Fast and Efficient Algorithms for TV Image Restoration

LIANG, Haixia

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy
in
Mathematics

The Chinese University of Hong Kong
September 2010

Thesis/Assessment Committee

Professor BLU Thierre (Chair)

Professor CHAN Hon Fu Raymond (Thesis Supervisor)

Professor AU Kwok Keung Thoms (Committee member)

Professor NG Michael (External Examiner)

Abstract of thesis entitled

Fast and Efficient Algorithms for TV Image Restoration

In this thesis, we study two aspects in image processing. Part I is on the fast and efficient algorithms for the TV-L1 image restoration. Part II is on the fast and efficient algorithms for the positively constraint maximum penalized TV image restoration.

In Part I of the thesis, we focus on the fast and efficient algorithms for the TV-L1 minimization problem which can be applied to recover the blurred images corrupted by impulse noise. We construct the half-quadratic algorithm (HQA) for TV-L1 image restoration based on the half-quadratic technique. By introducing the proximal point algorithm into the HQA, we then obtain a modified HQA. We call it the proximal point half-quadratice algorithm (PHA). We introduce the PHA aiming to decrease the condition number of the coefficient matrix as updating the iterator in HQA. Until recently, there have been many efficient methods to solve the TV-L1 minimization problem. Examples are the primal-dual method, the fast total variational deconvolution method (FTVDM), and the augmented Lagrangian method (ALM). By numerical results of the FTVDM and ALM, we see that the images restored by these methods may sometimes appear to be blocky. Come back to our methods. The HQA and the PHA are both fast and efficient algorithms to solve the TV-L1 minimization problem. We prove that our algorithms are both majorize-minimize algorithms for solving a regularized TV-L1 problem. Given the assumption $\ker(\nabla) \cap \ker(B^T B) = \{\mathbf{0}\}$, the convergence and linear convergence of the HQA is then easily obtained. Without such an assumption, a convergence result of PHA is also obtained. We apply our algorithms to deblur images corrupted with impulse noise. The results show that the HQA is faster and more accurate than the ALM and FTVDM for salt-and-pepper noise

and comparable to the two methods for random-valued impulse noise. The PHA is comparable to the HQA in both recovered effect and computing consuming. Comparing with ALM and FTVDM, the PHA is faster and more accurate than ALM and FTVDM for salt-and-pepper noise and comparable to the two methods for random-valued impulse noise. Furthermore, the recovered images by the HQA and the PHA are less blocky.

Part II of the thesis focuses on the positively constraint maximum penalized total variation image restoration. We develop and implement a multiplicative iteration approach for the positively constrained total variation image restoration. We call our algorithm MITV. The MITV algorithm is based on the multiplicative iterative algorithm originally developed for tomographic image reconstruction. The advantages of the MITV are that it is very easy to derive and implement under different image noise models and it respects the positivity constraint. Our method can be applied to kinds of noise models, the Gaussian noise model, Poisson noise model and the impulse noise model. In numerical test, we apply our algorithm to deblur images corrupted with Gaussian noise. The results show that our method give better restored images than the forward-backward splitting algorithm.

香港中文大學

數學哲學博士論文

基於變分原理的快速有效的圖像重構方法

梁海霞

2010 年 9 月

摘要

在這篇論文中，我們研究圖像處理中的兩大類問題。第一部分是關於 TV-L1 圖像重構模型的快速有效方法。第二部分是基於變分原理圖像處理的正約束問題的方法研究。

在第一部分中，我們研究 TV-L1 最優化模型的快速求解方法，這一模型成功地應用於脈沖噪音下的圖像去噪和去模糊。我們考慮半二次方法快速有效地求解基於 TV-L1 模型的圖像重構問題。進而通過近點算法改進半二次方法。我們稱這類改進的方法為近點半二次法。引入近點算法的目的是在於盡可能減小半二次方法迭代過程中線性方程組係數矩陣的條件數，從而使得此係數矩陣盡可能良態，從而改進算法。 到目前為止，已存在很多有效求解 TV-L1 極小化問題的算法，例如原始對偶法，重疊合法，增廣拉格朗日法。此外，我們討論半二次法和進點半二次法收斂性和線性收斂性質。 在數值試驗中，我們給出半二次法，近點半二次法同重疊合法，增廣拉格朗日法的比較結果。基於此，我們會在一個章節中首先簡要介紹重疊合法和增廣拉格朗日法。 在數值試驗中，我們應用我們的算法和重疊合法，以及增廣拉格朗日法處理脈沖噪音下去噪和去模糊問題，結果顯示，對椒鹽噪音，半二次法和進點半二次法比重疊合法和進點半二次法更快速更精確。 對隨機值噪音，他們結果類似。此外，由重疊合法和近點半二次法恢復的圖像較為分片常數的。 由半二次法和進點半二次法恢復的圖像更加自然。

在第二部分中，我們研究基於變分原理的圖像處理的正約束問題的快速方法。我們構造增殖迭代法求解基於變分原理的圖像重構問題。這一方法是在用於解決體層影像重構問題的增殖迭代法基礎上提出的。增殖方法的一個很大優點在於，即便是對於不同的模型，推導和實現都非常容易，並且迭代過程滿足

正約束條件。這一方法可以廣泛地應用於高斯噪音模型，泊松噪音模型和脈沖噪音模型。數值試驗中，我們應用我們的方法來處理高斯噪音下的去噪和去模糊問題，結果顯示，我們的方法比向前向後交替迭代法能得到更好的重構結果。

ACKNOWLEDGMENTS

First and foremost, I wish to express my deepest gratitude to my supervisor, Professor Raymond H. Chan. His encouragement, support and advice have been immensely valuable, both in personal and professional terms. I am particularly grateful to him for his influence over me by the attitude towards research and teaching, for his emphasis on simplicity and elegance in research, and for the genuine concern he has shown for my development to be an academic.

I also indebted to Prof. I-Liang Chern, National Taiwan University, Prof. Michael Hintermüller, Humboldt-University of Berlin, for their enthusiasm, constant counsel and support throughout my Ph.D. studies. I am grateful for their visiting CUHK when I learn a lot from them. I am specially grateful to Prof. Er-xiong Jiang, Shanghai University, my supervisor advising me for my master degree recommending me to Prof. Chan to continue my study for Ph. D. degree.

My thanks are also due to my collaborators Dr. Jun Ma, Macquarie University, Australia, for his beneficial contribution on part II of the thesis. In addition, I would like to thank Prof. Ke Chen, Prof. Tieyong, Zeng, Dr. Pierre Weiss, Dr. Yiqiu Dong, Dr. Haiwei Sun, Prof. Qianshun, Chang, Dr. Jianfeng, Cai, Dr. Xiaoqun, Zhang and Prof. Eric Zhung for their helpful discussion, encouragement, generous support and sincere care. Thank Chi kin, Tai for his help in MATLAB programming and helpful discussion.

Finally, I thank my parents, brother and sisters for their encouraging support, which enables me to devote my energy to my thesis and finish my Ph.D. study.

Contents

1 Introduction **12**

1.1 Introduction to Part I 13

1.1.1 Introduction to Chapter 2 on the Fast Total Variation Method and the Augmented Lagrangian Method for TV-L1 Image Restoration 13

1.1.2 Introduce to Chapter 3 on the Half-Quadratic Algorithm for TV-L1 Image Restoration . . . 14

1.1.3 Introduction to Chapter 4 on the Proximal Point Algorithm for TV-L1 Image Restoration 15

1.2 Introduction to Part II 16

1.2.1 Introduction to Chapter 5 on the Positive Constrained Penalized Total Variation Image Restoration 16

2 The FTVDM and the ALM for TV-L1 Minimization Problems **17**

2.1 The FTVDM for the TV-L1 Minimization Problem . 18

2.2 The ALM for the TV-L1 Minimization Problem . . . 19
2.3 Relationship Between the Shrinkage and the Huber Function . . . 20
2.4 The application of the Shrinkage and the Huber Function in the FTVDM and the ALM . . . 22

3 A Fast and Efficient Half-Quadratic Algorithm for TV-L1 Image Restoration 24
3.1 Introduction . . . 24
3.2 The Derivation of the HQA . . . 26
3.3 Convergence of HQA . . . 28
3.4 Numerical Examples . . . 37
3.5 Conclusions . . . 41

4 A Proximal Point Half-Quadratic Algorithm for TV-L1 Image Restoration 54
4.1 Introduction . . . 54
4.2 The Derivation of the PHA . . . 56
4.3 Convergence of PHA . . . 57
4.4 Numerical Examples . . . 59
4.5 Conclusions . . . 62

5 Positively Constrained Minimum Penalized Total Variation Image Restoration 68
5.1 Introduction . . . 68

5.2 The Forward-Backward Splitting Algorithm 72
5.3 Multiplicative Iterative TV Penalized Image Restoration 73
5.3.1 Derivation of the Algorithm 73
5.3.2 MITV Under Different Image Noise Models . 77
5.4 Numerical Examples 79
5.5 Conclusions 82

Bibliography **87**

Chapter 1

Introduction

"No time in human history has ever witnessed such explosive influence and impact of image processing on modern society, science, and technologies. From nanotechnologies, astronomy, medicine, vision psychology, remote security screening, and the entertainment industry to digital communication technologies, image have helped mankind to see objects in various environments and scales, to sense and communicate distinct spatial or temporal patterns of the physical world, as well as to make optimal decisions and take right actions. Image processing and understanding are therefore turning into a critical component in processing and understanding are therefore turning into a critical component in contemporary sciences and technologies with many important applications."

– Tony F. Chan and Jianhong (Jackie) Shen in [13]

In this thesis, we study two aspects in image processing. Part I is on the fast and efficient algorithms for the TV-L1 minimization problem that has been successfully applied in deblurring images corrupted by impulsive noise. Part II is on the multiplicative iterative algorithm for positively constraint maximum penalized total variation image restoration.

1.1 Introduction to Part I

1.1.1 Introduction to Chapter 2 on the Fast Total Variation Method and the Augmented Lagrangian Method for TV-L1 Image Restoration

Total variation (TV) regularization was first introduced in [45]. It has been demonstrated successfully in image restoration because of its good property in preserving edges. Usually, the TV minimization is considered with an L2 data fitting term, which is particularly suitable for recovering images corrupted by Gaussian noise. Replacing the L2 data fidelity term with an L1 data fidelity term offers us the TV-L1 minimization model. In many important data, the noise may not obey Gaussian distribution. The impulse noise is one of the examples. Impulse noise is usually generated by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or erroneous transmission [3]. It has two common types, salt-and-papper noise and random-valued noise. The salt-an-pepper (or random-valued) noise corrupts the a portion of the image pixels with minimal or maximal (or random-valued) intensities while keeping the other pixels unaffected. It is difficult to remove such kind of noise, since the corrupted pixels are randomly distributed in the image and the intensities at the corrupted pixels are usually distinguishable from those of their neighbors. The TV-L2 minimization model is not suitable to recover the images corrupted by impulse noise. Compared with the TV-L2 model, the TV-L1 model uses a non-smooth fidelity which has great advantage in impulse noise removal [39, 40]. It is shown that the L1 fidelity can fit uncorrupted pixels exactly and regularize the corrupted pixels perfectly. Until recently, there have been many efficient methods to solve the TV-L1 minimization problems. Examples are the primal-dual method, the fast total variational deconvolution method (FTVDM), and the augmented Lagrangian method (ALM).

In Chapter 2, we introduce the FTVDM and ALM briefly, since the numerical

comparison with the FTVDM and the ALM will be given in Chapter 3 and 4.

The FTVDM can either derived from the classical quadratic penalty function technique in optimization [17], or from the half-quadratic technique initially proposed in [22]. The FTVDM is indeed an alternating minimization algorithm with attractive convergence properties, which include the global convergence with a strong q-linear rate and convergence for some auxiliary variables. Under periodic boundary conditions, its computation can take advantages of simple high-dimentional shrinkage and fast Fourier transform (FFT).

The ALM is a fast and efficient algorithm for TV regularization with non-quadratic data fidelity function. It is based on the augmented Lagrangian method. The ALM can not only be applied to solve the TV-L1 minimization problem, but also be applied to deblur images corrupted by Poisson noise. Under periodic boundary conditions, its computation can also take advantage of simple high-dimentional shrinkage and FFT. It has been the fastest and the most efficient method until recently.

However, the images restored by the FTVDM and the ALM may sometimes appear to be blocky.

Chapter 2 is based on [52, 51]

1.1.2 Introduce to Chapter 3 on the Half-Quadratic Algorithm for TV-L1 Image Restoration

In Chapter 3, we propose a fast and efficient algorithm for TV-L1 minimization problem based on the half-quadratic (HQ) technique [22]. Considering that the TV-L1 functional is non-smooth functional, we begin our algorithm by smoothing the objective functional. Then the half-quadratic technique [22] is applied to the smooth functional to construct our HQA. We prove that the HQA is indeed a majorize-minimize algorithm for a regularized TV-L1 problem and hence the convergence and linear convergence of the HQA is easily obtained. We apply our

algorithm to deblur images that are corrupted with impulse noise The results show that our method is faster and more accurate than ALM and FTVDM for salt-and-pepper noise and comparable to the two methods for random-valued impulse noise Furthermore, the recovered images by the HQA are less blocky Chapter 3 is based on the paper

R Chan and H Liang, *A fast and efficient half-quadratic algorithm for TV-L1 image restoration*, submitted, 2010

1.1.3 Introduction to Chapter 4 on the Proximal Point Algorithm for TV-L1 Image Restoration

In Chapter 3, we design a fast and efficient half-quadratic algorithm to solve the TV-L1 minimization problem The convergence and linear convergence of HQA are discussed under the assumption that $\ker(\Delta)\cap\ker(B^TB) = \{\mathbf{0}\}$, where Δ is the Laplacian operator and B is the blurring operator Considering that the condition number of the coefficient matrix as updating the iteration in HQA may be very large in the absence of $\ker(\Delta)\cap\ker(B^TB) = \{\mathbf{0}\}$ In Chapter 4, we modify the HQA with the proximal-point algorithm (PPA) [43] such that the condition number of the coefficient matrix in HQA is small We call the modified HQA the proximal point half-quadratic algorithm (PHA) We prove that the PHA is also a majorize-minimize algorithm (MMA) as the HQA The convergence of PHA is obtained in the absence of the above assumption We apply the PHA to deblur images corrupted by impulse noise The numerical results show that the PHA is comparable to the HQA in both recovered effect and computing time consuming Comparing with the ALM and the FTVDM, the PHA is faster and more accurate than the ALM and FTVDM for salt-and-pepper noise and comparable to the two methods for random-valued impulse noise Furthermore, the recovered images by the PHA are less blocky as the HQA, while the recovered images by the ALM and FTVDM are blocky Chapter 3 is based on the paper

R. Chan and H. Liang, *A proximal point half-quadtatic algorithm for TV-L1 image restoration*, Preprint.

1.2 Introduction to Part II

1.2.1 Introduction to Chapter 5 on the Positive Constrained Penalized Total Variation Image Restoration

In Chapter 5, we also discuss the total variation model widely used in image processing. Notice that the 8-bit gray-scale images are usually constrained into $[0, 255]$ at every pixel. However, the most existing approaches for total variation image restoration lack of considering the positivity constraint. To obtain the recovered image in $[0, 255]$, in practice, one may solve the unconstrained problem first and then project or scale the solution into the dynamic range . However, for images with large zero backgrounds, such kind of projection methods may not be efficient enough.

In Chapter 5, we develop and implement a new approach for total variation image restoration. Our method is based on the multiplicative iterative algorithm originally developed for tomographic image reconstruction. The advantages of our algorithm are that it is very easy to derive and implement under different image noise models and it respects the positivity constraint. Our method can be applied to kinds of noise models, the Gaussian noise model, Poisson noise model and the impulse noise model. In numerical test, we apply our algorithm to deblur images corrupted with Gaussian noise. The results show that our method give better restored images than the forward-backward splitting algorithm. Chapter 4 is based on the paper

R. Chan, H. Liang, and J. Ma, *Positively constraint minimum penalized total variation image restoration*, submitted, 2010.

Chapter 2

The FTVDM and the ALM for TV-L1 Minimization Problems

In this chapter, we introduce the FTVDM and the ALM briefly. The FTVDM can either be derived from the classical quadratic penalty function technique in optimization [17], or from the half-quadratic technique initially proposed in [22]. The FTVDM is indeed an alternating minimization algorithm with attractive convergence properties, which include the global convergence with a strong q-linear rate and convergence for some auxiliary variables. Its computation can take advantages of simple high-dimentional shrinkage and take advantage of FFT under periodic boundary conditions.

The ALM is a fast and efficient algorithm for TV regularization with non-quadratic data fidelity function. It is based on the augmented Lagrangian method. The ALM can not only be applied to solve the TV-L1 minimization problem, but also be applied to Poisson noise model. The same to the FTVDM, its computation can also take advantages of the simple high-dimentional shrinkage and take advantage of the FFT under the periodic boundary conditions. It has been the fastest and the most efficient method until recently.

2.1 The FTVDM for the TV-L1 Minimization Problem

In this section, we introduce the FTVDM for the TV-L1 minimization problem. The TV-L1 minimization problem is the problem of finding $\mathbf{u}$ from $\mathbf{f}$ in

$$\min_{\mathbf{u}}\{\lambda\|\mathbf{u}\|_{TV} + \|B\mathbf{u} - \mathbf{f}\|_1\}, \tag{2.1}$$

where $\mathbf{u}$, $\mathbf{f} \in \mathbb{R}^n$ are column vectors concatenated from the original image and the observed image respectively, $B \in \mathbb{R}^{n\times n}$, and n is the number of pixels in the images, and $\|\mathbf{u}\|_{TV} = \||\nabla\mathbf{u}|\|_1 = \sum\sqrt{\nabla_x u_{i,j}^2 + \nabla_y u_{i,j}^2}$ is the isotropic total variation.

The FTVDM is to introduce two auxiliary variables to approximate $B\mathbf{u} - \mathbf{f}$ and $\nabla\mathbf{u}$ in the nondifferentiable norms in (2.1), respectively. By adding the quadratic terms to penalize the difference between every pair of the original and auxiliary, the following approximate problem to (2.1) is obtained:

$$\min_{\mathbf{w},\mathbf{z},\mathbf{u}}\{\lambda(\||\mathbf{w}|\|_1 + \frac{\theta_{\mathbf{w}}}{2}\|\mathbf{w} - \nabla\mathbf{u}\|_2^2) + \|\mathbf{z}\|_1 + \frac{\theta_{\mathbf{z}}}{2}\|\mathbf{z} - (B\mathbf{u} - \mathbf{f})\|_2^2\} \tag{2.2}$$

where $\theta_{\mathbf{w}}, \theta_{\mathbf{z}} \gg 0$ are penalty parameters. The approximate problem (2.2) turns to (2.1) as $\theta_{\mathbf{w}}, \theta_{\mathbf{z}} \to \infty$. (2.2) is then solved by an iterative and alternating approach due to the fact that with any two of the three variables $\mathbf{w}, \mathbf{z}$, and $\mathbf{u}$ fixed, the minimizer of (2.2) with respect to the third one. It has closed-form formulas based on the 1-dimensional shrinkage for $\mathbf{z}$, 2-dimensional shrinkage for $\mathbf{w}$ and FFT for $\mathbf{u}$ under the periodic boundary condition. Here, 1-dimensional shrinkage is defined by

$$\begin{aligned}\mathcal{T}_\gamma(x) &= \arg\min_y\{|y| + \frac{1}{2\gamma}|y - x|^2\},\\ &= \text{sign}(x)\max(|x| - \gamma, 0), \quad \text{for } x \in \mathbb{R}, \gamma > 0\end{aligned} \tag{2.3}$$

and 2-dimensional shrinkage is defined by

$$\begin{aligned}\mathcal{T}_\gamma(\mathbf{x}) &= \arg\min_{\mathbf{y}\in\mathbb{R}^2}\{\|\mathbf{y}\|_2 + \frac{1}{2\gamma}\|\mathbf{y}-\mathbf{x}\|_2^2\},\\ &= \begin{pmatrix} \max(\|\mathbf{x}\|_2-\gamma,0)\frac{x_1}{\|\mathbf{x}\|_2} \\ \max(\|\mathbf{x}\|_2-\gamma,0)\frac{x_2}{\|\mathbf{x}\|_2}\end{pmatrix}, \quad \text{for } x\in\mathbb{R}^2, \gamma>0. \end{aligned} \tag{2.4}$$

In addition, the algorithm is analyzed and shown to have attractive convergence properties, which include global convergence with strong q-linear rate and finite convergence for auxiliary variables. The interested readers can consult [52] for more details on the algorithm and convergence proof.

2.2 The ALM for the TV-L1 Minimization Problem

In this section, we introduce the ALM for the TV-L1 Minimization problem. It is using the augmented Lagrangian method to solve the TV regularization problem with the L1 fidelity term.

First, two new variables $\mathbf{w}$ and $\mathbf{z}$ are introduced and the problem (2.1) is then reformulated to the following constrained optimization problem

$$\begin{cases} \min\limits_{\mathbf{u},\mathbf{w},\mathbf{z}}\{\lambda\||\mathbf{w}|\|_1 + \|\mathbf{z}-\mathbf{f}\|_1\} \\ s.t. \mathbf{w} = \nabla\mathbf{u},\ \mathbf{z} = B\mathbf{u}\end{cases} \tag{2.5}$$

To solve (2.5), the following augmented Lagrangian functional

$$\begin{aligned}&\mathcal{L}(\mathbf{u},\mathbf{w},\mathbf{z};\lambda_\mathbf{w},\lambda_\mathbf{z})\\ =\ &\lambda(\||\mathbf{w}|\|_1 + \langle\lambda_\mathbf{w}, \mathbf{w}-\nabla\mathbf{u}\rangle + \frac{\gamma_\mathbf{w}}{2}\|\mathbf{w}-\nabla\mathbf{u}\|_2^2)\\ &+\|\mathbf{z}-\mathbf{f}\|_1 + \langle\lambda_\mathbf{z}, \mathbf{z}-B\mathbf{u}\rangle + \frac{\gamma_\mathbf{z}}{2}\|\mathbf{z}-B\mathbf{u}\|_2^2,\end{aligned} \tag{2.6}$$

with Lagrangian multipliers $\lambda_\mathbf{w}$ and $\lambda_\mathbf{z}$ and positive constants $\gamma_\mathbf{w}$ and $\gamma_\mathbf{z}$ is developed. Then the alternating minimization algorithm is used to solve

$$(\mathbf{u}^k,\mathbf{w}^k,\mathbf{z}^k) \approx \arg\min_{\mathbf{u},\mathbf{w},\mathbf{z}} \mathcal{L}(\mathbf{u},\mathbf{w},\mathbf{z};\lambda_\mathbf{w}^k,\lambda_\mathbf{z}^k), \tag{2.7}$$

with updating

$$\begin{aligned} \lambda_{\mathbf{w}}^{k+1} &= \lambda_{\mathbf{w}} + \gamma_{\mathbf{w}}(\mathbf{w}^k - \nabla \mathbf{u}^k), \\ \lambda_{\mathbf{z}}^{k+1} &= \lambda_{\mathbf{z}}^k + \gamma_{\mathbf{z}}(\mathbf{z}^k - B\mathbf{u}^k). \end{aligned}$$

In solving (2.7), the 1-dimensional shrinkage (2.3) is applied for updating $\mathbf{z}$ and 2-dimensional shrinkage (2.4) is used for updating $\mathbf{w}$. The same to the FTVDM, under periodic boundary conditions, the updating for $\mathbf{u}$ can also take advantages of the FFT. The introduction in detail to the ALM, its application to the Poisson noise removal and the convergence analysis can be found in [51]. The interested readers can consult [51] for more information.

With the high-dimension shrinkage, the fundamental ideas of the FTVDM and the ALM are both to solve the nondifferentiable minimization problem (2.1) by solving the approximated differentiable ones at each iteration. The differentiable functions are actually the Huber function corresponding to the nondifferentiable ones at each step. The Huber function defined in one dimension is of the following form:

$$\phi_\epsilon(t) = \begin{cases} \frac{1}{2\epsilon}t^2, & \text{if } |t| \leq \epsilon \\ |t| - \frac{\epsilon}{2}, & \text{otherwise} \end{cases} \tag{2.8}$$

which is usually used to approximate $\phi(t) = |t|, t \in \mathbb{R}$ with small ϵ . It has been used to approximate the anisotropic TV function. For isotropic TV function, the Huber function $\phi_\epsilon(\|\mathbf{t}\|_2)$ with respect to $|\mathbf{t}|$ is used to approximate $\|\mathbf{t}\|_2$, where $\phi(\cdot)$ is as defined in (2.8), $\mathbf{t} = (t_1, t_2)^T \in \mathbb{R}^2$ and $\|\mathbf{t}\|_2 = \sqrt{t_1^2 + t_2^2}$.

2.3 Relationship Between the Shrinkage and the Huber Function

In this section, we discuss the relationship between shrinkage and Huber function. With simple derivation, we show that the 1-dimensional (2-dimensional) shrinkage

is indeed the minimizer of the 1-dimensional (2-dimensional) Huber function.

We first consider the 1-dimensional case. In (2.3), denote

$$f(x) = \min_y \{|y| + \frac{1}{2\gamma}|y - x|^2\}. \tag{2.9}$$

We rewrite the 1-dimensional shrinkage $\mathcal{T}_\gamma(x)$ as

$$\mathcal{T}_\gamma(x) = \begin{cases} 0, & |x| \leq \gamma \\ \text{sign}(x)(|x| - \gamma). & \text{otherwise} \end{cases}$$

Substituting $\mathcal{T}_\gamma(x)$ into (2.9), we have $f(x) = \frac{1}{2\gamma}x^2$ as $|x| \leq \gamma$; and $f(x) = |x| - \frac{\gamma}{2}$ as $|x| > \gamma$, which is just the Huber function $\phi_\gamma(x)$.

We rewrite the 2-dimensional shrinkage into the following form:

$$\mathcal{T}_\gamma(\mathbf{x}) = \begin{cases} \mathbf{0}, & \|\mathbf{x}\|_2 \leq \gamma \\ \frac{\|\mathbf{x}\|_2 - \gamma}{\|\mathbf{x}\|_2}(x_1, x_2)^T. & \|\mathbf{x}\|_2 > \gamma \end{cases}$$

We substitute $\mathcal{T}_\gamma(\mathbf{x})$ into

$$g(\mathbf{x}) = \min_{\mathbf{y}} \{\|\mathbf{y}\|_2 + \frac{1}{2\gamma}\|\mathbf{y} - \mathbf{x}\|_2^2\},$$

then we have

$$g(\mathbf{x}) = \frac{1}{2\gamma}\|\mathbf{x}\|_2^2, \text{ as } \|\mathbf{x}\|_2 \leq \gamma$$

As $\|\mathbf{x}\|_2 \geq \gamma$, we have

$$g(\mathbf{x}) = \|\mathbf{x}\|_2 - \gamma + \frac{\gamma^2}{2\gamma} = \|\mathbf{x}\|_2 - \frac{\gamma}{2}.$$

It is the 2-dimensional Huber function $\phi_\gamma(\|\mathbf{x}\|_2)$.

With such a relationship between the shrinkage and the Huber function, we will discuss the applications of the shrinkage and the Huber function in the FTVDM and the ALM in next section.

2.4 The application of the Shrinkage and the Huber Function in the FTVDM and the ALM

In this section, we analyze the applications of the shrinkage and the Huber function in the FTVDM and ALM. By discussion on the Huber function and the high-dimensional shrinkage in §2.1 and §2.3, (2.2) can be rewritten to be the following form with the Huber function:

$$\min_{\mathbf{u}}\{\sum_{i=1}^{n}\lambda\phi_{1/\theta_{\mathbf{w}}}(\|\nabla u_i\|_2)+\phi_{1/\theta_{\mathbf{z}}}(B_i\mathbf{u}-f_i)\},$$

if TV is isotropic. If TV is taken to be anisotropic, then (2.2) can be rewritten as

$$\min_{\mathbf{u}}\{\sum_{i=1}^{n}\lambda(\phi_{1/\theta_{\mathbf{w}}}(\nabla_x u_i)+\phi_{1/\theta_{\mathbf{w}}}(\nabla_y u_i))+\phi_{1/\theta_{\mathbf{z}}}(B_i\mathbf{u}-f_i)\},$$

It is to use the Huber function to approximate non-differentiable TV and L1 norms.

For the ALM, introducing the Huber function into (2.7), we then have

$$\begin{aligned}&\min_{\mathbf{u},\mathbf{w},\mathbf{z}}\mathcal{L}(\mathbf{u},\mathbf{w},\mathbf{z};\lambda_{\mathbf{w}},\lambda_{\mathbf{z}})\\ =\ &\min_{\mathbf{u}}\sum_{i=1}^{n}(\phi_{1/\gamma_{\mathbf{w}}}(\|\nabla u_i-\frac{1}{\gamma_{\mathbf{w}}}(\lambda_{\mathbf{w}})_i\|_2)+\phi_{1/\gamma_{\mathbf{z}}}(B_i\mathbf{u}-f_i-\frac{1}{\gamma_{\mathbf{z}}}(\lambda_{\mathbf{z}})_i)+C(\gamma_{\mathbf{w}},\gamma_{\mathbf{z}}),\end{aligned}$$

if TV is isotropic, where $C(\gamma_{\mathbf{w}},\gamma_{\mathbf{z}})$ is a constant with respect to $\mathbf{u}$. If TV is anisotropic, then

$$\begin{aligned}&\min_{\mathbf{u},\mathbf{w},\mathbf{z}}\mathcal{L}(\mathbf{u},\mathbf{w},\mathbf{z};\lambda_{\mathbf{w}},\lambda_{\mathbf{z}})\\ =\ &\min_{\mathbf{u}}\sum_{i=1}^{n}(\phi_{1/\gamma_{\mathbf{w}}}(\nabla_x u_i-\frac{1}{\gamma_{\mathbf{w}}}(\lambda_{1,\mathbf{w}})_i)+\phi_{1/\gamma_{\mathbf{w}}}(\nabla_y u_i-\frac{1}{\gamma_{\mathbf{w}}}(\lambda_{2,\mathbf{w}})_i)\\ &+\phi_{1/\gamma_{\mathbf{z}}}(B_i\mathbf{u}-f_i-\frac{1}{\gamma_{\mathbf{z}}}(\lambda_{\mathbf{z}})_i)+C(\gamma_{\mathbf{w}},\gamma_{\mathbf{z}}),\end{aligned}$$

where $\lambda_{\mathbf{w}}=[\lambda_{1,\mathbf{w}},\lambda_{2,\mathbf{w}}]$. The same to the FTVDM, it is also to use the Huber function to approximate TV and L1 data fitting.

According to the above discussions, the FTVDM and the ALM are both to use the Huber functions to approximate the nondifferentible TV regularization function and the L1 data fitting function. In the FTVDM and the ALM respectively a sequence is generated by solving an approximate differentiable minimization problem which converges to the solution of the nondifferentiable problem (2.1).

Chapter 3

A Fast and Efficient Half-Quadratic Algorithm for TV-L1 Image Restoration

3.1 Introduction

For the convenience of reading, we repeat the TV-L1 minimization problem here:

$$\min_{\mathbf{u}}\{\lambda\|\mathbf{u}\|_{TV} + \|B\mathbf{u} - \mathbf{f}\|_1\}. \tag{3.1}$$

In (3.1), we rewrite the regularization term

$$\|\mathbf{u}\|_{TV} := \sum_{i=1}^{n} \sqrt{[(A_1)_i\mathbf{u}]^2 + [(A_2)_i\mathbf{u}]^2},$$

where A_1 and A_2 represent the finite difference operator in the x and y-direction respectively, and $(A_j)_i$ is the ith row of A_j. Note that $(A_1)_i\mathbf{u}$ and $(A_2)_i\mathbf{u}$ are the x and y-directional finite difference at the i-th component u_i of $\mathbf{u}$. In the following, we denote $A := (A_1^T, A_2^T)^T \in \mathbb{R}^{2n\times n}$ and $\nabla u_i := ((A_1)_i\mathbf{u}, (A_2)_i\mathbf{u})^T$.

Many image processing problems, such as deblurring under impulse noise [51], image cartoon-texture decomposition [54], feature selection [54], multiscale

decomposition [55], and computer version [9], can be formulated as (3.1). Since $\|\mathbf{u}\|_{TV}$ can be regarded as the L1-norm of the gradient of $\mathbf{u}$, (3.1) can be regarded as a L1-regularized problem with a L1 fidelity term.

The TV regularization was first introduced in [45], and has been successfully applied to image restoration [6, 11, 5, 32, 2, 7, 33, 8, 56, 57] because of its good property in preserving edges. Traditionally, the squared L2 data fidelity $\frac{1}{2}\|B\mathbf{u} - \mathbf{f}\|_2^2$ is commonly used in conjunction with the TV regularization, see [47, 45, 49, 23]. Recently, however, there are many applications of TV-L1 model using the L1 data fidelity $\|B\mathbf{u} - \mathbf{f}\|_1$ term, see [9, 54, 14, 53, 55, 15, 51, 52]. The interested readers can consult [52] for more background information. Several fast methods for solving TV-L1 problems have been proposed recently. Examples are the primal-dual method [18], fast alternating minimization approach [24, 38] the fast total variational deconvolution method (FTVDM) [52], and the augmented Lagrangian method (ALM) [51]. The FTVDM and the ALM can be understood as methods for solving a smooth functional approximation of (3.1). Among these methods, the FTVDM and the ALM are the most efficient and the ALM is the fastest one, see numerical results in [51]. However, the images recovered by the ALM and the FTVDM can sometimes be blocky, see Figure 3.13 (left and middle).

In this chapter, we propose a fast and efficient method for solving (3.1) that is based on the half-quadratic technique [22]. Our method also begins by smoothing the energy functional in (3.1) as in the FTVDM and ALM. More precisely, since TV and L1 functionals are not differentiable, we replace (3.1) by

$$\min_{\mathbf{u}} \left\{ \sum_{i=1}^{n} \lambda \sqrt{[(A_1)_i\mathbf{u}]^2 + [(A_2)_i\mathbf{u}]^2 + \beta} + \sqrt{(B_i\mathbf{u} - f_i)^2 + \gamma} \right\}, \qquad (3.2)$$

where B_i is the ith row of B, and β, γ are both small regularization parameters. The approximate problem (3.2) turns to (2.1) as $\beta, \gamma \to 0$. This smoothing idea has been used in [9] for TV-L1 denoising problem (i.e. when B is the identity matrix), where the smoothed functional is minimized by an explicit gradient

descent scheme.

Here we solve (3.2) by applying the half-quadratic regularization approach [22]. We call the algorithm HQA. We will prove that HQA is indeed a majorize-minimize algorithm [12, 30, 29] for solving (3.2). Hence the global convergence and the convergence rate of HQA can easily be established based on the convergence analysis in [12]. To test our algorithm, we apply it to deblur images that are corrupted by impulse noise [51]. The results show that our method is faster and more accurate than ALM and FTVDM for salt-and-pepper noise and comparable to the two methods for random-valued impulse noise. Furthermore, the recovered images by our method are less blocky.

The outline of the chapter is as follows: In §3.3, we review the half-quadratic approach, and then apply it to (3.2) to derive HQA. In §3.4, we prove that the HQA is indeed a majorize-minimize algorithm. Hence based on the convergence analysis in [12], the global convergence and the linear convergence rate of HQA are proven. Comparison with the ALM and the FTVDM on image deblurring under impulse noise is given in §3.5.

3.2 The Derivation of the HQA

The half-quadratic regularization approach [22] is equivalent to the gradient linearized iteration in [41]. It begins with the fact that, if $0 \neq t \in \mathbb{R}$, then

$$|t| = \min_{v>0}\{vt^2 + \frac{1}{4v}\}, \tag{3.3}$$

and the minimum value is at $v = \frac{1}{2|t|}$. Note that the function $vt^2 + 1/(4v)$ is quadratic in t but not in v and hence the name *half-quadratic*. Denote $|(x,y)^T|_\beta := \sqrt{x^2+y^2+\beta}$ and $|x|_\gamma := \sqrt{x^2+\gamma}$ for any $x, y \in \mathbb{R}$. Problem (3.2) can be written as

$$\min_{\mathbf{u}} \left\{ \sum_{i=1}^{n} \lambda |\nabla u_i|_\beta + |B_i\mathbf{u} - f_i|_\gamma \right\} := \min_{\mathbf{u}}\{\Phi(\mathbf{u})\}, \tag{3.4}$$

where $|\nabla u_i|_\beta = \sqrt{[(A_1)_i\mathbf{u}]^2 + [(A_2)_i\mathbf{u}]^2 + \beta}$. Using (3.3), equation (3.4) becomes

$$\begin{aligned} & \min_{\mathbf{u}} \left\{ \sum_{i=1}^{n} \lambda \min_{v_i>0}(v_i|\nabla u_i|_\beta^2 + \frac{1}{4v_i}) + \min_{w_i>0}(w_i|B_i\mathbf{u} - f_i|_\gamma^2 + \frac{1}{4w_i}) \right\} \\ = & \min_{\mathbf{u},\mathbf{v}>\mathbf{0},\mathbf{w}>\mathbf{0}} \left\{ \sum_{i=1}^{n} \left[\lambda(v_i|\nabla u_i|_\beta^2 + \frac{1}{4v_i}) + w_i|B_i\mathbf{u} - f_i|_\gamma^2 + \frac{1}{4w_i} \right] \right\} \qquad (3.5) \\ := & \min_{\mathbf{u},\mathbf{v}>\mathbf{0},\mathbf{w}>\mathbf{0}} \{\mathcal{L}(\mathbf{u},\mathbf{v},\mathbf{w})\}, \end{aligned}$$

where $\mathbf{v}, \mathbf{w} > 0$ means that all components of $\mathbf{v}, \mathbf{w}$ are greater than 0.

We apply the alternate minimization procedure to solve (3.6), namely

$$\mathbf{v}^{k+1} = \arg\min_{\mathbf{v}>0} \mathcal{L}(\mathbf{u}^k, \mathbf{v}, \mathbf{w}^k), \qquad (3.6)$$

$$\mathbf{w}^{k+1} = \arg\min_{\mathbf{w}>0} \mathcal{L}(\mathbf{u}^k, \mathbf{v}^{k+1}, \mathbf{w}), \qquad (3.7)$$

$$\mathbf{u}^{k+1} = \arg\min_{u} \mathcal{L}(\mathbf{u}, \mathbf{v}^{k+1}, \mathbf{w}^{k+1}). \qquad (3.8)$$

Applying (3.3) in (3.6), we see that (3.6) and (3.7) have explicit component minimizers

$$v_i^{k+1} = \frac{1}{2|\nabla u_i^k|_\beta} \quad \text{and} \quad w_i^{k+1} = \frac{1}{2|B_i\mathbf{u}^k - f_i|_\gamma}. \qquad (3.9)$$

Note that $\mathcal{L}(\mathbf{u}, \mathbf{v}^{k+1}, \mathbf{w}^{k+1})$ is continuous differentiable in $\mathbf{u}$. Hence $\mathbf{u}^{k+1}$ in (3.8) is the solution of

$$0 = \nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}, \mathbf{v}^{k+1}, \mathbf{w}^{k+1}) = \lambda A^T \tilde{D}_\beta(\mathbf{u}^k)A\mathbf{u} + B^T D_\gamma(\mathbf{u}^k)(B\mathbf{u} - \mathbf{f}). \qquad (3.10)$$

Here $\tilde{D}_\beta(\mathbf{u}^k) = \mathrm{diag}(D_\beta(\mathbf{u}^k), D_\beta(\mathbf{u}^k))$; and $D_\beta(\mathbf{u}^k)$ and $D_\gamma(\mathbf{u}^k) \in \mathbb{R}^{n\times n}$ are the diagonal matrices with their i-th diagonal entries being $2v_i^{k+1} = 1/|\nabla u_i^k|_\beta$ and $2w_i^{k+1} = 1/|B_i\mathbf{u}^k - f_i|_\gamma$ respectively. From (3.10) we obtain the linear system

$$\lambda A^T \tilde{D}_\beta(\mathbf{u}^k)A\mathbf{u} + B^T D_\gamma(\mathbf{u}^k)B\mathbf{u} = B^T D_\gamma(\mathbf{u}^k)\mathbf{f} \qquad (3.11)$$

for updating $\mathbf{u}^{k+1}$. Thus the algorithm of HQA is as follows:

(i) *Initialize* $\mathbf{u}^0$;

(ii) *For* $k = 1, 2, \cdots$ *until convergence, update* $\mathbf{u}^{k+1}$ by solving (3.11).

In Lemma 3.3.3 below, we show that if $\ker(A^TA) \cap \ker(B^TB) = \{\mathbf{0}\}$, then the coefficient matrix in the left hand side of (3.11) is indeed invertible and hence $\mathbf{u}^{k+1}$ is computable. We remark that this assumption is very general and usually satisfied. In fact,

(i) The matrix A^TA is the discrete Laplacian operator. Thus if the boundary condition for the finite difference is either periodic or reflective, then $\ker(A^TA)$ is spanned by $\mathbf{1}$, the vector of all ones.

(ii) If the matrix B is a blurring matrix, then it is a low pass filter, and hence $B^TB\mathbf{1} \neq \mathbf{0}$.

Hence, in these cases, $\ker(A^TA) \cap \ker(B^TB) = \{\mathbf{0}\}$ holds.

3.3 Convergence of HQA

In this section, we analyze the convergence of HQA based on the convergence analysis of the majorize-minimize algorithm (MMA) in [12]. Let us review the general idea of the MMA first.

The MMA optimization technique is to solve a minimization problem $\min_{\mathbf{u}} \Psi(\mathbf{u})$ by

$$\mathbf{u}^{k+1} = \arg\min_{\mathbf{u}}\{\mathcal{G}(\mathbf{u}, \mathbf{u}^k)\}, \tag{3.12}$$

where $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$, called a *tangent majorant function* of $\Psi(\mathbf{u})$ at $\mathbf{u}^k$, must satisfy

$$\mathcal{G}(\mathbf{u}, \mathbf{u}^k) \geq \Psi(\mathbf{u}), \qquad \forall\ \mathbf{u} \in \mathbb{R}^n, \tag{3.13}$$

$$\mathcal{G}(\mathbf{u}, \mathbf{u}^k) = \Psi(\mathbf{u}^k), \qquad \text{at } \mathbf{u} = \mathbf{u}^k, \tag{3.14}$$

$$\nabla_1\mathcal{G}(\mathbf{u}, \mathbf{u}^k) = \nabla\Psi(\mathbf{u}), \qquad \text{at } \mathbf{u} = \mathbf{u}^k. \tag{3.15}$$

Here, $\nabla_1\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ denotes the partial derivative with respect to the first component. Convergence analysis of MMAs can be found in [12, 30, 29]. For minimization problem (3.4), we define

$$\mathcal{G}(\mathbf{u}, \mathbf{u}^k) = \mathcal{L}(\mathbf{u}, \mathbf{v}^{k+1}, \mathbf{w}^{k+1}). \tag{3.16}$$

Substituting (3.9) into (3.16), we obtain the explicit form of $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$:

$$\mathcal{G}(\mathbf{u}, \mathbf{u}^k) = \sum_{i=1}^{n} \Big[\lambda\Big(\frac{|\nabla u_i|_\beta^2}{2|\nabla u_i^k|_\beta} + \frac{|\nabla u_i^k|_\beta}{2}\Big) + \frac{|B_i\mathbf{u} - f_i|_\gamma^2}{2|B_i\mathbf{u}^k - f_i|_\gamma} + \frac{|B_i\mathbf{u}^k - f_i|_\gamma}{2}\Big]. \quad (3.17)$$

We now show that HQA is the MMA for solving (3.4) with $\Psi := \Phi$.

Lemma 3.3.1 *HQA is the MMA for solving* (3.4) *with the tangent majorant function* $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ *defined in* (3.17).

Proof: We need to verify that $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ defined in (3.17) satisfies (3.13)–(3.15). Taking $\mathbf{u} = \mathbf{u}^k$, we obtain $\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k) = \Phi(\mathbf{u}^k)$, which is (3.14). The inequality $a^2 + b^2 \geq 2ab$ for all $a, b \in \mathbb{R}$ yields

$$\mathcal{G}(\mathbf{u}, \mathbf{u}^k) \geq \sum_{i=1}^{n} (\lambda|\nabla u_i|_\beta + |B_i\mathbf{u} - f_i|_\gamma) = \Phi(\mathbf{u}), \quad \forall \mathbf{u} \in \mathbb{R}^n.$$

Hence (3.13) holds. By taking the derivatives of $\Phi(\mathbf{u})$ and $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ with respect to $\mathbf{u}$, we have

$$\nabla\Phi(\mathbf{u}) = \lambda A^T \tilde{D}_\beta(\mathbf{u}) A\mathbf{u} + B^T D_\gamma(\mathbf{u})(B\mathbf{u} - \mathbf{f}), \quad (3.18)$$

$$\nabla_1\mathcal{G}(\mathbf{u}, \mathbf{u}^k) = \lambda A^T \tilde{D}_\beta(\mathbf{u}^k) A\mathbf{u} + B^T D_\gamma(\mathbf{u}^k)(B\mathbf{u} - \mathbf{f}). \quad (3.19)$$

Substituting $\mathbf{u} = \mathbf{u}^k$ into (3.18) and (3.19), we immediately have $\nabla\Phi(\mathbf{u}^k) = \nabla_1\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)$. □

In [12], the authors discuss the global convergence and linear convergence of MMAs when the energy function Φ and the majorant function $\mathcal{G}$ satisfy Hypotheses 4.1 and 4.2 there. We state them here as Hypotheses 1 and 2 respectively.

Hypothesis 1

(a) Φ *is twice continuously differentiable and strictly convex.*

(b) Φ *is coercive, i.e.,* $\lim_{\|\mathbf{u}\|_2 \to \infty} \Phi(\mathbf{u}) = \infty$.

(c) Φ *is bounded from below.*

Hypothesis 2

(a) *There exists a function* $C(\mathbf{v})$ *such that the followings hold:*

(i) $\mathcal{G}(\mathbf{u},\mathbf{v}) = \Phi(\mathbf{v}) + (\mathbf{u}-\mathbf{v})^T\nabla\Phi(\mathbf{v}) + \frac{1}{2}(\mathbf{u}-\mathbf{v})^T C(\mathbf{v})(\mathbf{u}-\mathbf{v})$ *for all* $\mathbf{u},\mathbf{v}\in\mathbb{R}^n$.

(ii) C *is continuous.*

(iii) $\lambda_{\min}(C(\mathbf{v})) \geq \eta > 0$, *for all* $\mathbf{v}\in\mathbb{R}^n$.

(b) $\Phi(\mathbf{u}) \leq \mathcal{G}(\mathbf{u},\mathbf{v})$ *for all* $\mathbf{u},\mathbf{v}\in\mathbb{R}^n$.

Thus in order to establish the linear convergence of HQA, we only have to show that Hypotheses 1 and 2 are satisfied. We start with Hypothesis 1. We will use the notation $U \succeq$ (respectively $\succ$)V to mean that $U - V$ is a positive semi-definite (respectively definite) matrix.

Lemma 3.3.2 *Let* $\ker(A^TA)\cap\ker(B^TB) = \{\mathbf{0}\}$. *Then* $\Phi(\mathbf{u})$ *defined in (3.4) satisfies Hypothesis 1. In particular,* $\Phi(\mathbf{u})$ *has a unique minimizer.*

Proof: By the definition of Φ in (3.4), Φ is obviously twice continuously differentiable and bounded from below by 0. We thus only need to prove the strict convexity and coercivity.

We start with the strict convexity. Taking derivatives on both sides of (3.18), we have

$$\nabla^2\Phi(\mathbf{u}) = \lambda\Big(\beta A^T\tilde{P}_\beta(\mathbf{u})A + T(\mathbf{u})^T P_\beta(\mathbf{u})T(\mathbf{u})\Big) + \gamma B^T P_\gamma(\mathbf{u})B, \tag{3.20}$$

where $\tilde{P}_\beta(\mathbf{u}) = \mathrm{diag}(P_\beta(\mathbf{u}), P_\beta(\mathbf{u}))$; $P_\beta(\mathbf{u})$ and $P_\gamma(\mathbf{u}) \in \mathbb{R}^{n\times n}$ are the diagonal matrices with their i-th diagonal entries being $1/|\nabla u_i|_\beta^3$ and $1/|B_i\mathbf{u} - f_i|_\gamma^3$ respectively; and $T(\mathbf{u}) \in \mathbb{R}^{n\times n}$ with $[T(\mathbf{u})]_{ij} = (A_2)_{ij}(A_1)_i\mathbf{u} - (A_1)_{ij}(A_2)_i\mathbf{u}$. Notice that

$$\begin{aligned}\nabla^2\Phi(\mathbf{u}) &\succeq \lambda\beta A^T\tilde{P}_\beta(\mathbf{u})A + \gamma B^T P_\gamma(\mathbf{u})B \\ &\succeq \frac{\lambda\beta}{|(2\|\mathbf{u}\|_\infty, 2\|\mathbf{u}\|_\infty)|_\beta^3}A^TA + \frac{\gamma}{|\|B\|_\infty\|\mathbf{u}\|_\infty + \|\mathbf{f}\|_\infty|_\gamma^3}B^TB.\end{aligned}$$

By the assumption $\ker(A^TA) \cap \ker(B^TB) = \{\mathbf{0}\}$, we have $\nabla^2\Phi(\mathbf{u}) \succ 0$, and the strict convexity of $\Phi(\mathbf{u})$ is proven.

Next we prove the coercivity. We apply the inequality $\sum_i a_i \geq \sqrt{\sum_i a_i^2}$ for any $a_i \in \mathbb{R}^+$ to $\Phi(\mathbf{u})$ in (3.4). After simple derivations, we obtain

$$\begin{aligned}\Phi(\mathbf{u}) &\geq \sum_{i=1}^{n} \sqrt{\lambda^2|\nabla u_i|_\beta^2 + |B_i\mathbf{u} - f_i|_\gamma^2} \geq \left[\sum_{i=1}^{n}\left(\lambda^2|\nabla u_i|_\beta^2 + |B_i\mathbf{u} - f_i|_\gamma^2\right)\right]^{1/2} \\ &= \sqrt{\mathbf{u}^T(\lambda^2A^TA + B^TB)\mathbf{u} - 2\mathbf{f}^TB\mathbf{u} + \|\mathbf{f}\|_2^2 + (\beta\lambda^2 + \gamma)n}\ .\end{aligned}$$

Since $\lambda^2A^TA + B^TB \succ 0$, $\sigma^2 := \lambda_{\min}(\lambda^2A^TA + B^TB) > 0$. Thus if $\|\mathbf{u}\|_2 \to \infty$, we see that $\Phi(\mathbf{u}) \approx \sigma\|\mathbf{u}\|_2 \to \infty$.

By [19, Proposition 1.2], the strict convexity and coercivity imply the existence and the uniqueness of the minimizer of $\Phi(\mathbf{u})$. □

Regarding Hypothesis 2, in fact, we cannot show that Hypothesis 2(a)(iii) holds for arbitrary vectors $\mathbf{v}$. We can only show that it holds for $\mathbf{v} = \mathbf{u}^k$, the sequence generated by HQA. However, as we will see later in Theorem 3.3.4, it is enough to prove the convergence of HQA by showing that Hypothesis 2 holds only for $\mathbf{v} = \mathbf{u}^k$, and this is what we are going to prove in the next Lemma.

Lemma 3.3.3 *Let* $\ker(A^TA)\cap\ker(B^TB) = \{\mathbf{0}\}$. *Then* $\Phi(\mathbf{u})$ *defined in (3.4) and* $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ *defined in (3.17) satisfy Hypothesis 2 with* $\mathbf{v} = \mathbf{u}^k$ *there. In particular, the coefficient matrix of the linear system in (3.11) is invertible.*

Proof: From (3.19), the Hessian of $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ is given by

$$\nabla_1^2\mathcal{G}(\mathbf{u}, \mathbf{u}^k) = \lambda A^T\tilde{D}_\beta(\mathbf{u}^k)A + B^TD_\gamma(\mathbf{u}^k)B, \tag{3.21}$$

which is independent of $\mathbf{u}$. Hence $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ is quadratic in $\mathbf{u}$. Taking its Taylor expansion at $\mathbf{u}^k$, we have

$$\begin{aligned}\mathcal{G}(\mathbf{u}, \mathbf{u}^k) &= \mathcal{G}(\mathbf{u}^k, \mathbf{u}^k) + \langle\nabla_1\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k), \mathbf{u} - \mathbf{u}^k\rangle \\ &\quad + \frac{1}{2}(\mathbf{u} - \mathbf{u}^k)^T\nabla_1^2\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)(\mathbf{u} - \mathbf{u}^k).\end{aligned} \tag{3.22}$$

Since we have proven that the HQA is indeed the MMA in Lemma 3.3.1, we can replace $\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)$ and $\nabla_1 \mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)$ by $\Phi(\mathbf{u}^k)$ and $\nabla\Phi(\mathbf{u}^k)$ respectively in the equality above and then we obtain

$$\begin{aligned} \mathcal{G}(\mathbf{u}, \mathbf{u}^k) &= \Phi(\mathbf{u}^k) + \langle \nabla\Phi(\mathbf{u}^k), \mathbf{u} - \mathbf{u}^k \rangle \\ &\quad + \frac{1}{2}(\mathbf{u} - \mathbf{u}^k)^T \nabla_1^2 \mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)(\mathbf{u} - \mathbf{u}^k). \end{aligned} \tag{3.23}$$

This is precisely Hypothesis 2(a)(i) with $C(\mathbf{u}^k) := \nabla_1^2 \mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)$ which is given by (3.21). Recalling the definitions of $D_\beta(\cdot)$ and $D_\gamma(\cdot)$ in (3.10), we see that Hypothesis 2(a)(ii) is satisfied.

By Lemma 3.3.1 and the definition of the majorant function, see (3.12)–(3.14), we immediately have

$$\Phi(\mathbf{u}^{k+1}) \leq \mathcal{G}(\mathbf{u}^{k+1}, \mathbf{u}^k) \leq \mathcal{G}(\mathbf{u}^k, \mathbf{u}^k) = \Phi(\mathbf{u}^k). \tag{3.24}$$

Hence $\{\Phi(\mathbf{u}^k)\}$ is monotonic decreasing and bounded from above by $\Phi(\mathbf{u}^0)$. Therefore, by coercivity, see Hypothesis 1(b), $\{\|\mathbf{u}^k\|_2\}$ must be bounded from above. Denote the bound by M. Recalling the definition of $D_\beta(\mathbf{u}^k), D_\gamma(\mathbf{u}^k)$ in (3.10), we have

$$\begin{aligned} &\lambda_{\min}(C(\mathbf{u}^k)) = \lambda_{\min}\left(\nabla_1^2 \mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)\right) \\ \geq\ &\lambda_{\min}\left(\frac{\lambda}{|(2M, 2M)|_\beta} A^T A + \frac{1}{|M\|B\|_\infty + \|\mathbf{f}\|_\infty|\gamma} B^T B\right) := \eta \end{aligned} \tag{3.25}$$

Since $\ker(A^T A) \cap \ker(B^T B) = \{\mathbf{0}\}$, we see that $\eta > 0$. Thus Hypothesis 2(a)(iii) holds.

Hypothesis 2(b) is just (3.13), and hence is true. Finally notice that the coefficient matrix of the linear system in (3.11) is precisely $\nabla_1^2 \mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)$ in (3.21) and hence by (3.25), it is invertible. □

Since Hypothesis 2 is only valid for $\mathbf{u}^k$ and not for arbitrary vectors $\mathbf{v}$, we cannot directly apply the convergence theorems in [12]. However, the proof in [12] can easily be adapted to prove the following two convergence theorems for

HQA. For the convenience of the readers to understand the following theorems, we give the proof after the two theorems as the illustrations.

Theorem 3.3.4 *For the sequence* $\{\mathbf{u}^k\}$ *generated by HQA, if* $\ker(A^TA)\cap\ker(B^TB) = \{\mathbf{0}\}$, *we have*

(a) $\{\Phi(\mathbf{u}^k)\}$ *is monotonic decreasing and convergent;*

(b) $\lim_{k\to\infty} \|\mathbf{u}^k - \mathbf{u}^{k+1}\|_2 = 0$;

(c) $\{\mathbf{u}^k\}$ *converges to the unique minimizer* $\mathbf{u}^*$ *of* $\Phi(\mathbf{u})$ *from any initial guess* $\mathbf{u}^0$.

Proof:

(a) By Hypothesis 1(c), the sequence $\{\Phi(\mathbf{u}^k)\}$ is bounded from below. By (3.24), it is also monotonically decreasing. Hence it converges and

$$\lim_{k\to\infty}\left[\Phi(\mathbf{u}^k) - \Phi(\mathbf{u}^{k+1})\right] = 0. \tag{3.26}$$

(b) Recall that by establishing (3.21), we see that $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ is quadratic in $\mathbf{u}$. Taking the Taylor expansion of $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ at $\mathbf{u}^{k+1}$, we have

$$\begin{aligned}\mathcal{G}(\mathbf{u}, \mathbf{u}^k) &= \mathcal{G}(\mathbf{u}^{k+1}, \mathbf{u}^k) + \langle \nabla_1\mathcal{G}(\mathbf{u}^{k+1}, \mathbf{u}^k), \mathbf{u} - \mathbf{u}^{k+1}\rangle \\ &\quad + \frac{1}{2}(\mathbf{u} - \mathbf{u}^{k+1})^T\nabla_1^2\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)(\mathbf{u} - \mathbf{u}^{k+1}). \end{aligned}\tag{3.27}$$

By (3.12), we have $\nabla_1\mathcal{G}(\mathbf{u}^{k+1}, \mathbf{u}^k) = 0$. By taking $\mathbf{u} = \mathbf{u}^k$ in (3.27) and using (3.25), we thus have

$$\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k) \geq \mathcal{G}(\mathbf{u}^{k+1}, \mathbf{u}^k) + \frac{\eta}{2}\|\mathbf{u}^k - \mathbf{u}^{k+1}\|_2^2,$$

where $\eta > 0$ by Hypothesis 2(a)(iii). From (3.24), we obtain that

$$\Phi(\mathbf{u}^k) - \Phi(\mathbf{u}^{k+1}) \geq (\eta/2)\|\mathbf{u}^k - \mathbf{u}^{k+1}\|_2^2.$$

Together with (3.26), we have $\lim_{k\to\infty} \|\mathbf{u}^k - \mathbf{u}^{k+1}\|_2 = 0$.

(c) In the proof of Lemma 3.3.3, we have shown that the sequence $\{\|\mathbf{u}^k\|_2\}$ is bounded by M. Hence it converges to the unique minimizer $\mathbf{u}^*$ if and only if all convergent subsequence of $\{\mathbf{u}^k\}$ converges to $\mathbf{u}^*$. Let $\{\mathbf{u}^{k_j}\}$ be an arbitrary convergence subsequence of $\{\mathbf{u}^k\}$ that converges to $\bar{\mathbf{u}}$. We have to prove that $\bar{\mathbf{u}} = \mathbf{u}^*$. Since $\mathcal{G}(\mathbf{u}, \mathbf{u}^{k_j})$ is quadratic in $\mathbf{u}$, we have

$$\begin{aligned}\mathcal{G}(\mathbf{u}, \mathbf{u}^{k_j}) &= \mathcal{G}(\mathbf{u}^{k_j}, \mathbf{u}^{k_j}) + \langle \nabla_1 \mathcal{G}(\mathbf{u}^{k_j}, \mathbf{u}^{k_j}), \mathbf{u} - \mathbf{u}^{k_j} \rangle \\ &\quad + \frac{1}{2}(\mathbf{u} - \mathbf{u}^{k_j})^T \nabla_1^2 \mathcal{G}(\mathbf{u}^{k_j}, \mathbf{u}^{k_j})(\mathbf{u} - \mathbf{u}^{k_j}).\end{aligned}$$

Substituting (3.15) here and taking partial derivative with respect to $\mathbf{u}$, we then have

$$\nabla_1 \mathcal{G}(\mathbf{u}, \mathbf{u}^{k_j}) = \nabla \Phi(\mathbf{u}^{k_j}) + \nabla_1^2 \mathcal{G}(\mathbf{u}^{k_j}, \mathbf{u}^{k_j})(\mathbf{u} - \mathbf{u}^{k_j}).$$

It is continuous since Φ is twice continuously differentiable by Hypothesis 1(a), and $C(\mathbf{u}^{k_j}) = \nabla_1^2 \mathcal{G}(\mathbf{u}^{k_j}, \mathbf{u}^{k_j})$ is continuous by Hypothesis 2(a)(ii). Letting $\mathbf{u} = \mathbf{u}^{k_j+1}$ and using (3.12), we then have

$$0 = \nabla_1 \mathcal{G}(\mathbf{u}^{k_j+1}, \mathbf{u}^{k_j}) = \nabla \Phi(\mathbf{u}^{k_j}) + \nabla_1^2 \mathcal{G}(\mathbf{u}^{k_j}, \mathbf{u}^{k_j})(\mathbf{u}^{k_j+1} - \mathbf{u}^{k_j}).$$

Taking limits, we obtain

$$\begin{aligned}0 = \lim_{j \to \infty} \nabla_1 \mathcal{G}(\mathbf{u}^{k_j+1}, \mathbf{u}^{k_j}) &= \nabla_1 \mathcal{G}(\lim_{j \to \infty} \mathbf{u}^{k_j+1}, \lim_{j \to \infty} \mathbf{u}^{k_j}) \\ &= \nabla_1 \mathcal{G}(\bar{\mathbf{u}}, \bar{\mathbf{u}}) = \nabla \Phi(\bar{\mathbf{u}}) + \nabla_1^2 \mathcal{G}(\bar{\mathbf{u}}, \bar{\mathbf{u}})(\bar{\mathbf{u}} - \bar{\mathbf{u}}) \\ &= \nabla \Phi(\bar{\mathbf{u}}).\end{aligned}$$

By the uniqueness of the minimizer, see Lemma 3.3.2, we can conclude that $\bar{\mathbf{u}} = \mathbf{u}^*$. □

We recall that the existence and uniqueness of the minimizer of $\Phi(\mathbf{u})$ is proven in Lemma 3.3.2. In the following theorem, we derive the linear convergence of the HQA and give a bound of its convergence rate.

Theorem 3.3.5 *Suppose* $\ker(A^TA) \cap \ker(B^TB) = \{\mathbf{0}\}$. *Let* $\mathbf{u}^*$ *be the unique minimizer of* $\Phi(\mathbf{u})$ *and*

$$\Lambda := 1 - \lambda_{\min}\Big(\nabla_1^2\mathcal{G}(\mathbf{u}^*, \mathbf{u}^*)^{-1}\nabla^2\Phi(\mathbf{u}^*)\Big).$$

Then $\Lambda < 1$ *and the sequence* $\{\Phi(\mathbf{u}^k)\}$ *has a linear convergence rate of at most* Λ *while the sequence* $\{\mathbf{u}^k\}$ *is r-linearly convergent with a convergence rate of at most* $\sqrt{\Lambda}$.

Proof: Define

$$\lambda_k = \frac{\mathcal{G}(\mathbf{u}^*, \mathbf{u}^k) - \Phi(\mathbf{u}^*)}{\frac{1}{2}(\mathbf{u}^* - \mathbf{u}^k)^T\nabla_1^2\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)(\mathbf{u}^* - \mathbf{u}^k)}.$$

We know that $\lambda_k \geq 0$ since $\Phi(\mathbf{u}^*) \leq \mathcal{G}(\mathbf{u}^*, \mathbf{u}^k)$ by (3 13) and Lemma 3.3.1 To finish the proof of the theorem, let us begin with proving the following claim:

Claim: $\Phi(\mathbf{u}^{k+1}) - \Phi(\mathbf{u}^*) \leq \lambda_k(\Phi(\mathbf{u}^k) - \Phi(\mathbf{u}^*))$.

Proof. Recall that $\nabla^2\Phi(\mathbf{u})$ and $\nabla_1^2\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ are the Hessian of $\Phi(\mathbf{u})$ and $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ respectively. By (3.22), definition of the tangent majorant function (3.13)–(3.15) and Lemma 3.3.1, we take the Taylor expansions of $\mathcal{G}(\mathbf{u}^*, \mathbf{u}^k)$ and $\Phi(\mathbf{u}^*)$ at $\mathbf{u}^k$,

$$\mathcal{G}(\mathbf{u}^*, \mathbf{u}^k) = \Phi(\mathbf{u}^k) + \langle\nabla\Phi(\mathbf{u}^k), \mathbf{u}^* - \mathbf{u}^k\rangle + \frac{1}{2}(\mathbf{u}^* - \mathbf{u}^k)^T\nabla_1^2\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)(\mathbf{u}^* - \mathbf{u}^k), \tag{3.28}$$

$$\Phi(\mathbf{u}^*) = \Phi(\mathbf{u}^k) + \langle\nabla\Phi(\mathbf{u}^k), \mathbf{u}^* - \mathbf{u}^k\rangle + \frac{1}{2}(\mathbf{u}^* - \mathbf{u}^k)^T\nabla^2\Phi(\mathbf{u}^k)(\mathbf{u}^* - \mathbf{u}^k) + \mathcal{O}(\|\mathbf{u}^* - \mathbf{u}^k\|^3) \tag{3.29}$$

We substract (3.29) from (3.28) and divide by $\frac{1}{2}(\mathbf{u}^* - \mathbf{u}^k)^T\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)(\mathbf{u}^* - \mathbf{u}^k)$, then we get

$$\begin{aligned}
\lambda_k &= \frac{\mathcal{G}(\mathbf{u}^*, \mathbf{u}^k) - \Phi(\mathbf{u}^*)}{\frac{1}{2}(\mathbf{u}^* - \mathbf{u}^k)^T\nabla_1^2\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)(\mathbf{u}^* - \mathbf{u}^k)} \\
&= \frac{\frac{1}{2}(\mathbf{u}^* - \mathbf{u}^k)^T\Big(\nabla_1^2\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k) - \nabla^2\Phi(\mathbf{u}^k)\Big)(\mathbf{u}^* - \mathbf{u}^k) + \mathcal{O}(\|\mathbf{u}^* - \mathbf{u}^k\|^3)}{\frac{1}{2}(\mathbf{u}^* - \mathbf{u}^k)^T\nabla_1^2\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)(\mathbf{u}^* - \mathbf{u}^k)} \\
&= 1 - \frac{\frac{1}{2}(\mathbf{u}^* - \mathbf{u}^k)^T\nabla^2\Phi(\mathbf{u}^k)(\mathbf{u}^* - \mathbf{u}^k)}{\frac{1}{2}(\mathbf{u}^* - \mathbf{u}^k)^T\nabla_1^2\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)(\mathbf{u}^* - \mathbf{u}^k)} + \mathcal{O}(\|\mathbf{u}^* - \mathbf{u}^k\|)
\end{aligned} \tag{3.30}$$

Notice that $\nabla^2\Phi(\mathbf{u}^k)$ and $\nabla_1^2\mathcal{G}(\mathbf{u}^k,\mathbf{u}^k)$ are continuous and symmetric positive definite by (3.20) and (3.21). Hence, as k is large enough, we have $0 \leq \lambda_k \leq 1$. By (3.13), (3.12), Lemma 3.3.1 and the convexity of $\mathcal{G}(\mathbf{u},\mathbf{u}^k)$, we have

$$\Phi(\mathbf{u}^{k+1}) \leq \mathcal{G}(\mathbf{u}^{k+1},\mathbf{u}^k) \leq \mathcal{G}(\lambda_k\mathbf{u}^k + (1-\lambda_k)\mathbf{u}^*,\mathbf{u}^k). \tag{3.31}$$

By (3.23), we rewrite

$$\begin{aligned}
& \mathcal{G}(\lambda_k\mathbf{u}^k + (1-\lambda_k)\mathbf{u}^*,\mathbf{u}^k) \\
= \;& \Phi(\mathbf{u}^k) + (1-\lambda_k)\langle\nabla\Phi(\mathbf{u}^k),\mathbf{u}^*-\mathbf{u}^k\rangle \\
& + \frac{(1-\lambda_k)^2}{2}(\mathbf{u}^*-\mathbf{u}^k)^T\nabla_1^2\mathcal{G}(\mathbf{u}^k,\mathbf{u}^k)(\mathbf{u}^*-\mathbf{u}^k) \\
= \;& \lambda_k\Phi(\mathbf{u}^k) + (1-\lambda_k)\Big(\mathcal{G}(\mathbf{u}^*,\mathbf{u}^k) - \frac{\lambda_k}{2}(\mathbf{u}^*-\mathbf{u}^k)\nabla\mathcal{G}(\mathbf{u}^k,\mathbf{u}^k)(\mathbf{u}^*-\mathbf{u}^k)\Big) \\
= \;& \lambda_k\Phi(\mathbf{u}^k) + (1-\lambda_k)\Phi(\mathbf{u}^*)
\end{aligned} \tag{3.32}$$

Substituting (3.32) into (3.31), we have

$$\Phi(\mathbf{u}^{k+1}) \leq \lambda_k\Phi(\mathbf{u}^k) + (1-\lambda_k)\Phi(\mathbf{u}^*).$$

Subtracting $\Phi(\mathbf{u}^*)$ on the both side of the above inequality, we then have

$$\Phi(\mathbf{u}^{k+1}) - \Phi(\mathbf{u}^*) \leq \lambda_k(\Phi(\mathbf{u}^k) - \Phi(\mathbf{u}^*)).$$

Recalling that $\nabla^2\Phi(\mathbf{u}^k)$ and $\nabla_1^2\mathcal{G}(\mathbf{u}^k,\mathbf{u}^k)$ are both symmetric and positive definite, (3.30) immediately implies that

$$\lambda_k \leq 1 - \lambda_{\min}((\nabla_1^2\mathcal{G}(\mathbf{u}^k,\mathbf{u}^k))^{-1}\nabla^2\Phi(\mathbf{u}^k)) + \mathcal{O}(\|\mathbf{u}^*-\mathbf{u}^k\|). \tag{3.33}$$

Since $\nabla_1^2\mathcal{G}$ and $\nabla^2\Phi$ are both continuous, hence

$$1 - \lambda_{\min}((\nabla_1^2\mathcal{G}(\mathbf{u}^k,\mathbf{u}^k))^{-1}\nabla^2\Phi(\mathbf{u}^k)) \to 1 - \lambda_{\min}((\nabla_1^2\mathcal{G}(\mathbf{u}^*,\mathbf{u}^*))^{-1}\nabla^2\Phi(\mathbf{u}^*). \tag{3.34}$$

By (3.20) and (3.21), $\nabla^2\Phi$ and $\nabla_1^2\mathcal{G}$ are both symmetric, positive definite at $\mathbf{u}^*$, therefore,

$$\begin{aligned}
& (\nabla_1^2\mathcal{G}(\mathbf{u}^*,\mathbf{u}^*))^{-1}\nabla^2\Phi(\mathbf{u}^*) \\
= \;& (\nabla_1^2\mathcal{G}(\mathbf{u}^*,\mathbf{u}^*))^{-\frac{1}{2}}(\nabla_1^2\mathcal{G}(\mathbf{u}^*,\mathbf{u}^*))^{-\frac{1}{2}}\nabla^2\Phi(\mathbf{u}^*)(\nabla_1^2\mathcal{G}(\mathbf{u}^*,\mathbf{u}^*))^{-\frac{1}{2}}(\nabla_1^2\mathcal{G}(\mathbf{u}^*,\mathbf{u}^*))^{\frac{1}{2}}.
\end{aligned}$$

Hence $(\nabla_1^2\mathcal{G}(\mathbf{u}^*,\mathbf{u}^*))^{-1}\nabla^2\Phi(\mathbf{u}^*)$ is the similar to $(\nabla_1^2\mathcal{G}(\mathbf{u}^*,\mathbf{u}^*))^{-\frac{1}{2}}\nabla^2\Phi(\mathbf{u}^*)(\nabla_1^2\mathcal{G}(\mathbf{u}^*,\mathbf{u}^*))^{-\frac{1}{2}}$, which is positive definite, thus

$$\begin{aligned} & \lambda_{\min}\Big((\nabla_1^2\mathcal{G}(\mathbf{u}^*,\mathbf{u}^*))^{-1}\nabla^2\Phi(\mathbf{u}^*)\Big) \\ = \; & \lambda_{\min}\Big((\nabla_1^2\mathcal{G}(\mathbf{u}^*,\mathbf{u}^*))^{-\frac{1}{2}}\nabla^2\Phi(\mathbf{u}^*)(\nabla_1^2\mathcal{G}(\mathbf{u}^*,\mathbf{u}^*))^{-\frac{1}{2}}\Big) > 0. \end{aligned} \tag{3.35}$$

By (3.34) and (3.35), we have

$$\Lambda = 1 - \lambda_{\min}\Big((\nabla_1^2\mathcal{G}(\mathbf{u}^*,\mathbf{u}^*))^{-1}\nabla^2\Phi(\mathbf{u}^*)\Big) < 1 \tag{3.36}$$

Together with (3.33), we have $\lambda_k \leq \Lambda < 1$ for k sufficiently large. Since $\mathbf{u}^*$ is the minimizer of $\Phi(\mathbf{u})$, hence $\nabla\Phi(\mathbf{u}^*) = \mathbf{0}$. Taking the Taylor expansion of $\Phi(\mathbf{u})$ at $\mathbf{u}^*$, we get

$$\begin{aligned} \Phi(\mathbf{u}^k) - \Phi(\mathbf{u}^*) &= \frac{1}{2}(\mathbf{u}^k - \mathbf{u}^*)^T\nabla^2\Phi(\mathbf{u}^*)(\mathbf{u}^k - \mathbf{u}^*) + \mathcal{O}(\|\mathbf{u}^k - \mathbf{u}^*\|^3) \\ &\geq \frac{1}{2}\lambda_{\min}(\nabla^2\Phi(\mathbf{u}^*))\|\mathbf{u}^k - \mathbf{u}^*\|^2 + \mathcal{O}(\|\mathbf{u}^k - \mathbf{u}^*\|^3). \end{aligned}$$

Thus, there exists $K > 2$ such that

$$\|\mathbf{u}^k - \mathbf{u}^*\| \leq \sqrt{\frac{K}{\lambda_{\min}(\nabla^2\Phi(\mathbf{u}^*))}(\Phi(\mathbf{u}^k) - \Phi(\mathbf{u}^*))} := y_k, \tag{3.37}$$

for k sufficiently large. By the Claim,

$$y_{k+1}^2 = \frac{K}{\lambda_{\min}(\nabla^2\Phi(\mathbf{u}^*))}(\Phi(\mathbf{u}^{k+1}) - \Phi(\mathbf{u}^*)) \leq \frac{K}{\lambda_{\min}(\nabla^2\Phi(\mathbf{u}^*))}\Lambda(\Phi(\mathbf{u}^k) - \Phi(\mathbf{u}^*)) = \Lambda y_k^2,$$

i.e. $y_{k+1} \leq \sqrt{\Lambda}y_k$, hence $\{\mathbf{u}^k\}$ is r-linearly convergent with at most $\sqrt{\Lambda}$ as the r-convergence rate.

□

3.4 Numerical Examples

In this section, we apply our algorithm to deblur images that are also corrupted by impulse noise. The deblurring problem has been discussed recently in many

papers, see for examples [18, 52, 51]. Among all these methods, the FTVDM and the ALM are the most efficient; and according to the numerical results in [51], ALM is the fastest one. Hence in this paper, we compare our HQA with FTVDM and ALM only. The FTVDM and ALM codes we used here are provided by the authors in [51]. For ALM, in (2.6), parameters $\gamma_{\mathbf{w}}$ and $\gamma_{\mathbf{z}}$ are fixed to be $\gamma_{\mathbf{w}} = 10$ and $\gamma_{\mathbf{z}} = 100$ in all the experiments. For FTVDM, the parameters $\theta_{\mathbf{w}}$ and $\theta_{\mathbf{z}}$ are upper limited to be $\theta_{\mathbf{w}} = 2^{10}, \theta_{\mathbf{z}} = 2^{15}$ in the approximate problem (2.2). Using $\theta_{\mathbf{w}} > 2^{10}, \theta_{\mathbf{z}} > 2^{15}$ would only increase the computational cost but not solution quality. To speed up the convergence, $\theta_{\mathbf{w}}$ and $\theta_{\mathbf{z}}$ are both implemented in a continuous scheme; that is, let $\theta_{\mathbf{w}}$ and $\theta_{\mathbf{z}}$ take small values at the beginning and gradually increase their values to 2^{10} and 2^{15} respectively. Specially, a $\theta_{\mathbf{w}}$-sequence $2^0, 2^{2/3}, 2^{4/3}, \cdots, 2^{10}$ is tested. Accordingly, $\theta_{\mathbf{z}}$ is set to be $2^0, 2^1, 2^2, \cdots, 2^{15}$.

We test three 256-by-256 images: Barbara, Bridge and Goldhill. The matrix B is the blurring matrix corresponds to the Gaussian blur generated by the MATLAB command

```
fspecial('Gaussian', [7 , 7], 5).
```

Then impulse noise is added to the blurred image to obtain the observed image $\mathbf{f}$. The noise levels are 30%, 40%, 50%, 60% for salt-and-pepper noise and 20%, 30%, 40%, 50% for random-valued noise. We use the same regularization parameter λ for all algorithms at the same noise level. For salt-and-pepper noise, $\lambda = 1/13, 1/10, 1/8, 1/4$ for noise level $30\%, 40\%, 50\%, 60\%$ respectively. For random-valued noise, $\lambda = 1/25, 1/10, 1/8, 1/4$ for noise level $20\%, 30\%, 40\%, 50\%$. In HQA, we initialize $\mathbf{u_0} = \text{rand}(\text{size}(\mathbf{f}))$. As in the FTVDM, similar settings to β and γ are used in HQA. We set β-sequence $10^{-3}, 10^{-4}, \cdots, 10^{-16}$ in (3.2). Corresponding, we set $\gamma = \beta^2$. To speed up the convergence and improve the resolution quality, we let β, γ take large values at the beginning and decrease their values to 10^{-10} and 10^{-20} respectively. Equation (3.11) is solved by the conjugate gradient method where we fix the maximum number of iterations to 10. In all

tests, we consider periodic boundary condition for the difference matrix A, as it is the boundary condition used in the tests in [51]. We compare the accuracy of the methods by the signal-to-noise ratio (SNR) used in [51]. It is defined by

$$\text{SNR} := 10\log_{10}\frac{\|\mathbf{u} - E(\mathbf{u})\|_2^2}{\|\hat{\mathbf{u}} - \mathbf{u}\|_2^2}(\text{dB}).$$

Here $\mathbf{u}$ and $\hat{\mathbf{u}}$ denote the original image and the restored image respectively, and $E(\mathbf{u})$ is the mean gray-level value of the original image.

Image	Method	Salt-and-pepper noise				Random-valued noise			
		30%	40%	50%	60%	20%	30%	40%	50%
Barbara	ALM	13.93	13.35	12.45	11.37	14.63	13.61	12.91	11.24
	HQA	14.24	13.59	12.83	11.72	15.17	13.81	12.86	11.06
Bridge	ALM	11.85	10.95	10.13	8.52	13.05	11.31	10.35	8.53
	HQA	12.03	11.12	10.27	9.00	13.46	11.37	10.13	8.72
Goldhill	ALM	16.08	15.03	13.78	12.05	17.29	15.53	14.41	12.14
	HQA	16.50	15.32	14.10	12.72	18.02	15.68	13.98	11.98

Table 3.1: SNR of the restored images.

First we compare the speed of the three methods. Figures 3.7–3.12 show the timing comparison of the three algorithms. Each point in the figures show the accumulated CPU time until that iteration and the corresponding SNR. For the deblurring problem with salt-and-pepper noise, see Figures 3.7–3.9, our method is the fastest amongst the three methods. It is also the most accurate one. For random-valued noise, when the noise level is low, HQA is comparable to the fastest method ALM. But when the noise level is higher, say 50%, HQA becomes the fastest. In Figure 3.13, we zoom into part of the restored Barbara image by the three methods. We see that the images by the ALM and the FTVDM are both blocky whereas that by our algorithm is less blocky.

From Figures 3.7–3.12, it is clear that FTVDM is the slowest amongst the three. In order to compare the accuracy of the two faster methods ALM and

HQA more precisely, we list in Table 3.1 the average SNR of the recovered images in five trials by the two methods. To compare the timing fairly, we first run ALM until it converges, say with t_0 CPU seconds. Then we let HQA run until the CPU time of the kth iteration is just greater than t_0. Then we record the SNR of the $(k-1)$th iteration as our result for HQA. We see from Table 3.1 that HQA is more accurate than ALM for the deblurring problem with the salt-and-pepper noise; and it is comparable to ALM for random-valued noise, especially when the noise level is low. See also Figures 3.1–3.6.

We illustrate the convergence rate of $\{\Phi(\mathbf{u}^k)\}$ and $\{\mathbf{u}^k\}$ in our HQA as mentioned in Theorem 3.3.5. We use the Barbara image as example. Since we do not have the true minimizer, we set $\mathbf{u}^* = \mathbf{u}^{30}$. In Figure 3.14, we plot the ratio $R_\Phi^k := [\Phi(\mathbf{u}^{k+1}) - \Phi(\mathbf{u}^{30})]/[\Phi(\mathbf{u}^k) - \Phi(\mathbf{u}^{30})]$ against the iteration number. We see that the ratios are all less than 1, indicating that $\{\Phi(\mathbf{u}^k)\}$ is converging linearly as stated in Theorem 3.3.5. In Figure 3.15, we plot $R_{\mathbf{u}}^k := [\|\mathbf{u}^{k+1} - \mathbf{u}^{30}\|_2/\|\mathbf{u}^k - \mathbf{u}^{30}\|_2]$ against the iteration number. We see that $R_{\mathbf{u}}^k < 1$, indicating that $\{\mathbf{u}^k\}$ indeed is linearly convergent.

As $\theta_{\mathbf{w}}$ and $\theta_{\mathbf{z}}$ are both small, the FTVDM converges fast, while the solution is far away from the true solution of (2.1). Hence, first problem (2.2) is solved at small $\theta_{\mathbf{w}}^k$ and $\theta_{\mathbf{z}}^k$ whose solution is taken to be the initial guess of the FTVDM for solving (2.2) at big $\theta_{\mathbf{w}}^{k+1}$ and $\theta_{\mathbf{z}}^{k+1}$ from $k = 0$. We take $\beta = 10^{-1}, 10^{-2}, \cdots, 10^{-17}$, correspondingly, $\gamma = \beta^2$ and compare the FTVDM and the HQA. Figure 3.16 shows their comparison results on the SNR versus CPU time and SNR versus iteration No.. The jump shows the improvements in SNR as $\theta_{\mathbf{w}}^k, \theta_{\mathbf{z}}^k, \beta^k, \gamma^k$ change to $\theta_{\mathbf{w}}^{k+1}, \theta_{\mathbf{z}}^{k+1}, \beta^{k+1}, \gamma^{k+1}$.

As β, γ decrease to $10^{-17}, 10^{-34}$ in HQA, from Figure 3.17, we see that the solution of the HQA tends to be blocky, which is close to the solution of (2.1).

3.5 Conclusions

In this chapter, we apply the half-quadratic technique to construct a method for solving TV-L1 minimization problems. We analyze its convergence using results of the majorize-minimize technique. Our algorithm is proven to be extremely efficient in terms of accuracy, computational time and the visual quality as demonstrated by the numerical experiments. In [42], the authors discuss the multiplicative and additive half-quadratic technique. Our future work is to apply the additive half-quadratic technique to TV-L1 minimization problem and extend our algorithm to color image restoration via TV and L1 data fitting and to general Lp regularized problems.

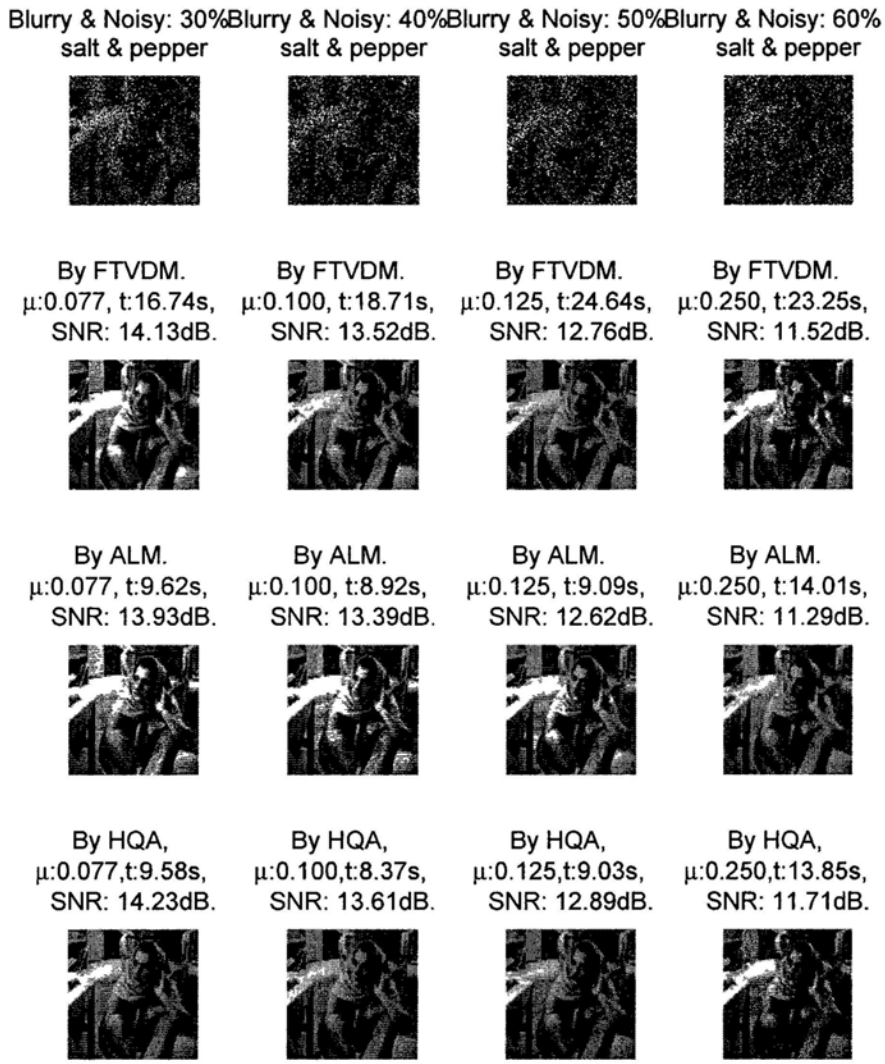

Figure 3.1: TV-L1 restoration for Barbara image from 7×7 sized Gaussian blur with salt & pepper noise from 30% t0 60%.

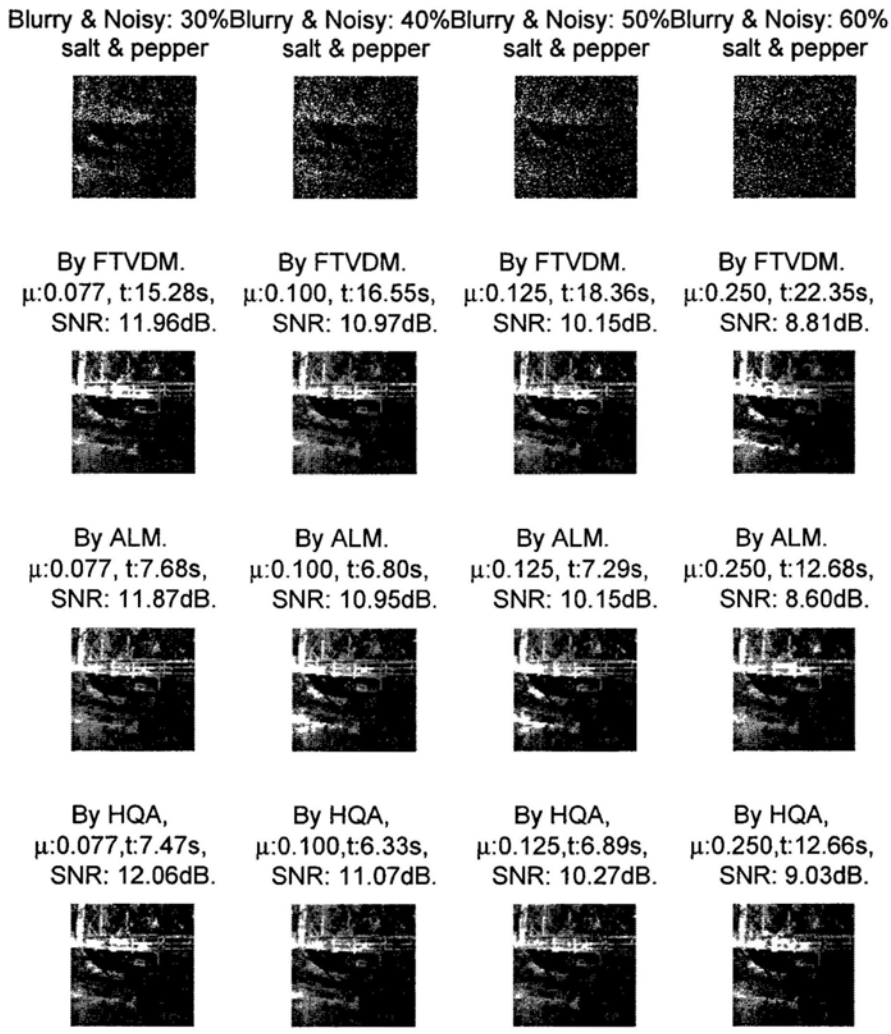

Figure 3.2: TV-L1 restoration for Bridge image from 7×7 sized Gaussian blur with salt & pepper noise from 30% t0 60%.

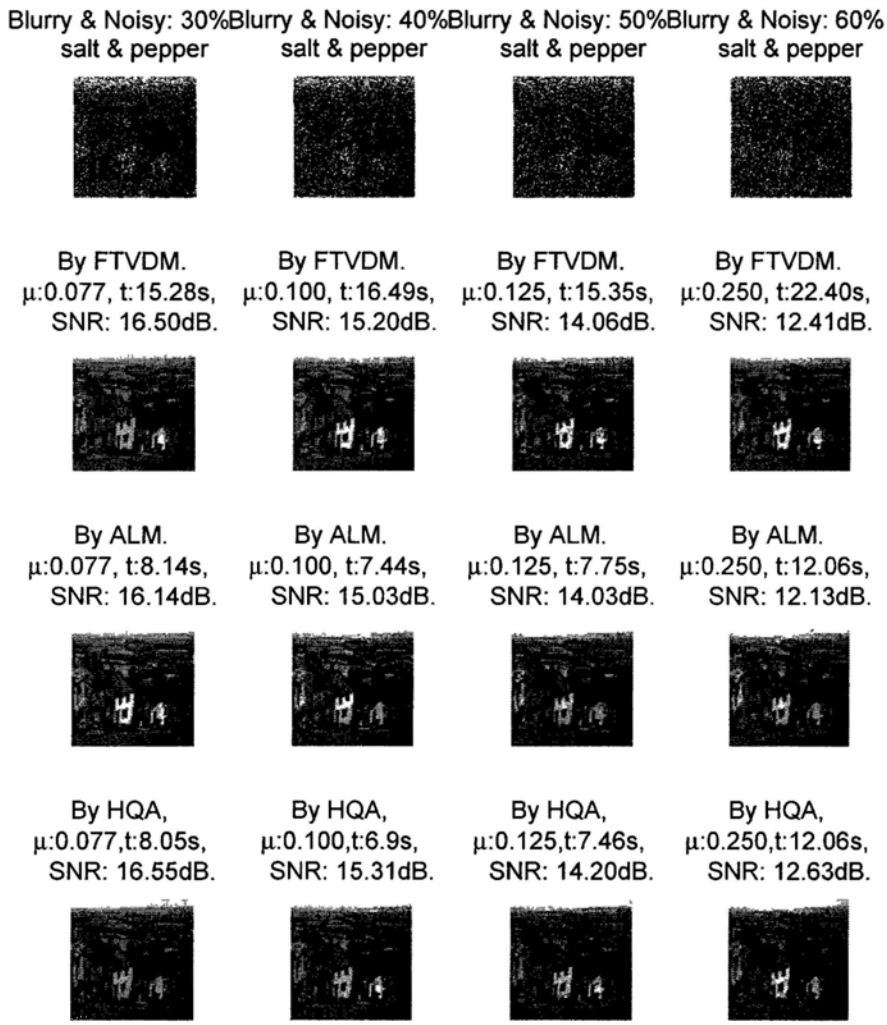

Figure 3.3: TV-L1 restoration for Goldhill image from 7×7 sized Gaussian blur with salt & pepper noise from 30% t0 60%.

Figure 3.4: TV-L1 restoration for Barbara image from 7×7 sized Gaussian blur with random-valued noise from 20% t0 50%.

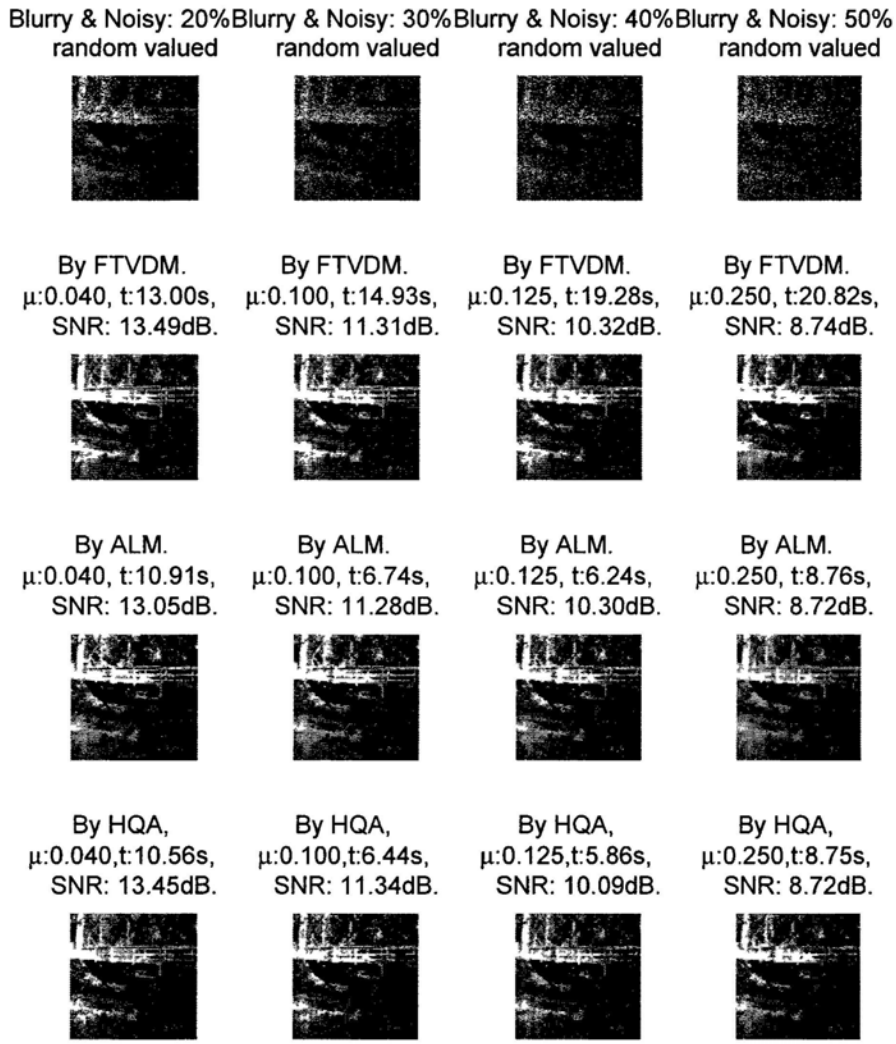

Figure 3.5: TV-L1 restoration for Bridge image from 7×7 sized Gaussian blur with random-valued noise from 20% t0 50%.

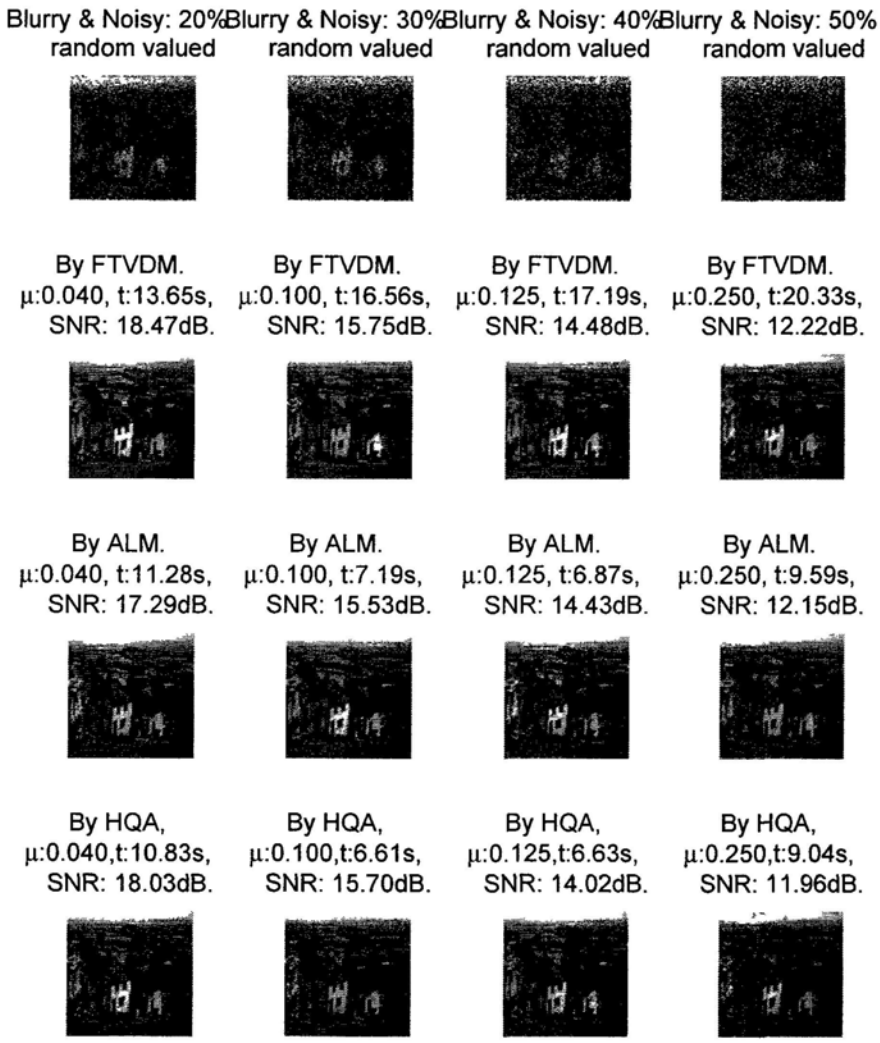

Figure 3.6: TV-L1 restoration for Goldhill image from 7×7 sized Gaussian blur with random-valued noise from 20% t0 50%.

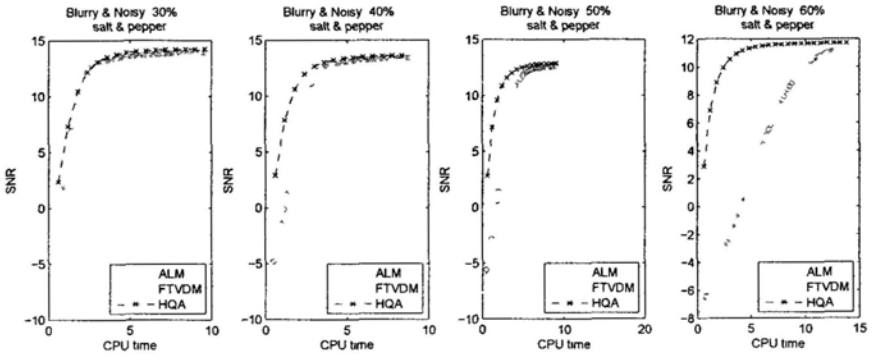

Figure 3.7: SNR versus CPU time in seconds for "Barbara" with salt-and-pepper noise.

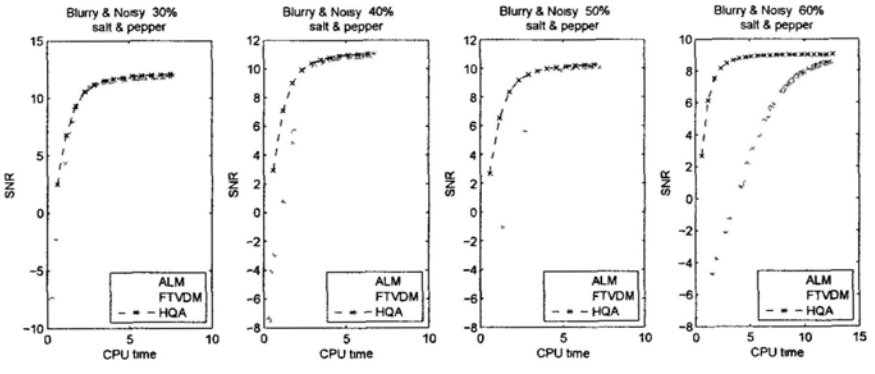

Figure 3.8: SNR versus CPU time in seconds for "Bridge" with salt-and-pepper noise.

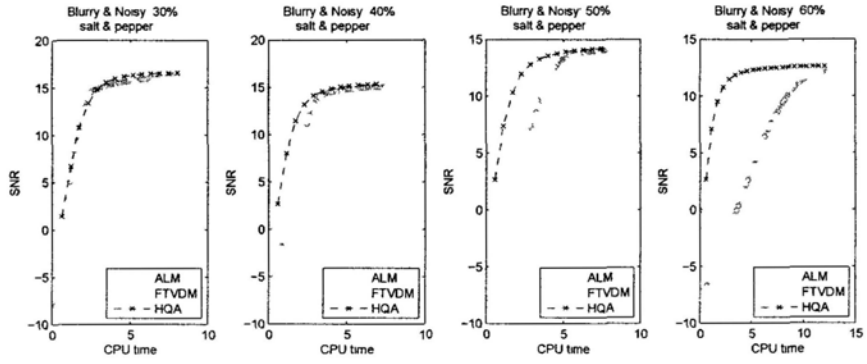

Figure 3.9: SNR versus CPU time in seconds for "Goldhill" with salt-and-pepper noise.

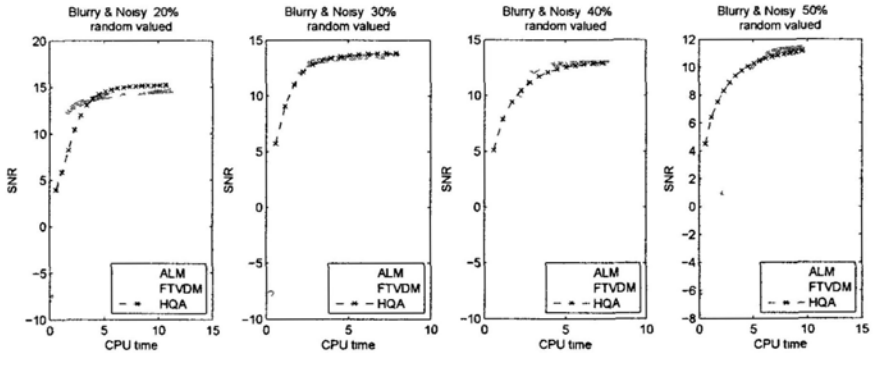

Figure 3.10: SNR versus CPU time in seconds for "Barbara" with random-valued noise.

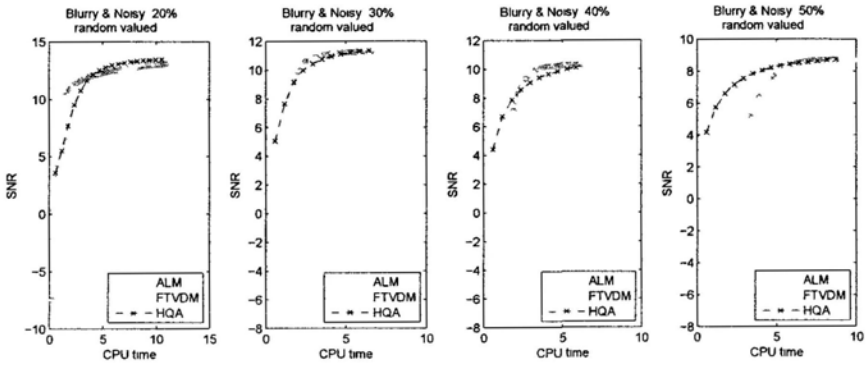

Figure 3.11: SNR versus CPU time in seconds for "Bridge" with random-valued noise.

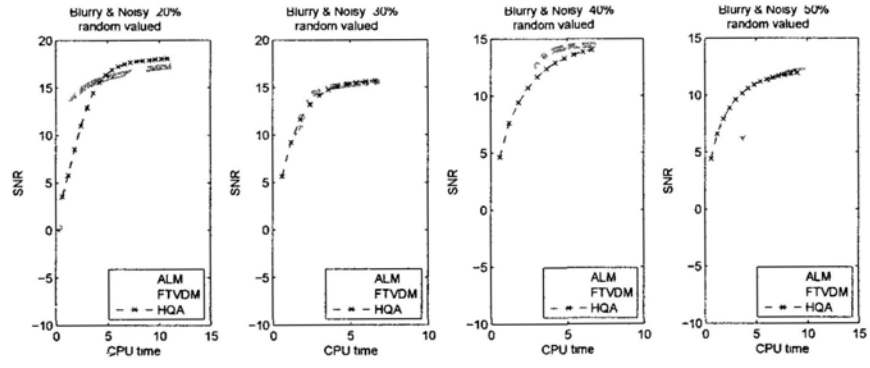

Figure 3.12: SNR versus CPU time in seconds for "Goldhill" with random-valued noise.

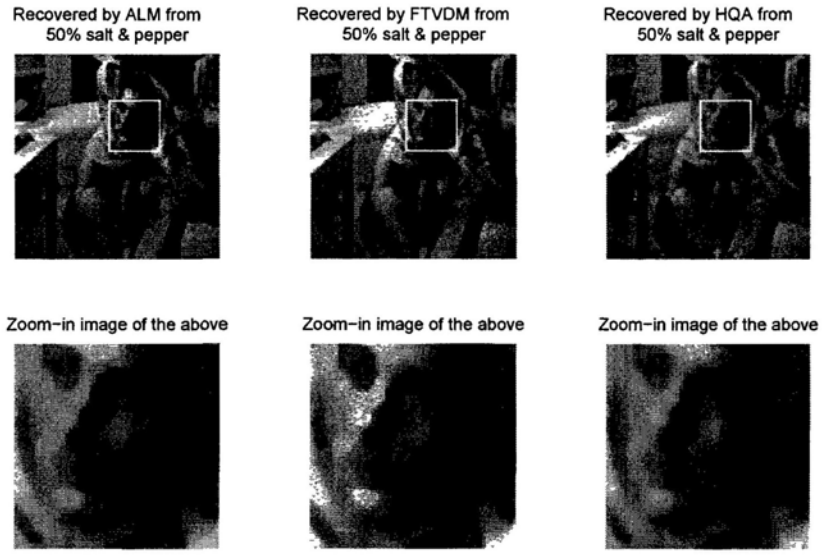

Figure 3.13: Zoom-in images of the recovered Barbara images.

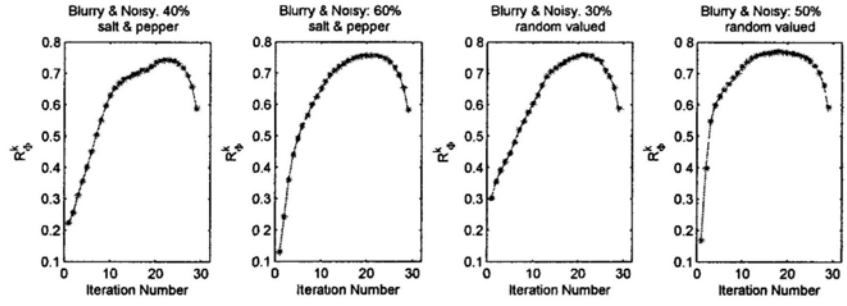

Figure 3.14: The ratio $R_\Phi^k := \frac{\Phi(\mathbf{u}^{k+1})-\Phi(\mathbf{u}^{30})}{\Phi(\mathbf{u}^{k})-\Phi(\mathbf{u}^{30})}$ versus iteration number for "Barbara". The ratios are less than 1, illustrating the linear convergence of $\{\Phi(\mathbf{u}^k)\}$.

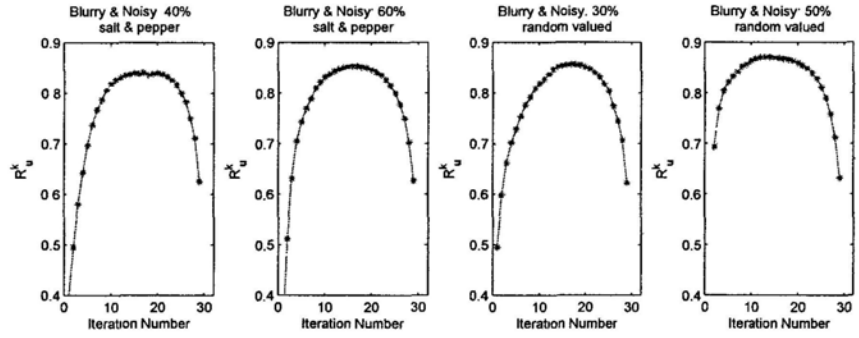

Figure 3.15: The ratio $R_{\mathbf{u}}^k = \frac{\|\mathbf{u}^{k+1}-\mathbf{u}^{30}\|_2}{\|\mathbf{u}^k-\mathbf{u}^{30}\|_2}$ versus iteration number for "Barbara". The ratios are less than 1, illustrating the linear convergence of $\{\mathbf{u}^k\}$.

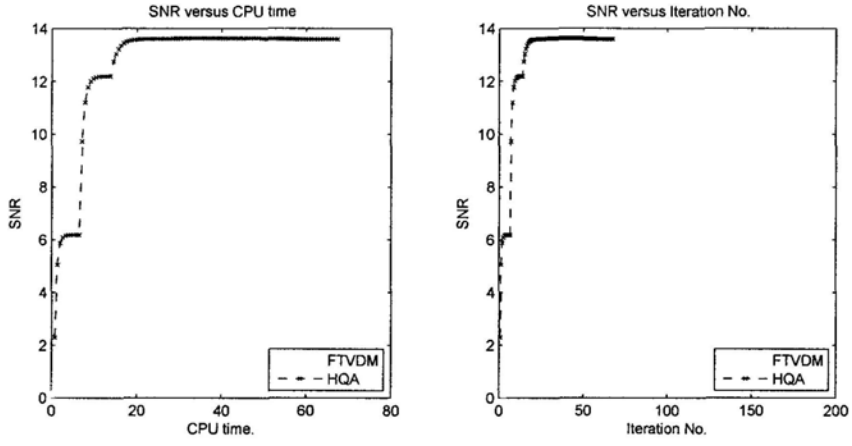

Figure 3.16: The figure shows the comparison results of the FTVDM and the HQA for Barbara image with salt-and-pepper noise removal at noise level 40%. For HQA, in (3.2), $\beta = 10^{-1}, 10^{-2}, \cdots, 10^{-17}$, and correspondingly $\gamma = \beta^2$. For FTVDM, in (2.2), $\theta_{\mathbf{w}} = 1, 2^{2/3}, \cdots, 2^{10}$, correspondingly, $\theta_{\mathbf{z}} = 1, 2, \cdots, 2^{15}$. At each jump, $\beta_k, \gamma_k, \theta_{\mathbf{w}}^k, \theta_{\mathbf{z}}^k$ jump to $\beta_{k+1}, \gamma_{k+1}, \theta_{\mathbf{w}}^{k+1}, \theta_{\mathbf{z}}^{k+1}$ in HQA and FTVDM.

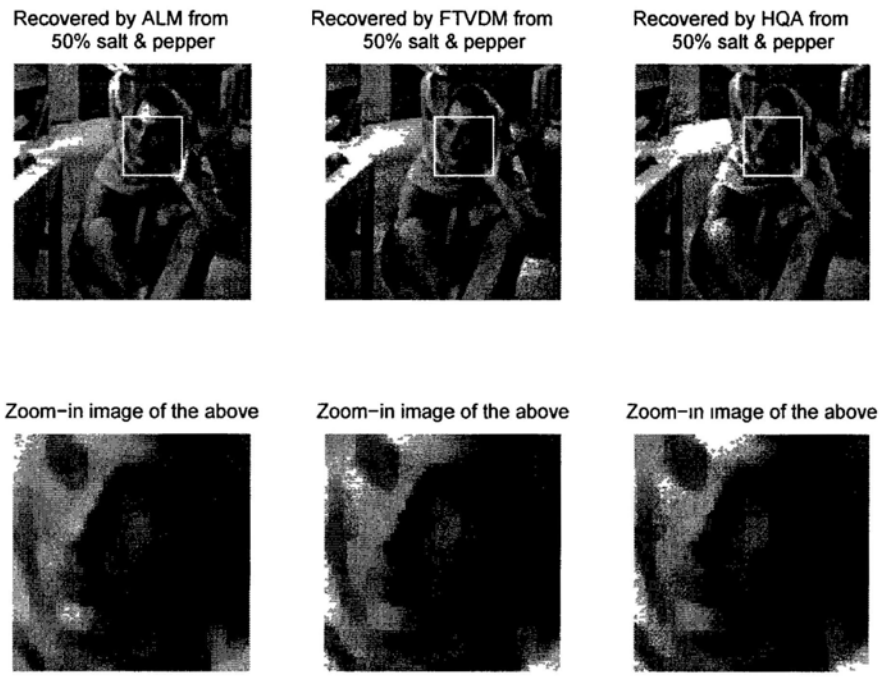

Figure 3.17: Zoom in images of the recovered Barbara image. The recovered image by the HQA is the final recovered at $\beta = 10^{-17}, \gamma = \beta^2$.

Chapter 4

A Proximal Point Half-Quadratic Algorithm for TV-L1 Image Restoration

4.1 Introduction

We follow the notations in Chapter 3 in this chapter and repeat the TV-L1 minimization problem again here:

$$\min_{\mathbf{u}}\{\lambda\|\mathbf{u}\|_{TV} + \|B\mathbf{u} - \mathbf{f}\|_1\}. \tag{4.1}$$

In Chapter 3, we proposed a fast and efficient algorithm HQA based on the half-quadratic technique to solve (4.1). There, the convergence analysis of the HQA is given under the assumption $\ker(\Delta)\cap\ker(B^TB) = \{\mathbf{0}\}$, where Δ is the Laplacian operator. In addition, without such an assumption, the condition number of the coefficient matrix in (3.11) is possible to be very large. To decrease its condition number, in this chapter, we modify the HQA with the PPA [43] and obtain the PHA. As what has been done for constructing HQA, we replace the TV and the

L1 functionals by differentiable functionals and then (4.1) turns to

$$\min_{\mathbf{u}}\{\sum_{i=1}^{n}\mu\sqrt{[(A_1)_i\mathbf{u}]^2+[(A_2)_i\mathbf{u}]^2+\beta}+\sqrt{(B_i\mathbf{u}-f_i)^2+\gamma}\}, \tag{4.2}$$

where B_i is the i-th row of B, and β, γ are both small regularization parameters. The half-quadratic technique is applied to solve (4.2) and the HQA is then constructed by updating

$$\mathbf{u}^{k+1}=\arg\min\{\Phi_{HQ}(\mathbf{u},\mathbf{u}^k)\}, \tag{4.3}$$

where $\Phi_{HQ}(\mathbf{u},\mathbf{u}^k):=\mathcal{L}(\mathbf{u},\mathbf{v}^{k+1},\mathbf{w}^{k+1})$ (see (3.8)). We then apply the PPA to (4.3) and obtain the PHA:

$$\mathbf{u}^{k+1}=\arg\min\{\Phi_{HQ}(\mathbf{u},\mathbf{u}^k)+\frac{1}{2\alpha_k}\|\mathbf{u}-\mathbf{u}^k\|_2^2)\}.$$

We prove that the PHA is indeed an MMA [29, 30, 12] for solving (4.2) and hence some convergence result of PHA is easily obtained. The proof is similar to the convergence proof of the lagged diffusivity fixed point iteration in [12]. The convergence of the PHA is obtained without the assumption $\ker(\Delta)\cap\ker(B^TB)=\{\mathbf{0}\}$ under which the HQA is convergent and linear convergent. To test our algorithm, we apply it to deblur images that are corrupted by impulse noise [51]. The results show that PHA is comparable to the HQA in both recovered effect and computing consuming. Comparing with augmented Lagrangian method (ALM) and the fast total variational decomposition method (FTVDM), the PHA is faster and more accurate than ALM and FTVDM for salt-and-pepper noise and comparable to the two methods for random-valued impulse noise. Furthermore, the recovered images by our method are less blocky as HQA, while the recovered images by ALM and FTVDM are blocky.

The outline of the chapter is as follows: In §3.3, we review the HQA [?] and the PPA [43], then we apply them to (4.2) to derive PHA. In §3.4, we prove the convergence of PHA based on the convergence theory of the MMA. Comparison with FTVDM, ALM and HQA on image deblurring is given in §3.5.

4.2 The Derivation of the PHA

In this section, We review the HQA and the PPA [43] first. Then we modify the HQA by PPA to construct our PHA.

Recalling that the algorithm of HQA in Chapter 3 is as follows:

(i) *Initialize* $\mathbf{u}^0$;

(ii) *For* $k = 1, 2, \cdots$ *until convergence, update* $\mathbf{u}^{k+1}$ by solving (3.11).

Notice that the coefficient matrix $\lambda A^T \tilde{D}_\beta(\mathbf{u}^k)A\mathbf{u} + B^T D_\gamma(\mathbf{u}^k)B$ on the left hand side of (3.11) may be ill-conditioned. In order to decrease its condition number, we introduce the PPA [43] into the HQA. The PPA is to solve a minimization problem $\min_{\mathbf{u}} \Psi(\mathbf{u})$ by $\mathbf{u}^{k+1} = \arg\min_{\mathbf{u}}\{\Psi(\mathbf{u}) + \frac{1}{2\alpha_k}\|\mathbf{u} - \mathbf{u}^k\|_2^2\}$. We combine PPA and (3.8) to obtain our PHA, i.e. we update $\mathbf{u}^{k+1}$ by

$$\mathbf{u}^{k+1} = \arg\min_{\mathbf{u}}\{\mathcal{L}(\mathbf{u}, \mathbf{v}^{k+1}, \mathbf{w}^{k+1}) + \frac{1}{2\delta}\|\mathbf{u} - \mathbf{u}^k\|_2^2\} \tag{4.4}$$

instead of by (3.8) in HQA. Here $\mathbf{u}^{k+1}$ is then the solution of

$$0 = \nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}, \mathbf{v}^{k+1}, \mathbf{w}^{k+1}) + \frac{1}{\delta}(\mathbf{u} - \mathbf{u}^k).$$

Substituting $\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}, \mathbf{v}^{k+1}, \mathbf{w}^{k+1})$ in (3.10) into the above equation, we obtain the following linear system for updating $\mathbf{u}^{k+1}$:

$$\lambda A^T \tilde{D}_\beta(\mathbf{u}^k)A\mathbf{u} + B^T D_\gamma(\mathbf{u}^k)B\mathbf{u} + \frac{1}{\delta}\mathbf{u} = B^T D_\gamma(\mathbf{u}^k)\mathbf{f} + \frac{1}{\delta}\mathbf{u}^k. \tag{4.5}$$

Thus the coefficient matrix in (4.5) is then well-conditioned with suitable δ. The algorithm of PHA is given as follows:

1. *Initialize* $\mathbf{u}^0$;

2. *For* $k = 1, 2, \cdots$ *until convergence, update* $\mathbf{u}^{k+1}$ by solving (4.5).

4.3 Convergence of PHA

In this section, we give the convergence proof of PHA based on the convergence analysis of the MMA.

Recall that the MM optimization technique is to solve a minimization problem $\min_{\mathbf{u}} \Psi(\mathbf{u})$ by

$$\mathbf{u}^{k+1} = \arg\min_{\mathbf{u}}\{\mathcal{G}(\mathbf{u}, \mathbf{u}^k)\}, \tag{4.6}$$

where $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$, called a *tangent majorant function* of $\Psi(\mathbf{u})$ at $\mathbf{u}^k$, must satisfy (3.13), (3.14) and (3.15). For minimization problem (3.4), here we define

$$\mathcal{G}(\mathbf{u}, \mathbf{u}^k) = \mathcal{L}(\mathbf{u}, \mathbf{v}^{k+1}, \mathbf{w}^{k+1}) + \frac{1}{2\delta}\|\mathbf{u} - \mathbf{u}^k\|_2^2. \tag{4.7}$$

Substituting (3.9) into (4.7), we obtain the explicit form of $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$:

$$\begin{aligned}\mathcal{G}(\mathbf{u}, \mathbf{u}^k) = & \sum_{i=1}^{n}\Big[\lambda\Big(\frac{|\nabla u_i|_\beta^2}{2|\nabla u_i^k|_\beta} + \frac{|\nabla u_i^k|_\beta}{2}\Big) + \frac{|B_i\mathbf{u} - f_i|_\gamma^2}{2|B_i\mathbf{u}^k - f_i|_\gamma} + \frac{|B_i\mathbf{u}^k - f_i|_\gamma}{2}\Big] \\ & + \frac{1}{2\delta}\|\mathbf{u} - \mathbf{u}^k\|_2^2.\end{aligned} \tag{4.8}$$

We now show that the PHA is the MMA for solving (3.4) with $\Psi := \Phi$.

Lemma 4.3.1 *The PHA is the MMA for solving the TV-L1 regularized problem* (3.4) *with tangent majorant function* $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ *defined in* (4.8).

Proof: We need to verify that $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ defined in (4.8) satisfies (3.13)–(3.15). Taking $\mathbf{u} = \mathbf{u}^k$, we obtain $\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k) = \Phi(\mathbf{u}^k)$, which is (3.14). The inequality $a^2 + b^2 \geq 2ab$ for $\forall\ a, b \in \mathbb{R}$ yields

$$\mathcal{G}(\mathbf{u}, \mathbf{u}^k) \geq \sum_{i=1}^{n}(\lambda|\nabla u_i|_\beta + |B_i\mathbf{u} - f_i|_\gamma) + \frac{1}{2\delta}\|\mathbf{u} - \mathbf{u}^k\|_2^2 \geq \Phi(\mathbf{u}), \quad \forall \mathbf{u}.$$

Hence (3.13) holds. By taking the derivatives of $\Phi(\mathbf{u})$ and $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ with respect to $\mathbf{u}$, we have

$$\nabla\Phi(\mathbf{u}) = \lambda\nabla^T\tilde{D}_\beta(\mathbf{u})\nabla\mathbf{u} + B^T D_\gamma(\mathbf{u})(B\mathbf{u} - \mathbf{f}), \tag{4.9}$$

$$\nabla_1\mathcal{G}(\mathbf{u}, \mathbf{u}^k) = \lambda\nabla^T\tilde{D}_\beta(\mathbf{u}^k)\nabla\mathbf{u} + B^T D_\gamma(\mathbf{u}^k)(B\mathbf{u} - \mathbf{f}) + \frac{1}{\delta}(\mathbf{u} - \mathbf{u}^k). \tag{4.10}$$

Substituting $\mathbf{u} = \mathbf{u}^k$ into (4.9) and (4.10), we immediately have $\nabla\Phi(\mathbf{u}^k) = \nabla_1\mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)$. □

From (4.10), the Hessian matrix of $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ is given by

$$\nabla_1^2\mathcal{G}(\mathbf{u}, \mathbf{u}^k) = \lambda\nabla^T\tilde{D}_\beta(\mathbf{u}^k)\nabla + B^T D_\gamma(\mathbf{u}^k)B + \frac{1}{\delta}I. \tag{4.11}$$

Obviously, $\nabla_1^2\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ is symmetric positive definite with its smallest eigenvalue $\lambda_{\min} \geq 1/\delta$. It implies the strict convexity of $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ with respect to $\mathbf{u}$.

Lemma 4.3.2 *$\Phi(\mathbf{u})$ in (3.4) is convex in $\mathbf{u}$.*

Proof: By the definition of Φ in (3.4), Φ is obviously twice continuously differentiable and bounded from below by 0.

Taking derivatives on both sides of (4.9), we have

$$\nabla^2\Phi(\mathbf{u}) = \lambda\Big(\beta A^T\tilde{P}_\beta(\mathbf{u})A + T(\mathbf{u})^T P_\beta(\mathbf{u})T(\mathbf{u})\Big) + \gamma B^T P_\gamma(\mathbf{u})B, \tag{4.12}$$

where $\tilde{P}_\beta(\mathbf{u}) = \mathrm{diag}(P_\beta(\mathbf{u}), P_\beta(\mathbf{u}))$; $P_\beta(\mathbf{u})$ and $P_\gamma(\mathbf{u}) \in \mathbb{R}^{n\times n}$ are the diagonal matrices with their i-th diagonal entries being $1/|\nabla u_i|_\beta^3$ and $1/|B_i\mathbf{u} - f_i|_\gamma^3$ respectively; and $T(\mathbf{u}) \in \mathbb{R}^{n\times n}$ with $[T(\mathbf{u})]_{ij} = (A_2)_{ij}(A_1)_i\mathbf{u} - (A_1)_{ij}(A_2)_i\mathbf{u}$. Obviously, the Hessian matrix $\nabla^2\Phi(\mathbf{u})$ in (4.12) is symmetric, positive, and semi-definite, hence the convexity of $\Phi(\mathbf{u})$ is obtained. □

Similar to Theorem 3.3 in Chapter 3, we give the following convergence theorem for the PHA.

Theorem 4.3.3 *For the sequence $\{\mathbf{u}^k\}$ generated by PHA, we have*

1. *monotonic decreasing convergence of $\{\Phi(\mathbf{u}^k)\}$ i.e. $\Phi(\mathbf{u}^{k+1}) \leq \Phi(\mathbf{u}^k)$ and converges;*

2. *Suppose that the problem (4.2) has solutions, then $\{\Phi(\mathbf{u}^k\}$ strictly monotonicly decreasing (i.e. $\Phi(\mathbf{u}^{k+1}) < \Phi(\mathbf{u}^k)$ unless $\Phi(\mathbf{u}^k) = \min_{\mathbf{u}}\{\Phi(\mathbf{u})\}$) and thus $\lim_{k\to\infty}\Phi(\mathbf{u}^k) = \min_{\mathbf{u}}\{\Phi(\mathbf{u})\} := \Phi^*$.*

Proof:

1. By Lemma 4.3.1 and the definition of the majorant function, we immediately have

$$\Phi(\mathbf{u}^{k+1}) \leq \mathcal{G}(\mathbf{u}^{k+1}, \mathbf{u}^k) \leq \mathcal{G}(\mathbf{u}^k, \mathbf{u}^k) = \Phi(\mathbf{u}^k). \tag{4.13}$$

   Since $\Phi(\mathbf{u}^k)$ is bounded from below by 0 and is monotonically decreasing, the sequence $\{\Phi(\mathbf{u}^k)\}$ converges and

$$\lim_{k\to\infty} \left[\Phi(\mathbf{u}^k) - \Phi(\mathbf{u}^{k+1})\right] = 0. \tag{4.14}$$

2. Notice that $\min_{\mathbf{u}}\{\Phi(\mathbf{u})\} \geq \mathbf{0}$ and $\Phi(\mathbf{u})$ is differentiable. Suppose $\Phi(\mathbf{u}^k) \neq \min_{\mathbf{u}}\{\Phi(\mathbf{u})\}$, then $\nabla\Phi(\mathbf{u}^k) \neq \mathbf{0}$ and thus $\nabla_1\mathcal{G}(\mathbf{u}, \mathbf{u}^k)|_{\mathbf{u}=\mathbf{u}^k} \neq \mathbf{0}$ from (4.9) and (4.10). It implies that at $\mathbf{u}^k$, the majorant function $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ can not reach the minimum. Recall that

$$\mathbf{u}^{k+1} = \arg\min_{\mathbf{u}}\{\mathcal{G}(\mathbf{u}, \mathbf{u}^k)\},$$

   we then have $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)|_{\mathbf{u}=\mathbf{u}^{k+1}} = \mathbf{0}$ and $\mathcal{G}(\mathbf{u}^{k+1}, \mathbf{u}^k) < \mathcal{G}(\mathbf{u}^k, \mathbf{u}^k)$ since $\mathcal{G}(\mathbf{u}, \mathbf{u}^k)$ is a strictly convex and continuous function with respect to $\mathbf{u}$. From the definition of the majorant function, we obtain an inequality

$$\Phi(\mathbf{u}^{k+1}) \leq \mathcal{G}(\mathbf{u}^{k+1}, \mathbf{u}^k) < \mathcal{G}(\mathbf{u}^k, \mathbf{u}^k) = \Phi(\mathbf{u}^k).$$

   It is said that sequence $\{\Phi(\mathbf{u}^k)\}$ is strictly monotonically decreasing and stop until $\Phi(\mathbf{u}^k)$ reaches the minimum value or $\{\Phi(\mathbf{u}^k)\}$ is strictly monotonically decreasing and converges to Φ^*.

4.4 Numerical Examples

In this section, we apply our algorithm to deblur images that are all corrupted by impulse noise. The deblurring problem has been discussed recently in many

papers, see for examples [18, 52, 51] and Chapter 3. Among all these methods, the FTVDM, the ALM and the HQA are the most efficient; and according to the numerical results in [51] and Chapter 3, ALM and HQA are the fastest ones. Hence in this chapter, we compare our PHA with the FTVDM, ALM and HQA. The FTVDM and ALM codes we used here are provided by the authors in [51]. For ALM, in (2.6), parameters $\gamma_{\mathbf{w}}$ and $\gamma_{\mathbf{z}}$ are fixed to be $\gamma_{\mathbf{w}} = 10$ and $\gamma_{\mathbf{z}} = 100$ in all the experiments. For FTVDM, the parameters $\theta_{\mathbf{w}}$ and $\theta_{\mathbf{z}}$ are upper limited to be $\theta_{\mathbf{w}} = 2^{10}, \theta_{\mathbf{z}} = 2^{15}$ in the approximate problem (2.2). Using $\theta_{\mathbf{w}} > 2^{10}, \theta_{\mathbf{z}} > 2^{15}$ would only increase the computational cost but not solution quality. To speed up the convergence, $\theta_{\mathbf{w}}$ and $\theta_{\mathbf{z}}$ are both implemented in a continuous scheme; that is, let $\theta_{\mathbf{w}}$ and $\theta_{\mathbf{z}}$ take small values at the beginning and gradually increase their values to 2^{10} and 2^{15} respectively. Specially, a $\theta_{\mathbf{w}}$-sequence $2^0, 2^{2/3}, 2^{4/3}, \cdots, 2^{10}$ is tested. Accordingly, $\theta_{\mathbf{z}}$ is set to be $2^0, 2^1, 2^2, \cdots, 2^{15}$.

We test three 256-by-256 images: Barbara, Bridge and Goldhill. The matrix B is the blurring matrix corresponds to the Gaussian blur generated by the MATLAB command

```
fspecial('Gaussian', [7 , 7], 5).
```

Then impulse noise is added to the blurred image to obtain the observed image $\mathbf{f}$. The noise levels are 30%, 40%, 50%, 60% for salt-and-pepper noise and 20%, 30%, 40%, 50% for random-valued noise. We use the same regularization parameter λ for all algorithms at the same noise level. For salt-and-pepper noise, $\lambda = 1/13, 1/10, 1/8, 1/4$ for noise level $30\%, 40\%, 50\%, 60\%$ respectively. For random-valued noise, $\lambda = 1/25, 1/10, 1/8, 1/4$ for noise level $20\%, 30\%, 40\%, 50\%$. In HQA and PHA, we take $\mathbf{u}_0 = \text{rand}(\text{size}(\mathbf{f}))$. As in the FTVDM, similar settings to β and γ are used in HQA. We set β-sequence $10^{-3}, 10^{-4}, \cdots, 10^{-16}$ in (3.2). Corresponding, we set $\gamma = \beta^2$. To speed up the convergence and improve the resolution quality, we let β, γ take large values at the beginning and decrease their values to 10^{-10} and 10^{-20} respectively. Equation (3.11) and (4.5) are solved by

the conjugate gradient method where we fix the maximum number of iterations to 10. In all tests, we also consider periodic boundary condition for the difference matrix A as in Chapter 3 in order to compare the PHA with the ALM [51] and FTVDM [52], as it is the boundary condition used in the tests in [52, 51] and Chapter 3. We compare the accuracy of the methods by the signal-to-noise ratio (SNR) used in [51]. It is defined by

$$\text{SNR} := 10 \log_{10} \frac{\|\mathbf{u} - E(\mathbf{u})\|_2^2}{\|\hat{\mathbf{u}} - \mathbf{u}\|_2^2} (\text{dB}).$$

Here $\mathbf{u}$ and $\hat{\mathbf{u}}$ denote the original image and the restored image respectively, and $E(\mathbf{u})$ is the mean gray-level value of the original image.

First we compare the speed of the four methods. Figures 4.2–4.7 show the timing comparison of the four algorithms. Each point in the figures show the accumulated CPU time until that iteration and the corresponding SNR. For the deblurring problem with salt-and-pepper noise, see Figures 4.2–4.4, our method is comparable with the HQA in speed and is faster than the FTVDM and the ALM. It is also more accurate than the FTVDM and the ALM. For random-valued noise, when the noise level is low, PHA is comparable to the ALM. But when the noise level is higher, say 50%, PHA becomes faster than the ALM. In Figure 4.1, we zoom into part of the restored Barbara image by the three methods. We see that the images by the ALM and the FTVDM are both blocky whereas that by the PHA is less blocky.

From Figures 4.2–4.7, it is clear that FTVDM is the slowest amongst the three. In order to compare the accuracy of the three faster methods ALM, HQA and PHAmore precisely, we list in Table 4.1 the average SNR of the recovered images in five trials by the three methods. To compare the timing fairly, we first run ALM until it converges, say with t_0 CPU seconds. Then we let HQA (PHA) run until the CPU time of the kth iteration is just greater than t_0. Then we record the SNR of the $(k-1)$th iteration as our result for HQA (PHA). We see from Table 4.1 that HQA and the PHA are more accurate than ALM for the

deblurring problem with the salt-and-pepper noise; and it is comparable to ALM for random-valued noise, especially when the noise level is low.

Figure 4.8 shows us the difference between the recovered images by HQA and PHA as β, γ decrease to $10^{-17}, 10^{-34}$ for Barbara image with salt and pepper noise removal. The Frobenius norm of the difference images are 0.0393, 0.0292, 0.0133, 0.0458 respectively from noise level 30% to 60%. Figure 4.8 shows the numerical illustration that the HQA and the PHA converges to the same solution.

Image	Method	Salt-and-pepper noise				Random-valued noise			
		30%	40%	50%	60%	20%	30%	40%	50%
Barbara	ALM	13.93	13.35	12.45	11.37	14.63	13.61	12.91	11.24
	HQA	14.24	13.59	12.83	11.72	15.17	13.81	12.86	11.06
	PHA	14.21	13.55	12.73	11.69	15.16	13.77	12.7	10.95
Bridge	ALM	11.85	10.95	10.13	8.52	13.05	11.31	10.35	8.53
	HQA	12.03	11.12	10.27	9.00	13.46	11.37	10.13	8.72
	PHA	11.97	11.01	10.14	8.98	13.43	11.25	9.97	8.64
Goldhill	ALM	16.08	15.03	13.78	12.05	17.29	15.53	14.41	12.14
	HQA	16.50	15.32	14.10	12.72	18.02	15.68	13.98	11.98
	PHA	16.44	15.19	13.90	12.66	18.00	15.54	13.71	11.85

Table 4.1: SNR of the restored images.

4.5 Conclusions

In this chapter, we apply the proximal point algorithm to modify the HQA for solving the TV-L1 minimization problem. We analyze its convergence using knowledge of the majorize-minimize technique without the assumption that $\ker(\Delta) \cap \ker(B^T B) = \{\mathbf{0}\}$. Our algorithm is proven to be extremely efficient in terms of accuracy, computational time and the visual quality as demonstrated by the numerical experiments. Our future work is to extend our algorithm to color

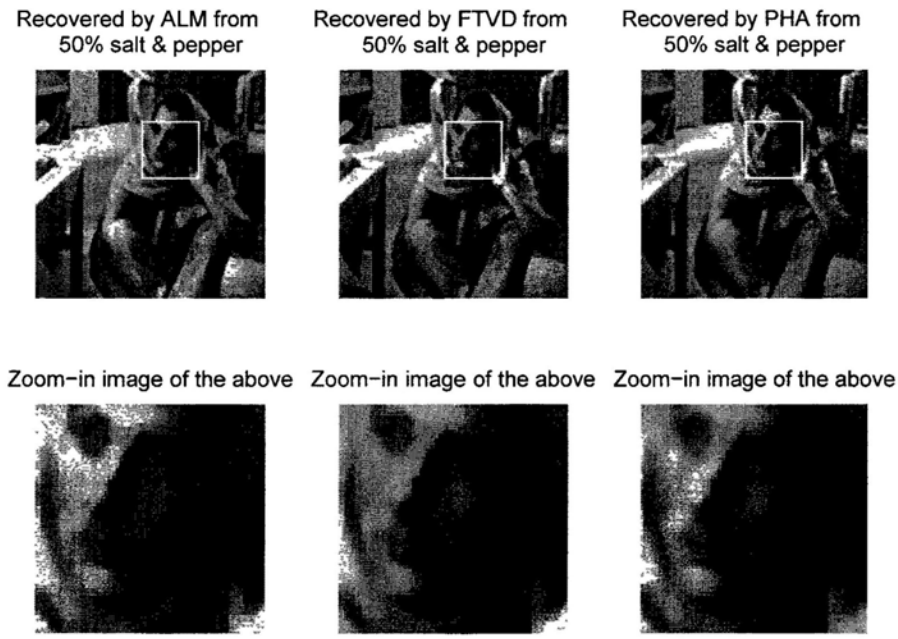

Figure 4.1: Zoom-in images of the recovered Barbara images by ALM and PHA.

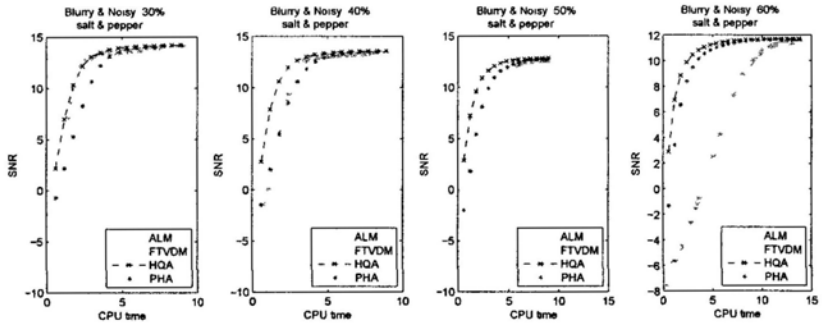

Figure 4.2: SNR versus CPU time in seconds for "Barbara" with salt-and-pepper noise.

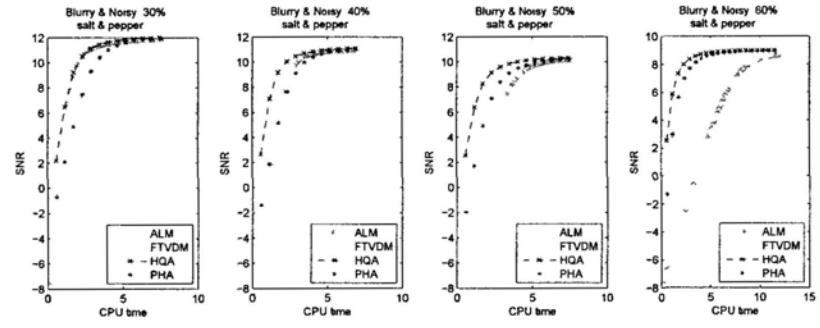

Figure 4.3: SNR versus CPU time in seconds for "Bridge" with salt-and-pepper noise.

image restoration via TV and L1 data fittings and to general Lp-regularized problems.

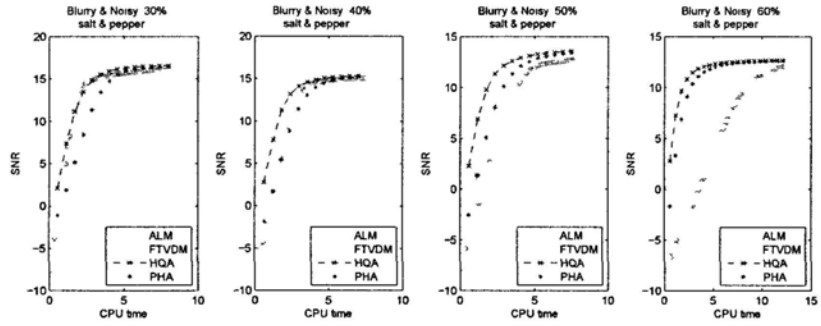

Figure 4.4: SNR versus CPU time in seconds for "Goldhill" with salt-and-pepper noise.

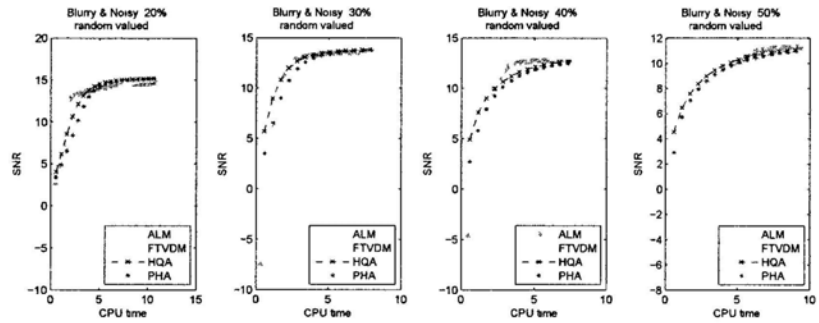

Figure 4.5: SNR versus CPU time in seconds for "Barbara" with random-valued noise.

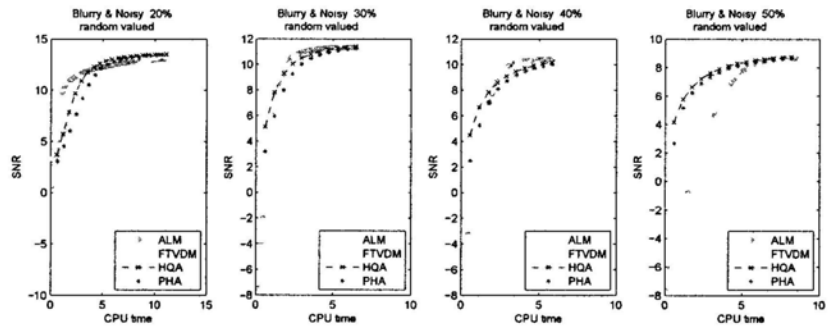

Figure 4.6: SNR versus CPU time in seconds for "Bridge" with random-valued noise.

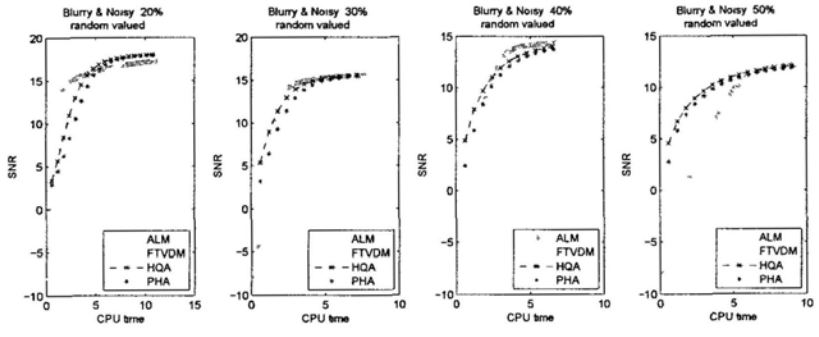

Figure 4.7: SNR versus CPU time in seconds for "Goldhill" with random-valued noise.

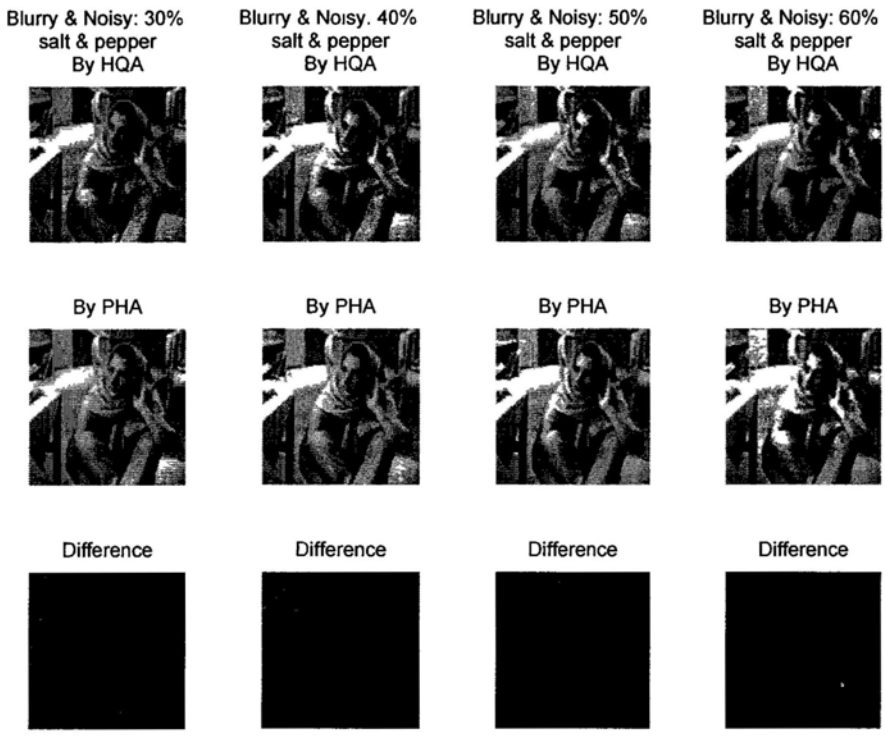

Figure 4.8: The figure shows the recovered images by HQA and the PHA and their difference as β, γ decrease to $10^{-17}, 10^{-34}$ in (3.2) for Barbara image. The first row shows the recovered images by the HQA for salt-and-pepper noise removal at noise level 30% to 60%. The second row shows the recovered images by the PHA. The third row shows the absolute value image of the difference between the first row and the second row. Form left to right, the Frobenius norm of the third row is 0.0393, 0.0292, 0.0133, 0.0458 respectively.

Chapter 5

Positively Constrained Minimum Penalized Total Variation Image Restoration

5.1 Introduction

The total variation (TV) image restoration is an important image processing method due to its ability in preserving sharp edges in the restored image [6, 49, 46]. The method is usually formulated as a penalized least squares with the TV penalty function [50, 49]. The existing numerical methods for solving this TV optimization problem include: partial differential equation (PDE) [45], primal-dual method using either Newton [10] or conjugate gradient [49] optimizations, etc. However, these methods do not consider the positivity constraint imposed on the restored image, and usually the pixel values are guaranteed to be nonnegative only in the last iteration by a simple projection or a scaling. In this chapter, we propose a new algorithm for solving TV penalized, and positively constrained, image restoration problems. Our approach uses the multiplicative iterative (MI) algorithm originally developed for tomographic image reconstruction [34] under

quadratic penalty terms. Here we modify it to solve the TV penalized image restoration problem. Moreover, we extend the error probability model from the traditional Gaussian distribution to other probability distributions, such as Poisson and Laplace (equivalent to L_1 norm) distributions.

The following notations are used throughout this chapter. Let $\mathcal{U} \in \mathbb{R}_+^{p\times q}$ be the unknown image which requires to be estimated and $\mathcal{F} \in \mathbb{R}_+^{p\times q}$ be the observed blurry image, where $\mathbb{R}_+$ denotes the positive orthant of $\mathbb{R}$. To simplify, images $\mathcal{U}$ and $\mathcal{F}$ are lexicographically ordered into vectors, and we let $\mathbf{u} = (u_1, \ldots, u_n)^T$ denote the vectorized $\mathcal{U}$ and $\mathbf{f} = (f_1, \ldots, f_n)^T$ denote the vectorized $\mathcal{F}$, where superscript T represents matrix transpose and $n = p \cdot q$. We will use these two notations of an image (i.e. 2D $\mathcal{U}$ and its vectorized $\mathbf{u}$) interchangeably when there is no confusion; an operation on $\mathbf{u}$ can be defined using $\mathcal{U}$, and vice versa. For a function $g(\mathbf{u})$ we use $\nabla g(\mathbf{u})$ to denote derivative of g with respect to $\mathbf{u}$ and $\nabla_j g(\mathbf{u})$ the derivative of g with respect to u_j.

In this chapter, we consider the statistical image restoration. Suppose the true (unobserved) image $\mathbf{u}$ is distorted by a blurring mechanism (such as a point spread function) which is denoted by an $n \times n$ matrix B. The expected observed image, denoted by an n-vector $\boldsymbol{\mu}$, is given by

$$\boldsymbol{\mu} = B\mathbf{u}.$$

However, due to noise contamination, we cannot observe $\boldsymbol{\mu}$ directly; instead, we observe the blurred noisy image $\mathbf{f}$. Our aim is to restore $\mathbf{u}$ from $\mathbf{f}$.

Statistical image restoration depends on the assumed probability model for the observed image and the penalty function (also known as the log prior density function). The penalty is used to restrict the restored image so that it follows certain local smoothness patterns. Assume that f_i are independent (given $\mathbf{u}$) and each follows a probability model, i.e.

$$f_i \sim p_i(f_i \,|\, \mu_i),$$

where $p_i(\cdot)$ represents the probability density function (pdf) of f_i and $\mu_i = B_i\mathbf{u}$ with B_i being the ith row of B. The penalized negative log-likelihood objective function $\Psi(\mathbf{u})$ for recovering $\mathbf{u}$ is given by

$$\Psi(\mathbf{u}) = -\sum_{i=1}^{n} l_i(\mu_i) + \lambda J(\mathbf{u}), \tag{5.1}$$

where $l_i(\mu_i) = \log p_i(f_i \mid \mu_i)$, $\lambda > 0$ is the smoothing parameter and $J(\mathbf{u})$ is the penalty function. The restored image $\hat{\mathbf{u}}$ is given as the minimizer of $\Psi(\mathbf{u})$ subject to $\mathbf{u} \geq 0$, namely

$$\hat{\mathbf{u}} = \underset{\mathbf{u}\geq 0}{\operatorname{argmin}}\, \Psi(\mathbf{u}). \tag{5.2}$$

The first term of (5.1) represents the negative log-likelihood and it measures fidelity of the restored blurry image $\hat{\boldsymbol{\mu}} = B\hat{\mathbf{u}}$ to the observed image $\mathbf{f}$. Its second term, on the other hand, measures smoothness of the restored image $\hat{\mathbf{u}}$. The smoothing parameter λ is included for the purpose of controlling the amount of smoothness of the restored image. A good smoothing parameter should balance well these two conflicting targets, namely data fidelity and smoothness. This chapter will not discuss how to select an optimal smoothing value for λ. Our focus is on how to compute efficiently the solution to problem (5.2) and how to obtain a good approximation to the target image.

The form of data fitting term in (5.1) depends on the statistical noise model. Three kinds of common noise models are as follows:

1. Gaussian noise model: Observed image intensities $f_i \sim N(\mu, \sigma^2)$. In this model, after combining variance σ^2 with λ to form a new smoothing parameter, we have

$$l_i(\mu_i) = -\frac{1}{2}(f_i - \mu_i)^2. \tag{5.3}$$

2. Poisson noise model: Observed image intensities $f_i \sim \text{Poisson}(\mu_i)$. This noise model gives

$$l_i(\mu_i) = -\mu_i + f_i \log \mu_i. \tag{5.4}$$

3. Impulsive noise model: Observed image intensities $f_i \sim \text{Laplace}(\mu_i, \sigma^2)$. Similar to the Gaussian noise model example, after combining σ^2 with λ we can write

$$l_i(\mu_i) = -|f_i - \mu_i|. \tag{5.5}$$

In this chapter, we consider the regularization function $J(\cdot)$ being the TV penalty function. Let $\|\cdot\|_2$ be the Euclidean norm of the relevant space and Ω be the domain of image $\mathcal{U}$. According to [49], the TV penalty $J(\mathbf{u})$ can be written as

$$J(\mathbf{u}) = \int_{(\xi_1,\xi_2)\in\Omega} \|\nabla \mathcal{U}\|_2 d\xi_1 d\xi_2 \approx \sum_{j=1}^{n} \sqrt{(R_j\mathbf{u})^2 + (C_j\mathbf{u})^2 + \beta}, \tag{5.6}$$

where R_j and C_j are respectively the jth row of the $n \times n$ matrices R and C. The entries of $R\mathbf{u}$ and $C\mathbf{u}$ represent the first-order differences of $\mathcal{U}$ along the row and column directions respectively. If $\mathbf{u}_j$ in $\mathbf{u}$ corresponds to $\mathcal{U}_{s,t}$ in $\mathcal{U}$, then

$$R_j\mathbf{u} = \begin{cases} \mathcal{U}_{s+1,t} - \mathcal{U}_{s,t} & \text{for } 1 \le s \le m-1 \\ \mathcal{U}_{1,t} - \mathcal{U}_{m,t} & \text{for } s = m \end{cases}$$

$$C_j\mathbf{u} = \begin{cases} \mathcal{U}_{s,t+1} - \mathcal{U}_{s,t} & \text{for } 1 \le t \le m-1 \\ \mathcal{U}_{s,1} - \mathcal{U}_{s,m} & \text{for } t = m \end{cases}$$

In (5.6), parameter $\beta > 0$ is included to avoid degenerate derivative of $J(\mathbf{u})$.

We will introduce a multiplicative iterative (MI) algorithm to solve the constrained optimization problem (5.2) where the penalty function $J(\mathbf{u})$ is TV. We call this new method the MITV algorithm. The advantages of MITV are that it is very easy to derive and implement under different image noise models, such as Gaussian, Poisson and impulsive, and it respects the positivity constraint. In the numerical tests, we apply our algorithm to deblur images corrupted with Gaussian noise. The results show that our method gives better restored images than the forward-backward splitting algorithm.

The rest of this chapter is arranged as follows. §5.2 reviews the forward-backward splitting algorithm for TV image restoration. It is the algorithm that

we will use to compare with MITV. §5.3 develops our new MITV image restoration algorithm. Two test images with Gaussian noises are used to compare MITV with forward-backward splitting, and the results are given in §5.4. Finally, concluding remarks are provided in §5.5.

5.2 The Forward-Backward Splitting Algorithm

In this section, we explain the forward-backward splitting (FBS) algorithm (e.g. [16, 4]) for solving problem (5.2) for the Gaussian noise model. This algorithm was designed without considering the positivity constraint. To obtain a positive solution, however, one usually project the iteration results into the nonnegative half space at every iteration or at the last iteration.

Let $H(\mathbf{u}) = -\sum_{i=1}^{n} l_i(\mu_i) = \frac{1}{2}\|B\mathbf{u} - \mathbf{f}\|_2^2$. We begin with the definition of the proximity operator. For any proper, convex and semi-continuous function $\phi(\cdot)$ with range $(-\infty, +\infty]$, its proximity operator is defined by

$$\text{prox}_{\delta\phi} \ : \mathbf{u} \rightarrow \underset{\mathbf{v}}{\text{argmin}}\{\frac{1}{2\delta}\|\mathbf{u} - \mathbf{v}\|_2^2 + \phi(\mathbf{v})\}. \tag{5.7}$$

The FBS algorithm can be used to solve the minimization problem of the following form:

$$\min_{\mathbf{u}}\{F_1(\mathbf{u}) + F_2(\mathbf{u})\}, \tag{5.8}$$

where F_1 is a proper, convex, lower semi-continuous function and F_2 is a convex, $1/\gamma$-Lipschitz continuous differentiable function. The FBS iteration formula for solving (5.8) is given as follows:

$$\mathbf{u}^{k+1} = \text{prox}_{\delta F_1}(\mathbf{u}^k - \delta\nabla F_2(\mathbf{u}^k)). \tag{5.9}$$

In [16], the authors show that this FBS algorithm converges to the solution of (5.8) when $0 < \delta < 2\gamma$.

We can apply the FBS algorithm to solve problem (5.2) with the Gaussian noise model, where the penalty term $F_1(\mathbf{u}) = \lambda J(\mathbf{u})$ and the data fidelity term $F_2(\mathbf{u}) = H(\mathbf{u})$ respectively. In this case, the Lipschitz constant $\gamma = 1/\|B^TB\|_2$. The gradient of the data fidelity term is: $\nabla H(\mathbf{u}) = B^T(B\mathbf{u} - \mathbf{f})$. The FBS iteration formula (5.9) is then given by

$$\mathbf{u}^{k+1} = \text{prox}_{\delta(\lambda J)}(\mathbf{u}^k - \delta\nabla H(\mathbf{u}^k)) \tag{5.10}$$

from any initial $\mathbf{u}^0$. By definition of the proximity operator (5.7), the iteration formula (5.10) is equivalent to

$$\mathbf{u}^{k+1} = \underset{\mathbf{u}}{\text{argmin}}\{\lambda J(\mathbf{u}) + \frac{1}{2\delta}\|\mathbf{u} - (\mathbf{u}^k - \delta\nabla H(\mathbf{u}^k))\|_2^2\}. \tag{5.11}$$

The FBS algorithm (5.11) converges to the solution of (5.2) for Gaussian noise removal if $\delta \in (0, \frac{2}{\|B^TB\|_2})$. Recall that $J(\mathbf{u})$ is the TV penalty function, so we can apply the Chambolle's denoising algorithm [7] to obtain the minimizer of (5.11) at each step. Interested readers can consult [16] for more details of the FBS algorithm, [7] for general idea of the Chambolle's denoising algorithm, and [27, 35, 36] for more background knowledge of convex analysis.

5.3 Multiplicative Iterative TV Penalized Image Restoration

5.3.1 Derivation of the Algorithm

In image processing, pixel values should be nonnegative numbers. In [25, 37], the authors discussed the box-constrained minimization problem based on the Karush-Kuhn-Tucker (KKT) condition. The interested readers can consult [25] for the affine-scalling interior-point cyclic Barzilai-Borwein [1] method for box-constrained minimization problem and [37] for an a reduced Newton method for box-constrained linear least-squares problems. In [34], the author proposed a

multiplicative iterative algorithm (MI) for problems in tomographic reconstructions. There the penalty function is taken to be a quadratic one. In this section, we develop the MI algorithm for TV penalized and positively constrained image restoration. The algorithm is flexible and can be applied to different image noise models. It also begins with the Karush-Kuhn-Tucker (KKT) necessary conditions.

We first introduce some notations needed for the derivation of the MITV algorithm below. For any vector function $g(\mathbf{u})$, it can be separated into positive part vector $[g(\mathbf{u})]^+$ and negative part vector $[g(\mathbf{u})]^-$ with their jth entry defined by

$$[g(\mathbf{u})]_j^+ = \max\{[g(\mathbf{u})]_j, 0\}, \qquad [g(\mathbf{u})]_j^- = \max\{-[g(\mathbf{u})]_j, 0\}, \tag{5.12}$$

where $[g(\mathbf{u})]_j$ is the jth component of $g(\mathbf{u})$. By (5.12),

$$g(\mathbf{u}) = [g(\mathbf{u})]^+ - [g(\mathbf{u})]^-. \tag{5.13}$$

Obviously $[g(\mathbf{u})]^+ \geq \mathbf{0}$ and $[g(\mathbf{u})]^- \geq \mathbf{0}$. We call this separation **Type 1** separation that we use in the following part. If $g(\mathbf{u})$ has an explicit separation form, i.e. $g(\mathbf{u}) = \mathbf{a} - \mathbf{b}$ with $\mathbf{a} \geq \mathbf{0}$ and $\mathbf{b} \geq \mathbf{0}$, we just take $[g(\mathbf{u})]^+ = \mathbf{a}$ and $[g(\mathbf{u})]^- = \mathbf{b}$. We call this separation **Type 2** separation in the following part.

If $\hat{\mathbf{u}}$ solves optimization problem (5.2), then $\hat{\mathbf{u}}$ satisfies the KKT conditions:

$$\begin{aligned} \nabla_j \Psi(\hat{\mathbf{u}}) &= 0, \text{ if } \hat{u}_j > 0 \\ \nabla_j \Psi(\hat{\mathbf{u}}) &\geq 0, \text{ if } \hat{u}_j = 0 \end{aligned}$$

for $j = 1, \ldots, n$. Equivalently $\hat{\mathbf{u}}$ solves the following linear system

$$D(\mathbf{u})\nabla\Psi(\mathbf{u}) = \mathbf{0}, \tag{5.14}$$

where $D(\mathbf{u}) = \text{diag}(d_1(\mathbf{u}), \cdots, d_n(\mathbf{u}))$ with

$$d_j(\mathbf{u}) = \begin{cases} u_j, & \nabla_j \Psi(\mathbf{u}) = 0 \\ 0. & \nabla_j \Psi(\mathbf{u}) > 0 \end{cases}$$

Using expression (5.1), equation (5.14) is a linear system of

$$u_j \left(-\sum_{i=1}^{n} b_{ij} \nabla l_i(\mu_i) + \lambda \nabla_j J(\mathbf{u}) \right) = 0, \qquad j = 1, \cdots, n \tag{5.15}$$

where $b_{ij} \geq 0$ represents the (i,j)th element of matrix B. First we apply the separation (5.13) to $\nabla l_i(\mu_i)$ and $\nabla_i J(\mathbf{u})$, to get $\nabla l_i(\mu_i) = [\nabla l_i(\mu_i)]^+ - [\nabla l_i(\mu_i)]^-$ and $\nabla_j J(\mathbf{u}) = [\nabla_j J(\mathbf{u})]^+ - [\nabla_j J(\mathbf{u})]^-$. Then we rewrite equation (5.15) as

$$u_j \left(\sum_{i=1}^{n} b_{ij} [\nabla l_i(\mu_i)]^- + \lambda [\nabla_j J(\mathbf{u})]^+ \right) = u_j \left(\sum_{i=1}^{n} b_{ij} [\nabla l_i(\mu_i)]^+ + \lambda [\nabla_j J(\mathbf{u})]^- \right), \tag{5.16}$$

where both sides of this equation are now nonnegative.

Equation (5.16) naturally suggests an iterative scheme for solving (5 15), and that is:

$$u_j^{k+1/2} = u_j^k \frac{\delta_{1j}^k}{\delta_{2j}^k}, \qquad j = 1, \cdots, n \tag{5.17}$$

where

$$\begin{aligned} \delta_{1j}^k &= \sum_{i=1}^{n} b_{ij} [\nabla l_i(\mu_i^k)]^+ + \lambda [\nabla_j J(\mathbf{u}^k)]^- \\ \delta_{2j}^k &= \sum_{i=1}^{n} b_{ij} [\nabla l_i(\mu_i^k)]^- + \lambda [\nabla_j J(\mathbf{u}^k)]^+, \end{aligned}$$

and

$$\delta_{2j}^k - \delta_{1j}^k = \nabla_j \Psi(\mathbf{u}^k). \tag{5.18}$$

Here $\mu_i^k = B_i \mathbf{u}^k$ and $\nabla_j J(\mathbf{u}^k)$ denotes $\nabla_j J(\mathbf{u})$ evaluated at $\mathbf{u}^k$. In (5.17) the iteration index $k + 1/2$ explains that this is merely a temporary updating, and further improvements are necessary to give $\mathbf{u}^{k+1}$ by the following rule:

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \alpha^k \mathbf{d}^k.$$

Here $\alpha^k \in (0, 1]$ is a positive step size and the direction vector $\mathbf{d}^k$ is given componentwise by

$$d_j^k = u_j^{k+1/2} - u_j^k.$$

Substituting (5.17) and (5.18) into above equality, we have

$$d_j^k = -\frac{u_j^k}{\delta_{2j}^k}\nabla_j\Psi(\mathbf{u}^k). \tag{5.19}$$

Hence $\mathbf{d}^k$ is a down-hill direction for $\Psi(\mathbf{u})$ when the denominator δ_{2j}^k is nonzero for all j.

In order to facilitate (5.17) we must provide $\nabla_j J(\mathbf{u})$ for the TV penalty function $J(\mathbf{u})$ given in (5.6). It is not difficult to derive that

$$\nabla_j J(\mathbf{u}) = \sum_{t=1}^{n} \frac{r_{tj}R_t\mathbf{u} + c_{tj}C_t\mathbf{u}}{\sqrt{(R_t\mathbf{u})^2 + (C_t\mathbf{u})^2 + \beta}}. \tag{5.20}$$

To ensure that the iteration scheme (5.17) is well defined, we have to explain how to handle the possibility of $\delta_{2j}^k = 0$. This problem can be rectified simply by replacing any zero δ_{2j}^k with a constant ε (such as $\varepsilon = 10^{-2}$). In this case, however, the corresponding numerator of (5.17) must also be altered by $\delta_{1j}^k + \varepsilon$ so that the estimating equation (5.15) is still maintained.

It is possible that $\mathbf{u}^{k+1/2}$ by (5.17) does not decrease $\Psi(\mathbf{u})$, i.e. $\Psi(\mathbf{u}^{k+1/2}) \geq \Psi(\mathbf{u}^k)$. In this case we need a line search step to improve $\mathbf{u}^{k+1/2}$ such that the final update decreases the objective function $\Psi(\mathbf{u})$. In this line search, we must find an $0 < \alpha^k \leq 1$ such that

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \alpha^k(\mathbf{u}^{k+1/2} - \mathbf{u}^k), \tag{5.21}$$

and $\mathbf{u}^{k+1}$ satisfies $\Psi(\mathbf{u}^{k+1}) < \Psi(\mathbf{u}^k)$. Here α^k can be obtained efficiently by approaches such as step half or Armijo rule [31]. By (5.17) and (5.19), we know that (5.21) maintain all the zero components of $\mathbf{u}^k$ in $\mathbf{u}^{k+1}$.

We call the iterative scheme defined by equations (5.21) and (5.17), where the penalty function $J(\mathbf{u})$ and its derivative $\nabla J(\mathbf{u})$ are given by, respectively, (5.6) and (5.20), the MITV algorithm. This algorithm is very easy to be implemented in TV image restoration tasks, and it can handle any image noise model. A very attractive feature is that this algorithm respects the positivity constraint usually

imposed on image restoration problems. In fact, from any initial guess $\mathbf{u}^0 \geq 0$, we have $\mathbf{u}^{1/2} > 0$ by (5.17). We then immediately obtain that $\mathbf{u}^1 \geq 0$ since $\mathbf{u}^1 = (1 - \alpha^0)\mathbf{u}^0 + \alpha^0\mathbf{u}^1$ and $0 < \alpha^0 \leq 1$. By induction, we know that $\mathbf{u}^k$ is always non-negative if initial guess $\mathbf{u}^0 \geq 0$. From the study reported in Section 5.4, the MITV algorithm is very competitive with the existing forward-backward splitting TV denoising algorithm.

Under certain regularity conditions, the MITV algorithm is convergent, and moreover, it converges to the solution satisfying the KKT necessary conditions. The proof of the general convergence result is available in [34].

5.3.2 MITV Under Different Image Noise Models

In this section, we provide the details of the MITV algorithms when the observed image contains Gaussian, Poisson or Impulse noises. From (5.17), these MITV algorithms differ only due to the fact that the derivatives of the log density functions, i.e. $\nabla l_i(\mu_i)$, are different for different noise models.

Gaussian Noise Model

For the Gaussian noise model, according to the data fitting term (5.3), then the gradient $\nabla l_i(\mu_i) = f_i - \mu_i$, where $f_i \geq 0$ and $\mu_i > 0$. Use **Type 2** separation to $\nabla_i(\mu_i)$, we have $[\nabla l_i(\mu_i)]^+ = f_i$ and $[\nabla l_i(\mu_i)]^- = \mu_i$. In addition, we use **Type 1** separation to $\nabla J(\mathbf{u})$. Then iteration (5.17) becomes

$$u_j^{(k+1/2)} = u_j^{(k)} \frac{\sum_{i=1}^n b_{ij} f_i + \lambda[\nabla_j J(\mathbf{u}^{(k)})]^-}{\sum_{i=1}^n b_{ij}\mu_i + \lambda[\nabla_j J(\mathbf{u}^{(k)})]^+}, \tag{5.22}$$

for $j = 1, \cdots, n$. If the denominator of (5.22) is zero for a pixel j, then we add a threshold ε to both numerator and denominator of (5.22). Once $\mathbf{u}^{k+1/2}$ is obtained, we move to $\mathbf{u}^{k+1}$ as follows. If $\Psi(\mathbf{u}^{k+1/2}) < \Psi(\mathbf{u}^k)$ then $\mathbf{u}^{k+1} = \mathbf{u}^{k+1/2}$; otherwise, using a line search to find an $0 < \alpha^k < 1$ such that

$$\Psi(\mathbf{u}^k + \alpha^k(\mathbf{u}^{k+1/2} - \mathbf{u}^k)) < \Psi(\mathbf{u}^k),$$

and then set $\mathbf{u}^{k+1} = \mathbf{u}^k + \alpha^k(\mathbf{u}^{k+1/2} - \mathbf{u}^k)$.

Poisson Noise Model

For the Poisson noise model, from (5.4), the gradient $\nabla l_i(\mu_i) = -1 + f_i/\mu_i$. Any zero μ_i will make both $l_i(\mu_i)$ and $\nabla l_i(\mu_i)$ not well defined. If $\mu_i^k = 0$ happens in iterations we will replace μ_i^k in denominator by a small constant $\theta > 0$. Hence we can assume all $\mu_i \neq 0$ without loss of generality. Using **Type 2** separation to $\nabla l_i(\mu_i)$, we have $[\nabla l_i(\mu_i)]^+ = f_i/\mu_i$ and $[\nabla l_i(\mu_i)]^- = 1$. In addition, we use **Type 1** separation to $\nabla J(\mathbf{u})$. Then iteration formula (5.17) now becomes,

$$u_j^{(k+1/2)} = u_j^{(k)} \frac{\sum_{i=1}^n b_{ij} f_i/\mu_i^k + \lambda[\nabla_j J(\mathbf{u}^{(k)})]^-}{\sum_{i=1}^n b_{ij} + \lambda[\nabla_j J(\mathbf{u}^{(k)})]^+}, \tag{5.23}$$

for $j = 1, \ldots, n$. As in the Gaussian noise above, if the denominator of (5.23) is zero for a pixel j, then we add a threshold ε to both numerator and denominator of (5.23). Then calculate $\mathbf{u}^{k+1}$ from $\mathbf{u}^{k+1/2}$ by line search.

Impulsive Noise Model

We use the Laplace distribution to model impulsive noises so that $l_i(\mu_i) = -|f_i - \mu_i|$ according to (5.5). A problem with this model is that $l_i(\mu_i)$ is not differentiable at $\mu_i = f_i$. However, adopting the idea of Huber function [28], we may *define* the derivative of $l_i(\mu_i)$ to be

$$\begin{aligned} \nabla l_i(\mu_i) &= \begin{cases} 1 & \text{if } \mu_i < f_i - \gamma \\ -1 & \text{if } \mu_i > f_i + \gamma \\ \frac{f_i - \mu_i}{\gamma} & \text{if } f_i - \gamma \leq \mu_i \leq f_i + \gamma \end{cases} \\ &= I_{\frac{f_i-\mu_i}{\gamma}>1} + \frac{f_i - \mu_i}{\gamma} I_{0\leq\frac{f_i-\mu_i}{\gamma}\leq 1} - I_{\frac{f_i-\mu_i}{\gamma}<-1} + \frac{f_i - \mu_i}{\gamma} I_{-1\leq\frac{f_i-\mu_i}{\gamma}<0}, \end{aligned}$$

We apply **Type 1** separation to $\nabla l_i(\mu_i)$, then we have $[\nabla l_i(\mu_i)]^+ = I_{\frac{f_i-\mu_i}{\gamma}>1} + \frac{f_i-\mu_i}{\gamma} I_{0\leq\frac{f_i-\mu_i}{\gamma}\leq 1}$ and $[\nabla l_i(\mu_i)]^- = I_{\frac{f_i-\mu_i}{\gamma}<-1} + \frac{\mu_i-f_i}{\gamma} I_{-1\leq\frac{f_i-\mu_i}{\gamma}<0}$, where $I_{\mathcal{A}}$ is an indicator function for event $\mathcal{A}$: $I_{\mathcal{A}} = 1$ if $\mathcal{A}$ is true, and $I_{\mathcal{A}} = 0$ otherwise. The

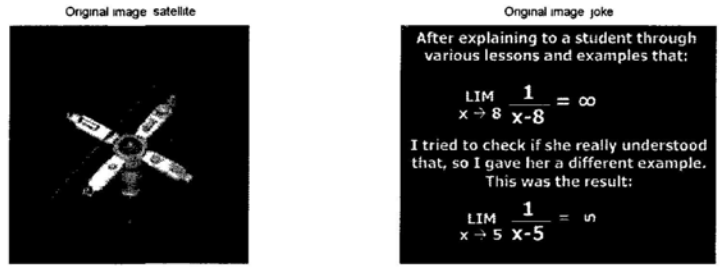

Figure 5.1: Original test images.

separation for $\nabla J(\mathbf{u})$ is still the **Type 1** separation. The updating formula (5.17) now becomes

$$u_j^{k+1/2} = u_j^k \frac{\sum_{i=1}^n b_{ij} \left(I_{\frac{f_i-\mu_i^k}{\gamma}>1} + \frac{f_i-\mu_i^k}{\gamma} I_{0\le\frac{f_i-\mu_i^k}{\gamma}\le 1} \right) + \lambda[\nabla_j J(\mathbf{u}^k)]^-}{\sum_{i=1}^n b_{ij} \left(I_{\frac{f_i-\mu_i^k}{\gamma}<-1} + \frac{\mu_i^k-f_i}{\gamma} I_{-1\le\frac{f_i-\mu_i^k}{\gamma}<0} \right) + \lambda[\nabla_j J(\mathbf{u}^k)]^+}, \quad (5.24)$$

for $j = 1, \ldots .n$. It is possible that at an iteration, the denominator of (5.24) is zero. In this case, we add a quantity $\varepsilon > 0$ to both numerator and denominator of (5.24). The update $\mathbf{u}^{k+1}$ is obtained from $\mathbf{u}^{k+1/2}$ by line search; see the Gaussian noise model example above.

5.4 Numerical Examples

In this section, we apply our MITV algorithm and the FBS algorithm to deblur images which are corrupted by Gaussian noises. For the Poisson and impulsive noise models, the augmented Lagrangian method (ALM) in [51] is a fast and efficient method. The interested readers can consult [51] for more details. Comparison of MITV with ALM on non-Gaussian noise models will be reported elsewhere.

In our tests, we use two images shown in Figure 5.1. The satellite image has size of 176×176, and the joke image has size of 284×378. We choose these

images because they both contain large zero backgrounds, so we expect that the positivity constraint is strongly informative in both examples. In the satellite image, there are 76.72% pixels equal to 0. In the joke image, this percentage is 86.50%. Due to such features, it is easier to test the effect of our algorithm in ensuring the positivity constraint.

In all the tests, matrix A is the blurring matrix corresponding to *motion* blur and is generated by the MATLAB command

```
fspecial('motion', 15, 30).
```

Gaussian noise is added to the blurry image to obtain the observed one. Denote the noise level by σ which represents the standard deviation of the Gaussian noise. We test two cases: one has noise level $\sigma = 5$, and the other one has noise level $\sigma = 10$.

Considering the good property of MITV in preserving positivity and the feature of our test images that most of the pixels equal to zero, we propose a projection in each MITV iteration:

$$u_j^{k+1} = \begin{cases} 0, & u_j^{k+1} < \eta \\ u_j^{k+1}, & \text{otherwise} \end{cases} \tag{5.25}$$

for $j = 1, \ldots, n$. We call this projection the lower projection, where η is called the lower projection parameter. This projection will help to further improve the restoration from our MITV algorithm. We call MITV with lower projection the PMITV algorithm. The optimal choice for η in all the tests is obtained experimentally. By trial and error, we find that the value of η almost increases in direct proportion to σ. In [37], Morini *et al* also discuss the box-constrained image restoration problems with projections. The interested readers can consult [37] for more details.

By trial and error, we find that for the satellite image the best smoothing parameter $\lambda = 0.4$ at noise level $\sigma = 5$, and the best smoothing parameter $\lambda = 1$

at noise level $\sigma = 10$. For the joke image, the best smoothing parameter $\lambda = 0.2$ at noise level $\sigma = 5$, and the best smoothing parameter $\lambda = 0.4$ at noise level $\sigma = 10$.

We use the peak signal-to-noise ratio (PSNR) to measure the quality of the restored images which is defined by

$$\text{PSNR} := 10 \log_{10} \frac{255^2}{\frac{1}{n}\|\hat{\mathbf{u}} - \mathbf{u}\|_2^2} (\text{dB}).$$

Here $\mathbf{u}$ and $\hat{\mathbf{u}}$ denote the original and restored images respectively, and n is the total pixel number of the image.

We show the comparison results of the PMITV algorithm and the FBS algorithm in Figures 5.2 – 5.5. Figures 5.2 and 5.3 show the observed and the recovered images by the PMITV and FBS algorithms. In PMITV, at noise level $\sigma = 5$, we choose $\eta = 4$; at noise level $\sigma = 10$, we choose $\eta = 9$. At each of these η's, the corresponding PMITV algorithm reaches the highest PSNR value among all the trials that we conducted. The PFBS algorithm is the FBS algorithm with a lower projection (5.25) at every iteration. Also from the trials, we find that 0 is the best choice for η in PFBS.

In the subfigures for the FBS algorithm in Figures 5.2 and 5.3, the PSNR records the PSNR value of the final iteration of the FBS algorithm without any projection, while the PPSNR records the PSNR value of FBS algorithm with a lower projection (5.25) only at the end of all the iterations, where the lower projection parameter is 0. At the final step in all the test algorithms, we take an upper projection

$$u_j^{rec} = \begin{cases} 255, & u_j^{rec} > 255 \\ u_j^{rec}, & \text{otherwise} \end{cases} \tag{5.26}$$

such that the recovered image is in $[0, 255]$. From Figures 5.2 and 5.3, we see that the PMITV always reach a higher PSNR value than PFBS (i.e. with projection) and FBS (without projection).

Figures 5.4 and 5.5 give computational time comparisons of the MITV algorithm and the FBS algorithm. From left to right, the first columns of these figures describe the time comparison of the energy value sequence $\{\Psi(\mathbf{u}^k)\}$. The second columns give the number of the pixels that are projected to 0 at each iteration in the PMITV algorithm. We observe that the number of pixels that are projected tends to stabilize as the iteration number increases. The third columns show the time comparison in relation to the PSNR value. We find that the PMITV always reaches higher PSNR value faster than the PFBS and FBS algorithms.

5.5 Conclusions

In this chapter, we develop and implement a new approach for total variation image restoration. Our method is based on the multiplicative iterative algorithm originally developed for tomographic image reconstruction. The advantages of our algorithm are that it is very easy to derive and implement under different noise models and it respects the positivity constraint. We discuss in the chapter how to apply this method to Gaussian, Poisson and impulsive noise models. In the numerical test, we apply our algorithm to deblur images corrupted with Gaussian noise. The results show that our method gives better restored images than the FBS algorithms.

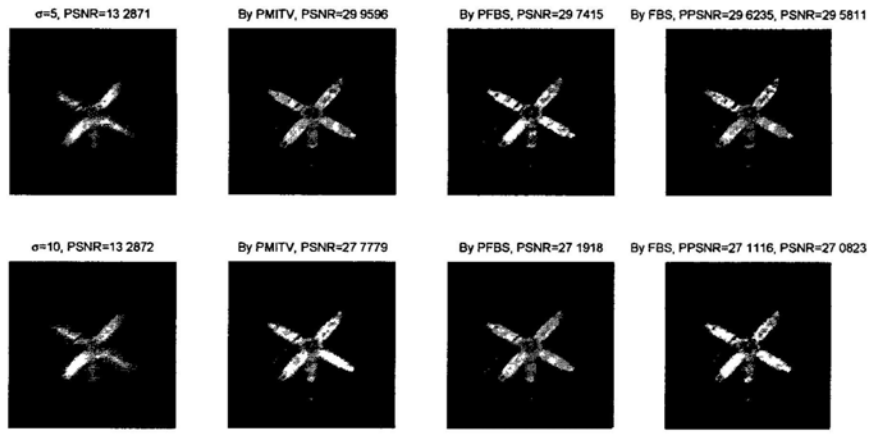

Figure 5.2: The figure shows the results of MITV algorithm and FBS algorithm when restoring blurred noisy image "satellite". From left to right, the first column shows the blurred noisy images (row 1: noise level $\sigma = 5$, PSNR = 13.2871; row 2: noise level $\sigma = 10$, PSNR = 13.2872). The right three columns show the restored results by the MITV algorithm and the FBS algorithm for restoring the corresponding degraded image in the first column on the same row. PMITV is the MITV (row 1: $\lambda = 0.4$; row 2: $\lambda = 1$) algorithm with a lower projection (5.25) (row 1: $\eta = 4$; row 2: $\eta = 9$) at every iteration; PFBS is the FBS algorithm with a lower projection (5.25) (row 1 and row 2: $\eta = 0$) at every iteration. In the last column, PPSNR denotes the PSNR value of the recovered image by the FBS algorithm only with a lower projection (5.25) ($\eta = 0$) at the final step; PSNR denotes the PSNR value without any projection. All the results shown in the figure are obtained with an additional upper projection (5.26) at the final step.

Figure 5.3: The figure shows the results of MITV algorithm and FBS algorithm when restoring blurred noisy image "joke". From left to right, the first column shows the blurred noisy images (row 1: noise level $\sigma = 5$, PSNR = 10.462; row 2: noise level $\sigma = 10$, PSNR = 10.462). The right three columns show the restored results by the MITV algorithm and the FBS algorithm for restoring the corresponding degraded image in the first column on the same row. PMITV is the MITV (row 1: $\lambda = 0.2$; row 2: $\lambda = 0.4$) algorithm with a lower projection (5.25) (row 1: $\eta = 4$; row 2: $\eta = 9$) at every iteration; PFBS is the FBS algorithm with a lower projection (5.25) (row 1 and row 2: $\eta = 0$) at every iteration. In the last column, PPSNR denotes the PSNR value of the recovered image by the FBS algorithm only with a lower projection (5.25) ($\eta = 0$) at the final step; PSNR denotes the PSNR value without any projection. All the results shown in the figure are obtained with an additional upper projection (5.26) at the final step.

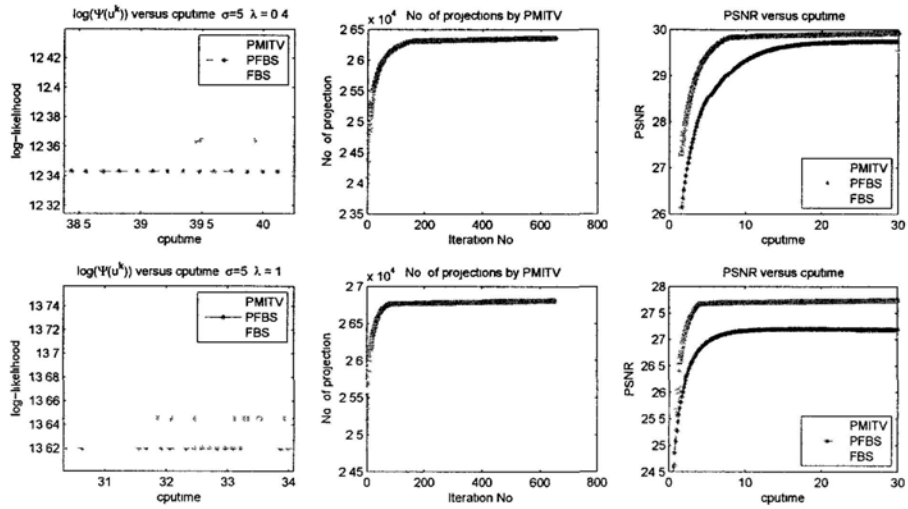

Figure 5.4: The figure shows the accuracy and the time comparison of the MITV algorithm and FBS algorithm for satellite image. PMITV algorithm is the MITV algorithm with a lower projection The first row is for noise level $\sigma = 5$, in PMITV, we take $\lambda = 0.4, \eta = 4$; the second row is for noise level $\sigma = 10$, in PMITV, we take $\lambda = 1. \eta = 9$. From left to right, the first column describes the decreasing of the energy value in time; the second column describes the number of projections at each iteration in PMITV; the third column shows the timing comparison of the PSNR values. From the second column, we see that the number of projections tends to a constant as time increasing. The third column shows us clearly that the MITV algorithm can reach a higher PSNR value faster than PFBS with projection and FBS without projection.

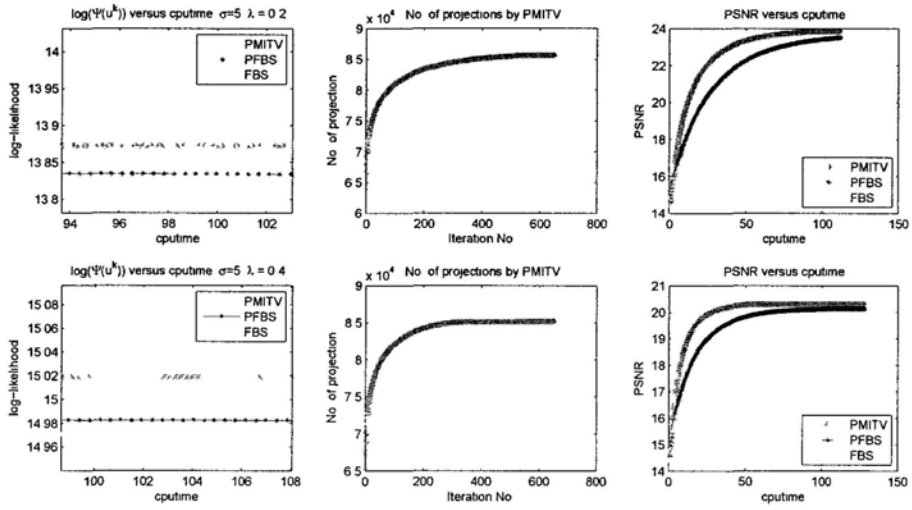

Figure 5.5: The figure shows the accuracy and the time comparison of the MITV algorithm and FBS algorithm for joke image. PMITV algorithm is the MITV algorithm with a lower projection. The first row is for noise level $\sigma = 5$, in PMITV, we take $\lambda = 0.2, \eta = 4$; the second row is for noise level $\sigma = 10$, in PMITV, we take $\lambda = 0.4, \eta = 9$. From left to right, the first column describes the decreasing of the energy value in time; the second column describes the number of projections at each iteration in PMITV; the third column shows the timing comparison of the PSNR values. From the second column, we see that the number of projections tends to a constant as time increasing. The third column shows us clearly that the MITV algorithm can reach a higher PSNR value faster than PFBS with projection and FBS without projection.

Bibliography

[1] J. Barzilai and J. Borwein, *Two point step size gradient methods,* IMA J. Numer. Anal., 8 (1988), pp. 141–148.

[2] P. Besbeas, I. De Fies, and T. Sapatinas, *A comparative simulation study of wavelet shrinkage estimators for Poisson counts*, Internat. Statist. Rev., 2004, pp. 209–237.

[3] A. Bovik, *Handbook of image and video processing*, New York, Academic, 2000.

[4] K. Bredies, *A forwardbackward splitting algorithm for the minimization of non-smooth convex functionals in Banach space,* Inverse Problems, 25(2009), pp. 1–20.

[5] J. Carter, *Dual Methods for Total Variation-Based Image Restoration,* Ph.D. thesis, UCLA, 2001.

[6] A. Chambolle and P. Lions, *Image recovery via total variation minimization and related problems*, Numer. Math., 76(1997), pp. 167–188.

[7] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vis., 20(2004), pp. 89–97.

[8] R. Chan and K. Chen, *Multilevel algorithms for a Poisson noise removal model with total variation regularization*, Int. J. Comput. Math., 84(2007), pp. 1183–1198.

[9] T. Chan, and S. Esedoglu, *Aspects of total variation regularized L^1 function approximation*, SIAM J. Appl. Math., 65(2005), pp. 1817–1837.

[10] T. Chan, G. Golub, and P. Mulet, *A non-linear primal-dual method for TV-based image restoration*, Lecture Notes in Control and Information Science no. 219 (Berger, M. et al eds.) (1996), pp. 241–252.

[11] T. Chan, A. Marquina, and P. Mulet, *High-order total variation-based image restoration*, SIAM J. Sci. Comput., 22(2000), pp. 503–516.

[12] T. Chan and P. Mulet, *On the convergence of the lagged diffusivity fixed point method in total variation image restoration*, SIAM J. Numer. Anal., 36(1999), pp. 354–367.

[13] T. Chan and J Shen, *Image processing and analysis*, Society for Industrial and Applied mathematics (SIAM), Philadelphia, PA, 2005. variational, PDE, wavelet, and stochastic methods.

[14] T. Chen, T. Huang, W. Yin, and X. Zhou, *A new coarse-to-fine framework for 3D brain MR image registration*, in Computer Vision of Biomedical Image, Lecture Notes in Computer Science, 3765(2005), pp. 114–124, Springer.

[15] T. Chen, W. Yin, X. Zhou, D. Comaniciu, and T. Huang, *Total variation models for variable lighting face recognition*, IEEE Trans. Pattern Anal. and Machine Intell., 28(2006), pp. 1519–1524.

[16] P. Combettes and V. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Model. and Simul., 4(2005), pp. 1168–1200.

[17] R. Courant, *Variational methods for the solution of problems with equilibrium and vibration*, Bull. Amer. Math. Soc., 49(1943), pp. 1–23.

[18] Y. Dong, M. Hintermüller, and M. Neri, *An efficient primal-dual method for L^1-TV image restoration*, SIAM J. Imaging Sci., 2(2009), pp. 1168–1189.

[19] I. Ekerland and R. Témam, *Convex Analysis and Variational Problems,* SIAM, 1999.

[20] E. Esser, *Applications of Lagrangian-based alternating direction methods and connections to split Bregman,* UCLA CAM Report, 09-31.

[21] H. Fu, M. Ng, M. Nikolova, and J. Barlow, *Efficient minimization methods of mixed l2-l1 and l1-l1 norms for image restoration*, SIAM J. Sci. Comput., 27(2006), pp. 1881–1902.

[22] D. Geman and C. Yang, *Nonlinear image recovery with half-quadratic regularization and FFTs*, IEEE Trans. Image Proc., 4(1995), pp. 932–946.

[23] T. Goldstein and S. Osher, *The split Bregman method for L1 regularization problems*, SIAM J. Imaging Sci., 2(2009), pp. 323–343

[24] X. Guo, F. Li, and M. Ng, *A fast l1-TV algorithm for image restoration and reconstruction*, SIAM J. Sci. Comput., 31(2009), pp. 2322–2341.

[25] W. Hager, B. Mair, and H. Zhang, *An affine-scalling interior-point CBB method for box-constrained optimization*, Mathematical Programming, 119(2009), pp. 1–32.

[26] M. Hintermüller, K. Ito, and K. Kunisch, *the primal-dual active set strategy as a semismooth newton method*, SIAM J. Optim., 13(2002), pp. 865–888.

[27] J. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and minimization Algorithms,* Springer-Verlag Berlin, I, 1996.

[28] P. Huber, *Robust regression: Asymptotics, conjectures, and Monte Carlo,* Ann. Statist., 1(1973), pp. 799–821.

[29] M. Jacobson and J. Fessler, *An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms*, IEEE Trans. Image Proc., 16(2007), pp. 2411–2422.

[30] M Jacobson and J. Fessler, *Properties of MM algorithms on convex feasible sets: extended version*, Tech. Rep. 353, Comm. and Sign. Proc. Lab., Dept. of EECS, Univ. of Michigan, Ann Arbor, MI, 48109-2122, 2004.

[31] D. Luenberger, *Linear and Nonlinear Programming (2nd edition)*, J. Wiley, 1984.

[32] M. Lysaker, A. Lundervold, and X. Tai, *Noise removal using fourth-order partial differential equation with applications to medical Magnetic Resonance Images in space and time*, IEEE Trans. Image Proc., 12(2003), pp. 1579–1590.

[33] M. Lysaker and X. Tai, *Iterative image restoration combining total variation minimization and a second order functional*, Int. J. Comput. Vis., pp. 5–18, 2006.

[34] J. Ma, *Positively constrained multiplicative iterative algorithm for maximum penalized likelihood tomographic reconstruction (to appear)*, IEEE Trans. Nuc. Scie., 2009.

[35] J. Moreau, *Function convexes duales et points proximaux dans un espace Hilbertien*, C. R. Acad. Sci. paris Sér. A math., 255(1962), pp. 1897–2899.

[36] J. Moreau, *Proximité et dualité dans un espace hilbertien*, Bull. Soc. math. France, 93(1995), pp. 273–299.

[37] B. Morini, M. Porcelli, and R. Chan, *A reduced Newton method for constrained linear least-squares problems*, J. Comp. Applied Math., 233(2010), pp. 2200–2212.

[38] M. Ng, P. Weiss, and X. Yuan, *Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods*, accepted by SIAM J. Sci. Comput.

[39] M. Nikolova, *Minimizers of cost-functions involving non-smooth data fidelity term*, SIAM J. Num. Ana., 40(2002), pp. 965–994.

[40] M. Nikolova, *A variational approach to remove outliers and impulse noise*, J. Math. imaging Vision, 20(2004), pp. 99–120.

[41] M. Nikolova and R. Chan, *The equivalence of half-quadratic minimization and the gradient linearization iteration*, IEEE Trans. Image Proc., 16(2007), pp. 1623–1627.

[42] M. Nikolova and M. Ng, *Analysis of half-quadratic minimization methods for signal and image recovery*, SIAM. J. Sci. Comput., 27(2006), pp. 937–966.

[43] R. Rockafellar, *Monotone operators and the proximal point algorithm.* SIAM J. Control Optim., 14(1976), pp. 877–898.

[44] P. Rodríigiez and B. Wohlberg, *Efficient minimization method for a generalized total variation functional*, IEEE Trans. Image Process., 18(2009), pp. 322–332.

[45] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D. 60(1992), pp. 259–268.

[46] D. Strong and T. Chan, *Edge-preserving and scale-dependent properties of total variation regularization*, Inverse Problems, 19(2003), pp. 165–187.

[47] A. Tikhonov and V. Arsenin, *Solutions of ill-posed problems*, SIAM Rev., 21(1979), pp. 266–267.

[48] Y. Vardi, L. Shepp and A. Kaufman, *A statistical model for positron emission tomography (with discussion)*, J. Amer. Stat. Assoc., 80(1985), pp. 8–37.

[49] C. Vogel and M. Oman, *Fast, robust total variation-based reconstruction of noisy, blurred images*, IEEE Trans. Image Proc., 7(1998), pp. 813–824.

[50] C. Vogel and M. Oman, *Iterative methods for total variation denoising,* SIAM J. Sci. Comput., 17(1996), pp. 227 – 238.

[51] C. Wu, J. Zhang, and X. Tai, *Augmented Lagrangian method for total variation restoration with non-quadratic fidelity,* UCLA CAM Report 09-82.

[52] J. Yang, Y. Zhang, and W. Yin, *An efficient TVL1 algorithm for image deblurring multichannel images corrupted by impulse noise,* SIAM J. Sci. Comput., 31(2009), pp. 2842–2865.

[53] W Yin, T. Chen, X. Zhou, and A. Chakraborty, *Background correction for cDNA microarray image using the TV+L1 model,* Bioinformatics, 21(2005), pp. 2410–2416.

[54] W Yin, D. Goldfarb, and S. Osher, *Image cartoon-texture decomposition and feature selection using the total variation regularized L1 functional,* in Variational, Geometric, and Level Set Methods in Computer Vision 2005, LNCS, 3752(2005), pp. 73–84.

[55] W. Yin, D. Goldfarb, and S. Osher, *The total variation regularized L^1 model for multiscale decomposition,* Multis. Model. Simul., 6(2006), pp. 190–211.

[56] M. Zhu and T. Chan, *An efficient primal-dual hybrid gradient algorithm for total variation image restoration,* UCLA CAM Report 08-34, 2008.

[57] M. Zhu, S. Wright, and T. Chan, *Duality-based algorithms for total-variation-regularized image restoration,* UCLA CAM Report 08-33, 2008.