Title
Image and Signal Processing with Non-Gaussian Noise: EM-Type Algorithms and Adaptive Outlier Pursuit

Author
Yan, Ming

Publication Date
2012

Peer reviewed|Thesis/dissertation

University of California
Los Angeles

Image and Signal Processing with Non-Gaussian Noise: EM-Type Algorithms and Adaptive Outlier Pursuit

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mathematics

by

Ming Yan

2012

Abstract of the Dissertation

Image and Signal Processing with Non-Gaussian Noise: EM-Type Algorithms and Adaptive Outlier Pursuit

by

Ming Yan

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 2012

Professor Luminita A. Vese, Chair

Most of the studies of noise-induced phenomena assume that the noise source is Gaussian because of the possibility of obtaining some analytical results when working with Gaussian noises. The use of non-Gaussian noises is rare, mainly because of the difficulties in handling them. However, there is experimental evidence indicating that in many phenomena, the noise sources could be non-Gaussian, for example Poisson data and sparsely corrupted data. This thesis provides two classes of algorithms for dealing with some special types of non-Gaussian noise.

Obtaining high quality images is very important in many areas of applied sciences, and the first part of this thesis is on expectation maximization (EM)-Type algorithms for image reconstruction with Poisson noise and weighted Gaussian noise. In these two chapters, we proposed general robust expectation maximization (EM)-Type algorithms for image reconstruction when the measured data is corrupted by Poisson noise and weighted Gaussian noise, without and with background emission. This method is separated into two steps: EM step and regularization step. In order to overcome the contrast reduction introduced by some regularizations, we suggested EM-Type algorithms with Bregman iteration by applying a sequence of modified EM-Type algorithms. One algorithm with total variation being the regularization is used for image reconstruction in computed tomography application.

The second part of this thesis is on adaptive outlier pursuit method for sparsely corrupted

data. In many real world applications, there are all kinds of errors in the measurements during data acquisition and transmission. Some errors will damage the data seriously and make the obtained data containing no information about the true signal, for example, sign flips in measurements for 1-bit compressive sensing and impulse noise in images. Adaptive outlier pursuit is used to detect the outlier and reconstruct the image or signal by iteratively reconstructing the image or signal and adaptively pursuing the outlier. Adaptive outlier pursuit method is used for robust 1-bit compressive sensing and impulse noise removal in chapters 4 and 5 respectively.

The dissertation of Ming Yan is approved.

Alex Bui

Jason Cong

Stanley Osher

Lieven Vandenberghe

Luminita A. Vese, Committee Chair

University of California, Los Angeles

2012

To my family

for their constant source of

love, concern, support and strength all these years.

Table of Contents

List of Figures

List of Tables

Acknowledgments

Foremost, I would like to express my deep appreciation to my advisor Professor Luminita A. Vese for her professional advice, dedicated guidance, constant support, collaboration and insight. I learned not only numerous ideas and ways to conduct research, but also many other things beyond academics.

Besides, I would like to thank Professor Alex Bui and Professor Jason Cong for their stimulating discussion in the group meetings at CDSC (Center of Domain Specific Computing) and support for the last three years. I must absolutely thank Professor Stanley Osher for organizing the level set collective every week. I also want to thank Professor Lieven Vandenberghe for offering the optimization courses.

I am thankful to Professor Andrea Bertozz and Professor Joseph Teran for their help in my studies and research. I also appreciate the helpful discussions and suggestions I received from postdoctoral and graduate students in CDSC and the math department, which includes Dr. Jianfeng Cai, Dr. Jianwen Chen, Dr. Bin Dong, Feng Guan, Dr. Rongjie Lai, Dr. Wenye Ma, Xiaokui Yang, Yi Yang, Yi Zou and many others.

My gratitude further goes to my friends who made my graduate study colorful and joyful. I am also thankful to all of the people who work in Department of Mathematics and Department of Computer Science here at UCLA for all of their help, especially, Maggie Albert, Martha Contreras, Babette Dalton and Alexandra Luong.

Finally, I would like to thank my parents and my younger sister in China, whom this thesis is dedicated to.

The research presented in this dissertation was supported by NSF Grant DMS-0714945 and Center for Domain-Specific Computing (CDSC) under the NSF Expeditions in Computing Award CCF-0926127.

Vita

2005	B.S. (Computational Mathematics), University of Science and Technology of China, P. R. China.
2008	M.S. (Computational Mathematics), University of Science and Technology of China, P. R. China.
2008–2012	Teaching and Research Assistant, Department of Mathematics, University of California, Los Angeles, California, USA.

Publications

M. Yan, Y. Yang and S. Osher, Robust 1-bit compressive sensing using adaptive outlier pursuit, *IEEE Transactions on Signal Processing*, to appear.

J. Chen, J. Cong, M. Yan and Y. Zou, FPGA-accelerated 3D reconstruction using compressive sensing, *In: Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA 2012)*, 163-166.

M. Yan, EM-type algorithms for image reconstruction with background emission and Poisson noise, *In: Proceedings of 7th International Symposium on Visual Computing*, Lecture Notes in Computer Science (LNCS), 6938 (2011), 33-42.

M. Yan, J. Chen, L. A. Vese, J. Villasenor, A. Bui and J. Cong, EM+TV based reconstruction for cone-beam CT with reduced radiation, *In: Proceedings of 7th International Symposium on Visual Computing*, Lecture Notes in Computer Science (LNCS), 6938 (2011), 1-10.

J. Chen, M. Yan, L. A. Vese, J. Villasenor, A. Bui and J. Cong, EM+TV for reconstruction of cone-beam CT with curved detectors using GPU, *In: Proceedings of International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine,* 2011, 363-366.

M. Yan and L. A. Vese, Expectation maximization and total variation based model for computed tomography reconstruction from undersampled data, *In: Proceedings of SPIE Medical Imaging: Physics of Medical Imaging*, 7961 (2011), 79612X. doi:10.1117/12.878238

H. Han and M. Yan, A mixed finite element method on a staggered mesh for Navier-Stokes equations, *Journal of Computational Mathematics*, 26 (2008), 816-824.

H. Han, M. Yan and C. Wu, An energy regularization method for the backward diffusion problem and its applications to image deblurring, *Communications in Computational Physics*, 4 (2008), 177-194.

CHAPTER 1

Introduction

Most of the studies of noise-induced phenomena assume that the noise source is Gaussian because of the possibility of obtaining some analytical results when working with Gaussian noises. The use of non-Gaussian noises is rare, mainly because of the difficulties in handling them. However, there is experimental evidence indicating that in many phenomena, the noise sources could be non-Gaussian, for example Poisson data and sparsely corrupted data. The methods for image and signal processing are different when different types of non-Gaussian noise are considered. In this thesis, we provide two classes of algorithms: expectation maximization (EM)-Type algorithms for image reconstruction with Poisson and weighted Gaussian noise, and adaptive outlier pursuit for image and signal reconstruction with sparsely corrupted data.

Obtaining high quality images is very important in many areas of applied sciences, such as medical imaging, optical microscopy and astronomy. The degradation model can be formulated as a linear inverse and ill-posed problem:

$$y = Ax + b + n. \tag{1.1}$$

Here x is the image to be reconstructed, A represents a model for the imaging device and is assumed to be linear, b is the background emission, y is the measured data and n is the non-Gaussian noise depending on $Ax + b$. The problem is to find x with A, b and y given. We proposed EM-Type algorithms for solving this problem when the noise n is Poisson noise and weighted Gaussian noise. These algorithms are performed by iteratively applying the EM step and regularization step.

Chapter 2 deals with the case without background emission ($b = 0$). In this chapter we proposed a class of EM-Type algorithms for image reconstruction with Poisson noise and weighted Gaussian noise. We proved the convergence of these algorithms by showing that the proposed EM-Type algorithms are equivalent to EM algorithms [1] with *a priori* information and alternating minimization methods for equivalent optimization problems. The performance of one algorithm with total variation (TV) [2] being the regularization, named EM-TV, is shown for image reconstruction in a computed tomography application.

Chapter 3 considers the case with background emission ($b \neq 0$), which occurs in astronomy and fluorescence microscopy. Similarly, we proposed a class of EM-Type algorithms for image reconstruction with Poisson noise and proved the convergence by showing the equivalence of EM-Type algorithms, EM algorithms with *a priori* information and alternating minimization methods for equivalent optimization problems. However, for some regularizations, the reconstructed images will lose contrast. To overcome this problem, EM-Type algorithms with Bregman iteration are introduced. The performance of EM-Type algorithms with or without Bregman iterations is shown for image deblurring.

The second part is on image and signal reconstruction with sparsely corrupted data. In many real world applications such as signal and image processing, there are all kinds of errors in the measurements during data acquisition and transmission. Some errors will damage the data seriously and make the obtained data containing no information about the true signal. Therefore, using this damaged data for signal reconstruction is useless and may worsen the performance of reconstruction methods, and methods robust to these outliers are strongly needed. For some applications like impulse noise removal, there are methods for detecting the damaged pixels: adaptive median filter (AMF) [3] is used in salt-and-pepper impulse noise removal and adaptive center wighted median filter (ACWMF) [4] is used in random-valued impulse noise removal. But the performance of ACWMF is not good enough when the number of pixels corrupted is very large. For other applications like 1-bit compressive sensing [5], there are no methods for detecting the sign flips in the measurements. Adaptive outlier pursuit is proposed to pursue the outliers adaptively by iteratively reconstructing the image or signal and detect the outliers.

Chapter 4 applies the adaptive outlier pursuit on robust 1-bit compressive sensing (CS) when some of the measurements are wrong. The classic compressive sensing (CS) theory assumes the measurements to be real-valued and have infinite bit precision. The quantization of CS measurements has been studied recently and it has been shown that accurate and stable signal acquisition is possible even when each measurement is quantized to only one single bit. The performance of all existing algorithms is worsened when there are a lot of sign flips in the measurements. We propose a robust method for recovering signals from 1-bit measurements using adaptive outlier pursuit. This method will detect the positions where sign flips happen and recover the signals using "correct" measurements. Numerical experiments show the accuracy of sign flips detection and high performance of signal recovery for our algorithms compared with other algorithms.

Chapter 5 is to solve the problem of image restoration of observed images corrupted by impulse noise (and mixed Gaussian impulse noise). Since the pixels damaged by impulse noise contain no information about the true image, how to find this set correctly is a very important problem. We proposed two methods based on blind inpainting and ℓ_0 minimization that can simultaneously find the damaged pixels and restore the image. By iteratively restoring the image and updating the set of damaged pixels, these methods have better performance than other methods, as shown in the experiments. In addition, we provide convergence analysis for these methods; these algorithms will converge to coordinatewise minimum points. In addition, they will converge to local minimum points (with probability one) with some modifications in the algorithms.

Part I

EM-Type Algorithms

CHAPTER 2

General Convergent Expectation Maximization (EM)-Type Algorithms for Image Reconstruction Without Background Emission

2.1 Introduction

Obtaining high quality images is very important in many areas of applied science, such as medical imaging, optical microscopy and astronomy. For some applications such as positron-emission-tomography (PET) and computed tomography (CT), analytical methods for image reconstruction are available. For instance, filtered back projection (FBP) is the most commonly used method for image reconstruction from CT by manufacturers of commercial imaging equipments [6]. However, it is sensitive to noise and suffers from streak artifacts (star artifacts). An alternative to this analytical reconstruction is the use of the iterative reconstruction technique, which is quite different from FBP. The main advantages of the iterative reconstruction technique over FBP are insensitivity to noise and flexibility [7]. The data can be collected over any set of lines, the projections do not have to be distributed uniformly in angle, and the projections can be even incomplete (limited angle). With the help of parallel computing and graphics processing units (GPUs), even iterative methods can be solved very fast. Therefore, iterative methods become more and more important, and we will focus on the iterative reconstruction technique only.

The degradation model can be formulated as a linear inverse and ill-posed problem:

$$y = Ax + b + n. \tag{2.1}$$

Here, y is the measured data (vector in $\mathbf{R}^M$ for the discrete case). A is a compact operator (matrix in $\mathbf{R}^{M\times N}$ for the discrete case). For all the applications we will consider, the entries of A are nonnegative and A does not have full column rank. x is the desired exact image (vector in $\mathbf{R}^N$ for the discrete case). b is the background emission and n is the noise (both are vectors in $\mathbf{R}^M$ for the discrete case). We will consider the case without background emission ($b = 0$) in this chapter. The case with background emission ($b \neq 0$) is considered in the next chapter. Since the matrix A does not have full column rank, the computation of x directly by finding the inverse of A is not reasonable because (2.1) is ill-posed and n is unknown. Even for the case without noise ($n = 0$), there are many solutions because A does not have full column rank. When there is noise in the measured data ($n \neq 0$), finding x is more difficult because of the unknown n. Therefore regularization techniques are needed for solving these problems efficiently.

One powerful technique for applying regularization is the Bayesian model, and a general Bayesian model for image reconstruction was proposed by Geman and Geman [8], and Grenander [9]. The idea is to use *a priori* information about the image x to be reconstructed. In the Bayesian approach, we assume that measured data y is a realization of a multi-valued random variable, denoted by Y and the image x is also considered as a realization of another multi-valued random variable, denoted by X. Therefore the Bayesian formula gives us

$$p_X(x|y) = \frac{p_Y(y|x)p_X(x)}{p_Y(y)}. \tag{2.2}$$

This is a conditional probability of having $X = x$ given that y is the measured data. After inserting the detected value of y, we obtain *a posteriori* probability distribution of X. Then we can find x^* such that $p_X(x|y)$ is maximized, as maximum a posteriori (MAP) likelihood estimation.

In general, X is assigned as a Gibbs random field, which is a random variable with the following probability distribution

$$p_X(x) \sim e^{-\beta J(x)}, \tag{2.3}$$

where $J(x)$ is a given convex energy function, and β is a positive parameter. There are many different choices for $J(x)$ depending on the applications. Some examples are, for instance, quadratic penalization $J(x) = \|x\|_2^2/2$ [10, 11], quadratic Laplacian $J(x) = \|\nabla x\|_2^2/2$ [12], total variation $J(x) = \||\nabla x|\|_1$ [13, 14, 15, 16, 17], and Good's roughness penalization $J(x) = \||\nabla x|^2/x\|_1$ [18], where $\|\cdot\|_1$ and $\|\cdot\|_2$ are the ℓ_1 and ℓ_2 norms respectively.

For the choices of probability densities $p_Y(y|x)$, we can choose

$$p_Y(y|x) \sim e^{-\|Ax-y\|_2^2/(2\sigma^2)} \tag{2.4}$$

in the case of additive Gaussian noise, and the minimization of the negative log-likelihood function gives us the famous Tikhonov regularization method [19]

$$\underset{x}{\text{minimize}} \ \frac{1}{2}\|Ax - y\|_2^2 + \beta J(x). \tag{2.5}$$

If the random variable Y of the detected values y follows a Poisson distribution [20, 21] with an expectation value provided by Ax instead of Gaussian distribution, we have

$$y_i \sim \text{Poisson}\{(Ax)_i\}, \quad i.e., \quad p_Y(y|x) \sim \prod_i \frac{(Ax)_i^{y_i}}{y_i!} e^{-(Ax)_i}. \tag{2.6}$$

By minimizing the negative log-likelihood function, we obtain the following optimization problem

$$\underset{x\geq 0}{\text{minimize}} \ \sum_i \big((Ax)_i - y_i \log(Ax)_i\big) + \beta J(x). \tag{2.7}$$

In this chapter, we will focus on solving (2.5) and (2.7). It is easy to see that the objective functions in (2.5) and (2.7) are convex. Additionally, with suitably chosen regularization $J(x)$, the objective functions are strictly convex, and the solutions to these problems are unique.

The work is organized as follows. The uniqueness of the solutions to problems (2.5) and (2.7) are provided in section 2.2 for the discrete modeling. In section 2.3, we will give a short introduction of expectation maximization (EM) iteration, or Richardson-Lucy algorithm,

used in image reconstruction without background emission from the view of optimization. In section 2.4, we will propose general EM-Type algorithms for image reconstruction without background emission when the measured data is corrupted by Poisson noise. This is based on the maximum *a posteriori* likelihood estimation and an EM step. In this section, these EM-Type algorithms are shown to be equivalent to EM algorithms with *a priori* information, and their convergence is shown in two different ways. In addition, these EM-Type algorithms are also considered as alternating minimization methods for equivalent optimization problems. When the noise is weighted Gaussian noise, we also have the similar EM-Type algorithms. Simultaneous algebraic reconstruction technique is shown to be EM algorithm in section 2.5, and EM-Type algorithms for weighted Gaussian noise are introduced in section 2.6. In section 2.6, we also show the convergence analysis of EM-Type algorithms for weighted Gaussian noise via EM algorithms with *a priori* information and alternating minimization methods. Some numerical experiments in CT reconstruction are given in section 2.7 to show the efficiency of the EM-Type algorithms. We will end this work by a short conclusion section.

2.2 Uniqueness of Solutions to Problems (2.5) and (2.7)

As mentioned in the introduction, the original problem without regularization is ill-posed. Therefore at least one of these three properties: (i) a solution of the problem exists, (ii) the solution is unique, and (iii) the solution depends continuously on the data, are not fulfilled. For the well-posedness of the continuous modeling of problems (2.5) and (2.7), the analysis will be different depending on different regularizations. If $J(x) = \||\nabla x|\|_1$, i.e,. the regularization is the total variation, the well-posedness of the regularization problems is shown in [22] and [15] for Gaussian and Poisson noise respectively. However, for discrete modeling, the well-posedness of the problems is easy to show, because problems (2.5) and (2.7) are convex. We have to just show that the solutions are unique.

In discrete modeling, the operator A is a matrix and x is a vector. After imposing some reasonable assumptions on $J(x)$ and A, the objective functions are strictly convex, therefore

the solutions are unique. The strict convexity means that given two different vectors x^1 and x^2, then for any $w \in (0,1)$, the new vector $x_w = wx^1 + (1-w)x^2$ satisfies

$$\begin{aligned}\frac{1}{2}\|Ax_w - y\|_2^2 + \beta J(x_w) <& w\frac{1}{2}\|Ax^1 - y\|_2^2 + w\beta J(x^1) \\ &+ (1-w)\frac{1}{2}\|Ax^2 - y\|_2^2 + (1-w)\beta J(x^2). \end{aligned} \tag{2.8}$$

If the objective function is not strictly convex, then we can find two different vectors x^1 and x^2 and $w \in (0,1)$ such that

$$\begin{aligned}\frac{1}{2}\|Ax_w - y\|_2^2 + \beta J(x_w) \geq& w\frac{1}{2}\|Ax^1 - y\|_2^2 + w\beta J(x^1) \\ &+ (1-w)\frac{1}{2}\|Ax^2 - y\|_2^2 + (1-w)\beta J(x^2). \end{aligned} \tag{2.9}$$

From the convexity of the objective function, we have

$$\begin{aligned}\frac{1}{2}\|Ax_w - y\|_2^2 + \beta J(x_w) =& w\frac{1}{2}\|Ax^1 - y\|_2^2 + w\beta J(x^1) \\ &+ (1-w)\frac{1}{2}\|Ax^2 - y\|_2^2 + (1-w)\beta J(x^2), \end{aligned} \tag{2.10}$$

for all $w \in (0,1)$. Since $\frac{1}{2}\|Ax - y\|_2^2$ and $J(x)$ are convex, we have

$$\frac{1}{2}\|Ax_w - y\|_2^2 = w\frac{1}{2}\|Ax^1 - y\|_2^2 + (1-w)\frac{1}{2}\|Ax^2 - y\|_2^2, \tag{2.11}$$

$$J(x_w) = wJ(x^1) + (1-w)J(x^2), \tag{2.12}$$

for all $w \in (0,1)$. From the equation (2.11), we have $Ax^1 = Ax^2$. If A is injective, i.e., the null space of A is trivial, x^1 and x^2 have to be equal, then the objective function is strictly convex. If A is not injective (A does not have full column rank), for instance, reconstruction from PET and CT with undersampled data, we have to also consider equation (2.12). The equality in (2.12) depends on the regularization $J(x)$. For quadratic penalization, $J(x)$ is strictly convex, which implies $x^1 = x^2$, while for quadratic Laplacian, the equation (2.12) gives us $\nabla x^1 = \nabla x^2$. If $J(x)$ is the total variation, we obtain, from the equality, that $\nabla x^1 = \alpha \nabla x^2$

with $\alpha \geq 0$ and depending on the pixel (or voxel). When Good's roughness penalization is used, we have $\frac{\nabla x^1}{x^1} = \frac{\nabla x^2}{x^2}$ from the equality. Thus, if the matrix A is chosen such that we can not find two different vectors (images) satisfying $Ax^1 = Ax^2$ and $\nabla x^1 = \alpha \nabla x^2$, the objective function is strictly convex. Actually, this assumption is reasonable and in the applications mentioned above, it is satisfied. Therefore, for the discrete modeling, the optimization problem has a unique solution. If Poisson noise, instead of Gaussian noise, is assumed, the objective function is still strictly convex, and the problem has a unique solution.

2.3 Expectation Maximization (EM) Iteration

A maximum likelihood (ML) method for image reconstruction based on Poisson data was introduced by Shepp and Vardi [21] in 1982 for image reconstruction in emission tomography. In fact, this algorithm was originally proposed by Richardson [23] in 1972 and Lucy [24] in 1974 for image deblurring in astronomy. The ML method is a method for solving the special case of problem (2.7) without regularization term, i.e., $J(x)$ is a constant, which means we do not have any *a priori* information about the image. From equation (2.6), for given measured data y, we have a function of x, the likelihood of x, defined by $p_Y(y|x)$. Then a ML estimation of the unknown image is defined as any maximizer x^* of $p_Y(y|x)$.

By taking the negative log-likelihood, one obtains, up to an additive constant,

$$f_0(x) = \sum_i \big((Ax)_i - y_i \log(Ax)_i\big), \tag{2.13}$$

and the problem is to minimize this function $f_0(x)$ on the nonnegative orthant, because we have the constraint that the image x is nonnegative. In fact, we have

$$f(x) = D_{KL}(y, Ax) := \sum_i \big(y_i \log \frac{y_i}{(Ax)_i} + (Ax)_i - y_i\big) = f_0(x) + C, \tag{2.14}$$

where $D_{KL}(y, Ax)$ is the Kullback-Leibler (KL) divergence of Ax from y, and C is a constant

independent of x. The KL divergence is considered as a data-fidelity function for Poisson data just like the standard least-square $\|Ax-y\|_2^2$ is the data-fidelity function for additive Gaussian noise. It is convex, nonnegative and coercive on the nonnegative orthant, so the minimizers exist and are global.

In order to find a minimizer of $f(x)$ with the constraint $x_j \geq 0$ for all j, we can solve the Karush-Kuhn-Tucker (KKT) conditions [25, 26],

$$\begin{aligned} \sum_i \left(A_{i,j}(1-\frac{y_i}{(Ax)_i}) \right) - s_j = 0, \qquad & j=1,\cdots,N, \\ s_j \geq 0, \quad x_j \geq 0, \qquad & j=1,\cdots,N, \\ s^T x = 0, & \end{aligned}$$

where s_j is the Lagrangian multiplier corresponding to the constraint $x_j \geq 0$. By the positivity of $\{x_j\}$, $\{s_j\}$ and the complementary slackness condition $s^T x = 0$, we have $s_j x_j = 0$ for every $j \in \{1,\cdots,N\}$. Multiplying by x_j gives us

$$\sum_i \left(A_{i,j}(1-\frac{y_i}{(Ax)_i}) \right) x_j = 0, \qquad j=1,\cdots,N.$$

Therefore, we have the following iteration scheme

$$x_j^{k+1} = \frac{\sum_i \left(A_{i,j}(\frac{y_i}{(Ax^k)_i}) \right)}{\sum_i A_{i,j}} x_j^k. \tag{2.15}$$

This is the well-known EM iteration or Richardson-Lucy algorithm in image reconstruction, and an important property of it is that it preserves positivity. If x^k is positive, then x^{k+1} is also positive if A preserves positivity. It is also shown that for each iteration, $\sum_i (Ax)_i$ is fixed and equals $\sum_i y_i$. Since $\sum_i (Ax)_i = \sum_j (\sum_i A_{i,j}) x_j$, the minimizer has a weighted l_1 constraint.

Shepp and Vardi showed in [21] that this is equivalent to the EM algorithm proposed by Dempster, Laird and Rubin [1]. To make it clear, EM iteration means the special EM

method used in image reconstruction, while EM algorithm means the general EM algorithm for solving missing data problems.

2.4 EM-Type Algorithms for Poisson data

The method shown in the last section is also called maximum-likelihood expectation maximization (ML-EM) reconstruction, because it is a maximum likelihood approach without any Bayesian assumption on the images. If additional *a priori* information about the image is given, we have maximum a posteriori probability (MAP) approach [27, 28], which is the case with regularization term $J(x)$. Again we assume here that the detected data is corrupted by Poisson noise, and the regularization problem is

$$\begin{cases} \underset{x}{\text{minimize}} & E^P(x) := \beta J(x) + \sum_i \left((Ax)_i - y_i \log(Ax)_i\right), \\ \text{subject to} & x_j \geq 0, \quad j = 1, \cdots, N. \end{cases} \tag{2.16}$$

This is still a convex constraint optimization problem if J is convex and we can find the optimal solution by solving the KKT conditions:

$$\begin{aligned} \beta \partial J(x)_j + \sum_i \left(A_{i,j}(1 - \frac{y_i}{(Ax)_i}) \right) - s_j = 0, \qquad & j = 1, \cdots, N, \\ s_j \geq 0, \quad x_j \geq 0, \qquad & j = 1, \cdots, N, \\ s^T x = 0. & \end{aligned}$$

Here s_j is the Lagrangian multiplier corresponding to the constraint $x_j \geq 0$. By the positivity of $\{x_j\}$, $\{s_j\}$ and the complementary slackness condition $s^T x = 0$, we have $s_j x_j = 0$ for every $j \in \{1, \cdots, N\}$. Thus we obtain

$$\beta x_j \partial J(x)_j + \sum_i \left(A_{i,j}(1 - \frac{y_i}{(Ax)_i}) \right) x_j = 0, \qquad j = 1, \cdots, N,$$

or equivalently

$$\beta \frac{x_j}{\sum\limits_i A_{i,j}} \partial J(x)_j + x_j - \frac{\sum\limits_i \left(A_{i,j} (\frac{y_i}{(Ax)_i}) \right)}{\sum\limits_i A_{i,j}} x_j = 0, \qquad j = 1, \cdots, N.$$

Notice that the last term on the left hand side is an EM step (2.15). After plugging the EM step into the equation, we obtain

$$\beta \frac{x_j}{\sum\limits_i A_{i,j}} \partial J(x)_j + x_j - x_j^{EM} = 0, \qquad j = 1, \cdots, N, \tag{2.17}$$

which is the optimality condition for the following optimization problem

$$\underset{x}{\text{minimize}} \; E_1^P(x, x^{EM}) := \beta J(x) + \sum_j (\sum_i A_{i,j}) \left(x_j - x_j^{EM} \log x_j \right). \tag{2.18}$$

Therefore we propose the general EM-Type algorithms in Algorithm 1. The initial guess x^0 can be any positive initial image, and ϵ, chosen for the stopping criteria, is a small constant. Num_Iter is the maximum number of iterations. If $J(x)$ is constant, the second step is just $x^k = x^{k-\frac{1}{2}}$ and this is exactly the ML-EM from the previous section. When $J(x)$ is not constant, we have to solve an optimization problem for each iteration. In general, the problem can not be solved analytically, and we have to use iterative methods to solve it. However, in practice, we do not have to solve it exactly by stopping it after a few iterations. We will show that the algorithms will also converge without solving it exactly.

Algorithm 1 Proposed EM-Type algorithms.

Input: x^0, ϵ
Initialization: $k = 1$
while $k < Num\_Iter$ & $\|x^k - x^{k-1}\| < \epsilon$ **do**
  $x^{k-\frac{1}{2}} = EM(x^{k-1})$ using (2.15),
  $x^k = \underset{x}{\text{argmin}} \; E_1^P(x, x^{k-\frac{1}{2}})$ by solving (2.18),
  k=k+1.
end while

2.4.1 Equivalence to EM Algorithms with *a priori* Information

In this subsection, the EM-Type algorithms are shown to be equivalent to EM algorithms with *a priori* information. The EM algorithm is a general approach for maximizing *a posterior* distribution when some of the data is missing [1]. It is an iterative method which alternates between expectation (E) steps and maximization (M) steps. For image reconstruction, we assume that the missing data is the latent variables $\{z_{ij}\}$, describing the intensity of pixel (or voxel) j observed by detector i. Therefore the observed data are $y_i = \sum_j z_{ij}$. We can have the assumption that z is a realization of multi-valued random variable Z, and for each (i, j) pair, z_{ij} follows a Poisson distribution with expected value $A_{i,j}x_j$, because the summation of two Poisson distributed random variables also follows a Poisson distribution, whose expected value is summation of the two expected values.

The original E-step is to find the expectation of the log-likelihood given the present variables x^k:

$$Q(x|x^k) = E_{z|x^k,y} \log p(x, z|y).$$

Then, the M-step is to choose x^{k+1} to maximize the expected log-likelihood $Q(x|x^k)$ found in the E-step:

$$\begin{aligned} x^{k+1} &= \operatorname*{argmax}_x \; E_{z|x^k,y} \log p(x, z|y) = \operatorname*{argmax}_x \; E_{z|x^k,y} \log(p(y, z|x)p(x)) \\ &= \operatorname*{argmax}_x \; E_{z|x^k,y} \sum_{ij} \left(z_{ij} \log(A_{i,j}x_j) - A_{i,j}x_j\right) - \beta J(x) \\ &= \operatorname*{argmin}_x \; \sum_{ij} (A_{i,j}x_j - E_{z|x^k,y} z_{ij} \log(A_{i,j}x_j)) + \beta J(x). \end{aligned} \tag{2.19}$$

From (2.19), what we need before solving it is just $\{E_{z|x^k,y} z_{ij}\}$. Therefore we can compute the expectation of missing data $\{z_{ij}\}$ given present x^k and the condition $y_i = \sum_j z_{ij}$, denoting this as an E-step. Because for fixed i, $\{z_{ij}\}$ are Poisson variables with mean $\{A_{i,j}x_j^k\}$ and $\sum_j z_{ij} = y_i$, the conditional distribution of z_{ij} is binomial distribution $\left(y_i, \frac{A_{i,j}x_j^k}{(Ax^k)_i}\right)$. Thus we

can find the expectation of z_{ij} with all these conditions by the following E-step

$$z_{ij}^{k+1} = E_{z|x^k,y} z_{ij} = \frac{A_{i,j} x_j^k y_i}{(Ax^k)_i}. \tag{2.20}$$

After obtaining the expectation for all z_{ij}, we can solve the M-step (2.19).

We will show that EM-Type algorithms are exactly the described EM algorithms with *a priori* information. Recalling the definition of x^{EM}, we have

$$x_j^{EM} = \frac{\sum_i z_{ij}^{k+1}}{\sum_i A_{i,j}}.$$

Therefore, the M-step is equivalent to

$$\begin{aligned} x^{k+1} &= \operatorname*{argmin}_x \sum_{ij} (A_{i,j} x_j - z_{ij}^{k+1} \log(A_{i,j} x_j)) + \beta J(x) \\ &= \operatorname*{argmin}_x \sum_j (\sum_i A_{i,j})(x_j - x_j^{EM} \log(x_j)) + \beta J(x). \end{aligned}$$

We have shown that EM-Type algorithms are EM algorithms with *a priori* information. The convergence of EM-Type algorithms is shown in the next subsection from the convergence of the EM algorithms with *a priori* information.

2.4.2 Convergence of EM-Type Algorithms

In this subsection, we will show that the negative log-likelihood is decreasing in the following theorem.

Theorem 2.4.1. *The objective function (negative log-likelihood) $E^P(x^k)$ in (2.16) with x^k given by Algorithm 1 will decrease until it attaints a minimum.*

Proof. For all k and i, we always have the constraint satisfied

$$\sum_j z_{ij}^k = y_i.$$

Therefore, we have the following inequality

$$\begin{aligned} y_i \log\left((Ax^{k+1})_i\right) - y_i \log\left((Ax^k)_i\right) &= y_i \log\left(\frac{(Ax^{k+1})_i}{(Ax^k)_i}\right) = y_i \log\left(\sum_j \frac{A_{i,j}x_j^{k+1}}{(Ax^k)_i}\right) \\ &= y_i \log\left(\sum_j \frac{A_{i,j}x_j^k x_j^{k+1}}{(Ax^k)_i x_j^k}\right) = y_i \log\left(\sum_j \frac{z_{ij}^{k+1} A_{i,j} x_j^{k+1}}{y_i A_{i,j} x_j^k}\right) \\ &\geq y_i \sum_j \frac{z_{ij}^{k+1}}{y_i} \log\left(\frac{A_{i,j}x_j^{k+1}}{A_{i,j}x_j^k}\right) \qquad \text{(Jensen's inequality)} \\ &= \sum_j z_{ij}^{k+1} \log(A_{i,j}x_j^{k+1}) - \sum_j z_{ij}^{k+1} \log(A_{i,j}x_j^k). \qquad (2.21) \end{aligned}$$

This inequality gives us

$$\begin{aligned} E^P(x^{k+1}) - E^P(x^k) =& \sum_i ((Ax^{k+1})_i - y_i \log(Ax^{k+1})_i) + \beta J(x^{k+1}) \\ & - \sum_i \left((Ax^k)_i - y_i \log(Ax^k)_i\right) - \beta J(x^k) \\ \leq& \sum_{ij} (A_{i,j}x_j^{k+1} - z_{ij}^{k+1} \log(A_{i,j}x_j^{k+1})) + \beta J(x^{k+1}) \\ & - \sum_{ij} (A_{i,j}x_j^k - z_{ij}^{k+1} \log(A_{i,j}x_j^k)) - \beta J(x^k) \\ \leq& 0. \end{aligned}$$

The first inequality comes from (2.21) and the second inequality comes from the M-step (2.19). When $E^P(x^{k+1}) = E^P(x^k)$, these two equalities have to be satisfied. The first equality is satisfied if and only if $x_j^{k+1} = \alpha x_j^k$ for all j with α being a constant, while the second one is satisfied if and only if x^k and x^{k+1} are minimizers of the M-step (2.19). The objective function to be minimized in M-step (2.19) is strictly convex, which means that α has to be 1 and

$$\beta x_j^k \partial J(x^k)_j + \sum_i A_{i,j}x_j^k - \sum_i z_{ij}^{k+1} = 0, \qquad j = 1, \cdots, N.$$

After plugging the E-step (2.20) into these equations, we have

$$\beta x_j^k \partial J(x^k)_j + \sum_i A_{i,j} x_j^k - \sum_i \frac{A_{i,j} x_j^k y_i}{(Ax^k)_i} = 0, \qquad j = 1, \cdots, N.$$

Therefore, x^k is one minimizer of the original problem. □

The log-likelihood function will increase for each iteration until the solution is found, and in the proof, we do not fully use the M-step. Even if the M-step is not solved exactly, it will still increase as long as $Q(x^{k+1}|x^k) > Q(x^k|x^k)$ is satisfied before x^k converges.

The increasing of log-likelihood function can be proved in another way by using the M-step. From $x^{k+1} = \underset{x}{\operatorname{argmax}}\ Q(x|x^k)$, we have

$$\beta x_j^{k+1} \partial J(x^{k+1})_j + \sum_i A_{i,j} x_j^{k+1} - \sum_i z_{ij}^{k+1} = 0, \qquad j = 1, \cdots, N.$$

Multiplying by $(x_j^{k+1} - x_j^k)/x_j^{k+1}$ and taking summation over j gives us

$$\beta \sum_j (x_j^{k+1} - x_j^k) \partial J(x^{k+1})_j + \sum_{ij} A_{i,j}(x_j^{k+1} - x_j^k) - \sum_{ij} z_{ij}^{k+1} \frac{x_j^{k+1} - x_j^k}{x_j^{k+1}} = 0.$$

From the convexity of $J(x)$, we have

$$J(x^k) \geq J(x^{k+1}) + (x^k - x^{k+1}) \partial J(x^{k+1}) = J(x^{k+1}) + \sum_j (x_j^k - x_j^{k+1}) \partial J(x^{k+1})_j.$$

Therefore we have

$$\begin{aligned} 0 &\geq \beta J(x^{k+1}) - \beta J(x^k) + \sum_{ij} A_{i,j}(x_j^{k+1} - x_j^k) - \sum_{ij} z_{ij}^{k+1} \frac{x_j^{k+1} - x_j^k}{x_j^{k+1}} \\ &= E^P(x^{k+1}) - E^P(x^k) + \sum_i y_i \log\left(\frac{(Ax^{k+1})_i}{(Ax^k)_i}\right) - \sum_{ij} z_{ij}^{k+1} \frac{x_j^{k+1} - x_j^k}{x_j^{k+1}} \end{aligned}$$

$$
\begin{aligned}
&\geq E^P(x^{k+1}) - E^P(x^k) + \sum_i y_i \left(1 - \frac{(Ax^k)_i}{(Ax^{k+1})_i}\right) - \sum_{ij} z_{ij}^{k+1} \frac{x_j^{k+1} - x_j^k}{x_j^{k+1}} \\
&= E^P(x^{k+1}) - E^P(x^k) - \sum_i y_i \frac{\sum_j A_{i,j} x_j^k}{\sum_j A_{i,j} x_j^{k+1}} + \sum_{ij} z_{ij}^{k+1} \frac{x_j^k}{x_j^{k+1}} \\
&\geq E^P(x^{k+1}) - E^P(x^k).
\end{aligned}
$$

The second inequality comes from $\log(x) \geq 1 - 1/x$ for $x > 0$, and the last inequality comes from Cauchy-Schwarz inequality. If $E^P(x^{k+1}) = E^P(x^k)$, from the last inequality, we have $x_j^{k+1} = \alpha x_j^k$ for all j with a constant α, and from the second inequality, we have $(Ax^k)_i = (Ax^{k+1})_i$ which makes $\alpha = 1$. Therefore, the log-likelihood function will increase until the solution is found.

2.4.3 EM-Type Algorithms are Alternating Minimization Methods

In this section, we will show that these algorithms can also be derived from alternating minimization methods of other problems with variables x and z. The new optimization problems are

$$
\begin{aligned}
&\underset{x,z}{\text{minimize}}\ E^P(x,z) := \sum_{ij} \left(z_{ij} \log \frac{z_{ij}}{A_{i,j} x_j} + A_{i,j} x_j - z_{ij} \right) + \beta J(x). \\
&\text{subject to } \sum_j z_{ij} = y_i, \qquad i = 1, \cdots, M.
\end{aligned} \tag{2.22}
$$

Here E^P is used again to define the new function. $E^P(\cdot)$ means the negative log-likelihood function of x, while $E^P(\cdot,\cdot)$ means the new function of x and z defined in new optimization problems.

Having initial guess $x^0,\ z^0$ of x and z, the iteration for $k = 0, 1, \cdots$ is as follows:

$$
\begin{aligned}
z^{k+1} &= \underset{z}{\text{argmin}}\ E^P(x^k, z), \qquad \text{subject to } \sum_j z_{ij} = y_i, \\
x^{k+1} &= \underset{x}{\text{argmin}}\ E^P(x, z^{k+1}).
\end{aligned}
$$

Firstly, in order to obtain z^{k+1}, we fix $x = x^k$ and easily derive

$$z_{ij}^{k+1} = \frac{A_{i,j}x_j^k y_i}{(Ax^k)_i}. \tag{2.23}$$

After finding z^{k+1}, let $z = z^{k+1}$ fixed and update x, then we have

$$\begin{aligned} x^{k+1} &= \underset{x}{\operatorname{argmin}} \sum_{ij} \left(A_{i,j}x_j + z_{ij}^{k+1} \log \frac{z_{ij}^{k+1}}{A_{i,j}x_j} \right) + \beta J(x) \\ &= \underset{x}{\operatorname{argmin}} \sum_{ij} \left(A_{i,j}x_j - z_{ij}^{k+1} \log(A_{i,j}x_j) \right) + \beta J(x), \end{aligned}$$

which is the M-Step (2.19) in section **2.4.1**. The equivalence of problems (2.16) and (2.22) is provided in the following theorem.

Theorem 2.4.2. *If (x^*, z^*) is a solution of problem (2.22), then x^* is also a solution of (2.16), i.e., $x^* = \underset{x}{\operatorname{argmin}}\ E^P(x)$. If x^* is a solution of (2.16), then we can find z^* from (2.23) and (x^*, z^*) is a solution of problem (2.22).*

Proof. The equivalence can be proved in two steps. Firstly, we will show that $E^P(x, z) \geq E^P(x) + C$ for all z, here C is a constant dependent on y only.

$$\begin{aligned} E^P(x, z) &= \sum_{ij} \left(z_{ij} \log \frac{z_{ij}}{A_{i,j}x_j} + A_{i,j}x_j - z_{ij} \right) + \beta J(x) \\ &= \sum_{ij} \left(\frac{z_{ij}}{y_i} \log \frac{z_{ij}}{A_{i,j}x_j} \right) y_i + \sum_i ((Ax)_i - y_i) + \beta J(x) \\ &\geq \sum_i y_i \log \left(\frac{y_i}{(Ax)_i} \right) + \sum_i ((Ax)_i - y_i) + \beta J(x) \\ &= E^P(x) + \sum_i (y_i \log y_i - y_i). \end{aligned}$$

The inequality comes form Jensen's inequality, and the equality is satisfied if and only if

$$\frac{z_{ij}}{A_{i,j}x_j} = C_i, \qquad \forall j = 1, \cdot, N,$$

where C_i are constants, which depends on x, y and i, can be found from the constraint

$\sum_j z_{ij} = y_i$. Therefore $\min_z E^P(x,z) = E^P(x) + C$, which means that problems (2.22) and (2.16) are equivalent. □

2.5 Simultaneous Algebraic Reconstruction Technique (SART) is EM

Among all the iterative reconstruction algorithms, there are two important classess. One is EM from statistical assumptions mentioned above, and the other is algebraic reconstruction technique (ART)-type algorithms [29, 30]. Simultaneous algebraic reconstruction technique (SART) [31, 32], as a refinement of ART, is used widely [33, 34, 35] and the convergence analysis of SART is well studied by Jiang and Wang [36, 37], Wang and Zheng [38], Censor and Elfving [39] and Yan [40]. In this section, we will show that SART is also an EM algorithm, building the connection between these two classes.

From the convergence analysis of SART in [40], SART is also an algorithm for solving a maximum likelihood problem

$$p_Y(y|x) = \prod_i \frac{1}{\sqrt{2\pi w_i}} e^{-\frac{(y_i-(Ax)_i)^2}{2w_i}}, \tag{2.24}$$

where $w_i = \sum_j A_{i,j}$. Similarly, we assume that the missing data $\{z_{ij}\}$ follow normal distributions with expected values $\{A_{i,j}x_j\}$ and variances $\{A_{i,j}\}$ respectively. The original E-step is to find the expectation of the log-likelihood given the present variables x^k and the constraints $y_i = \sum_j z_{ij}$. It is easy to derive that under the constraints, $\{z_{ij}\}$ are still realizations of normally distributed random variables, but with different expected values $\{A_{i,j}x_j + \frac{A_{i,j}(y_i-(Ax)_i)}{w_i}\}$ and variances $\{\frac{A_{i,j}(w_i-A_{i,j})}{w_i}\}$ respectively.

In this section, we consider the special case without regularization function, i.e., there is no *a priori* information about the image to be reconstructed. The M-step is to maximize

the expected value of the log-likelihood function,

$$E_{z|x^k,y} \log p(y,z|x) = -E_{z|x^k,y} \sum_{ij} \frac{(z_{ij} - A_{i,j}x_j)^2}{2A_{i,j}} + C \tag{2.25}$$

$$= -\sum_{ij} \frac{(E_{z|x^k,y} z_{ij} - A_{i,j}x_j)^2}{2A_{i,j}} + C, \tag{2.26}$$

where C is a constant independent of x and z. Therefore, for the E-step we have to just find the expected value of z_{ij} given x^k and the constraints, which is

$$z_{ij}^{k+1} = A_{i,j}x_j^k + \frac{A_{i,j}(y_i - (Ax^k)_i)}{w_i}. \tag{2.27}$$

For the M-step, we find x^{k+1} by maximizing $p(y, z^{k+1}|x)$ with respect to x, which has an analytical solution

$$x_j^{k+1} = \frac{\sum_i z_{ij}^{k+1}}{\sum_i A_{i,j}} = x_j^k + \frac{1}{\sum_i A_{i,j}} \sum_i \frac{A_{i,j}(y_i - (Ax^k)_i)}{w_i}. \tag{2.28}$$

This is the original SART algorithm proposed by Andersen [31].

From the convergence analysis of SART in [40], the result of SART depends on the initialization x^0 for both noiseless and noisy cases when A is underdetermined.

Remark: SART is just one example of Landweber-like schemes for solving systems of linear equations. By changing the variance of y_i and z_{ij}, different schemes can be proposed. For other Landweber-like schemes such as component averaging in [39, 41], they can also be derived from the EM algorithm similarly by choosing different variances. Furthermore, new schemes can be derived by choosing different variances.

2.6 EM-Type Algorithms for Gaussian Noise

It is shown in the last section that SART is an EM algorithm based on weighted Gaussian assumption for the problem without regularization. Without regularization, the original

problem is ill-posed, and the result will depend on the initialization x^0. In this section, we will consider the regularized problem

$$\underset{x}{\text{minimize}}\ E^G(x) := \beta J(x) + \sum_i \frac{((Ax)_i - y_i)^2}{2w_i}, \tag{2.29}$$

and derive EM-Type algorithms with Gaussian noise assumption for solving it. The E-step is the same as in the case without regularization,

$$z_{ij}^{k+1} = A_{i,j}x_j^k + \frac{A_{i,j}(y_i - (Ax^k)_i)}{w_i}. \tag{2.30}$$

However, the M-step is different because we have *a priori* information on the image x to be reconstructed. The new M-step is to solve the following optimization problems

$$\underset{x}{\text{minimize}}\ \sum_{ij} \frac{(z_{ij}^{k+1} - A_{i,j}x_j)^2}{2A_{i,j}} + \beta J(x), \tag{2.31}$$

which is equivalent to

$$\underset{x}{\text{minimize}}\ \frac{1}{2}\sum_j (\sum_i A_{i,j})(x_j - \frac{\sum_i z_{ij}^{k+1}}{\sum_i A_{i,j}})^2 + \beta J(x). \tag{2.32}$$

From the SART iteration (2.28) in the last section, we can define

$$x^{SART} = x_j^k + \frac{1}{\sum_i A_{i,j}} \sum_i \frac{A_{i,j}(y_i - (Ax^k)_i)}{w_i}. \tag{2.33}$$

and have

$$x^{k+1} = \underset{x}{\text{argmin}}\ E_1^G(x, x^{SART}) := \frac{1}{2}\sum_j (\sum_i A_{i,j})(x_j - x_j^{SART})^2 + \beta J(x). \tag{2.34}$$

Therefore, the proposed EM-Type algorithms for image reconstruction with Gaussian noise are as follows.

Algorithm 2 Proposed EM-Type algorithms for Gaussian noise.

Input: x^0, ϵ,
Initialization: $k = 1$.
while $k < Num\_Iter$ & $\|x^k - x^{k-1}\| < \epsilon$ **do**
$x^{k-\frac{1}{2}} = SART(x^{k-1})$ using (2.33)
$x^k = \text{argmin } E_1^G(x, x^{k-\frac{1}{2}})$ by solving (2.34)
k=k+1.
end while

The initial guess x^0 can be any initial image and ϵ, chosen for the stopping criteria, is very small. Num_Iter is the maximum number of iterations. When $J(x)$ is not constant, we have to solve an optimization problem for each iteration. In general, the problem can not be solved analytically, and we have to use iterative methods to solve it. Similarly, we will show that the algorithms also converge without solving the optimization problem exactly, so we can stop it after a few iterations. The convergence analysis of these algorithms is shown in two different ways as for the case with Poisson noise.

2.6.1 Convergence Analysis of EM-Type Algorithms for Gaussian Noise

Same as EM-Type algorithms for Poisson noise, we have convergence analysis of EM-Type algorithms for Gaussian Noise. Firstly, we will show that the objective function (negative log-likelihood function) is decreasing.

Theorem 2.6.1. *The objective function (negative log-likelihood) $E^G(x^k)$ in (2.29) with x^k given by Algorithm 2 will decrease until it attains a minimum.*

Proof. Since for all k and i, we always have

$$\sum_j z_{ij}^k = y_i,$$

we have the following inequalities,

$$\begin{aligned}
&\frac{((Ax^{k+1})_i - y_i)^2}{w_i} - \frac{((Ax^k)_i - y_i)^2}{w_i} = \frac{\left(\sum_j A_{i,j}(x_j^{k+1} + x_j^k) - 2y_i\right)\left(\sum_j A_{i,j}(x_j^{k+1} - x_j^k)\right)}{w_i} \\
&= \left(\sum_j \frac{A_{i,j}}{w_i}(x_j^{k+1} - x_j^k) + \frac{2}{w_i}\left(\sum_j A_{i,j}x_j^k - y_i\right)\right)\left(\sum_j \frac{A_{i,j}}{w_i}(x_j^{k+1} - x_j^k)\right) w_i \\
&\leq \sum_j A_{i,j}\left((x_j^{k+1} - x_j^k) + \frac{2}{w_i}\left(\sum_j A_{i,j}x_j^k - y_i\right)\right)(x_j^{k+1} - x_j^k) \qquad \text{(Jensen's inequality)} \\
&= \sum_j A_{i,j}\left((x_j^{k+1} - x_j^k) + \frac{2}{A_{i,j}}\left(A_{i,j}x_j^k - z_{ij}^{k+1}\right)\right)(x_j^{k+1} - x_j^k) \\
&= \sum_j \left(\frac{(z_{ij}^{k+1} - A_{i,j}x_j^{k+1})^2}{A_{i,j}} - \frac{(z_{ij}^{k+1} - A_{i,j}x_j^k)^2}{A_{i,j}}\right). \qquad (2.35)
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
E^G(x^{k+1}) - E^G(x^k) &= \sum_i \frac{((Ax^{k+1})_i - y_i)^2}{2w_i} + \beta J(x^{k+1}) - \sum_i \frac{((Ax^k)_i - y_i)^2}{2w_i} - \beta J(x^k) \\
&\leq \sum_{ij} \frac{(z_{ij}^{k+1} - A_{i,j}x_j^{k+1})^2}{2A_{i,j}} + \beta J(x^{k+1}) - \sum_{ij} \frac{(z_{ij}^{k+1} - A_{i,j}x_j^k)^2}{2A_{i,j}} - \beta J(x^k) \\
&\leq 0.
\end{aligned}$$

The first inequality comes from (2.35) and the second inequality comes from the M-step (2.31). When $E^G(x^{k+1}) = E^G(x^k)$, these two equalities have to be satisfied. The first equality is satisfied if and only if $x_j^{k+1} = x_j^k + \alpha$ for all j with α being a constant, while the second one is satisfied if and only if both x^k and x^{k+1} are minimizers of the M-step (2.31). The objective function of this optimization problem is strictly convex, and the minimizer is unique, which means α has to be 0. From the KKT conditions, we have

$$\beta \partial J(x^k)_j + \sum_i A_{i,j}x_j^k - \sum_i z_{ij}^{k+1} = 0 \qquad j = 1, \cdots, N.$$

After plugging the E-step (2.30) into this function, we have

$$\beta \partial J(x)_j + \sum_i \frac{A_{i,j}(y_i - (Ax^k)_i)}{w_i} = 0 \qquad j = 1, \cdots, N.$$

Therefore, x^k is one minimizer of the original problem. □

We can see, from the proof above, that the optimization problem of the M-step in each iteration does not have to be solved exactly. The log-likelihood function will still increase as long as $E_1^G(x^{k+1}, x^{k+1/2}) < E_1^G(x^k, x^{k+1/2})$ is satisfied before x^k converges.

The convergence can be proved in another way. If the M-step is solved exactly, we have

$$\beta \partial J(x^{k+1})_j + \sum_i A_{i,j} x_j^{k+1} - \sum_i z_{ij}^{k+1} = 0.$$

Multiplying by $(x_j^{k+1} - x_j^k)$ and taking summation over j gives us

$$\beta \sum_j (x_j^{k+1} - x_j^k) \partial J(x^{k+1})_j + \sum_{ij} A_{i,j} x_j^{k+1} (x_j^{k+1} - x_j^k) - \sum_{ij} z_{ij}^{k+1} (x_j^{k+1} - x_j^k) = 0.$$

From the convexity of $J(x)$, we have

$$J(x^k) \geq J(x^{k+1}) + (x^k - x^{k+1}) \partial J(x^{k+1}) = J(x^{k+1}) + \sum_j (x_j^k - x_j^{k+1}) \partial J(x^{k+1})_j.$$

Therefore we have

$$\begin{aligned}
0 &\geq \beta J(x^{k+1}) - \beta J(x^k) + \sum_{ij} A_{i,j} x_j^{k+1} (x_j^{k+1} - x_j^k) - \sum_{ij} z_{ij}^{k+1} (x_j^{k+1} - x_j^k) \\
&= E^G(x^{k+1}) - E^G(x^k) + \sum_{ij} (A_{i,j} x_j^{k+1} - z_{ij}^{k+1})(x_j^{k+1} - x_j^k) \\
&\quad + \sum_i \frac{(y_i - (Ax^k)_i)^2 - (y_i - (Ax^{k+1})_i)^2}{2w_i} \\
&= E^G(x^{k+1}) - E^G(x^k) + \sum_{ij} A_{i,j} (x_j^{k+1} - x_j^k)^2 - \sum_i \frac{((Ax^k)_i - (Ax^{k+1})_i)^2}{2w_i}
\end{aligned}$$

$$\geq E^G(x^{k+1}) - E^G(x^k) + \sum_{ij} A_{i,j}(x_j^{k+1} - x_j^k)^2 - \sum_i \frac{((Ax^k)_i - (Ax^{k+1})_i)^2}{w_i}$$
$$\geq E^G(x^{k+1}) - E^G(x^k).$$

The last inequality comes from Cauchy-Schwarz inequality and the equality holds if and only if $x_j^{k+1} = x_j^k + \alpha$ for all j with α being a constant. If we have $E^G(x^{k+1}) = E^G(x^k)$, from the second inequality, we have $Ax^k = Ax^{k+1}$. If constant vectors are not in the null space of A, which is the case satisfied by all the applications mentioned above, α has to be 0.

2.6.2 EM-Type Algorithms are Alternating Minimization Methods

As shown in the case of Poisson data, the algorithms can also be derived from an alternating minimization method of other problems with variables x and z. The new problems are

$$\begin{cases} \underset{x,z}{\text{minimize}} & E^G(x,z) := \sum_{ij} \frac{(z_{ij} - A_{i,j}x_j)^2}{2A_{i,j}} + \beta J(x), \\ \text{subject to} & \sum_j z_{ij} = y_i, \qquad i = 1, \cdots M. \end{cases} \tag{2.36}$$

Here E^G is used again to define the new function. $E^G(\cdot)$ means the negative log-likelihood function of x, while $E^G(\cdot,\cdot)$ means the new function of x and z defined in new optimization problems. The iteration is as follows:

$$z^{k+1} = \underset{z}{\text{argmin}}\ E(x^k, z), \qquad \text{subject to } \sum_j z_{ij} = y_i.$$
$$x^{k+1} = \underset{x}{\text{argmin}}\ E(x, z^{k+1}).$$

First, let us fix $x = x^k$ and update z. It is easy to derive

$$z_{ij}^{k+1} = A_{i,j}x_j^k + \frac{A_{i,j}}{w_i}\left(y_i - (Ax^k)_i\right).$$

Then, by fixing $z = z^{k+1}$ and updating x, we have

$$\begin{aligned} x^{k+1} &= \underset{x}{\operatorname{argmin}} \sum_{ij} \frac{(z_{ij} - A_{i,j}x_j)^2}{2A_{i,j}} + \beta J(x) \\ &= \underset{x}{\operatorname{argmin}} \frac{1}{2}\sum_j (\sum_i A_{i,j})(x_j - \frac{\sum\limits_i z_{ij}}{2\sum\limits_i A_{i,j}})^2 + \beta J(x). \end{aligned}$$

Since the problem (2.36) is convex, we can find the minimizer with respect to z for fixed x first and obtain a function of x as follows,

$$\sum_i \frac{((Ax)_i - y_i)^2}{2w_i} + \beta J(x), \tag{2.37}$$

which is also convex and equals $E^G(x)$. Therefore EM-Type algorithms will converge to the solution of (2.29).

2.6.3 Relaxation

In practice, other authors use a relaxation of SART reconstruction, which is

$$x_j^{k+1} = x_j^k + \frac{w}{\sum\limits_i A_{i,j}} \sum_i \frac{A_{i,j}(y_i - (Ax^k)_i)}{w_i},$$

with a relaxant coefficient w. The convergence of this relaxation is shown in [36, 37, 40] for any $w \in (0, 2)$. Inspired by this strategy, we have a relaxation of the EM-Type algorithms for image reconstruction with Gaussian noise. The EM-step is the relaxed SART with relaxant coefficient w:

$$x_j^{k+\frac{1}{2}} = x_j^k + \frac{w}{\sum\limits_i A_{i,j}} \sum_i \frac{A_{i,j}(y_i - (Ax^k)_i)}{w_i}.$$

The corresponding regularization step is

$$x^{k+1} = \underset{x}{\operatorname{argmin}} \ \frac{1}{2}\sum_j (\sum_i A_{i,j})(x_j - x_j^{k+\frac{1}{2}})^2 + w\beta J(x).$$

When $w = 1$, we have already discussed the convergence in the previous subsections by EM algorithms with *a priori* information and alternating minimization methods. For $w \neq 1$, we will show the convergence of the relaxed EM-Type algorithms for $w \in (0, 1)$ by alternating minimization methods.

We will show that the relaxed EM-Type algorithms are equivalent to solve the unconstrained problems

$$\underset{x,z}{\text{minimize}} \ E_R^G(x, z) := \sum_{ij} \frac{(z_{ij} - A_{i,j}x_j)^2}{2A_{i,j}} + \gamma \sum_i \frac{(\sum_j z_{ij} - y_i)^2}{2w_i} + w\beta J(x), \tag{2.38}$$

where $\gamma = \frac{w}{1-w}$, by alternating minimization between x and z. First, fix $x = x^k$, we can solve the problem of z only, and the analytical solution is

$$z_{ij}^{k+1} = A_{i,j}x_j^k + \frac{\gamma}{1+\gamma}\frac{A_{i,j}}{w_i}\left(y_i - (Ax^k)_i\right) = A_{i,j}x_j^k + w\frac{A_{i,j}}{w_i}\left(y_i - (Ax^k)_i\right). \tag{2.39}$$

Then let $z = z^{k+1}$ fixed, and we can find x^{k+1} by solving

$$\underset{x}{\text{minimize}} \ \sum_{ij} \frac{(z_{ij} - A_{i,j}x_j)^2}{2A_{i,j}} + w\beta J(x) = \frac{1}{2}\sum_j (\sum_i A_{i,j})(x_j - \frac{\sum_i z_{ij}}{\sum_i A_{i,j}})^2 + w\beta J(x) + C,$$

where C is a constant independent of x. Having z^{k+1} from (2.39), we can calculate

$$\frac{\sum_i z_{ij}^{k+1}}{\sum_i A_{i,j}} = x_j^k + \frac{w}{\sum_i A_{i,j}} \sum_i \frac{A_{i,j}(y_i - (Ax^k)_i)}{w_i} = x_j^{k+\frac{1}{2}}.$$

Therefore this relaxed EM-Type algorithm is an alternating minimization method. We will show next that the result of this relaxed EM-Type algorithm is the solution to (2.29).

Because the objective function $E_R^G(x, z)$ in (2.38) is convex, we can first minimize the

function with respect to z with x fixed. Then the problem becomes

$$\underset{x}{\text{minimize}} \ \frac{\gamma}{1+\gamma}\sum_i \frac{((Ax)_i - y_i)^2}{2w_i} + w\beta J(x) \tag{2.40}$$

$$= w\sum_i \frac{((Ax)_i - y_i)^2}{2w_i} + w\beta J(x). \tag{2.41}$$

We have shown in this subsection that the relaxed EM-Type algorithm will also converge to the solution of the original problem (2.29) when $\alpha \in (0, 1]$.

2.7 Numerical Experiments

In this section, several numerical experiments are provided to show the efficiency of EM-Type algorithms. Though these EM-Type algorithms can be used in many applications, we choose computed tomography (CT) image reconstruction as our application in this work. CT is a medical imaging method which utilizes X-ray equipment to produce a two dimensional (or three dimensional) image of the inside of an object from a large series of one dimensional (or two dimensional) X-ray images taken along a single axis of rotation [30]. In CT reconstruction, the operator A is the Radon transform, and the discrete version of A is constructed by Siddon's algorithm [42, 43].

We recall the continuous Radon transform in two dimensions, applied to an image function $f(x, y)$: assuming that a straight line in the plane can be represented by the equation depending on two parameters (θ, t),

$$x\cos\theta + y\sin\theta = t,$$

the Radon transform $f \mapsto g$ (with $(\theta, t) \mapsto g(\theta, t)$ the image in the sinogram domain (θ, t) of the function $(x, y) \mapsto f(x, y)$) is defined by the projections along lines of f,

$$g(\theta, t) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x, y)\delta(x\cos\theta + y\sin\theta - t)dxdy,$$

where δ is the one-dimensional Dirac delta function, with support on the line of equation $x\cos\theta + y\sin\theta = t$. In the two-dimensional discrete case, f represents a discrete image matrix and g (or a noisy version of g) is known only at a finite number of samples (θ_i, t_i). In computed tomography, we must recover f from its projections g. Although the Radon transform is an invertible operator, in the real discrete case only a small finite number of projections (θ_i, t_i) are given, which may be also noisy. Thus, in the real case, the recovery of f from projections g is a difficult inverse problem. For the rest of the presentation, we will continue to use the notations x for the image to be recovered, y for the measurements or projections in the sinogram domain, and A for the discrete Radon transform with a finite number of samples.

The problem is to reconstruct the image from the measurements, which is equivalent to solving $Ax = y$. Poisson noise is assumed. Total variation (TV) and a modified version of TV are chosen to be the regularization. In order to compare the results of different methods, root mean square error (RMSE) is utilized.

2.7.1 CT Reconstruction by EM-TV (2D)

At first, we illustrate one method (EM-TV) at a simple synthetic object (two dimensional 256x256 Shepp-Logan phantom), see Figure 2.1.

Original x

Figure 2.1: 2D Shepp-Logan phantom

The most common method used in commercial CT is filtered back projection (FBP), which is straightforward to implement and can be computed rapidly. However, FBP has limitations due to the presence of streak artifacts and noise enhancement, which is inherent

in the reconstruction. Furthermore, in order to obtain an accurate image, many views are taken. Algorithms that can perform accurate image reconstruction from few views are very important in reducing patient dose and speeding up scans. Optimization based methods, including EM-TV, can reconstruct images from few views, but require more computing time. However, with the development of graphics processing units (GPUs), the computing time has been reduced greatly and this kind of technique becomes more and more important.

In the following experiment, we will compare the reconstruction results obtained by EM-TV with those obtained by filtered back projection. To solve the above minimization problem (2.18) with $J(x)$ being the total variation, we use an iterative semi-implicit finite-differences scheme. Each iteration is called a TV step. Given x^{EM}, already computed from the EM step, we compute the new x discretizing (2.17) (as a discrete minimizer of (2.18)), by the following simple iterative scheme, presented below in two spatial dimensions and assuming now that x is represented as a 2D matrix: from an initial guess x^0, we compute x^{n+1} with $n \geq 0$ from the following linearized discrete equation,

$$
\begin{aligned}
&- \beta \frac{x^n_{i,j}}{v_{i,j}} \frac{x^n_{i+1,j} - x^{n+1}_{i,j}}{\sqrt{\epsilon + (x^n_{i+1,j} - x^n_{i,j})^2 + (x^n_{i,j+1} - x^n_{i,j})^2}} \\
&+ \beta \frac{x^n_{i,j}}{v_{i,j}} \frac{x^{n+1}_{i,j} - x^n_{i-1,j}}{\sqrt{\epsilon + (x^n_{i,j} - x^n_{i-1,j})^2 + (x^n_{i-1,j+1} - x^n_{i-1,j})^2}} \\
&- \beta \frac{x^n_{i,j}}{v_{i,j}} \frac{x^n_{i,j+1} - x^{n+1}_{i,j}}{\sqrt{\epsilon + (x^n_{i+1,j} - x^n_{i,j})^2 + (x^n_{i,j+1} - x^n_{i,j})^2}} \\
&+ \beta \frac{x^n_{i,j}}{v_{i,j}} \frac{x^{n+1}_{i,j} - x^n_{i,j-1}}{\sqrt{\epsilon + (x^n_{i+1,j-1} - x^n_{i,j-1})^2 + (x^n_{i,j} - x^n_{i,j-1})^2}} + x^{n+1}_{i,j} - x^{EM}_{i,j} = 0,
\end{aligned}
\tag{2.42}
$$

where ϵ is a small constant and $v_{i,j} = \sum_{i'=1}^{M} A_{i',j'}$ with j' being the index corresponding to the 2D index (i, j).

Finally, the two steps (EM and TV) are solved in an alternating fashion. Usually, for each main iteration, we apply 2 or 3 EM steps, followed by 5 to 8 TV steps. For the TV step, the initial guess can be defined as the result from the previous EM update, or from the last TV update [16].

The measurements are obtained using Siddon's algorithm. We consider both the noise-free and noisy cases. For the FBP method, we present results using 36 views (every 10 degrees; for each view there are 301 measurements), 180 views, and 360 views. In order to show that we can reduce the number of views by using EM-TV, we only use 36 views for the proposed method. The results are shown in Figure 2.2. We notice the much improved results obtained with EM-TV using only 36 views, by comparison with FBP using 36, 180 or even 360 views.

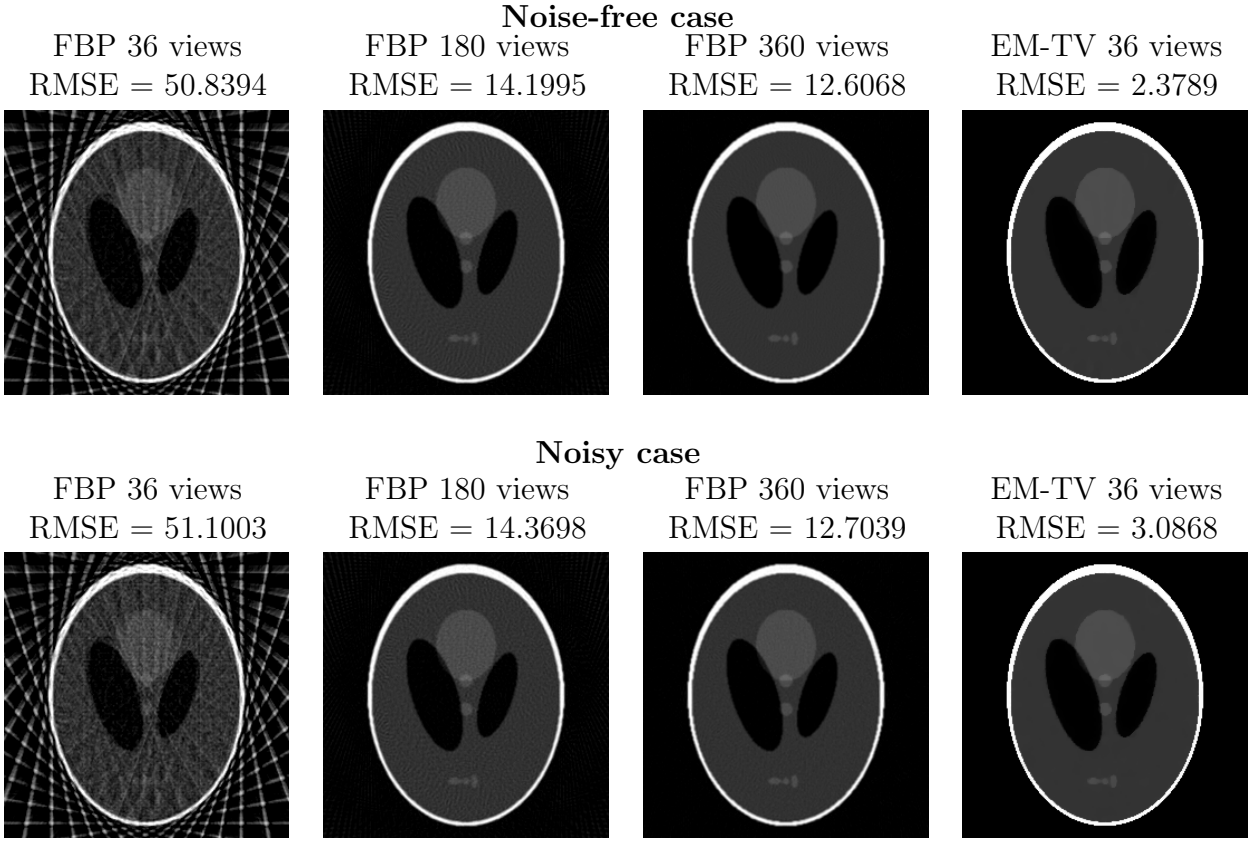


Figure 2.2: Top from left to right: reconstruction result in the noise-free case using FBP with 36, 180 and 360 views, and result using EM-TV with 36 views. Bottom from left to right: reconstruction result in the noisy case using FBP with 36, 180 and 360 views, and result using EM-TV with 36 views. The root mean square errors are also given.

2.7.2 Reconstruction using EM-MSTV (2D)

Instead of TV regularization, we also show the results by using a modified TV, which is called Mumford-Shah TV (MSTV) [44]. The new regularization is

$$J(x,v) = \int_\Omega v^2 |\nabla x| + \alpha \int_\Omega \left(\epsilon |\nabla v|^2 + \frac{(v-1)^2}{4\epsilon} \right),$$

which has two variables x and v, and Ω is the image domain. It is shown by Alicandro et al. [45] that $J(x,v)$ will Γ-convergent to

$$\int_{\Omega \setminus K} |\nabla x| + \alpha \int_K \frac{|x^+ - x^-|}{1 + |x^+ - x^-|} d\mathcal{H}^1 + |D^c x|(\Omega),$$

where x^+ and x^- denote the image values on two sides of the edge set K, $\mathcal{H}^1$ is the one-dimensional Hausdorff measure and $D^c x$ is the Cantor part of the measure-valued derivative Dx.

The comparisons of EM-TV and EM-MSTV in both noise-free and noisy cases are in Figure 2.3. From the results, we can see that with MSTV, the reconstructed images will be better than with TV only, visually and according to the RMSE.

2.7.3 Reconstruction using EM-TV (3D)

In this experiment, we will show the reconstruction results by EM-TV for three dimensional images. The image chosen is the $128 \times 128 \times 128$ Shepp-Logan phantom, and the sinogram data is obtained from 36 views. The result is compared with that obtained by using EM only in Figure 2.4.

2.8 Conclusion

In this chapter, we proposed general robust EM-Type algorithms for image reconstruction without background emission. Both Poisson noise and Gaussian noise are considered. The EM-Type algorithms are performed using iteratively EM (or SART for weighted Gaussian

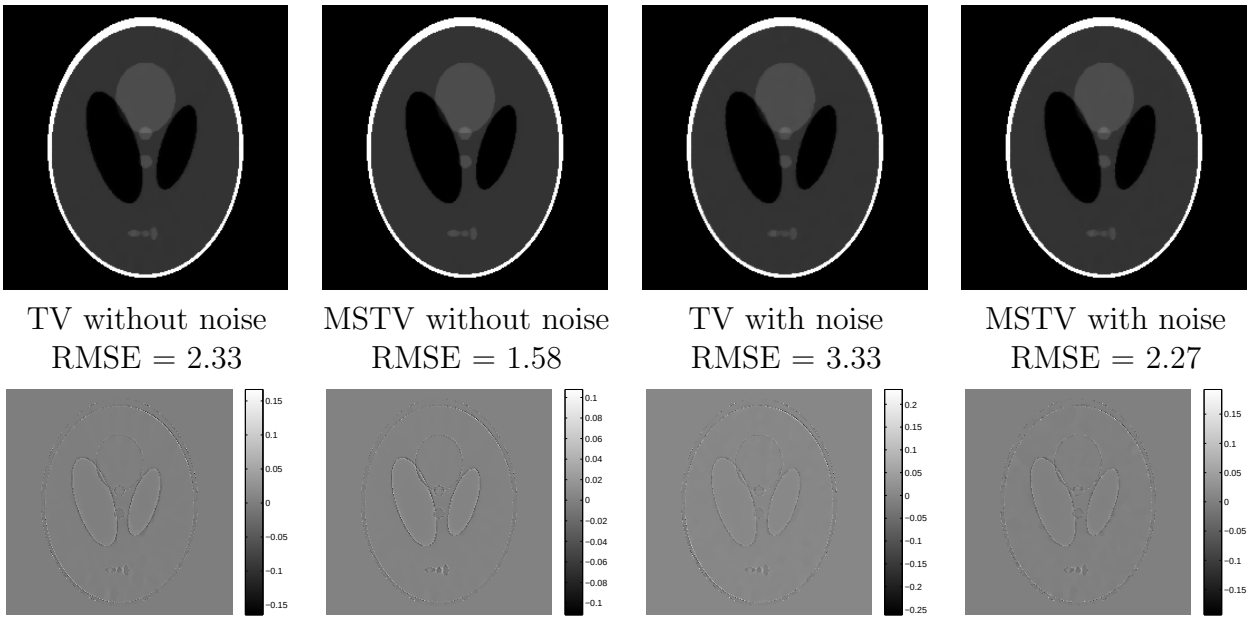


Figure 2.3: Comparisons of TV regularization and MSTV regularization for both without and with noise cases. Top row shows the reconstructed images by these two methods in both cases, Bottom row shows the differences between the reconstructed images and original phantom image. The RMSEs and differences show that MSTV can provide better results than TV only.

noise) and regularization in the image domain. The convergence of these algorithms is proved in several ways: EM with *a priori* information and alternating minimization methods. To show the efficiency of EM-Type algorithms, the application in CT reconstruction is chosen. We compared EM-TV and EM-MSTV for 2D CT reconstruction. Both methods can give us good results by using undersampled data comparing to the filtered back projection. Results from EM-MSTV have sharper edges than those from EM-TV. Also EM-TV is used for 3D CT reconstruction and the performance is better than using EM only (without regularization term) for undersampled data.

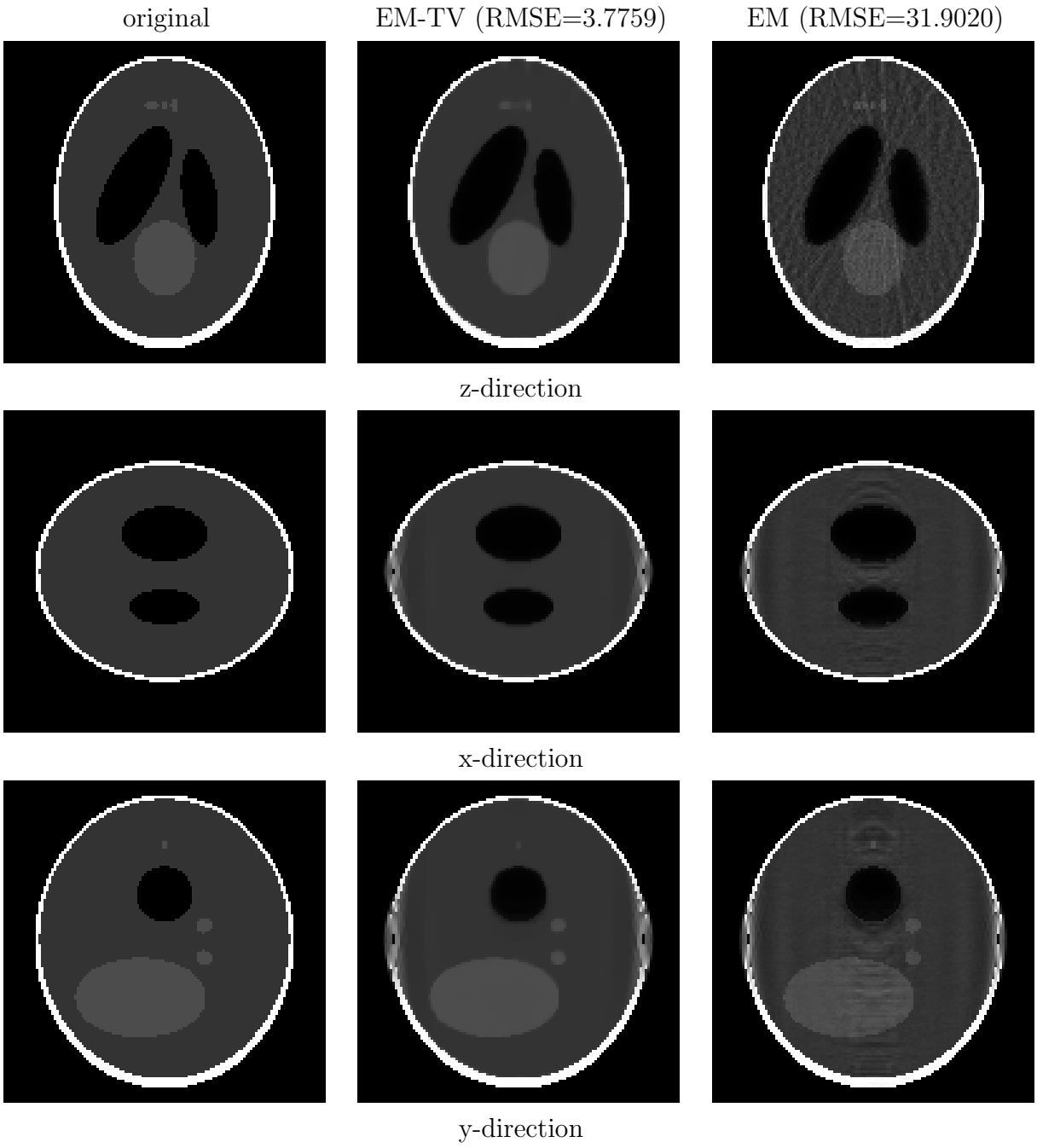


Figure 2.4: Reconstruction results in three dimensions for the noise-free case. First column: two-dimensional views of the original three-dimensional Shepp-Logan phantom. Middle column: two-dimensional views of reconstruction results obtained using EM-TV algorithm. Last column: two-dimensional views of reconstruction results obtained using EM iteration. The root mean square errors are also given.

CHAPTER 3

General Convergent EM-Type Algorithms for Image Reconstruction With Background Emission and Poisson Noise

3.1 Introduction

As mentioned in the previous chapter, the degradation model can be formulated as a linear inverse and ill-posed problem,

$$y = Ax + b + n. \tag{3.1}$$

Here, y is the measured data (vector in $\mathbf{R}^M$ for the discrete case). A is a compact operator (matrix in $\mathbf{R}^{M\times N}$ for the discrete case). For all the applications we will consider, the entries of A are nonnegative and A does not have full column rank. x is the desired exact image (vector in $\mathbf{R}^N$ for the discrete case). b is the background emission, which is assumed to be known, and n is the noise. In the last chapter, we considered the case without background emission ($b = 0$), and the case with background emission ($b \neq 0$) is considered in this chapter. In astronomy, this is due to sky emission [46, 47], while in fluorescence microscopy, it is due to auto-fluorescence and reflections of the excitation light. Since the matrix A does not have full column rank, the computation of x directly by finding the inverse of A is not reasonable because (3.1) is ill-posed and n is unknown. Therefore regularization techniques are needed for solving these problems efficiently.

Same as in the last chapter, we assume that measured data y is a realization of a multi-valued random variable, denoted by Y and the image x is also considered as a realization of another multi-valued random variable, denoted by X. Therefore the Bayesian formula gives

us

$$p_X(x|y) = \frac{p_Y(y|x)p_X(x)}{p_Y(y)}. \tag{3.2}$$

This is a conditional probability of having $X = x$ given that y is the measured data. After inserting the detected value of y, we obtain *a posteriori* probability distribution of X. Then we can find x^* such that $p_X(x|y)$ is maximized, as maximum *a posteriori* (MAP) likelihood estimation.

In general, X is assigned as a Gibbs random field, which is a random variable with the following probability distribution

$$p_X(x) \sim e^{-\beta J(x)}, \tag{3.3}$$

where $J(x)$ is a given convex energy function, and β is a positive parameter. The choice of $p_Y(y|x)$ depends on the noise model. If the random variable Y of the detected values y follows a Poisson distribution [20, 21] with an expectation value provided by $Ax+b$, we have

$$y_i \sim \text{Poisson}\{(Ax+b)_i\}, \quad i.e., \quad p_Y(y|x) \sim \prod_i \frac{(Ax+b)_i^{y_i}}{y_i!} e^{-(Ax+b)_i}. \tag{3.4}$$

By minimizing the negative log-likelihood function, we obtain the following optimization problem

$$\underset{x\geq 0}{\text{minimize}} \sum_i \big((Ax+b)_i - y_i \log(Ax+b)_i\big) + \beta J(x). \tag{3.5}$$

In this chapter, we will focus on solving (3.5). It is easy to see that the objective function in (3.5) is convex when $J(x)$ is convex. Additionally, with suitably chosen regularization $J(x)$, the objective function is strictly convex, and the solution to this problem is unique.

The work is organized as follows. In section 3.2, we will give a short introduction of expectation maximization (EM) iteration, or the Richardson-Lucy algorithm, used in image reconstruction with background emission from the view of optimization. In section 3.3, we will propose general EM-Type algorithms for image reconstruction with background emission when the measured data is corrupted by Poisson noise. This is based on the maximum a

posteriori likelihood estimation and EM step. In this section, these EM-Type algorithms are shown to be equivalent to EM algorithms with *a priori* information, and their convergence is shown in two different ways. In addition, these EM-Type algorithms are also considered as alternating minimization methods for equivalent optimization problems. For the case without regularization, more analysis on the convergence (the distance to the solution is decreasing) is provided. However, for some regularizations, the reconstructed images will lose contrast. To overcome this problem, EM-Type algorithms with Bregman iteration are introduced in section 3.4. Some numerical experiments are given in section 3.5 to show the efficiency of the EM-Type algorithms with different regularizations. We will end this work by a short conclusion section.

3.2 Expectation Maximization (EM) Iteration

A maximum likelihood (ML) method for image reconstruction based on Poisson data was introduced by Shepp and Vardi [21] in 1982 for applications in emission tomography. In fact, this algorithm was originally proposed by Richardson [23] in 1972 and Lucy [24] in 1974 for astronomy. In this section, we consider the special case without regularization term, i.e., $J(x)$ is a constant, we do not have any *a priori* information about the image. From equation (3.4), for given measured data y, we have a function of x, the likelihood of x, defined by $p_Y(y|x)$. Then a ML estimate of the unknown image is defined as any maximizer x^* of $p_Y(y|x)$.

By taking the negative log-likelihood, one obtains, up to an additive constant

$$f_0(x) = \sum_i \big((Ax+b)_i - y_i \log(Ax+b)_i\big), \tag{3.6}$$

and the problem is to minimize this function $f_0(x)$ on the nonnegative orthant, because we have the constraint that the image x is nonnegative. In fact, we have

$$f(x) = D_{KL}(y, Ax+b) := \sum_i \Big(y_i \log \frac{y_i}{(Ax+b)_i} + (Ax+b)_i - y_i\Big) = f_0(x) + C,$$

where $D_{KL}(y, Ax+b)$ is the Kullback-Leibler (KL) divergence of $Ax+b$ from y, and C is a constant independent of x. The KL divergence is considered as a data-fidelity function for Poisson data just like the standard least-square $\|Ax+b-y\|_2^2$ is the data-fidelity function for additive Gaussian noise. It is convex, nonnegative and coercive on the nonnegative orthant, so the minimizers exist and are global.

In order to find a minimizer of $f(x)$ with the constraint $x \geq 0$, we can solve the Karush-Kuhn-Tucker (KKT) conditions [25, 26],

$$\begin{aligned} \sum_i \left(A_{i,j}(1 - \frac{y_i}{(Ax+b)_i}) \right) - s_j &= 0, \qquad j = 1, \cdots, N, \\ s_j \geq 0, \quad x_j &\geq 0, \qquad j = 1, \cdots, N, \\ s^T x &= 0. \end{aligned}$$

Here s_j is the Lagrangian multiplier corresponding to the constraint $x_j \geq 0$. By the positivity of $\{x_j\}$, $\{s_j\}$ and the complementary slackness condition $s^T x = 0$, we have $s_j x_j = 0$ for every $j = 1, \cdots, N$. Multiplying by x_j gives us

$$\sum_i \left(A_{i,j}(1 - \frac{y_i}{(Ax+b)_i}) \right) x_j = 0, \qquad j = 1, \cdots, N. \tag{3.7}$$

Therefore, we have the following iterative scheme

$$x_j^{k+1} = \frac{\sum_i \left(A_{i,j}(\frac{y_i}{(Ax^k+b)_i}) \right)}{\sum_i A_{i,j}} x_j^k. \tag{3.8}$$

This is the well-known EM iteration or Richardson-Lucy algorithm in image reconstruction, and an important property of it is that it preserves positivity. If x^k is positive, then x^{k+1} is also positive if A preserves positivity.

Shepp and Vardi showed in [21] that when $b = 0$, this is equivalent to the EM algorithm proposed by Dempster, Laird and Rubin [1]. Actually, when $b \neq 0$, this is also equivalent to the EM algorithm and it will be shown in the next section. To make it clear, EM iteration means the special EM method used in image reconstruction, while EM algorithm means the

general EM algorithm for solving missing data problems.

3.3 EM-Type Algorithms for Image Reconstruction

The method shown in the last section is also called maximum-likelihood expectation maximization (ML-EM) reconstruction, because it is a maximum likelihood approach without any Bayesian assumption on the images. If additional *a priori* information about the image is given, we have maximum a posteriori probability (MAP) approach [27, 28], which is the case with regularization term $J(x)$. Again we assume here that the detected data is corrupted by Poisson noise, and the regularization problem is

$$\begin{cases} \underset{x}{\text{minimize}} & E^P(x) := \beta J(x) + \sum_i \left((Ax+b)_i - y_i \log(Ax+b)_i \right), \\ \text{subject to} & x_j \geq 0, \quad j = 1, \cdots, N. \end{cases} \tag{3.9}$$

This is still a convex constraint optimization problem when $J(x)$ is convex and we can find the optimal solution by solving the KKT conditions:

$$\begin{aligned} \beta \partial J(x)_j + \sum_i \left(A_{i,j}(1 - \frac{y_i}{(Ax+b)_i}) \right) - s_j &= 0, \qquad j = 1, \cdots, N, \\ s_j \geq 0, \quad x_j &\geq 0, \qquad j = 1, \cdots, N, \\ s^T x &= 0. \end{aligned}$$

Here s_j is the Lagrangian multiplier corresponding to the constraint $x_j \geq 0$. By the positivity of $\{x_j\}$, $\{s_j\}$ and the complementary slackness condition $s^T x = 0$, we have $s_j x_j = 0$ for every $j = 1, \cdots, N$. Thus we obtain

$$\beta x_j \partial J(x)_j + \sum_i \left(A_{i,j}(1 - \frac{y_i}{(Ax+b)_i}) \right) x_j = 0, \qquad j = 1, \cdots, N,$$

or equivalently

$$\beta \frac{x_j}{\sum_i A_{i,j}} \partial J(x)_j + x_j - \frac{\sum_i \left(A_{i,j} (\frac{y_i}{(Ax+b)_i}) \right)}{\sum_i A_{i,j}} x_j = 0, \qquad j = 1, \cdots, N.$$

Notice that the last term on the left hand side is an EM step (3.8). After plugging the EM step into the equation, we obtain

$$\beta \frac{x_j}{\sum_i A_{i,j}} \partial J(x)_j + x_j - x_j^{EM} = 0, \qquad j = 1, \cdots, N,$$

which is the optimality condition for the following optimization problem

$$\underset{x}{\text{minimize}} \;\; E_1^P(x, x^{EM}) := \beta J(x) + \sum_j (\sum_i A_{i,j}) \left(x_j - x_j^{EM} \log x_j \right). \tag{3.10}$$

Therefore we propose the general EM-Type algorithms in Algorithm **3**. The initial guess x^0 can be any positive initial image, and ϵ, chosen for the stopping criteria, is very small. Num_Iter is the maximum number of iterations. If $J(x)$ is constant, the second step is just $x^k = x^{k-\frac{1}{2}}$ and this is exactly the ML-EM from the previous section. When $J(x)$ is not constant, we have to solve an optimization problem for each iteration. In general, the problem can not be solved analytically, and we have to use iterative methods to solve it. However, in practice, we do not have to solve it exactly by stopping it after a few iterations. We will show that the algorithms will also converge without solving it exactly.

Algorithm 3 Proposed EM-Type algorithms.

Input: x^0, ϵ
Initialization: $k = 1$
while $k < Num\_Iter$ & $\|x^k - x^{k-1}\| < \epsilon$ **do**
  $x^{k-\frac{1}{2}} = EM(x^{k-1})$ using (3.8)
  $x^k = \text{argmin} \;\; E_1^P(x, x^{k-\frac{1}{2}})$ by solving (3.10)
  k=k+1
end while

3.3.1 Equivalence to EM Algorithms with *a priori* Information

In this subsection, the EM-Type algorithms are shown to be equivalent to EM algorithms with *a priori* information. The EM algorithm is a general approach for maximizing *a posterior* distribution when some of the data is missing [1]. It is an iterative method that alternates between expectation (E) steps and maximization (M) steps. For image reconstruction, we assume that the missing data is $\{z_{ij}\}$, describing the intensity of pixel (or voxel) j observed by detector i and $\{\bar{b}_i\}$, the intensity of background observed by detector i. Therefore the observed data are $y_i = \sum_j z_{ij} + \bar{b}_i$. We can have the assumption that z is a realization of multi-valued random variable Z, and for each (i, j) pair, z_{ij} follows a Poisson distribution with expected value $A_{i,j}x_j$, and $\bar{b}_i$ follows a Poisson distribution with expected value b_i, because the summation of two Poisson distributed random variables also follows a Poisson distribution, whose expected value is summation of the two expected values.

The original E-step is to find the expectation of the log-likelihood given the present variables x^k:

$$Q(x|x^k) = E_{z|x^k,y} \log p(x, z|y)$$

Then, the M-step is to choose x^{k+1} to maximize the expected log-likelihood $Q(x|x^k)$ found in the E-step:

$$\begin{aligned} x^{k+1} &= \underset{x}{\operatorname{argmax}}\ E_{z|x^k,y} \log p(x, z|y) = \underset{x}{\operatorname{argmax}}\ E_{z|x^k,y} \log(p(y, z|x)p(x)) \\ &= \underset{x}{\operatorname{argmax}}\ E_{z|x^k,y} \sum_{ij} (z_{ij} \log(A_{i,j}x_j) - A_{i,j}x_j) - \beta J(x) \\ &= \underset{x}{\operatorname{argmin}} \sum_{ij} (A_{i,j}x_j - E_{z|x^k,y} z_{ij} \log(A_{i,j}x_j)) + \beta J(x). \end{aligned} \tag{3.11}$$

From (3.11), what we need before solving it is just $\{E_{z|x^k,y} z_{ij}\}$. Therefore we compute the expectation of missing data $\{z_{ij}\}$ given present x^k, denoting this as an E-step. Because for fixed i, $\{z_{ij}\}$ are Poisson variables with mean $\{A_{i,j}x_j^k\}$ and $\bar{b}_i$ is Poisson variable with mean b_i, then the distribution of z_{ij} is binomial distribution $\left(y_i, \frac{A_{i,j}x_j^k}{(Ax^k+b)_i}\right)$, thus we can find the

expectation of z_{ij} with all these conditions by the following E-step

$$z_{ij}^{k+1} = E_{z|x^k,y} z_{ij} = \frac{A_{i,j} x_j^k y_i}{(Ax^k + b)_i}, \qquad \bar{b}_i^{k+1} = \frac{b_i y_i}{(Ax^k + b)_i}. \tag{3.12}$$

After obtaining the expectation for all z_{ij}, then we can solve the M-step (3.11).

We will show that EM-Type algorithms are exactly the described EM algorithms with *a priori* information. Recalling the definition of x^{EM}, we have

$$x_j^{EM} = \frac{\sum\limits_i z_{ij}^{k+1}}{\sum\limits_i A_{i,j}}. \tag{3.13}$$

Therefore, the M-step is equivalent to

$$\begin{aligned} x^{k+1} &= \operatorname*{argmin}_x \sum_{ij} (A_{i,j} x_j - z_{ij}^{k+1} \log(A_{i,j} x_j)) + \beta J(x) \\ &= \operatorname*{argmin}_x \sum_j (\sum_i A_{i,j})(x_j - x_j^{EM} \log(x_j)) + \beta J(x). \end{aligned}$$

We have shown that EM-Type algorithms are EM algorithms with *a priori* information. The convergence of EM-Type algorithms is shown in the next subsection from the convergence of the EM algorithms with *a priori* information.

3.3.2 Convergence of EM-Type Algorithms

In this subsection, we will show that the negative log-likelihood is decreasing in the following theorem.

Theorem 3.3.1. *The objective function (negative log-likelihood) $E^P(x^k)$ in (3.9) with x^k given by Algorithm 3 will decrease until it attaints a minimum.*

Proof. For all k and i, we always have the constraint satisfied

$$\sum_j z_{ij}^k + \bar{b}_i^k = y_i.$$

Therefore, we have the following inequality

$$
\begin{aligned}
& y_i \log\left((Ax^{k+1}+b)_i\right) - y_i \log\left((Ax^{k}+b)_i\right) \\
&= y_i \log\left(\frac{(Ax^{k+1}+b)_i}{(Ax^{k}+b)_i}\right) = y_i \log\left(\frac{\sum_j A_{i,j}x_j^{k+1}+b_i}{(Ax^{k}+b)_i}\right) \\
&= y_i \log\left(\sum_j \frac{A_{i,j}x_j^k x_j^{k+1}}{(Ax^k+b)_i x_j^k} + \frac{b_i}{(Ax^k+b)_i}\right) \\
&= y_i \log\left(\sum_j \frac{z_{ij}^{k+1} A_{i,j}x_j^{k+1}}{y_i A_{i,j}x_j^k} + \frac{\bar{b}_i^{k+1}}{y_i}\right) \\
&\geq y_i \sum_j \frac{z_{ij}^{k+1}}{y_i} \log\left(\frac{A_{i,j}x_j^{k+1}}{A_{i,j}x_j^k}\right) \qquad \text{(Jensen's inequality)} \\
&= \sum_j z_{ij}^{k+1} \log(A_{i,j}x_j^{k+1}) - \sum_j z_{ij}^{k+1} \log(A_{i,j}x_j^k). \qquad (3.14)
\end{aligned}
$$

This inequality gives us

$$
\begin{aligned}
E^P(x^{k+1}) - E^P(x^k) &= \sum_i ((Ax^{k+1}+b)_i - y_i \log(Ax^{k+1}+b)_i) + \beta J(x^{k+1}) \\
&\quad - \sum_i \left((Ax^k+b)_i - y_i \log(Ax^k+b)_i\right) - \beta J(x^k) \\
&\leq \sum_{ij} (A_{i,j}x_j^{k+1} - z_{ij}^{k+1} \log(A_{i,j}x_j^{k+1})) + \beta J(x^{k+1}) \\
&\quad - \sum_{ij} (A_{i,j}x_j^k - z_{ij}^{k+1} \log(A_{i,j}x_j^k)) - \beta J(x^k) \\
&\leq 0.
\end{aligned}
$$

The first inequality comes from (3.14) and the second inequality comes from the M-step (3.11). When $E^P(x^{k+1}) = E^P(x^k)$, these two equalities have to be satisfied. The first equality is satisfied if and only if $x_j^{k+1} = x_j^k$ for all j, while the second one is satisfied if and only if x^k and x^{k+1} are minimizers of the M-step (3.11). Since the objective function to be

minimized in M-step (3.11) is strictly convex, we have

$$\beta x_j^k \partial J(x^k)_j + \sum_i A_{i,j} x_j^k - \sum_i z_{ij}^{k+1} = 0, \qquad j = 1, \cdots, N,$$

after plugging the E-step (3.12) into these equations, we have

$$\beta x_j^k \partial J(x^k)_j + \sum_i A_{i,j} x_j^k - \sum_i \frac{A_{i,j} x_j^k y_i}{(Ax^k + b)_i} = 0, \qquad j = 1, \cdots, N.$$

Therefore, x^k is one minimizer of the original problem. □

The log-likelihood function will increase for each iteration until the solution is found, and from the proof, we do not fully use the M-step. Even if the M-step is not solved exactly, it will still increase as long as $Q(x^{k+1}|x^k) > Q(x^k|x^k)$ is satisfied before x^k converges.

The increasing of log-likelihood function can be proved in another way by using the M-step. From $x^{k+1} = \underset{x}{\operatorname{argmax}}\ Q(x|x^k)$, we have

$$\beta x_j^{k+1} \partial J(x^{k+1})_j + \sum_i A_{i,j} x_j^{k+1} - \sum_i z_{ij}^{k+1} = 0, \qquad j = 1, \cdots, N.$$

Multiplying by $(x_j^{k+1} - x_j^k)/x_j^{k+1}$ and taking summation over j gives us

$$\beta \sum_j (x_j^{k+1} - x_j^k) \partial J(x^{k+1})_j + \sum_{ij} A_{i,j}(x_j^{k+1} - x_j^k) - \sum_{ij} z_{ij}^{k+1} \frac{x_j^{k+1} - x_j^k}{x_j^{k+1}} = 0.$$

From the convexity of $J(x)$, we have

$$J(x^k) \geq J(x^{k+1}) + (x^k - x^{k+1}) \partial J(x^{k+1}) = J(x^{k+1}) + \sum_j (x_j^k - x_j^{k+1}) \partial J(x^{k+1}).$$

Therefore we have

$$0 \geq \beta J(x^{k+1}) - \beta J(x^k) + \sum_{ij} A_{i,j}(x_j^{k+1} - x_j^k) - \sum_{ij} z_{ij}^{k+1} \frac{x_j^{k+1} - x_j^k}{x_j^{k+1}}$$

$$
\begin{aligned}
&= E^P(x^{k+1}) - E^P(x^k) + \sum_i y_i \log\left(\frac{(Ax^{k+1}+b)_i}{(Ax^k+b)_i}\right) - \sum_{ij} z_{ij}^{k+1}\frac{x_j^{k+1}-x_j^k}{x_j^{k+1}} \\
&\geq E^P(x^{k+1}) - E^P(x^k) + \sum_i y_i \left(1 - \frac{(Ax^k+b)_i}{(Ax^{k+1}+b)_i}\right) - \sum_{ij} z_{ij}^{k+1}\frac{x_j^{k+1}-x_j^k}{x_j^{k+1}} \\
&= E^P(x^{k+1}) - E^P(x^k) - \sum_i y_i \frac{\sum_j A_{i,j}x_j^k + b_i}{\sum_j A_{i,j}x_j^{k+1} + b_i} + \sum_{ij} z_{ij}^{k+1}\frac{x_j^k}{x_j^{k+1}} + \sum_i (y_i - \sum_j z_{ij}^{k+1}) \\
&\geq E^P(x^{k+1}) - E^P(x^k).
\end{aligned}
$$

The second inequality comes from $\log(x) \geq 1 - 1/x$ for $x > 0$, and the last inequality comes from Cauchy-Schwarz inequality. If $E^P(x^{k+1}) = E^P(x^k)$, from the last inequality, we have $x_j^{k+1} = x_j^k$ for all j. Therefore, the log-likelihood function will increase until the solution is found.

3.3.3 EM-Type Algorithms are Alternating Minimization Methods

In this section, we will show that these algorithms can also be derived from alternating minimization methods of other problems with variables x and z. The new optimization problems are

$$
\begin{aligned}
\underset{x,z}{\text{minimize}} \;\; E^P(x,z) := &\sum_{ij}\left(z_{ij}\log\frac{z_{ij}}{A_{i,j}x_j} + A_{i,j}x_j - z_{ij}\right) \\
&+ \sum_i \left(\bar{b}_i \log\frac{\bar{b}_i}{b_i} + b_i - \bar{b}_i\right) + \beta J(x),
\end{aligned} \tag{3.15}
$$

where $\bar{b}_i = y_i - \sum_j z_{ij}$, for all $i = 1, \cdots, M$. Here E^P is used again to define the new function. $E^P(\cdot)$ means the negative log-likelihood function of x, while $E^P(\cdot,\cdot)$ means the new function of x and z defined in new optimization problems.

Having initial guess x^0, z^0 of x and z, the iteration for $k = 0, 1, \cdots$ is as follows:

$$
\begin{aligned}
z^{k+1} &= \underset{z}{\text{argmin}} \;\; E^P(x^k, z), \\
x^{k+1} &= \underset{x}{\text{argmin}} \;\; E^P(x, z^{k+1}).
\end{aligned}
$$

Firstly, in order to obtain z^{k+1}, we fix $x = x^k$ and easily derive

$$z_{ij}^{k+1} = \frac{A_{i,j}x_j^k y_i}{(Ax^k + b)_i}. \tag{3.16}$$

After finding z^{k+1}, let $z = z^{k+1}$ fixed and update x, then we have

$$\begin{aligned} x^{k+1} &= \underset{x}{\operatorname{argmin}} \sum_{ij} \left(A_{i,j}x_j + z_{ij}^{k+1} \log \frac{z_{ij}^{k+1}}{A_{i,j}x_j} \right) + \beta J(x) \\ &= \underset{x}{\operatorname{argmin}} \sum_{ij} \left(A_{i,j}x_j - z_{ij}^{k+1} \log(A_{i,j}x_j) \right) + \beta J(x), \end{aligned}$$

which is the M-Step (3.11) in section 3.3.1. The equivalence of problems (3.9) and (3.15) is provided in the following theorem.

Theorem 3.3.2. *If (x^*, z^*) is a solution of problem (3.15), then x^* is also a solution of (3.9), i.e., $x^* = \underset{x}{\operatorname{argmin}}\ E^P(x)$. If x^* is a solution of (3.9), then we can find z^* from (3.16) and (x^*, z^*) is a solution of problem (3.15).*

Proof. The equivalence can be proved in two steps. Firstly, we will show that $E^P(x, z) \geq E^P(x) + C$ for all z, here C is constant dependent on y only:

$$\begin{aligned} E^P(x,z) &= \sum_{ij} \left(z_{ij} \log \frac{z_{ij}}{A_{i,j}x_j} + A_{i,j}x_j - z_{ij} \right) + \sum_i \left(\bar{b}_i \log \frac{\bar{b}_i}{b_i} + b_i - \bar{b}_i \right) + \beta J(x) \\ &= \sum_{ij} \left(\frac{z_{ij}}{y_i} \log \frac{z_{ij}}{A_{i,j}x_j} \right) y_i + \sum_i \frac{\bar{b}_i}{y_i} \log \frac{\bar{b}_i}{b_i} y_i + \sum_i \left((Ax+b)_i - y_i \right) + \beta J(x) \\ &\geq \sum_i y_i \log \left(\frac{y_i}{(Ax+b)_i} \right) + \sum_i \left((Ax+b)_i - y_i \right) + \beta J(x) \\ &= E^P(x) + \sum_i (y_i \log y_i - y_i). \end{aligned}$$

The inequality comes form Jensen's inequality, and the equality is satisfied if and only if

$$\frac{z_{ij}}{A_{i,j}x_j} = \frac{\bar{b}_i}{b_i} = C_i, \qquad \forall j = 1, \cdots, N, \tag{3.17}$$

where C_i are constants, which depends on x, y and i and can be found from the constraint

$\sum_j z_{ij} + \bar{b}_i = y_i$. Therefore $\min_z E^P(x,z) = E^P(x) + C$, which means that problems (3.15) and (3.9) are equivalent. □

From these two convergence analyses, if the second part of the EM-Type algorithms can not be solved exactly, we can choose the initial guess to be the result from the previous iteration, then use any method for solving convex optimization problem to obtain a better result.

3.3.4 Further Analysis for the Case Without Regularization

For the case without regularization, we will show that for each limit point $\tilde{x}$ of the sequence $\{x^k\}$, we have $D_{KL}(\tilde{x}, x^{k+1}) \leq D_{KL}(\tilde{x}, x^k)$ if $\sum_i A_{i,j} = 1$ for all j. If this condition is not fulfilled, similarly, we can show that $D_{KL}(\tilde{x}', x^{k+1'}) \leq D_{KL}(\tilde{x}', x^{k'})$, where $\tilde{x}'_j = \sum_i A_{i,j}\tilde{x}_j$ and ${x^{k'}}_j = \sum_i A_{i,j} x_j^k$ for all j.

Theorem 3.3.3. *If $\sum_i A_{i,j} = 1$ for all j, $D_{KL}(\tilde{x}, x^k)$ is decreasing for the case without regularization.*

Proof. Define vectors f^j, g^j such that their components are

$$f_i^j = \frac{A_{i,j} y_i / (A\tilde{x} + b)_i}{(A^T(y/(A\tilde{x}+b)))_j}, \qquad g_i^j = \frac{A_{i,j} y_i / (Ax^k + b)_i}{(A^T(y/(Ax^k+b)))_j}, \qquad i = 1, \cdots n, \tag{3.18}$$

then we have $\sum_i f_i^j = \sum_i g_i^j = 1$ and

$$\begin{aligned}
0 &\leq \sum_j \tilde{x}_j D_{KL}(f^j, g^j) \\
&= \sum_j \tilde{x}_j \sum_i f_i^j \log \frac{f_i^j}{g_i^j} \\
&= \sum_j \tilde{x}_j \sum_i \frac{A_{i,j} y_i / (A\tilde{x}+b)_i}{(A^T(y/(A\tilde{x}+b)))_j} \log \frac{(Ax^k+b)_i (A^T(y/(Ax^k+b)))_j}{(A\tilde{x}+b)_i (A^T(y/(A\tilde{x}+b)))_j} \\
&= \sum_j \tilde{x}_j \sum_i \frac{A_{i,j} y_i / (A\tilde{x}+b)_i}{(A^T(y/(A\tilde{x}+b)))_j} \log \frac{(Ax^k+b)_i x_j^{k+1} \tilde{x}_j}{(A\tilde{x}+b)_i \tilde{x}_j x_j^k}.
\end{aligned}$$

Since

$$\tilde{x}_j = \frac{(A^T(y/(A\tilde{x}+b))_j)}{(A^T 1)_j}\tilde{x}_j,$$

we have

$$\frac{(A^T(y/(A\tilde{x}+b)))_j}{(A^T 1)_j} = 1.$$

It follows that

$$\begin{aligned}
0 &\leq \sum_j \tilde{x}_j \sum_i \frac{A_{i,j}y_i}{(A\tilde{x}+b)_i} \log \frac{(Ax^k+b)_i x_j^{k+1}}{(A\tilde{x}+b)_i x_j^k} \\
&= \sum_j \tilde{x}_j \sum_i \frac{A_{i,j}y_i}{(A\tilde{x}+b)_i} \left(\log \frac{(Ax^k+b)_i}{(A\tilde{x}+b)_i} + \log \frac{x_j^{k+1}}{x_j^k} \right) \\
&= \sum_j \tilde{x}_j \sum_i \frac{A_{i,j}y_i}{(A\tilde{x}+b)_i} \log \frac{(Ax^k+b)_i}{(A\tilde{x}+b)_i} + \sum_j \tilde{x}_j \log \frac{x_j^{k+1}}{x_j^k} \\
&= \sum_i \frac{(A\tilde{x})_i y_i}{(A\tilde{x}+b)_i} \log \frac{(Ax^k+b)_i}{(A\tilde{x}+b)_i} + \sum_j \tilde{x}_j \log \frac{x_j^{k+1}}{x_j^k} \\
&= D_{KL}(y, A\tilde{x}+b) - D_{KL}(y, Ax^k+b) + D_{KL}(\tilde{x}, x^k) - D_{KL}(\tilde{x}, x^{k+1}) \\
&\quad - \sum_i \frac{b_i y_i}{(A\tilde{x}+b)_i} \log \frac{(Ax^k+b)_i}{(A\tilde{x}+b)_i} - \sum_j \tilde{x}_j + \sum_j x_j^{k+1}.
\end{aligned}$$

Since $\sum_i y_i - \sum_j x_j^{k+1} = \sum_i (y_i - \sum_j z_{ij}^{k+1}) = \sum_i \frac{b_i y_i}{(Ax^k+b)_i}$, we have

$$\begin{aligned}
&- \sum_i \frac{b_i y_i}{(A\tilde{x}+b)_i} \log \frac{(Ax^k+b)_i}{(A\tilde{x}+b)_i} - \sum_j \tilde{x}_j + \sum_j x_j^{k+1} \\
=& - \sum_i \frac{b_i y_i}{(A\tilde{x}+b)_i} \log \frac{(Ax^k+b)_i}{(A\tilde{x}+b)_i} + \sum_i \frac{b_i y_i}{(A\tilde{x}+b)_i} - \sum_i \frac{b_i y_i}{(Ax^k+b)_i} \\
=& - D_{KL}(\frac{b_i y_i}{(A\tilde{x}+b)_i}, \frac{b_i y_i}{(Ax^k+b)_i}) \leq 0.
\end{aligned}$$

The decreasing of the objective function $D_{KL}(y, Ax^k+b)$ gives us $D_{KL}(y, A\tilde{x}+b) \leq D_{KL}(y, Ax^k+$

b) and it follows that

$$0 \leq D_{KL}(\tilde{x}, x^k) - D_{KL}(\tilde{x}, x^{k+1})$$

which is $D_{KL}(\tilde{x}, x^{k+1}) \leq D_{KL}(\tilde{x}, x^k)$. □

If $\sum_i A_{i,j} = 1$ is not satisfied, we have the same property for $\tilde{x}'$ and $x^{k'}$, which are just weighted vectors with the j^{th} weight being $\sum_i A_{i,j}$, from the same proof.

3.4 EM-Type Algorithms with Bregman Iteration

In the previous section, the EM-Type algorithms are presented to solve problem (3.9). However, the regularization may lead to reconstructed images suffering from contrast reduction [48]. Therefore, we suggest a contrast improvement in EM-Type algorithms by Bregman iteration, which is introduced in [49, 50, 51]. An iterative refinement is obtained from a sequence of modified EM-Type algorithms.

For the problem with Poisson noise, we start with the basic EM-Type algorithms, i.e., finding the minimum x^1 of (3.9). After that, variational problems with a modified regularization term

$$x^{k+1} = \underset{x}{\operatorname{argmin}} \ \beta(J(x) - \langle p^k, x\rangle) + \sum_i \left((Ax+b)_i - y_i \log(Ax+b)_i\right) \tag{3.19}$$

where $p^k \in \partial J(x^k)$, are solved sequentially. From the optimality of (3.19), we have the following formula for updating p^{k+1} from p^k and x^{k+1}:

$$p^{k+1} = p^k - \frac{1}{\beta} A^T \left(1 - \frac{y}{Ax^{k+1}+b}\right). \tag{3.20}$$

Therefore the EM-Type algorithms with Bregman iteration are as follows:

Algorithm 4 Proposed EM-Type algorithms with Bregman iteration.

Input: x^0, δ, ϵ

Initialization: $k = 1$, $p^0 = 0$

while k $\leq$ Num_outer & $D_{KL}(y, Ax^{k-1} + b) < \delta$ **do**

$x^{temp,0} = x^{k-1}$, $l = 0$,

while $l \leq$ Num_inner & $\|x^{temp,l} - x^{temp,l-1}\| \leq \epsilon$ **do**

$l = l + 1$,

$x^{temp,l-\frac{1}{2}} = EM(x^{temp,l-1})$ using (3.8),

$x^{temp,l} = \underset{x}{\operatorname{argmin}}\ E_1^P(x, x^{temp,l-\frac{1}{2}})$ with $J(x) - \langle p^{k-1}, x\rangle$

end while

$x^k = x^{temp,l}$

$p^k = p^{k-1} - \frac{1}{\beta}A^T\left(1 - \frac{y}{Ax^k + b}\right)$,

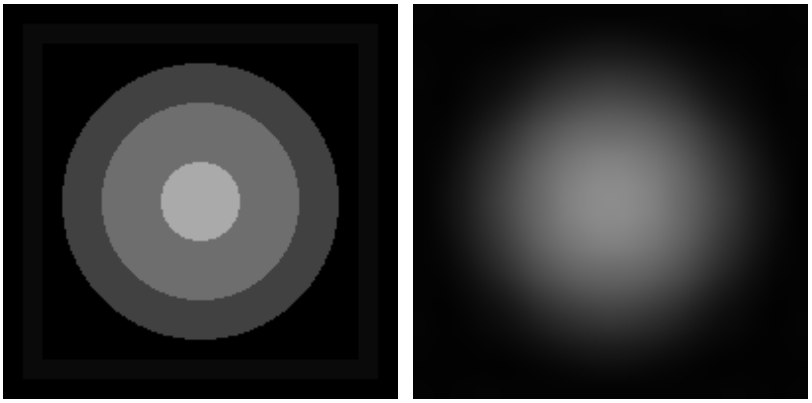
k=k+1

end while

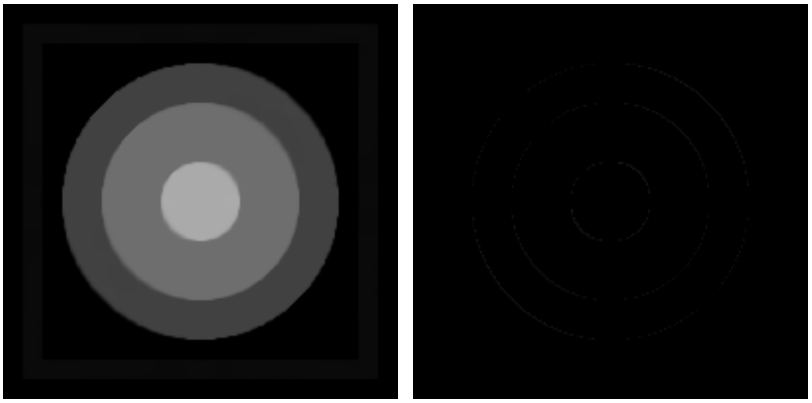
The initial guess x^0 can be any positive image, and $\delta = D_{KL}(y, Ax^* + b)$, where x^* is the ground truth, is assumed to be known, ϵ is the stopping criteria which is small. Num_inner and Num_outer are maximum numbers of inner iterations and outer iterations.

3.5 Numerical Experiments

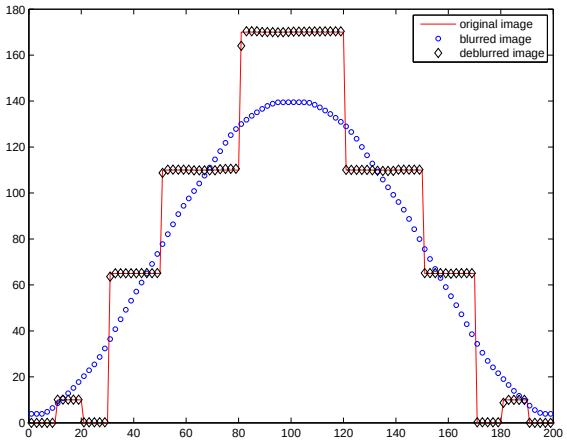
In this section, we will illustrate the proposed EM-Type algorithms for image reconstruction (more specifically, image deblurring). In the beginning, we present some deblurring results on a phantom with the proposed EM-TV algorithm, one example of EM-Type algorithms with total variation (TV) regularization, and the Bregman version of it. The phantom used in this section is a synthetic 200×200 phantom. It consists of circles with intensities 65, 110 and 170, enclosed by a square frame of intensity 10. For the experiment, we choose the backgroud $b = 20$. Firstly, we consider the case without noise. The blurred image is obtained from the original image using a Gaussian blur kernel K with standard deviation $\sigma = 100$. The result is shown in Figure 3.1. The root mean square error (RMSE) is 2.5629 and the KL distance is 0.0080.



(a) (b)



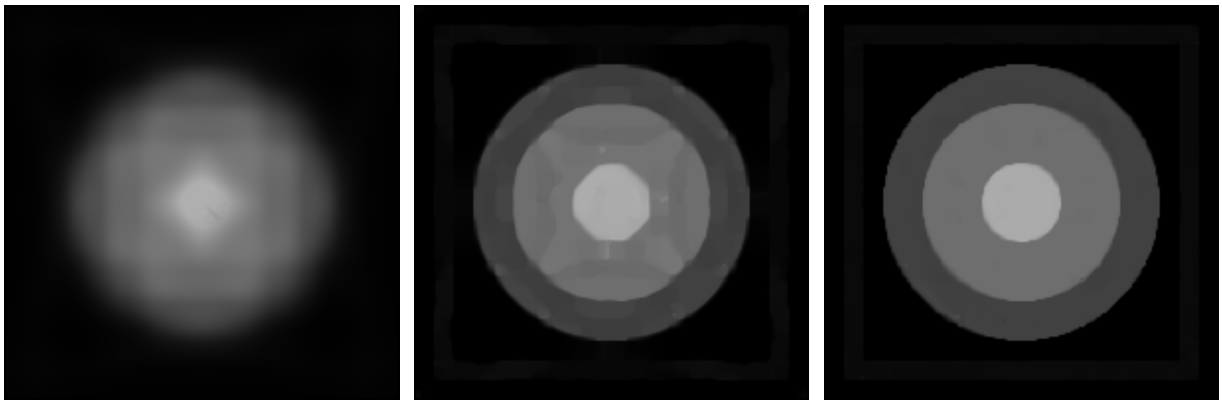
(c) (d)



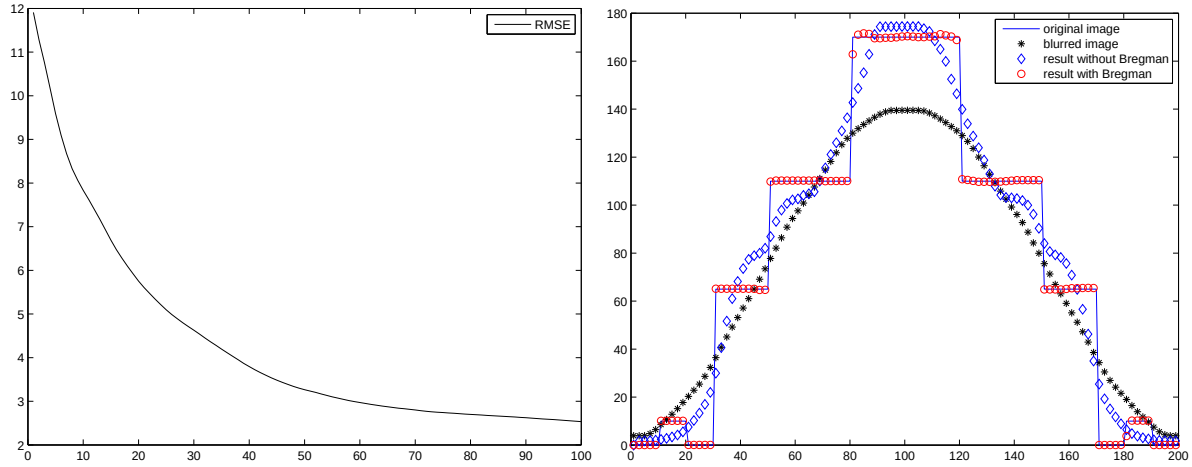
(e)

Figure 3.1: (a) The orginal image u_*. (b) Blurred image $K * u_*$ using a Gaussian blur kernel K. (c) The deblurred image using the proposed EM-TV with Bregman iteration. (d) The difference between the deblurred image and the original image. (e) The lineouts of original image, blurred image and deblurred image in the middle row. Some parameters chosen are $\beta = 5$, Num_inner $= 1$ and Num_outer $= 10000$.

To illustrate the advantage of Bregman iterations, we show the comparison results of EM-TV with different numbers of Bregman iterations in Figure 3.2. The RMSE for 3.2(a), 3.2(b) and 3.2(c) are 11.9039, 5.0944 and 2.5339, respectively. The corresponding KL distances are 93.0227, 0.8607 and 0.0172, respectively.



(a) (b) (c)



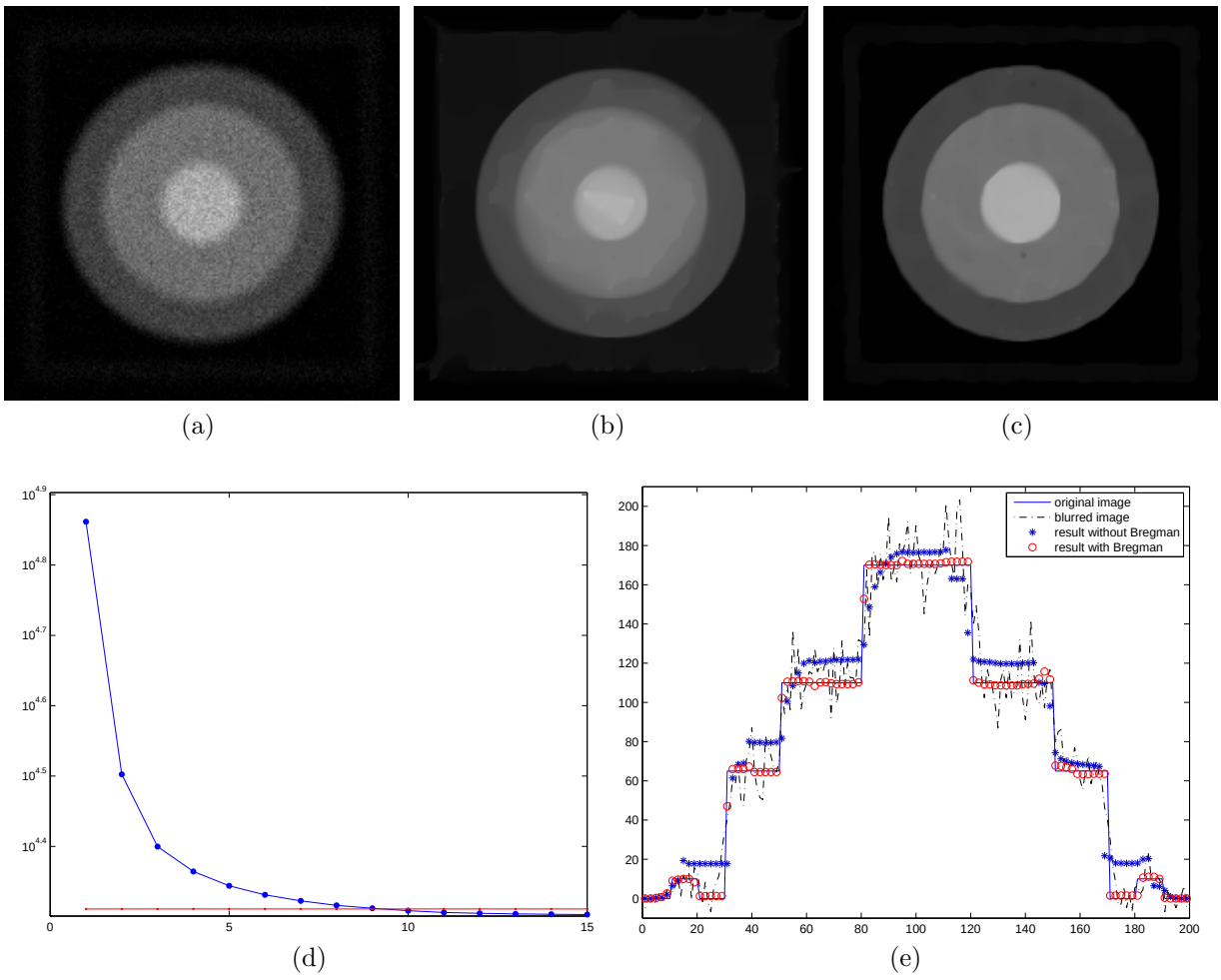
(d) (e)

Figure 3.2: (a) The result without Bregman iteration. (b) The result with 25 Bregman iterations. (c) The result with 100 Bregman iterations. (d) The plot of RMSE versus Bregman iterations. (e) The lineouts of original image, blurred image, the results with and without Bregman iterations. Some parameters chosen are $\beta = 0.001$, Num_inner $= 100$ and Num_outer $= 100$.

How to choose a good parameter β is important for algorithms without Bregman iteration because it balances the regularization and data fidelity, while it is not sensitive for algorithms with Bregman iteration. For this experiment, though β is not chosen to be op-

timal, the results of Bregman iteration show that we can still obtain a good result after several iterations. From the lineouts we can see that the result with Bregman iteration fits the original image very well.

EM-TV with Bregman iteration can provide us with very good result if there is no noise in the blurred images. However, the noise is unavoidable in applications. The next experiment is to illustrate the EM-TV algorithm in the noisy case. The RMSE for 3.3(b) and 3.3(c) are 12.9551 and 4.1176, respectively.



(a) (b) (c)

(d) (e)

Figure 3.3: (a) The noisy blurred image. (b) The result without Bregman iteration. (c) The result with 9 Bregman iterations. (d) The plot of KL distances versus Bregman iterations. (e) The lineouts of original image, blurred image, the results with and without Bregman iterations. Some parameters chosen are $\beta = 1$, Num_inner $= 200$ and Num_outer $= 15$.

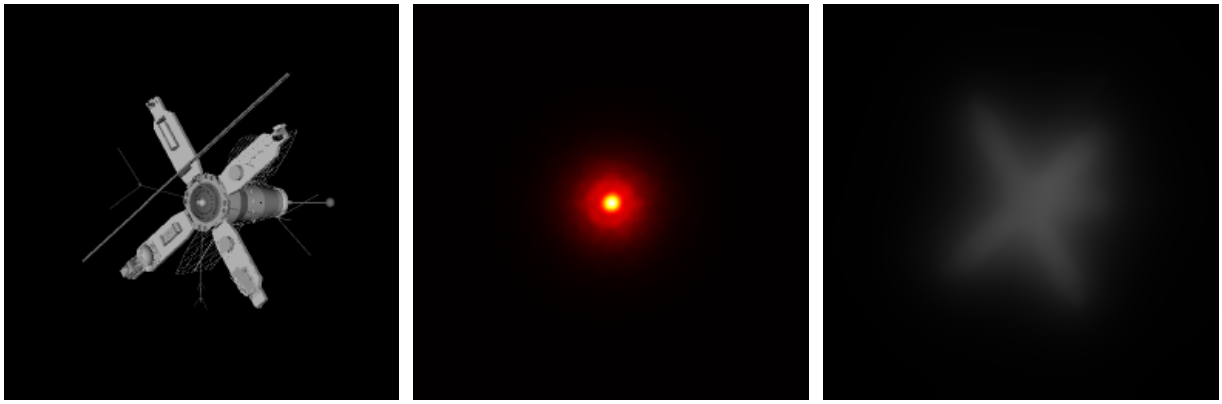
The results show that even with noise, EM-TV with Bregman iteration gives better result

than EM-TV without Bregman iteration.

Next, the EM-TV algorithm is used to perform deconvolution on an image of a satellite (Figure 3.4(a)), and the point spread function (PSF) is shown in Figure 3.4(b). In order to make the algorithm fast, we choose the initial guess x^0 to be the result from solving $Ax = y - b$ using conjugate gradient (CG). The negative values are changed into zero before applying the EM-TV algorithm. The corresponding RMSE for x^0 and the result are 13.6379 and 11.8127, respectively. By using the EM-TV with Bregman iteration, we get a better image with sharp edges and artifacts are removed.

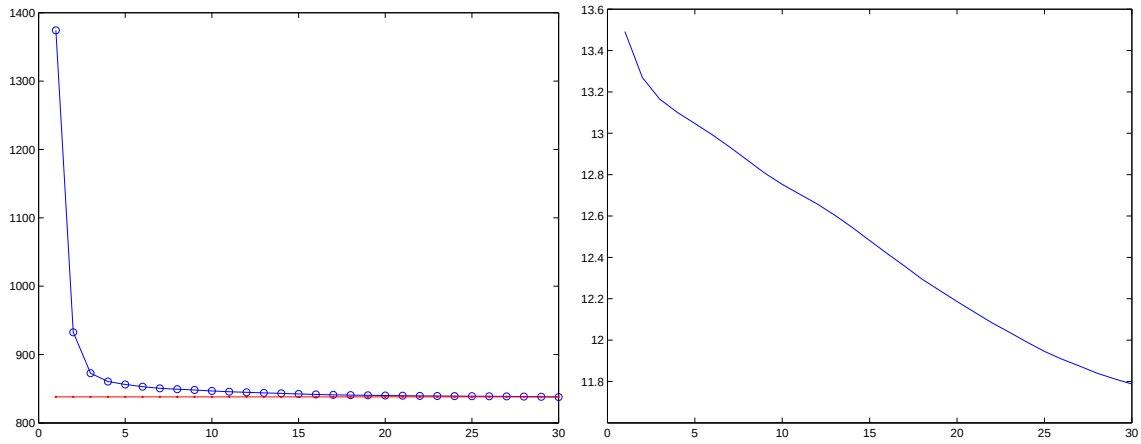
The same EM-TV algorithm is also tested on an image of text (Figure 3.5(a)) and the point spread function (PSF) is shown in Figure 3.5(b). In order to make the algorithm fast, we choose the initial guess x^0 to be the result from solving $Ax = y - b$ using Hybrid Bidiagonalization Regularization (HyBR) [52]. The negative values are changed into zero before applying the EM-TV algorithm. The corresponding RMSE for x^0 and the result are 45.8918 and 37.8574, respectively. By using the EM-TV with Bregman iteration, we get a better image with sharp edges and artifacts are removed.

The convergence analysis of EM-Type algorithms is for the case when $J(x)$ is convex. When $J(x)$ is not convex, we still have the same algorithm, and from the equivalence with alternating minimization method, the algorithm will converge to a local minimum of the function. For the last experiment (Figure 3.6), we try to separate the spare objects in lensfree fluorescent imaging [53] using EM-Type algorithm with a non-convex $J(x)$. The result of EM (or Richardson-Lucy) method will tend to be sparse, because the l_1 norm is almost fixed for all the iterations (when $b = 0$, the l_1 norm is fixed), but the EM method cannot separate the particles when they are close to each other ($13\mu m$ and $10\mu m$ in this experiment). Therefore, we can choose $J(x) = \sum_j |x_j|^p$ for $p \in (0, 1)$, and these two particles can be separated even when the distance is very small). For the numerical experiment, we choose the same numbers of iterations for EM-l_p and EM method, and the results show that with $p < 1$, we can obtain better results.



(a) (b) (c)

(d) (e) (f)



(g) (h)

Figure 3.4: (a) The original image. (b) The PSF image. (c) The blurred image. (d) The noisy blurred image. (e) Initial guess from CG. (f) The result of EM-Type algorithm with Bregman iterations. (g) The plot of KL versus Bregman iterations. (h) The RMSE versus Bregman iterations. Some parameters chosen are $\beta = 1$, Num_inner = 200 and Num_outer = 30.

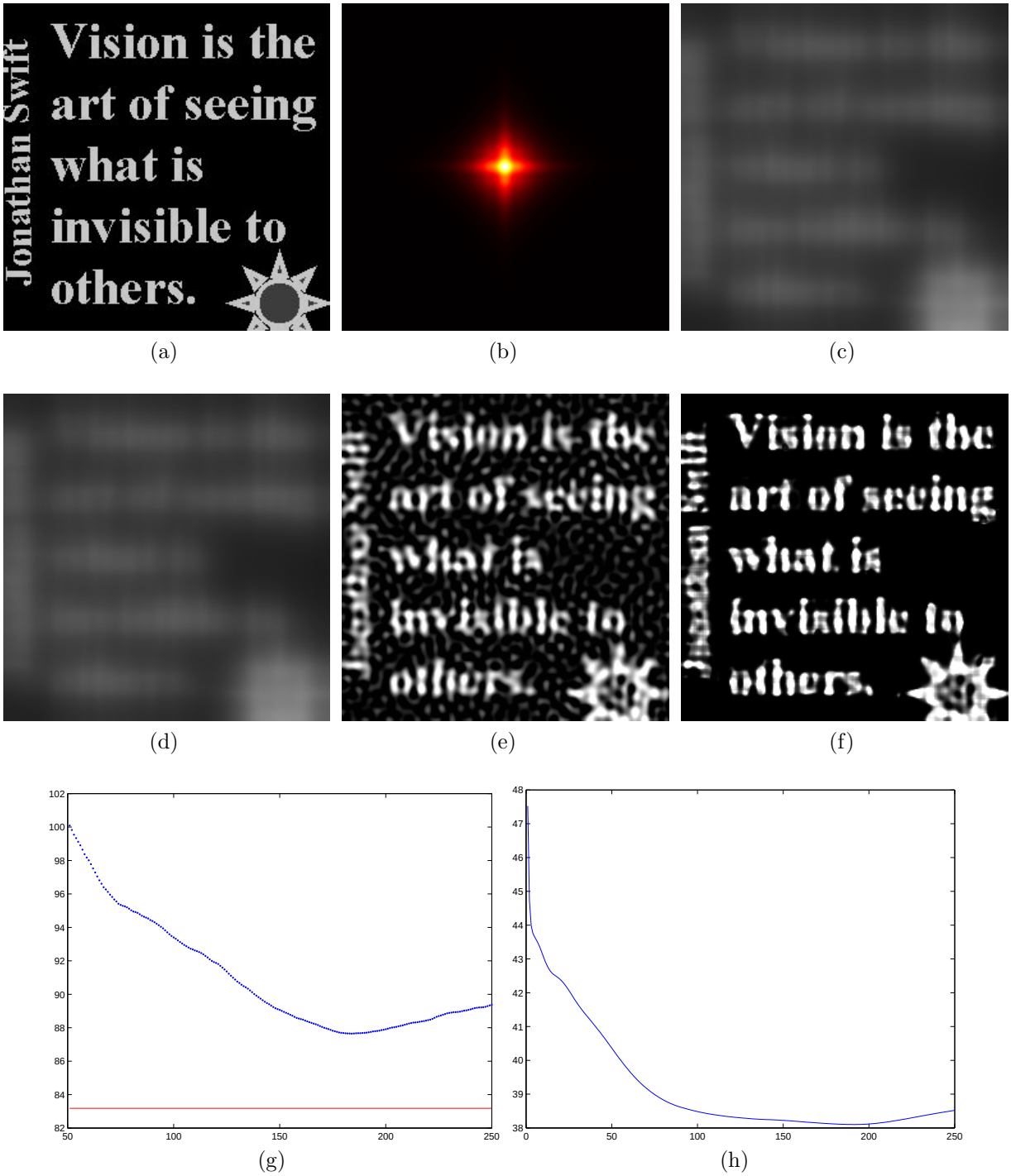
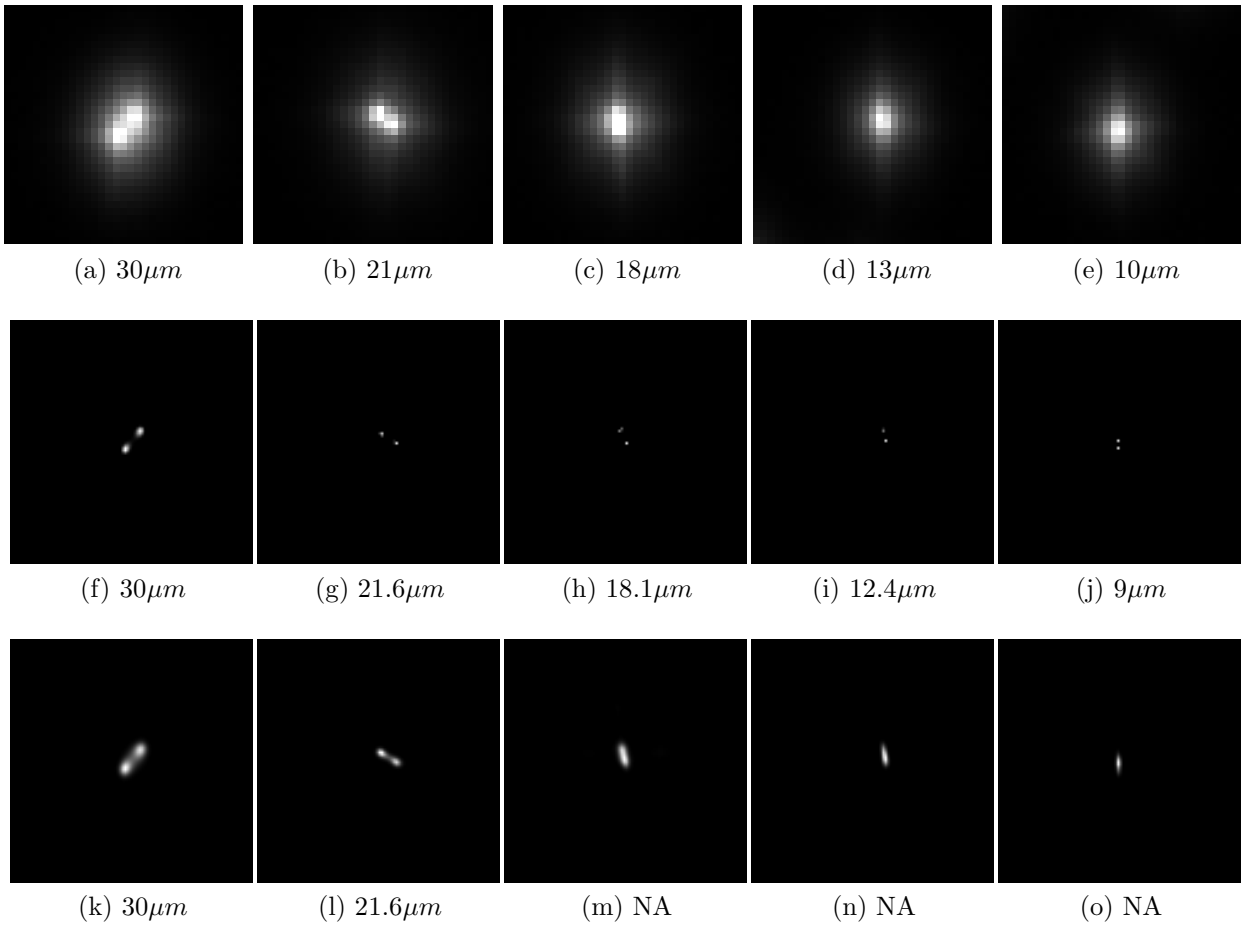


Figure 3.5: (a) The original image. (b) The PSF image. (c) The blurred image. (d) The noisy blurred image. (e) Initial guess from HyBR. (f) The result of EM-Type algorithm with Bregman iterations. (g) The plot of KL versus Bregman iterations. (h) The RMSE versus Bregman iterations. Some parameters chosen are $\beta = 10^{-5}$, Num_inner = 10 and Num_outer = 250.



(a) $30\mu m$ (b) $21\mu m$ (c) $18\mu m$ (d) $13\mu m$ (e) $10\mu m$

(f) $30\mu m$ (g) $21.6\mu m$ (h) $18.1\mu m$ (i) $12.4\mu m$ (j) $9\mu m$

(k) $30\mu m$ (l) $21.6\mu m$ (m) NA (n) NA (o) NA

Figure 3.6: Top row shows raw lensfree fluorescent images of different pairs of particles. The distances betweens thes two particles are $30\mu m$, $21\mu m$, $18\mu m$, $13\mu m$ and $9\mu m$, from left to right. Middle row shows the results of EM-Type algorithm with $p = 0.5$. Bottom row shows the results for EM (or Richardson-Lucy) method.

3.6 Conclusion

In this chapter, we proposed general robust EM-Type algorithms for image reconstruction with background emission when the measured data is corrupted by Poisson noise: iteratively performing EM and regularization in the image domain. The convergence of these algorithms is proved in several ways. For the case without regularization, the KL distance to the limit of the sequence of iterations is decreasing. The problem with regularization will lead to contrast reduction in the reconstructed images. Therefore, in order to improve the contrast, we suggested EM-Type algorithms with Bregman iteration by applying a sequence of modified

EM-Type algorithms. We have tested EM-Type algorithms with different $J(x)$. With TV regularization, this EM-TV algorithm can provide images with preserved edges and artifacts are removed. When l_p regularization is used, this EM-l_p algorithm can be used to separate sparse particles even when the distance is small, in which case the EM method can not do a good job.

Part II

Adaptive Outlier Pursuit

CHAPTER 4

Adaptive Outlier Pursuit for Robust 1-Bit Compressive Sensing

4.1 Introduction

The theory of compressive sensing (CS) enables reconstruction of sparse or compressible signals from a small number of linear measurements relative to the dimension of the signal space [54, 55, 56, 57, 58]. In this setting, we have

$$y = \Phi x, \tag{4.1}$$

where $x \in \mathbf{R}^N$ is the signal, $\Phi \in \mathbf{R}^{M \times N}$ with $M < N$ is an underdetermined measurement system, and $y \in \mathbf{R}^M$ is the set of linear measurements. It was demonstrated that K-sparse signals, i.e., $x \in \Sigma_K$ where $\Sigma_K := \{x \in \mathbf{R}^N : \|x\|_0 := |\text{supp}(x)| \leq K\}$, can be reconstructed exactly if Φ satisfies the restricted isometry property (RIP) [59]. It was also shown that random matrices will satisfy the RIP with high probability if the entries are chosen according to independent and identically distributed (i.i.d.) Gaussian distribution.

Classic compressive sensing assumes that the measurements are real valued and have infinite bit precision. However, in practice, CS measurements must be quantized, i.e., each measurement has to be mapped from a real value (over a potentially infinite range) to a discrete value over some finite range, which will induce error in the measurements. The quantization of CS measurements has been studied recently and several new algorithms were proposed [60, 61, 62, 63, 64, 65].

Furthermore, for some real world problems, severe quantization may be inherent or pre-

ferred. For example, in analog-to-digital conversion (ADC), the acquisition of 1-bit measurements of an analog signal only requires a comparator to zero, which is an inexpensive and fast piece of hardware that is robust to amplification of the signal and other errors, as long as they preserve the signs of the measurements, see [5, 66]. In this paper, we will focus on the CS problem when a 1-bit quantizer is used.

The 1-bit compressive sensing framework proposed in [5] is as follows. Measurements of a signal $x \in \mathbf{R}^N$ are computed via

$$y = A(x) := \text{sign}(\Phi x). \tag{4.2}$$

Therefore, the measurement operator $A(\cdot)$ is a mapping from $\mathbf{R}^N$ to the Boolean cube [1] $\mathcal{B}^M := \{-1, 1\}^M$. We have to recover a signal $x \in \sum_K^* := \{x \in S^{N-1} : \|x\|_0 \leq K\}$ where $S^{N-1} := \{x \in \mathbf{R}^N : \|x\|_2 = 1\}$ is the unit hyper-sphere of dimension N. Since the scale of the signal is lost during the quantization process, we can restrict the sparse signals to be on the unit hyper-sphere. Jacques et al. provided two flavors of results for the 1-bit CS framework [67]: 1) a lower bound is provided on the best achievable performance of this 1-bit CS framework, and if the elements of Φ are drawn randomly from i.i.d. Gaussian distribution or its rows are drawn uniformly from the unit sphere, then the solution will have bounded error on the order of the optimal lower bound; and 2) a condition on the mapping A, binary ϵ-stable embedding (BϵSE), that enables stable reconstruction is given to characterize the reconstruction performance even when some of the measurement signs have changed (e.g., due to noise in the measurements).

Since this problem was introduced and studied by Boufounos and Baraniuk in 2008 [5], it has been studied by many people and several algorithms have been developed [5, 67, 68, 69, 70, 71, 72]. Binary iterative hard thresholding (BIHT) [67] is shown to perform better than other algorithms such as matching sign pursuit (MSP) [68] and restricted-step shrinkage (RSS) [70] in reconstruction error as well as consistency, see [67] for more details. The experiment in [67] shows that the one-sided ℓ_1 objective (BIHT) performs better when there

[1] Generally, the M-dimensional Boolean cube is defined as $\{0, 1\}^M$. Without loss of generality, we use $\{-1, 1\}^M$ instead.

are only a few errors, and the one-sided ℓ_2 objective (BIHT-ℓ_2) performs better when there are significantly more errors, which implies that BIHT-ℓ_2 is useful when the measurements contain significant noise that might cause a large number of sign flips.

In practice, there will always be noise in the measurements during acquisition and transmission, therefore, a robust algorithm for 1-bit compressive sensing when the measurements flip their signs is strongly needed. One possible way to build this robust algorithm is to introduce an outlier detection technique.

There are many applications where the accurate detection of outliers is needed. For example, when an image is corrupted by random-valued impulse noise, the corrupted pixels are useless in image denoising. There are some methods (e.g., adaptive center-weighted median filter (ACWMF) [4]) for detecting the damaged pixels. But these methods will miss quite a lot of real noise and false-hit some noise-free pixels when the noise level is high. In [73], we proposed a method to adaptively detect the noisy pixels and restore the image with ℓ_0 minimization. Instead of detecting the damaged pixels before recovering the image, we iteratively restore the image and detect the damaged pixels. This idea works really well for impulse noise removal. In this 1-bit compressive sensing framework, when there is a sign flip in one measurement, this measurement will worsen the reconstruction performance. If we can detect all the measurements with sign flips, then we can change the signs for these measurements and improve the reconstruction performance a lot. However, it is much more difficult than detecting impulse noise and there is no method for detecting sign flips, but we can still utilize the idea in [73] to adaptively find the sign flips. In this chapter, we will introduce a method for robust 1-bit compressive sensing which can detect the sign flips and reconstruct the signals with very high accuracy even when there are a large number of sign flips.

This chapter is organized as follows. We will introduce several algorithms for reconstructing the signal and detecting the sign flips in section 4.2. Section 4.3 studies the case when the noise information is not given. The performance of these algorithms is illustrated in section 4.4 with comparison to BIHT and BIHT-ℓ_2. We will end this work by a short conclusion.

4.2 Robust 1-bit Compressive Sensing using Adaptive Outlier Pursuit

Binary iterative hard thresholding (BIHT or BIHT-ℓ_2) in [67] is the algorithm for solving

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \sum_{i=1}^{M} \phi(y_i, (\Phi x)_i) \\ \text{subject to:} \quad & \|x\|_2 = 1, \quad \|x\|_0 \leq K, \end{aligned} \tag{4.3}$$

where ϕ is the one-sided ℓ_1 (or ℓ_2) objective:

$$\phi(x, y) = \begin{cases} 0, & \text{if } x \cdot y > 0, \\ |x \cdot y| \text{ (or } |x \cdot y|^2/2), & \text{otherwise.} \end{cases} \tag{4.4}$$

The high performance of BIHT is demonstrated when all the measurements are noise-free. However when there are a lot of sign flips, the performance of BIHT and BIHT-ℓ_2 is worsened by the noisy measurements. There is no method to detect the sign flips in the measurements, but adaptively finding the sign flips and reconstructing the signals can be combined together as in [73] to obtain better performance.

Let us assume firstly that the noise level (the ratio of the number of sign flips over the number of measurements for 1-bit compressive sensing) is provided. Based on this information, we can choose a proper integer L such that at most L elements of the total measurements are wrongly detected (having sign flips). For measurements $y \in \{-1, 1\}^M$, $\Lambda \in \mathbf{R}^M$ is a binary vector denoting the "correct" data:

$$\Lambda_i = \begin{cases} 1, & \text{if } y_i \text{ is "correct"}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.5}$$

According to the assumption, we have $\sum_{i=1}^{M}(1 - \Lambda_i) \leq L$.

Introducing Λ into the old problem solved by BIHT, we have the following new problem

with unknown variables x and Λ:

$$\begin{aligned} \underset{x,\Lambda}{\text{minimize}} \quad & \sum_{i=1}^{M} \Lambda_i \phi(y_i, (\Phi x)_i) \\ \text{s.t.} \quad & \sum_{i=1}^{M} (1 - \Lambda_i) \leq L, \\ & \Lambda_i \in \{0, 1\} \quad i = 1, 2, \cdots, M, \\ & \|x\|_2 = 1, \quad \|x\|_0 \leq K. \end{aligned} \tag{4.6}$$

The above model can also be interpreted in the following way. Let us consider the noisy measurements y as the signs of Φx with additive unknown noise n, i.e., $y = \text{sign}(\Phi x + n)$. Though the binary measurement is robust to noise as long as the sign does not change, there exist some n_i's such that the corresponding measurements change. In our problem, only a few measurements are corrupted, and only these corresponding n_i's are important. Therefore, n can be considered as sparse noise with nonzero entries at these locations, and we have to recover the signal x from sparsely corrupted measurements [74, 75], even when the measurements are acquired by taking the signs of $\Phi x + n$. This equivalent problem is

$$\begin{aligned} \underset{x,n}{\text{minimize}} \quad & \sum_{i=1}^{M} \phi(y_i, (\Phi x)_i + n_i) \\ \text{s.t.} \quad & \|n\|_0 \leq L, \\ & \|x\|_2 = 1, \quad \|x\|_0 \leq K. \end{aligned} \tag{4.7}$$

The equivalence is described in the appendix at the end of this chapter.

The problem defined in (4.6) is non-convex and has both continuous and discrete variables. It is difficult to find (x, Λ) together, thus we use an alternating minimization method, which separates the energy minimization over x and Λ into two steps:

- Fix Λ and solve for x:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \sum_{i=1}^{M} \Lambda_i \phi(y_i, (\Phi x)_i) \\ \text{s.t.} \quad & \|x\|_2 = 1, \quad \|x\|_0 \leq K. \end{aligned} \tag{4.8}$$

This is the same as (4.3) with revised Φ and y. We only need to use the ith rows of Φ

and y where $\Lambda_i = 1$.

- Fix x and update Λ:

$$\begin{aligned} \underset{\Lambda}{\text{minimize}} \quad & \sum_{i=1}^{M} \Lambda_i \phi(y_i, (\Phi x)_i) \\ \text{s.t.} \quad & \sum_{i=1}^{M} (1 - \Lambda_i) \leq L, \\ & \Lambda_i \in \{0, 1\} \quad i = 1, 2, \cdots, M. \end{aligned} \tag{4.9}$$

This problem is to choose $M-L$ elements with least sum from M elements $\{\phi(y_i, (\Phi x)_i)\}_{i=1}^{M}$. Given an x estimated from (4.8), we can update Λ in one step:

$$\Lambda_i = \begin{cases} 0, & \text{if } \phi(y_i, (\Phi x)_i) \geq \tau, \\ 1, & \text{otherwise,} \end{cases} \tag{4.10}$$

where τ is the L^{th} largest term of $\{\phi(y_i, (\Phi x)_i)\}_{i=1}^{M}$. If the L^{th} and $(L+1)^{th}$ largest terms are equal, then we can choose any Λ such that $\sum_{i=1}^{M} \Lambda_i = M - L$ and

$$\min_{i, \Lambda_i = 0} \phi(y_i, (\Phi x)_i) \geq \max_{i, \Lambda_i = 1} \phi(y_i, (\Phi x)_i).$$

Since for each step, the updated Λ identifies the outliers, this method is named as adaptive outlier pursuit (AOP). When $L = 0$, this is exactly the BIHT proposed in [67]. Our algorithm is as follows:

Algorithm 5 AOP

Input: $\Phi \in \mathbf{R}^{M\times N}$, $y \in \{-1,1\}^M$, $K > 0$, $L \geq 0$, $\alpha > 0$, Miter > 0
Initialization: $x^0 = \Phi^T y/\|\Phi^T y\|$, $k = 0$, $\Lambda = \mathbf{1} \in \mathbf{R}^M$, Loc $= 1 : M$, tol $=$ inf, TOL $=$ inf.
while $k \leq$ Miter and $L \leq$ tol **do**
  Compute $\beta^{k+1} = x^k + \alpha\Phi(\text{Loc},:)^T(y(\text{Loc}) - \text{sign}(\Phi(\text{Loc},:)x^k))$.
  Update $x^{k+1} = \eta_K(\beta^{k+1})$,
  Set tol $= \|y - A(x^{k+1})\|_0$.
  if tol $\leq$TOL **then**,
    Compute Λ with (4.10).
    Update Loc to be the location of 1-entries of Λ.
    Set TOL $=$ tol.
  end if
  $k = k + 1$.
end while
return $x^k/\|x^k\|$.

$\eta_K(v)$ computes the best K-term approximation of v by thresholding. Since $y_i \in \{-1,\ 1\}$, once we find the locations of the errors, instead of deleting these data, we can also "correct" them by flipping their signs. Hence x can also be updated with Φ and these new measurements. This algorithm with changing signs is called AOP with flips.

Remark: Similar to BIHT-ℓ_2, we can also choose the one-sided ℓ_2 objective instead of the ℓ_1 objective and obtain two other algorithms.

4.3 The case with L unknown

In the previous section, we assume that L, the number of corrupted measurements, is known in advance. However in real world applications there are cases when no pre-knowledge about the noise is given. If L is chosen smaller or larger than the true value, the performance of these algorithms will get worse. As shown numerically in section 4.4, when L is less than the true value, even if the L detections are completely correct, some sign flips still remain in the measurements. On the other hand, some correct measurements will be lost if L is too large, and the problem will have more solutions if the number of total measurements is not large enough, which will affect the accuracy of the algorithm. Therefore, in this scenario we

have to apply an L detection skill to find an L which is not far from the true value.

When no noise information is given, the following procedure can be applied to predict L. The first-phase preparation is to do extensive experiments on simulated data with known L and record the Hamming distances between $A(x)$ and noisy y of BIHT-ℓ_2 and AOP. Here we can simply use the results in our first experiment in section 4.4. The average of the results describes nicely the behavior of these two algorithms at different noise levels. Hence a formula can be derived to predict the Hamming distance of AOP based on the results obtained by BIHT-ℓ_2. This could be a fair initial guess for the noise level, and we can derive an L based on the result, labeled as L_0. Then we calculate $L_t = \|A(x) - y\|_0$ using the result x gained by AOP with L_0 as the input for L. If L_t is greater than L_0, which means that L_0 is too small while L_t is too large, we set L_t as the upper bound $L_{\max}$ and L_0 as the lower bound $L_{\min}$. Otherwise, if L_t is smaller than or equal to L_0, which means L_0 may be too large, we use μL_t $(0 < \mu < 1)$ as the new L_0 to look for new L_t. We will keep doing this until L_t is greater than L_0. Then the previous L_0 is defined as the upper bound $L_{\max}$ and the new L_0 is defined as the lower bound $L_{\min}$. This is just one method for finding lower and upper bounds for L, and there are certainly other possible ways to decide the bounds. Then we use the bisection method to find a better L. The mean of $L_{\max}$ and $L_{\min}$ (L_{mean}) is then used as input to derive L_t with AOP. If L_t is greater than L_{mean}, we update $L_{\min}$ with L_{mean}. Otherwise, L_{mean} is set as $L_{\max}$. This bisection method is applied to update these two bounds until $L_{\max} - L_{\min} \leq 1$. The final $L_{\min}$ is our input L.

4.4 Numerical Results

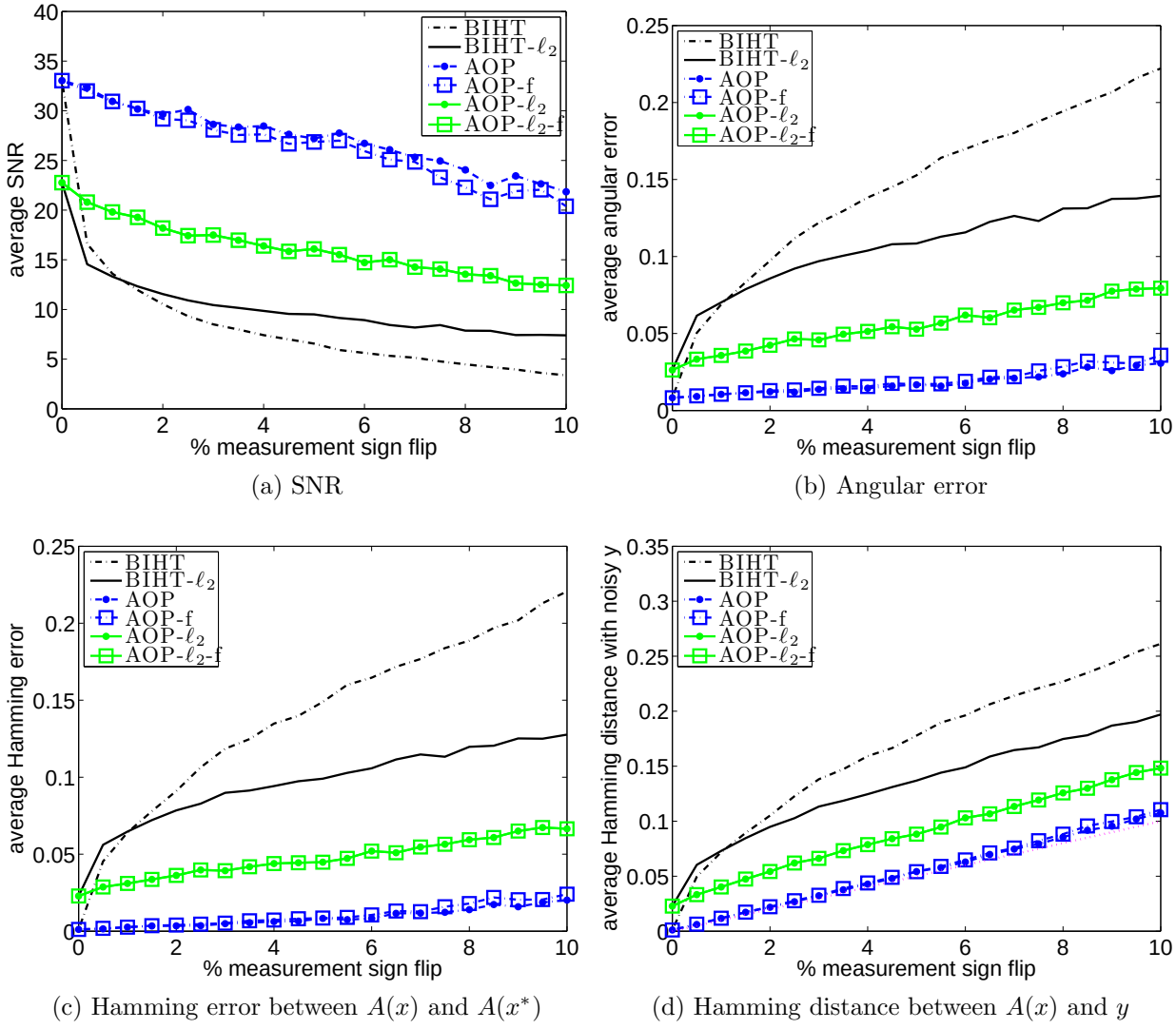
In this section we use several numerical experiments to demonstrate the effectiveness of AOP algorithms. Here AOP is implemented in the following four ways: 1) AOP with one-sided ℓ_1 objective (AOP); 2) AOP with flips and one-sided ℓ_1 objective (AOP-f); 3) AOP with one-sided ℓ_2 objective (AOP-ℓ_2); and 4) AOP with flips and one-sided ℓ_2 objective (AOP-ℓ_2-f). The four algorithms, together with BIHT and BIHT-ℓ_2, are studied and compared in the following experiments.

The setup for our experiments is as follows. We first generate a matrix $\Phi \in \mathbf{R}^{M\times N}$ whose elements follow i.i.d. Gaussian distribution. Then we generate the original K-sparse signal $x^* \in \mathbf{R}^N$. Its non-zero entries are drawn from standard Gaussian distribution and then normalized to have norm 1. $y^* \in \{-1, 1\}^M$ is computed by $A(x^*)$.

4.4.1 Noise levels test

In our first experiment, we set $M = N = 1000$, $K = 10$, and examine the performance of these algorithms on data with different noise levels. Here in each test, we choose a few measurements at random and flip their signs. The noise level is between 0% and 10% and we assume it is known in advance. For each level, we perform 100 trials and record the average signal-to-noise ratio (SNR), average reconstruction angular error for each reconstructed signal x with respect to x^*, average Hamming error between $A(x)$ and $A(x^*)$, and average Hamming distance between $A(x)$ and the noisy measurements y. Here SNR is denoted by $10\log_{10}(\|x\|^2/\|x-x^*\|^2)$, angular error is defined as $\arccos\langle x, x^*\rangle/\pi$, Hamming error stands for $\|A(x) - A(x^*)\|_0/M$ and the Hamming distance between $A(x)$ and y, defined as $\|A(x)-y\|_0/M$, is used to measure the difference between $A(x)$ and the noisy measurements y. The results are depicted in Figure 4.1. The plots demonstrate that in these comparisons four AOP algorithms outperform BIHT and BIHT-ℓ_2 for all noise levels, significantly so when more than 2% of the measurements are corrupted. Compared with BIHT, BIHT-ℓ_2 tends to give worse results when there are only a few sign flips in y and better results if we have high noise level. This has been shown and studied in [67]. Of all the AOP series, AOP and AOP-f give better results compared with AOP-ℓ_2 and AOP-ℓ_2-f. We can also see that there is a lot of overlap between the results obtained by AOP and the ones acquired by AOP with flips, especially when one-sided ℓ_2 objective is used, the results are almost the same. Figure 4.1(d) compares the average Hamming distances between $A(x)$ and the noisy measurements y for all algorithms. If the sign flips can be found correctly, then the Hamming distance between $A(x)$ and y should be equal to the noise level. The result shows that average Hamming distances for AOP and AOP-f are slightly above the noise levels, which means that AOP with one-sided ℓ_1 objective performs better in consistency than other algorithms in noisy

case.



(a) SNR

(b) Angular error

(c) Hamming error between $A(x)$ and $A(x^*)$

(d) Hamming distance between $A(x)$ and y

Figure 4.1: Algorithm comparison on corrupted data with different noise levels. (a) average SNR versus noise level, (b) average angular error versus noise level, (c) average Hamming error between $A(x)$ and $A(x^*)$ versus noise level, (d) average Hamming distance between $A(x)$ and noisy measurements y versus noise level. AOP proves to be more robust to measurement sign flips compared with BIHT.

In order to show that our algorithms can find the positions of sign flips with high accuracy, we measure the probabilities of correct detections of sign flips in the noisy measurements for different noise levels from 0.5% to 10% in Figure 4.2 ($M = N = 1000$, $K = 10$). The exact number of sign flips is used as L in the algorithms and we compare the exact locations of sign flips in measurements y with those detected from the algorithms for all 100 trials, then

the average probabilities of correct detections are shown for different algorithms at different noise levels. From this figure, we can see that all four algorithms have high accuracy in detecting the sign flips. When the noise level is low ($\leq$4%), the accuracy of AOP and AOP-f can be as high as 95%, even when the noise level is high (e.g., 10%), the accuracy of AOP and AOP-f is still above 90%. Comparing to algorithms with one-sided ℓ_1 objective, algorithms with one-sided ℓ_2 objective have lower accuracy. The accuracy for AOP-ℓ_2 and AOP-ℓ_2-f is around 80%.

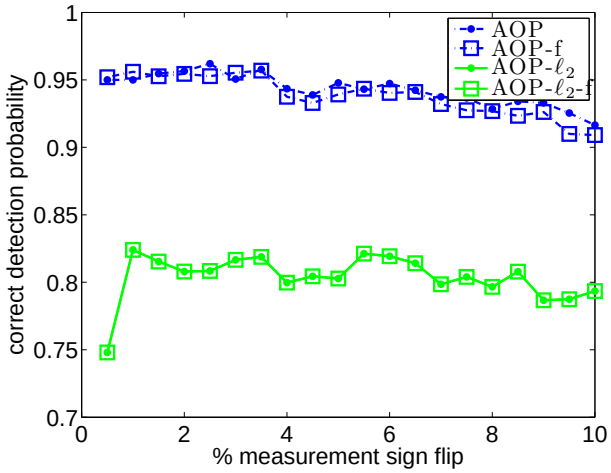
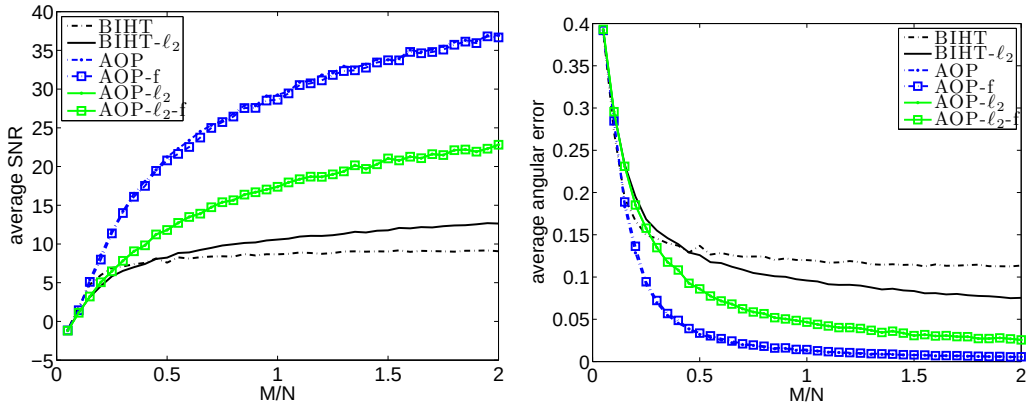


Figure 4.2: The probabilities of correct detections of sign flips for different noise levels ranging from 0.5% to 10%. AOP and AOP-f have very high accuracy (great than 90%) in detecting the sign flips, while AOP-ℓ_2 and AOP-ℓ_2-f have relatively lower accuracy (around 80%).

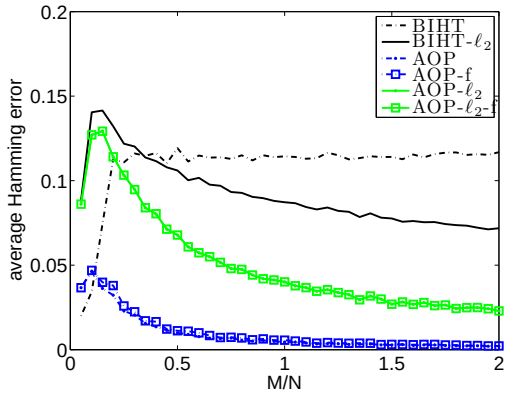
4.4.2 M/N test

In the second experiment, $N = 1000$, $K = 10$ and the noise level 3% are fixed, and we change M/N values within the range $(0, 2]$. 40 different M/N values are considered and we perform 300 tests for each value. The results are displayed in five different ways: the average SNR, average angular error, average Hamming error between $A(x)$ and $A(x^*)$, average Hamming distance between $A(x)$ and y and average percentage of coefficient misses. Here misses stand for the coefficients where $x_i^* \neq 0$ while $x_i = 0$. According to Figure 4.3, although all the algorithms show the same trend as M/N increases, AOP and AOP-f always obtain a much smaller angular error (higher SNR) than BIHT and BIHT-ℓ_2. There are also fewer coefficient misses in the results acquired by AOP series. Furthermore, we see that even when 3% of

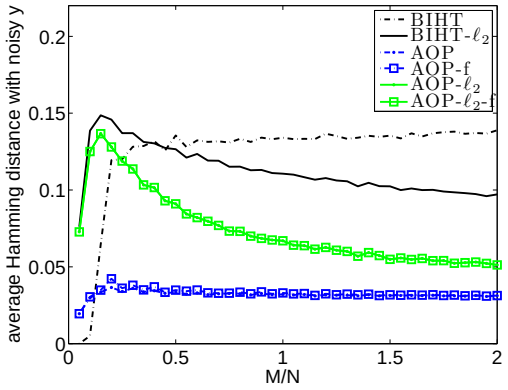


(a) SNR

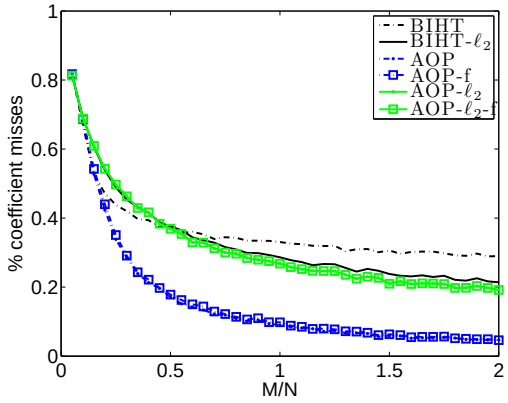
(b) Angular error



(c) Hamming distance between $A(x)$ and $A(x^*)$

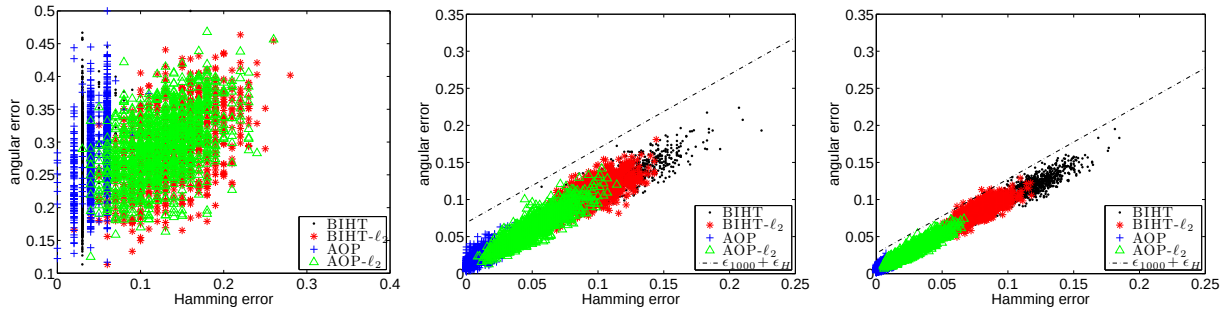


(d) Hamming error between $A(x)$ and y



(e) % coefficient misses

Figure 4.3: Algorithm comparison on corrupted data with different M/N. (a) average SNR versus M/N, (b) average angular error versus M/N, (c) average Hamming error between $A(x)$ and $A(x^*)$ versus M/N, (d) average Hamming distance between $A(x)$ and y versus M/N, (e) average percentage of coefficient misses versus M/N. AOP yields a remarkable improvement in reducing the Hamming and angular error and achieving higher SNR.



(a) M=0.1N (b) M=0.7N (c) M=1.5N

Figure 4.4: Hamming error versus angular error with different M's. AOP gives the most consistent results for $M = 0.7N$ and $M = 1.5N$. In these two cases we can see a linear relationship $\epsilon_{\text{sim}} \approx C + \epsilon_H$ between the average angular error ϵ_{sim} and average Hamming error ϵ_H, where C is a constant. For really small M ($M = 0.1N$) BIHT returns almost the same results as AOP as AOP may fail to find the exact sign flips in the noisy measurements. The dashed line $\epsilon_{1000} + \epsilon_H$ is an upper bound for 1000 trials.

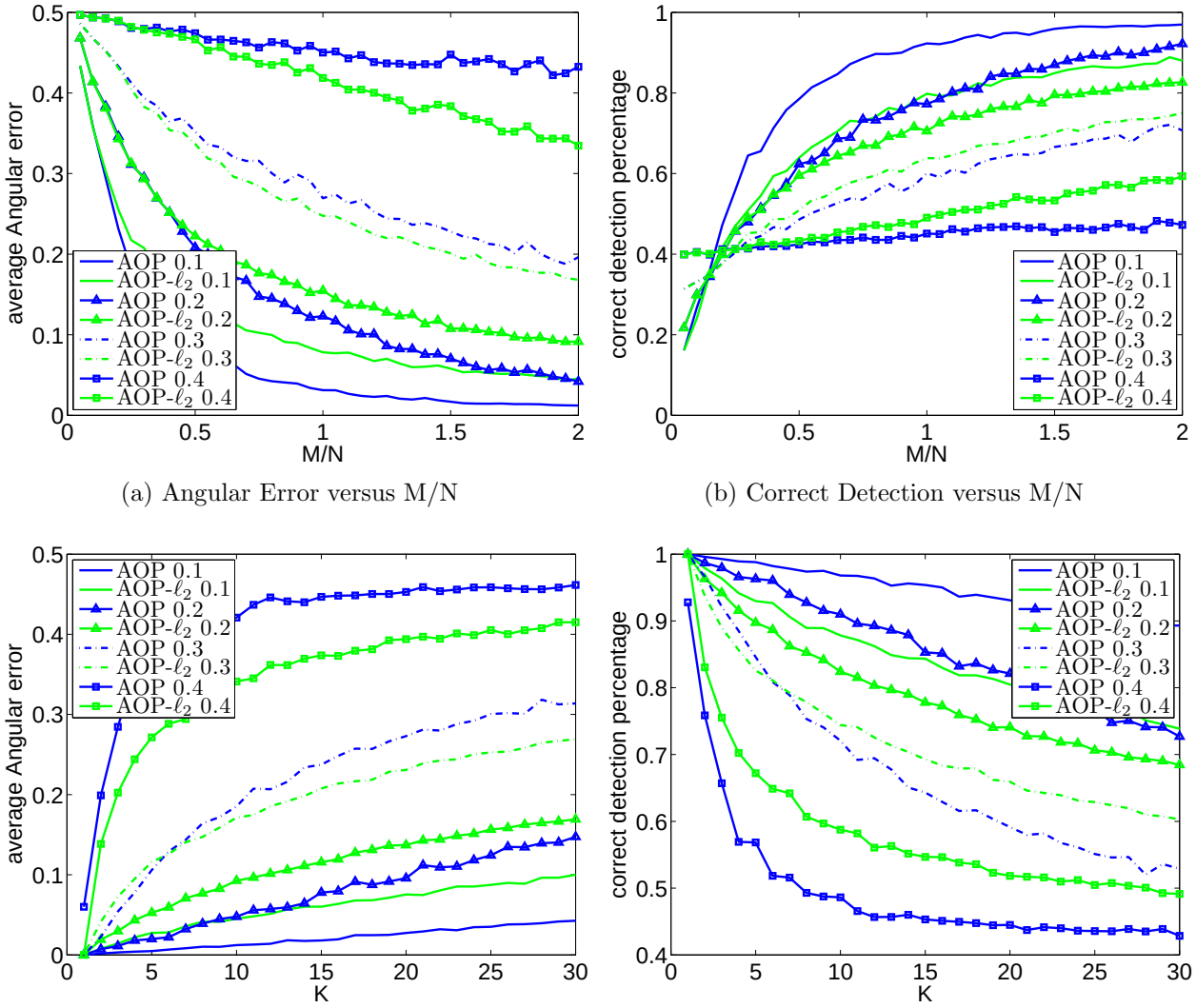
the measurements are corrupted, AOP can still recover a signal with SNR greater than 20 using less than 0.5 bits per coefficient of x^*. In Hamming error comparison, AOP and AOP-f beat other algorithms significantly when $M/N > 0.15$. Moreover, we see that the average Hamming error of AOP and AOP-f is extremely close to zero when $M/N > 0.5$. When $M/N < 0.15$, the seeming failure of AOP and AOP-f compared with BIHT is due to the fact that there is usually more than one solution to (4.6) for really small M, and with high probability our method will return one solution with L sign flips, which may not be the actual one. Hence we may not be able to detect the actual errors in the measurements.

We also try to explore the relationship between the Hamming error between $A(x)$ and $A(x^*)$ and the reconstruction angular error. With $N = 1000$, $K = 10$ and the noise level 3% fixed, we plot the Hamming error versus angular error for three different M values in Figure 4.4. Since AOP and AOP with flips tend to return almost the same results if we use the same objective (one-sided ℓ_1 or one-sided ℓ_2) for x update, we only compare the results acquired by BIHT, BIHT-ℓ_2, AOP and AOP-ℓ_2. We can see clearly that almost all the blue (+) points stay in the lower left part of the graph for $M = 0.7N$ and $M = 1.5N$, which proves that AOP gives more consistent results compared with other three algorithms. For these two values of M, the average angular error is close to a linear function of average Hamming

error, which is predicted by BϵSE property in [67]. We also plot an empirical upper bound $\epsilon_{1000} + \epsilon_H$ for AOP defined in [67], where ϵ_{1000} is the largest angular error of AOP and ϵ_H is the Hamming distance. Especially for the "under-sampled" case like $M = 0.1N$, none of these algorithms are able to return consistent reconstructions, as we can see the points scatter almost randomly over the domain. In this case the results obtained by BIHT stay really close to those gained by AOP. As mentioned above, this is because AOP may not be able to detect the exact sign flips in the noisy measurements when M is too small.

4.4.3 High noise levels

In this subsection, we study the performance of AOP and AOP-ℓ_2 when a large number of measurements are corrupted. Two settings are considered. In the first experiment, we fix $N = 1000$, $K = 10$, and change the M/N ratio between 0.05 and 2. Four different noise levels are considered from 0.1 to 0.4 and we record the average angular error and correct detection probability from 100 tests. In the second setting, we fix $M = 2000$, $N = 1000$ and change K from 1 to 30. Still, four noise levels are considered and the mean results from 100 tests are recorded. From Figure 4.5 (a) and (b), we can see a similar trend for the behavior of angular error and correct detection probability as we have discovered in Figure 4.3. According to (c), (d), for all the noise levels the performance of these two algorithms tends to get worse as K increases. We also have another interesting discovery that when the noise level is greater than 0.2, AOP-ℓ_2 turns out to be a better choice than AOP. This is because when the noise level is extremely high, even with outlier detection technique, lots of sign flips remain in the recovered measurements, and this new "noise level" is still relatively high. According to [67], BIHT-ℓ_2 outperforms BIHT when the measurements contain lots of sign flips. Therefore, when the noise level is high enough, AOP-ℓ_2 is considered as a better choice compared with AOP.



(a) Angular Error versus M/N

(b) Correct Detection versus M/N

(c) Angular Error versus K

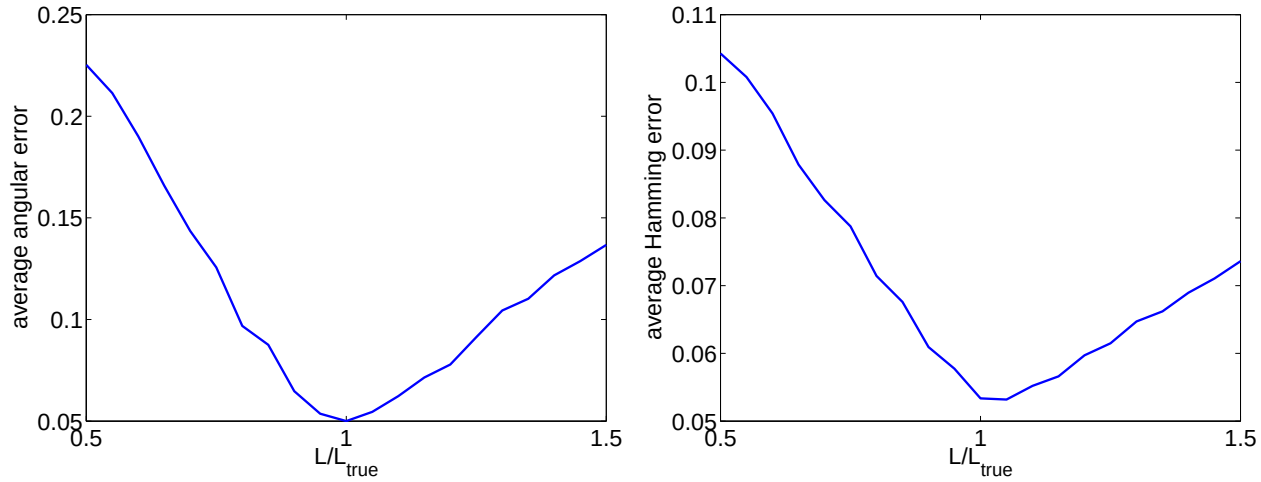
(d) Correct Detection versus K

Figure 4.5: The performance of AOP and AOP-ℓ_2 under different noise levels. (a) average angular error versus M/N with different noise levels, (b) correct detection percentage versus M/N with different noise levels, (c) average angular error versus K with different noise levels, (d) correct detection percentage versus K with different noise levels. The performance gets better when we increase M/N or decrease K.

4.4.4 L mismatch

In Figure 4.6, we analyze the influence of incorrect selection of L on AOP. Here we choose $M = N = 1000$, $K = 10$, noise level 5%, and change the input value from $0.5L$ to $1.5L$. 100 tests are conducted and the mean results are recorded. It is easily seen that the error will become larger when the input L digresses from its true value. According to this plot, we

know that in order to obtain good performance for our method, we should choose a proper L as input.



(a) Angular Error (b) Hamming Error

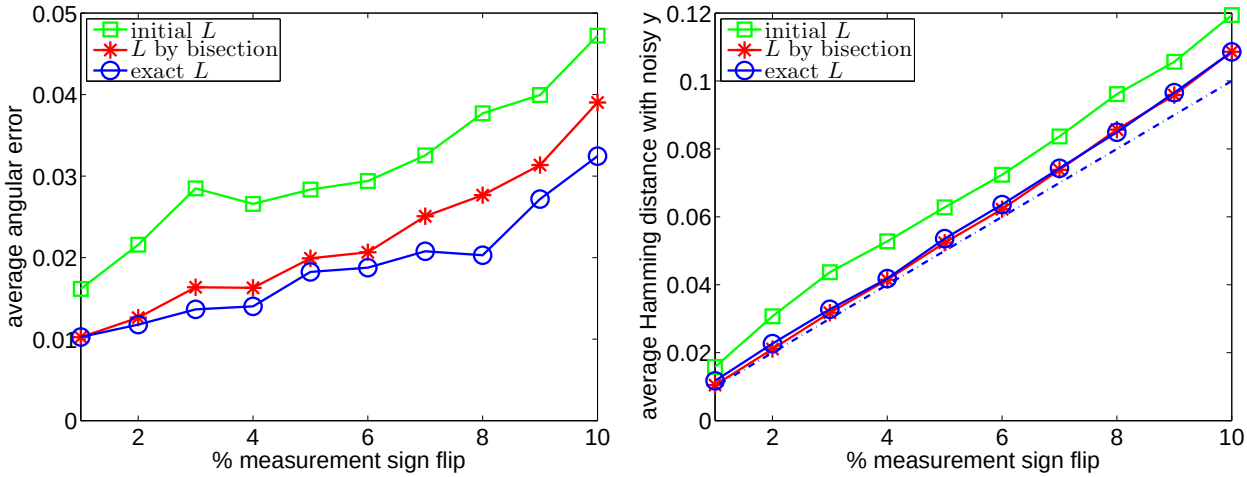
Figure 4.6: The performance of AOP with different L inputs. L has to stay close to its true value in order to get good performance.

4.4.5 Unknown L

To show that our method works even when L is not given, we use the method described in Section 4.3 to find an approximation of L, and compare the results of AOP with different L's. Here $M = N = 1000$, $K = 10$ are fixed, and 10 different noise levels (from 1% to 10%) are tested. Three inputs for L: the initial L_0 predicted from the result of BIHT-ℓ_2, L obtained from bisection method, exact L, are used in AOP to obtain the results. The following Figure 4.7 is depicted with the average results from 100 trials. Even with the initial L_0, the results are comparable to those with exact L, and the bisection method can provide a better approximation for L with longer time for predicting L.

4.5 Conclusion

In this chapter, we propose a method based on adaptive outlier pursuit for robust 1-bit compressive sensing. By iteratively detecting the sign flips in measurements and recovering



(a) Angular error

(b) Hamming distance with noisy y

Figure 4.7: Comparison of results by different L's at different noise levels from 1% to 10%. (a) average angular error versus noise level, (b) average Hamming distance between $A(x)$ and noisy y versus noise level. By choosing appropriate L as the input, we can obtain the results comparable to those with exact L.

the signals from "correct" measurements, this method can obtain better results in both finding the noisy measurements and recovering the signals, even when there are a lot of sign flips in the measurements. Four algorithms (AOP, AOP-f, AOP-ℓ_2 and AOP-ℓ_2-f) are given based on this method, and the performance of these four algorithms is shown in the numerical experiments. The algorithms based on one-sided ℓ_1 objective (AOP and AOP-f) have better performance compared to the other two algorithms (AOP-ℓ_2 and AOP-ℓ_2-f), which are based on one-sided ℓ_2 objective when the noise level is not high (less than 20%). When the noise level is extremely high, AOP-ℓ_2 is a better choice compared with AOP. In addition, we propose a simple method to find a candidate for the number of sign flips L when L is unknown and the numerical experiments show that the performance of AOP with this inexact input L is comparable with that of exact L.

Appendix

In this appendix, we show the equivalence of problems (4.6) and (4.7). If (x, n) satisfies the constraints of problem (4.7), we can define

$$\Lambda_i = \begin{cases} 1, & \text{if } n_i = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4.11}$$

Then we have $\phi(y_i, (\Phi x)_i + n_i) = 0$ if $\Lambda_i = 0$, as we can always find n_i such that $\phi(y_i, (\Phi x)_i + n_i) = 0$ for fixed x. If $\Lambda_i = 1$, we have $n_i = 0$, thus $\phi(y_i, (\Phi x)_i + n_i) = \phi(y_i, (\Phi x)_i)$. Therefore, problem (4.7) is equivalent to

$$\begin{aligned} \underset{x,n}{\text{minimize}} \quad & \sum_{i=1}^{M} \Lambda_i \phi(y_i, (\Phi x)_i) \\ \text{s.t.} \quad & \|n\|_0 \leq L, \\ & \|x\|_2 = 1, \quad \|x\|_0 \leq K. \end{aligned} \tag{4.12}$$

From the relation of Λ and n in (4.11), we know that the constraint $\|n\|_0 \leq L$ in the above problem can be replaced with the constraint on Λ defined in (4.6). Therefore, problems (4.6) and (4.7) are equivalent.

CHAPTER 5

Impulse Noise Removal

5.1 Introduction

Observed images are often corrupted by impulse noise during image acquisition and transmission, caused by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or bit errors in transmission [76]. There are two common types of impulse noise: salt-and-pepper impulse noise and random-valued impulse noise. Assume that the dynamic range of an image is $[d_{\min}, d_{\max}]$. For images corrupted by salt-and-pepper impulse noise, the noisy pixels can take only two values $d_{\min}$ and $d_{\max}$, while for images corrupted by random-valued impulse noise, the noisy pixels can take any random value between $d_{\min}$ and $d_{\max}$.

In this work, the original unknown image u is defined on a domain Ω, and the observed image f is modeled as

$$f = \begin{cases} Hu + n_1 & x \in \Omega_1 \\ n_2 & x \in \Omega_1^c := \Omega \backslash \Omega_1. \end{cases} \tag{5.1}$$

Here, n_2 is the impulse noise, and n_1 is the additive zero-mean Gaussian white noise. H is the identity or a blurring operator. The subset Ω_1^c of Ω denotes the region where the information of Hu is missing. The problem is to find the true image u from observed f given the operator H.

If Ω_1^c is empty, there is no impulse noise, then we have $f = Hu + n_1$, which is an image denoising (and deblurring) problem, and it has been extensively studied by both signal processing researchers and mathematicians. If Ω_1^c is not empty and known, this can be considered as an image inpainting (and deblurring) problem.

Here, we will consider the last and most difficult case where Ω_1^c is not empty and unknown. The challenge of this problem is to restore the lost details, and remove the impulse noise simultaneously. If $n_1 = 0$, this problem is an impulse noise removal (and deblurring) problem and if $n_1 \neq 0$ it becomes a mixed Gaussian impulse noise removal (and deblurring) problem. There are already several types of approaches for solving this problem.

The first type of approaches treats n_2 as outliers and uses the L_1 norm in the fidelity term to increase the robustness of inpainting to outliers [77, 78, 79, 80], and the problem is to solve

$$\underset{u}{\text{minimize}} \int_\Omega |Hu - f| dx + J(u), \tag{5.2}$$

where $J(u)$ is a regularization on the true image u. There are many candidates for the regularization $J(u)$, some examples are Tikhonov regularization [81]; Geman and Reynolds' half quadratic variational models [82]; Rudin, Osher and Fatemi's total variation models [2, 83]; and framelet based models [84, 85]. This approach does not need to find the damaged pixels and performs well when there is only impulse noise. However, for the case of images corrupted by mixed Gaussian impulse noise, the Gaussian noise is not treated properly.

The second type of approaches are the two-stage approach [86, 87, 88, 89, 85, 90], which estimates the inpainting region Ω_1^c before estimating u. In these approaches, the second stage becomes a regular image inpainting (and deblurring) problem [91, 92, 93, 94]:

$$\underset{u}{\text{minimize}} \frac{1}{2} \int_{\Omega_1} (Hu - f)^2 dx + J(u). \tag{5.3}$$

The success of these two-stage approaches relies on the accurate detection of Ω_1^c, e.g., adaptive median filter (AMF) [3] is used to detect salt-and-pepper impulse noise, while adaptive center-weighted median filter (ACWMF) [4] and rank-ordered logarithmic difference (ROLD) [95] are utilized to detect random-valued impulse noise.

Though adaptive median filter can detect most pixels damaged by salt-and-pepper impulse noise, it is more difficult to detect pixels corrupted by random-valued impulse noise

than salt-and-pepper impulse noise. Recently, by considering two different types of noise, Dong et.al. [96] proposed a new method using framelets to remove random-valued impulse noise plus Gaussian noise by solving

$$\underset{u,v}{\text{minimize}} \ \frac{1}{2}\int_{\Omega}(Hu+v-f)^2dx+\lambda\|Wu\|_1+\int_{\Omega}|v|dx, \tag{5.4}$$

where W is a transformation from the image to the framelet coefficients. Two unknowns u (restored image) and v (noise) are introduced into this variational method, and their methods can simultaneously find u and v using split Bregman iterations [97].

Dong et.al.'s method uses ℓ_1 norm as a convex approximation of ℓ_0 to make the result v sparse, and keep the problem convex in the meantime. However, using non-convex optimization (ℓ_p when $p<1$) has better performance than convex optimization in dealing with sparsity, as shown in compressive sensing [98]. Even ℓ_0 minimization and smoothed ℓ_0 minimization are used in many algorithms [99, 100]. In this chapter, we will use ℓ_0 instead of ℓ_1 minimization in the problem, and by using ℓ_0, the problem in u and v is shown to be equivalent to a problem in u and Ω_1. Thus using alternating minimization algorithms, it can be solved easily by alternately solving the image inpainting problem and finding the damaged pixels.

The work is organized as follows. An introduction of ACWMF is given in section 5.2. We will show in this section that ACWMF cannot detect the pixels corrupted by random-valued impulse noise correctly. In section 5.3 and 5.4, we introduce our general methods for removing impulse noise using two different treatments for ℓ_0: I) the ℓ_0 term is put in the objective function, II) the ℓ_0 term is in the constraint. The algorithms for these two methods and performances are similar. The convergence analysis is shown in section 5.5. These algorithms will converge to coordinatewise minimum points. In addition, they will converge to local minimum points (with probability one) with some modifications in the algorithms. Some experiments are given in section 5.6 to show the efficiency of proposed method for removing impulse noise and mixed Gaussian impulse noise. We will end this work by a short conclusion section.

5.2 The Adaptive Center-Weighted Median Filter

In order to remove random-valued impulse noise, adaptive center-weighted median filter (ACWMF) [4] is considered as a good method for detecting the damaged pixels when the noise level is not too high. So the result of ACWMF is often utilized in two-stage approaches [88, 89, 85] to estimate the set of damaged pixels Ω_1^c.

If u is a noisy M-by-N $(M \times N)$ grayscale image, and $u_{i,j}$ is the gray level at pixel $(i,j) \in \{1,\cdots,M\} \times \{1,\cdots,N\}$, the expression of the ACWMF filter is as follows:

$$y_{i,j}^{2k} = \text{median}\{u_{i-s,j-t}, (2k) \diamond u_{i,j} | -h \leq s,t \leq h\},$$

where $(2h+1) \times (2h+1)$ is the window size, and $\diamond$ represents the repetition operation. For $k = 0, 1, \cdots, J-1$, where $J = 2h(h+1)$, we can determine the differences $d_k = |y_{i,j}^{2k} - u_{i,j}|$. They satisfy the condition $d_k \leq d_{k-1}$ for $k \geq 1$. To determine if the considered pixel (i,j) is noisy, a set of thresholds T_k is utilized, where $T_{k-1} > T_k$ for $k = 1, \cdots, J-1$. The output of the filter is defined in the following manner:

$$u_{\text{ACWMF}} = \begin{cases} y_{i,j}^0, & \text{if } d_k > T_k \text{ for some } k, \\ u_{i,j}, & \text{otherwise.} \end{cases} \tag{5.5}$$

Usually, the window size is chosen as 3×3 (i.e., $h = 1$ and $J = 4$), four thresholds T_k $(k = 0, \cdots, 3)$ are needed, and they are calculated as follows:

$$T_k = s \cdot \text{MAD} + \delta_k, \tag{5.6}$$

$$\text{MAD} = \text{median}\{|u_{i-s,j-t} - y_{i,j}^1| - h \leq s,t \leq h\}, \tag{5.7}$$

where $[\delta_0, \delta_1, \delta_2, \delta_3] = [40, 25, 10, 5]$ and $0 \leq s \leq 0.6$.

The performance of ACWMF is demonstrated in Figure 5.1 on a 256x256 blurry cameraman image when 25% of the pixels are corrupted by random-valued impulse noise. For the first case (top row), the set having the corrupted pixels is chosen randomly, and from the

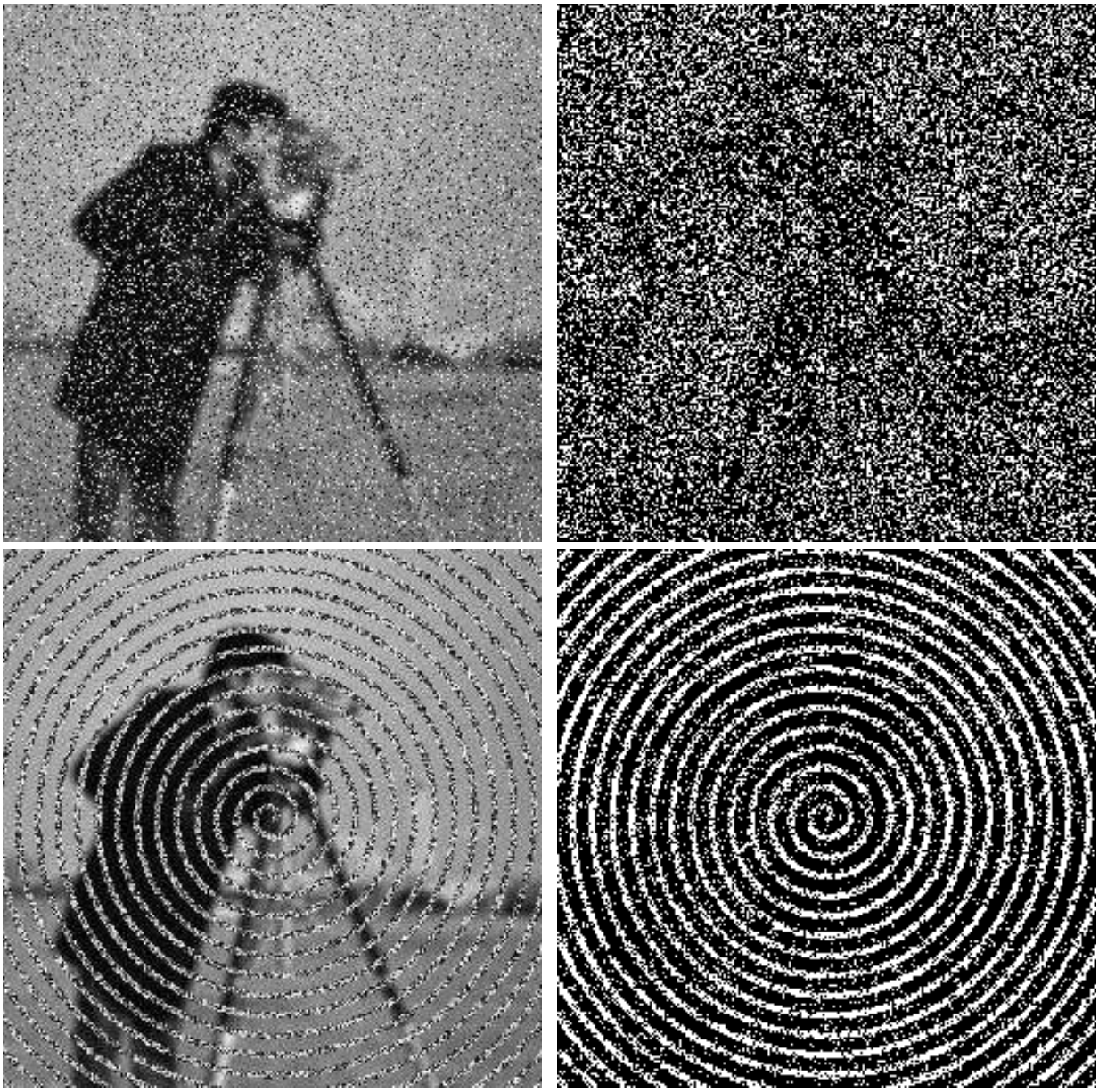


Figure 5.1: Noisy images and the sets detected by ACWMF. Left column: noisy images corrupted by random-valued impulse noise; Right column: the sets of damaged pixels detected by ACWMF. White point means that the corresponding pixel is corrupted by impulse noise.

result obtained from ACWMF, we can still see some features of the cameraman image. For the other case (bottom row), we specify a set containing the damaged pixels, and ACWMF misses quite a lot of real noise and false-hits some noise-free pixels. The success of two-stage methods depends on the accuracy of detecting damaged pixels. As we will show in the numerical experiments, the performance of two-stage methods using ACWMF for detect-

ing damaged pixels is not good enough, and even worse than the methods using L_1 fidelity term. If the detection of damaged pixels can be updated during the image restoration with increasing accuracy, then the performance will be better. In the next two sections, we will provide two methods that iteratively update the detection of damaged pixels and restore the images.

5.3 Blind Inpainting Models using ℓ_0 Norm

5.3.1 Formulation

For a $M \times N$ image, $\Lambda \in \{0,1\}^{M\times N}$ is a binary matrix denoting the undamaged pixels (pixels not corrupted by impulse noise):

$$\Lambda_{i,j} = \begin{cases} 1, & \text{if pixel } (i,j) \in \Omega_1, \\ 0, & \text{otherwise.} \end{cases} \tag{5.8}$$

We use $\mathcal{P}_\Lambda(\cdot)$ to represent the projection of an image onto a matrix supported by Λ:

$$\mathcal{P}_\Lambda(u)_{i,j} = \begin{cases} 0, & \text{if } \Lambda_{i,j} = 0, \\ u_{i,j}, & \text{if } \Lambda_{i,j} = 1. \end{cases} \tag{5.9}$$

Given a degraded image f, our objective is to estimate the damaged (or missing) pixels and restore them. We propose the following model using ℓ_0 minimization to solve this problem:

$$\underset{u,v}{\text{minimize}} \ \frac{1}{2}\int_\Omega (Hu + v - f)^2 dx + \lambda_1 J(u) + \lambda_2 \|v\|_0, \tag{5.10}$$

where $J(u)$ is the regularization term on the image, λ_1 and λ_2 are two positive parameters. The parameter λ_1 is dependent on the noise level of n_1. The higher the noise level, the larger the parameter should be. The parameter λ_2 is dependent on the noise level of impulse noise. The difference from Dong et.al.'s method is that ℓ_1 norm is replaced by ℓ_0. However, it is

difficult to solve this problem because of the ℓ_0 term in the function. We will show that this problem in u and v is equivalent to a problem in u and Ω_1, which is easy to solve.

Assume that (u^*, v^*) is the optimal solution to problem (5.10). When $v^*_{i,j} \neq 0$, we have $v^*_{i,j} = f_{i,j} - (Hu^*)_{i,j}$. Therefore if we denote

$$\Lambda_{i,j} = \begin{cases} 0, & \text{if } v_{i,j} \neq 0, \\ 1, & \text{if } v_{i,j} = 0. \end{cases} \tag{5.11}$$

then problem (5.10) is equivalent to

$$\underset{u,\Lambda}{\text{minimize}} \;\; F_1(u, \Lambda) \equiv \frac{1}{2} \sum_{i,j} \Lambda_{i,j}((Hu)_{i,j} - f_{i,j})^2 + \lambda_1 J(u) + \lambda_2 \sum_{i,j} (1 - \Lambda_{i,j}), \tag{5.12}$$

which is a problem in u and Ω_1, because each Ω_1 corresponds to a unique binary matrix Λ. Problem (5.12) can be solved easily by alternating minimization method, and the algorithm for solving (5.12) is described below.

5.3.2 Algorithm

The objective function defined in (5.12) is non-convex and has both continuous and discrete variables. It is still difficult to solve it in the pair (u, Λ), but we can use alternating minimization method, which separates the energy minimization over u and Λ into two steps. For solving the problem in u with Λ fixed, it is a convex optimization problem for image inpainting and finding Λ with u fixed is a combinatorial optimization problem, which can be solved in one step. These two subproblems are

1) Finding u: Given an estimate of the support matrix Λ, the minimization over u is just an image inpainting (and deblurring) problem [101]:

$$\underset{u}{\text{minimize}} \;\; \frac{1}{2} \int_{\Omega_1} (Hu - f)^2 dx + \lambda_1 J(u). \tag{5.13}$$

There are many existing methods for solving this problem.

2) Finding Λ: Given an estimate of the image u, the minimization over Λ becomes:

$$\underset{\Lambda}{\text{minimize}} \ \frac{1}{2}\sum_{i,j}\Lambda_{i,j}((Hu)_{i,j}-f_{i,j})^2-\lambda_2\sum_{i,j}\Lambda_{i,j}. \tag{5.14}$$

Since this minimization problem of Λ is separable, it can be solved exactly in only one step:

$$\Lambda_{i,j}=\begin{cases} 0 & \text{if } ((Hu)_{i,j}-f_{i,j})^2/2>\lambda_2, \\ 0 \text{ or } 1 & \text{if } ((Hu)_{i,j}-f_{i,j})^2/2=\lambda_2, \\ 1 & \text{if } ((Hu)_{i,j}-f_{i,j})^2/2<\lambda_2. \end{cases} \tag{5.15}$$

Therefore, the proposed algorithm for blind inpainting with ℓ_0 minimization is iteratively finding u and Λ. If the regularization $J(\cdot)$ is chosen to be total variation, the detailed algorithm is described below.

5.3.3 TV Blind Inpainting

If the regularization for the image u is the total variation, H is the identity operator and the noise n_1 is zero-mean Gaussian white noise, then problem (5.12) becomes

$$\underset{u,\Lambda}{\text{minimize}} \ F_1(u,\Lambda)\equiv\sum_{i,j}\frac{1}{2}\Lambda_{i,j}(u_{i,j}-f_{i,j})^2+\lambda_1\int_\Omega|\nabla u|+\lambda_2\sum_{i,j}(1-\Lambda_{ij}). \tag{5.16}$$

The step for finding u is

$$\underset{u}{\text{minimize}} \ \sum_{i,j}\frac{1}{2}\Lambda_{i,j}(u_{i,j}-f_{i,j})^2+\lambda_1\int_\Omega|\nabla u|, \tag{5.17}$$

which is the famous TV inpainting model [102, 103]. Numerous algorithms proposed for solving TV denoising problem can be adopted to solve this TV inpainting problem with some necessary modifications. Some examples are algorithms based on duality [104, 105], augmented Lagrangian methods [106, 107], and split Bregman iterations [97, 108].

The algorithm for total variation blind inpainting is described below, the initial Λ^0 is chosen by the methods for detecting the impulse noise (AMF for salt-and-pepper impulse

noise and ACWMF for random-valued impulse noise). Usually three iterations are sufficient, as shown in the experiments.

Algorithm 6 Proposed total variation blind inpainting algorithm.

Input: f, λ_1, λ_2, Λ^0, ϵ
Initialization: $k = 1$.
while $k < 2$ or $F_1(u^k, \Lambda^k) - F_1(u^{k-1}, \lambda^{k-1}) > \epsilon$ **do**
    Obtain u^k by solving (5.17),
    Obtain Λ^k by (5.15).
    $k = k + 1$.
end while

5.4 Blind Inpainting Using Adaptive Outlier Pursuit

In the previous section, we proposed a method for blind inpainting by putting a ℓ_0 term in the objective function, which can be solved by iteratively updating the set Λ and restoring the image. Instead of putting the ℓ_0 term in the objective function, we can also put a constraint on $\|v\|_0$, which will be equivalent to a constraint on $\sum_{i,j} \Lambda_{i,j}$. This technique has been applied to robust 1-bit compressive sensing where there are sign flips in the binary measurements belonging to $\{-1, 1\}$ [109]. We proposed an algorithm, named adaptive outlier pursuit (AOP), which can adaptively find the sign flips and reconstruct the signal by using other measurements assumed to be correct. Since images corrupted by impulse noise can also be considered as sparsely corrupted measurements, the same idea can be applied to impulse noise (and mixed Gaussian impulse noise) removal by iteratively finding the pixels corrupted by impulse noise and recovering the image using the other pixels.

Let us assume that the number of pixels corrupted by impulse noise is bounded above by L, this can be obtained from the noise level of impulse noise. Therefore, the new problem is

$$\begin{aligned} \underset{u,v}{\text{minimize}} \quad & \tfrac{1}{2}\textstyle\int_\Omega (Hu + v - f)^2 dx + \lambda_1 J(u), \\ \text{subject to} \quad & \|v\|_0 \leq L \end{aligned} \tag{5.18}$$

Similarly, we have the equivalent form in (u, Λ), which is

$$\begin{array}{ll} \underset{u,\Lambda}{\text{minimize}} & F_2(u,\Lambda) \equiv \sum_{i,j} \frac{1}{2}\Lambda_{i,j}((Hu)_{i,j} - f_{i,j})^2 + \lambda_1 J(u), \\ \text{subject to} & \sum_{i,j}(1-\Lambda_{i,j}) \le L, \quad \Lambda_{i,j} \in \{0,1\}. \end{array} \tag{5.19}$$

This problem can also be solved iteratively as in previous section.

The u-subproblem is the same as the previous one, and the Λ-subproblem is slightly different. In order to update Λ, we have to solve

$$\begin{array}{ll} \underset{\Lambda}{\text{minimize}} & \sum_{i,j}((Hu)_{i,j} - f_{i,j})^2/2 \\ \text{subject to} & \sum_{i,j}(1-\Lambda_{i,j}) \le L, \Lambda_{i,j} \in \{0,1\}. \end{array} \tag{5.20}$$

This problem is to choose $M \times N - L$ elements with least sum from $M \times N$ elements $\{((Hu)_{i,j} - f_{i,j})^2/2\}_{i=1,j=1}^{M,N}$. Given an u estimated from (5.17), we can update Λ in one step:

$$\Lambda_{i,j} = \begin{cases} 0, & \text{if } ((Hu)_{i,j} - f_{i,j})^2/2 \ge \lambda_2, \\ 1, & \text{if } ((Hu)_{i,j} - f_{i,j})^2/2 < \lambda_2, \end{cases} \tag{5.21}$$

where λ_2 is the L^{th} largest term of $\{((Hu)_{i,j} - f_{i,j})^2/2\}_{i=1,j=1}^{M,N}$. If the L^{th} and $(L+1)^{th}$ largest terms are equal, then we can choose any binary matrix Λ such that $\sum_{i,j}\Lambda_{i,j} = M \times N - L$ and

$$\min_{i,j,\Lambda_{i,j}=0}((Hu)_{i,j}f_{i,j})^2/2 \ge \max_{i,j,\Lambda_{i,j}=1}((Hu)_{i,j} - f_{i,j})^2/2. \tag{5.22}$$

The algorithm for total variation blind inpainting using AOP is described below.

Algorithm 7 Proposed total variation blind inpainting using AOP.

Input: f, λ_1, L, Λ^0, ϵ
Initialization: $k = 1$.
while $k < 2$ or $F_2(u^k, \Lambda^k) - F_2(u^{k-1}, \lambda^{k-1}) > \epsilon$ **do**
  Obtain u^k by solving (5.17),
  Obtain Λ^i by (5.21).
  $k = k + 1$.
end while

The difference between these two algorithms is the Λ-subproblem, the threshold λ_2 is fixed for algorithm 6, while λ_2 is changing for AOP. AOP can be considered as one special case of algorithm 6 with changing λ_2. However the performance of these two algorithms is similar, and the parameter L is easier to obtain than λ_2. So we will only use algorithm 7 for numerical experiments.

These two algorithms are both for the case where H is the identity, and there is no blurring in the observed image. In order to show that our idea can be applied to other two-stage approaches easily, we will introduce in the next subsection a method for dealing with the case when H is a blurring operator using framelet.

5.4.1 Framelet-Based Deblurring

Though total variation (TV) is popular for regularization in recent years as it preserves edges, its limitation is that TV-based regularization can not preserve the details and textures very well in the regions of complex structures due to the stair-casing effects [110]. Therefore, framelet-based algorithms are introduced in [84, 85] for impulse noise removal. In [85], iterative framelet-based approximation/sparsity deblurring algorithm (IFASDA) and accelerated algorithm of IFASDA (Fast_IFASDA) are proposed to deblur images corrupted by impulse plus Gaussian noise. They both have two steps, the first step is to apply AMF or ACWMF on f to estimate the set Ω_1^c; the second step is deblurring the image using framelet from $\mathcal{P}_\Lambda(f)$.

The energy function to be minimized is

$$E(u) := \|\phi(\mathcal{P}_\Lambda(Hu - f))\|_1 + \lambda\|\mathcal{O}_{wH}u\|_1, \tag{5.23}$$

where $\mathcal{O}_{wH} := [w_1\mathcal{O}_1^2\ w_2\mathcal{O}_2^T\ \cdots\ w_{17}\mathcal{O}_{17}^T]^T$ and $\phi(x) = \frac{\eta x^2}{\eta+|x|}$. $\{\mathcal{O}_k\}_{k=0}^{17}$ are the matrix representations of the tight framelet filters $\{\tau_k\}_{k=0}^{17}$ under a proper boundary condition [111] (see Section III in [85] for the tight framelet filters $\mathcal{O}$ and weight w). In addition a matrix $\mathcal{O}_H$ is

formed by stacking the matrices $\mathcal{O}_k, 1 \leq k \leq 17$ together, that is

$$\mathcal{O}_H = [\mathcal{O}_1^2 \ \mathcal{O}_2^T \ \cdots \ \mathcal{O}_{17}^T]^T. \tag{5.24}$$

Associated with the matrix $\mathcal{O}_H$ and a diagonal matrix

$$\Gamma := \text{diag}(\cdots, \gamma_l, \cdots), \qquad \gamma_l \geq 0, \tag{5.25}$$

the shrinkage function $Sh_\Gamma : \mathbf{R}^{M\times N} \to \mathbf{R}^{M\times N}$ is defined as follow:

$$Sh_\Gamma(f) := \mathcal{O}_0^T\mathcal{O}_0 f + \mathcal{O}_H^T \cdot \text{shrink}(\mathcal{O}_H f, \Gamma) \tag{5.26}$$

where shrink is the componentwise thresholding operator

$$\text{shrink}(x, \Gamma)[l] = \text{sign}(x[l]) \max\{|x[l]| - \gamma_l, 0\}. \tag{5.27}$$

The algorithm is

$$u^{n+1} = Sh_{\Gamma_n}(u^n - \beta_n \nabla J_n(u^n)) \tag{5.28}$$

where

$$J_n(y) := \|\phi_n(\mathcal{P}_\Lambda(Hy - f))\|_1 + \lambda_n \|\mathcal{O}_{wH} y\|_1, \tag{5.29}$$

with

$$\phi_n(x) = \frac{\eta_n x^2}{\eta_n + |x|}. \tag{5.30}$$

The ACWMF can not estimate the set ${\Omega_1}^c$ perfectly, especially when noise level is high and two types of impulse noise are present, thus frequently updating the set is necessary.

The new constrainted optimization problem using AOP is

$$\begin{aligned} \underset{u,\Lambda}{\text{minimize}} \quad & \|\phi(\mathcal{P}_\Lambda(Hu-f))\|_1 + \lambda\|\mathcal{O}_{wH}u\|_1 \\ \text{subject to:} \quad & \textstyle\sum_{i,j}(1-\Lambda_{i,j}) \le K. \end{aligned}$$

If alternating minimizing method is utilized, the first step for finding u is the same as IFASDA, and the second step for updating Ω_1 is very easy to find from (5.21). The modified IFASDA algorithm Ada_IFASDA is shown below:

Algorithm 8 Proposed framelet-based deblurring algorithm (Ada_IFASDA).

Input: f, L, Λ,
for $k = 1, 2, \cdots,$ **do**
  Estimate η_k, β_k, λ_k, Γ_k.
  Compute u^{k+1}.
  Stop if u^{k+1} meets a stopping criteria
  if mod(k,5)=0 **then**
    Obtain Λ by (5.21).
  end if
  $k = k + 1$.
end for

Here, the input Λ is obtained from ACWMF. In fact, we do not need to wait until the step for finding u converges, several iterations in IFASDA are sufficient for the full algorithm to converge. We choose to update the set ${\Omega_1}^c$ for every five iterations in IFASDA.

Similarly, Fast_IFASDA can also be modified into Ada_Fast_IFASDA by adding the steps for updating ${\Omega_1}^c$ every five iterations in Fast_IFASDA.

5.5 Convergence Analysis

First of all, we will show that the constraint $\Lambda_{i,j} \in \{0,1\}$ in (5.12) and (5.19) can be replaced by the relaxed constraint $\Lambda_{i,j} \in [0,1]$. Because for every fixed $\bar{u}$, we can always find a minimum point of $F_1(\bar{u},\Lambda)$ (or $F_2(\bar{u},\Lambda)$) satisfying the constraint $\Lambda_{i,j} \in \{0,1\}$. Therefore, for any local minimum of $F_1(u,\Lambda)$ (or $F_2(u,\Lambda)$) with the relaxed constraint, we can find a corresponding Λ satisfying the unrelaxed constraint such that the local minimum is the

same.

In this section, we establish some convergence results for these two algorithms. We will show that the algorithm will stop in finite steps, and the output is a coordinatewise minimum point of $F_1(u,\Lambda)$ (or $F_2(u,\Lambda)$) with relaxed constraint $\Lambda_{i,j} \in [0,1]$. In addition, we can modify a little bit the algorithm and the output will be a local minimum point (with probability one).

Let us define a new function combining $F_1(u,\Lambda)$ and $F_2(u,\Lambda)$ as follows

$$F(u,\Lambda) = \sum_{i,j} \frac{1}{2}\Lambda_{i,j}((Hu)_{i,j} - f_{i,j})^2 + \lambda_1 J(u) + R(\Lambda), \tag{5.31}$$

where $R(\Lambda) = \lambda_2 \sum_{i,j}(1-\Lambda_{i,j}) + \chi_A(\Lambda)$ with $A = \{\Lambda : \Lambda_{i,j} \in [0,1]\}$ for $F_1(u,\Lambda)$. $\chi_A(\Lambda)$ is the characteristic function having value 0 if $\Lambda \in A$ and $+\infty$ if $\Lambda \notin A$. For $F_2(u,\Lambda)$, $R(\Lambda) = \chi_B(\Lambda)$ with $B = \{\Lambda : \Lambda_{i,j} \in [0,1], \sum_{i,j}(1-\Lambda_{i,j}) \leq L\}$.

Theorem 5.5.1. *Both algorithms will converge in a finite number of steps and the output* (u^*,Λ^*) *is a coordinatewise minimum point of* $F(u,\Lambda)$.

Proof. Since $\Lambda_{i,j} \in \{0,1\}$, there are only finite number of Λ's and the algorithm will stop in a finite number of steps if the u-subproblem and Λ-subproblem are solved exactly. Assume that at step i, the function $F(u,\Lambda)$ stops decreasing, which means

$$F(u^k,\Lambda^k) = F(u^{k+1},\Lambda^{k+1}). \tag{5.32}$$

Together with the nonincreasing property of the algorithm

$$F(u^k,\Lambda^k) \geq F(u^{k+1},\Lambda^k) \geq F(u^{k+1},\Lambda^{k+1}), \tag{5.33}$$

we have

$$F(u^k,\Lambda^k) = F(u^{k+1},\Lambda^k) = F(u^{k+1},\Lambda^{k+1}), \tag{5.34}$$

Thus

$$F(u^k, \Lambda^k) = F(u^{k+1}, \Lambda^k) = \underset{u}{\text{minimize}}\ F(u, \Lambda^k), \tag{5.35}$$

$$F(u^k, \Lambda^k) = \underset{\Lambda}{\text{minimize}}\ F(u^k, \Lambda). \tag{5.36}$$

Then $(u^*, \Lambda^*) = (u^k, \Lambda^k)$ is a coordinatewise minimum point of $F(u, \Lambda)$. □

However, (u^*, Λ^*) may not be a local minimum point of $F(u, \Lambda)$. As shown in the next theorem, Λ^* being the unique minimum point of $F(u^*, \Lambda)$ is a sufficient condition for (u^*, Λ^*) to be a local minimum point.

Theorem 5.5.2. *If Λ^* is the unique minimum point of $F(u^*, \Lambda)$, then (u^*, Λ^*) is a local minimum point of $F(u, \Lambda)$.*

Proof. We have to show that there exists $\epsilon > 0$, such that for all $(\partial u, \partial\Lambda)$ satisfying $\|(\partial u, \partial\Lambda)\| \leq \epsilon$, $F(u^* + \partial u, \Lambda^* + \partial\Lambda) \geq F(u^*, \Lambda^*)$. $F_1(u, \Lambda)$ and $F_2(u, \Lambda)$ cases are proved separately. For $F_1(u, \Lambda)$, as

$$\begin{aligned}
&F_1(u^* + \partial u, \Lambda^* + \partial\Lambda) \\
&= \frac{1}{2}\sum_{i,j}(\Lambda^* + \partial\Lambda)_{i,j}((H(u^* + \partial u))_{i,j} - f_{i,j})^2 + \lambda_1 J(u^* + \partial u) + \lambda_2 \sum_{i,j}(1 - (\Lambda + \partial\Lambda)_{i,j}) \\
&= F_1(u^* + \partial u, \Lambda^*) + \frac{1}{2}\sum_{i,j}\partial\Lambda_{i,j}((H(u^* + \partial u))_{i,j} - f_{i,j}))^2 - \lambda_2 \sum_{i,j}\partial\Lambda_{i,j} \\
&\geq F_1(u^*, \Lambda^*) + \sum_{i,j}\partial\Lambda_{i,j}(((H(u^* + \partial u))_{i,j} - f_{i,j})^2/2 - \lambda_2).
\end{aligned}$$

We have to show that $\sum_{i,j}\partial\Lambda_{i,j}(((Hu^* + \partial u)_{i,j} - f_{i,j})^2/2 - \lambda_2) \geq 0$.

Λ^* being the unique minimum point of $F_1(u^*, \Lambda)$ implies

$$\min_{i,j,\Lambda^*_{i,j}=0}((Hu^*)_{i,j} - f_{i,j})^2/2 > \lambda_2. \tag{5.37}$$

Therefore, we can find a $\epsilon > 0$ such that for all $\|\partial u\| \leq \epsilon$,

$$\min_{i,j,\Lambda^*_{i,j}=0}((H(u^* + \partial u))_{i,j} - f_{i,j})^2/2 > \lambda_2. \tag{5.38}$$

Combining with the constraint of Λ, we have $\partial\Lambda_{i,j} \geq 0$ if $\Lambda^*_{i,j} = 0$, $\partial\Lambda_{i,j} \leq 0$ if $\Lambda^*_{i,j} = 1$. Thus $\sum_{i,j} \partial\Lambda_{i,j}(((Hu^* + \partial u)_{i,j} - f_{i,j})^2/2 - \lambda_2) \geq 0$.

For AOP, as

$$\begin{aligned} F_2(u^* + \partial u, \Lambda^* + \partial\Lambda) &= \frac{1}{2}\sum_{i,j}(\Lambda^* + \partial\Lambda)_{i,j}((H(u^* + \partial u))_{i,j} - f_{i,j})^2 + \lambda_1 J(u^* + \partial u) \\ &= F_2(u^* + \partial u, \Lambda^*) + \frac{1}{2}\sum_{i,j}\partial\Lambda_{i,j}((H(u^* + \partial u))_{i,j} - f_{i,j})^2 \\ &\geq F_2(u^*, \Lambda^*) + \frac{1}{2}\sum_{i,j}\partial\Lambda_{i,j}((H(u^* + \partial u))_{i,j} - f_{i,j})^2. \end{aligned}$$

We have to show that $\sum_{i,j} \partial\Lambda_{i,j}((Hu + \partial u)_{i,j} - f_{i,j})^2/2 \geq 0$.

Λ^* being the unique minimum point of $F_2(u^*, \Lambda)$ implies that

$$\min_{i,j,\Lambda^*_{i,j}=0}((Hu^*)_{i,j} - f_{i,j})^2/2 > \min_{i,j,\Lambda^*_{i,j}=1}((Hu^*)_{i,j} - f_{i,j})^2/2. \tag{5.39}$$

Therefore, we can find a $\epsilon > 0$ such that for all $\|\partial u\| \leq \epsilon$, we have

$$\min_{i,j,\Lambda^*_{i,j}=0}((H(u^* + \partial u))_{i,j} - f_{i,j})^2/2 > \min_{i,j,\Lambda^*_{i,j}=1}((H(u^* + \partial u))_{i,j} - f_{i,j})^2/2. \tag{5.40}$$

Combining with the constraint of Λ, we have $\partial\Lambda_{i,j} \geq 0$ if $\Lambda^*_{i,j} = 0$, $\partial\Lambda_{i,j} \leq 0$ if $\Lambda^*_{i,j} = 1$. In addition, there is a constraint $\sum_{i,j} \partial\Lambda_{i,j} \geq 0$. Thus $\sum_{i,j} \partial\Lambda_{i,j}((Hu+\partial u)_{i,j} - f_{i,j})^2/2 \geq 0$. □

From theorem 5.5.2, if (u^*, Λ^*) is not a local minimum point, then there are many minimum points for $F(u^*, \Lambda)$. A more accurate necessary condition is stated in the next theorem.

Theorem 5.5.3. *If (u^*, Λ^*) is not a local minimum point of $F(u, \Lambda)$, then there exists another minimum point $\bar{\Lambda}$ of $F(u^*, \Lambda)$ being a binary matrix, such that $F(u^*, \bar{\Lambda}) > \min_u F(u, \bar{\Lambda})$.*

Proof. Since $F(u,\Lambda)$ is linear with respect to Λ, and $\Lambda^* + \partial\Lambda$ is a linear combination of $\{\Lambda^i\}_{i=1}^{K_1+K_2}$, where Λ^i are binary matrices, we have

$$F(u^*+\partial u,\Lambda^*+\partial\Lambda) = \sum_{i=1}^{K_1}\alpha_i F(u^*+\partial u,\Lambda^i) + \sum_{i=K_1+1}^{K_1+K_2}\alpha_i F(u^*+\partial u,\Lambda^i), \tag{5.41}$$

where $\Lambda^1,\Lambda^2,\cdots\Lambda^{K_1}$ are the minimum points of $F(u^*,\Lambda)$, $\alpha_i > 0$ and $\sum_{i=1}^{K_1+K_2}\alpha_i = 1$. Because for all $i > K_1$, we have $F(u^*,\Lambda^i) > F(u^*,\Lambda^*)$, there exist $\epsilon > 0$ such that when $\|\partial u\| \le \epsilon$, we still have $F(u^*+\partial u,\Lambda^i) \ge F(u^*,\Lambda^*)$ for all $i > K_1$. Therefore, in order to find $(\partial u,\partial\Lambda)$ small enough to make $F(u^*+\partial u,\Lambda^*+\partial\Lambda) < F(u^*,\Lambda^*)$, there exist $i \le K_1$ such that $F(u^*,\Lambda^i) > \min_u F(u,\Lambda^i)$. □

From theorem 5.5.3, when the Λ-subproblem has many solutions, we can choose the best Λ with lowest $\min_u F(u,\Lambda)$, then the algorithm will stop at a local minimum. Also, we can modify the objective function $F(u,\Lambda)$ by adding $\tau\sum_{i,j}\Lambda_{i,j}r_{i,j}$, where $\{r_{i,j}\}$ are random values uniformly distributed in $[0,1]$ and τ is a small number, as Wang et.al. did in [112]. Then the algorithm will converge to a local minimum with probability one, because the minimum point of $F(u,\Lambda)$ for fixed u will be unique with probability one.

5.6 Experiments

Because the performance of these two algorithms is similar, in this section, we applied the total variation blind inpainting algorithm using AOP to remove impulse noise and mixed Gaussian impulse noise. To evaluate the quality of the restoration results, peak signal to noise ratio (PSNR) is employed. Given an image $u \in [0,255]^{M\times N}$, the PSNR of the restoration result $\hat{u}$ is defined as follows:

$$\mathrm{PSNR}(\hat{u},u) = 10\log_{10}\frac{255^2}{\frac{1}{MN}\sum_{i,j}(\hat{u}_{ij}-u_{ij})^2}. \tag{5.42}$$

There are two important types of impulse noise: salt-and-pepper impulse noise and

random-valued impulse noise. The pixels damaged by salt-and-pepper impulse noise are much easier to find as the values are either $d_{\min}$ or $d_{\max}$. The adaptive median filter (AMF) has been widely used to accurately identify most pixels damaged by salt-and-pepper impulse noise (See e.g. [113, 3]). The detection of pixels corrupted by random-valued impulse noise is much harder than those corrupted by salt-and-pepper impulse noise because the value of damaged pixels can be any number between $d_{\min}$ and $d_{\max}$. ACWMF was proposed to detect pixels damaged by random-valued impulse noise.

For the first experiment, salt-and-pepper impulse noise is considered. Because the pixels corrupted by this kind of impulse noise can only take two values, the detection of damaged pixels is easy. As an efficient method for detecting the damaged pixels, AMF is used widely in salt-and-pepper impulse noise removal. We will compare total variation blind inpainting using AOP with AMF and TVL1 [114, 115], where TVL1 is the result of solving the following problem,

$$\underset{u}{\text{minimize}} \sum_{i,j} |u_{i,j} - f_{i,j}| + \lambda \int_{\Omega} |\nabla u|, \tag{5.43}$$

using split Bregman [108]. The parameter λ is tuned to achieve the best quality of the restoration images.

Four test images are corrupted by Gaussian noise of zero mean and standard deviations $\sigma = 5, 10, 15$, then we add salt-and-pepper impulse noise with different levels ($s = 30\%, 50\%, 70\%$) on the test images, with or without the Gaussian noise. The PSNR values of the results from three methods are summarized in Table 5.1.

From Table 5.1, we can see that for salt-and-pepper impulse noise, the results from total variation blind inpainting using AOP are better than those by AMF and TVL1 for all noise levels. The visual comparison of some results is shown in Figure 5.2. We can see noisy artifacts in the background of the images obtained by AMF, and the images obtained by TVL1 are blurred with some lost details. Images restored by total variation blind inpainting using AOP are smooth in flat regions of the background and the details are kept.

We do not compare AOP with two-stage approaches because the detection of damaged

Salt-and-Pepper Impulse Noise								
$\sigma + s$	"Lena"				"House"			
	Noisy	AMF	TVL1	AOP	Noisy	AMF	TVL1	AOP
0+30%	10.68	33.80	30.97	**37.75**	10.42	38.97	36.53	**47.14**
5+30%	10.66	31.47	30.32	**34.56**	10.40	33.69	34.49	**39.09**
10+30%	10.62	27.93	29.40	**32.25**	10.39	28.90	32.79	**35.73**
15+30%	10.54	25.14	28.59	**30.41**	10.30	25.66	31.41	**33.49**
0+50%	8.44	30.35	27.98	**33.98**	8.22	34.60	31.70	**42.50**
5+50%	8.45	29.00	27.58	**32.61**	8.19	31.73	31.36	**37.43**
10+50%	8.42	26.54	27.25	**30.88**	8.18	27.84	30.37	**34.60**
15+50%	8.40	24.15	26.46	**29.50**	8.14	24.96	29.60	**32.50**
0+70%	7.00	26.85	24.90	**30.61**	6.75	30.05	26.80	**36.84**
5+70%	6.97	26.11	24.65	**29.94**	6.73	28.80	26.81	**34.44**
10+70%	6.98	24.62	24.57	**29.05**	6.74	26.15	26.36	**32.28**
15+70%	6.97	22.83	24.20	**28.11**	6.72	23.86	25.85	**31.23**
	"Cameraman"				"Boat"			
	Noisy	AMF	TVL1	AOP	Noisy	AMF	TVL1	AOP
0+30%	10.32	33.62	30.43	**38.43**	10.70	30.16	27.64	**33.32**
5+30%	10.28	31.34	30.09	**35.33**	10.69	30.27	27.84	**33.39**
10+30%	10.25	28.15	29.35	**32.47**	10.69	30.34	27.70	**33.06**
15+30%	10.21	25.39	28.40	**30.45**	10.68	30.24	27.70	**32.96**
0+50%	8.08	29.78	26.80	**34.58**	8.49	27.27	25.00	**30.54**
5+50%	8.08	28.35	26.55	**32.78**	8.51	27.20	25.02	**30.49**
10+50%	8.09	26.38	26.16	**30.57**	8.48	27.24	25.24	**30.19**
15+50%	8.04	24.29	26.09	**29.08**	8.49	27.12	25.02	**30.12**
0+70%	6.62	25.73	23.20	**29.85**	7.02	24.33	22.42	**27.20**
5+70%	6.62	25.22	23.21	**29.06**	7.02	24.19	22.35	**27.14**
10+70%	6.60	24.02	23.01	**28.04**	7.02	24.18	22.42	**27.08**
15+70%	6.60	22.48	22.66	**26.96**	7.01	24.23	22.37	**26.97**

Table 5.1: PSNR(dB) for denoising results of different algorithms for noisy images corrupted by salt-and-pepper impulse noise and mixed Gaussian impulse noise. σ is the standard deviation for the Gaussian noise and s is the level of salt-and-pepper impulse noise.

pixels by salt-and-pepper impulse noise using AMF is very accurate, and Λ will not change too much in the iterations, thus the performance of AOP will be similar to the two-stage approach by first detecting the damaged pixels by AMF and then solve the total variation image inpainting problem, which is just the first iteration of AOP.

For random-valued impulse noise removal, it is more difficult to detect the corrupted pixels because they can take any value between d_{min} and d_{max}. ACWMF and BOLD are

Figure 5.2: Denoising results of images contaminated by both Gaussian noise and salt-and-pepper impulse noise with $\sigma = 10$ and $s = 30\%$. Top row: noisy images; Second row: the results restored by AMF; Third row: the results restored by TVL1; Bottom row: the results restored by total variation blind inpainting using AOP.

used in two-stage approaches for detecting the damaged pixels [4, 95]. In this experiment, we will compare AOP with ACWMF, TVL1, and two-stage approaches (TS) for random-valued impulse noise removal. The two-stage approach we used here is just one step of AOP, and the parameter for second stage (total variation image inpainting) is also tuned to achieve

the best quality of the restoration images.

Again four test images are corrupted by Gaussian noise of zero mean and standard deviation ($\sigma = 5, 10, 15$), then we add random-valued impulse noise with different levels ($s = 25\%, 40\%$) onto the test images, with or without Gaussian noise. The PSNR values of the results from these four methods are summarized in Table 5.2.

Random-Valued Impulse Noise										
$\sigma + s$	"Lena"					"House"				
	Noisy	acwmf	TVL1	TS	BI	Noisy	acwmf	TVL1	TS	BI
0+25%	15.25	30.53	31.75	32.65	**33.74**	14.71	31.50	36.88	35.55	**42.11**
5+25%	15.25	30.51	31.88	32.72	**33.85**	14.68	31.35	36.95	35.56	**41.58**
10+25%	15.26	30.44	31.87	32.65	**33.66**	14.71	31.38	36.36	35.55	**41.61**
15+25%	15.33	30.62	31.88	32.81	**33.89**	14.65	31.28	36.81	35.43	**41.49**
0+40%	13.27	24.62	29.22	28.44	**30.77**	12.65	23.90	32.77	30.07	**37.39**
5+40%	13.23	24.61	29.20	28.51	**30.51**	12.63	23.90	32.49	30.05	**36.86**
10+40%	13.22	24.31	28.94	28.15	**30.34**	12.62	23.82	32.44	29.97	**36.71**
15+40%	13.25	24.60	29.10	28.43	**30.75**	12.65	23.90	32.50	30.07	**36.66**
$\sigma + s$	"Cameraman"					"Boat"				
	Noisy	acwmf	TVL1	TS	BI	Noisy	acwmf	TVL1	TS	BI
0+25%	14.48	28.93	31.24	31.75	**33.16**	15.29	28.18	28.63	29.37	**29.60**
5+25%	14.41	28.77	31.26	31.72	**33.21**	15.32	28.16	28.62	29.42	**29.67**
10+25%	14.52	28.99	31.32	31.86	**33.26**	15.35	28.16	28.76	29.48	**29.78**
15+25%	14.47	28.98	31.24	31.63	**32.55**	15.35	27.99	28.56	29.19	**29.47**
0+40%	12.37	22.26	27.36	26.51	**29.16**	13.30	23.56	26.16	26.03	**27.12**
5+40%	12.46	22.50	27.67	26.70	**29.26**	13.26	23.40	26.11	25.96	**26.99**
10+40%	12.39	22.42	27.75	26.74	**29.21**	13.31	23.42	26.22	25.96	**26.99**
15+40%	12.43	22.42	27.89	26.69	**29.08**	13.28	23.47	26.18	26.03	**27.02**

Table 5.2: PSNR(dB) for denoising results of different algorithms for noisy images corrupted by random-valued impulse noise and mixed Gaussian impulse noise. σ is the standard deviation for the Gaussian noise and s is the level of random-valued impulse noise.

From Table 5.2, we can see that for random-valued impulse noise, the results from total variation blind inpainting using AOP are better than those by other methods for all noise levels. The comparison of ACWMF and TVL1 shows that TVL1 outperforms ACWMF for all noise levels tested, because ACWMF misses quite a lot of real noise and false-hits some noise-free pixels. TVL1 has better performance than two-stage approach for the cases when noise level is high ($s = 40\%$ in the numerical experiments), because the accuracy of detecting corrupted pixels by random-valued impulse noise using ACWMF is very low when the noise

level is high. The accuracy of detecting corrupted pixels can be improved by our method via iteratively updating the binary matrix Λ, as shown in the comparison.

The visual comparison of some results is shown in Figure 5.3. We can see noisy artifacts in the background of the images obtained by ACWMF, and the images obtained by TVL1 are blurred with some lost details. Images restored by total variation blind inpainting are smooth in flat regions of the background and the details are kept.

Both experiments show that our method by iteratively updating the inpainting region and performing image inpainting provides better results in identifying the outliers and recovering damaged pixels. For salt-and-pepper impulse noise, because there are very accurate methods for detecting the corrupted pixels such as AMF, our method has similar performance as two-stage approaches. However, for random-valued impulse noise, there is no method that can detect corrupted pixels accurately, especially when the noise level is high. Our method by iteratively updating the corrupted pixels is a better choice.

When H is a blurring operator instead of the identity, Ada_IFASDA (Ada_Fast_IFASDA) are proposed for deblurring of images corrupted by mixed Gaussian and random-valued impulse noise by iteratively updating the inpainting region Ω_1^c . The last experiment is to compare the performance of Ada_IFASDA (Ada_Fast_IFASDA) and IFASDA (Fast_IFASDA), for deblurring of images corrupted by Gaussian noise of mean zero and random-valued impulse noise. Four test images are blurred by the kernel `fspecial('disk',3)`, and corrupted by Gaussian noise of mean zero and standard deviation $\sigma = 5$. Several noise levels ($s = 25\%, 40\%, 55\%$ for random set, and $s = 25.32\%, 31.40\%, 36.83\%$ for specified set) are added into those blurry and noisy images.

The quantitative qualities (PSNR values) of restored images are listed in Table 5.3. From Table 5.3, the performances of Ada_IFASDA and Ada_Fast_IFASDA are better than those of IFASDA and Fast_IFASDA respectively. The restored images of Fast_IFASDA and Ada_Fast_IFASDA for noise levels $s = 55\%$ and $s = 36.83\%$ are shown in Figures 5.4 and 5.5 respectively. From the results, a better estimate for the damaged pixels is very crucial to two-stage methods (e.g. IFASDA), especially when the noise level is high. The results show

Figure 5.3: Denoising results of images contaminated by both Gaussian noise and random-valued impulse noise with $\sigma = 10$ and $s = 25\%$. Top row: noisy images; Second row: the results restored by ACWMF; Third row: the results restored by TVL1; Bottom row: the results restored by total variation blind inpainting using AOP.

the advantage of simultaneously detecting damaged pixels and restoring images.

Furthermore, we compared the damaged pixels detected by ACWMF and obtained from Ada_IFASDA in Figure 5.6 for the cameraman image. For the first case where the damaged pixels are chosen randomly ($s = 40\%$), the set obtained from our method is also random and

Algorithn	"Lena"		"Goldhill"		"Cameraman"		"Boat"		Case (s)
	PSNR	Sec	PSNR	Sec	PSNR	Sec	PSNR	Sec	
	Random-Valued Impulse Noise at Random Set								
IFASDA	27.20	49	26.23	85	**24.71**	51	26.59	44	25%
Ada_IFASDA	**27.59**	83	**26.34**	52	24.70	45	**26.99**	84	
Fast_IFASDA	27.43	49	26.23	44	25.03	54	26.83	44	
Ada_Fast_IFASDA	**27.61**	54	**26.37**	48	**25.21**	82	**27.03**	44	
IFASDA	27.02	82	26.00	51	24.27	45	26.38	87	40%
Ada_IFASDA	**27.30**	64	**26.17**	49	**24.47**	45	**26.62**	68	
Fast_IFASDA	27.04	52	26.03	46	24.63	54	26.41	44	
Ada_Fast_IFASDA	**27.37**	68	**26.21**	55	**24.90**	81	**26.66**	48	
IFASDA	24.44	59	25.01	43	22.65	43	25.24	48	55%
Ada_IFASDA	**26.33**	76	**25.69**	47	**23.71**	48	**26.01**	55	
Fast_IFASDA	24.15	80	24.99	81	22.70	80	25.32	69	
Ada_Fast_IFASDA	**26.15**	80	**25.85**	81	**24.52**	82	**26.20**	79	
	Random-Valued Impulse Noise at Specific Set								
IFASDA	26.27	40	25.44	39	23.99	40	26.10	41	25.32%
Ada_IFASDA	**26.97**	43	**25.90**	39	**24.45**	42	**26.46**	42	
Fast_IFASDA	26.57	59	25.62	58	24.55	61	26.36	59	
Ada_Fast_IFASDA	**27.30**	59	**26.25**	60	**25.03**	60	**26.81**	54	
IFASDA	24.91	42	24.66	40	23.19	42	25.27	40	31.40%
Ada_IFASDA	**26.32**	43	**25.70**	44	**24.36**	46	**26.00**	40	
Fast_IFASDA	24.92	58	24.93	59	23.54	59	25.68	59	
Ada_Fast_IFASDA	**27.00**	59	**26.01**	59	**24.76**	61	**26.50**	59	
IFASDA	22.77	44	23.13	43	21.50	42	23.82	42	36.83%
Ada_IFASDA	**25.69**	47	**24.95**	44	**23.79**	47	**25.65**	46	
Fast_IFASDA	17.73	58	22.82	58	19.11	58	23.77	58	
Ada_Fast_IFASDA	**26.47**	59	**25.60**	58	**24.44**	59	**26.03**	59	

Table 5.3: PSNR(dB) and CPU computing time (seconds) for deblurred results of different algorithms for blurred images corrupted by random-valued impulse noise plus Gaussian noise. The images are blurred by the blurring kernel `fspecial('disk',3)`.

does not contain any information from the image, while the set detected by ACWMF still has some information. For the second case where the set of damaged pixels is not random ($s = 31.40\%$), the set obtained by our method is still better than the set from ACWMF.

In the modified algorithms, parameter L plays an important role in the results of restored images. If we know the number of damaged pixels (pixels in ${\Omega_1}^c$) $|{\Omega_1}^c|$, then L can be chosen as the exact number of damaged pixels. However the difference between Hu and f may be very small at some damaged pixels and these pixels may not be considered as damaged. Thus, a number slightly less than $|{\Omega_1}^c|$ is better. To find a best rate for $L/|{\Omega_1}^c|$, we test on

Figure 5.4: The restored results of images blurred by `fspecial('disk',3)` and corrupted by random-valued impulse noise (level $s = 55\%$) at random set and Gaussian noise (STD $\sigma = 5$). Top row: blurry and noisy images; Middle row: the results restored by Fast_IFASDA; Bottom row: the results restored Ada_Fast_IFASDA.

the cameraman image when the level of noise is 40%, and the results are shown in Figure 5.7. For both methods, we can obtain image with highest PSNR when the rate is 0.85, and this number is chosen for all previous experiments.

5.7 Conclusion

This chapter presents two general algorithms based on blind inpainting and ℓ_0 minimization for removing impulse noise. The difference is in the treatment of the ℓ_0 term: I) the ℓ_0 term is put in the objective function, II) the ℓ_0 term is in the constraint. Both problems can be solved by iteratively restoring the images and identifying the damaged pixels. The

Figure 5.5: The restored results of images blurred by `fspecial('disk',3)` and corrupted by random-valued impulse noise (level $s = 36.83\%$) at specific set and Gaussian noise (STD $\sigma = 5$). Top row: blurry and noisy images; Middle row: the results restored by Fast_IFASDA; Bottom row: the results restored Ada_Fast_IFASDA.

performance of these two methods is similar, and the connection between these two methods is shown. It is also shown in the experiments that the proposed methods performed better than two-stage approaches and TVL1. This simple idea can also be applied to other cases where the noise model is not Gaussian.

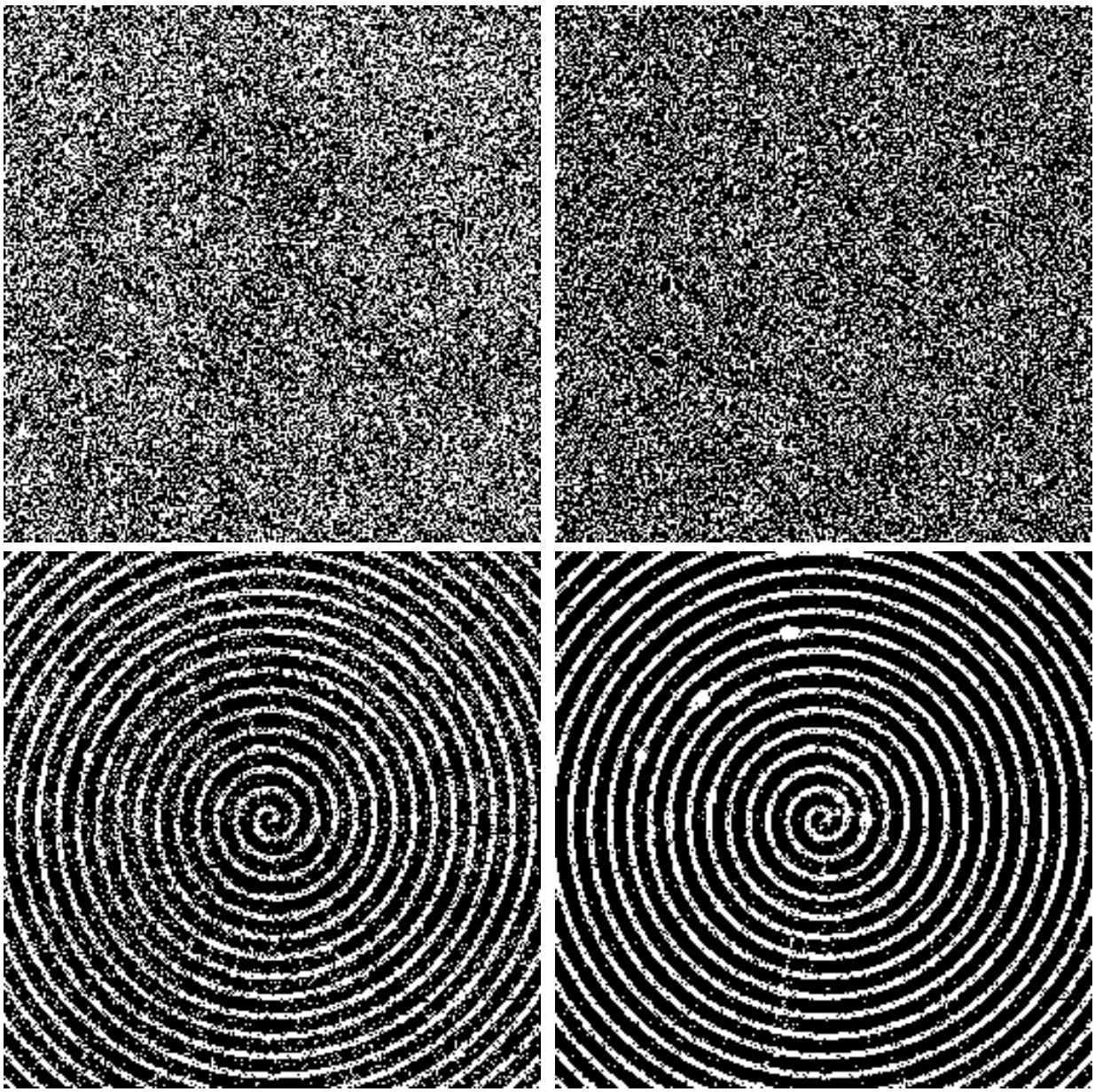


Figure 5.6: The damaged pixels detected by ACWMF and Ada_IFASDA. Left column: the set obtained by ACWMF; Right column: the set obtained by Ada_IFASDA

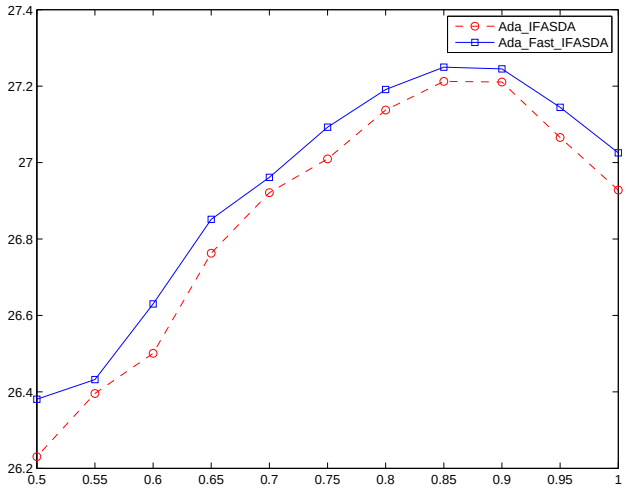


Figure 5.7: PSNR values for different $K/|\Lambda^c|$ for cameraman image when the level of random-valued impulse noise is 40%.

References

[1] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society Series B*, vol. 39, pp. 1–38, 1977. 2, 11, 14, 39, 42

[2] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms", *Physics D*, vol. 60, pp. 259–268, 1992. 2, 80

[3] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001. 2, 80, 96

[4] T. Chen and H. R. Wu, "Adaptive impulse dectection using center-weighted median filters", *IEEE Signal Processing Letters*, vol. 8, no. 1, pp. 1–3, 2001. 2, 63, 80, 82, 98

[5] P. T. Boufounos and R. G. Baraniuk, "One-bit compressive sensing", in *Conference on Information Sciences and Systems (CISS)*, Princeton, March 2008. 2, 62

[6] L. Shepp and B. Logan, "The Fourier reconstruction of a head section", *IEEE Transaction on Nuclear Science*, vol. 21, pp. 21–34, 1974. 5

[7] A. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, Society of Industrial and Applied Mathematics, 2001. 5

[8] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984. 6

[9] U. Grenander, "Tutorial in pattern theory", Lecture Notes Volume, Division of Applied Mathematics, Brown University, 1984. 6

[10] J. A. Conchello and J. G. McNally, "Fast regularization technique for expectation maximization algorithm for optical sectioning microscopy", in *Proceedings of SPIE Symposium on Electronic Imaging Science and Technology*, 1996, vol. 2655, pp. 199–208. 7

[11] J. Markham and J. A. Conchello, "Fast maximum-likelihood image-restoration algorithms for three-dimensional fluorescence microscopy", *Journal of the Optical Society America A*, vol. 18, pp. 1052–1071, 2001. 7

[12] D. Zhu, M. Razaz, and R. Lee, "Adaptive penalty likelihood for reconstruction of multi-dimensional confocal microscopy images", *Computerized Medical Imaging and Graphics*, vol. 29, pp. 319–331, 2005. 7

[13] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms", *Phys D,*, vol. 60, pp. 259–268, 1992. 7

[14] N. Dey, L. Blanc-Feraud, C. Zimmer, P. Roux, Z. Kam, J. C. Olivo-Marin, and J. Zerubia, "Richardson-Lucy algorithm with total variation regularization for 3D confocal microscope deconvolution", *Microscopy Research and Technique*, vol. 69, pp. 260–266, 2006. 7

[15] C. Brune, A. Sawatzky, F. Wubbeling, T. Kosters, and M. Burger, "An analytical view on EM-TV based methods for inverse problems with Poisson noise", Preprint, University of Münster, Aug. 2009. 7, 8

[16] M. Yan and L. A. Vese, "Expectation maximization and total variation based model for computed tomography reconstruction from undersampled data", in *Proceedings of SPIE Medical Imaging: Physics of Medical Imaging*, 2011, vol. 7961, p. 79612X. 7, 31

[17] M. Yan, "Em-type algorithms for image reconstruction with background emission and poisson noise", in *ISVC (1)*, G. Bebis, R. D. Boyle, B. Parvin, D. Koracin, S. Wang, K. Kim, B. Benes, K. Moreland, C. W. Borst, S. DiVerdi, Y.-J. Chiang, and J. Ming, Eds. 2011, vol. 6938 of *Lecture Notes in Computer Science*, pp. 33–42, Springer. 7

[18] S. Joshi and M. I. Miller, "Maximum a posteriori estimation with Good's roughness for optical sectioning microscopy", *Journal of the Optical Society of America A*, vol. 10, pp. 1078–1085, 1993. 7

[19] A. N. Tychonoff and V. Y. Arsenin, *Solution of Ill-posed Problems*, Washington, Winston, 1977. 7

[20] T. Le, R. Chartrand, and T. J. Asaki, "A variational approach to reconstructing images corrupted by poisson noise", *Journal of Mathematical Imaging and Vision*, vol. 27, pp. 257–263, 2007. 7, 37

[21] L. Shepp and Y. Vardi, "Maximum likelihood reconstruction for emission tomography", *IEEE Transaction on Medical Imaging*, vol. 1, pp. 113–122, 1982. 7, 10, 11, 37, 38, 39

[22] R. Acar and C. R. Vogel, "Analysis of bounded variation penalty methods for ill-posed problems", *Inverse Problem*, vol. 10, pp. 1217–1229, 1994. 8

[23] W. H. Richardson, "Bayesian-based iterative method of image restoration", *Journal of the Optical Society America*, vol. 62, pp. 55–59, 1972. 10, 38

[24] L. B. Lucy, "An iterative technique for the rectification of observed distributions", *Astronomical Journal*, vol. 79, pp. 745–754, 1974. 10, 38

[25] H. Kuhn and A. Tucker, "Nonlinear programming", in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, 1951, pp. 481–492. 11, 39

[26] W. Karush, "Minima of functions of several variables with inequalities as side constraints", Master's thesis, Department of Mathematics, University of Chicago, Chicago, Illinois, 1939. 11, 39

[27] H. Hurwitz, "Entropy reduction in Bayesian analysis of measurements", *Physics Review A*, vol. 12, pp. 698–706, 1975. 12, 40

[28] E. Levitan and G. T. Herman, "A maximum a posteriori probability expectation maximization algorithm for image reconstruction in emission tomography", *IEEE Transactions on Medial Imaging*, vol. 6, pp. 185–192, 1987. 12, 40

[29] R. Gordon, R. Bender, and G. Herman, "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography", *Journal of Theoretical Biology*, vol. 29, pp. 471–481, 1970. 20

[30] G. Herman, *Fundamentals of Computerized Tomography: Image Reconstruction From Projection*, Springer, 2009. 20, 29

[31] A. Andersen, "Algebraic reconstruction in CT from limited views", *IEEE Transactions on Medical Imaging*, vol. 8, pp. 50–55, 1989. 20, 21

[32] A. Andersen and A. Kak, "Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm", *Ultrasonic Imaging*, vol. 6, pp. 81–94, 1984. 20

[33] C. Atkinson and J. Soria, "An efficient simultaneous reconstruction technique for tomographic particle image velocimetry", *Experiments in Fluids*, vol. 47, pp. 553–568, 2009. 20

[34] Y. Pan, R. Whitaker, A. Cheryauka, and D. Ferguson, "Feasibility of gpu-assisted iterative image reconstruction for mobile c-arm ct", *Proceedings of International Society for Photonics and Optonics (SPIE)*, vol. 7258, pp. 72585J, 2009. 20

[35] H. Yu and G. Wang, "SART-type image reconstruction from a limited number of projections with the sparsity constraint", *International Journal of Biomedical Imaging*, p. 934847, 2010. 20

[36] M. Jiang and G. Wang, "Convergence studies on iterative algorithms for image reconstruction", *IEEE Transactions on Medical Imaging*, vol. 22, pp. 569–579, 2003. 20, 27

[37] M. Jiang and G. Wang, "Convergence of the simultaneous algebraic reconstruction technique (SART)", *IEEE Transaction on Image Processing*, vol. 12, pp. 957–961, 2003. 20, 27

[38] J. Wang and Y. Zheng, "On the convergence of generalized simultaneous iterative reconstruction algorithms", *IEEE Transaction on Image Processing*, vol. 16, pp. 1–6, 2007. 20

[39] Y. Censor and T. Elfving, "Block-iterative algorithms with diagonally scaled oblique projections for the linear feasibility problem", *SIAM Journal on Matrix Analysis and Applications*, vol. 24, pp. 40–58, 2002. 20, 21

[40] M. Yan, "Convergence analysis of sart: Optimization and statistics", submitted, 2011. 20, 21, 27

[41] Y. Censor, D. Gordon, and R. Gordon, "Component averaging: An efficient iterative parallel algorithm for large and sparse unstructured problems", *Parallel Computing*, vol. 27, pp. 777–808, 2001. 21

[42] R. Siddon, "Fast calculation of the exact radiological path for a three-dimensional CT array", *Medical Physics*, vol. 12, pp. 252–255, 1986. 29

[43] H. Zhao and A. Reader, "Fast ray-tracing technique to calculate line integral paths in voxel arrays", *IEEE Nuclear Science Symposium Conference Record*, pp. M11–197, 2003. 29

[44] H. Shah, "A common framework for curve evolution, segmentation and anisotropic diffusion", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 136–142, 1996. 33

[45] R. Alicandro, A. braides, and J. Shah, "Free-discontinuity problems via functionals involving the L^1-norm of the gradient and their approximation", *Interfaces and Free Boundaries*, vol. 1, pp. 17–37, 1999. 33

[46] M. Bertero, P. Boccacci, G. Desidera, and G. Vicidomini, "Image deblurring with Poisson data: from cells to galaxies", *Inverse Problem*, vol. 25, pp. 123006, 2009. 36

[47] D. G. Politte and D. L. Snyder, "Corrections for accidental coincidences and attenuation in maximum-likelihood image reconstruction for positron-emission tomography", *IEEE Transaction on Medical Imaging*, vol. 10, pp. 82–89, 1991. 36

[48] Y. Meyer, *Oscillating Patterns in Image Processing and in some Nonlinear Evolution Equations*, American Mathematical Society, 2001. 50

[49] L. Bregman, "The relaxation method for finding common points of convex sets and its application to the solution of problems in convex programming", *USSR Computational Mathematics and Mathematical Physics*, vol. 7, pp. 200–217, 1967. 50

[50] S. Osher, Y. Mao, B. Dong, and W. Yin, "Fast linearized Bregman iteration for compressed sensing and sparse denoising", *Communications in Mathematical Sciences*, vol. 8, pp. 93–111, 2010. 50

[51] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for l1-minimization with applications to compressed sensing", *Journal on Imaging Sciences*, vol. 1, pp. 143–168, 2008. 50

[52] J. Chung, J. Nagy, and D. O'Leary, "A weighted GCV method for Lanczos hybrid regularization", *Electronic Transactions on Numerical Analysis*, vol. 28, pp. 149–167, 2008. 55

[53] A. F. Coskun, I. Sencan, T. W. Su, and A. Ozcan, "Lensless wide field fluorescent imaging on a chip using compressive decoding of sparse objects", *Optics Express*, vol. 18, pp. 10510–10523, 2010. 55

[54] E. Candès, "Compressive sampling", in *Int. Congress of Mathematics*, Madrid, Spain, 2006, vol. 3, pp. 1433–1452. 61

[55] E. J. Candès, J. K. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information", *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006. 61

[56] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements", *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, Aug. 2006. 61

[57] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?", *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006. 61

[58] D. L. Donoho, "Compressed sensing", *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006. 61

[59] E. Candès, "The restricted isometry property and its implications for compressed sensing", *Comptes Rendus Mathematique*, vol. 346, no. 9-10, pp. 589–592, May 2008. 61

[60] J. Z. Sun and V. K. Goyal, "Quantization for Compressed Sensing Reconstruction", in *SAMPTA'09, International Conference on Sampling Theory and Applications*, L. Fesquet and B. Torrésani, Eds., Marseille, France, 2009, p. Special session on sampling and quantization. 61

[61] W. Dai, H. V. Pham, , and O. Milenkovic, "Distortion-rate functions for quantized compressive sensing", in *IEEE Information Theory Workshop on Networking and Information Theory*, 2009. 61

[62] A. Zymnis, S. Boyd, and E. Candes, "Compressed sensing with quantized measurements", *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 149–152, 2010. 61

[63] J. N. Laska, P. T. Boufounos, M. A. Davenport, and R. G. Baraniuk, "Democracy in action: Quantization, saturation, and compressive sensing", *Applied and Computational Harmonic Analysis*, vol. 31, no. 3, pp. 429–443, Nov. 2011. 61

[64] L. Jacques, D. K. Hammond, and M.-J. Fadili, "Dequantizing compressed sensing: When oversampling and non-gaussian constraints combine.", *IEEE Transactions on Information Theory*, pp. 559–571, 2011. 61

[65] Y. Plan and R. Vershynin, "Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach", *submitted*, 2012. 61

[66] J. N. Laska and R. G. Baraniuk, "Regime change: Bit-depth versus measurement-rate in compressive sensing", *IEEE Transactions on Signal Processing*, accepted. 62

[67] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, "Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors", *submitted*, 2011. 62, 64, 66, 69, 74

[68] P. T. Boufounos, "Greedy sparse signal reconstruction from sign measurements", in *Proceedings of the 43rd Asilomar conference on Signals, systems and computers*, Piscataway, NJ, USA, 2009, Asilomar'09, pp. 1305–1309, IEEE Press. 62

[69] A. Gupta, R. Nowak, and B. Recht, "Sample complexity for 1-bit compressed sensing and sparse classification", in *Proceedings of the IEEE International Symposium on Information Theory*, 2010. 62

[70] J. N. Laska, Z. Wen, W. Yin, and R. G. Baraniuk, "Trust, but verify: Fast and accurate signal recovery from 1-bit compressive measurements", *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5289–5301, 2011. 62

[71] Y. Plan and R. Vershynin, "One-bit compressed sensing by linear programming", *submitted*, 2011. 62

[72] T. Zhou and D. Tao, "Hamming compressed sensing", *submitted*, 2011. 62

[73] M. Yan, "Restoration of images corrupted by impulse noise and mixed Gaussian impulse noise using blind inpainting", *submitted*, 2011. 63, 64

[74] J. N. Laska, M. A. Davenport, and R. G. Baraniuk, "Exact signal recovery from sparsely corrupted measurements through the pursuit of justice", in *Proceedings of the 43rd Asilomar conference on Signals, systems and computers*, Piscataway, NJ, USA, 2009, Asilomar'09, pp. 1556–1560, IEEE Press. 65

[75] C. Studer, P. Kuppinger, G. Pope, and H. Bölcskei, "Recovery of sparsely corrupted signals", *IEEE Transactions on Information Theory*, vol. 58, pp. 3115–3130, 2012. 65

[76] A. C. Bovik, *Handbook of Image and Video Processing (Communications, Networking and Multimedia)*, Academic Press, Inc., Orlando, FL, USA, 2005. 79

[77] M. Nikolova, "A variational approach to remove outliers and impulse noise", *Journal of Mathematical Imaging and Vision*, vol. 20, pp. 99–120, 2004. 80

[78] L. Bar, N. A. Sochen, and N. Kiryati, "Image deblurring in the presence of salt-and-pepper noise", in *Scale-Space*, R. Kimmel, N. A. Sochen, and J. Weickert, Eds. 2005, vol. 3459 of *Lecture Notes in Computer Science*, pp. 107–118, Springer. 80

[79] L. Bar, N. Kiryati, and N. Sochen, "Image deblurring in the presence of impulsive noise", *International Journal of Computer Vision*, vol. 70, pp. 279–298, 2006. 80

[80] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing", *Multiscale Modeling & Simulation*, vol. 7, no. 3, pp. 1005–1028, 2008. 80

[81] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*, V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York,, 1977. 80

[82] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 3, pp. 367–383, 1992. 80

[83] L. I. Rudin and S. Osher, "Total variation based image restoration with free local constraints", in *Proceedings of 1st International Conference on Image Processing*. 1994, vol. 1, pp. 31–35, IEEE Comput. Soc. Press. 80

[84] J. Cai, R. H. Chan, and Z. Shen, "A framelet-based image inpainting algorithm", *Applied and Computational Harmonic Analysis*, vol. 24, pp. 131–149, 2008. 80, 89

[85] Y. Li, L. Shen, D. Dai, and B. Suter, "Framelet algorithms for de-blurring images corrupted by impulse plus Gaussian noise", *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1822–1837, 2011. 80, 82, 89

[86] R. Chan, C. Hu, and M. Nikolova, "An iterative procedure for removing random-valued impulse noise", *IEEE Signal Processing Letters*, vol. 11, no. 12, pp. 921–924, 2004. 80

[87] R. H. Chan, C.-W. Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization", *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1479–1485, 2005. 80

[88] J. Cai, R. H. Chan, and M. Nikolova, "Two-phase approach for deblurring images corrupted by impulse plus Gaussian noise", *Inverse Problem and Imaging*, vol. 2, pp. 187–204, 2008. 80, 82

[89] J. Cai, R. H. Chan, and M. Nikolova, "Fast two-phase image deblurring under impulse noise", *Journal of Mathematical Imaging and Vision*, vol. 36, pp. 46–53, 2010. 80, 82

[90] Y. Xiao, T. Zeng, J. Yu, and M. Ng, "Restoration of images corrupted by mixed Gaussian-impulse noise via l_1-l_0 minimization", *Pattern Recognition*, vol. 44, pp. 1708–1720, 2011. 80

[91] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting", in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, SIGGRAPH '00, pp. 417–424. 80

[92] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting", in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003, vol. 2, p. 707. 80

[93] M. Bertalmío, L. A. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting", *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 882–889, 2003. 80

[94] T. F. Chan, J. Shen, and H. Zhou, "Total variation wavelet inpainting", *Journal of Mathematical Imaging and Vision*, vol. 25, pp. 107–125, 2006. 80

[95] Y. Dong, R. H. Chan, and S. Xu, "A detection statistic for random-valued impulse noise", *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 1112–1120, 2007. 80, 98

[96] B. Dong, H. Ji, J. Li, Z. Shen, and Y. Xu, "Wavelet frame based blind image inpainting", *Applied and Computational Harmonic Analysis*, vol. accepted, 2011. 81

[97] T. Goldstein and S. Osher, "The split bregman method for L1-regularized problems", *SIAM Journal on Imaging Sciences*, vol. 2, pp. 323–343, 2009. 81, 86

[98] R. Chartrand and V. Staneva, "Restricted isometry properties and nonconvex compressive sensing", *Inverse Problems*, vol. 24, no. 3, pp. 035020, 2008. 81

[99] H. Mohimani, M. Babaie-Zadeh, I. Gorodnitsky, and C. Jutten, "Sparse recovery using smoothed l0 (sl0): convergence analysis", *IEEE Trans. On Information Theory*, vol. submitted, 2011. 81

[100] Z. Lu and Y. Zhang, "Penalty Decomposition Methods for $L0$-Norm Minimization", *ArXiv preprint arXiv:1008.5372*, 2010. 81

[101] T. Chan and J. Shen, "Variational image inpainting", *Communications on Pure and Applied Mathematics*, vol. 58, pp. 579–619, 2005. 85

[102] T. F. Chan and J. Shen, "Mathematical models for local nontexture inpaintings", *SIAM Journal of Applied Mathematics*, vol. 62, no. 3, pp. 1019–1043, 2002. 86

[103] T. Chan and J. Shen, *Image Processing And Analysis: Variational, Pde, Wavelet, And Stochastic Methods*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2005. 86

[104] A. Chambolle, "An algorithm for total variation minimization and applications", *Journal of Mathematical Imaging and Vision*, vol. 20, pp. 89–97, 2004. 86

[105] M. Zhu, S. J. Wright, and T. F. Chan, "Duality-based algorithms for total-variation-regularized image restoration", *Computational Optimization and Applications*, vol. 47, pp. 377–400, 2010. 86

[106] X.-C. Tai and C. Wu, "Augmented lagrangian method, dual methods and split bregman iteration for ROF model", in *Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision*, 2009, pp. 502–513. 86

[107] C. Wu and X.-C. Tai, "Augmented lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models", *SIAM Journal on Imaging Sciences*, vol. 3, pp. 300–339, 2010. 86

[108] P. Getreuer, "tvreg v2: Variational imaging methods for denoising, deconvolution, inpainting, and segmentation", 2010. 86, 96

[109] M. Yan, Y. Yang, and S. Osher, "Robust 1-bit compressive sensing using adaptive outlier pursuit", *IEEE Transactions on Signal Processing*, accepted. 87

[110] Y. Meyer, *Oscillating patterns in image processing and nonlinear evolution equations*, vol. 22 of *University Lecture Series*, American Mathematical Society, Providence, RI, 2001, The fifteenth Dean Jacqueline B. Lewis memorial lectures. 89

[111] R. H. Chan, S. D. Riemenschneider, L. Shen, and Z. Shen, "Tight frame: an efficient way for high-resolution image reconstruction.", *Applied and Computational Harmonic Analysis*, vol. 17, no. 1, pp. 91–115, 2004. 89

[112] Y. Wang, A. Szlam, and G. Lerman, "Robust locally linear analysis with applications to image denoising and blind inpainting", *submitted*, 2011. 95

[113] H. Hwang and R. A. Haddad, "Adaptive median filters: new algorithms and results", *IEEE Transactions on Image Processing*, vol. 4, no. 4, pp. 499–502, 1995. 96

[114] S. Alliney, "A property of the minimum vectors of a regularizing functional defined by means of the absolute norm", *IEEE Transactions on Signal Processing*, vol. 45, no. 4, pp. 913–917, 1997. 96

[115] T. F. Chan and S. Esedoglu, "Aspects of total variation regularized l1 function approximation", *SIAM Journal on Applied Mathematics*, vol. 65, no. 5, pp. 1817, 2005. 96