary with each element of the dictionary representing one unique pixel that has a positive probability of being present in the true kernel. In this setting, our model would behave very similar to any blind deblurring problem, where every single pixel of the kernel is computed separately. Although this approach may be a requirement for the blind problem, it would not be efficient to stick with this approach if we have additional information about the kernel.

7.1 Motion Blurring Problem

Let us now restrict our attention to a single motion deblurring problem. Assume that we have the following prior information about the blurring kernel:

1. The image is affected by a shift-invariant motion blur.
2. The angle of motion is known or can be easily estimated.
3. The length of motion blur is unknown, but has estimated upper and lower bounds.

From a blind deblurring perspective, one needs create a dictionary as large as the maximum number of pixels. However, in this problem we can use our prior knowledge to create a much smaller dictionary where each element supplements

some unique pixels, which collectively contain the sufficient pixels groups to perfectly restore any motion blur kernel within the search space. For example, let us assume that our motion blur kernel is horizontal with an unknown true length of 31 pixels, but is estimated to be between between 28 and 35 pixels in length. In this setting, we can design a dictionary, where the first element is a motion blur kernel with a length of 28, and each additional element is a single pixel not represented in the smaller elements. For this problem, they would look as in fig. 7.1.

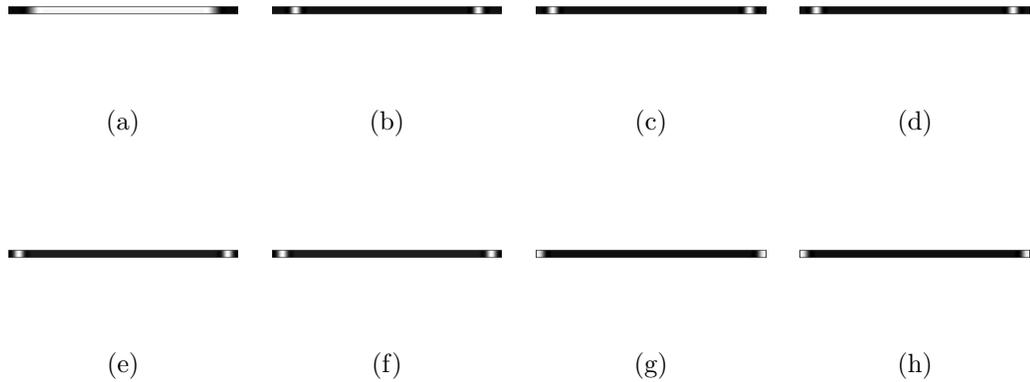


Figure 7.1: Sample dictionary to solve semi-blind motion deblurring problem

Notice that by using linear combinations of these 8 kernels, one can construct any motion blur kernel, which has an angle of 0° and has a length between 28 and 35 pixels. Whereas another blind deblurring model may determine kernels by solving a problem with as many as 35 variables, by this approach, one can find it by using 8 variables. This difference arises from the flexibility of our approach, which allows us to create a custom dictionary for a specific problem setting. By using fewer variables, we do not only reduce the computation time, but also by using the specific structure groups, we can represent a motion blur kernel with greater precision. If one uses this example structure to pursue the true image using our proposed methods, the solution is as in fig. 7.2.

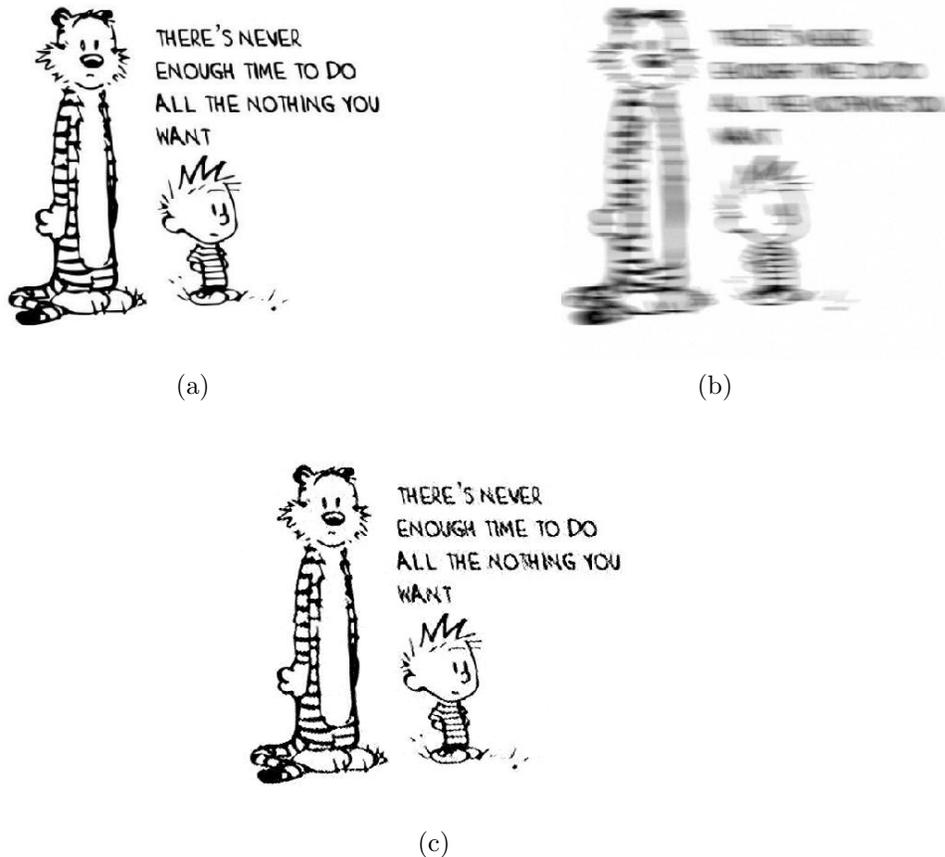


Figure 7.2: Images affected by semi-blind motion blurring kernels recovered using a specially designed dictionary. (a) True Image. (b) Observed Image. (c) Restored Image.

7.2 Out-of-Focus Blurring Problem

Let us consider a blurring problem affected by an out-of-focus blur with a radius estimated to be between two predetermined values. To create an efficient dictionary for the out-of-focus blur, we can decompose the kernel into its elements in the form of concentric circles. By each circle providing the unique pixel not represented by any smaller element, one can design a dictionary that can be used to construct any out-of-focus blur kernel with a radius between desired values.

For example, assume that we are pursuing an out-of-focus blur kernel with an unknown true radius of 15 pixels, but is estimated to be an odd number between

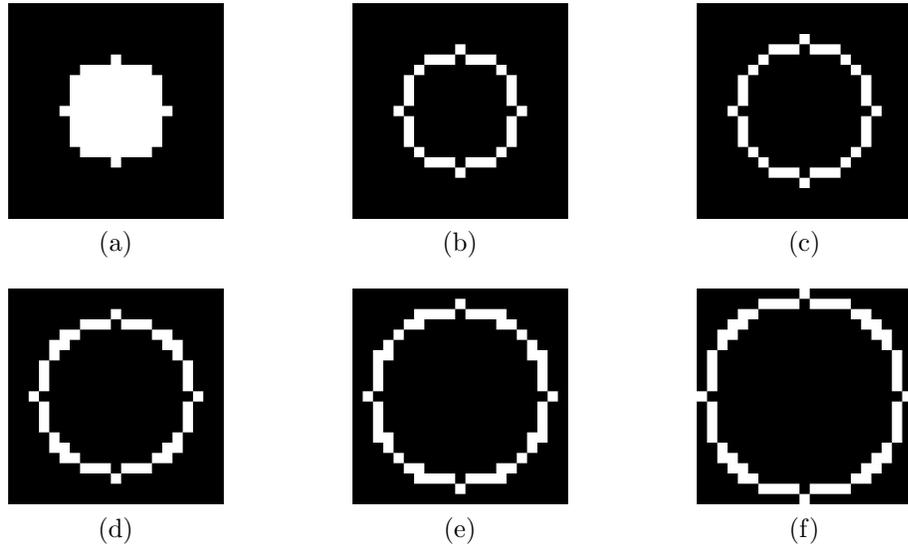


Figure 7.3: Sample dictionary to solve semi-blind out-of-focus deblurring problem

11 and 21 pixels. In this setting, we can design a new dictionary, with the first element being an out-of-focus blur with a radius of 11, and each additional element containing the unique pixels in the form of circles. They would look as in fig. 7.3.

By using the elements of this dictionary, one can construct any out-of-focus kernel with an odd numbered radius between 11 and 21 pixels. The major advantage of this approach is that to estimate the kernel using this dictionary, one needs to use only 6 variables, whereas pursuing the blurring kernel without incorporating the prior knowledge would require as many as 441 variables. Again, using kernel's unique structure allows us to reduce the number of variables in our problem, which can lead to sizable performance and quality improvements. When we solve this semi-blind problem using the dictionary above, the recovered image is as in fig. 7.4.



(a)



(b)



(c)

Figure 7.4: Images affected by semi-blind out-of-focus blurring kernels recovered using a specially designed dictionary. (a) True Image. (b) Observed Image. (c) Restored Image.

7.3 Gaussian Blurring Problem

Let us now consider an image that is blurred by a Gaussian kernel. In a practical setting, we may have a prior knowledge about some characteristic of the kernel, such as its size or standard deviation. Similar to the methods we employed above, we can utilize this knowledge to create a model setting that would allow us to determine the kernel and the original image very effectively. For example, we can consider a setting where our aim is to deblur an image that is affected by a gaussian blur with a size less than or equal to 11. We can limit our knowledge to its size and not pursue any information about the standard deviation of the

kernel. Even without such knowledge we can construct a sufficient dictionary for this problem. In fact, one only needs a dictionary of 22 elements to sufficiently restore any Gaussian kernel with any standard deviation with size less than or equal to 11. Compared to a potential size of 121 elements, this approach reduces the complexity of the problem and allows one to find solutions faster and more accurately. Assuming that we pursue a solution for this problem where the original kernel has a size of 11 and a standard deviation of 4, we recover the image and kernel provided in fig. 7.5.

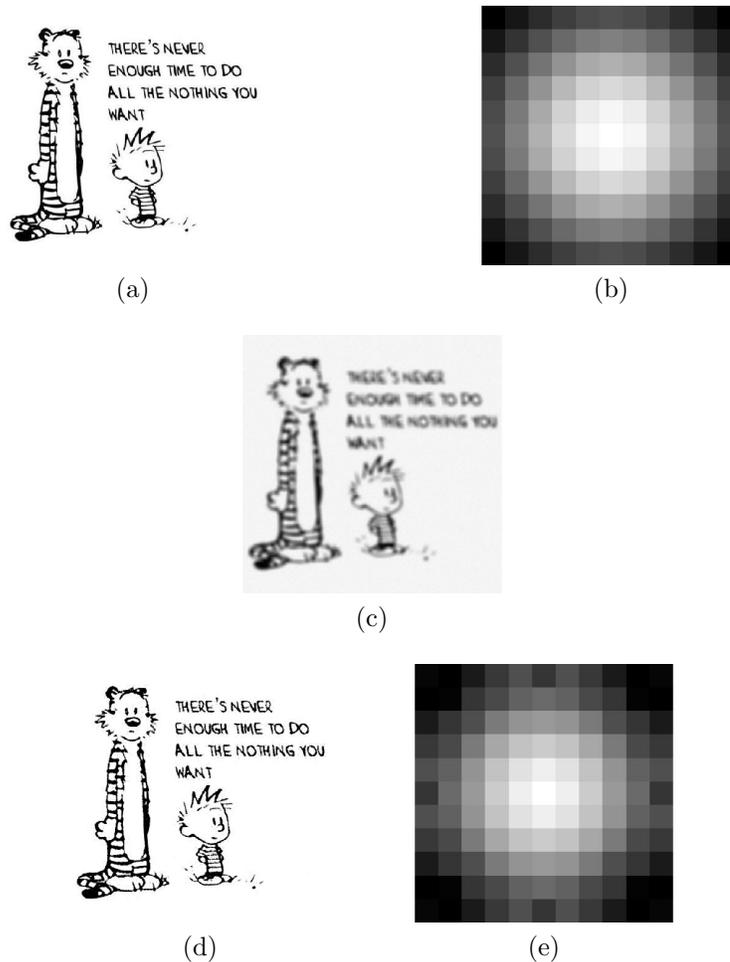


Figure 7.5: Images affected by semi-blind gaussian blurring kernels recovered using a specially designed dictionary. (a) True Image. (b) True Kernel. (c) Observed Image. (d) Restored Image. (e) Restored Kernel.

Although we restrict our efforts to providing examples for only three types of

kernels, it is possible for one to carry the same principles to many other types of kernels. As many problem settings inherently contain a prior knowledge about the type of blurring kernel one needs to consider, we can exploit the kernel structures to determine the kernels using a small number of elements. Especially in cases where the observation setting is self-designed, which involves many practices such as license plate recognition, barcode reading or hand-writing detection, one can utilize the dictionary to restrict the search space and find kernels faster and more accurately using our semi-blind deblurring model.

Chapter 8

Conclusion

In the thesis, we proposed a new method for deblurring text images under non-blind and semi-blind settings. We provided an algorithm that can estimate a deblurred image by using efficient arithmetic operations. One of the core contributions of our proposed method is its effectiveness in restoring almost binary text images. As found in the computational analysis, our algorithm can effectively deblur a text-like image, comparing favorably with many state-of-the-art methods in the Literature.

Although one thing we did not focus extensively on is the converge of our proposed methods, we find it to be a secondary concern for our approach. As many image restoration approaches rely on using very ill-defined models, many methods strive to find a local solution, which may provide inconsistent estimation performances. We instead focused our attention to designing an algorithm that performs specific tasks with predictable outcomes. In the process, we strived to design our algorithm in such a way that each iteration addresses a specific characteristic of the desired image to improve the current estimation. Due to this, our approach relies largely on the idea of regularization, not only in the sense of adding terms to our objective function, but even in our design of the algorithm our aim is for the model to regularize itself through iterations to find a good solution without requiring a tight termination condition. Such an approach

can be a valuable addition to the existing Literature.

Another advantage of our current model is its extensibility. We find that despite the specialization of our model towards specific image types, it can be adapted to work on various different settings by the addition of new terms and choice of solution methods. One such interesting addition may be to alter the model to handle images that have multiple accumulation points, which can allow one to work on images with multiple layers. Another future research direction we consider important is extending the model to handle images with backgrounds that has a change in intensity through some direction. Modifying the model to work with such images can allow one to work in setting where the object of interest is a text under various illumination conditions.

Last addition we hope will be relevant to literature is our approach to construction of kernel dictionary for the semi-blind deblurring problem. Currently, most of the Literature is divided into blind and non-blind problem settings. However, in many applications, there is some inherent knowledge about the blurring process for one to take advantage of. In such a case it is most beneficial to focus exclusively on the type of blur to eliminate any potential errors. In our setting we remedy this need by our proposed setting with dictionaries. We find that our approach not only allows one to restrict the search space for the deblurring problem, but also does this without requiring any assumptions about the image.

Bibliography

- [1] W. H. Richardson, “Bayesian-based iterative method of image restoration*,” *J. Opt. Soc. Am.*, vol. 62, pp. 55–59, Jan 1972.
- [2] L. B. Lucy, “An iterative technique for the rectification of observed distributions,” *The Astronomical Journal*, vol. 79, p. 745, June 1974.
- [3] L. I. 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.
- [4] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, “Deblurring text images via l0-regularized intensity and gradient prior,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2901–2908, June 2014.
- [5] D. Krishnan and R. Fergus, “Fast image deconvolution using hyper-laplacian priors,” in *NIPS* (Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, eds.), pp. 1033–1041, Curran Associates, Inc., 2009.
- [6] S. Cho, J. Wang, and S. Lee, “Handling outliers in non-blind image deconvolution,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 495–502, IEEE, 2011.
- [7] S. Murray and P. J. Bex, “Perceived blur in naturally contoured images depends on phase,” *Frontiers in psychology*, vol. 1, p. 185, 2010.
- [8] C.-C. Lee and W.-L. Hwang, “Mixture of gaussian blur kernel representation for blind image restoration,” *IEEE Transactions on Computational Imaging*, vol. 3, no. 4, pp. 783–797, 2017.

- [9] T. G. Stockham, T. M. Cannon, and R. B. Ingebretsen, “Blind deconvolution through digital signal processing,” *Proceedings of the IEEE*, vol. 63, pp. 678–692, April 1975.
- [10] M. Cannon, “Blind deconvolution of spatially invariant image blurs with phase,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, pp. 58–63, Feb 1976.
- [11] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. The MIT Press, 1964.
- [12] Y. Zheng, “A fast image deblurring algorithm using the wiener filter and the hartley transform,” in *Review of Progress in Quantitative Nondestructive Evaluation*, pp. 735–742, Springer, 1989.
- [13] M. I. Sezan and A. M. Tekalp, “Adaptive regularized image restoration using the theory of convex projections,” in *Decision and Control, 1987. 26th IEEE Conference on*, vol. 26, pp. 1208–1209, IEEE, 1987.
- [14] P. C. Hansen, J. G. Nagy, and D. P. O’Leary, *6. Regularization by Spectral Filtering*, pp. 71–86. Society for Industrial and Applied Mathematic, 2006.
- [15] L. A. Shepp and Y. Vardi, “Maximum likelihood reconstruction for emission tomography,” *IEEE Transactions on Medical Imaging*, vol. 1, pp. 113–122, Oct 1982.
- [16] M. Temerinac-Ott, O. Ronneberger, P. Ochs, W. Driever, T. Brox, and H. Burkhardt, “Multiview deblurring for 3-d images from light-sheet-based fluorescence microscopy,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1863–1873, 2012.
- [17] M. Keuper, T. Schmidt, M. Temerinac-Ott, J. Padeken, P. Heun, O. Ronneberger, and T. Brox, “Blind deconvolution of widefield fluorescence microscopic data by regularization of the optical transfer function (otf),” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 2179–2186, IEEE, 2013.

- [18] A. N. Tikhonov, “Solution of incorrectly formulated problems and the regularization method,” *Soviet Math. Dokl.*, vol. 4, pp. 1035–1038, 1963.
- [19] L. Xu, S. Zheng, and J. Jia, “Unnatural l0 sparse representation for natural image deblurring,” in *CVPR*, pp. 1107–1114, IEEE Computer Society, 2013.
- [20] D. Krishnan, T. Tay, and R. Fergus, “Blind deconvolution using a normalized sparsity measure,” in *CVPR*, pp. 233–240, IEEE Computer Society, 2011.
- [21] C. C. Lee and W. L. Hwang, “Sparse representation of a blur kernel for out-of-focus blind image restoration,” in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 2698–2702, Sept 2016.
- [22] H. Cho, J. Wang, and S. Lee, “Text image deblurring using text-specific properties,” in *European Conference on Computer Vision*, pp. 524–537, Springer, 2012.
- [23] X. Cao, W. Ren, W. Zuo, X. Guo, and H. Foroosh, “Scene text deblurring using text-specific multiscale dictionaries,” *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1302–1314, 2015.
- [24] S. Anwar, C. Phuoc Huynh, and F. Porikli, “Class-specific image deblurring,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 495–503, 2015.
- [25] T. Köhler, A. Maier, and V. Christlein, “Binarization driven blind deconvolution for document image restoration,” in *German Conference on Pattern Recognition*, pp. 91–102, Springer, 2015.
- [26] T.-H. Li and K.-S. Lii, “A joint estimation approach for two-tone image deblurring by blind deconvolution,” *IEEE transactions on image processing*, vol. 11, no. 8, pp. 847–858, 2002.
- [27] X. Jiang, H. Yao, and S. Zhao, “Text image deblurring via two-tone prior,” *Neurocomputing*, vol. 242, pp. 1–14, 2017.
- [28] M. Hradiš, J. Kotera, P. Zemčík, and F. Šroubek, “Convolutional neural networks for direct text deblurring,” in *Proceedings of BMVC*, vol. 10, 2015.

- [29] N. Joshi, W. Matusik, E. H. Adelson, and D. J. Kriegman, “Personal photo enhancement using example images.,” *ACM Trans. Graph.*, vol. 29, no. 2, pp. 12–1, 2010.
- [30] T. Goldstein and S. Osher, “The split bregman method for l1-regularized problems,” *SIAM J. Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.
- [31] T. F. Chan and C.-K. Wong, “Total variation blind deconvolution,” *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 370–375, 1998.
- [32] L. Xu, C. Lu, Y. Xu, and J. Jia, “Image smoothing via l0 gradient minimization,” *ACM Trans. Graph.*, vol. 30, no. 6, p. 174, 2011.
- [33] M. Tofighi, K. Kose, and A. E. Cetin, “Denoising images corrupted by impulsive noise using projections onto the epigraph set of the total variation function (pes-tv),” *Signal, Image and Video Processing*, vol. 9, no. 1, pp. 41–48, 2015.
- [34] A. E. Cetin, A. Bozkurt, O. Gunay, Y. H. Habiboglu, K. Kose, I. Onaran, M. Tofighi, and R. A. Sevimli, “Projections onto convex sets (pocs) based optimization by lifting,” in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pp. 623–623, IEEE, 2013.
- [35] A. Tekalp, H. Kaufman, and J. Woods, “Fast recursive estimation of the parameters of a space-varying autoregressive image model,” *IEEE transactions on acoustics, speech, and signal processing*, vol. 33, no. 2, pp. 469–472, 1985.
- [36] C. A. Micchelli, L. Shen, Y. Xu, and X. Zeng, “Proximity algorithms for image models ii: L1/tv denosing,” *Advances in Computational Mathematics, online version available*, 2011.
- [37] V. S. Prasath and A. Singh, “Ring artifact reduction in blind image deblurring and denoising problems by regularization methods,” in *Advances in Pattern Recognition, 2009. ICAPR’09. Seventh International Conference on*, pp. 333–336, IEEE, 2009.

- [38] J. Pan, D. Sun, H. Pfister, and M.-H. Yang, “Blind image deblurring using dark channel prior,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1628–1636, 2016.
- [39] B. Gu, Z. Huo, and H. Huang, “Inexact proximal gradient methods for non-convex and non-smooth optimization,” *arXiv preprint arXiv:1612.06003*, 2016.
- [40] W. H. Press, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.
- [41] M. H. Hayes, *Schaum’s outline of digital signal processing*. McGraw-Hill, Inc., 1998.
- [42] R. Bro and S. D. Jong, “A fast non-negativity-constrained least squares algorithm,” *Journal of Chemometrics*, vol. 11, pp. 393–401, 1997.
- [43] D. K. Nguyen and T. B. Ho, “Anti-lopsided algorithm for large-scale non-negative least square problems,” *arXiv preprint arXiv:1502.01645*, 2015.
- [44] Y. Nesterov, “A method of solving a convex programming problem with convergence rate $o(1/k^2)$,” in *Soviet Mathematics Doklady*, vol. 27, pp. 372–376, 1983.
- [45] D. Gabay and B. Mercier, “A dual algorithm for the solution of nonlinear variational problems via finite element approximation,” *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.
- [46] R. Glowinski and A. Marroco, “Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de dirichlet non linéaires,” *Revue française d’automatique, informatique, recherche opérationnelle. Analyse numérique*, vol. 9, no. R2, pp. 41–76, 1975.
- [47] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

Appendix A

Variables and Derivative Operators

Table A.1: Definition of Variables

Symbol	Description
x	True Image
y	Observed Blurred Image
ϵ	Random Noise
h	Blurring Kernel
H	Blurring matrix
D	Kernel Dictionary
w	Kernel Weights
∇_h	Horizontal Derivative
∇_v	Vertical Derivative
∇	Derivative
∇_{hh}	Horizontal Second Derivative
∇_{vv}	Vertical Second Derivative

Table A.2: Formulations for Derivative Operators

Operator	Formulation
$\nabla_h x$	$[1 \ -1] \otimes x$
$\nabla_v x$	$\begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes x$
∇x_i	$\sqrt{(\nabla_h x)_i^2 + (\nabla_v x)_i^2}$
∇_{hh}	$[1 \ -2 \ 1] \otimes x$
∇_{vv}	$\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \otimes x$

Appendix B

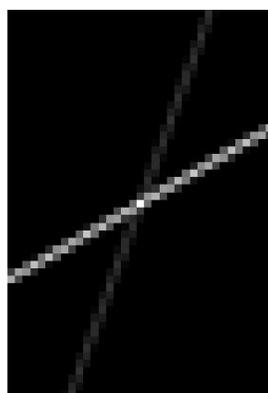
Testing Images and Kernels



(a)



(b)

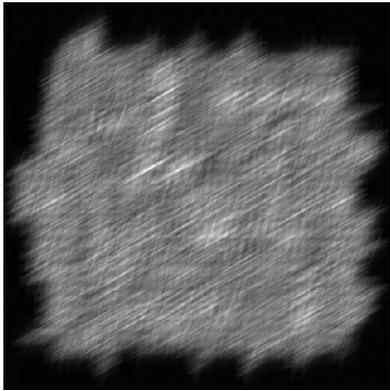


(c)

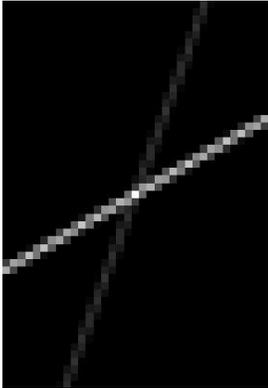
Figure B.1: True images, observed images and blurring kernels for cartoon image



(a)

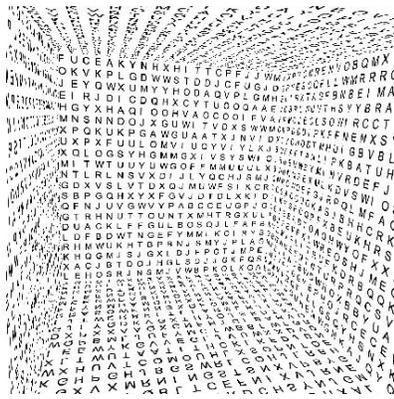


(b)

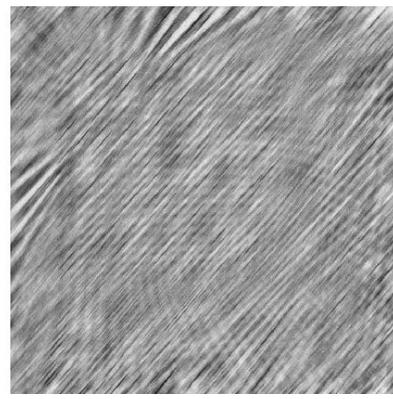


(c)

Figure B.2: True images, observed images and blurring kernels for vector image



(a)



(b)



(c)

Figure B.4: True images, observed images and blurring kernels for cube image

Appendix C

Additional Results

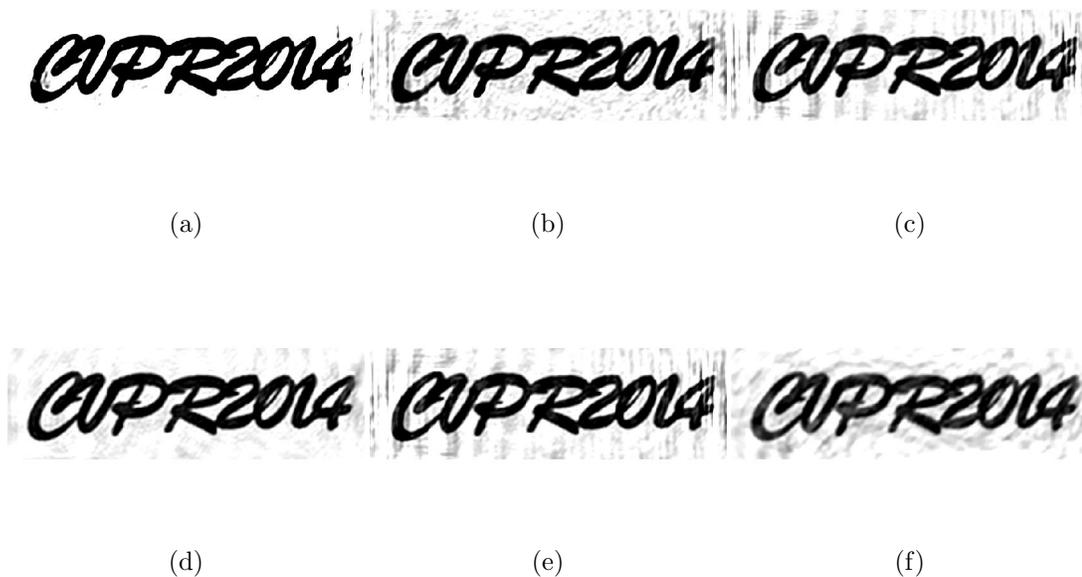


Figure C.1: (a)Our method. (b)Lucy-Richardson Algorithm [1, 2]. (c)Total Variation [3]. (d)l0 regularized model [4]. (e) Hyper-laplacian priors [5]. (f) [6].

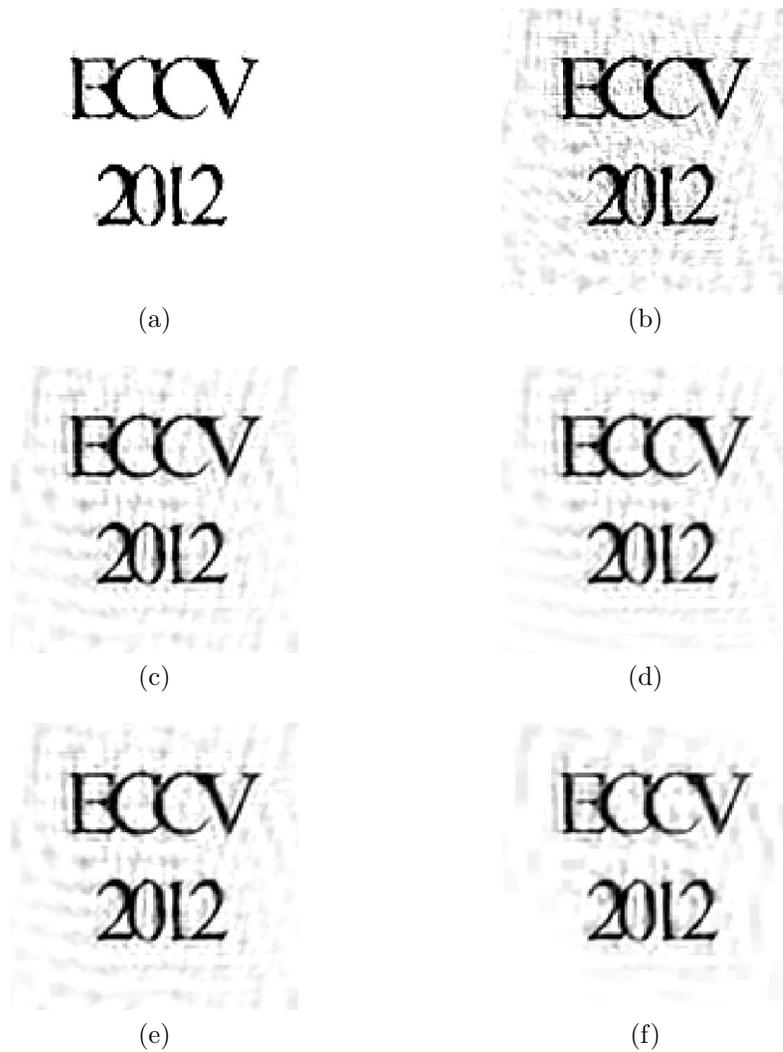


Figure C.2: (a)Our method. (b)Lucy-Richardson Algorithm [1, 2]. (c)Total Variation [3]. (d)l0 regularized model [4]. (e) Hyper-laplacian priors [5]. (f) [6].

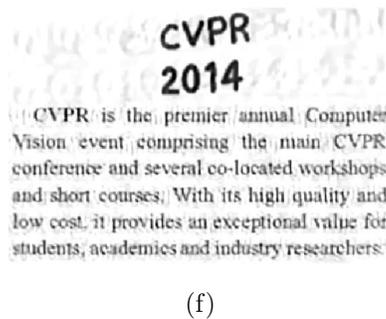
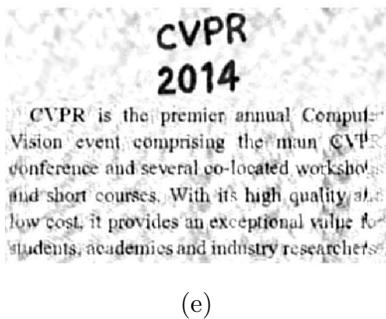
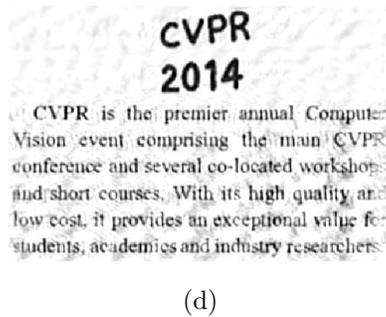
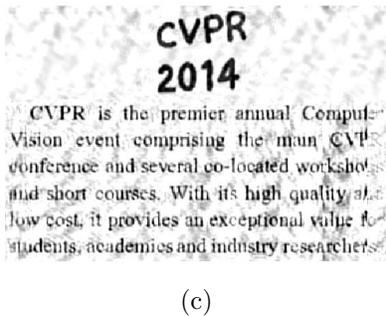
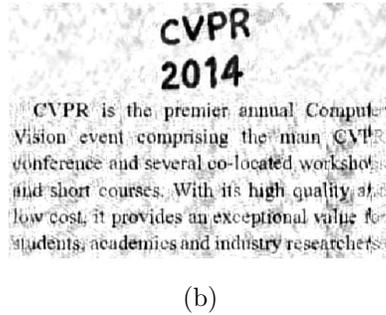
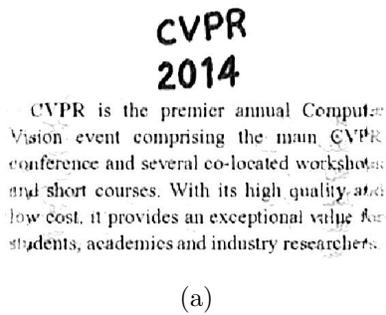


Figure C.3: (a)Our method. (b)Lucy-Richardson Algorithm [1,2]. (c)Total Variation [3]. (d)l0 regularized model [4]. (e) Hyper-laplacian priors [5]. (f) [6].



(a)



(b)



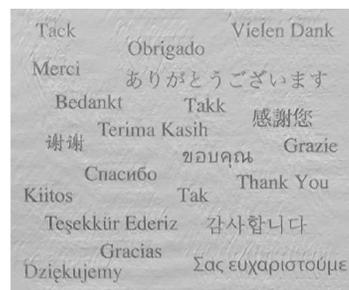
(c)



(d)



(e)



(f)

Figure C.4: (a)Our method. (b)Lucy-Richardson Algorithm [1,2]. (c)Total Variation [3]. (d)l0 regularized model [4]. (e) Hyper-laplacian priors [5]. (f) [6].

Appendix D

Effects of Regularization Terms



Figure D.1: Estimated solution (a)Without ϕ (b)With ϕ

The regularizer with weight ϕ eliminates the ringing noise that is observed near text parts. Using the regularizer eliminates some of the large noise blocks, leading to smoother images without causing a loss in image detail.



Figure D.2: Estimated solution (a)Without ν (b)With ν

The regularizer with weight ν eliminates the noise by enhancing each pixel using its neighbors. However, it also eliminates some of the details in the image. For this reason, we use it with ϕ term to balance between denoised and detailed image.



Figure D.3: Estimated solution (a)Without μ (b)With μ

The regularizer with weight μ pursues to restore some of the details lost due to noise, by satisfying the sparsity property of text images. It strengthens the details of the text by providing solutions with less disconnected pixels.

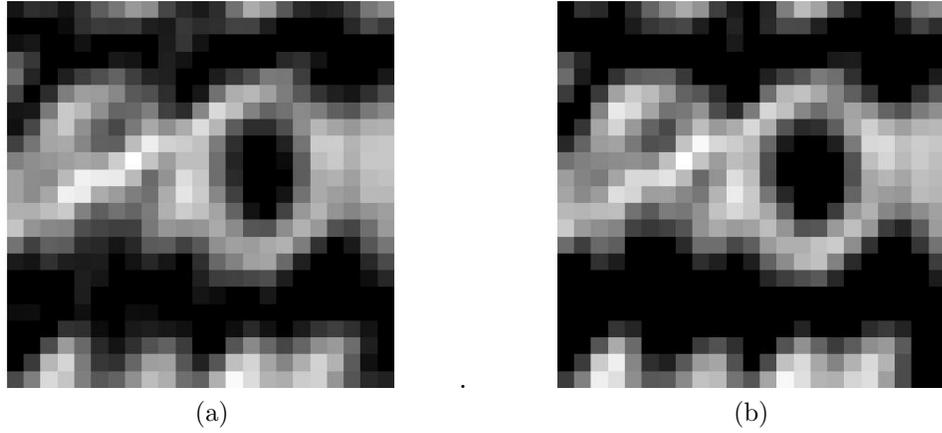


Figure D.4: Estimated solution (a)Without γ (b)With γ

The regularizer with weight γ is most useful for text images with black backgrounds. It strengthens the intensity of the background, which allows the resulting image to have a better readability of the text.



Figure D.5: Estimated solution (a)Without ζ (b)With ζ

The regularizer with weight γ strengthens the intensity of a white background. This not only reduces noise, but also brings out the text details by increasing the difference between text and background pixels.



Figure D.6: Estimated solution (a)Without η (b)With η

The regularizer with weight γ separates the black and white pixels iteratively. It leads to solutions where the text and background are further separated. It results in solutions that have a more uniform background and greater text details.