

Model-Based X-Ray CT Image and Light Field Reconstruction Using Variable Splitting Methods

by

Hung Nien

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in the University of Michigan
2014

Doctoral Committee:

Professor Jeffrey A. Fessler, Chair
Assistant Professor Laura Balzano
Professor Anna C. Gilbert
Professor Alfred O. Hero

© Hung Nien 2014

All Rights Reserved

ACKNOWLEDGEMENTS

I could not finish this work without the help of many people. First of all, I would like to thank my advisor, Professor Jeffrey A. Fessler, for his careful guidance, constant encouragement, and the maximum freedom he gave me for developing independent thinking and research skills. Unlike most of my colleagues, I knew nothing about medical imaging and worked on a totally different research topic when I first arrived at the University of Michigan. When I could not find a supervisor for taking the qualifying exams, it was Jeff who volunteeringly served as my directed research supervisor and gave me insightful advice on my preliminary research on light field reconstruction. When I had a hard and had to quit from the PhD program since I did not have a research advisor supporting my PhD research, it was Jeff who willingly served as my research advisor and supported my following doctoral study. Without Professor Jeff Fessler's kindly help, this work would not have been possible, even from the very beginning.

Furthermore, I would like to thank Professor Laura Balzano, Professor Anna C. Gilbert, and Professor Alfred O. Hero, who served as my thesis committee and gave me enlightening suggestion and valuable feedback about my dissertation and future research plan. I really appreciate their help in sharing their deep knowledge of statistics, optimization, and machine learning, which greatly improved the quality of this thesis. I am grateful to my current and former colleagues: Michael Allison, Jang Hwan Cho, Donghwan Kim, Jung Kuk Kim, Young Song Kwon, Mai Le, Antonis Matakos, Madison McGaffin, Gopal Nataraj, Sathish Ramani, Stephen Schmitt, Hao Sun, and Daniel Weller, for their contributed ideas and discussions. I would also like to thank all my friends in the EE:Systems program (too many to be listed) for their emotional support and timely assistance, and thank the EE:Systems staff who helped me pass different stages of my doctoral study at the University of Michigan.

Finally, I would like to thank my mother and my grandparents for their unwavering support, letting me pursue my dream, and always believing in me, from the other side of the world. I would also like to thank my girlfriend, Huiying (Yanna) Yan, who always stands on my side and helps me become a better man.

This work is supported in part by NIH grant R01-HL-098686 and by an equipment donation from Intel Corporation. We thank GE Healthcare for providing sinogram data for our experiments.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ii
LIST OF FIGURES	vii
LIST OF APPENDICES	xi
ABSTRACT	xii
CHAPTER	
I. Introduction	1
1.1 Contributions	2
1.2 Outline	3
II. Background of X-ray CT and its reconstruction	4
2.1 Review of X-ray computed tomography	4
2.2 Model-based X-ray CT image reconstruction	6
2.2.1 Gradient-based methods	8
2.2.2 Splitting-based methods	13
III. Fast X-ray CT image reconstruction using variable splitting methods with ordered subsets	19
3.1 Non-iterative image update using a linearized AL method	19
3.1.1 Linearized AL method	20
3.1.2 OS-LALM: a splitting-based ordered-subset method	26
3.2 Fast convergence with a downward continuation approach	30
3.2.1 Second-order system analysis with a fixed ρ	31
3.2.2 Experimental results: low-dose CT	36
3.3 Efficient memory usage when considering additional splits	43
3.3.1 Split OS-LALM: OS-LALM with additional split	45
3.3.2 Memory-efficient implementation	47
3.3.3 Experimental results: sparse-view CT	50

3.4	Conclusion	52
IV.	Blind gain correction for X-ray CT image reconstruction . . .	55
4.1	Introduction	55
4.2	Method	56
4.2.1	Joint gain-image estimation for X-ray CT image re- construction	56
4.2.2	Applying prior knowledge of gain parameter to the joint gain-image estimation	57
4.2.3	Joint gain-image estimation using other optimization methods	58
4.3	Result	59
4.3.1	2D fan beam X-ray CT image reconstruction	59
4.3.2	3D axial X-ray CT image reconstruction	60
4.4	Conclusion	62
V.	Model-based light field reconstruction	65
5.1	Introduction	65
5.2	Discrete light fields	66
5.3	Light field transformations of typical camera operations	68
5.3.1	Light field transformation: refocusing	69
5.3.2	Light field transformation: zooming	70
5.3.3	Light field transformation: camera translation	71
5.4	Model-based light field reconstruction	72
5.4.1	Sparsity-regularized minimization using ADMM	73
5.4.2	Prior of the elongated light field structure	73
5.5	The back-project filter method for light field reconstruction	78
5.6	Flatland simulations	79
5.7	Conclusion	81
VI.	Conclusion and future work	83
APPENDICES	85
A.1	Introduction	86
A.2	The split Bregman method as an ADMM algorithm	87
B.1	Frequency analysis	90
B.1.1	Case I	92
B.1.2	Case II	94
B.1.3	Case III	95
B.2	Parameter selection for image restoration problems	95
B.3	Numerical experiments	96
C.1	Introduction	99

C.2 Proof of Theorem 1	100
C.3 Proof of Theorem 2	103
BIBLIOGRAPHY	111

LIST OF FIGURES

<u>Figure</u>		
3.1	Optimal asymptotic convergence rate ρ^* of a system with six distinct eigenvalues ($\mu = 0.05L, 0.1L, 0.3L, 0.7L, 0.9L$, and L).	34
3.2	Shoulder scan: cropped images (displayed from 800 to 1200 HU) from the central transaxial plane of the initial FBP image $\mathbf{x}^{(0)}$ (left), the reference reconstruction \mathbf{x}^* (center), and the reconstructed image using the proposed algorithm (OS-LALM-40-c-1) at the 30th iteration $\mathbf{x}^{(30)}$ (right).	39
3.3	Shoulder scan: cropped difference images (displayed from -30 to 30 HU) from the central transaxial plane of $\mathbf{x}^{(30)} - \mathbf{x}^*$ using OS-based algorithms.	40
3.4	Shoulder scan: RMS differences between the reconstructed image $\mathbf{x}^{(k)}$ and the reference reconstruction \mathbf{x}^* as a function of iteration using OS-based algorithms with (a) 20 subsets and (b) 40 subsets, respectively. The dotted lines show the RMS differences using the standard OS algorithm with one subset as the baseline convergence rate.	40
3.5	Shoulder scan: RMS differences between the reconstructed image $\mathbf{x}^{(k)}$ and the reference reconstruction \mathbf{x}^* as a function of iteration using the proposed algorithm with different number of FISTA iterations n (1, 2, and 5) for solving the inner constrained denoising problem. . .	41
3.6	Truncated abdomen scan: cropped images (displayed from 800 to 1200 HU) from the central transaxial plane of the initial FBP image $\mathbf{x}^{(0)}$ (left), the reference reconstruction \mathbf{x}^* (center), and the reconstructed image using the proposed algorithm (OS-LALM-20-c-1) at the 30th iteration $\mathbf{x}^{(30)}$ (right).	41
3.7	Truncated abdomen scan: cropped difference images (displayed from -30 to 30 HU) from the central transaxial plane of $\mathbf{x}^{(30)} - \mathbf{x}^*$ using OS-based algorithms.	42
3.8	Truncated abdomen scan: RMS differences between the reconstructed image $\mathbf{x}^{(k)}$ and the reference reconstruction \mathbf{x}^* as a function of iteration using OS-based algorithms with 10, 20, and 40 subsets. The dotted line shows the RMS differences using the standard OS algorithm with one subset as the baseline convergence rate.	42

3.9	GE performance phantom: cropped images (displayed from 800 to 1200 HU) from the central transaxial plane of the initial FBP image $\mathbf{x}^{(0)}$ (left), the reference reconstruction \mathbf{x}^* (center), and the reconstructed image using the proposed algorithm (OS-LALM-24-c-1) at the 30th iteration $\mathbf{x}^{(30)}$ (right).	44
3.10	GE performance phantom: cropped difference images (displayed from -30 to 30 HU) from the central transaxial plane of $\mathbf{x}^{(30)} - \mathbf{x}^*$ using the relaxed OS+momentum algorithm and the proposed algorithm.	44
3.11	GE performance phantom: RMS differences between the reconstructed image $\mathbf{x}^{(k)}$ and the reference reconstruction \mathbf{x}^* as a function of iteration using the relaxed OS+momentum algorithm and the proposed algorithm with 24 subsets. The dotted line shows the RMS differences using the standard OS algorithm with one subset as the baseline convergence rate.	45
3.12	Chest scan: cropped images (displayed from 800 to 1200 HU) from the central transaxial plane of the initial FBP image $\mathbf{x}^{(0)}$ (left), the reference reconstruction \mathbf{x}^* with $\delta = 0.1$ (center), and the converged reconstruction with $\delta = 1$ (right).	52
3.13	Chest scan: RMS differences between the reconstructed image $\mathbf{x}^{(k)}$ and the reference reconstruction \mathbf{x}^* as a function of (a) iteration and (b) time, respectively, using the proposed algorithm with $M = 5$ and different values of η	52
3.14	Chest scan: RMS differences between the reconstructed image $\mathbf{x}^{(k)}$ and the reference reconstruction \mathbf{x}^* as a function of (a) iteration and (b) time, respectively, using the proposed algorithm with $\eta = 0.05$ and different values of M	53
3.15	Chest scan: RMS differences between the reconstructed image $\mathbf{x}^{(k)}$ and the reference reconstruction \mathbf{x}^* as a function of (a) iteration and (b) time, respectively, using the OS+momentum algorithm and the proposed algorithm with $M = 5$	53
3.16	Chest scan: cropped images (displayed from 800 to 1200 HU) from the central transaxial plane of the reconstructed images $\mathbf{x}^{(100)}$ using the OS+momentum algorithm (left) and the proposed algorithm (center) after 100 iterations and the reference reconstruction \mathbf{x}^* (right).	54
4.1	2D simulation: (a) The noisy sinogram with gain fluctuations, and (b) the corresponding gain fluctuations, where the horizontal and vertical axes are the radial axis r and the projection view angle θ , respectively.	60
4.2	2D simulation: (a) The true phantom (in cm^{-1}), (b) the initial guess using the FBP reconstruction, (c) the conventional reconstruction without gain correction, (d) the proposed reconstruction with blind gain correction, and (e) the reference reconstruction from a noisy sinogram without gain fluctuations.	61

4.3	2D simulation: (a) The estimated gain parameter vector as a function of projection view angle, and (b) the RMS error of the estimated gain parameter vector versus iteration.	62
4.4	3D simulation: (a) The true phantom (in cm1), (b) the initial guess using the FBP reconstruction, (c) the conventional reconstruction without gain correction, (d) the proposed reconstruction with blind gain correction, (e) the proposed reconstruction with non-blind gain correction, and (f) the reference reconstruction from a noisy sinogram without gain fluctuations. Each subfigure shows the middle transaxial, coronal, and sagittal planes of the volume.	63
4.5	3D simulation: (a) the true gain fluctuations, (b) the estimated gain fluctuations for the blind case, and (c) the estimated gain fluctuations for the non-blind case, where the horizontal and vertical axes are the projection view angle θ and the transaxial axis z , respectively.	64
4.6	3D simulation: for each method, RMS difference between the image at the n th iteration and the converged reference reconstruction.	64
5.1	Different parameterizations of light fields.	66
5.2	Parameterization of light fields in a camera.	66
5.3	Ray diagram of a planar object with Lambertian reflectance h in a simplified camera.	67
5.4	An example of a 4D light field. (a) A 2D representation of a 4D light field $L(\mathbf{x}, \mathbf{u})$. Each block denotes a 2D slice or a sub-aperture image with sub-aperture centered at \mathbf{u} . (b) An example of a 2D slice $L(\mathbf{x}, \mathbf{u}_0)$ of the 4D light field $L(\mathbf{x}, \mathbf{u})$	68
5.5	Ray diagrams for (a) refocusing, (b) zooming, and (c) camera translation.	69
5.6	Demo of refocusing: (a) Image refocused at the first object ($z = 1$ m). (b) Image refocused at the second object ($z = 2$ m). (c) Image refocused at the third object ($z = 3$ m).	70
5.7	Demo of zooming: (a) Image captured with $\alpha = 1.0$. (b) Image captured with $\alpha = 1.1$. (c) Image captured with $\alpha = 1.2$	71
5.8	Demo of camera translation: (a) Pinhole image with pinhole centered at $\mathbf{u}_0 = (-2.5 \text{ mm}, 0.0 \text{ mm})$. (b) Pinhole image with pinhole centered at $\mathbf{u}_0 = (0.0 \text{ mm}, 0.0 \text{ mm})$. (c) Pinhole image with pinhole centered at $\mathbf{u}_0 = (2.5 \text{ mm}, 0.0 \text{ mm})$	72
5.9	Characteristic slope s_0 of practical camera.	75
5.10	Depth-invariant blur kernel.	76
5.11	Demonstration of the depth-invariant prior.	77
5.12	A scene with two planar objects at different depth.	79
5.13	2D flatland simulation: (a) The true light field of a scene in Figure 5.12. (b) The reconstructed light field using the framelet prior. (c) The reconstructed light field using the curvelet prior. (d) The reconstructed light field using the TV prior. (e) The reconstructed light field using the proposed light field prior. (f) The reconstructed light field using the BPF method.	80

5.14	2D flatland simulation: (a) The true light field of a scene with three planar objects at different depths. (b) The reconstructed light field using the proposed light field prior. (c) The reconstructed light field using the BPF method.	82
B.1	An image restoration problem instance: the true image (left), the noisy blurred image (middle), and the converged reference reconstruction (right).	97
B.2	The convergence rate curves with different parameter settings: (a) the relative error of cost value and (b) the RMS difference between the reconstructed image and the reference reconstruction as a function of the number of iterations.	98

LIST OF APPENDICES

Appendix

- A. A (primal) convergence proof of the SB method for regularized least-squares problems 86
- B. Frequency analysis and parameter selection of ADMM: the quadratic case 90
- C. Convergence analyses of the inexact linearized AL method 99

ABSTRACT

Model-Based X-Ray CT Image and Light Field Reconstruction Using
Variable Splitting Methods

by
Hung Nien

Chair: Jeffrey A. Fessler

Model-based image reconstruction (MBIR) is a powerful technique for solving ill-posed inverse problems. Compared with direct methods, it can provide better estimates from noisy measurements and from incomplete data, at the cost of much longer computation time. In this work, we focus on accelerating and applying MBIR for solving reconstruction problems, including X-ray computed tomography (CT) image reconstruction and light field reconstruction, using variable splitting based on the augmented Lagrangian (AL) methods. For X-ray CT image reconstruction, we combine the AL method and ordered subsets (OS), a well-known technique in the medical imaging literature for accelerating tomographic reconstruction, by considering a linearized variant of the AL method and propose a fast splitting-based ordered-subset algorithm, OS-LALM, for solving X-ray CT image reconstruction problems with penalized weighted least-squares (PWLS) criterion. Practical issues such as the non-trivial parameter selection of AL methods and remarkable memory overhead when considering the finite difference image variable splitting are carefully studied, and several variants of the proposed algorithm are investigated for solving practical model-based X-ray CT image reconstruction problems. Experimental results show that the proposed algorithm significantly accelerates the convergence of X-ray CT image reconstruction with negligible overhead and greatly reduces the noise-like OS artifacts in the reconstructed image when using many subsets for OS acceleration. For light field reconstruction, considering decomposing the camera imaging process into a linear convolution and a non-linear slicing operations for faster forward projection, we propose to reconstruct light field from a sequence of photos taken with

different focus settings, i.e., a focal stack, using an alternating direction method of multipliers (ADMM). To improve the quality of the reconstructed light field, we also propose a signal-independent sparsifying transform by considering the elongated structure of light fields. Flatland simulation results show that our proposed sparse light field prior produces high resolution light field with fine details compared with other existing sparse priors for natural images.

CHAPTER I

Introduction

Inverting a measurement process to recover a phenomenon of interest from noisy measurements is of fundamental interest in science and mathematics. This problem is known as the inverse problem. It is central to any context where a phenomenon we are truly interested in cannot be observed directly. As an example, and one focus of this thesis, consider X-ray computed tomography. In this application, X-ray photons with known intensity emitting from known locations are transmitted through the patient to image what is inside the body in a non-invasive way. The measurements, however, are not direct measurements of the body's internal structures but instead measured attenuation properties that indicate the density distribution over different parts of the body. In this case, the inverse problem refers to reconstructing internal structures of the human body from the indirect X-ray measurements. Unfortunately, for most cases, it is impossible to solve the inverse problem exactly and directly, either because the inverse problem itself is highly ill-posed, or because the measurements are severely corrupted by noise.

Model-based image reconstruction (MBIR) methods (also known as statistical image reconstruction methods) are a collection of techniques that solve inverse problems iteratively to obtain better reconstructions at the cost of longer computation time. There are five key components to MBIR methods [1]: (1) the object model that describes the unknown continuous object using finitely many unknown coefficients of a certain basis; (2) the system model that relates the unknown coefficients to the ideal measurements assuming that there is no error in the measurement process; (3) the statistical model that measures how the observations deviate from the ideal measurements in the noisy environment; (4) the cost function that evaluates the goodness of fit between the estimate and the statistical model together with the prior knowledge of the object; and (5) an algorithm that minimizes the cost function iteratively to find a better estimate of the unknown coefficients.

It has been shown in the literature [2–4] that MBIR methods can provide better estimates from noisy measurements and from incomplete data, comparing to the conventional but widely used filtered back-projection (FBP) method in X-ray CT image reconstruction. This is extremely useful because it implies that we can acquire a CT scan with relatively lower dose but have even better reconstructed image quality. The growing concern about radiation dose from CT scans comes from the increased use of CT procedures. In the past three decades, the average American’s dose from medical exposure (not including radiotherapy) has increased from 0.54 mSv in 1982 to 3.0 mSv in 2006, where CT procedures account for about half of the collective dose from all medical procedures [5]. Compared with the natural background yearly dose of 3.6 mSv, the standard radiation dose used currently can increase the possible risk of cancers, especially for body screening with multiple scans. Veo, the first commercial product using MBIR methods recently introduced by GE, demonstrates significantly improved image quality under sub-mSv dose level in clinical scans [6]. However, the much longer computation time (estimated from 20 to 80 minutes) still restrains the applicability of MBIR methods in practice.

1.1 Contributions

In this work, we focus on accelerating and applying MBIR for solving reconstruction problems, including X-ray CT image reconstruction and light field reconstruction, using an optimization technique called variable splitting based on the augmented Lagrangian (AL) methods. For X-ray CT image reconstruction, we propose to accelerate the existing splitting-based X-ray CT image reconstruction algorithms [7–9] using ordered subsets (OS) [10] and a proposed downward continuation approach. Experimental results show that the proposed algorithm converges much faster than the existing splitting- and OS-based algorithms, illustrating the efficiency of our proposed combination and continuation approach. We also propose a new penalized weighted least-squares (PWLS) formulation for model-based X-ray CT image reconstruction, which automatically handles the model mismatch due to the incorrect calibration of reference channels in a CT scan. The new PWLS formulation is applicable to all existing X-ray CT image reconstruction algorithms with just a mild modification but significantly removes the visible shading artifacts in the reconstructed images. For light field reconstruction, we propose to decompose the imaging process of a camera into a linear convolution and a non-linear slicing operation for faster forward projection, and then we can solve the light field reconstruction problem using a splitting-

based algorithm with non-iterative updates. We also propose a signal-independent sparse light field prior that considers the elongated structure of light fields, and flat-land simulation results show that our proposed sparse light field prior produces high resolution light field with fine details compared with other existing sparse priors for natural images.

1.2 Outline

This thesis is organized as follows. In Chapter II, a brief review of the background of X-ray CT and its reconstruction is presented. In Chapter III, we consider accelerating splitting-based X-ray CT image reconstruction algorithms using OS. In Chapter IV, a new variational formulation that takes gain correction of CT scans into account in model-based X-ray CT image reconstruction is proposed. In Chapter V, we use MBIR methods to reconstruct light field from a focal stack. Finally, Chapter VI draws a conclusion and outlines the future work.

CHAPTER II

Background of X-ray CT and its reconstruction

2.1 Review of X-ray computed tomography

Unlike conventional single-shot projection radiography in which X-ray photons are transmitted through the human body to view its internal structure, X-ray computed tomography (CT) is a non-invasive imaging technique that combines a series of X-ray projections of an object taken from many different angles and computer processing (i.e., reconstruction methods) to create tomographic (i.e., cross-sectional) images of the spatial attenuation distribution (e.g., the bones and soft tissues) inside the object. It is incredibly important and used for diagnostic purposes in various disciplines in the medical community since its introduction into clinical practice in the 1970s [11]. It has more recently been used for preventive medicine or screening for disease. For example, CT screening of lungs is used to detect small nodules or tumors in the lung parenchyma that are usually not shown in conventional chest X-rays in an early stage.

A typical (transmission) CT scanner consists of an X-ray tube, a rotating gantry, a table, and a detector with one or more detector cells. There are various CT geometries (cone-beam vs. multi-slice fan-beam) and scan trajectories (axial vs. spiral/helical) in modern CT scanners. However, regardless of the different geometries and trajectories, the physics of X-ray CT measurement is governed by the Beer-Lambert law, which states that there is a logarithmic dependence between the transmission (i.e., the ratio of the exit to the incident intensity) of X-ray photons through the object and the integral of the (linear) attenuation coefficient of the substance along the path that X-ray photons travel through the object. That is,

$$I_{\text{out}}(\mathcal{E}) = I_{\text{in}}(\mathcal{E}) \exp\left(-\int_L \mu(x, y, z; \mathcal{E}) \, d\ell\right), \quad (2.1)$$

where I_{out} and I_{in} are the energy-dependent exit and incident X-ray intensity, μ is

the spatially-varying energy-dependent attenuation coefficient of the substance, and L denotes the path that X-ray photons travel through the object. Therefore, for a ray L_i with infinitesimal width, the mean projection measurement can be expressed as:

$$\mathbb{E}[Y_i] \propto \int I_{\text{in}}(\mathcal{E}) \exp\left(-\int_{L_i} \mu(x, y, z; \mathcal{E}) \, d\ell\right) \, d\mathcal{E} + r_i, \quad (2.2)$$

where Y_i denotes the measurement for the i th ray of the incident spectrum, and r_i accounts for the background signals such as Compton scatter, dark current, and read-out noise. Consider a monoenergetic X-ray source and assume no background signals for simplicity. The post-logarithmic projection measurement y_i (i.e., sinogram in CT terminology) can be represented as a line integral of the attenuation coefficient (up to some scaling factor that is determined by the reference projection measurement).

The image reconstruction problem for X-ray CT is to estimate the attenuation coefficient μ from the measurement $\mathbf{y} \triangleq [y_1, \dots, y_{n_d}]'$, where n_d denotes the total number of measurements. However, reconstructing an unknown continuous attenuation coefficient from a finite number of measurements is highly ill-posed and intractable in general. To make the reconstruction problem tractable, we usually parameterize the unknown continuous object using a finitely many unknown coefficients x_1, \dots, x_{n_p} of a certain basis. Then, the line integral of the unknown continuous attenuation coefficient can be approximated as a linear combination of x_1, \dots, x_{n_p} . That is,

$$y_i = \sum_{j=1}^{n_p} a_{ij} x_j + \varepsilon_i, \quad (2.3)$$

where a_{ij} defines the system matrix (or the projection matrix) \mathbf{A} that serves as the forward model of X-ray CT, and ε_i denotes the additive noise for the i th measurement. Note that although the projection matrix \mathbf{A} in X-ray CT is usually sparse (but highly shift-variant), it is still too large to store due to the large problem size, and the matrix-vector multiplication by \mathbf{A} (forward projection) and \mathbf{A}' (back-projection) should be computed on the fly (and usually very slow without other hardware acceleration [12]). Finally, the system equation (2.3) can be written in a more compact form as:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\varepsilon}, \quad (2.4)$$

where $\mathbf{x} \triangleq [x_1, \dots, x_{n_p}]'$ and $\boldsymbol{\varepsilon} \triangleq [\varepsilon_1, \dots, \varepsilon_{n_d}]'$, and X-ray CT image reconstruction refers to estimating the unknown coefficient vector \mathbf{x} from the noisy post-logarithmic measurement \mathbf{y} .

The conventional X-ray CT image reconstruction relies on analytical (i.e., non-iterative) image reconstruction methods, and the mathematical theory behind those analytical image reconstruction methods is the Radon transform [13], which relates a continuous function to the collection of its projections in the frequency domain by the Fourier transform. The most widely used analytical image reconstruction method is called the filtered back-projection (FBP) method, or the FDK method [14] for 3D axial cone-beam CT in particular. The method is pretty mature nowadays, and one can reconstruct an image volume with typical image size in a minute with modern computer architecture.

Although the analytical reconstruction methods are simple and fast these days, there are still several limitations that impair their performance due to somewhat unrealistic simplified models when developing those methods. For instance, analytical formulations usually assume continuous measurements and provide integral-form solutions, and the sampling issues are treated by discretizing these solutions after the fact. Furthermore, analytical methods generally ignore measurement noise in the problem formulation and solve the noise-related problems by post-processing filters after the image is reconstructed. When the radiation dose is medium or high, the fast approximate solution by analytical methods (such as the FBP method) is tolerable; however, when the radiation dose is low, analytical methods might amplify the noise and degrade reconstructed image quality for medical diagnosis. Note that there is a growing concern about the radiation dose from increasingly used CT procedures because the standard radiation dose used currently can increase the potential risk of cancers. Hence, noise-robust image reconstruction from “low-dose” (i.e., low SNR) CT scans becomes one of the main issues in modern X-ray CT research. One possible way to solve this problem is to estimate the unknown coefficient vector \mathbf{x} based on a more realistic system model that takes the noise statistics into account, which usually refers to the model-based image reconstruction (MBIR) methods or the statistical image reconstruction (SIR) methods in the medical imaging literature.

2.2 Model-based X-ray CT image reconstruction

The most straightforward MBIR method for estimating image \mathbf{x} is maximum-likelihood (ML) estimation, which estimates the parameters by maximizing the likelihood (or minimizing the negative log-likelihood) of the unknown parameters. Math-

ematically, the ML estimator of the image \mathbf{x} can be represented as:

$$\hat{\mathbf{x}}_{\text{ML}} \in \arg \min_{\mathbf{x}} \bar{L}(\mathbf{x}; \mathbf{y}) , \quad (2.5)$$

where $\bar{L}(\mathbf{x}; \mathbf{y}) \triangleq -\log l(\mathbf{x}; \mathbf{y})$ denotes the negative log-likelihood of the noisy measurement \mathbf{y} and unknown image \mathbf{x} , and $l(\mathbf{x}; \mathbf{y})$ is the corresponding likelihood, which happens to be the probability mass function $p(\mathbf{y}; \mathbf{x})$ of the noisy measurement \mathbf{y} given some underlying image \mathbf{x} .

One simple and widely used noise model of X-ray CT is the Poisson model (ignoring electronic noise). In this case, the negative log-likelihood $\bar{L}(\mathbf{x}; \mathbf{y})$ is

$$\bar{L}(\mathbf{x}; \mathbf{y}) = \sum_{i=1}^{n_d} \left\{ -y_i (\log b_i - [\mathbf{A}\mathbf{x}]_i) + b_i \exp(-[\mathbf{A}\mathbf{x}]_i) \right\} , \quad (2.6)$$

where b_i is a constant determined by the reference projection measurement of the i th measurement y_i . Unfortunately, typical ML reconstruction $\hat{\mathbf{x}}_{\text{ML}}$ is extremely noisy and impossible for clinical use. Therefore, people usually add an additional regularization term (or regularizer for short) \mathbf{R} to the ML cost function for better reconstructed image quality (e.g., sharper and less noisy), leading to a maximum *a posteriori* (MAP) or penalized likelihood (PL) reconstruction. Mathematically, the MAP or PL estimator of the image \mathbf{x} can be represented as:

$$\hat{\mathbf{x}}_{\text{MAP}} \in \arg \min_{\mathbf{x}} \left\{ \bar{L}(\mathbf{x}; \mathbf{y}) + \mathbf{R}(\mathbf{x}) \right\} , \quad (2.7)$$

and we focus on a family of regularizers \mathbf{R} that penalizes the “roughness” of the reconstructed image and is defined as:

$$\mathbf{R}(\mathbf{x}) \triangleq \Phi(\mathbf{C}\mathbf{x}) = \sum_{m=1}^{n_c} \sum_{j=1}^{n_p} \omega_{\text{reg}}(j; m) \phi([\mathbf{C}_m \mathbf{x}]_j) , \quad (2.8)$$

where \mathbf{C} is a tall matrix that stacks up the first-order finite difference matrices \mathbf{C}_m for $m = 1, \dots, n_c$, ω_{reg} is a direction- and voxel-dependent weight that controls the regularization force in each direction by β_m for $m = 1, \dots, n_c$ and the spatial resolution uniformity by κ_j for $j = 1, \dots, n_p$ [15], and ϕ is a convex, even, and possibly non-smooth potential function. For smooth potential functions, define $\omega_\phi(t) \triangleq \dot{\phi}(t)/t$ as the potential weighting function of ϕ . Suppose it is non-negative, bounded above (without loss of generality, we assume $\max_t \omega_\phi(t) = 1$), and non-increasing for $t \geq 0$. This implies that $\omega_\phi(t)$ attains its maximum at $t = 0$, i.e., $\max_t \omega_\phi(t) = \omega_\phi(0) = 1$.

Furthermore, when ϕ is twice differentiable, we also have $\max_t \ddot{\phi}(t) = \ddot{\phi}(0) = 1$. The gradient of R with bounded ω_ϕ can be expressed as:

$$\nabla R(\mathbf{x}) \triangleq \mathbf{C}' \nabla \Phi(\mathbf{C}\mathbf{x}) = \mathbf{C}' \mathbf{D}_{\text{reg}} \mathbf{D}_\phi \{\mathbf{C}\mathbf{x}\} \mathbf{C}\mathbf{x}, \quad (2.9)$$

where \mathbf{D}_{reg} and $\mathbf{D}_\phi \{\mathbf{v}\}$ denote diagonal matrices with diagonal entries ω_{reg} and $\omega_\phi(v)$ arranged in proper orders, respectively. Note that since both $\omega_{\text{reg}}(j; m)$ and $\omega_\phi([\mathbf{C}_m \mathbf{x}]_j)$ are computed locally, gradients of R can be computed efficiently on the fly with the expression in (2.9).

However, due to the space-varying “shape” (i.e., local curvature or Hessian) of the non-quadratic negative log-likelihood $\bar{L}(\mathbf{x}; \mathbf{y})$, the minimization problem is usually hard to solve analytically and numerically. Furthermore, for some more general noise models of X-ray CT, $\bar{L}(\mathbf{x}; \mathbf{y})$ can be non-convex, making the minimization problem prone to local minimums. One typical way to solve this problem is to approximate the negative log-likelihood with a second-order Taylor expansion, thus leading to a penalized weighted least-squares (PWLS) formulation for model-based X-ray CT image reconstruction [3]:

$$\hat{\mathbf{x}}_{\text{PWLS}} \in \arg \min_{\mathbf{x} \in \Omega} \left\{ \Psi_{\text{PWLS}}(\mathbf{x}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + R(\mathbf{x}) \right\}, \quad (2.10)$$

where \mathbf{W} is the statistical diagonal weighting matrix whose diagonal entries are inverse proportional to the measurement variances in the simple Poisson noise model (thus with extremely huge dynamic range), and Ω denotes a convex set for a box constraint (usually the non-negativity constraint) on \mathbf{x} . With high photon flux, the PWLS formulation (2.10) leads to negligible bias [16] and a simpler cost function. We will focus on solving this formulation with some advance convex optimization algorithms in this thesis.

2.2.1 Gradient-based methods

The conventional way to solve (2.10) is to use gradient-based methods. Let $\ell(\mathbf{x})$ denote the weighted quadratic data-fitting term in (2.10) and $[\cdot]_\Omega$ denote an operator that projects a vector onto Ω . The projected gradient descent method solving (2.10) is described in Algorithm II.1. Convergence of Algorithm II.1 is guaranteed if the step size $t > 0$ satisfies some convergence condition. For example, when $1/t$ is greater than the Lipschitz constant of Ψ_{PWLS} , Algorithm II.1 guarantees convergence and monotone decrease of the cost function.

Algorithm II.1 Projected gradient descent method solving (2.10)

Require: $\mathbf{x}^{(0)} \in \mathbb{R}^{n_p}$, $n_{\text{iter}} \geq 1$, $t > 0$

- 1: **for** $k = 0, 1, \dots, n_{\text{iter}} - 1$ **do**
 - 2: $\mathbf{x}^{(k+1)} = [\mathbf{x}^{(k)} - t(\nabla\ell(\mathbf{x}^{(k)}) + \nabla R(\mathbf{x}^{(k)}))]_{\Omega}$
 - 3: **end for**
 - 4: **return** $\mathbf{x}^{(n_{\text{iter}})}$
-

Another closely-related convex optimization method is called the optimization transfer method, the majorization-minimization (MM) method [17], or the separable quadratic surrogate (SQS) method. In the SQS method, the cost function Ψ_{PWLS} in (2.10) is replaced by its SQS function $\check{\Psi}_{\text{PWLS}}$ (evaluated at the previous iterate) at each iteration, and the image update is performed by exactly minimizing the separable quadratic majorized cost function. The SQS function of a function φ evaluated at $\mathbf{x} = \bar{\mathbf{x}}$ with diagonal Hessian $\mathbf{D}(\bar{\mathbf{x}})$ is defined as

$$\check{\varphi}(\mathbf{x}; \bar{\mathbf{x}}) \triangleq \varphi(\bar{\mathbf{x}}) + (\mathbf{x} - \bar{\mathbf{x}})' \nabla \varphi(\bar{\mathbf{x}}) + \frac{1}{2} \|\mathbf{x} - \bar{\mathbf{x}}\|_{\mathbf{D}(\bar{\mathbf{x}})}^2 \quad (2.11)$$

which satisfies the “majorization” condition:

$$\begin{cases} \check{\varphi}(\mathbf{x}; \bar{\mathbf{x}}) \geq \varphi(\mathbf{x}) & , \forall \mathbf{x}, \bar{\mathbf{x}} \in \text{Dom} \varphi \\ \check{\varphi}(\bar{\mathbf{x}}; \bar{\mathbf{x}}) = \varphi(\bar{\mathbf{x}}) & , \forall \bar{\mathbf{x}} \in \text{Dom} \varphi. \end{cases} \quad (2.12)$$

In X-ray CT image reconstruction, the standard choice of $\check{\Psi}_{\text{PWLS}}$ at the k th iteration is the sum of $\check{\ell}(\mathbf{x}; \mathbf{x}^{(k)})$ and $\check{R}(\mathbf{x}; \mathbf{x}^{(k)})$ with diagonal Hessian matrices

$$\mathbf{D}_L \triangleq \text{diag}\{\mathbf{A}'\mathbf{W}\mathbf{A}\mathbf{1}\} \succeq \nabla^2 \ell(\mathbf{x}^{(k)}) = \mathbf{A}'\mathbf{W}\mathbf{A} \quad (2.13)$$

and

$$\mathbf{D}_R^{(k)} \triangleq \text{diag}\{|\mathbf{C}'\mathbf{D}_{\text{reg}}\mathbf{D}_{\phi}\{\mathbf{C}\mathbf{x}^{(k)}\}|\mathbf{C}\mathbf{1}\} \succeq \mathbf{C}'\mathbf{D}_{\text{reg}}\mathbf{D}_{\phi}\{\mathbf{C}\mathbf{x}^{(k)}\}\mathbf{C}, \quad (2.14)$$

respectively. The rightmost term in (2.14) is called Huber’s curvature of R evaluated at $\mathbf{x} = \mathbf{x}^{(k)}$ [18]. Furthermore, since $\max_t \omega_{\phi}(t) = 1$, we can find another looser but universal diagonal Hessian matrix for \check{R} as:

$$\mathbf{D}_R \triangleq \text{diag}\{|\mathbf{C}'\mathbf{D}_{\text{reg}}|\mathbf{C}\mathbf{1}\} \succeq \mathbf{C}'\mathbf{D}_{\text{reg}}\mathbf{C}, \quad (2.15)$$

where the rightmost term is called the maximum curvature of R everywhere. In

terms of convergence rate, Huber’s curvature is preferable since it is tighter and data-dependent. We use it in the SQS method solving (2.10) described in Algorithm II.2. Due to the majorization conditions (2.12), the image update in Algorithm II.2 also guarantees monotone decrease of the cost function. In fact, projected gradient descent method is just an SQS method that majorizes the cost function by an “isotropic” SQS function (i.e., with a “constant-times-identity” diagonal Hessian matrix). However, unlike Algorithm II.1, Algorithm II.2 allows voxel-dependent step sizes in the image update and does not need to estimate the Lipschitz constant of Ψ_{PWLS} . Hence, the SQS method is usually more preferable for X-ray CT image reconstruction problems.

Algorithm II.2 Separable quadratic surrogate (SQS) method solving (2.10)

Require: $\mathbf{x}^{(0)} \in \mathbb{R}^{n_p}$, $n_{\text{iter}} \geq 1$

1: **for** $k = 0, 1, \dots, n_{\text{iter}} - 1$ **do**

2: $\mathbf{x}^{(k+1)} = \left[\mathbf{x}^{(k)} - (\mathbf{D}_L + \mathbf{D}_R^{(k)})^{-1} (\nabla \ell(\mathbf{x}^{(k)}) + \nabla R(\mathbf{x}^{(k)})) \right]_{\Omega}$

3: **end for**

4: **return** $\mathbf{x}^{(n_{\text{iter}})}$

However, the standard SQS method is not used in practice because it takes one forward/back-projection per iteration for evaluating the gradient of ℓ and converges very slowly. As mentioned before, since the coefficient of \mathbf{A} is computed on the fly, multiplication by \mathbf{A} and \mathbf{A}' is very slow. A more practical SQS method is call the OS-SQS method [10]. By grouping the projections into M ordered subsets (OS) that satisfy the “subset balance condition” and updating the image incrementally using the M subset gradients, the OS-SQS method effectively performs M times as many as image updates per outer iteration as the standard SQS method, leading to approximately M times acceleration in early iterations. Mathematically, ℓ is decomposed into M smaller quadratic functions ℓ_1, \dots, ℓ_M , where

$$\ell_m(\mathbf{x}) \triangleq \frac{1}{2} \|\mathbf{y}_m - \mathbf{A}_m \mathbf{x}\|_{\mathbf{W}_m}^2 \quad (2.16)$$

for $m = 1, \dots, M$, and $\{\mathbf{y}_1, \dots, \mathbf{y}_M\}$, $\{\mathbf{A}_1, \dots, \mathbf{A}_M\}$, and $\{\mathbf{W}_1, \dots, \mathbf{W}_M\}$ are the corresponding non-overlapping partition of \mathbf{y} , \mathbf{A} , and \mathbf{W} , respectively. Suppose the projections are grouped properly. We should have

$$\nabla \ell(\mathbf{x}) \approx M \nabla \ell_1(\mathbf{x}) \approx \dots \approx M \nabla \ell_M(\mathbf{x}) \quad (2.17)$$

when \mathbf{x} is far away from the optimum $\hat{\mathbf{x}}_{\text{PWLS}}$, and we can use the subset gradients to approximate the full gradient of ℓ [10]. The OS-SQS method [10] solving (2.10) is

described in Algorithm II.3. The net computational complexity per outer iteration of Algorithm II.3 is one forward and back-projection (for computing subset gradients of ℓ) and M regularizer gradient evaluations. That is, the OS-SQS method evaluates regularizer gradients more frequently than the standard SQS method. However, since the most computationally expensive operation in CT is the forward/back-projection, the OS-SQS method is still faster than the standard SQS method when M is not too big. Furthermore, since the gradients in the OS-SQS method are computed inexactly, and the subset gradient approximation (2.17) is valid only when the current iterate is far away from the optimum, the OS-SQS method eventually approaches a “limit cycle” in which updates stop approaching the optimum. The OS-SQS method is not convergent in general unless relaxation [19] or incremental majorization [20] is used, unsurprisingly, at the cost of slower convergence rate. We can interpret the OS-SQS method as an incremental gradient method [21]; when the subsets are chosen randomly with some constraints so that the subset gradient is unbiased and with finite variance, the OS-SQS method can also be referred as a stochastic gradient method [22] in the machine learning literature.

Algorithm II.3 Ordered-subset SQS (OS-SQS) method solving (2.10)

Require: $\mathbf{x}^{(0)} \in \mathbb{R}^{n_p}$, $n_{\text{iter}} \geq 1$, $M \geq 1$

- 1: **for** $k = 0, 1, \dots, n_{\text{iter}} - 1$ **do**
- 2: $\mathbf{x}^{(k,1)} = \mathbf{x}^{(k)}$
- 3: **for** $m = 1, \dots, M$ **do**
- 4: $\mathbf{x}^{(k,m+1)} = \left[\mathbf{x}^{(k,m)} - (\mathbf{D}_L + \mathbf{D}_R^{(k,m)})^{-1} (M \nabla \ell_m(\mathbf{x}^{(k,m)}) + \nabla R(\mathbf{x}^{(k,m)})) \right]_{\Omega}$
- 5: **end for**
- 6: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k,M+1)}$
- 7: **end for**
- 8: **return** $\mathbf{x}^{(n_{\text{iter}})}$

Recently, OS variants of fast gradient methods [23–25] were proposed and demonstrated dramatic accelerations (about M^2 times in early iterations) in convergence rate over their one-subset counterparts [26, 27]. People call these fast OS methods the OS+momentum methods because the momentum of update trajectory is used for acceleration in these methods. An OS+momentum method [27] is described in Algorithm II.4. Note that the net computational complexity per outer iteration of the OS+momentum (OS-Nes05) method is almost unchanged comparing with the OS-SQS method. With negligible (memory and computational) overhead, OS+momentum methods improve the early convergence rate significantly! However, experimental results also showed that OS+momentum methods seem to have “larger” limit cycles

and exhibit noise-like OS artifacts in the reconstructed images when M increases [28]. This problem is also studied in the machine learning literature. Devolder showed that the error accumulation in fast gradient methods is inevitable when an inexact oracle is used, but it can be reduced by using relaxed momentum, i.e., a growing diagonal majorizer (or equivalently, a diminishing step size), at the cost of slower convergence rate [29]. An OS variant of the relaxed momentum method was investigated for solving X-ray CT image reconstruction problems [28]. A simplified relaxed OS+momentum method (OS-rNes05) is described in Algorithm II.5. There is an additional parameter $\gamma \geq 0$ in Algorithm II.5. This parameter affects the amount of relaxation. When $\gamma = 0$, Algorithm II.5 reverts to a standard OS+momentum method. Schmidt *et al.* also showed that an accelerated proximal gradient method is more sensitive to errors in the gradient and proximal mapping calculation of the smooth and non-smooth cost function components, respectively [30]. In OS-based methods, M affects the accuracy of gradient calculation. When M is too large, gradient errors accumulate too fast, and the algorithm diverges. Therefore, choosing an appropriate M for OS+momentum methods that ensures both fast convergence and stable reconstruction is very important when implementing these methods for real applications.

Algorithm II.4 OS+momentum (OS-Nes05) method [27] solving (2.10)

Require: $\mathbf{x}^{(0)} \in \mathbb{R}^{n_p}$, $n_{\text{iter}} \geq 1$, $M \geq 1$

- 1: Initialize $\mathbf{z}^{(0)} = \mathbf{x}^{(0)}$, $\mathbf{v}^{(0)} = \mathbf{0}$, $t_0 = 1$
- 2: **for** $k = 0, 1, \dots, n_{\text{iter}} - 1$ **do**
- 3: $\mathbf{c}^{(k,1)} = \mathbf{c}^{(k)}$ for $\mathbf{c} \in \{\mathbf{x}, \mathbf{v}, \mathbf{z}\}$
- 4: $t_{k,1} = t_k$
- 5: **for** $m = 1, \dots, M$ **do**
- 6: $\mathbf{D}^{(k,m+1)} = \mathbf{D}_L + \mathbf{D}_R^{(k,m)}$
- 7: $\mathbf{g}^{(k,m+1)} = M \nabla \ell_m(\mathbf{z}^{(k,m)}) + \nabla R(\mathbf{z}^{(k,m)})$
- 8: $\mathbf{x}^{(k,m+1)} = \left[\mathbf{z}^{(k,m)} - (\mathbf{D}^{(k,m+1)})^{-1} \mathbf{g}^{(k,m+1)} \right]_{\Omega}$
- 9: $\mathbf{v}^{(k,m+1)} = \mathbf{v}^{(k,m)} + t_{k,m} \mathbf{g}^{(k,m+1)}$
- 10: $t_{k,m+1} = \frac{1 + \sqrt{1 + 4t_{k,m}^2}}{2}$
- 11: $\mathbf{z}^{(k,m+1)} = \frac{t_{k,m+1} - 1}{t_{k,m+1}} \mathbf{x}^{(k,m+1)} + \frac{1}{t_{k,m+1}} \left[\mathbf{z}^{(0)} - (\mathbf{D}^{(k,m)})^{-1} \mathbf{v}^{(k,m+1)} \right]_{\Omega}$
- 12: **end for**
- 13: $\mathbf{c}^{(k+1)} = \mathbf{c}^{(k,M+1)}$ for $\mathbf{c} \in \{\mathbf{x}, \mathbf{v}, \mathbf{z}\}$
- 14: $t_{k+1} = t_{k,M+1}$
- 15: **end for**
- 16: **return** $\mathbf{x}^{(n_{\text{iter}})}$

Algorithm II.5 Relaxed OS+momentum (OS-rNes05) method [27] solving (2.10)

Require: $\mathbf{x}^{(0)} \in \mathbb{R}^{n_p}$, $n_{\text{iter}} \geq 1$, $M \geq 1$, $\gamma \geq 0$

- 1: Initialize $\mathbf{z}^{(0)} = \mathbf{x}^{(0)}$, $\mathbf{v}^{(0)} = \mathbf{0}$, $\mathbf{\Gamma} = \gamma \cdot \text{median}([\mathbf{D}_L]_j) \mathbf{I}$, $r = 1$
 - 2: **for** $k = 0, 1, \dots, n_{\text{iter}} - 1$ **do**
 - 3: $\mathbf{c}^{(k,1)} = \mathbf{c}^{(k)}$ for $\mathbf{c} \in \{\mathbf{x}, \mathbf{v}, \mathbf{z}\}$
 - 4: **for** $m = 1, \dots, M$ **do**
 - 5: $\mathbf{D}^{(k,m+1)} = \mathbf{D}_L + \mathbf{D}_R^{(k,m)} + (r + 2) \mathbf{\Gamma}$
 - 6: $\mathbf{g}^{(k,m+1)} = M \nabla \ell_m(\mathbf{z}^{(k,m)}) + \nabla \mathbf{R}(\mathbf{z}^{(k,m)})$
 - 7: $\mathbf{x}^{(k,m+1)} = \left[\mathbf{z}^{(k,m)} - (\mathbf{D}^{(k,m+1)})^{-1} \mathbf{g}^{(k,m+1)} \right]_{\Omega}$
 - 8: $\mathbf{v}^{(k,m+1)} = \mathbf{v}^{(k,m)} + \frac{r+1}{2} \mathbf{g}^{(k,m+1)}$
 - 9: $\mathbf{z}^{(k,m+1)} = \frac{r+1}{r+3} \mathbf{x}^{(k,m+1)} + \frac{2}{r+3} \left[\mathbf{z}^{(0)} - (\mathbf{D}^{(k,m+1)})^{-1} \mathbf{v}^{(k,m+1)} \right]_{\Omega}$
 - 10: $r = r + 1$
 - 11: **end for**
 - 12: $\mathbf{c}^{(k+1)} = \mathbf{c}^{(k,M+1)}$ for $\mathbf{c} \in \{\mathbf{x}, \mathbf{v}, \mathbf{z}\}$
 - 13: **end for**
 - 14: **return** $\mathbf{x}^{(n_{\text{iter}})}$
-

2.2.2 Splitting-based methods

Unlike gradient-based methods that devote themselves to finding a minimizer that minimizes the cost function of the original minimization problem, splitting-based methods solve a minimization problem by divide-and-conquer. More precisely, they decompose the original minimization problem into a series of easier penalized least-squares problems using a mathematical technique called variable splitting. These methods are especially useful for minimization problems with (tractable) non-smooth cost function such as the LASSO regression [31], a least-squares problem with an ℓ_1 regularization, because the non-differentiable nature of those problems precludes optimization by conventional gradient-based methods. They become more popular in X-ray CT image reconstruction with non-smooth regularizations these days due to the recent resurgence of the classic augmented Lagrangian (AL) methods in fields like total-variation (TV) denoising and compressed sensing (CS). In fact, splitting-based methods are also very useful for X-ray CT image reconstructions even when smooth approximations (of non-smooth regularizers, e.g., using corner-rounding) are employed because approximations with such modifications usually have very high curvature, leading to very slow convergence of gradient-based methods again.

To demonstrate how splitting-based methods solve an X-ray CT image reconstruction problem, let's consider a sparse-view X-ray CT image reconstruction problem for example. Undersampling projection views is one way to reduce radiation dose in

CT scans; however, this causes strong streak artifacts in FBP images that degrade image quality. To overcome this problem, the split Bregman (SB) method [32], an alias of the AL method in the context of ℓ_1 -regularized image reconstruction problems, has been investigated using strong non-smooth TV and sparsity-promoting regularizations [7, 33]. Mathematically, instead of solving (2.10) directly, consider an equivalent equality-constrained minimization problem (we ignore the box constraint here for simplicity):

$$(\hat{\mathbf{x}}, \hat{\mathbf{v}}) \in \arg \min_{\mathbf{x}, \mathbf{v}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \Phi(\mathbf{v}) \right\} \text{ s.t. } \mathbf{v} = \mathbf{C}\mathbf{x}. \quad (2.18)$$

The SB method solves this problem by finding a saddle point of the corresponding scaled augmented Lagrangian of (2.18):

$$\mathcal{L}_A(\mathbf{x}, \mathbf{v}, \mathbf{e}; \eta) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \Phi(\mathbf{v}) + \frac{\eta}{2} \|\mathbf{C}\mathbf{x} - \mathbf{v} - \mathbf{e}\|_2^2, \quad (2.19)$$

where \mathbf{e} is the scaled Lagrange multiplier of the auxiliary variable \mathbf{v} , and $\eta > 0$ is the corresponding AL penalty parameter, using an alternating minimization method followed by a gradient ascent of the (unscaled) Lagrangian multiplier. For instance, suppose we update variables in the order of \mathbf{x} , \mathbf{v} , and then \mathbf{e} . At the k th iteration, the \mathbf{x} -update is a least-squares problem:

$$\mathbf{x}^{(k+1)} \in \arg \min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \frac{\eta}{2} \|\mathbf{C}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)}\|_2^2 \right\} \quad (2.20)$$

with an analytical solution:

$$\hat{\mathbf{x}}^{(k+1)} = (\mathbf{A}'\mathbf{W}\mathbf{A} + \eta\mathbf{C}'\mathbf{C})^{-1} (\mathbf{A}'\mathbf{W}\mathbf{y} + \eta\mathbf{C}'(\mathbf{v}^{(k)} + \mathbf{e}^{(k)})). \quad (2.21)$$

Due to the large problem size, inverting the big non-circulant matrix $\mathbf{A}'\mathbf{W}\mathbf{A} + \eta\mathbf{C}'\mathbf{C}$ is impossible; however, one can still find an approximate solution to (2.20) using limited iterations of the conjugate gradient (CG) method or the preconditioned CG (PCG) method. We call the corresponding iterates (with iteration limiter) an “inexact” SB method due to the inevitable inexact image update. Once $\mathbf{x}^{(k+1)}$ is determined, one can proceed to the \mathbf{v} -update, which is a penalized least-squares problem:

$$\mathbf{v}^{(k+1)} \in \arg \min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \|\mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)}\|_2^2 \right\}. \quad (2.22)$$

This can be written as the proximal mapping of Φ :

$$\mathbf{v}^{(k+1)} \in \text{prox}_{\eta^{-1}\Phi}(\mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{e}^{(k)}) , \quad (2.23)$$

where prox_φ denotes the proximal mapping of a function φ defined as:

$$\text{prox}_\varphi(\mathbf{z}) \triangleq \arg \min_{\mathbf{x}} \left\{ \varphi(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 \right\} . \quad (2.24)$$

For some φ , its proximal mapping can be computed very efficiently, e.g., the soft-shrinkage operation for the ℓ_1 -norm. For those Φ 's that do not have efficient proximal mappings, the \mathbf{v} -update in (2.22) can only be solved inexactly. We also refer to the corresponding iterates as an inexact SB method. Finally, the \mathbf{e} -update is a simple gradient ascent:

$$\mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} + (-\mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)}) . \quad (2.25)$$

In summary, the SB method solving (2.18) is described in Algorithm II.6. As a final remark, due to the inevitable inexact image (\mathbf{x} -)update and the (column) rank deficiency of the finite difference matrix \mathbf{C} , the (primal) convergence of Algorithm II.6 is not straightforward. To conclude the discussion of the SB method, a (primal) convergence proof of the (inexact) SB method for regularized least-squares problems with rank deficient analysis regularization matrices is included in Appendix A.

Algorithm II.6 Split Bregman (SB) method [7] solving (2.18)

Require: $\mathbf{x}^{(0)} \in \mathbb{R}^{n_p}$, $n_{\text{iter}} \geq 1$, $\eta > 0$

- 1: Initialize $\mathbf{v}^{(0)} = \text{prox}_{\eta^{-1}\Phi}(\mathbf{C}\mathbf{x}^{(0)})$, $\mathbf{e}^{(0)} = -\mathbf{C}\mathbf{x}^{(0)} + \mathbf{v}^{(0)}$
 - 2: **for** $k = 1, \dots, n_{\text{iter}} - 1$ **do**
 - 3: Compute $\mathbf{x}^{(k+1)}$ by solving (2.20) using (P)CG
 - 4: Compute $\mathbf{v}^{(k+1)}$ by solving (2.22) using the proximal mapping of Φ
 - 5: $\mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)}$
 - 6: **end for**
 - 7: **return** $\mathbf{x}^{(n_{\text{iter}})}$
-

Theoretically, the SB method should work pretty well for X-ray CT image reconstruction problems with non-smooth regularizers, just like the great success it achieves in TV denoising and CS. However, there are still some practical issues when solving X-ray CT image reconstruction problems using the SB method, e.g., how long the SB method takes for an image update. This is usually not a big problem for gradient-based methods because image updates in gradient-based methods are very straightforward: forward projection, back-projection, vector addition, and finally an element-wise projection. However, image updates in splitting-based methods are a lit-

tle bit tricky because they are minimization problems! Due to the enormous dynamic range of the statistical weighting \mathbf{W} , the Hessian of the statistical data-fitting term (i.e., $\mathbf{A}'\mathbf{W}\mathbf{A}$) is highly shift-variant. This forces people to run more (P)CG iterations to get an acceptable approximation to the analytical solution (2.21). For example, [7] suggested solving the challenging inner least-squares problem using up to 100 iterations of the CG method, that is, hundreds of forward/back-projection pairs for a single outer-loop image update! As mentioned before, the forward/back-projection is the most time-consuming operation in CT because we have to compute the coefficients of the system matrix \mathbf{A} on the fly. Hundreds of forward/back-projection pairs for a single image update is absolutely undesirable. To solve this problem, Ramani *et al.* introduced an additional auxiliary variable that separates the shift-variant and approximate shift-invariant parts of the statistically weighted quadratic data-fitting term so that one can find an appropriate circulant preconditioner for the better-conditioned inner least-squares problem and solve the inner least-squares problem efficient using the PCG method [8]. More precisely, the authors consider another equivalent equality-constrained minimization problem that is also equivalent to (2.10) (ignore the box constraint here again) but uses two auxiliary variables:

$$(\hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{v}}) \in \arg \min_{\mathbf{x}, \mathbf{u}, \mathbf{v}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{u}\|_{\mathbf{W}}^2 + \Phi(\mathbf{v}) \right\} \text{ s.t. } \mathbf{u} = \mathbf{A}\mathbf{x}, \mathbf{v} = \mathbf{C}\mathbf{x}. \quad (2.26)$$

The corresponding scaled augmented Lagrangian of (2.26) is

$$\begin{aligned} \mathcal{L}_A(\mathbf{x}, \mathbf{u}, \mathbf{v}, \mathbf{d}, \mathbf{e}; \rho, \eta) \\ \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{u}\|_{\mathbf{W}}^2 + \Phi(\mathbf{v}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \mathbf{u} - \mathbf{d}\|_2^2 + \frac{\eta}{2} \|\mathbf{C}\mathbf{x} - \mathbf{v} - \mathbf{e}\|_2^2, \end{aligned} \quad (2.27)$$

where \mathbf{d} and \mathbf{e} are the scaled Lagrange multipliers of the auxiliary variables \mathbf{u} and \mathbf{v} , respectively, and $\rho, \eta > 0$ are the corresponding AL penalty parameters. As can be seen in (2.27), the statistical weighting \mathbf{W} is no longer tied to the system matrix \mathbf{A} . Therefore, if we solve (2.26) using an alternating minimization method followed by a gradient ascent of the (unscaled) Lagrange multipliers in the order of \mathbf{x} , \mathbf{u} , \mathbf{v} , and then the Lagrange multipliers \mathbf{d} and \mathbf{e} , at the k th iteration, the \mathbf{x} -update is a least-squares problem:

$$\mathbf{x}^{(k+1)} \in \arg \min_{\mathbf{x}} \left\{ \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}\|_2^2 + \frac{\eta}{2} \|\mathbf{C}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)}\|_2^2 \right\} \quad (2.28)$$

with an analytical solution:

$$\hat{\mathbf{x}}^{(k+1)} = (\rho \mathbf{A}' \mathbf{A} + \eta \mathbf{C}' \mathbf{C})^{-1} (\rho \mathbf{A}' (\mathbf{u}^{(k)} + \mathbf{d}^{(k)}) + \eta \mathbf{C}' (\mathbf{v}^{(k)} + \mathbf{e}^{(k)})) . \quad (2.29)$$

Although it is still impractical to compute (2.29) exactly, the Hessian of the cost function in (2.28) (i.e., $\rho \mathbf{A}' \mathbf{A} + \eta \mathbf{C}' \mathbf{C}$) is approximately shift-invariant in 2D CT. Hence, one can precondition the problem effectively using FFTs with cone-type filters! The \mathbf{u} -update is a much simpler least-squares problem:

$$\mathbf{u}^{(k+1)} \in \arg \min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{u}\|_{\mathbf{W}}^2 + \frac{\rho}{2} \|\mathbf{A} \mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\|_2^2 \right\} \quad (2.30)$$

with an analytical solution:

$$\hat{\mathbf{u}}^{(k+1)} = (\mathbf{W} + \rho \mathbf{I})^{-1} (\mathbf{W} \mathbf{y} + \rho (\mathbf{A} \mathbf{x}^{(k+1)} - \mathbf{d}^{(k)})) . \quad (2.31)$$

Unlike the analytical solution in (2.29), $\mathbf{W} + \rho \mathbf{I}$ in (2.31) is a diagonal matrix, so the \mathbf{u} -update (2.30) can be solved exactly and efficiently by element-wise division. The \mathbf{v} -update here is exactly the same as that in the SB method, i.e., (2.23), and it can be solved using the proximal mapping of Φ . Finally, the Lagrange multipliers \mathbf{d} and \mathbf{e} perform gradient ascents accordingly:

$$\begin{cases} \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} + (-\mathbf{A} \mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)}) \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} + (-\mathbf{C} \mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)}) . \end{cases} \quad (2.32)$$

The complete alternating direction method of multipliers (ADMM) is described in Algorithm II.7. We call this algorithm ADMM to distinguish it from the SB method, which uses a variable splitting scheme with a rank deficient analysis regularization matrix. A more detailed discussion about the SB method and ADMM can be found in Appendix B. Experimental results showed that ADMM provides significant acceleration over the SB method in 2D CT [8]; however, in 3D CT, due to different cone-beam geometries and scan trajectories, it is more difficult to construct a good preconditioner for the inner least-squares problem, and ADMM has yet to achieve the same acceleration as in 2D CT. Furthermore, splitting-based methods simplify optimization at the expense of storing additional auxiliary variables (and Lagrange multipliers). The additional memory requirement might be still affordable for 2D problems. For 3D problems, the remarkable memory overhead (26 extra image volumes if one considers all 13 neighbors when computing the finite differences) can

be problematic and restrains the applicability of splitting-based methods from an implementation perspective.

Algorithm II.7 Alternating direction method of multipliers (ADMM) [8] solving (2.26)

Require: $\mathbf{x}^{(0)} \in \mathbb{R}^{n_p}$, $n_{\text{iter}} \geq 1$, $\rho, \eta > 0$

1: Initialize $\mathbf{u}^{(0)} = (\mathbf{W} + \rho\mathbf{I})^{-1} (\mathbf{W}\mathbf{y} + \rho\mathbf{A}\mathbf{x}^{(0)})$, $\mathbf{d}^{(0)} = -\mathbf{A}\mathbf{x}^{(0)} + \mathbf{u}^{(0)}$,
 $\mathbf{v}^{(0)} = \text{prox}_{\eta^{-1}\Phi}(\mathbf{C}\mathbf{x}^{(0)})$, $\mathbf{e}^{(0)} = -\mathbf{C}\mathbf{x}^{(0)} + \mathbf{v}^{(0)}$

2: **for** $k = 1, \dots, n_{\text{iter}} - 1$ **do**

3: Compute $\mathbf{x}^{(k+1)}$ by solving (2.28) using (P)CG

4: $\mathbf{u}^{(k+1)} = (\mathbf{W} + \rho\mathbf{I})^{-1} (\mathbf{W}\mathbf{y} + \rho(\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{d}^{(k)}))$

5: Compute $\mathbf{v}^{(k+1)}$ by solving (2.22) using the proximal mapping of Φ

6: $\mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)}$

7: $\mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)}$

8: **end for**

9: **return** $\mathbf{x}^{(n_{\text{iter}})}$

CHAPTER III

Fast X-ray CT image reconstruction using variable splitting methods with ordered subsets

In Chapter II, we have reviewed two families of methods, gradient-based methods and splitting-based methods, solving model-based X-ray CT image reconstruction problems. From an implementation perspective, image updates of gradient-based methods are more straightforward and memory-efficient. However, unlike splitting-based methods which have more flexibility to deal with complicated regularizations using variable splitting, gradient-based methods have a practical limitation that restrains their applications to smooth regularizers with modest curvature. In this chapter, we propose to combine these two methods by considering a linearized variant of the AL method [39–42], in which we majorize the quadratic AL penalty term in the scaled augmented Lagrangian using a fixed diagonal majorizer, thus leading to a much simpler splitting-based order-subsets algorithm, OS-LALM, for model-based X-ray CT image reconstruction with PWLS criterion. In the following sections, we will go through our proposed algorithm by solving three practical issues (iterative image updates, non-trivial parameter selection, and enormous memory requirement) of splitting-based methods.

3.1 Non-iterative image update using a linearized AL method

The first practical issue of splitting-based methods is the iterative image updates due to the challenging inner least-squares problem. This issue is not highlighted before in the context of image restoration because the blur kernel is usually assumed to be shift-invariant in image restoration problems, leading to efficient image updates using FFTs. However, in X-ray CT image reconstruction problems, due to the huge

This chapter is based on [34–38].

dynamic range of transmission data in X-ray CT, the minimum (non-zero) statistical weighting can be about 10^{-5} times of the peak value typically. This makes the matrix $\mathbf{A}'\mathbf{W}\mathbf{A}$ highly ill-conditioned, so the SB method (Algorithm II.6) will be very slow in practice. In fact, even for the unweighted case, the matrix $\mathbf{A}'\mathbf{A}$ is still highly ill-conditioned in 3D CT due to the different geometries and scan trajectories. For example, in 3D helical CT scan, the local impulse response (or point spread function) rotates from slice to slice because the X-ray source rotates and moves in the z -direction at the same time. Therefore, ADMM (Algorithm II.7) is also very slow in 3D CT. One possible way to deal with the issue is to simplify the image update with some approximation that still guarantees convergence. Here, we focus on a linearized AL method that simplify the image update to a single prox-linear step [43].

3.1.1 Linearized AL method

We first review the linearized AL method in a general setting and show new convergence properties of the linearized AL method with “inexact” updates. Consider a general composite convex optimization problem:

$$\hat{\mathbf{x}} \in \arg \min_{\mathbf{x}} \{g(\mathbf{A}\mathbf{x}) + h(\mathbf{x})\} \quad (3.1)$$

and its equivalent constrained minimization problem:

$$(\hat{\mathbf{x}}, \hat{\mathbf{u}}) \in \arg \min_{\mathbf{x}, \mathbf{u}} \{g(\mathbf{u}) + h(\mathbf{x})\} \text{ s.t. } \mathbf{u} = \mathbf{A}\mathbf{x}, \quad (3.2)$$

where both g and h are closed and proper convex functions. Typically, g is a weighted quadratic data-fitting term, and h is an edge-preserving regularization (together with the box constraint Ω) term in CT. One way to solve the constrained minimization problem (3.2) is to use the (alternating direction) AL method, which alternately minimizes the scaled augmented Lagrangian:

$$\mathcal{L}_A(\mathbf{x}, \mathbf{u}, \mathbf{d}; \rho) \triangleq g(\mathbf{u}) + h(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \mathbf{u} - \mathbf{d}\|_2^2 \quad (3.3)$$

with respect to \mathbf{x} and \mathbf{u} , followed by a gradient ascent of \mathbf{d} , yielding the following AL iterates [44, 45]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg \min_{\mathbf{x}} \left\{ h(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}\|_2^2 \right\} \\ \mathbf{u}^{(k+1)} \in \arg \min_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)}, \end{cases} \quad (3.4)$$

where \mathbf{d} is the scaled Lagrange multiplier of the auxiliary variable \mathbf{u} , and $\rho > 0$ is the corresponding AL penalty parameter.

In the linearized AL method [39–42] (also known as the split inexact Uzawa method in the image processing literature [46–48]), one replaces the quadratic AL penalty term in the \mathbf{x} -update of (3.4):

$$\theta_k(\mathbf{x}) \triangleq \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}\|_2^2 \quad (3.5)$$

by its SQS function defined in (2.11):

$$\begin{aligned} & \check{\theta}_k(\mathbf{x}; \mathbf{x}^{(k)}) \\ & \triangleq \theta_k(\mathbf{x}^{(k)}) + \langle \nabla \theta_k(\mathbf{x}^{(k)}), \mathbf{x} - \mathbf{x}^{(k)} \rangle + \frac{\rho L}{2} \|\mathbf{x} - \mathbf{x}^{(k)}\|_2^2 \\ & = \frac{\rho}{2t} \left\| \mathbf{x} - (\mathbf{x}^{(k)} - t\mathbf{A}'(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)})) \right\|_2^2 + (\text{const. independent of } \mathbf{x}), \end{aligned} \quad (3.6)$$

where $L > \|\mathbf{A}\|_2^2 = \lambda_{\max}(\mathbf{A}'\mathbf{A})$ ensures that $L\mathbf{I} \succ \mathbf{A}'\mathbf{A}$, and $t \triangleq 1/L$. It is trivial to generalize L to a symmetric positive semi-definite matrix \mathbf{L} , e.g., the diagonal matrix $\text{diag}\{|\mathbf{A}'|\mathbf{A}|\mathbf{1}\}$ used in OS-based methods [10, 49], and still ensure (2.12). When $\mathbf{L} = \mathbf{A}'\mathbf{A}$, the linearized AL method reverts to the standard AL method. Majorizing with a diagonal matrix removes the entanglement of \mathbf{x} introduced by the system matrix \mathbf{A} and leads to a simpler \mathbf{x} -update. The corresponding linearized AL iterates are as follows [39–42]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg \min_{\mathbf{x}} \left\{ \phi_k(\mathbf{x}) \triangleq h(\mathbf{x}) + \check{\theta}_k(\mathbf{x}; \mathbf{x}^{(k)}) \right\} \\ \mathbf{u}^{(k+1)} \in \arg \min_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)}. \end{cases} \quad (3.7)$$

The \mathbf{x} -update can be written as the proximal mapping (2.24) of h :

$$\begin{aligned}\mathbf{x}^{(k+1)} &\in \text{prox}_{(\rho^{-1}t)h}(\mathbf{x}^{(k)} - t\mathbf{A}'(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)})) \\ &= \text{prox}_{(\rho^{-1}t)h}(\mathbf{x}^{(k)} - (\rho^{-1}t)\mathbf{s}^{(k+1)}),\end{aligned}\quad (3.8)$$

where

$$\mathbf{s}^{(k+1)} \triangleq \rho\mathbf{A}'(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}) \quad (3.9)$$

denotes the “search direction” of the proximal gradient \mathbf{x} -update in (3.8). Furthermore, $\check{\theta}_k$ can also be written as:

$$\check{\theta}_k(\mathbf{x}; \mathbf{x}^{(k)}) = \theta_k(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{x} - \mathbf{x}^{(k)}\|_{\mathbf{G}}^2, \quad (3.10)$$

where $\mathbf{G} \triangleq L\mathbf{I} - \mathbf{A}'\mathbf{A} \succ 0$ by the definition of L . Hence, the linearized AL iterates (3.7) can be represented as a proximal-point variant of the standard AL iterates (3.4) (also known as the preconditioned AL iterates) by plugging (3.10) into (3.7) [46, 50, 51]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg \min_{\mathbf{x}} \left\{ h(\mathbf{x}) + \theta_k(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{x} - \mathbf{x}^{(k)}\|_{\mathbf{G}}^2 \right\} \\ \mathbf{u}^{(k+1)} \in \arg \min_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)}. \end{cases} \quad (3.11)$$

The linearized AL method (3.7) is convergent with any fixed AL penalty parameter $\rho > 0$ for any \mathbf{A} [39–42], while the standard AL method is (primal) convergent only if \mathbf{A} has full column rank [44, Theorem 8]. Furthermore, even if the AL penalty parameter varies every iteration, (3.7) is convergent when ρ is non-decreasing and bounded above [39]. However, all these convergence analyses assume that all updates are exact. In this chapter, we are more interested in the linearized AL method with inexact updates. Specifically, instead of the exact linearized AL method (3.7), we focus on inexact linearized AL methods:

$$\begin{cases} \left\| \mathbf{x}^{(k+1)} - \arg \min_{\mathbf{x}} \phi_k(\mathbf{x}) \right\| \leq \delta_k \\ \mathbf{u}^{(k+1)} \in \arg \min_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)}, \end{cases} \quad (3.12)$$

where ϕ_k was defined in (3.7), and

$$\begin{cases} \left| \phi_k(\mathbf{x}^{(k+1)}) - \min_{\mathbf{x}} \phi_k(\mathbf{x}) \right| \leq \varepsilon_k \\ \mathbf{u}^{(k+1)} \in \arg \min_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)}. \end{cases} \quad (3.13)$$

The \mathbf{u} -update can also be inexact; however, for simplicity, we focus on exact updates of \mathbf{u} . Considering an inexact update of \mathbf{u} is a trivial extension.

Our convergence analysis of the inexact linearized AL method is twofold. First, we show that the equivalent proximal-point variant of the standard AL iterates (3.11) can be interpreted as a convergent ADMM that solves another equivalent constrained minimization problem of (3.1) with a redundant auxiliary variable (the proof can be found in Appendix C):

$$(\hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{v}}) \in \arg \min_{\mathbf{x}, \mathbf{u}, \mathbf{v}} \{g(\mathbf{u}) + h(\mathbf{x})\} \text{ s.t. } \mathbf{u} = \mathbf{A}\mathbf{x} \text{ and } \mathbf{v} = \mathbf{G}^{1/2}\mathbf{x}. \quad (3.14)$$

Therefore, the linearized AL method is a convergent ADMM, and it has all the nice properties of ADMM, including the tolerance of inexact updates [44, Theorem 8]. More formally, we have the following theorem:

Theorem III.1. *Consider a constrained composite convex optimization problem (3.2) where both g and h are closed and proper convex functions. Let $\rho > 0$ and $\{\delta_k\}_{k=0}^{\infty}$ denote a non-negative sequence such that*

$$\sum_{k=0}^{\infty} \delta_k < \infty. \quad (3.15)$$

If (3.2) has a solution $(\hat{\mathbf{x}}, \hat{\mathbf{u}})$, then the sequence of updates $\{(\mathbf{x}^{(k)}, \mathbf{u}^{(k)})\}_{k=0}^{\infty}$ generated by the inexact linearized AL method (3.12) converges to $(\hat{\mathbf{x}}, \hat{\mathbf{u}})$; otherwise, at least one of the sequences $\{(\mathbf{x}^{(k)}, \mathbf{u}^{(k)})\}_{k=0}^{\infty}$ or $\{\mathbf{d}^{(k)}\}_{k=0}^{\infty}$ diverges.

Theorem III.1 shows that the inexact linearized AL method (3.12) converges if the error δ_k is absolutely summable. However, it does not describe how fast the algorithm converges and more importantly, how inexact updates affect the convergence rate. This leads to the second part of our convergence analysis.

In this part, we rely on the equivalence between the linearized AL method and the Chambolle-Pock first-order primal-dual algorithm (CPPDA) [50]. Consider a

minimax problem:

$$(\hat{\mathbf{z}}, \hat{\mathbf{x}}) \in \arg \min_{\mathbf{z}} \max_{\mathbf{x}} \Omega(\mathbf{z}, \mathbf{x}) , \quad (3.16)$$

where

$$\Omega(\mathbf{z}, \mathbf{x}) \triangleq \langle -\mathbf{A}'\mathbf{z}, \mathbf{x} \rangle + g^*(\mathbf{z}) - h(\mathbf{x}) , \quad (3.17)$$

and φ^* denotes the convex conjugate of a function φ . Note that since both g and h are closed, proper, and convex, we have $g^{**} = g$ and $h^{**} = h$. The sequence of updates $\{(\mathbf{z}^{(k)}, \mathbf{x}^{(k)})\}_{k=0}^{\infty}$ generated by the CPPDA iterates:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \text{prox}_{\sigma h}(\mathbf{x}^{(k)} - \sigma \mathbf{A}'\bar{\mathbf{z}}^{(k)}) \\ \mathbf{z}^{(k+1)} \in \text{prox}_{\tau g^*}(\mathbf{z}^{(k)} + \tau \mathbf{A}\mathbf{x}^{(k+1)}) \\ \bar{\mathbf{z}}^{(k+1)} = \mathbf{z}^{(k+1)} + (\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)}) \end{cases} \quad (3.18)$$

converges to a saddle-point $(\hat{\mathbf{z}}, \hat{\mathbf{x}})$ of (3.16), and the non-negative primal-dual gap $\Omega(\mathbf{z}_k, \hat{\mathbf{x}}) - \Omega(\hat{\mathbf{z}}, \mathbf{x}_k)$ converges to zero with rate $O(1/k)$ [50, Theorem 1], where \mathbf{x}_k and \mathbf{z}_k denote the arithmetic mean of all previous \mathbf{x} - and \mathbf{z} -updates up to the k th iteration, respectively. Since the CPPDA iterates (3.18) solve the minimax problem (3.16), they also solve the primal problem:

$$\hat{\mathbf{z}} \in \arg \min_{\mathbf{z}} \{h^*(-\mathbf{A}'\mathbf{z}) + g^*(\mathbf{z})\} \quad (3.19)$$

and the dual problem:

$$\hat{\mathbf{x}} \in \arg \max_{\mathbf{x}} \{-g(\mathbf{A}\mathbf{x}) - h(\mathbf{x})\} \quad (3.20)$$

of (3.16), and the latter happens to be the composite convex optimization problem (3.1). Therefore, the CPPDA iterates (3.18) solve (3.1) with rate $O(1/k)$ in an ergodic sense. Furthermore, Chambolle *et al.* showed that their proposed primal-dual algorithm is equivalent to a preconditioned ADMM solving (3.2) with a preconditioner $\mathbf{M} \triangleq \sigma^{-1}\mathbf{I} - \tau\mathbf{A}'\mathbf{A}$ provided that $0 < \sigma\tau < 1/\|\mathbf{A}\|_2^2$ [50, Section 4.3]. Letting $\mathbf{z}^{(k)} = -\tau\mathbf{d}^{(k)}$ and choosing $\sigma = \rho^{-1}t$ and $\tau = \rho$, the CPPDA iterates (3.18) reduce to (3.11) and hence, the linearized AL method (3.7). This suggests that we can measure the convergence rate of the linearized AL method using the primal-dual gap that is vanishing ergodically with rate $O(1/k)$. Finally, to take inexact updates into account, we apply the error analysis technique developed in [30] to the convergence rate analysis of CPPDA, leading to the following theorem (the proof can be found in Appendix C):

Theorem III.2. *Consider a minimax problem (3.16) where both g and h are closed*

and proper convex functions. Suppose it has a saddle-point $(\hat{\mathbf{z}}, \hat{\mathbf{x}})$, where $\hat{\mathbf{z}}$ and $\hat{\mathbf{x}}$ are the solutions of the primal problem (3.19) and the dual problem (3.20) of (3.16), respectively. Let $\rho > 0$ and $\{\varepsilon_k\}_{k=0}^{\infty}$ denote a non-negative sequence such that

$$\sum_{k=0}^{\infty} \sqrt{\varepsilon_k} < \infty. \quad (3.21)$$

Then, the sequence of updates $\{(-\rho \mathbf{d}^{(k)}, \mathbf{x}^{(k)})\}_{k=0}^{\infty}$ generated by the inexact linearized AL method (3.13) is a bounded sequence that converges to $(\hat{\mathbf{z}}, \hat{\mathbf{x}})$, and the primal-dual gap of $(\mathbf{z}_k, \mathbf{x}_k)$ has the following bound:

$$\Omega(\mathbf{z}_k, \hat{\mathbf{x}}) - \Omega(\hat{\mathbf{z}}, \mathbf{x}_k) \leq \frac{(C + 2A_k + \sqrt{B_k})^2}{k}, \quad (3.22)$$

where $\mathbf{z}_k \triangleq \frac{1}{k} \sum_{j=1}^k (-\rho \mathbf{d}^{(j)})$, $\mathbf{x}_k \triangleq \frac{1}{k} \sum_{j=1}^k \mathbf{x}^{(j)}$,

$$C \triangleq \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\rho^{-1}t}} + \frac{\|(-\rho \mathbf{d}^{(0)}) - \hat{\mathbf{z}}\|_2}{\sqrt{2\rho}}, \quad (3.23)$$

$$A_k \triangleq \sum_{j=1}^k \sqrt{\frac{\varepsilon_{j-1}}{(1 - t\|\mathbf{A}\|_2^2)\rho^{-1}t}}, \quad (3.24)$$

and

$$B_k \triangleq \sum_{j=1}^k \varepsilon_{j-1}. \quad (3.25)$$

Theorem III.2 shows that the inexact linearized AL method (3.13) converges with rate $O(1/k)$ if the square root of the error ε_k is absolutely summable. In fact, even if $\{\sqrt{\varepsilon_k}\}_{k=0}^{\infty}$ is not absolutely summable, say, $\sqrt{\varepsilon_k}$ decreases as $O(1/k)$, A_k grows as $O(\log k)$ (note that B_k always grows slower than A_k), and the primal-dual gap converges to zero in $O(\log^2 k/k)$. To obtain convergence of the primal-dual gap, a necessary condition is that the partial sum of $\{\sqrt{\varepsilon_k}\}_{k=0}^{\infty}$ grows no faster than $o(\sqrt{k})$.

The primal-dual gap convergence bound above is measured at the average point $(-\rho \mathbf{d}_k, \mathbf{x}_k)$ of the update trajectory. In practice, the primal-dual gap of $(-\rho \mathbf{d}^{(k)}, \mathbf{x}^{(k)})$ converges much faster than that. Minimizing the constant in (3.22) need not provide the fastest convergence rate of the linearized AL method. However, the ρ -, t -, and ε_k -dependence in (3.22) suggests how these factors affect the convergence rate of the linearized AL method. Finally, although the linearized AL method has gradient-based image updates that pretty much solves the iterative image updates problem of

splitting-based methods, it is still not suitable for OS acceleration in general because the \mathbf{d} -update takes a full forward projection. We use the linearized AL method for analysis and to motivate the proposed algorithm in Section 3.1.2, but it is not recommended for practical implementation in CT reconstruction. By restricting g to be a quadratic loss function, we show that the linearized AL method becomes OS-acceleratable.

3.1.2 OS-LALM: a splitting-based ordered-subset method

Now, we restrict g to be a quadratic loss function, i.e., $g(\mathbf{u}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{u}\|_2^2$, and then the minimization problem (3.1) becomes a regularized least-squares problem:

$$\hat{\mathbf{x}} \in \arg \min_{\mathbf{x}} \left\{ \Psi(\mathbf{x}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + h(\mathbf{x}) \right\}. \quad (3.26)$$

It is very easy to generalize the above regularized least-squares problem to a weighted one such as (2.10) by using substitutions $\mathbf{y} \leftarrow \mathbf{W}^{1/2}\mathbf{y}$, $\mathbf{A} \leftarrow \mathbf{W}^{1/2}\mathbf{A}$, and $h \leftarrow \mathbf{R} + \iota_{\mathcal{C}}$, where $\iota_{\mathcal{C}}$ denotes the characteristic function of a convex set \mathcal{C} . Let $\ell(\mathbf{x}) \triangleq g(\mathbf{A}\mathbf{x})$ denote the quadratic data-fitting term in (3.26). We assume that ℓ is suitable for OS acceleration; i.e., ℓ can be decomposed into M smaller quadratic functions ℓ_1, \dots, ℓ_M satisfying the ‘‘subset balance condition’’ (2.17).

Since g is quadratic, its proximal mapping is linear. The \mathbf{u} -update in the linearized AL method (3.7) has the following simple closed-form solution:

$$\mathbf{u}^{(k+1)} = \frac{\rho}{\rho+1} (\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{d}^{(k)}) + \frac{1}{\rho+1} \mathbf{y}. \quad (3.27)$$

Combining (3.27) with the \mathbf{d} -update of (3.7) yields the identity

$$\mathbf{u}^{(k+1)} + \rho \mathbf{d}^{(k+1)} = \mathbf{y} \quad (3.28)$$

if we initialize \mathbf{d} as $\mathbf{d}^{(0)} = \rho^{-1} (\mathbf{y} - \mathbf{u}^{(0)})$. Letting $\tilde{\mathbf{u}} \triangleq \mathbf{u} - \mathbf{y}$ denote the split residual and substituting (3.28) into (3.7) leads to the following simplified linearized AL iterates:

$$\begin{cases} \mathbf{s}^{(k+1)} = \mathbf{A}'(\rho(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{y}) + (1 - \rho)\tilde{\mathbf{u}}^{(k)}) \\ \mathbf{x}^{(k+1)} \in \text{prox}_{(\rho^{-1}t)h}(\mathbf{x}^{(k)} - (\rho^{-1}t)\mathbf{s}^{(k+1)}) \\ \tilde{\mathbf{u}}^{(k+1)} = \frac{\rho}{\rho+1} (\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{y}) + \frac{1}{\rho+1} \tilde{\mathbf{u}}^{(k)}. \end{cases} \quad (3.29)$$

The net computational complexity of (3.29) per iteration reduces to one multiplication by \mathbf{A} , one multiplication by \mathbf{A}' , and one proximal mapping of h that often can be

solved non-iteratively or solved iteratively without using \mathbf{A} or \mathbf{A}' . Since the gradient of ℓ is $\mathbf{A}'(\mathbf{A}\mathbf{x} - \mathbf{y})$, letting $\mathbf{g} \triangleq \mathbf{A}'\tilde{\mathbf{u}}$ (a back-projection of the split residual) denote the split gradient, we can rewrite (3.29) as:

$$\begin{cases} \mathbf{s}^{(k+1)} = \rho \nabla \ell(\mathbf{x}^{(k)}) + (1 - \rho) \mathbf{g}^{(k)} \\ \mathbf{x}^{(k+1)} \in \text{prox}_{(\rho^{-1}t)h}(\mathbf{x}^{(k)} - (\rho^{-1}t) \mathbf{s}^{(k+1)}) \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1} \nabla \ell(\mathbf{x}^{(k+1)}) + \frac{1}{\rho+1} \mathbf{g}^{(k)}. \end{cases} \quad (3.30)$$

We call (3.30) the gradient-based linearized AL method because only the gradients of ℓ are used to perform the updates, and the net computational complexity of (3.30) per iteration becomes one gradient evaluation of ℓ and one proximal mapping of h .

We interpret the gradient-based linearized AL method (3.30) as a generalized proximal gradient descent of a regularized least-squares cost function Ψ with step size $\rho^{-1}t$ and search direction $\mathbf{s}^{(k+1)}$ that is a linear average of the gradient and split gradient of ℓ . A smaller ρ can lead to a larger step size. When $\rho = 1$, (3.30) happens to be the proximal gradient method or the iterative shrinkage/thresholding algorithm (ISTA) [52]. In other words, by using the linearized AL method, we can arbitrarily increase the step size of the proximal gradient method by decreasing ρ , thanks to the simple ρ -dependent correction of the search direction in (3.30). To have a concrete example, suppose all updates are exact, i.e., $\varepsilon_k = 0$ for all k . From (3.28) and Theorem III.2, we have $-\rho \mathbf{d}^{(k)} = \mathbf{u}^{(k)} - \mathbf{y} \rightarrow \mathbf{A}\hat{\mathbf{x}} - \mathbf{y} = \hat{\mathbf{z}}$ as $k \rightarrow \infty$. Furthermore, $(-\rho \mathbf{d}^{(0)}) - \hat{\mathbf{z}} = \mathbf{u}^{(0)} - \mathbf{A}\hat{\mathbf{x}}$. Therefore, with a reasonable initialization, e.g., $\mathbf{u}^{(0)} = \mathbf{A}\mathbf{x}^{(0)}$ and consequently, $\mathbf{g}^{(0)} = \nabla \ell(\mathbf{x}^{(0)})$, the constant C in (3.23) can be rewritten as a function of ρ :

$$C(\rho) = \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\rho^{-1}t}} + \frac{\|\mathbf{A}(\mathbf{x}^{(0)} - \hat{\mathbf{x}})\|_2}{\sqrt{2\rho}}. \quad (3.31)$$

This constant attains its minimum at

$$\rho_{\text{opt}} = \frac{\|\mathbf{A}(\mathbf{x}^{(0)} - \hat{\mathbf{x}})\|_2}{\sqrt{L} \|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2} \leq 1, \quad (3.32)$$

and it suggests that unity might be a reasonable upper bound on ρ for fast convergence. When the majorization is loose, i.e., $L \gg \|\mathbf{A}\|_2^2$, then $\rho_{\text{opt}} \ll 1$. In this case, the first term in (3.31) dominates C for $\rho_{\text{opt}} < \rho \leq 1$, and the upper bound of the

primal-dual gap becomes

$$\Omega(\mathbf{z}_k, \hat{\mathbf{x}}) - \Omega(\hat{\mathbf{z}}, \mathbf{x}_k) \leq \frac{C^2}{k} \approx O\left(\frac{L}{\rho^{-1}k}\right). \quad (3.33)$$

That is, comparing to the proximal gradient method ($\rho = 1$), the convergence rate (bound) of our proposed algorithm is accelerated by a factor of ρ^{-1} for $\rho_{\text{opt}} < \rho \leq 1$!

Finally, since the proposed gradient-based linearized AL method (3.30) requires only the gradients of ℓ to perform the updates, it is OS-acceleratable! For OS acceleration, we simply replace $\nabla\ell$ in (3.30) with $M\nabla\ell_m$ using the approximation (2.17) and incrementally perform (3.30) for M times as a complete iteration, thus leading to the final proposed OS-acceleratable linearized AL method (OS-LALM):

$$\begin{cases} \mathbf{s}^{(k,m+1)} = \rho M \nabla \ell_m(\mathbf{x}^{(k,m)}) + (1 - \rho) \mathbf{g}^{(k,m)} \\ \mathbf{x}^{(k,m+1)} \in \text{prox}_{(\rho^{-1}t)h}(\mathbf{x}^{(k,m)} - (\rho^{-1}t) \mathbf{s}^{(k,m+1)}) \\ \mathbf{g}^{(k,m+1)} = \frac{\rho}{\rho+1} M \nabla \ell_{m+1}(\mathbf{x}^{(k,m+1)}) + \frac{1}{\rho+1} \mathbf{g}^{(k,m)} \end{cases} \quad (3.34)$$

with $\mathbf{c}^{(k,M+1)} = \mathbf{c}^{(k+1)} = \mathbf{c}^{(k+1,1)}$ for $\mathbf{c} \in \{\mathbf{s}, \mathbf{x}, \mathbf{g}\}$ and $\ell_{M+1} = \ell_1$. The proposed algorithm (OS-LALM-FISTA) solving the X-ray CT image reconstruction problem (2.10) is described in Algorithm III.1. Note that when we majorize the quadratic AL penalty term using diagonal matrix \mathbf{D}_L in (2.13), the image update becomes a constrained weighted denoising problem:

$$\mathbf{x}^{(k,m+1)} \in \arg \min_{\mathbf{x} \in \Omega} \left\{ \frac{1}{2} \|\mathbf{x} - (\mathbf{x}^{(k,m)} - (\rho \mathbf{D}_L)^{-1} \mathbf{s}^{(k,m+1)})\|_{\rho \mathbf{D}_L}^2 + \mathbf{R}(\mathbf{x}) \right\} \quad (3.35)$$

that does not have simple analytical solution in general. Therefore, in Algorithm III.1, it is solved by using the fast iterative shrinkage/thresholding algorithm (FISTA) [25] starting from the previous update as a warm start, which is described in Algorithm III.2. Intuitively, the more FISTA iterations (i.e., the lower minimization errors), the faster convergence rate of OS-LALM-FISTA according to Theorem III.2. However, using more FISTA iterations for solving the denoising problem also increase the computational overhead of Algorithm III.1 per outer iteration. For example, suppose one use n FISTA iterations for solving the denoising problem, the net computational complexity of Algorithm III.1 per outer iterations is approximately one forward/back-projection pair and nM regularizer gradient evaluations! The computational overhead of computing regularizer gradients is not negligible when n or M is large.

Algorithm III.1 Proposed algorithm (OS-LALM-FISTA) solving (2.10)

Require: $\mathbf{x}^{(0)} \in \mathbb{R}^{n_p}$, $n_{\text{iter}} \geq 1$, $M \geq 1$, $\rho > 0$

- 1: Initialize $\mathbf{p}^{(0)} = \mathbf{g}^{(0)} = M \nabla \ell_M(\mathbf{x}^{(0)})$
 - 2: **for** $k = 0, 1, \dots, n_{\text{iter}} - 1$ **do**
 - 3: $\mathbf{c}^{(k,1)} = \mathbf{c}^{(k)}$ for $\mathbf{c} \in \{\mathbf{x}, \mathbf{p}, \mathbf{g}\}$
 - 4: **for** $m = 1, \dots, M$ **do**
 - 5: $\mathbf{s}^{(k,m+1)} = \rho \mathbf{p}^{(k,m)} + (1 - \rho) \mathbf{g}^{(k,m)}$
 - 6: Compute $\mathbf{x}^{(k,m+1)}$ by solving (3.35) using Algorithm III.2
 - 7: $\mathbf{p}^{(k,m+1)} = M \nabla \ell_m(\mathbf{x}^{(k,m+1)})$
 - 8: $\mathbf{g}^{(k,m+1)} = \frac{\rho}{\rho+1} \mathbf{p}^{(k,m+1)} + \frac{1}{\rho+1} \mathbf{g}^{(k,m)}$
 - 9: **end for**
 - 10: $\mathbf{c}^{(k+1)} = \mathbf{c}^{(k,M+1)}$ for $\mathbf{c} \in \{\mathbf{x}, \mathbf{p}, \mathbf{g}\}$
 - 11: **end for**
 - 12: **return** $\mathbf{x}^{(n_{\text{iter}})}$
-

Algorithm III.2 Fast iterative shrinkage/thresholding algorithm (FISTA) solving constrained denoising problem $\arg \min_{\mathbf{x} \in \Omega} \left\{ \frac{1}{2} \|\mathbf{x} - \mathbf{b}\|_{\mathbf{D}}^2 + \mathbf{R}(\mathbf{x}) \right\}$

Require: $\mathbf{x}^{(0)} \in \mathbb{R}^{n_p}$, $n_{\text{deno}} \geq 1$

- 1: Initialize $\mathbf{z}^{(0)} = \mathbf{x}^{(0)}$, $t_0 = 1$
 - 2: **for** $k = 0, 1, \dots, n_{\text{deno}} - 1$ **do**
 - 3: $\mathbf{x}^{(k+1)} = \left[\mathbf{z}^{(k)} - (\mathbf{D} + \mathbf{D}_R^{(k)})^{-1} (\mathbf{D}(\mathbf{x}^{(k)} - \mathbf{b}) + \nabla \mathbf{R}(\mathbf{z}^{(k)})) \right]_{\Omega}$
 - 4: $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2$
 - 5: $\mathbf{z}^{(k+1)} = \mathbf{x}^{(k+1)} + \frac{t_k - 1}{t_{k+1}} (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$
 - 6: **end for**
 - 7: **return** $\mathbf{x}^{(n_{\text{deno}})}$
-

Fortunately, in typical X-ray CT image reconstruction problems, the majorization is usually very loose (i.e., $\mathbf{D}_L \succ \mathbf{A}'\mathbf{W}\mathbf{A}$) probably due to the huge dynamic range of the statistical weight \mathbf{W} . This leads to a very strong weighting on the data-fitting term in (3.35), which equivalently diminishes greatly the regularization force in the denoising problem. In practice, the constrained denoising problem can be solved up to some acceptable tolerance within just one or two iterations. Therefore, an inexact version of the proposed algorithm (inexact OS-LALM) that uses only one FISTA denoising step to solve the constrained denoising problem is described in Algorithm III.3. Note that if we solve denoising problems with one FISTA (or ISTA) iteration, the net computational complexity of the algorithm is the same as conventional OS-based methods. By the way, the inexact algorithm is equivalent to replacing the regularizer \mathbf{R} by its SQS function $\check{\mathbf{R}}$ in a convergent ADMM (the linearized AL method). This particular approximation of ADMM is called the stochastic ADMM [51] in the machine learning literature. It is proven to be convergent only with a growing majorization (even when no gradient error is presented), thus leading to slower convergence rate, in the original paper [51]. However, a more recent paper showed that the stochastic ADMM is also convergent without the growing majorization in the noiseless case with rate $O(1/k)$ [53]. Hence, when $M = 1$, the inexact OS-LALM is convergent.

Algorithm III.3 Proposed algorithm (inexact OS-LALM) solving (2.10)

Require: $\mathbf{x}^{(0)} \in \mathbb{R}^{n_p}$, $n_{\text{iter}} \geq 1$, $M \geq 1$, $\rho > 0$

- 1: Initialize $\mathbf{p}^{(0)} = \mathbf{g}^{(0)} = M\nabla\ell_M(\mathbf{x}^{(0)})$
- 2: **for** $k = 0, 1, \dots, n_{\text{iter}} - 1$ **do**
- 3: $\mathbf{c}^{(k,1)} = \mathbf{c}^{(k)}$ for $\mathbf{c} \in \{\mathbf{x}, \mathbf{p}, \mathbf{g}\}$
- 4: **for** $m = 1, \dots, M$ **do**
- 5: $\mathbf{s}^{(k,m+1)} = \rho\mathbf{p}^{(k,m)} + (1 - \rho)\mathbf{g}^{(k,m)}$
- 6: $\mathbf{x}^{(k,m+1)} = \left[\mathbf{x}^{(k,m)} - (\rho\mathbf{D}_L + \mathbf{D}_R^{(k,m)})^{-1} (\mathbf{s}^{(k,m+1)} + \nabla\mathbf{R}(\mathbf{x}^{(k,m)})) \right]_{\Omega}$
- 7: $\mathbf{p}^{(k,m+1)} = M\nabla\ell_m(\mathbf{x}^{(k,m+1)})$
- 8: $\mathbf{g}^{(k,m+1)} = \frac{\rho}{\rho+1}\mathbf{p}^{(k,m+1)} + \frac{1}{\rho+1}\mathbf{g}^{(k,m)}$
- 9: **end for**
- 10: $\mathbf{c}^{(k+1)} = \mathbf{c}^{(k,M+1)}$ for $\mathbf{c} \in \{\mathbf{x}, \mathbf{p}, \mathbf{g}\}$
- 11: **end for**
- 12: **return** $\mathbf{x}^{(n_{\text{iter}})}$

3.2 Fast convergence with a downward continuation approach

One drawback of the AL method with a fixed AL penalty parameter ρ is the difficulty of finding the value that provides the fastest convergence. For example,

although the optimal AL penalty parameter ρ_{opt} in (3.32) minimizes the constant C in (3.31) that governs the convergence rate of the primal-dual gap, one cannot know its value beforehand because it depends on the solution $\hat{\mathbf{x}}$ of the problem. Intuitively, a smaller ρ is better because it leads to a larger step size. However, when the step size is too large, one can encounter overshoots and oscillations that slow down the convergence rate at first and when nearing the optimum. In fact, ρ_{opt} in (3.32) also suggests that ρ should not be arbitrarily small. Rather than estimating ρ_{opt} heuristically, we focus on using an iteration-dependent ρ , i.e., a continuation approach, for acceleration.

The classic continuation approach increases ρ as the algorithm proceeds so that the previous iterate can serve as a warm start for the subsequent worse-conditioned but more penalized inner minimization problem [54, Proposition 4.2.1]. However, in classic continuation approaches such as [39], one must specify both the initial value and the update rules of ρ . This introduces even more parameters to be tuned. In this section, unlike classic continuation approaches, we consider a downward continuation approach. The intuition is that, for a fixed ρ , the step length $\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|$ is typically a decreasing sequence because the gradient norm vanishes as we approach the optimum, and an increasing sequence ρ_k (i.e., a diminishing step size) would aggravate the shrinkage of step length and slow down the convergence rate. In contrast, a decreasing sequence ρ_k can compensate for step length shrinkage and accelerate convergence. Of course, ρ_k cannot decrease too fast; otherwise, the soaring step size might make the algorithm unstable or even divergent. To design a “good” decreasing sequence ρ_k for “effective” acceleration, we first analyze how our proposed algorithm (the one-subset version (3.30) for simplicity) behaves for different values of ρ .

3.2.1 Second-order system analysis with a fixed ρ

Consider a very simple quadratic problem:

$$\hat{\mathbf{x}} \in \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x}\|_2^2. \quad (3.36)$$

It is just an instance of (3.26) with $h = 0$ and $\mathbf{y} = \mathbf{0}$. One trivial solution of (3.36) is $\hat{\mathbf{x}} = \mathbf{0}$. To ensure a unique solution, we assume that $\mathbf{A}'\mathbf{A}$ is positive definite (for this analysis only). Let $\mathbf{A}'\mathbf{A}$ have eigenvalue decomposition $\mathbf{V}\mathbf{\Lambda}\mathbf{V}'$, where $\mathbf{\Lambda} \triangleq \text{diag}\{\lambda_i\}$ and $\mu = \lambda_1 \leq \dots \leq \lambda_n = L$. The updates generated by (3.30) that solve (3.36) can

be written as

$$\begin{cases} \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (1/L)(\mathbf{V}\Lambda\mathbf{V}'\mathbf{x}^{(k)} + (\rho^{-1} - 1)\mathbf{g}^{(k)}) \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1}\mathbf{V}\Lambda\mathbf{V}'\mathbf{x}^{(k+1)} + \frac{1}{\rho+1}\mathbf{g}^{(k)}. \end{cases} \quad (3.37)$$

Furthermore, letting $\bar{\mathbf{x}} = \mathbf{V}'\mathbf{x}$ and $\bar{\mathbf{g}} = \mathbf{V}'\mathbf{g}$, the linear system can be further diagonalized, and we can represent the i th components of $\bar{\mathbf{x}}$ and $\bar{\mathbf{g}}$ as

$$\begin{cases} \bar{x}_i^{(k+1)} = \bar{x}_i^{(k)} - (1/L)(\lambda_i\bar{x}_i^{(k)} + (\rho^{-1} - 1)\bar{g}_i^{(k)}) \\ \bar{g}_i^{(k+1)} = \frac{\rho}{\rho+1}\lambda_i\bar{x}_i^{(k+1)} + \frac{1}{\rho+1}\bar{g}_i^{(k)}. \end{cases} \quad (3.38)$$

Solving this system of recurrence relations of \bar{x}_i and \bar{g}_i , one can show that both \bar{x}_i and \bar{g}_i satisfy a second-order recursive system determined by the characteristic polynomial:

$$(1 + \rho)r^2 - 2(1 - \lambda_i/L + \rho/2)r + (1 - \lambda_i/L). \quad (3.39)$$

We analyze the roots of this polynomial to examine the convergence rate.

When $\rho = \rho_i^c$, where

$$\rho_i^c \triangleq 2\sqrt{\frac{\lambda_i}{L}\left(1 - \frac{\lambda_i}{L}\right)} \in (0, 1], \quad (3.40)$$

the characteristic equation has repeated roots. Hence, the system is critically damped, and \bar{x}_i and \bar{g}_i converge linearly to zero with convergence rate

$$r_i^c = \frac{1 - \lambda_i/L + \rho_i^c/2}{1 + \rho_i^c} = \sqrt{\frac{1 - \lambda_i/L}{1 + \rho_i^c}}. \quad (3.41)$$

When $\rho > \rho_i^c$, the characteristic equation has distinct real roots. Hence, the system is over-damped, and \bar{x}_i and \bar{g}_i converge linearly to zero with convergence rate that is governed by the dominant root

$$r_i^o(\rho) = \frac{1 - \lambda_i/L + \rho/2 + \sqrt{\rho^2/4 - \lambda_i/L(1 - \lambda_i/L)}}{1 + \rho}. \quad (3.42)$$

It is easy to check that $r_i^o(\rho_i^c) = r_i^c$, and r_i^o is non-decreasing. This suggests that the critically damped system always converges faster than the over-damped system. Finally, when $\rho < \rho_i^c$, the characteristic equation has complex roots. In this case, the system is under-damped, and \bar{x}_i and \bar{g}_i converge linearly to zero with convergence

rate

$$r_i^u(\rho) = \frac{1 - \lambda_i/L + \rho/2}{1 + \rho}, \quad (3.43)$$

and oscillate at the damped frequency $\psi_i/(2\pi)$, where

$$\cos\psi_i = \frac{1 - \lambda_i/L + \rho/2}{\sqrt{(1 + \rho)(1 - \lambda_i/L)}} \approx \sqrt{1 - \lambda_i/L} \quad (3.44)$$

when $\rho \approx 0$. Furthermore, by the small angle approximation:

$$\cos\sqrt{\theta} \approx 1 - \theta/2 \approx \sqrt{1 - \theta}, \quad (3.45)$$

if $\lambda_i \ll L$, $\psi_i \approx \sqrt{\lambda_i/L}$. Again, $r_i^u(\rho_i^c) = r_i^c$, but r_i^u behaves differently from r_i^o . Specifically, r_i^u is non-increasing if $\lambda_i/L < 1/2$, and it is non-decreasing otherwise. This suggests that the critically damped system converges faster than the under-damped system if $\lambda_i/L < 1/2$, but it can be slower otherwise. In sum, the critically damped system is optimal for those eigencomponents with smaller eigenvalues (i.e., $\lambda_i < L/2$), while for eigencomponents with larger eigenvalues (i.e., $\lambda_i > L/2$), the under-damped system is optimal.

In practice, the asymptotic convergence rate of the system is dominated by the smallest eigenvalue $\lambda_1 = \mu$. As the algorithm proceeds, only the component oscillating at the frequency $\psi_1/(2\pi)$ persists. Therefore, to achieve the fastest asymptotic convergence rate, we would like to choose

$$\rho^* = \rho_1^c = 2\sqrt{\frac{\mu}{L} \left(1 - \frac{\mu}{L}\right)} \in (0, 1]. \quad (3.46)$$

Figure 3.1 shows an example of the optimal asymptotic convergence rate ρ^* of a system with six distinct eigenvalues ($\mu = 0.05L, 0.1L, 0.3L, 0.7L, 0.9L$, and L). As we mentioned before, for those eigencomponents with smaller eigenvalues ($0.05L, 0.1L$, and $0.3L$), the critically damped system has the fastest asymptotic convergence rate, while eigencomponents with larger eigenvalues ($0.7L, 0.9L$, and L) attain the fastest asymptotic convergence rate in the under-damping regime. Moreover, when μ is less than $L/2$, the eigencomponent with the smallest eigenvalue μ determines the asymptotic convergence rate, i.e., $\min_{\rho} \{\max_i r_i(\rho)\} = \rho^*$, of the system as shown in (3.46). Unlike ρ_{opt} in (3.32), this choice of ρ does not depend on the initialization. It depends only on the geometry of the Hessian $\mathbf{A}'\mathbf{A}$. Furthermore, notice that both ρ_{opt} and ρ^* fall in the interval $(0, 1]$. Hence, although the linearized AL method converges for any $\rho > 0$, we consider only $\rho \leq 1$ in our downward continuation approach.

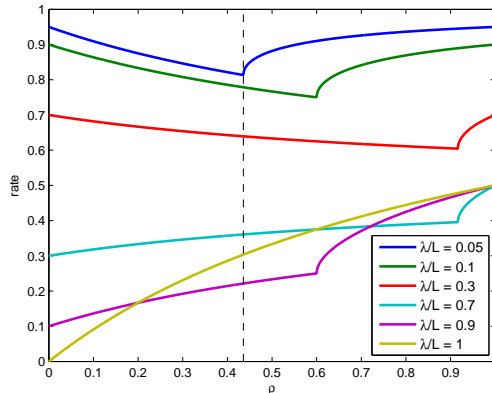


Figure 3.1: Optimal asymptotic convergence rate ρ^* of a system with six distinct eigenvalues ($\mu = 0.05L, 0.1L, 0.3L, 0.7L, 0.9L$, and L).

We can now interpret the classic (upward) continuation approach based on the second-order recursive system analysis. The classic continuation approach usually starts from a small ρ for better-conditioned inner minimization problem. Therefore, initially, the system is under-damped. Although the under-damped system has a slower asymptotic convergence rate, the oscillation can provide dramatic acceleration before the first zero-crossing of the oscillating components. We can think of the classic continuation approach as a greedy strategy that exploits the initial fast convergence rate of the under-damped system and carefully increases ρ to avoid oscillation and move toward the critical damping regime. However, this greedy strategy requires a “clever” update rule for increasing ρ . If ρ increases too fast, the acceleration ends prematurely; if ρ increases too slow, the system starts oscillating.

In contrast, we consider a more conservative strategy that starts from the over-damped regime, say, $\rho = 1$ as suggested in (3.46), and gradually reduces ρ to the optimal AL penalty parameter ρ^* . It sounds impractical at first because we do not know μ beforehand. To solve this problem, we adopt the adaptive restart proposed in [55] and generate a decreasing sequence ρ_k that starts from $\rho = 1$ and reaches ρ^* every time the algorithm restarts (i.e., resetting \mathbf{g} to be the current gradient of ℓ)! As mentioned before, the system oscillates at frequency $\psi_1/(2\pi)$ when it is under-damped. This oscillating behavior can also be observed from the trajectory of updates. For example,

$$\xi(k) \triangleq (\mathbf{g}^{(k)} - \nabla \ell(\mathbf{x}^{(k+1)}))' (\nabla \ell(\mathbf{x}^{(k+1)}) - \nabla \ell(\mathbf{x}^{(k)})) \quad (3.47)$$

oscillates at the frequency about ψ_1/π [55]. Hence, if we restart every time $\xi(k) > 0$, we restart the decreasing sequence about every $(\pi/2) \sqrt{L/\mu}$ iterations. Suppose we

restart at the l th iteration, we have the approximation $\sqrt{\mu/L} \approx \pi/(2l)$, and the ideal AL penalty parameter at the l th iteration should be

$$2\sqrt{\left(\frac{\pi}{2l}\right)^2\left(1 - \left(\frac{\pi}{2l}\right)^2\right)} = \frac{\pi}{l}\sqrt{1 - \left(\frac{\pi}{2l}\right)^2}. \quad (3.48)$$

Finally, the proposed downward continuation approach has the form (3.30), while we replace every ρ in (3.30) with

$$\rho_r = \begin{cases} 1 & , \text{ if } r = 0 \\ \max\left\{\frac{\pi}{r+1}\sqrt{1 - \left(\frac{\pi}{2r+2}\right)^2}, \rho_{\min}\right\} & , \text{ otherwise,} \end{cases} \quad (3.49)$$

where r is a counter that starts from zero, increases by one, and is reset to zero whenever $\xi(k) > 0$. For the M -subset version (3.34), we usually do not use adaptive restart because gradients are inaccurate in this case, and one might restart prematurely due to the false restart signal. The lower bound ρ_{\min} is a small positive number for guaranteeing convergence. Note that ADMM is convergent if ρ is non-increasing and bounded below away from zero [56, Corollary 4.2]. As shown in Section 3.1.1, the linearized AL method is in fact a convergent ADMM. Therefore, we can ensure convergence (of the one-subset version) of the proposed downward continuation approach if we set a non-zero lower bound for ρ_l , e.g., $\rho_{\min} = 10^{-3}$ in our experiments. The proposed algorithm (inexact OS-LALM-c) is described in Algorithm III.4.

Algorithm III.4 Proposed algorithm (inexact OS-LALM-c) solving (2.10)

Require: $\mathbf{x}^{(0)} \in \mathbb{R}^{n_p}$, $n_{\text{iter}} \geq 1$, $M \geq 1$

- 1: Initialize $\mathbf{p}^{(0)} = \mathbf{g}^{(0)} = M\nabla\ell_M(\mathbf{x}^{(0)})$, $r = 1$
- 2: **for** $k = 0, 1, \dots, n_{\text{iter}} - 1$ **do**
- 3: $\mathbf{c}^{(k,1)} = \mathbf{c}^{(k)}$ for $\mathbf{c} \in \{\mathbf{x}, \mathbf{p}, \mathbf{g}\}$
- 4: **for** $m = 1, \dots, M$ **do**
- 5: Compute ρ_r using (3.49)
- 6: $\mathbf{s}^{(k,m+1)} = \rho_r \mathbf{p}^{(k,m)} + (1 - \rho_r) \mathbf{g}^{(k,m)}$
- 7: $\mathbf{x}^{(k,m+1)} = \left[\mathbf{x}^{(k,m)} - (\rho_r \mathbf{D}_L + \mathbf{D}_R^{(k,m)})^{-1} (\mathbf{s}^{(k,m+1)} + \nabla R(\mathbf{x}^{(k,m)})) \right]_{\Omega}$
- 8: $\mathbf{p}^{(k,m+1)} = M\nabla\ell_m(\mathbf{x}^{(k,m+1)})$
- 9: $\mathbf{g}^{(k,m+1)} = \frac{\rho_r}{\rho_r+1} \mathbf{p}^{(k,m+1)} + \frac{1}{\rho_r+1} \mathbf{g}^{(k,m)}$
- 10: $r = r + 1$
- 11: **end for**
- 12: $\mathbf{c}^{(k+1)} = \mathbf{c}^{(k,M+1)}$ for $\mathbf{c} \in \{\mathbf{x}, \mathbf{p}, \mathbf{g}\}$
- 13: **end for**
- 14: **return** $\mathbf{x}^{(n_{\text{iter}})}$

Note that ρ_r in (3.49) is the same for any \mathbf{A} . The adaptive restart condition

takes care of the dependence on \mathbf{A} . When h is non-zero and/or $\mathbf{A}'\mathbf{A}$ is not positive definite, our analysis above does not hold. However, the downward continuation approach works well in practice for CT. One possible explanation is that the cost function can usually be well approximated by a quadratic near the optimum when the minimization problem is well-posed and h is locally quadratic.

3.2.2 Experimental results: low-dose CT

This section reports numerical results for 3D X-ray CT image reconstruction from real low-dose CT scans with different geometries using various OS-based methods, including

- **OS-SQS- M** : the standard OS algorithm [10] with M subsets (Algorithm II.3),
- **OS-Nes05- M** : the OS+momentum algorithm [27] based on Nesterov’s fast gradient method [24] with M subsets (Algorithm II.4),
- **OS-rNes05- M - γ** : the relaxed OS+momentum algorithm [28] based on Nesterov’s fast gradient method [24] and Devolder’s growing diagonal majorizer with M subsets (Algorithm II.5),
- **OS-LALM- M - ρ -1**: the proposed algorithm using a fixed AL penalty parameter ρ with M subsets and one FISTA iterations for solving the inner constrained denoising problem (Algorithm III.3), and
- **OS-LALM- M -c-1**: the proposed algorithm using the downward continuation approach with M subsets and one FISTA iterations for solving the inner constrained denoising problem (Algorithm III.4).

All algorithms listed above have approximately the same computational complexity, i.e., total number of flops, with one forward/back-projection pair and M regularizer gradient evaluations per iteration. Therefore, comparing the convergence rate as a function of iteration is fair. We measured the convergence rate using the RMS difference (in the region of interest) between the reconstructed image $\mathbf{x}^{(k)}$ and the almost converged reference reconstruction \mathbf{x}^* that is generated by running several iterations of the standard OS+momentum algorithm with a small M , followed by 2000 iterations of a convergent (i.e., one-subset) FISTA with adaptive restart [55].

Since OS-based methods are in fact incremental gradient methods, the order of incremental image update affects both the convergence rate and stability of OS-based algorithms. In our experiments, we choose the bit-reversal order [57] that

heuristically minimizes the subset gradient variance as in other OS-based algorithms [10, 27, 28, 49]. The number of subsets M also affects the stability of OS-based algorithms. When M is too large, OS algorithms typically become unstable, and one can observe artifacts in the reconstructed image. Therefore, finding an appropriate number of subsets is very important. Since errors of OS-based algorithms come from the gradient approximation using subset gradients, artifacts might be suppressed using a better gradient approximation. Intuitively, to have an acceptable gradient approximation, each voxel in a subset should be sampled by a minimum number of views s . For simplicity, we consider the central voxel in the transaxial plane. In axial CT, the views are uniformly distributed in each subset, so we want

$$\frac{1}{M_{\text{axial}}} \cdot (\text{number of views}) \geq s_{\text{axial}}. \quad (3.50)$$

This leads to our maximum number of subsets for axial CT:

$$M_{\text{axial}} \leq (\text{number of views}) \cdot \frac{1}{s_{\text{axial}}}. \quad (3.51)$$

In helical CT, the situation is more complicated. Since the X-ray source moves in the z -direction, a central voxel is only covered by $d_{\text{so}}/(p \cdot d_{\text{sd}})$ turns, where p is the pitch, d_{so} denotes the distance from the X-ray source to the isocenter, and d_{sd} denotes the distance from the X-ray source to the detector. Therefore, we want

$$\frac{1}{M_{\text{helical}}} \cdot (\text{number of views per turn}) \cdot \frac{d_{\text{so}}}{p \cdot d_{\text{sd}}} \geq s_{\text{helical}}. \quad (3.52)$$

This leads to our maximum number of subsets for helical CT:

$$M_{\text{helical}} \leq (\text{number of views per turn}) \cdot \frac{d_{\text{so}}}{p \cdot s_{\text{helical}} \cdot d_{\text{sd}}}. \quad (3.53)$$

Note that the maximum number of subsets for helical CT M_{helical} is inversely proportional to the pitch p . That is, the maximum number of subsets for helical CT decreases for a larger pitch. We set $s_{\text{axial}} \approx 40$ and $s_{\text{helical}} \approx 24$ for the proposed algorithm in our experiments.

3.2.2.1 Shoulder scan

In this experiment, we reconstructed a $512 \times 512 \times 109$ image from a shoulder region helical CT scan, where the sinogram has size $888 \times 32 \times 7146$ and pitch 0.5. The suggested maximum number of subsets according to (3.53) is about 40. Figure 3.2

shows the cropped images from the central transaxial plane of the initial FBP image, the reference reconstruction, and the reconstructed image using the proposed algorithm (OS-LALM-40-c-1) at the 30th iteration (i.e., after 30 forward/back-projection pairs). As can be seen in Figure 3.2, the reconstructed image using the proposed algorithm looks almost the same as the reference reconstruction in the display window from 800 to 1200 Hounsfield unit (HU). The reconstructed image using the standard OS+momentum algorithm (not shown here) also looks quite similar to the reference reconstruction.

To see the difference between the standard OS+momentum algorithm and our proposed algorithm, Figure 3.3 shows the difference images, i.e., $\mathbf{x}^{(30)} - \mathbf{x}^*$, for different OS-based algorithms. We can see that the standard OS algorithm (with both 20 and 40 subsets) exhibits visible streak artifacts and structured high frequency noise in the difference image. When $M = 20$, the difference images look similar for the standard OS+momentum algorithm and our proposed algorithm, although that of the standard OS+momentum algorithm is slightly structured and non-uniform. When $M = 40$, the difference image for our proposed algorithm remains uniform, whereas some noise-like OS artifacts appear in the standard OS+momentum algorithm’s difference image. The OS artifacts in the reconstructed image using the standard OS+momentum algorithm become worse when M increases, e.g., $M = 80$. This shows the better gradient error tolerance of our proposed algorithm when OS is used, probably due to the way we compute the search direction.

As can be seen in (3.30), the search direction \mathbf{s} is a linear average of the current gradient and the split gradient of ℓ . Specifically, (3.30) computes the search direction using a low-pass infinite-impulse-response (IIR) filter (across iterations), and therefore, the gradient error might be suppressed by the low-pass filter, leading to a more stable reconstruction. A similar averaging technique (with a low-pass finite-impulse-response or FIR filter) is also used in the stochastic average gradient (SAG) method [58, 59] for acceleration and stabilization. In comparison, the standard OS+momentum algorithm computes the search direction using only the current gradient (of the auxiliary image), so the gradient error accumulates when OS is used, providing a less stable reconstruction.

Figure 3.4 shows the convergence rate curves (RMS differences between the reconstructed image $\mathbf{x}^{(k)}$ and the reference reconstruction \mathbf{x}^* as a function of iteration) using OS-based algorithms with (a) 20 subsets and (b) 40 subsets, respectively. By exploiting the linearized AL method, the proposed algorithm accelerates the standard OS algorithm remarkably. As mentioned in Section 3.1.2, a smaller ρ can provide

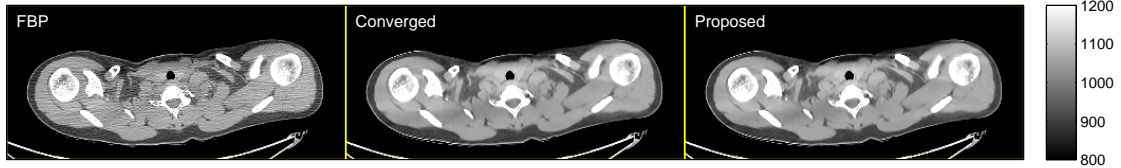


Figure 3.2: Shoulder scan: cropped images (displayed from 800 to 1200 HU) from the central transaxial plane of the initial FBP image $\mathbf{x}^{(0)}$ (left), the reference reconstruction \mathbf{x}^* (center), and the reconstructed image using the proposed algorithm (OS-LALM-40-c-1) at the 30th iteration $\mathbf{x}^{(30)}$ (right).

greater acceleration due to the increased step size. We can see the approximate 5, 10, and 20 times acceleration (comparing to the standard OS algorithm, i.e., $\rho = 1$) using $\rho = 0.2$, 0.1, and 0.05 in both figures. Note that too large step sizes can cause overshoots in early iterations. For example, the proposed algorithm with $\rho = 0.05$ shows slower convergence rate in first few iterations but decreases more rapidly later. This trade-off can be overcome by using our proposed downward continuation approach. As can be seen in Figure 3.4, the proposed algorithm using downward continuation reaches the lowest RMS differences (lower than 1 HU) within only 30 iterations! Furthermore, the slightly higher RMS difference of the standard OS+momentum algorithm with 40 subsets shows evidence of OS artifacts.

Figure 3.5 demonstrates the effectiveness of solving the inner constrained denoising problem using FISTA (for X-ray CT image reconstruction). As can be seen in Figure 3.5, the convergence rate improves only slightly when more FISTA iterations are used for solving the inner constrained denoising problem. In practice, one FISTA iteration, i.e., $n = 1$, suffices for fast and “convergent” X-ray CT image reconstruction. When the inner constrained denoising problem is more difficult to solve (e.g., when regularization term dominates the cost function), one might want to introduce an additional split variable for the regularizer as in [8] at the cost of higher memory burden, thus leading to a “high-memory” version of OS-LALM. We will discuss this variant of the proposed algorithm in the next section.

3.2.2.2 Truncated abdomen scan

In this experiment, we reconstructed a $600 \times 600 \times 239$ image from an abdomen region helical CT scan with transaxial truncation, where the sinogram has size $888 \times 64 \times 3516$ and pitch 1.0. The suggested maximum number of subsets according to (3.53) is about 20. Figure 3.6 shows the cropped images from the central transaxial plane of the initial FBP image, the reference reconstruction, and the reconstructed

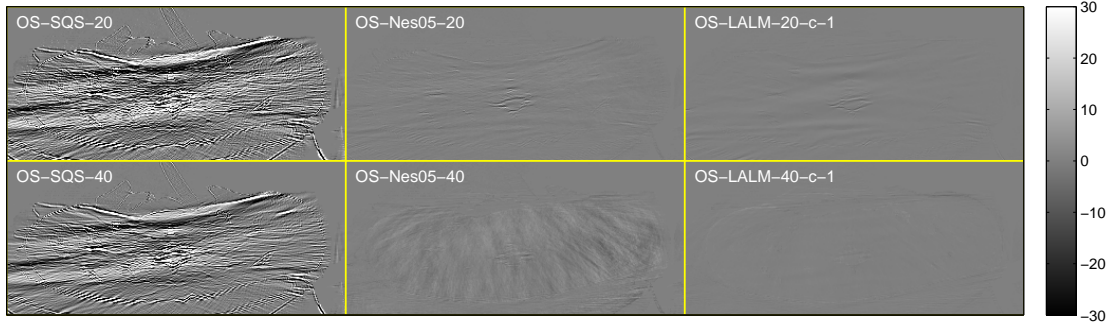


Figure 3.3: Shoulder scan: cropped difference images (displayed from -30 to 30 HU) from the central transaxial plane of $\mathbf{x}^{(30)} - \mathbf{x}^*$ using OS-based algorithms.

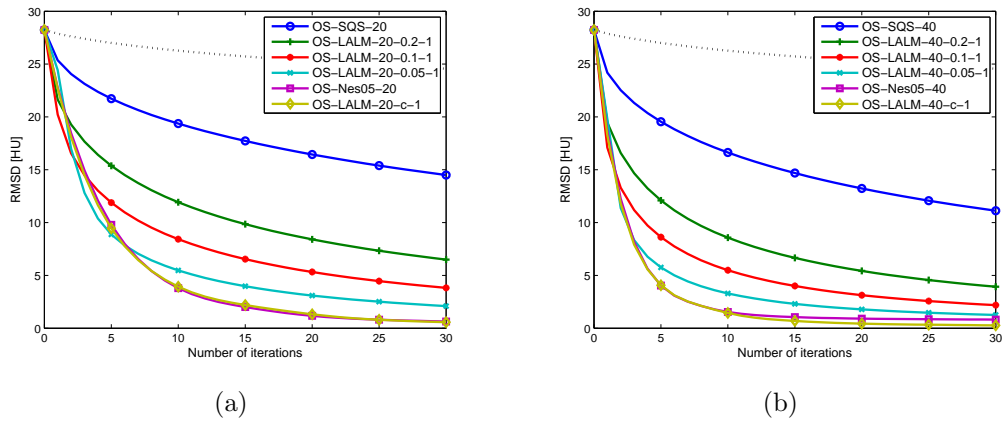


Figure 3.4: Shoulder scan: RMS differences between the reconstructed image $\mathbf{x}^{(k)}$ and the reference reconstruction \mathbf{x}^* as a function of iteration using OS-based algorithms with (a) 20 subsets and (b) 40 subsets, respectively. The dotted lines show the RMS differences using the standard OS algorithm with one subset as the baseline convergence rate.

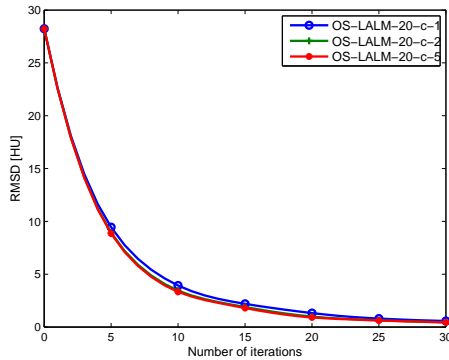


Figure 3.5: Shoulder scan: RMS differences between the reconstructed image $\mathbf{x}^{(k)}$ and the reference reconstruction \mathbf{x}^* as a function of iteration using the proposed algorithm with different number of FISTA iterations n (1, 2, and 5) for solving the inner constrained denoising problem.

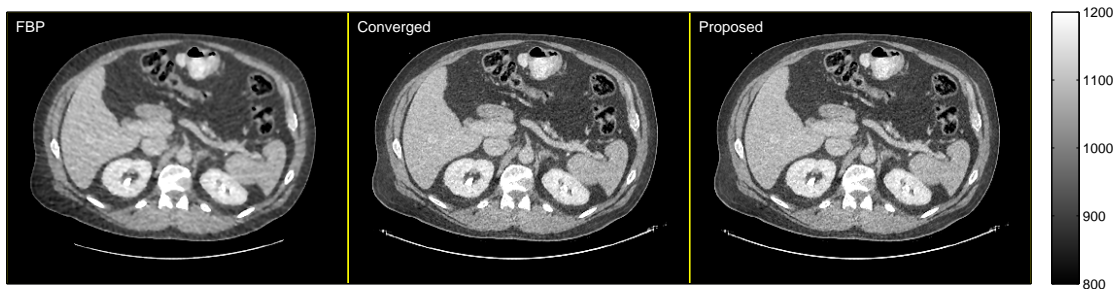


Figure 3.6: Truncated abdomen scan: cropped images (displayed from 800 to 1200 HU) from the central transaxial plane of the initial FBP image $\mathbf{x}^{(0)}$ (left), the reference reconstruction \mathbf{x}^* (center), and the reconstructed image using the proposed algorithm (OS-LALM-20-c-1) at the 30th iteration $\mathbf{x}^{(30)}$ (right).

image using the proposed algorithm (OS-LALM-20-c-1) at the 30th iteration. This experiment demonstrates how different OS-based algorithms behave when the number of subsets exceeds the suggested maximum number of subsets. Figure 3.7 shows the difference images for different OS-based algorithms with 10, 20, and 40 subsets. As can be seen in Figure 3.7, the proposed algorithm works best for $M = 20$; when M is larger ($M = 40$), ripples and light OS artifacts appear due to the insufficient sampling rate. However, it is still much better than the standard OS+momentum algorithm [27]. In fact, the OS artifacts in the reconstructed image using the standard OS+momentum algorithm with 40 subsets are visible with the naked eye in the display window from 800 to 1200 HU. The convergence rate curves in Figure 3.8 support our observation. In sum, the proposed algorithm exhibits fast convergence rate and excellent gradient error tolerance even in the case with truncation.

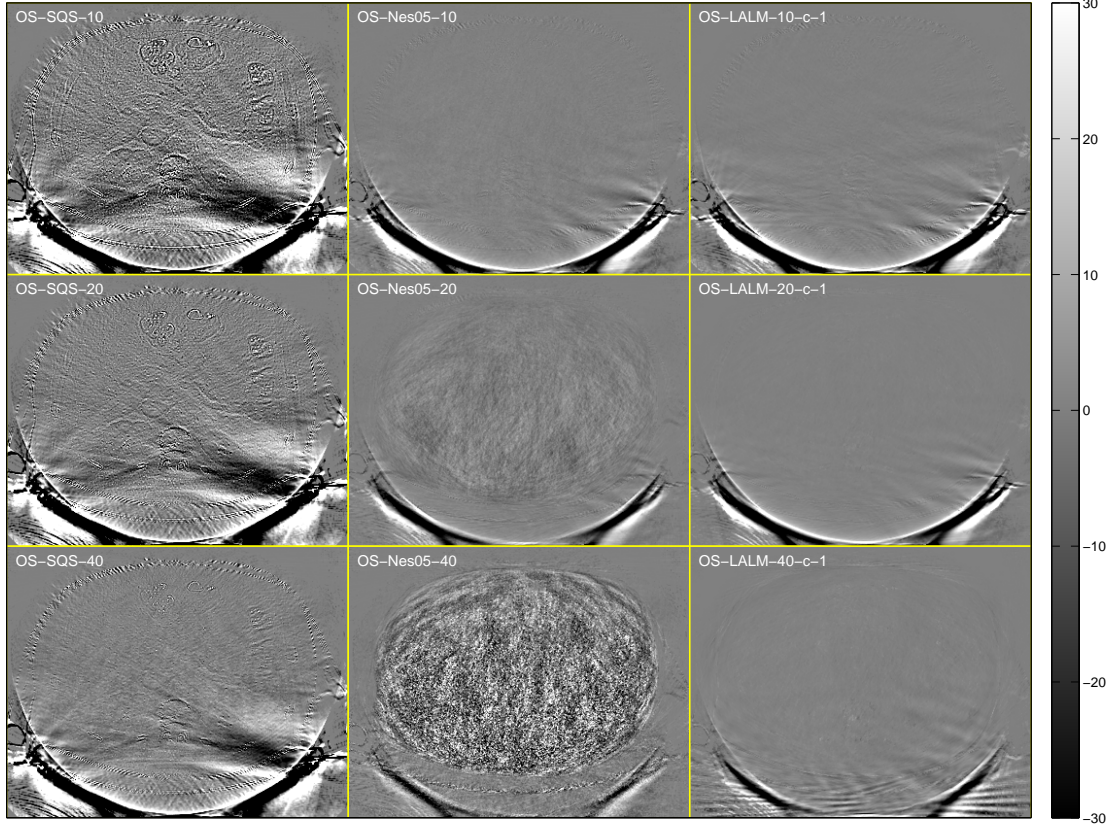


Figure 3.7: Truncated abdomen scan: cropped difference images (displayed from -30 to 30 HU) from the central transaxial plane of $\mathbf{x}^{(30)} - \mathbf{x}^*$ using OS-based algorithms.

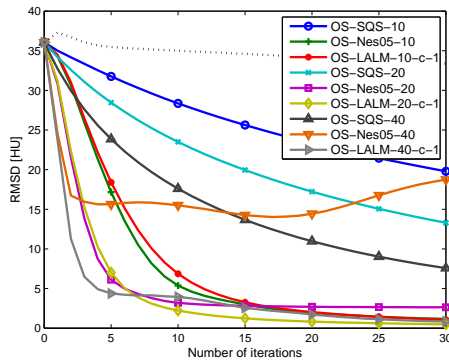


Figure 3.8: Truncated abdomen scan: RMS differences between the reconstructed image $\mathbf{x}^{(k)}$ and the reference reconstruction \mathbf{x}^* as a function of iteration using OS-based algorithms with 10, 20, and 40 subsets. The dotted line shows the RMS differences using the standard OS algorithm with one subset as the baseline convergence rate.

3.2.2.3 GE performance phantom

In this experiment, we reconstructed a $1024 \times 1024 \times 90$ image from the GE performance phantom (GEPP) axial CT scan, where the sinogram has size $888 \times 64 \times 984$. The suggested maximum number of subsets according to (3.51) is about 24. Figure 3.9 shows the cropped images from the central transaxial plane of the initial FBP image, the reference reconstruction, and the reconstructed image using the proposed algorithm (OS-LALM-24-c-1) at the 30th iteration. Again, the reconstructed image using the proposed algorithm at the 30th iteration is very similar to the reference reconstruction.

The goal of this experiment is to evaluate the gradient error tolerance of our proposed algorithm and the recently proposed relaxed OS+momentum algorithm [28] that trades reconstruction stability with speed by introducing relaxed momentum (i.e., a growing diagonal majorizer). We vary γ to investigate different amounts of relaxation. When $\gamma = 0$, the relaxed OS+momentum algorithm reverts to the standard OS+momentum algorithm. A larger γ can lead to a more stable reconstruction but slower convergence. Figure 3.10 and Figure 3.11 show the difference images and convergence rate curves using these OS-based algorithms, respectively. As can be seen in Figure 3.10 and 3.11, the standard OS+momentum algorithm has even more OS artifacts than the standard OS algorithm, probably because 24 subsets in axial CT is too aggressive for the standard OS+momentum algorithm, and we can see clear OS artifacts in the difference image and large limit cycle in the convergence rate curve. The OS artifacts are less visible as γ increases. The case $\gamma = 0.005$ achieves the best trade-off between OS artifact removal and fast convergence rate. When γ is even larger, the relaxed OS+momentum algorithm is significantly slowed down although the difference image looks quite uniform (with some structured high frequency noise). The proposed OS-LALM algorithm avoids the need for such parameter tuning; one only needs to choose the number of subsets M . Furthermore, even for $\gamma = 0.005$, the relaxed OS+momentum algorithm still has more visible OS artifacts and slower convergence rate comparing to our proposed algorithm.

3.3 Efficient memory usage when considering additional splits

As mentioned in the previous section, due to the inexact updates, the proposed algorithm can be slow when the regularization term dominates the cost function. To have a concrete example, let's take a look at the gradient-based image update of Algorithm III.4. The voxel-dependent step sizes are determined by the diagonal entries of

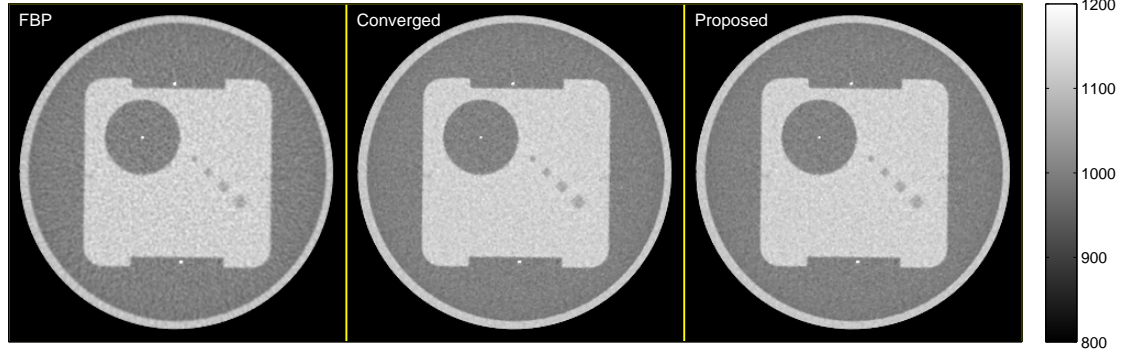


Figure 3.9: GE performance phantom: cropped images (displayed from 800 to 1200 HU) from the central transaxial plane of the initial FBP image $\mathbf{x}^{(0)}$ (left), the reference reconstruction \mathbf{x}^* (center), and the reconstructed image using the proposed algorithm (OS-LALM-24-c-1) at the 30th iteration $\mathbf{x}^{(30)}$ (right).

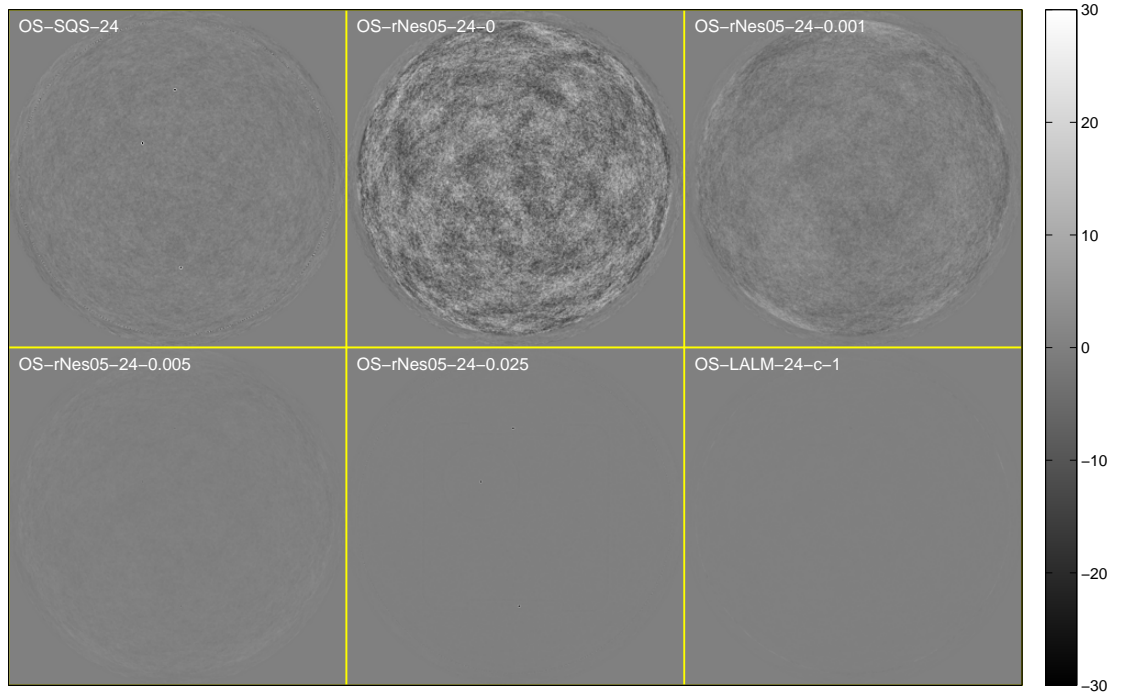


Figure 3.10: GE performance phantom: cropped difference images (displayed from -30 to 30 HU) from the central transaxial plane of $\mathbf{x}^{(30)} - \mathbf{x}^*$ using the relaxed OS+momentum algorithm and the proposed algorithm.

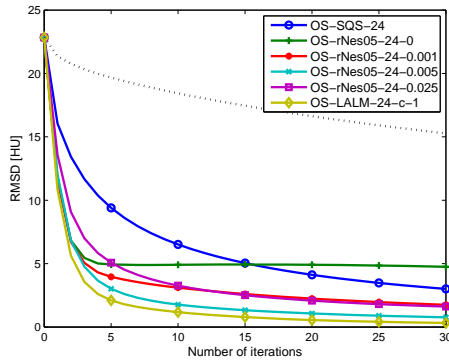


Figure 3.11: GE performance phantom: RMS differences between the reconstructed image $\mathbf{x}^{(k)}$ and the reference reconstruction \mathbf{x}^* as a function of iteration using the relaxed OS+momentum algorithm and the proposed algorithm with 24 subsets. The dotted line shows the RMS differences using the standard OS algorithm with one subset as the baseline convergence rate.

$\rho_r \mathbf{D}_L + \mathbf{D}_R^{(k)}$. Suppose the regularization force is large (hence, diagonal entries of $\mathbf{D}_R^{(k)}$ are large). The effect of continuation (decrease of ρ_r) is marginal because the latter term dominates the step sizes. One possible way to solve this problem is to introduce an additional auxiliary variable that separates the regularization effect from image updates, just like the high-memory ADMM [8]. By introducing the high-memory $\mathbf{v} = \mathbf{C}\mathbf{x}$ auxiliary variable, in (2.28), the non-quadratic regularizer \mathbf{R} whose Hessian can have very high (or even infinite) curvature is replaced by a quadratic AL penalty term with Hessian $\eta \mathbf{C}'\mathbf{C}$ that has an η -scalable curvature. This looks very promising, except for the inevitable high memory requirement.

3.3.1 Split OS-LALM: OS-LALM with additional split

Consider a regularized least-squares problem:

$$\hat{\mathbf{x}} \in \arg \min_{\mathbf{x} \in \Omega} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \Phi(\Theta\mathbf{x}) \right\}, \quad (3.54)$$

where \mathbf{A} is the system matrix, \mathbf{y} is the noisy measurement, Θ is an analysis regularization matrix, Φ is some convex (and possibly non-smooth) potential function defined in (2.8), and Ω denotes the convex set for a box constraint (usually the non-negativity constraint) on \mathbf{x} . For example, in (anisotropic) TV-regularized X-ray CT image reconstruction problems, \mathbf{A} is the weighted forward projection matrix, \mathbf{y} is the weighted noisy sinogram, Θ is a finite difference matrix, and Φ is an ℓ_1 -norm, probably with some weighting. The minimization problem (3.54) is non-trivial in general

since Θ might not be an identity matrix, and Φ can be non-smooth.

To solve (3.54), consider solving an equivalent constrained minimization problem:

$$(\hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{v}}) \in \arg \min_{\mathbf{x}, \mathbf{u}, \mathbf{v}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{u}\|_2^2 + \Phi(\mathbf{v}) + \iota_\Omega(\mathbf{x}) \right\} \text{ s.t. } \mathbf{u} = \mathbf{A}\mathbf{x}, \mathbf{v} = \Theta\mathbf{x} \quad (3.55)$$

by finding a saddle-point of the corresponding scaled augmented Lagrangian (with a weighted quadratic AL penalty term [34, 56]):

$$\begin{aligned} \mathcal{L}_A(\mathbf{x}, \mathbf{u}, \mathbf{v}, \mathbf{d}, \mathbf{e}; \rho, \eta) &\triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{u}\|_2^2 + \Phi(\mathbf{v}) + \iota_\Omega(\mathbf{x}) \\ &\quad + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \mathbf{u} - \mathbf{d}\|_2^2 + \frac{\eta}{2} \|\Theta\mathbf{x} - \mathbf{v} - \mathbf{e}\|_{\mathbf{D}_{\text{reg}}\mathbf{D}_\phi\{\Theta\mathbf{x}\}}^2 \end{aligned} \quad (3.56)$$

with the linearized AL method described in Section 3.1.1:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg \min_{\mathbf{x}} \left\{ \iota_\Omega(\mathbf{x}) + \check{\theta}_k(\mathbf{x}; \mathbf{x}^{(k)}) + \check{\phi}_k(\mathbf{x}; \mathbf{x}^{(k)}) \right\} \\ \mathbf{u}^{(k+1)} \in \arg \min_{\mathbf{u}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{u}\|_2^2 + \frac{\rho}{2} \|\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\|_2^2 \right\} \\ \mathbf{v}^{(k+1)} \in \arg \min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \|\Theta\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)}\|_{\mathbf{D}_{\text{reg}}\mathbf{D}_\phi\{\Theta\mathbf{x}^{(k)}\}}^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \Theta\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)}, \end{cases} \quad (3.57)$$

where ι_Ω is the characteristic function of the convex set Ω that handles the box constraint on \mathbf{x} , \mathbf{d} and \mathbf{e} are the scaled Lagrange multipliers of the auxiliary variables \mathbf{u} and \mathbf{v} , respectively, and $\rho, \eta > 0$ are the corresponding AL penalty parameters. We weight the second quadratic AL penalty term by $\eta \mathbf{D}_{\text{reg}}\mathbf{D}_\phi\{\Theta\mathbf{x}^{(k)}\}$ to impose more penalty for finite difference that has higher curvature in a scalable way. The functions $\check{\theta}_k(\mathbf{x}; \mathbf{x}^{(k)})$ and $\check{\phi}_k(\mathbf{x}; \mathbf{x}^{(k)})$ are two SQS functions that majorize

$$\theta_k(\mathbf{x}) \triangleq \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}\|_2^2 \quad (3.58)$$

and

$$\phi_k(\mathbf{x}) \triangleq \frac{\eta}{2} \|\Theta\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)}\|_{\mathbf{D}_{\text{reg}}\mathbf{D}_\phi\{\Theta\mathbf{x}^{(k)}\}}^2 \quad (3.59)$$

at $\mathbf{x} = \mathbf{x}^{(k)}$ with diagonal Hessian matrices $\rho\mathbf{D}_L$ and $\eta\mathbf{D}_R^{(k)}$, respectively.

As can be seen in (3.57), introducing an additional auxiliary variable \mathbf{v} only modestly changes the updates from the one-split linearized AL iterates (3.7). By letting $h_k \triangleq \iota_\Omega + \check{\phi}_k$, the two-split linearized AL iterates (3.57) become the one-split linearized AL iterates (3.7) with an iteration-dependent regularization term h_k , where

the effect of h_k is fully determined by the \mathbf{v} - and \mathbf{e} -updates in (3.57)! Hence, we can easily rewrite the two-split linearized AL iterates (3.57) to the two-split gradient-based linearized AL iterates:

$$\begin{cases} \mathbf{s}^{(k+1)} = \rho \nabla \ell(\mathbf{x}^{(k)}) + (1 - \rho) \mathbf{g}^{(k)} \\ \mathbf{x}^{(k+1)} \in \text{prox}_{h_k, \rho \mathbf{D}_L}(\mathbf{x}^{(k)} - (\rho \mathbf{D}_L)^{-1} \mathbf{s}^{(k+1)}) \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1} \nabla \ell(\mathbf{x}^{(k+1)}) + \frac{1}{\rho+1} \mathbf{g}^{(k)} \\ \mathbf{v}^{(k+1)} \in \text{prox}_{\Phi, \eta \mathbf{D}_{\text{reg}} \mathbf{D}_\phi \{\Theta \mathbf{x}^{(k)}\}}(\Theta \mathbf{x}^{(k+1)} - \mathbf{e}^{(k)}) \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \Theta \mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)}, \end{cases} \quad (3.60)$$

where ℓ denotes the quadratic data-fitting term in (3.54), and $\text{prox}_{\varphi, \mathbf{D}}$ denotes the (diagonal-)generalized proximal mapping of φ with diagonal weighting \mathbf{D} defined as:

$$\text{prox}_{\varphi, \mathbf{D}}(\mathbf{z}) \triangleq \arg \min_{\mathbf{x}} \left\{ \varphi(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_{\mathbf{D}}^2 \right\}. \quad (3.61)$$

Since both ι_Ω and $\check{\phi}_k$ are separable, the \mathbf{x} -update of the two-split gradient-based linearized AL iterates (3.60) has a closed-form solution:

$$\mathbf{x}^{(k+1)} = \left[\mathbf{x}^{(k)} - (\rho \mathbf{D}_L + \eta \mathbf{D}_R^{(k)})^{-1} (\mathbf{s}^{(k+1)} + \boldsymbol{\sigma}^{(k+1)}) \right]_{\Omega}, \quad (3.62)$$

where

$$\boldsymbol{\sigma}^{(k+1)} \triangleq \eta \Theta' \mathbf{D}_{\text{reg}} \mathbf{D}_\phi \{\Theta \mathbf{x}^{(k)}\} (\Theta \mathbf{x}^{(k)} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)}) \quad (3.63)$$

is the search direction attributed to the regularization term. Finally, the two-split gradient-based linearized AL method (3.60) is an extension of the one-split gradient-based linearized AL method, so we can accelerate it by using OS and the downward continuation approach. When OS is used for acceleration, we call our proposed algorithm split OS-LALM, by an analogy of the SB method. The proposed algorithm (split OS-LALM-c) solving (2.10) is described in Algorithm III.5.

3.3.2 Memory-efficient implementation

As mentioned before, the issue of the high-memory ADMM [8] is the remarkable memory overhead. To solve this problem, we consider majorizing the (smooth and L_Φ -Lipschitz) potential function Φ in Step 11 of Algorithm III.5 by its SQS function

$$\check{\Phi}(\mathbf{v}; \Theta \mathbf{x}^{(k, m+1)}) \propto \mathbf{v}' \nabla \Phi(\Theta \mathbf{x}^{(k, m+1)}) + \frac{1}{2} \|\mathbf{v} - \Theta \mathbf{x}^{(k, m+1)}\|_{\mathbf{D}_{\text{reg}} \mathbf{D}_\phi \{\Theta \mathbf{x}^{(k, m+1)}\}}^2 \quad (3.64)$$

Algorithm III.5 Proposed algorithm (split OS-LALM-c) solving (2.10)

Require: $\mathbf{x}^{(0)} \in \mathbb{R}^{n_p}$, $n_{\text{iter}} \geq 1$, $M \geq 1$

- 1: Initialize $\mathbf{p}^{(0)} = \mathbf{g}^{(0)} = M \nabla \ell_M(\mathbf{x}^{(0)})$, $r = 1$,
 $\mathbf{v}^{(0)} = \text{prox}_{\Phi, \eta \mathbf{D}_{\text{reg}} \mathbf{D}_{\phi}}\{\Theta \mathbf{x}^{(0)}\}(\Theta \mathbf{x}^{(0)})$, $\mathbf{e}^{(0)} = -\Theta \mathbf{x}^{(0)} + \mathbf{v}^{(0)}$
- 2: **for** $k = 0, 1, \dots, n_{\text{iter}} - 1$ **do**
- 3: $\mathbf{c}^{(k,1)} = \mathbf{c}^{(k)}$ for $\mathbf{c} \in \{\mathbf{x}, \mathbf{p}, \mathbf{g}, \mathbf{v}, \mathbf{e}\}$
- 4: **for** $m = 1, \dots, M$ **do**
- 5: Compute ρ_r using (3.49)
- 6: $\mathbf{s}^{(k,m+1)} = \rho_r \mathbf{p}^{(k,m)} + (1 - \rho_r) \mathbf{g}^{(k,m)}$
- 7: $\boldsymbol{\sigma}^{(k,m+1)} = \eta \Theta' \mathbf{D}_{\text{reg}} \mathbf{D}_{\phi} \{\Theta \mathbf{x}^{(k,m)}\} (\Theta \mathbf{x}^{(k,m)} - \mathbf{v}^{(k,m)} - \mathbf{e}^{(k,m)})$
- 8: $\mathbf{x}^{(k,m+1)} = \left[\mathbf{x}^{(k,m)} - (\rho_r \mathbf{D}_L + \eta \mathbf{D}_R^{(k,m)})^{-1} (\mathbf{s}^{(k,m+1)} + \boldsymbol{\sigma}^{(k,m+1)}) \right]_{\Omega}$
- 9: $\mathbf{p}^{(k,m+1)} = M \nabla \ell_m(\mathbf{x}^{(k,m+1)})$
- 10: $\mathbf{g}^{(k,m+1)} = \frac{\rho_r}{\rho_r + 1} \mathbf{p}^{(k,m+1)} + \frac{1}{\rho_r + 1} \mathbf{g}^{(k,m)}$
- 11: Compute $\mathbf{v}^{(k,m+1)}$ using the proximal mapping of Φ
- 12: $\mathbf{e}^{(k,m+1)} = \mathbf{e}^{(k,m)} - \Theta \mathbf{x}^{(k,m+1)} + \mathbf{v}^{(k,m+1)}$
- 13: $r = r + 1$
- 14: **end for**
- 15: $\mathbf{c}^{(k+1)} = \mathbf{c}^{(k, M+1)}$ for $\mathbf{c} \in \{\mathbf{x}, \mathbf{p}, \mathbf{g}, \mathbf{v}, \mathbf{e}\}$
- 16: **end for**
- 17: **return** $\mathbf{x}^{(n_{\text{iter}})}$

using Huber's curvature. Since $\check{\Phi}$ is quadratic, its proximal mapping is linear. This makes all \mathbf{v} - and \mathbf{e} -related steps linear. Then, let

$$\begin{cases} \bar{\mathbf{v}}^{(k,m+1)} \triangleq \Theta' \mathbf{D}_{\text{reg}} \mathbf{D}_{\phi} \{\Theta \mathbf{x}^{(k,m+1)}\} (\Theta \mathbf{x}^{(k,m+1)} - \mathbf{v}^{(k,m+1)}) \\ \tilde{\mathbf{e}}^{(k,m+1)} \triangleq \Theta' \mathbf{D}_{\text{reg}} \mathbf{D}_{\phi} \{\Theta \mathbf{x}^{(k,m+1)}\} \mathbf{e}^{(k,m+1)} \end{cases} \quad (3.65)$$

denote the ‘‘compressed’’ auxiliary variables. After simple calculations, the iterates of the compressed auxiliary variables are as follows:

$$\begin{cases} \bar{\mathbf{v}}^{(k,m+1)} = \frac{\eta}{\eta+1} \tilde{\mathbf{e}}^{(k,m)} + \frac{1}{\eta+1} \nabla \mathbf{R}(\mathbf{x}^{(k,m+1)}) \\ \tilde{\mathbf{e}}^{(k,m+1)} = \tilde{\mathbf{e}}^{(k,m)} - \bar{\mathbf{v}}^{(k,m+1)}. \end{cases} \quad (3.66)$$

Note that no explicit shrinkage has to be done now. The search direction attributed to the regularization term can be computed simply as:

$$\boldsymbol{\sigma}^{(k,m+1)} = \eta (\bar{\mathbf{v}}^{(k,m)} - \tilde{\mathbf{e}}^{(k,m)}) . \quad (3.67)$$

In other words, unlike the original (high-memory) split OS-LALM in which we have to store (one or) two extra image volumes of difference images for each direction, we only have to store (one or) two extra image volumes of difference images for all direction in the proposed approximate algorithm (inexact split OS-LALM)! The complete algorithm of inexact split OS-LALM is described in Algorithm III.6. So far, we do not have any convergence proof of the inexact split OS-LALM, even for $M = 1$, unless we think of the resulting update with majorization as an inexact update whose errors satisfy the absolutely summable condition. However, we believe algorithms that majorize functions in those updates of auxiliary variables are still convergent, just like the inexact OS-LALM (Algorithm III.3) that majorizes the regularizer in the image update. In fact, the majorization (3.64) equivalently majorizes the regularizer using a tighter non-separable quadratic surrogate function with Huber’s curvature! Proving the convergence of the inexact split OS-LALM (for $M = 1$) is a very interesting future work because it will be very useful for problems with lots of auxiliary variables.

For example, in the prior image constrained compressed sensing (PICCS) CT reconstruction [60], images are reconstructed from very limited number of view angles (i.e., highly downsampled projection views) with the help of a “prior image” that is, for example, reconstructed from the union of interleaved dynamic dataset (4D cone-beam CT). Mathematically, the PICCS CT reconstruction can be formulated as:

$$\hat{\mathbf{x}}_{\text{PICCS}} \in \arg \min_{\mathbf{x} \in \Omega} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \beta_1 \|\mathbf{x}\|_{\text{TV}} + \beta_2 \|\mathbf{x} - \mathbf{x}_{\text{prior}}\|_{\text{TV}} \right\}, \quad (3.68)$$

where $\mathbf{x}_{\text{prior}}$ denotes the prior image, $\|\cdot\|_{\text{TV}}$ denotes the TV norm of an image, and $\beta_1, \beta_2 \geq 0$ are the regularization forces of the two TV norms. In this case, suppose we solve this PICCS reconstruction problem (3.68) using splitting-based methods, say, the SB method, and consider all 13 neighbors on one side of a $3 \times 3 \times 3$ cube when computing the finite difference images. We have to store $2 \times 2 \times 13 = 52$ extra image volumes in the memory in order to solve this non-smooth minimization problem! That is definitely intractable even for a small dataset. Gradient-based methods can solve the smoothed PICCS reconstruction problem without additional auxiliary variables but are usually very slow due to the high curvature nature of the smoothed TV norm. In comparison, with our proposed low-memory OS-LALM with compressed variable splitting, we can have both faster convergence (via the η -scalable curvature) rate and efficient memory usage (by compressed variable splitting) at the same time!

Algorithm III.6 Proposed algorithm (inexact split OS-LALM-c) solving (2.10)

Require: $\mathbf{x}^{(0)} \in \mathbb{R}^{n_p}$, $n_{\text{iter}} \geq 1$, $M \geq 1$

- 1: Initialize $\mathbf{p}^{(0)} = \mathbf{g}^{(0)} = M \nabla \ell_M(\mathbf{x}^{(0)})$, $r = 1$,
 $\bar{\mathbf{v}}^{(0)} = \nabla R(\mathbf{x}^{(0)}) / (\eta + 1)$, $\tilde{\mathbf{e}}^{(0)} = -\bar{\mathbf{v}}^{(0)}$
- 2: **for** $k = 0, 1, \dots, n_{\text{iter}} - 1$ **do**
- 3: $\mathbf{c}^{(k,1)} = \mathbf{c}^{(k)}$ for $\mathbf{c} \in \{\mathbf{x}, \mathbf{p}, \mathbf{g}, \bar{\mathbf{v}}, \tilde{\mathbf{e}}\}$
- 4: **for** $m = 1, \dots, M$ **do**
- 5: Compute ρ_r using (3.49)
- 6: $\mathbf{s}^{(k,m+1)} = \rho_r \mathbf{p}^{(k,m)} + (1 - \rho_r) \mathbf{g}^{(k,m)}$
- 7: $\boldsymbol{\sigma}^{(k,m+1)} = \eta (\bar{\mathbf{v}}^{(k,m)} - \tilde{\mathbf{e}}^{(k,m)})$
- 8: $\mathbf{x}^{(k,m+1)} = \left[\mathbf{x}^{(k,m)} - (\rho_r \mathbf{D}_L + \eta \mathbf{D}_R^{(k,m)})^{-1} (\mathbf{s}^{(k,m+1)} + \boldsymbol{\sigma}^{(k,m+1)}) \right]_{\Omega}$
- 9: $\mathbf{p}^{(k,m+1)} = M \nabla \ell_m(\mathbf{x}^{(k,m+1)})$
- 10: $\mathbf{g}^{(k,m+1)} = \frac{\rho_r}{\rho_r + 1} \mathbf{p}^{(k,m+1)} + \frac{1}{\rho_r + 1} \mathbf{g}^{(k,m)}$
- 11: $\bar{\mathbf{v}}^{(k,m+1)} = \frac{\eta}{\eta + 1} \tilde{\mathbf{e}}^{(k,m)} + \frac{1}{\eta + 1} \nabla R(\mathbf{x}^{(k,m+1)})$
- 12: $\tilde{\mathbf{e}}^{(k,m+1)} = \tilde{\mathbf{e}}^{(k,m)} - \bar{\mathbf{v}}^{(k,m+1)}$
- 13: $r = r + 1$
- 14: **end for**
- 15: $\mathbf{c}^{(k+1)} = \mathbf{c}^{(k,M+1)}$ for $\mathbf{c} \in \{\mathbf{x}, \mathbf{p}, \mathbf{g}, \bar{\mathbf{v}}, \tilde{\mathbf{e}}\}$
- 16: **end for**
- 17: **return** $\mathbf{x}^{(n_{\text{iter}})}$

3.3.3 Experimental results: sparse-view CT

This section reports numerical results for simulated 3D X-ray sparse-view CT image reconstruction from a real low-dose CT scan using two different OS-based methods, including

- **OS-Nes05-M**: the OS+momentum algorithm [27] based on Nesterov’s fast gradient method [24] with M subsets (Algorithm II.4), and
- **OS-LALM-M-c- η (low-mem)**: the low-memory inexact split OS-LALM-c algorithm with M subsets and AL penalty parameter η using the downward continuation approach (Algorithm III.6).

We reconstructed a $718 \times 718 \times 122$ image from an undersampled chest axial CT scan. The size of the original sinogram is $888 \times 64 \times 642$ (half scan), and we uniformly undersampled the number of projection views from 642 to 81 (about 12.6% of projection views are used for reconstruction). Consider a smooth approximation of the anisotropic TV regularizer using the (scaled) Fair potential [61]:

$$\phi_{\text{Fair}}(x) \triangleq \delta^2 (|x/\delta| - \log(1 + |x/\delta|)) \quad (3.69)$$

with $\delta = 0.1$ HU. Due to the small δ , even though the regularizer is smooth, gradient-based methods (including the fast variants) converge very slow.

Figure 3.12 shows the cropped images from the central transaxial plane of the initial FBP image, the reference reconstruction (with $\delta = 0.1$), and the converged reconstruction with $\delta = 1$. As can be seen in Figure 3.12, due to the strong under-sampling, there are severe streak artifacts in the FBP image; while by using proper regularization, most streak artifacts can be removed. Furthermore, the value of δ also affects the image quality of the reconstructed image. When δ is larger, the regularizer has smaller curvature and is easier to minimize; however, the converged image looks smoother, and one can still see some mild streak artifacts in it.

To evaluate the effect of η and M , we run each algorithms for 100 iterations (in this case, 100 undersampled forward/back-projection pairs, about 13 full forward/back-projection pairs, are used for reconstruction) and plot the convergence rate curves as a function of both iteration and time. Figure 3.13 demonstrates how η affects the convergence rate of the proposed algorithm with fixed $M = 5$. As can be seen in Figure 3.13, algorithm converges faster with smaller η due to the lower curvature weighted quadratic AL penalty term in image updates. However, η cannot be arbitrarily small; otherwise, algorithm might be unstable because some voxels might have too large step sizes. To ensure empirical fast convergence rate, we tuned the value of η so that the median of $\eta[\mathbf{D}_R^{(0)}]_{ii}/[\mathbf{D}_L]_{ii}$ is about 2% to 10%. The intuition is that as the algorithm proceeds, $\rho_r \mathbf{D}_L$ becomes smaller and smaller, and we would like to have small enough step sizes for stable image updates with a large enough $\eta \mathbf{D}_R^{(k)}$. Note that since the number of subsets M is fixed, algorithms with different configurations have the same computational complexity. Therefore, RMS differences as a function of iteration and time do not look very different.

In contrast, Figure 3.14 shows how M affects the convergence rate of the proposed algorithm with fixed $\eta = 0.05$. By using ordered subsets, we can perform image updates more frequently, so the convergence rate curves as a function of iteration scale accordingly. However, since the forward/back-projection in sparse-view CT is not as time-consuming as in clinical CT, the overhead of regularizer gradient evaluations is not negligible. Hence, we do not see such significant acceleration of convergence when plotting convergence rate curves as a function of time. Finally, we compare our proposed algorithm with the OS+momentum algorithm. Figure 3.15 and Figure 3.16 show the convergence rate curves (as a function of both iteration and time) and the reconstructed images after 100 iterations, respectively. As can be seen in Figure 3.15, due to the very high curvature of the regularizer, the OS+momentum

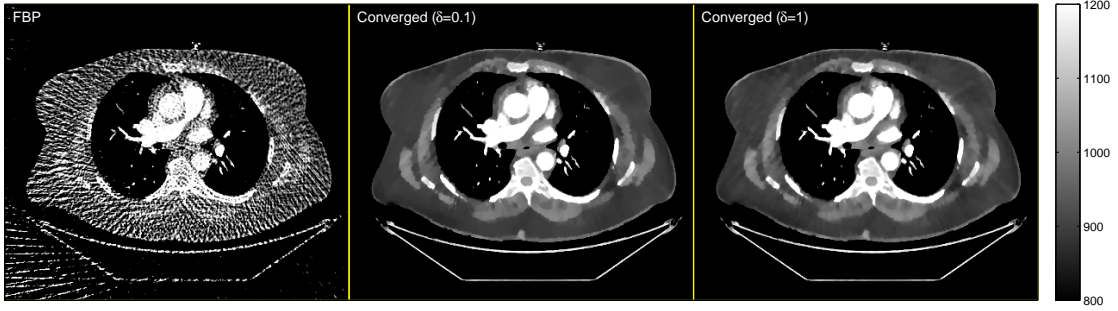


Figure 3.12: Chest scan: cropped images (displayed from 800 to 1200 HU) from the central transaxial plane of the initial FBP image $\mathbf{x}^{(0)}$ (left), the reference reconstruction \mathbf{x}^* with $\delta = 0.1$ (center), and the converged reconstruction with $\delta = 1$ (right).

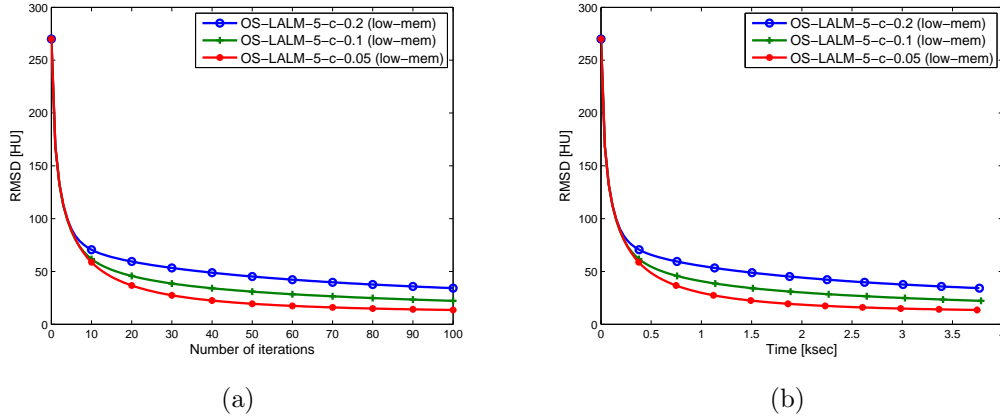


Figure 3.13: Chest scan: RMS differences between the reconstructed image $\mathbf{x}^{(k)}$ and the reference reconstruction \mathbf{x}^* as a function of (a) iteration and (b) time, respectively, using the proposed algorithm with $M = 5$ and different values of η .

algorithm becomes very slow after the 40th iteration, while the image reconstructed by the proposed algorithm keep approaching to the solution. Furthermore, the convergence rate of low frequency components depends highly on the high curvature region (around zero) of the regularizer. In Figure 3.15, we can easily see that the low frequencies of the image reconstructed by the OS+momentum algorithm converges much slower than the low frequencies of the image reconstructed by the proposed algorithm.

3.4 Conclusion

The AL method and OS are two powerful techniques for accelerating optimization algorithms using decomposition and approximation, respectively. In this chapter, we

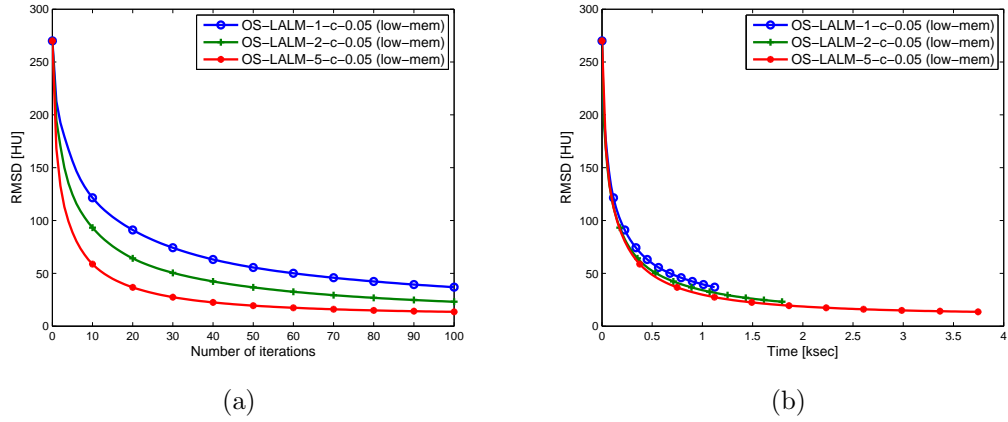


Figure 3.14: Chest scan: RMS differences between the reconstructed image $\mathbf{x}^{(k)}$ and the reference reconstruction \mathbf{x}^* as a function of (a) iteration and (b) time, respectively, using the proposed algorithm with $\eta = 0.05$ and different values of M .

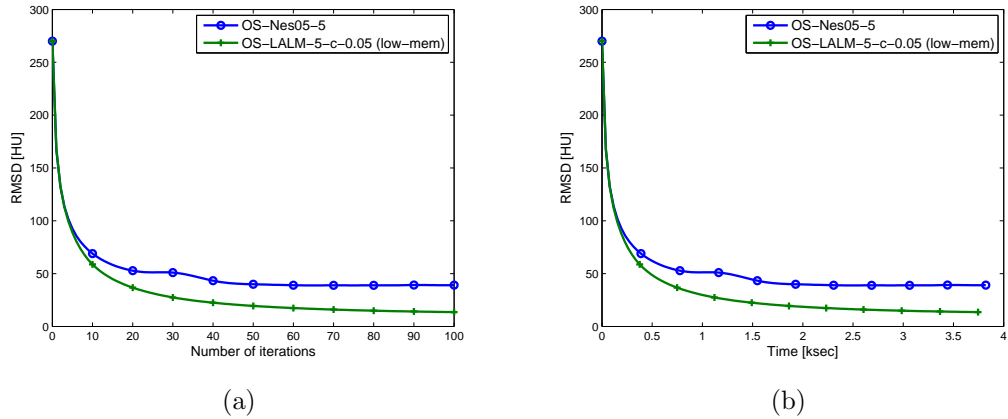


Figure 3.15: Chest scan: RMS differences between the reconstructed image $\mathbf{x}^{(k)}$ and the reference reconstruction \mathbf{x}^* as a function of (a) iteration and (b) time, respectively, using the OS+momentum algorithm and the proposed algorithm with $M = 5$.

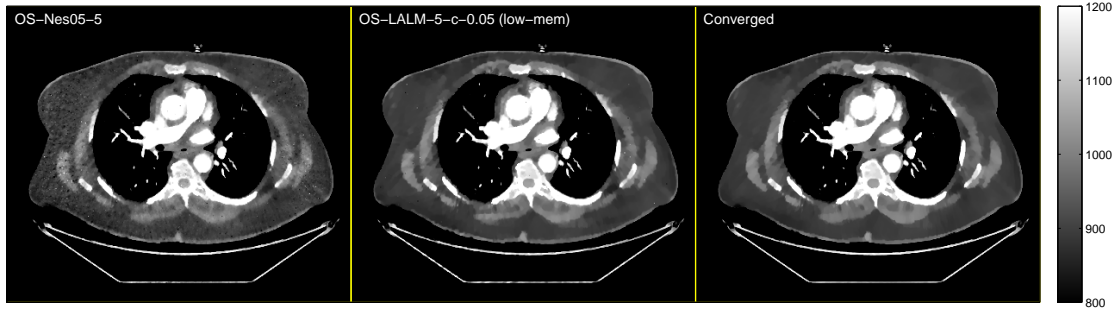


Figure 3.16: Chest scan: cropped images (displayed from 800 to 1200 HU) from the central transaxial plane of the reconstructed images $\mathbf{x}^{(100)}$ using the OS+momentum algorithm (left) and the proposed algorithm (center) after 100 iterations and the reference reconstruction \mathbf{x}^* (right).

combined these two techniques by considering a linearized variant of the AL method and proposed a fast splitting-based ordered-subset algorithm, OS-LALM, for solving X-ray CT image reconstruction problem with PWLS criterion. Several variants of the proposed algorithm were investigated for solving three practical issues of conventional splitting-based method: iterative image updates, non-trivial parameter selection, and enormous memory requirement. Experimental results showed that the proposed algorithm exhibits fast convergence rate and excellent gradient error tolerance when OS is used. Furthermore, the proposed algorithm also demonstrated more rapid low frequency convergence than conventional gradient-based methods when regularizers with very high curvature are used.

CHAPTER IV

Blind gain correction for X-ray CT image reconstruction

4.1 Introduction

The effective X-ray source intensity in CT scan can fluctuate from view to view due to the attenuation of thin items like sheets that partially block reference channels. Thibault *et al.* proposed to modify the cost function so that it depends on both the unknown image \mathbf{x} and an unknown gain parameter vector \mathbf{g} , where $[\mathbf{g}]_j$ denotes the gain fluctuation of the j th view, and to minimize jointly over both \mathbf{x} and \mathbf{g} by solving the following convex optimization problem [63]:

$$(\hat{\mathbf{x}}, \hat{\mathbf{g}}) \in \arg \min_{\mathbf{x}, \mathbf{g}} \left\{ \Psi(\mathbf{x}, \mathbf{g}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x} - \mathbf{g} \otimes \mathbf{1}\|_{\mathbf{W}}^2 + \mathbf{R}(\mathbf{x}) \right\}, \quad (4.1)$$

where \mathbf{y} denotes the noisy post-logarithm sinogram that may suffer from gain fluctuations, \mathbf{A} denotes the system matrix, \mathbf{W} denotes the diagonal weighting matrix that accounts for measurement variance, \otimes denotes the Kronecker product operator, $\mathbf{1}$ denotes the vector with all entries equal to unity and of length equal to the number of beams, and \mathbf{R} is an edge-preserving regularizer. Compared with (4.1), existing splitting-based iterative algorithms reconstruct image without considering the effect of \mathbf{g} , or equivalently, setting \mathbf{g} to be $\mathbf{0}$. This introduces visible shading artifacts as shown in Figure 4.1(c). We propose splitting-based iterative algorithms based on a simplification of the joint cost function in (4.1) that improves image quality compared to conventional splitting-based iterative algorithms [8, 32] that assume $\mathbf{g} = \mathbf{0}$.

This chapter is based on [62].

4.2 Method

4.2.1 Joint gain-image estimation for X-ray CT image reconstruction

Let \mathbf{y}_j , \mathbf{A}_j , and \mathbf{W}_j for $j = 1, \dots, J$ denote data, system matrix, and diagonal weighting matrix associated with the j th view in a CT scan, respectively. The optimization problem in (4.1) is equivalent to

$$\begin{aligned} (\hat{\mathbf{x}}, \hat{\mathbf{g}}) &\in \arg \min_{\mathbf{x}, \mathbf{g}} \left\{ \sum_{j=1}^J \frac{1}{2} \|\mathbf{y}_j - \mathbf{A}_j \mathbf{x} - g_j \mathbf{1}\|_{\mathbf{W}_j}^2 + R(\mathbf{x}) \right\} \\ &= \arg \min_{\mathbf{x}} \left\{ \sum_{j=1}^J \left(\min_{g_j} \frac{1}{2} \|\mathbf{y}_j - \mathbf{A}_j \mathbf{x} - g_j \mathbf{1}\|_{\mathbf{W}_j}^2 \right) + R(\mathbf{x}) \right\}. \end{aligned} \quad (4.2)$$

The inner minimization problem in (4.2) has a minimizer

$$\hat{g}_j(\mathbf{x}) = \frac{\mathbf{1}' \mathbf{W}_j (\mathbf{y}_j - \mathbf{A}_j \mathbf{x})}{\mathbf{1}' \mathbf{W}_j \mathbf{1}} \quad (4.3)$$

with minimum

$$\|\mathbf{y}_j - \mathbf{A}_j \mathbf{x} - \hat{g}_j(\mathbf{x}) \mathbf{1}\|_{\mathbf{W}_j}^2 = \left\| \left(\mathbf{I} - \frac{\mathbf{1} \mathbf{1}' \mathbf{W}_j}{\mathbf{1}' \mathbf{W}_j \mathbf{1}} \right) (\mathbf{y}_j - \mathbf{A}_j \mathbf{x}) \right\|_{\mathbf{W}_j}^2 = \|\mathbf{y}_j - \mathbf{A}_j \mathbf{x}\|_{\widetilde{\mathbf{W}}_j}^2, \quad (4.4)$$

where we define the following positive semi-definite symmetric “diagonal + rank-1” weighting matrix:

$$\widetilde{\mathbf{W}}_j = \mathbf{W}_j - \frac{\mathbf{W}_j \mathbf{1} \mathbf{1}' \mathbf{W}_j}{\mathbf{1}' \mathbf{W}_j \mathbf{1}}. \quad (4.5)$$

Plugging (4.4) into (4.2) yields the following problem formulation that is equivalent to (4.1) yet also equivalent to the kind of cost function used in “conventional” statistical X-ray CT image reconstruction except that it uses a non-diagonal weighting matrix:

$$\hat{\mathbf{x}} \in \arg \min_{\mathbf{x}} \left\{ \Psi(\mathbf{x}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{A} \mathbf{x}\|_{\widetilde{\mathbf{W}}}^2 + R(\mathbf{x}) \right\}, \quad (4.6)$$

where $\widetilde{\mathbf{W}}$ is a block diagonal matrix with block described in (4.5). Since $\widetilde{\mathbf{W}}$ is positive semi-definite, (4.6) is a convex optimization problem, and any existing optimization methods such as nonlinear conjugate gradient (NCG) [64], OS-SQS [10], and OS-momentum [26, 27] are still applicable. Here, we first propose to solve (4.6) by using splitting-based iterative algorithms [8, 9] with the “ $\mathbf{u} = \mathbf{A} \mathbf{x}$ ” split that involves the sinogram. The subproblem of \mathbf{u} has a closed form solution:

$$\mathbf{u}^{(k+1)} = \mathbf{D}_\rho^{-1} \left(\widetilde{\mathbf{W}} \mathbf{y} + \rho (\mathbf{A} \mathbf{x}^{(k+1)} + \mathbf{d}^{(k)}) \right), \quad (4.7)$$

where $\mathbf{D}_\rho \triangleq \widetilde{\mathbf{W}} + \rho\mathbf{I}$, and \mathbf{d} is the scaled dual variable of \mathbf{u} in ADMM [45]. When there is no gain correction, i.e., $\mathbf{g} = \mathbf{0}$, \mathbf{D}_ρ is a diagonal matrix, and (4.7) can be computed efficiently in $\mathcal{O}(S)$, where S is the size of the sinogram. When there is gain correction, i.e., $\mathbf{g} \neq \mathbf{0}$, \mathbf{D}_ρ is block diagonal matrix with block:

$$\mathbf{D}_{\rho,j} = (\mathbf{W}_j + \rho\mathbf{I}) + \begin{pmatrix} -\mathbf{w}_j \\ \mathbf{1}'\mathbf{w}_j \end{pmatrix} \mathbf{w}_j', \quad (4.8)$$

where $\mathbf{w}_j \triangleq \mathbf{W}_j\mathbf{1}$ is the diagonal entries of \mathbf{W}_j . The inverse of \mathbf{D}_ρ will also be a block diagonal matrix with block:

$$(\mathbf{D}_{\rho,j})^{-1} = (\mathbf{W}_j + \rho\mathbf{I})^{-1} + \frac{(\mathbf{W}_j + \rho\mathbf{I})^{-1}\mathbf{w}_j\mathbf{w}_j'(\mathbf{W}_j + \rho\mathbf{I})^{-1}}{\mathbf{1}'\mathbf{w}_j - \mathbf{w}_j'(\mathbf{W}_j + \rho\mathbf{I})^{-1}\mathbf{w}_j} \quad (4.9)$$

by the Sherman-Morrison formula. Note that the matrix-vector multiplication of $(\mathbf{D}_{\rho,j})^{-1}$ and a vector of proper size involves only componentwise division, vector inner product, and vector outer product. Therefore, the computational complexity of (4.7) is still $\mathcal{O}(S)$ in the presence of gain correction. That is, we can estimate the unknown gain parameter vector and the image jointly with almost unchanged computational complexity per iteration.

4.2.2 Applying prior knowledge of gain parameter to the joint gain-image estimation

The optimization problem described in (4.1) can be thought of as an X-ray CT image reconstruction with *blind* gain correction since we have no prior knowledge of the gain parameter vector, and we apply gain correction to every view. However, sometimes we do have some prior knowledge of the gain parameter vector. For example, when the sinogram is truncated between some view angles, we know that the object is outside the field of view, and the reference channels might be blocked by the object in these views with high probability. Hence, it is better to apply gain correction to these views. Similarly, when the object is well bounded in the field of view in some view angles, i.e., the projection is not truncated, the reference channel is less likely to be blocked, and we can turn off gain corrections in these views. To incorporate such prior knowledge about the support of the gain parameter vector, we propose a constrained optimization problem for *non-blind* gain correction:

$$(\hat{\mathbf{x}}, \hat{\mathbf{g}}) \in \arg \min_{\mathbf{x}, \mathbf{g}} \left\{ \Psi(\mathbf{x}, \mathbf{g}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x} - \mathbf{g} \otimes \mathbf{1}\|_{\mathbf{W}}^2 + \mathbf{R}(\mathbf{x}) \right\} \text{ s.t. } [\mathbf{g}]_{j \notin \mathcal{J}=0}, \quad (4.10)$$

where $\mathcal{J} \subseteq \{1, \dots, J\}$ is the index set of the *candidate* views which may suffer from gain fluctuations. Note that since the gain fluctuations are assumed to be only view-dependent, following the same procedure in the blind case, we can get exactly the same equivalent problem formulation as in (4.6), where $\widetilde{\mathbf{W}}$ is a block diagonal matrix with block:

$$\widetilde{\mathbf{W}}_j = \begin{cases} \mathbf{W}_j & , \text{ if } j \notin \mathcal{J} \\ \mathbf{W}_j - \frac{\mathbf{w}_j \mathbf{1} \mathbf{1}' \mathbf{W}_j}{\mathbf{1}' \mathbf{W}_j \mathbf{1}} & , \text{ otherwise.} \end{cases} \quad (4.11)$$

When we solve the X-ray CT image reconstruction with non-blind gain correction using splitting-based methods with the “ $\mathbf{u} = \mathbf{Ax}$ ” split that involves the sinogram, we will solve the subproblem of \mathbf{u} using (4.7), where $(\mathbf{D}_\rho)^{-1}$ is a block diagonal matrix with block:

$$(\mathbf{D}_{\rho,j})^{-1} = (\mathbf{W}_j + \rho \mathbf{I})^{-1} + \frac{(\mathbf{W}_j + \rho \mathbf{I})^{-1} \mathbf{w}_j \mathbf{w}_j' (\mathbf{W}_j + \rho \mathbf{I})^{-1}}{\mathbf{1}' \mathbf{w}_j - \mathbf{w}_j' (\mathbf{W}_j + \rho \mathbf{I})^{-1} \mathbf{w}_j} \mathbb{1}_{j \in \mathcal{J}}, \quad (4.12)$$

where $\mathbb{1}_{j \in \mathcal{J}}$ is the indicator function of \mathcal{J} . Clearly, when $\mathcal{J} = \emptyset$, it reduces to the X-ray CT image reconstruction without gain correction; when $\mathcal{J} = \{1, \dots, J\}$, it is the X-ray CT image reconstruction with blind gain correction. Furthermore, if desired, we can shrink the set \mathcal{J} as the iterative algorithm proceeds. For example, we can reset the very small estimated gain fluctuations to be zero after several iterations.

4.2.3 Joint gain-image estimation using other optimization methods

We can also solve (4.6) using any other convex optimization method. For example, the cost function in (4.6) has gradient

$$\nabla \Psi(\mathbf{x}) = \mathbf{A}' \widetilde{\mathbf{W}} (\mathbf{Ax} - \mathbf{y}) + \nabla R(\mathbf{x}) \quad (4.13)$$

assuming that the regularization term $R(\mathbf{x})$ is differentiable. This gradient can be used for any first-order method such as NCG [64]. In OS-based algorithms [10, 26, 27], we have to find a (separable) quadratic surrogate function that majorizes the original cost function and to minimize it. Since the only difference between the conventional variational formulation and our proposed formulation is the weighted least-squares (WLS) term, we just focus on the majorizer of that part. A quadratic majorizer of a function f has the general form

$$f(\mathbf{x}^{(k)}) + (\mathbf{x} - \mathbf{x}^{(k)})' \nabla f(\mathbf{x}^{(k)}) + \frac{1}{2} \|\mathbf{x} - \mathbf{x}^{(k)}\|_{\mathcal{G}}^2 \quad (4.14)$$

with $\mathcal{G} \succeq \nabla^2 f$. When f is the conventional WLS cost function, one particular choice of \mathcal{G} is $\mathcal{G}_{\text{diag}} \triangleq \text{diag}\{|\mathbf{A}'\mathbf{W}\mathbf{A}|\mathbf{1}\} \succeq \mathbf{A}'\mathbf{W}\mathbf{A}$ [10]. Since $\mathbf{W} \succeq \widetilde{\mathbf{W}}$, we have $\mathbf{A}'\mathbf{W}\mathbf{A} \succeq \mathbf{A}'\widetilde{\mathbf{W}}\mathbf{A}$. Therefore, $\mathcal{G}_{\text{diag}}$ is a valid diagonal matrix that majorizes the proposed WLS cost function, and it is very easy to modify the existing OS-based algorithms to enable gain correction.

4.3 Result

To evaluate our proposed method, we consider both 2D and 3D X-ray CT image reconstruction problems. In each case, the statistical weight w_i is set to be $\exp(-y_i)$, where y_i denotes the line-integral projection *with* gain fluctuations. We are interested in edge-preserving regularizer \mathbf{R} in the form:

$$\mathbf{R}(\mathbf{x}) \triangleq \beta \sum_{n=1}^N \sum_{m=1}^M \kappa_n \kappa_{n+s(m)} \Phi([\mathbf{C}_m \mathbf{x}]_n), \quad (4.15)$$

where β is the regularization parameter, N is the number of voxels, M is the number of offsets, κ_n is the voxel-dependent weight for $n = 1, \dots, N$, \mathbf{C}_m is the first-order finite-difference matrix in the m th direction with offset $s(m)$ for $m = 1, \dots, M$, and Φ is an edge-preserving potential function. We choose Φ to be the Fair potential function $\Phi_{\text{FP}}(x) \triangleq |x|/\delta - \ln(1 + |x|/\delta)$ with parameter δ . Following the voxel-dependent weight proposed by Fessler *et al.* [15], κ_n is set to be $\sqrt{[\mathbf{A}'\mathbf{W}\mathbf{1}]_n / [\mathbf{A}'\mathbf{1}]_n}$. For 2D case, $M = 2$ for the horizontal and vertical neighbors; for 3D case, $M = 13$ for the thirteen nearest neighbors. The minimization problem in (4.6) is solved by using ADMM for 500 iterations [8]. The FBP reconstruction from the gain-fluctuated noisy sinogram is used as the initial guess $\mathbf{x}^{(0)}$ for the iterative algorithm.

4.3.1 2D fan beam X-ray CT image reconstruction

We first consider a 2D X-ray CT image reconstruction from simulated NCAT phantom data with gain fluctuations. We use a 256×256 2D slice of NCAT phantom to numerically generate a 444×492 gain-fluctuated noisy sinogram with GE LightSpeed fan-beam geometry downsampled by two corresponding to a monoenergetic source with 10^5 incident photons per ray without background events. Two sections of angular samples suffer from gain fluctuations due to partially blocked reference channels with 20% and 18% attenuation, respectively, as shown in Figure 4.1. We set $\delta = 10^{-5}$ and $\beta = 3 \times 10^{-6}$ for edge-preserving regularization in this case. Figure 4.2 shows the true

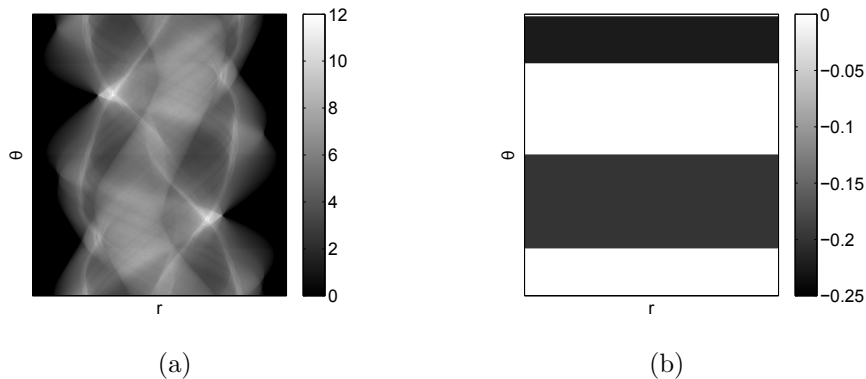


Figure 4.1: 2D simulation: (a) The noisy sinogram with gain fluctuations, and (b) the corresponding gain fluctuations, where the horizontal and vertical axes are the radial axis r and the projection view angle θ , respectively.

image, the initial guess, the conventional reconstruction without gain correction, the proposed reconstruction with blind gain correction, and the reference reconstruction from a noisy sinogram without gain fluctuations as a comparison, from left to right, top to bottom. As can be seen from Figure 4.2, our proposed method greatly reduced the shading artifacts resulting from gain fluctuations. Figure 4.3 shows the estimated gain parameter vector and its RMS error. As can be seen from Figure 4.3, our proposed method estimates the gain parameter vector accurately, and therefore, we have a comparable reconstruction with the reconstruction from the gain-fluctuation-free noisy sinogram as shown in Figure 4.1(d) and Figure 4.1(e). The RMS difference between them is about $3.12 \times 10^{-5} \text{ cm}^{-1}$, which means that they are very close to each other.

4.3.2 3D axial X-ray CT image reconstruction

We now consider a 3D X-ray CT image reconstruction from simulated phantom data with gain fluctuations. Assuming that the gain fluctuations are changing linearly in the angular direction and are constant in the transaxial direction, we use a $128 \times 120 \times 100$ 3D phantom (cylinder bone-like inserts) to analytically generate a $128 \times 120 \times 144$ noisy sinogram with axial geometry corresponding to a monoenergetic source with 10^4 incident photons per ray without background events. Then, we numerically add 2% and 5% (peak) attenuation to two separate sections of views, respectively. In this case, we set $\delta = 10^3$ and $\beta = 2 \times 10^{-3}$ for edge-preserving regularization. Figure 4.4 shows the middle transaxial, coronal, and sagittal planes of the true image, the initial guess, the conventional reconstruction without gain correction,

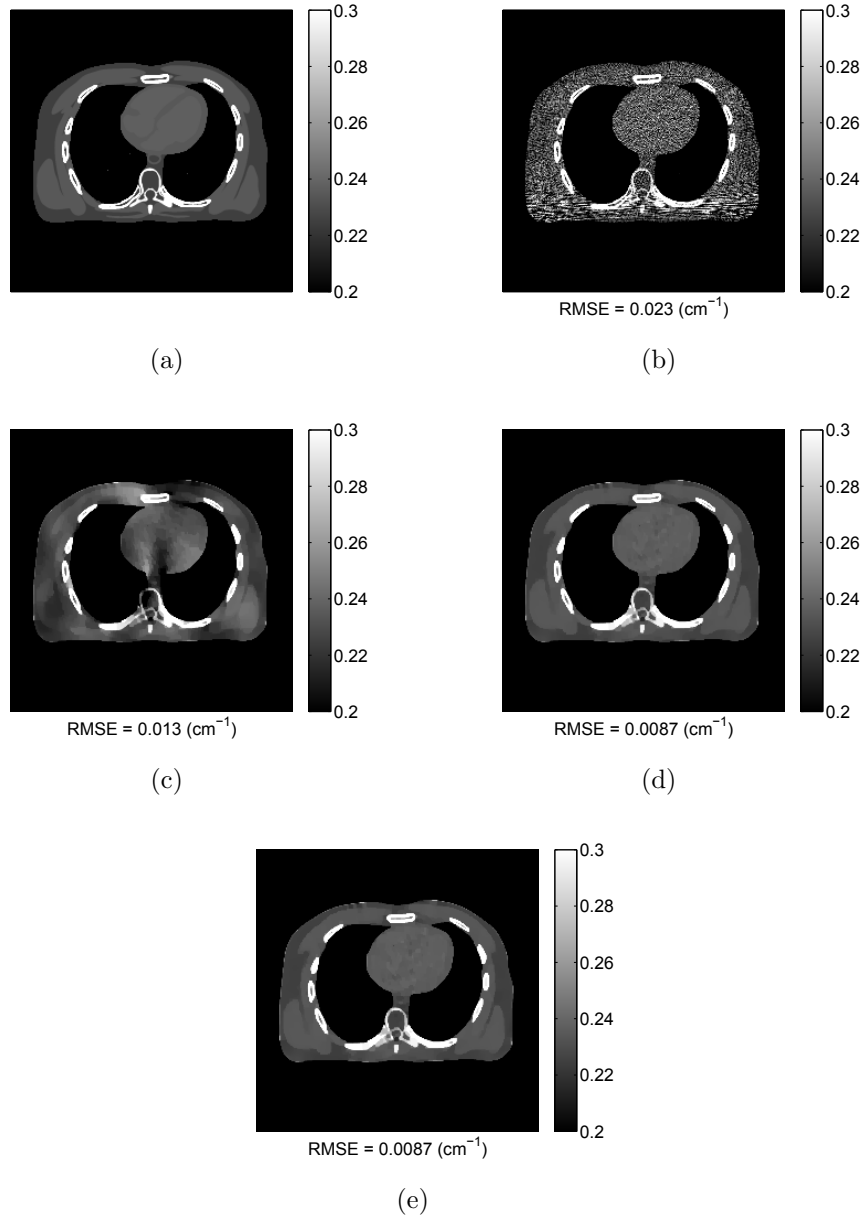


Figure 4.2: 2D simulation: (a) The true phantom (in cm^{-1}), (b) the initial guess using the FBP reconstruction, (c) the conventional reconstruction without gain correction, (d) the proposed reconstruction with blind gain correction, and (e) the reference reconstruction from a noisy sinogram without gain fluctuations.

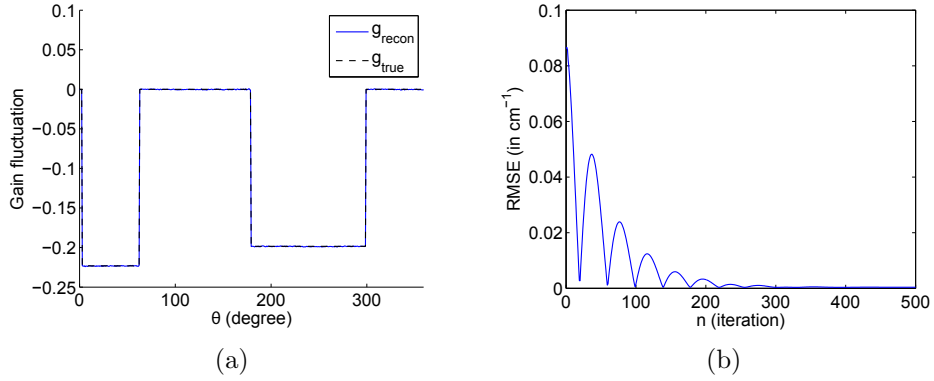


Figure 4.3: 2D simulation: (a) The estimated gain parameter vector as a function of projection view angle, and (b) the RMS error of the estimated gain parameter vector versus iteration.

the proposed reconstruction with blind gain correction, the proposed reconstruction with non-blind gain correction, and the reference reconstruction from a noisy sinogram without gain fluctuations as a comparison, from left to right, top to bottom. As can be seen from Figure 4.4, our proposed method effectively reduced the shading artifacts under such small attenuations. Figure 4.5 shows the true gain fluctuations and the estimated gain fluctuations for both blind and non-blind cases. The estimated gain fluctuations are a little bit noisier due to the small peak attenuation and show a ringing pattern no matter the reconstruction is blind or not. Figure 4.6 shows the RMS difference between the image at the k th iteration and the converged reference reconstruction for each method. Note that all methods show almost the same convergence rates in the early iterations before they start deviating from the solution. That is, gain correction does not change the convergence rate of the algorithm very much but rather improves the overall accuracy of the method.

4.4 Conclusion

In this chapter, a new variational formulation of statistical X-ray CT image reconstruction for jointly estimating the true gain parameter vector and the image was proposed. We evaluated our proposed method in both 2D and 3D cases. The shading artifacts due to gain fluctuations are greatly reduced, while the computational complexity per iteration is almost unchanged in our proposed method. Similar concepts can be applied to any convex optimization methods.

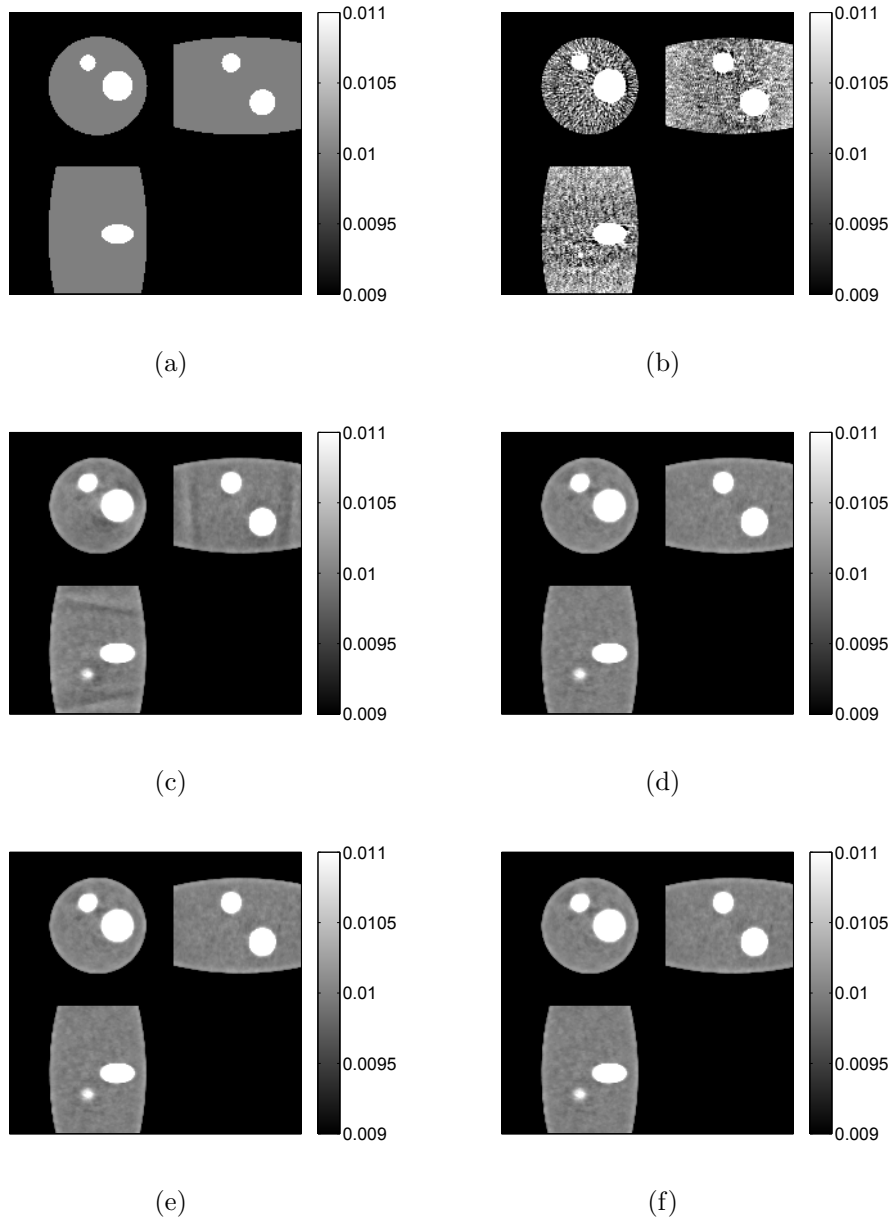


Figure 4.4: 3D simulation: (a) The true phantom (in cm^1), (b) the initial guess using the FBP reconstruction, (c) the conventional reconstruction without gain correction, (d) the proposed reconstruction with blind gain correction, (e) the proposed reconstruction with non-blind gain correction, and (f) the reference reconstruction from a noisy sinogram without gain fluctuations. Each subfigure shows the middle transaxial, coronal, and sagittal planes of the volume.

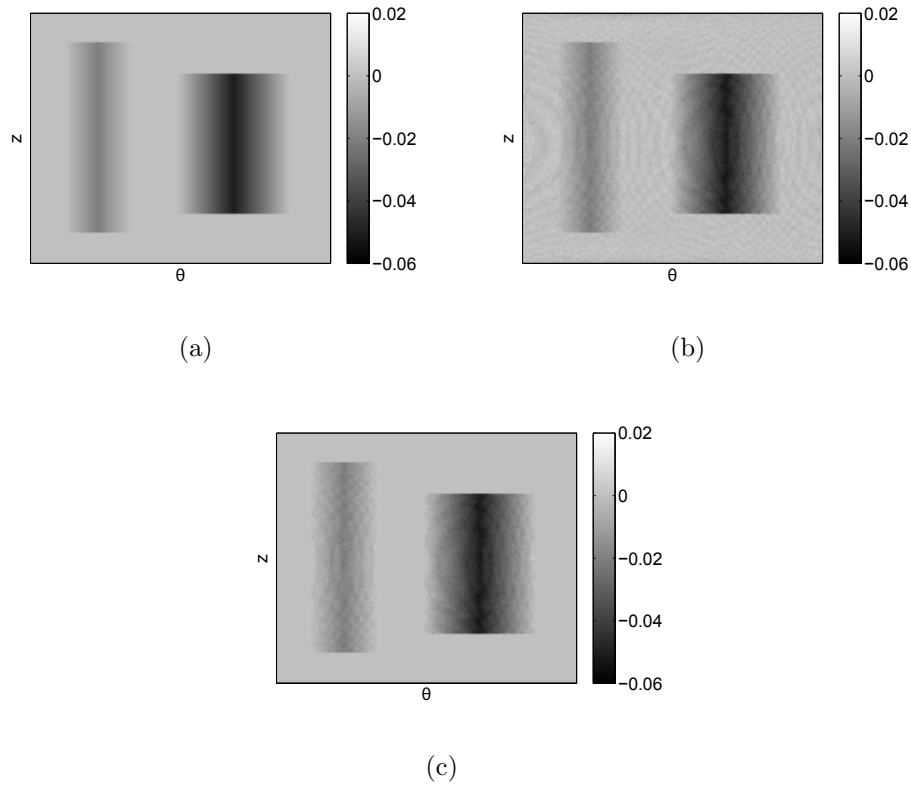


Figure 4.5: 3D simulation: (a) the true gain fluctuations, (b) the estimated gain fluctuations for the blind case, and (c) the estimated gain fluctuations for the non-blind case, where the horizontal and vertical axes are the projection view angle θ and the transaxial axis z , respectively.

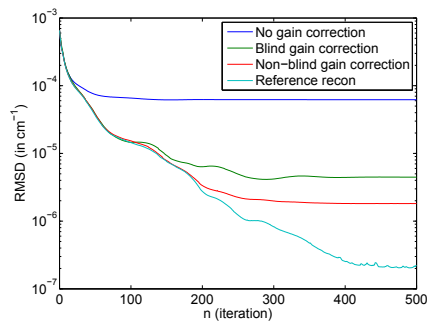


Figure 4.6: 3D simulation: for each method, RMS difference between the image at the n th iteration and the converged reference reconstruction.

CHAPTER V

Model-based light field reconstruction

5.1 Introduction

Visual information results from complex interactions between lighting, object geometry, materials, observer location and the characteristic of human visual system (or image sensor). To generate a photorealistic image, all parameters have to be precisely modeled, and numerous computations are required for simulating the interactions between parameters. This makes it difficult to have a simple systematic analysis. However, if we focus on geometric optics, the overall effects of these interactions are simply re-direction, scattering and attenuation of light rays. Therefore, once light rays in space can be described explicitly, we can re-create the visual contents without simulating the complex interactions. To describe the distribution of light rays explicitly, we should specify the radiance carried by a light ray propagating through position (x, y, z) in direction (θ, ϕ) with wavelength λ at time t . This leads to a seven-dimensional radiance function $L(x, y, z, \theta, \phi, \lambda, t)$, which is called *plenoptic function*, proposed by Edward H. Adelson and James R. Bergen for early vision [65].

However, the 7D plenoptic function is never used in practice because it is expensive and difficult to sample, store and reconstruct. Therefore, some simplifications are imposed. First, one can remove the time axis by focusing on a static scene. Second, one can use the trichromatic property of human visual system to replace the wavelength axis with, for example, RGB. This leads to three 5D plenoptic functions, one for each color [66]. Furthermore, if we concentrate on the plenoptic function in an *empty bounding box*, for example, the plenoptic function in a camera, since the bounding box is empty, there is no unexpected redirection, gain, and attenuation when light rays are propagating in the bounding box. It is sufficient to describe the

This chapter is based on my research for the qualifying exam.

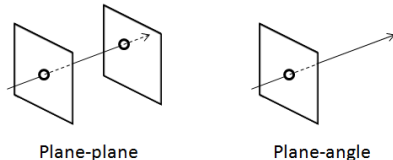


Figure 5.1: Different parameterizations of light fields.

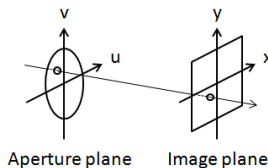


Figure 5.2: Parameterization of light fields in a camera.

5D plenoptic function by specifying its 4D slice. For example, we can describe the 5D plenoptic function $L(x, y, z, \theta, \phi)$ given a 4D slice $L(x, y, z_0, \theta, \phi)$ of the complete 5D plenoptic function since the observed radiances carried by a light ray propagating through (x, y, z_0) with angle (θ, ϕ) will be the same (or decay in a known manner) along the line. Therefore, the 5D plenoptic function is redundant, leaving us with a four-dimensional function, called *light field* [67] or *lumigraph* [68].

5.2 Discrete light fields

There are many ways to parameterize the 4D light field as shown in Figure 5.1. We can parameterize a light ray by its intersections with two parallel planes. We can also parameterize a light ray by its intersection and angle to a reference plane. If our main interest is the light field inside a camera, it can be intuitively and uniquely represented by the plane-plane parameterization which takes the aperture plane (uv -plane) and image plane (xy -plane) as the reference planes as shown in Figure 5.2. In this parameterization, each light ray in the camera passes through the aperture at some point and hits some point on the image plane. The origins of these two planes are defined by their intersections with the optical axis of the camera. We can define $L(\mathbf{x}, \mathbf{u})$ to be a 4D light field representing the radiance carried by light ray passing through $\mathbf{u} = (u, v)$ on the aperture plane and $\mathbf{x} = (x, y)$ on the sensor plane when it reaches the aperture plane.

To start our analysis of 4D light fields, let's consider the light field of a planar object with a Lambertian reflectance pattern $h(\mathbf{x}'', \mathbf{u}'') = h(\mathbf{x}'')$ in a (simplified) camera as shown in Figure 5.3. As can be seen in Figure 5.3, $(\mathbf{x}'', \mathbf{u}'')$ and (\mathbf{x}, \mathbf{u})

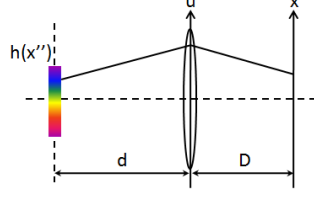


Figure 5.3: Ray diagram of a planar object with Lambertian reflectance h in a simplified camera.

can be represented as $h(\mathbf{x}'')/d^2$, where the $1/d^2$ term denotes the r^{-2} -decay when light propagating. We can find the expression of \mathbf{x}'' in terms of \mathbf{x} and \mathbf{u} by using *ray transfer matrix analysis*, and the light field of a planar object with reflectance pattern h can be represented as

$$L(\mathbf{x}, \mathbf{u}) = h\left(\left(1 - \frac{d}{f} + \frac{d}{D}\right)\mathbf{u} + \left(-\frac{d}{D}\right)\mathbf{x}\right)/d^2, \quad (5.1)$$

where d denotes the object depth, f denotes the focal length of the main lens, and D denotes the separation between the aperture plane and the sensor plane. Note that from (5.1), we can find that the 4D light field of a planar scene is nothing but an oblique backprojection of the 2D object reflectance.

The light field we derived so far is continuous, which is intractable for computation. To find the tractable discrete light field, we need more derivations. Suppose the continuous light field $L(\mathbf{x}, \mathbf{u})$ can be represented as a linear combination of a shift-invariant basis function $\beta(\mathbf{x}, \mathbf{u})$ with grid sizes $\Delta_{x,y}$ and $\Delta_{u,v}$, i.e.,

$$L(\mathbf{x}, \mathbf{u}) = \sum_{\mathbf{m}} \sum_{\mathbf{k}} L[\mathbf{m}, \mathbf{k}] \beta(\mathbf{x} - \mathbf{m}\Delta_{x,y}, \mathbf{u} - \mathbf{k}\Delta_{u,v}). \quad (5.2)$$

For simplicity, we just let the basis function to be a separable rectangular function with widths $\Delta_{x,y}$ and $\Delta_{u,v}$ in xy - and uv -dimensions, respectively. The coefficient $L[\mathbf{m}, \mathbf{k}]$ is set to be the average of $L(\mathbf{x}, \mathbf{u})$ over the rectangular region centered at $(\mathbf{m}\Delta_{x,y}, \mathbf{k}\Delta_{u,v})$, and the continuous light field $L(\mathbf{x}, \mathbf{u})$ can then be approximated by

$$L(\mathbf{x}, \mathbf{u}) \approx \sum_{\mathbf{m}} \sum_{\mathbf{k}} L[\mathbf{m}, \mathbf{k}] \text{rect}_{\Delta_{x,y}}(\mathbf{x} - \mathbf{m}\Delta_{x,y}) \text{rect}_{\Delta_{u,v}}(\mathbf{u} - \mathbf{k}\Delta_{u,v}). \quad (5.3)$$

and the specific coefficient $L[\mathbf{m}, \mathbf{k}]$ is define as the *discrete light field*. For example, the discrete light field $L[\mathbf{m}, \mathbf{k}]$ of the planar object shown in Figure 5.3 is the oblique backprojection of the 2D object reflectance onto the 4D rectangular region with grid sizes $\Delta_{x,y}$ and $\Delta_{u,v}$ centered at $(\mathbf{m}\Delta_{x,y}, \mathbf{k}\Delta_{u,v})$.

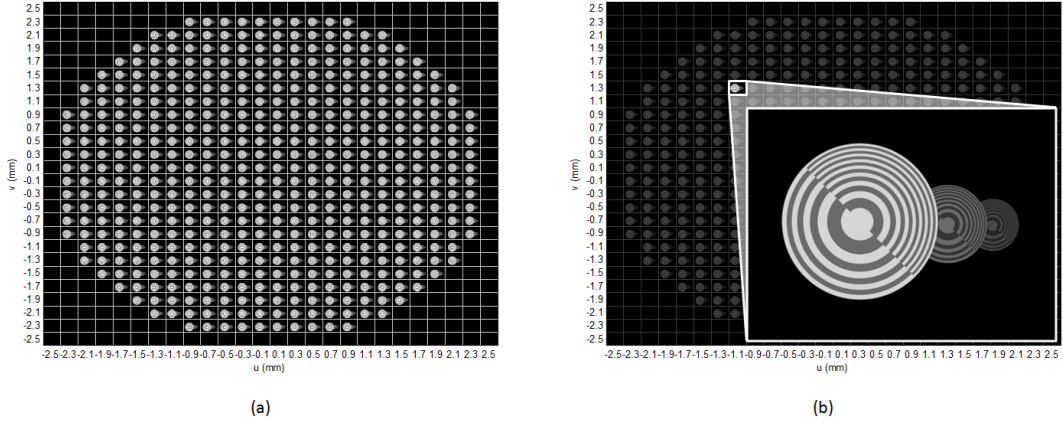


Figure 5.4: An example of a 4D light field. (a) A 2D representation of a 4D light field $L(\mathbf{x}, \mathbf{u})$. Each block denotes a 2D slice or a sub-aperture image with sub-aperture centered at \mathbf{u} . (b) An example of a 2D slice $L(\mathbf{x}, \mathbf{u}_0)$ of the 4D light field $L(\mathbf{x}, \mathbf{u})$.

Figure 5.4 shows an example discrete light field of a scene with three identical planar target objects placed at 1 m, 2 m, and 3 m, respectively, in front of a camera with aperture radius $r = 2.5$ mm, focal length $f = 50$ mm and aperture-sensor separation $D = 51$ mm. The grid sizes $\Delta_{x,y}$ and $\Delta_{u,v}$ are set to be 10^{-2} mm and 10^{-1} mm, respectively. Note that we downsample the light field by a factor of 2 in uv -dimension for alias-free display. Each block in Figure 5.4(a) shows a 2D slice of $L(\mathbf{x}, \mathbf{u})$ or a sub-aperture image with sub-aperture centered at \mathbf{u} . Figure 5.4(b) shows an example sub-aperture image with sub-aperture centered at \mathbf{u}_0 . The sub-aperture image looks sharp due to the small sub-aperture.

5.3 Light field transformations of typical camera operations

In this section, we derive the light field transformations of typical camera operations including *refocusing*, *zooming*, and *camera translation*. We consider a general light field transformation as

$$L'(\mathbf{x}', \mathbf{u}') = A \cdot L(a\mathbf{x}' + b\mathbf{u}' + \mathbf{e}, c\mathbf{x}' + d\mathbf{u}' + \mathbf{f}), \quad (5.4)$$

where $\mathbf{e} = (e_x, e_y)'$ and $\mathbf{f} = (f_u, f_v)'$. The discrete image captured by the camera with light field L' will be

$$I_{\text{general}}[\mathbf{m}] = \int \left(\int L'(\mathbf{x}, \mathbf{u}) d\mathbf{u} \right) \text{rect}_{\Delta_{x,y}}(\mathbf{x} - \mathbf{m}\Delta_{x,y}) d\mathbf{x}. \quad (5.5)$$

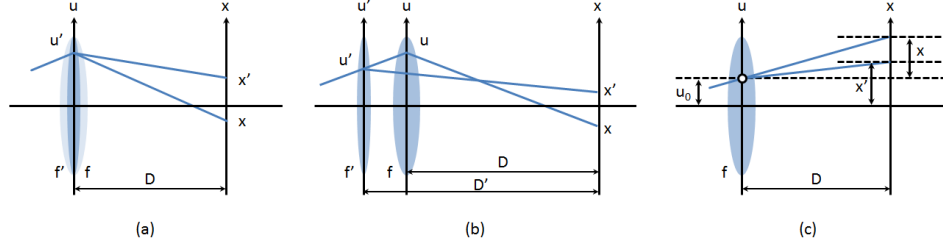


Figure 5.5: Ray diagrams for (a) refocusing, (b) zooming, and (c) camera translation.

Now, by plugging the discrete light field defined in (5.3), we can simplify (5.5) as

$$I_{\text{general}}[\mathbf{m}] = (L * g_{\text{general}})[\mathbf{m}', \mathbf{k}'] \Big|_{\mathbf{m}'=a\mathbf{m}, \mathbf{k}'=\frac{c\Delta_{x,y}}{\Delta_{u,v}}\mathbf{m}}, \quad (5.6)$$

where

$$g_{\text{general}}[\mathbf{m}, \mathbf{k}] \triangleq A \cdot (s * t)(\mathbf{m}\Delta_{x,y}, \mathbf{k}\Delta_{u,v}), \quad (5.7)$$

$$s(\mathbf{x}, \mathbf{u}) \triangleq \text{rect}_{\Delta_{x,y}}(\mathbf{x}) \text{rect}_{\Delta_{u,v}}(\mathbf{u}), \quad (5.8)$$

and

$$t(\mathbf{x}, \mathbf{u}) \triangleq \frac{1}{|ad-bc|} \text{rect}_{|ad-bc|\Delta_{x,y}}(d(\mathbf{x} + \mathbf{e}) - b(\mathbf{u} + \mathbf{f})). \quad (5.9)$$

The complete derivation is omitted here. In sum, we factor the vertical projection of a linear-transformed light field into a 4D convolution followed by a 4D-to-2D slicing. This gives us a fast algorithm to simulate typical camera operations given the light field. Note that 4D FFT has high memory allocation overhead, in real implementation, we decompose the 4D FFT and 4D-to-2D slicing into several 2D FFTs on the 2D slice. This decomposition has the same complexity but lower overhead.

5.3.1 Light field transformation: refocusing

In this subsection, we derive the light field transformation for refocusing the camera at a different depth. Consider the refocusing procedure as shown in Figure 5.5(a). We want to refocus the camera which originally focuses at some scene depth with a lens of focal length f at a different scene depth with a lens of focal length f' . Let L denote the light field captured by the camera with the original focus setting and L' denote the light field captured by the camera with the new focus setting. Note that since $(\mathbf{x}', \mathbf{u}')$ and $(\mathbf{x}, \mathbf{u}')$ represent an identical light ray, by using ray transfer matrix

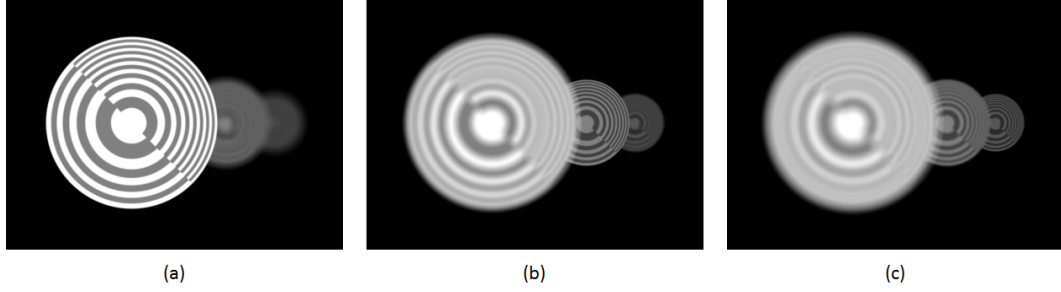


Figure 5.6: Demo of refocusing: (a) Image refocused at the first object ($z = 1$ m). (b) Image refocused at the second object ($z = 2$ m). (c) Image refocused at the third object ($z = 3$ m).

analysis, we can represent L' by L as

$$L'(\mathbf{x}', \mathbf{u}') = L\left(\mathbf{x}' + \left(\frac{D}{f'} - \frac{D}{f}\right)\mathbf{u}', \mathbf{u}'\right). \quad (5.10)$$

In other words, changing focus setting from f to f' results in shearing light field by a factor of $D/f' - D/f$. Obviously, refocusing falls into our general light field transformation. Figure 5.6 shows a demonstration of refocusing using the algorithm derived in this subsection. The discrete light field we used in this demo is the same as the discrete light field we showed in Figure 5.4. We refocus the camera at each object. As can be seen in Figure 5.6, in-focus objects look sharp while out-of-focus objects look blurry in the refocused images.

5.3.2 Light field transformation: zooming

In this subsection, we derive the light field transformation for zooming the camera to have different angle of view. Consider the zooming scheme to change angle of view as shown in Figure 5.5(b). We change the angle of view by moving the main lens and changing its focal length at the same time. Similar to the analysis in the previous subsection, let L denote the light field captured by the camera with focal length f and aperture-sensor separation D and L' denote the light field captured by the camera with focal length f' and aperture-sensor separation D' . Since (\mathbf{x}, \mathbf{u}) and $(\mathbf{x}', \mathbf{u}')$ represent the same light ray, L' can be represented by L as

$$L'(\mathbf{x}', \mathbf{u}') = L(\bar{a}\mathbf{x}' + \bar{b}\mathbf{u}', \bar{c}\mathbf{x}' + \bar{d}\mathbf{u}') , \quad (5.11)$$

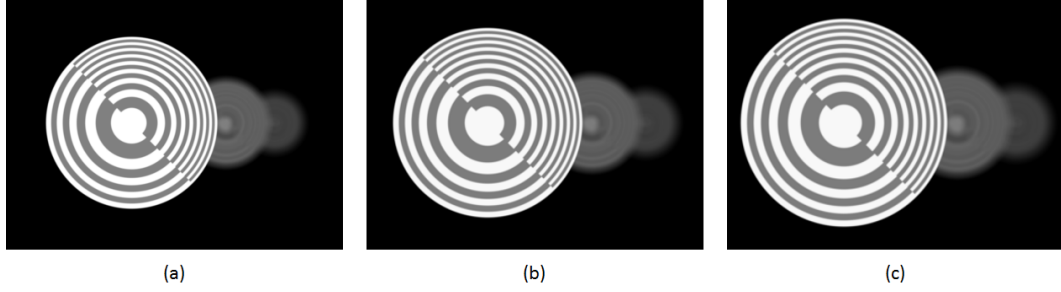


Figure 5.7: Demo of zooming: (a) Image captured with $\alpha = 1.0$. (b) Image captured with $\alpha = 1.1$. (c) Image captured with $\alpha = 1.2$.

where

$$\begin{cases} \bar{a} = \left(1 - \frac{D}{f}\right)\left(1 - \frac{D}{D'}\right) + \frac{D}{D'} \\ \bar{b} = \left(1 - \frac{D}{f}\right)\left(\frac{D'}{f'} - \frac{D}{f'} + \frac{D}{D'}\right) + \frac{D}{f'} - \frac{D}{D'} \\ \bar{c} = 1 - \frac{D}{D'} \\ \bar{d} = \frac{D'}{f'} - \frac{D}{f'} + \frac{D}{D'}. \end{cases} \quad (5.12)$$

Unlike refocusing, the light field transformation for zooming is a combination of shearing and scaling, and the optical zooming effect results from the scaling of light field. However, since the resolution of discrete light field is finite, we still have to do digital zooming. Again, zooming falls in our general light field transformation. Furthermore, in zooming, we usually have a non-zero c , and therefore, a non-horizontal slicing. Fortunately, the magnitude of $c\Delta_{x,y}/\Delta_{u,v}$ is usually fairly small, so a horizontal slicing is usually a good approximation of the non-horizontal slicing. Figure 5.7 shows a demonstration of zooming using the algorithm derived in this subsection. The discrete light field we used in this demo is the same as the discrete light field we showed in Figure 5.4. We focus at the first object and zoom the camera with $D' = \alpha D$, where $\alpha = 1.0, 1.1$, and 1.2 in our example. As can be seen in Figure 5.7, as we increase α , the angle of view decreases, and therefore the objects look larger in the captured images.

5.3.3 Light field transformation: camera translation

In this subsection, we derive the light field transformation for translating the camera on the aperture plane. Consider the setting shown in Figure 5.4(c), in which we synthesize an image of a pinhole camera centering at \mathbf{u}_0 on the aperture plane. It is trivial to extend the pinhole camera to the lens camera. According to the definition of light field, the pinhole image $I_{\mathbf{u}_0}(\mathbf{x}) = L(\mathbf{x}', \mathbf{u}_0)/D^2$. Furthermore, $\mathbf{x}' =$

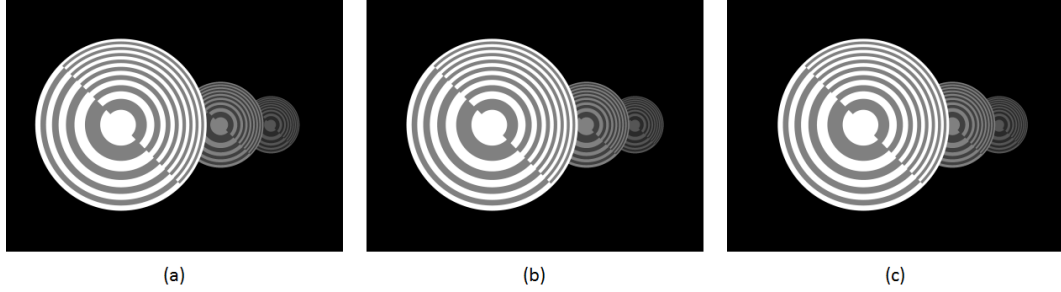


Figure 5.8: Demo of camera translation: (a) Pinhole image with pinhole centered at $\mathbf{u}_0 = (-2.5 \text{ mm}, 0.0 \text{ mm})$. (b) Pinhole image with pinhole centered at $\mathbf{u}_0 = (0.0 \text{ mm}, 0.0 \text{ mm})$. (c) Pinhole image with pinhole centered at $\mathbf{u}_0 = (2.5 \text{ mm}, 0.0 \text{ mm})$.

$\mathbf{x} + (1 - D/f)\mathbf{u}_0$ by ray transfer matrix analysis. Therefore, the pinhole image can be represented as

$$I_{\mathbf{u}_0}(\mathbf{x}) = L\left(\mathbf{x} + \left(1 - \frac{D}{f}\right)\mathbf{u}_0, \mathbf{u}_0\right) / D^2, \quad (5.13)$$

which is a shifted sub-aperture image. Figure 5.8 shows a demonstration of camera translation using the algorithm derived in this subsection. The discrete light field we used in this demo is the same as the discrete light field we showed in Figure 5.4. We translate the camera from the right to the left. As can be seen in Figure 5.8, the right viewer, i.e., Figure 5.8(a), can see larger occluded region of the second object than the left viewer, i.e., Figure 5.8(c), does. Furthermore, we can see the parallax when the camera moves from right to left. As expected, the front object shows larger displacement than the rear object does.

5.4 Model-based light field reconstruction

In this section, we consider light field reconstruction as solving an inverse problem based on the forward model we described in the previous section. To achieve better reconstruction, we add a sparsity-promoting prior, which works well in many inverse problems, to our reconstruction. We will first derive the algorithm we used to solve sparsity-regularized minimization problems. Then, a new analysis light field prior without explicit depth estimation is proposed to capture the elongated structure of light fields.

5.4.1 Sparsity-regularized minimization using ADMM

Consider a focal stack of size n generated by taking pictures using n different focus settings. Let \mathbf{x} denote the discrete light field, \mathbf{B}_i denote the convolution matrix of the i th focus setting, \mathbf{S}_i denote the diagonal sampling matrix of the i th focus setting, and \mathbf{y}_i denote the i th image in the focal stack. Suppose Gaussian noise is added when taking photos. We will have $\|\mathbf{y}_i - \mathbf{S}_i\mathbf{B}_i\mathbf{x}\|_2 \leq \epsilon_i$, where ϵ_i is proportional to the noise level of the i th image. In addition, as shown before, the light field should be non-negative, i.e., $\mathbf{x} \succeq \mathbf{0}$. Combining with the sparsity regularizer, our proposed model-based light field reconstruction formulation becomes

$$\hat{\mathbf{x}} \in \arg \min_{\mathbf{x} \succeq \mathbf{0}} \|\Theta\mathbf{x}\|_1 \text{ s.t. } \|\mathbf{y}_i - \mathbf{S}_i\mathbf{B}_i\mathbf{x}\|_2 \leq \epsilon_i \text{ for } i = 1, \dots, n, \quad (5.14)$$

where Θ is a sparsifying matrix, e.g., the discrete framelet transform matrix [69], the discrete curvelet transform matrix [70], and the finite different matrix \mathbf{C} . To solve this *constrained* convex optimization problem, we apply C-SALSA [45], which is an ADMM algorithm for constrained convex optimization problems.

5.4.2 Prior of the elongated light field structure

So far, we have formulated our model-based light field reconstruction problem by considering light fields as natural images and applying popular natural image priors, such as framelet prior, curvelet prior and TV prior, in our reconstruction. However, light fields can be very different from natural images. If we take a closer look at the light field data, we can find light field data has a special *elongated structure* with *linear characteristics* as we showed in (5.1). Some light field priors have been proposed in recent years. The classical prior used in early light field sampling and reconstruction works [71–73] is based on the band-limited assumption of captured light field. In this case, light field is assumed to be pre-filtered by a proper low-pass filter to meet the Nyquist rate. Therefore, the classical prior penalizes high frequency components of the reconstructed light field *isotropically*. The simplest way to express this band-limited assumption is to use a zero-mean isotropic Gaussian prior on the high-passed light field, i.e., $\mathbf{H}\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \sigma_0\mathbf{I})$, where \mathbf{H} denotes matrix of a proper high pass filter based on the frequency analysis of plenoptic sampling. It is easy to see that this classical prior does not capture the elongated structure of light fields at all. To account for the strong elongated structure of light fields, Anat Levin *et al.* [74] proposed a light field prior using a mixture of oriented Gaussians, where each

Gaussian component corresponds to a depth interpretation of the scene. That is, if the scene depth (and hence light field orientation) is known, they define an anisotropic Gaussian prior that accounts for the oriented structure. Therefore, they penalize the amount

$$\mathbf{x}'\Psi_S^{-1}\mathbf{x} \triangleq \sum_i \left(\frac{1}{\sigma_s} |\mathbf{g}'_{s(i),i}\mathbf{x}| + \frac{1}{\sigma_0} |\mathbf{g}'_{0,i}\mathbf{x}| \right), \quad (5.15)$$

where $\mathbf{g}'_{s(i),i}\mathbf{x}$ computes the directional derivative centered at the i th light field entry along direction $s(i)$, which is the orientation at the i th light field entry, $\mathbf{g}'_{0,i}\mathbf{x}$ computes the derivative at the i th light field entry in horizontal/spatial direction, and $\sigma_s \ll \sigma_0$. This prior constrains the light field tighter than the classical prior. However, inference is more complicated since the correct mixture of oriented Gaussians, or the scene depth, needs to be estimated ahead. Unfortunately, depth estimation is usually noise-sensitive and *requires human assistance*. Although there are many depth estimation algorithms available, a precious and fully automatically generated depth map is still not achievable. Therefore, priors that involve depth estimation are undesirable. A good light field prior should be a prior that has a linear analysis form, captures the elongated structure of light fields, and is *invariant to the scene depth*.

Consider a scene with characteristic $\{x - s_0u = k \mid k \in \mathbb{R}\}$. Note that here we consider the case of 2D light fields, i.e., light fields in flatland. It is fairly easy to extend the 2D result to 4D. Let u_0 be some value on the u -axis. Since the light field has characteristic $\{x - s_0u = k \mid k \in \mathbb{R}\}$, $L(x, u) = L(x_0, u_0) = L_{u_0}(x_0)$ for some x_0 . x_0 can be found by solving the equation $x - s_0u = k = x_0 - s_0u_0$ for some k , and the solution is

$$x_0 = x - s_0(u - u_0). \quad (5.16)$$

Therefore, we have the identity

$$L(x, u) = L_{u_0}(x - s_0(u - u_0)). \quad (5.17)$$

Now, let $B^s(x) \triangleq \int_{u \in \text{rect}_{(R,0)}} L(x + su, u) du$, where $\text{rect}_{(r,c)}$ denotes an interval centered at c with radius R . Similar to the light field transformations we derived before, the shear-projection operation can be expressed as a horizontal slice of a 2D convolution. Furthermore, after inserting the identity in (5.17) to the definition of B^s , we can represent B^s as

$$B^s(x) = \left(L_{u_0} * \frac{1}{|s-s_0|} \text{rect}_{(R|s-s_0|, -s_0u_0)} \right)(x), \quad (5.18)$$

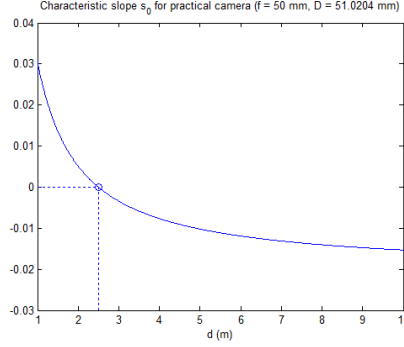


Figure 5.9: Characteristic slope s_0 of practical camera.

so a horizontal slice of a 2D convolution can be expressed as a 1D convolution of a horizontal slice given the linear characteristic $\{x - s_0 u = k \mid k \in \mathbb{R}\}$. However, $B^s(x)$ is still s_0 -dependent. To remove the s_0 -dependency, suppose $|s_0| \ll S$ for some positive S . This assumption does make sense in practice. Consider a 35 mm equivalent camera with lens focal length 50 mm, which is typical for photography of street and documentary. Suppose the camera is focused at scene depth 2.5 m, that is, $D = 1/(1/50 - 1/2500) \approx 51.0204$ mm. The corresponding characteristic (inverse) slope s_0 for scene depth between 1 m and 10 m can be computed by comparing with (5.1) and is shown in Figure 5.9. As can be seen in Figure 5.9, the characteristic slope falls in a relative small interval, which can be bounded by, for example, $S = 3$.

Let $\bar{B}(x) \triangleq \int_{-S}^S B^s(x) ds$. On the one hand, since each $B^s(x)$ is a horizontal slice of a 2D convolution of $L(x, u)$ with a projection kernel, $\bar{B}(x)$ can be represented as a horizontal slice of a 2D convolution of $L(x, u)$ with some kernel $g(x, u)$. On the other hand, by using the property of linear characteristic in (5.18), we can also express $\bar{B}(x)$ as

$$\bar{B}(x) = \int_{-S}^S B^s(x) ds = (L_0 * \phi_{s_0})(x), \quad (5.19)$$

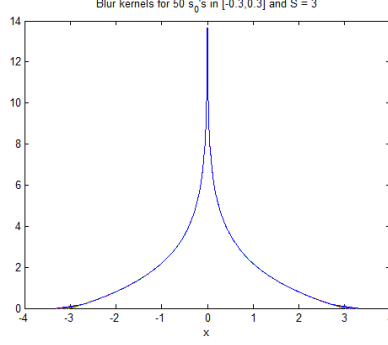


Figure 5.10: Depth-invariant blur kernel.

where

$$\begin{aligned} \phi_{s_0}(x) &= \int_{-S}^S \frac{1}{|s-s_0|} \text{rect}_{(R|s-s_0|,0)}(x) \, ds \\ &= \ln\left(\frac{(S-s_0)R}{|x|}\right) \cdot \mathbb{1}(S-s_0 \leq |x|/R) + \ln\left(\frac{(S+s_0)R}{|x|}\right) \cdot \mathbb{1}(S+s_0 \leq |x|/R) \end{aligned} \quad (5.20)$$

$$\approx 2\ln\left(\frac{SR}{|x|}\right) \cdot \mathbb{1}(SR \leq |x|) \quad (5.21)$$

$$\begin{aligned} &= \phi_0(x) \\ &= \int_{u \in \text{rect}_{(R,0)}} g(x, u) \, du \end{aligned} \quad (5.22)$$

is depth-invariant blur kernel [75, 76]. Figure 5.10 shows the approximate depth-invariant blur kernels for $s_0 \in [-0.3, 0.3]$ and $S = 3$. As can be seen in Figure 5.10, the 50 blur kernels are almost the same except for the tail distributions.

The discrete version of g and ϕ_0 can be found using the same technique we used when deriving the discrete light fields. Analytically, they are

$$g[n, m] = \int_{-R}^R \left(\int_{-S}^S (\text{rect}_{\Delta_x} * \text{rect}_{\Delta_x})(su + n\Delta_x) \, ds \right) \text{rect}_{\Delta_u}(u + m\Delta_u) \, du \quad (5.23)$$

and

$$\phi_0[n] = (\phi_0 * \text{rect}_{\Delta_x} * \text{rect}_{\Delta_x})(n\Delta_x) . \quad (5.24)$$

Although both g and ϕ_0 have closed-form expressions, for simplicity, we just compute them numerically. Finally, we do not have to always consider the entire light field. Since the support of g and ϕ_0 depend on R , if R is large, the number of samples that are free from boundary effect becomes fewer. Instead, we can think of the aperture as many overlapped sub-apertures and consider the elongated structure locally. In this

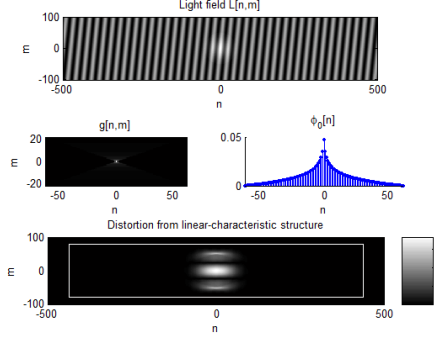


Figure 5.11: Demonstration of the depth-invariant prior.

case, we decompose the entire light field into multiple overlapped horizontal stripes. For each stripe, we compute the difference between the two expressions of $\bar{B}[n]$ using the filters derived in (5.23) and (5.24). If the stripe has the elongated structure, the 1D slice (except for boundary samples) must have small magnitude. Furthermore, we can stack these 1D slices and form a 2D matrix, which indicates the distortion from the elongated structure. To express this analysis form elongated structure indicator, let \mathbf{B}_{2D} be the 2D convolution matrix of g , \mathbf{B}_{1D} be the 1D convolution matrix of ϕ_0 , and \mathbf{S}_{ROI} denotes the sampling matrix of the region of interest. If \mathbf{x} is a light field with the elongated structure, then most of entries in $\mathbf{S}_{ROI}(\mathbf{B}_{2D} - \mathbf{B}_{1D})\mathbf{x}$ should be close to zero and have large magnitude only in occlusion region. Suppose the scene of interest has locally constant depth with finitely many depth discontinuities, $\mathbf{S}_{ROI}(\mathbf{B}_{2D} - \mathbf{B}_{1D})\mathbf{x}$ should be sparse. That is, $\mathbf{S}_{ROI}(\mathbf{B}_{2D} - \mathbf{B}_{1D})\mathbf{x} \sim \text{Laplacian}(\lambda\mathbf{I})$. Apparently, the null space of $\mathbf{S}_{ROI}(\mathbf{B}_{2D} - \mathbf{B}_{1D})$ is non-trivial. This depth-invariant prior may be a prior for the elongated structure. Figure 5.11 demonstrates the depth-invariant prior. In this demonstration, the light field $L[n, m]$ ($R = 1$ mm, $L = 10$ mm, and $\Delta_x = \Delta_u = 10^{-2}$ mm) is a superposition of an elongated structure with characteristic slope $s_0 = 0.1$ and a Gaussian disk at the center of the rectangular region. g and ϕ_0 are computed by setting $S = 3$ and stripe width to be $0.2R$. The wedge-shape $g[n, m]$ is due to superposition of infinitely many projection kernels. $\phi_0[n]$ is a long-tailed blur kernel. The white rectangular box in the bottom row of Figure 5.11 indicates the region of interest. As can be seen in Figure 5.11, $\mathbf{S}_{ROI}(\mathbf{B}_{2D} - \mathbf{B}_{1D})\mathbf{x}$ indicates regions that do not have locally elongated structure. Finally, to apply our proposed light field prior to the light field reconstruction problem, we simply set $\Theta = \mathbf{S}_{ROI}(\mathbf{B}_{2D} - \mathbf{B}_{1D})$ and solve the convex optimization problem (5.14) using C-SALSA [45].

5.5 The back-project filter method for light field reconstruction

In the proposed light field reconstruction scheme, we reconstruct the light field from a focal stack. We have already shown that a focal stack is samples of oblique projections of the light field. Suppose we define a focal stack diagram to be a collection of oblique projections of the light field. The proposed method reconstructs light fields from samples of the focal stack diagram. This is very similar to the X-ray CT image reconstruction problem, which reconstructs object from samples of the sinogram, where a sinogram denotes a collection of vertical projections of the object. Therefore, it is possible to solve the light field reconstruction problem by using methods that solve the X-ray CT image reconstruction problem.

One of the most well-known method in X-ray CT image reconstruction is called the back-project filtered (BPF) method. In the BPF method, one first gets a laminogram by back-projecting the sinogram. Because of the natural of vertical projection/back-projection, the back-projection results in a shift-invariant blur of the image, and therefore, one can reconstruct the image by removing the shift-invariant blur from the laminogram. In our light field reconstruction problem, we first obliquely back-project the focal stack diagram to get the laminogram. However, due to the oblique projection/back-projection, the blur is no longer shift-invariant. To solve this problem, we have to go back to our model. Suppose \mathbf{y}_i denotes the image taken by the i th focus setting, \mathbf{S}_i denotes the sampling matrix of the i th focus setting, and \mathbf{B}_i denotes the projection matrix of the i th focus setting. The measurements should satisfy the system of linear equations

$$\begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_n \end{pmatrix} \approx \begin{pmatrix} \mathbf{S}_1 \mathbf{B}_1 \\ \vdots \\ \mathbf{S}_n \mathbf{B}_n \end{pmatrix} \mathbf{x}. \quad (5.25)$$

Note that since $\mathbf{S}_i \mathbf{B}_i$ denotes the oblique projection operator, $\mathbf{B}_i' \mathbf{S}_i'$ should be the operator of oblique back-projection. Therefore, the oblique back-projection of focal stack diagram should satisfy the system of linear equations

$$\begin{pmatrix} \mathbf{S}_1 \mathbf{B}_1 \\ \vdots \\ \mathbf{S}_n \mathbf{B}_n \end{pmatrix}' \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_n \end{pmatrix} \approx \begin{pmatrix} \mathbf{S}_1 \mathbf{B}_1 \\ \vdots \\ \mathbf{S}_n \mathbf{B}_n \end{pmatrix}' \begin{pmatrix} \mathbf{S}_1 \mathbf{B}_1 \\ \vdots \\ \mathbf{S}_n \mathbf{B}_n \end{pmatrix} \mathbf{x}. \quad (5.26)$$

This equation is very similar to the *normal equation* we encounter in least-squares

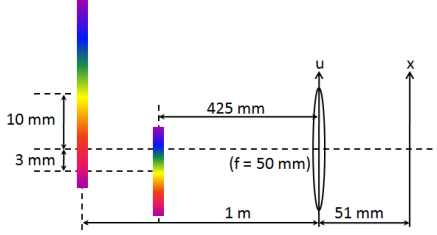


Figure 5.12: A scene with two planar objects at different depth.

problems. The only difference is that in our case, (5.25) is ill-posed, so does (5.26). However, we can still solve (5.26) by using conjugate gradient (CG) method, and this can be seen as a generalized BPF method for light field reconstruction.

5.6 Flatland simulations

In this section, we report flatland simulations based on the proposed algorithms described in the previous sections. Throughout the simulation, we consider the scene as shown in Figure 5.12. The focal length of the lens is set to be 50 mm. The separation between the main lens and the image sensor is 51 mm. The grid sizes Δ_u and Δ_x are set to be 10^{-2} mm, the width of sensor is 5 mm, and the radius of aperture is 1 mm. We consider the size of focal stack to be 16. We generate the focal stack by moving the sensor back and forth with α varying uniformly from 1 to 1.5. The sampling rate is about $16/200 = 8\%$. The maximum number of iterations is set to be 20000. Framelet, curvelet, TV, and the proposed light field priors are considered in the reconstruction. We also consider the BPF method in our simulation. The maximum number of CG iterations is set to be 1000.

Figure 5.13 shows the true and reconstructed light fields using different priors for the two-object scene shown in Figure 5.12. Figure 5.12(b) shows the reconstructed light field with the framelet prior. The RMS error is 24.6778. As can be seen in Figure 5.12(b), the reconstructed light field with the framelet prior suffers from blocky artifacts. This results from the separable basis and shift-variant property of framelets. We can also find that the reconstruction shows aliasing patterns, or the Moiré patterns, in the fast-varying region of the light field. Figure 5.12(c) shows the reconstructed light field with the curvelet prior. The RMS error is 37.4821. As can be seen in Figure 5.12(c), although the reconstructed light field with the curvelet prior recovers the elongated structure of light fields, it fails to satisfy the Lambertian scene assumption in our simulation. In addition, the reconstruction shows smear-like artifacts in the slow-varying region of the light field. Figure 5.12(d) shows the recon-

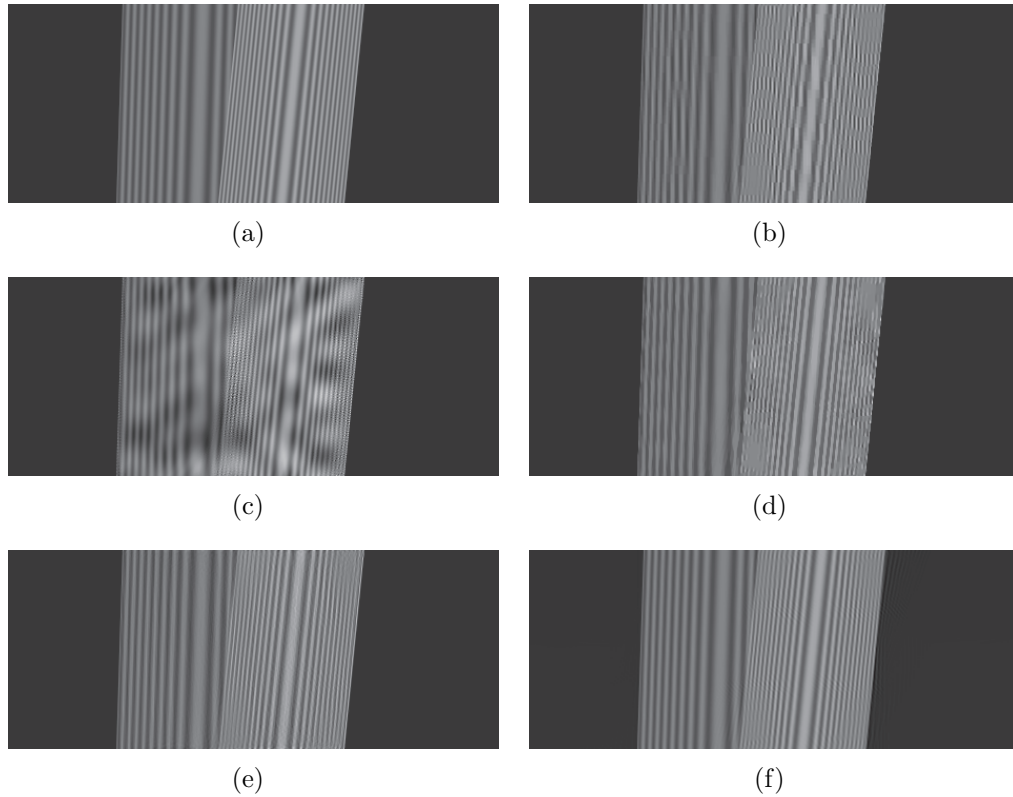


Figure 5.13: 2D flatland simulation: (a) The true light field of a scene in Figure 5.12. (b) The reconstructed light field using the framelet prior. (c) The reconstructed light field using the curvelet prior. (d) The reconstructed light field using the TV prior. (e) The reconstructed light field using the proposed light field prior. (f) The reconstructed light field using the BPF method.

structed light field with the TV prior. The RMS error is 22.7264. As can be seen in Figure 5.12(d), we can find that the TV prior prefers piecewise constant solutions, and therefore, some fine details of light field is smeared out in the reconstructed light field with TV prior, and the reconstruction suffers from staircase artifacts. Figure 5.12(e) shows the reconstructed light field with the proposed light field prior. The RMS error is 18.1181. As can be seen in Figure 5.12(e), we can find that the proposed light field prior captures the elongated structure of light fields well and recovers most of the fine details of the light field. Furthermore, the ℓ_1 objective function automatically handles the distribution of model mismatch error and suppresses the artifacts around the occlusion regions. Figure 5.12(f) shows the reconstructed light field using the BPF method. The RMS error is 16.1507. As can be seen in Figure 5.12(f), the BPF method reconstructs a relatively smooth light field which has the smallest RMS error. However, if we take a closer look at the reconstructed light field, we can find streak artifacts around the fast-varying region of the light field. In practice, we can use the reconstructed light field using the BPF method as the initial guess of the reconstruction with the light field prior for faster convergence. Figure 5.13(a) shows another example light field with three objects in the scene. Figure 5.13(b) and Figure 5.13(c) show the reconstructed light fields with the light field prior and the BPF method, respectively. Again, the reconstruction using the BPF method is smoother and with streak artifacts. On the other hand, the reconstruction with the light field prior is sharper and free from streak artifacts.

5.7 Conclusion

In this chapter, we proposed a fast algorithm to compute the projection of a linear-transformed light field and built a camera model using the discrete light fields. Based on the proposed camera model, we proposed to reconstruct the light field from a focal stack by solving a convex optimization problem with a sparsity-promoting prior. We found that existing natural image priors are not adequate for light fields. Therefore, we proposed an analysis light field prior which does not require explicit depth estimation. The proposed light field prior outperforms all the investigated natural image priors according to our flatland simulations. We also proposed a fast BPF method to find a good initial guess of the light field, which is useful for other iterative light field reconstruction methods.

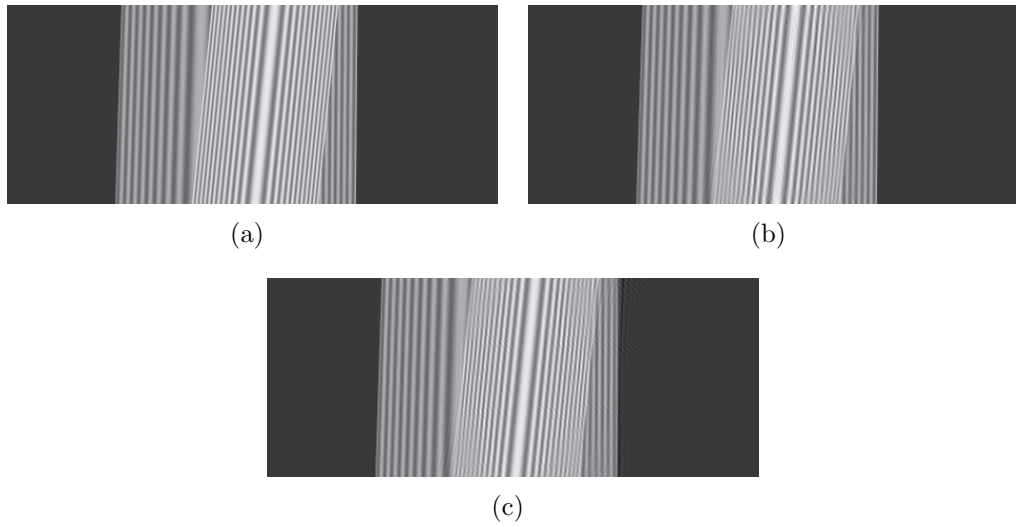


Figure 5.14: 2D flatland simulation: (a) The true light field of a scene with three planar objects at different depths. (b) The reconstructed light field using the proposed light field prior. (c) The reconstructed light field using the BPF method.

CHAPTER VI

Conclusion and future work

In this work, we developed a fast splitting-based ordered-subset algorithm for practical model-based X-ray CT image reconstruction and proposed a model-based light field reconstruction from a focal stack using ADMM. Experimental results showed promising preliminary results of both X-ray CT image and light field reconstructions. More rigorous and practical experiments have to be done in the future. For X-ray CT image reconstruction, the proposed future work is two-fold. In the theory part, we would like to investigate more about the convergence rate analysis of the proposed algorithm with the downward continuation approach and a more rigorous convergence analysis of the proposed algorithm with majorizations and for $M > 1$. The latter topic is closely related to the stochastic analysis in the machine learning literature and needs an accurate model (or “oracle”) of gradient errors using OS. We are also interested in the convergence analysis of the proposed inexact low-memory algorithm using the compressed variable splitting due to the potential impact on image reconstruction problems with lots of auxiliary variables, e.g., the prior image constrained compressed sensing CT reconstruction. In the application part, we would like to apply the proposed algorithm to CT reconstruction problems with more general noise model (i.e., other than the PWLS criterion). To do that, we can do successive quadratic majorization of the (Lipschitz) cost function. In fact, we use this trick very often in this work, so it looks promising that we can solve more general convex optimization problems, e.g., machine learning problems, using the proposed algorithm. Furthermore, we also think about applying techniques like smoothing (continuation of the smoothness parameter δ in the potential function) to solve reconstruction problems with high curvature regularizers. For light field reconstruction, thanks to the inherent similarity between X-ray CT image and light field reconstruction, it is natural to solve light field reconstruction problems using the same technique we used in X-ray CT image reconstruction. As a future work, we will build a more accurate forward

model of camera imaging process and investigate real 4D light field reconstructions with the proposed algorithm for X-ray CT.

APPENDICES

APPENDIX A

A (primal) convergence proof of the SB method for regularized least-squares problems

A.1 Introduction

Consider a regularized least-squares optimization problem with a general convex regularizer:

$$\hat{\mathbf{x}} \in \arg \min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \Phi(\Theta\mathbf{x}) \right\}, \quad (\text{A.1})$$

where \mathbf{y} is the noisy measurement, \mathbf{A} is the system matrix, Φ is a convex potential function, and Θ is an analysis regularization matrix. For example, when Φ is the ℓ_1 -norm and Θ is the discrete framelet transform matrix [69], the regularized least-squares problem (A.1) is a frame-based image restoration problem [78]; when Φ is a smooth “ ℓ_1 -like” potential function (such as the Huber function [18, 79] and the Fair function [61, 64]) and Θ is the finite difference matrix, the regularized least-squares problem (A.1) is an image restoration problem with an edge-preserving regularizer. To solve (A.1), one can use the split Bregman (SB) method proposed by Goldstein *et al.* [32], which solves an equivalent constrained minimization problem:

$$(\hat{\mathbf{x}}, \hat{\mathbf{v}}) \in \arg \min_{\mathbf{x}, \mathbf{v}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \Phi(\mathbf{v}) \right\} \quad \text{s.t.} \quad \mathbf{v} = \Theta\mathbf{x} \quad (\text{A.2})$$

This appendix is based on [77].

using the (alternating direction) augmented Lagrangian (AL) method. The iterates of the SB method are as follows:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg \min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \frac{\eta}{2} \|\Theta\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)}\|_2^2 \right\} \\ \mathbf{v}^{(k+1)} \in \arg \min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \|\Theta\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)}\|_2^2 \right\} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \Theta\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)}, \end{cases} \quad (\text{A.3})$$

where the \mathbf{x} -update is a least-squares problem, and the \mathbf{v} -update is a proximal mapping of Φ , which often can be solved efficiently, e.g., by soft-thresholding for the ℓ_1 potential.

To prove the convergence of the SB method, Esser [80] showed that the SB method is equivalent to the alternating direction method of multipliers (ADMM) [44, 45, 81, 82], and Setzer [83] showed that the SB method can be interpreted as the Douglas-Rachford splitting (DRS) method [44, 84, 85] applied to the dual problem. However, both [80] and [83] implicitly assume that Θ has full column rank, i.e., $\Theta'\Theta$ is invertible, and show the primal convergence proofs using [44, Theorem 8]. The full column rank condition holds when Θ is a tight frame as in frame-based image restoration problems. When Θ is the finite difference matrix as in edge-preserving image restoration problems (and also in [32]), this assumption does not hold anymore, and the the proofs in [80] and [83] are inapplicable. Differently, in [78], assuming all the inner updates in (A.3) are exact, Cai *et al.* proved the convergence of the SB method without using [44, Theorem 8] and therefore did not impose the “full column rank” assumption. In other words, the SB method is a convergent algorithm for any Θ if all the inner minimization problems in (A.3) are solved exactly! Unfortunately, when some of the inner updates are inexact, e.g., the \mathbf{x} -update in parallel magnetic resonance (MR) and X-ray computed tomography (CT) image reconstructions [8, 32, 86, 87], we still lack convergence proofs of the SB method.

A.2 The split Bregman method as an ADMM algorithm

To show the convergence of the inexact SB method, we first consider another constrained minimization problem that is also equivalent to (A.1) but uses two split variables:

$$(\hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{v}}) \in \arg \min_{\mathbf{x}, \mathbf{u}, \mathbf{v}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{u}\|_2^2 + \Phi(\mathbf{v}) \right\} \quad \text{s.t.} \quad \mathbf{u} = \mathbf{A}\mathbf{x}, \mathbf{v} = \Theta\mathbf{x}. \quad (\text{A.4})$$

The ADMM algorithm for this constrained minimization problem is [45]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg \min_{\mathbf{x}} \left\{ \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}\|_2^2 + \frac{\eta}{2} \|\Theta\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)}\|_2^2 \right\} \\ \mathbf{u}^{(k+1)} \in \arg \min_{\mathbf{u}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{u}\|_2^2 + \frac{\rho}{2} \|\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\|_2^2 \right\} \\ \mathbf{v}^{(k+1)} \in \arg \min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \|\Theta\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)}\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \Theta\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)}, \end{cases} \quad (\text{A.5})$$

where \mathbf{d} and \mathbf{e} are the scaled Lagrange multipliers (i.e., dual variables) of the split variables \mathbf{u} and \mathbf{v} , respectively, and $\rho, \eta > 0$ are the corresponding AL penalty parameters. By stacking \mathbf{u} and \mathbf{v} , we can represent the equality constraint in (A.4) more compactly as

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A} \\ \Theta \end{bmatrix}}_{\mathbf{S}} \mathbf{x}. \quad (\text{A.6})$$

When \mathbf{S} has full column rank, this ADMM algorithm (A.5) is convergent, even with inexact updates, providing the error in the inexact updates satisfies the conditions of [44, Theorem 8]. In many applications such as image restoration and X-ray CT image reconstruction, $\mathbf{A}'\mathbf{A}$ is a low-pass filter (but not necessarily shift-invariant). When $\Theta = \mathbf{C}$ is the finite difference matrix, $\Theta'\Theta$ is the Laplacian, which is a high-pass filter. The non-zero vectors in the null space of $\Theta'\Theta$ are usually not in the null space of $\mathbf{A}'\mathbf{A}$, and vice versa, so the null space of $\mathbf{S}'\mathbf{S} = \mathbf{A}'\mathbf{A} + \Theta'\Theta$ is usually $\{\mathbf{0}\}$. That is, \mathbf{S} usually has full column rank in applications like image restoration and X-ray CT image reconstruction! Therefore, (A.5) is a convergent ADMM algorithm that allows inexact updates for image restoration and X-ray CT image reconstruction according to [44, Theorem 8]. More specifically, $\mathbf{x}^{(k+1)}$ in (A.5) converges to $\hat{\mathbf{x}}$ if the error of the inner minimization problem (i.e., the ℓ_2 distance between the iterate and the optimum of the inner problem) is absolutely summable.

Now, let's take a closer look at (A.5). The \mathbf{u} -update in (A.5) has a closed-form solution

$$\mathbf{u}^{(k+1)} = \frac{\rho}{\rho+1} (\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{d}^{(k)}) + \frac{1}{\rho+1} \mathbf{y}. \quad (\text{A.7})$$

Combining with the \mathbf{d} -update in (A.5), we have the identity

$$\mathbf{u}^{(k+1)} + \rho\mathbf{d}^{(k+1)} = \mathbf{y} \quad (\text{A.8})$$

if we initialize \mathbf{d} as $\mathbf{d}^{(0)} = \rho^{-1} (\mathbf{y} - \mathbf{u}^{(0)})$. Substituting (A.8) into (A.5), we have the simplified ADMM iterates:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg \min_{\mathbf{x}} \left\{ \frac{\rho}{2} \|\mathbf{Ax} - \rho^{-1}\mathbf{y} - (1 - \rho^{-1})\mathbf{u}^{(k)}\|_2^2 + \frac{\eta}{2} \|\Theta\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)}\|_2^2 \right\} \\ \mathbf{u}^{(k+1)} = \frac{\rho}{\rho+1}\mathbf{Ax}^{(k+1)} + \frac{1}{\rho+1}\mathbf{u}^{(k)} \\ \mathbf{v}^{(k+1)} \in \arg \min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \|\Theta\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)}\|_2^2 \right\} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \Theta\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)}. \end{cases} \quad (\text{A.9})$$

By comparing the SB method (A.3) and the simplified ADMM algorithm (A.9) side by side, we can easily find that they have common \mathbf{v} - and \mathbf{e} -updates. The \mathbf{u} -update in (A.9) can be seen as a perturbation of its \mathbf{x} -update. In fact, when $\rho = 1$, the \mathbf{x} -update in (A.9) is independent of $\mathbf{u}^{(k)}$, and the simplified ADMM algorithm (A.9) reduces to the SB method (A.3). In other words, the SB method is a convergent ADMM algorithm when we solve a regularized least-squares problem, and this proves the convergence of the inexact SB method for image restoration and X-ray CT image reconstruction provided \mathbf{S} in (A.6) has full column rank, and the inner minimization error is absolute summable! Note that \mathbf{S} has full column rank in many applications whereas Θ often does not. This is the main difference between our new convergence condition and the conventional one.

APPENDIX B

Frequency analysis and parameter selection of ADMM: the quadratic case

B.1 Frequency analysis

In the previous appendix, we showed that when the data-fitting term is quadratic, the SB method is a convergent ADMM algorithm, and therefore proved the convergence of the inexact SB method. Although the convergence of the SB method for general convex data-fitting term is still an open problem, the convergence proof in Section A.2 is applicable to many popular image reconstruction problems. Note that the equivalence of the SB method and the ADMM algorithm holds for the choice $\rho = 1$; however, the ADMM algorithm (A.9) is convergent for any $\rho > 0$. Thus, it seems likely that the ADMM algorithm will be faster than the SB method if ρ is selected appropriately. To have a more concrete example and mathematically tractable analysis, we analyze the convergence rate properties of (A.9) for a quadratically regularized image restoration problem:

$$\hat{\mathbf{x}} \in \arg \min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \frac{\alpha}{2} \|\mathbf{C}\mathbf{x}\|_2^2 \right\}, \quad (\text{B.1})$$

where \mathbf{y} denotes the noisy blurred measurement of an image \mathbf{x} , degraded by a degradation matrix \mathbf{A} , $\alpha > 0$ is the regularization parameter, and \mathbf{C} denotes the tall masked finite difference matrix in multiple directions. To simplify our analysis, we will further assume that both $\mathbf{A}'\mathbf{A}$ and $\mathbf{C}'\mathbf{C}$ are approximately block circulant with circulant

This appendix is based on [77].

blocks (BCCB), i.e., $\mathbf{A}'\mathbf{A} \approx \mathbf{U}\mathbf{\Lambda}\mathbf{U}'$ and $\mathbf{C}'\mathbf{C} \approx \mathbf{U}\mathbf{\Omega}\mathbf{U}'$, where $\mathbf{\Lambda} \triangleq \text{diag}\{\lambda_i \geq 0\}$, $\mathbf{\Omega} \triangleq \text{diag}\{\omega_i \geq 0\}$, and \mathbf{U} denotes the normalized 2D inverse DFT matrix.

Clearly, the quadratically regularized image restoration problem (B.1) is simply an instance of the regularized least-squares problem (A.1) with $\mathbf{\Theta} \triangleq \mathbf{C}$ and $\Phi(\cdot) \triangleq \frac{\alpha}{2} \|\cdot\|_2^2$. Therefore, the simplified ADMM algorithm solving (B.1) is:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg \min_{\mathbf{x}} \left\{ \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \rho^{-1}\mathbf{y} - (1 - \rho^{-1})\mathbf{u}^{(k)}\|_2^2 + \frac{\eta}{2} \|\mathbf{C}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)}\|_2^2 \right\} \\ \mathbf{u}^{(k+1)} = \frac{\rho}{\rho+1}\mathbf{A}\mathbf{x}^{(k+1)} + \frac{1}{\rho+1}\mathbf{u}^{(k)} \\ \mathbf{v}^{(k+1)} \in \arg \min_{\mathbf{v}} \left\{ \frac{\alpha}{2} \|\mathbf{v}\|_2^2 + \frac{\eta}{2} \|\mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)}\|_2^2 \right\} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)}. \end{cases} \quad (\text{B.2})$$

Furthermore, since Φ is quadratic, it has a linear proximal mapping, and therefore, the \mathbf{v} -update in (B.2) has a closed-form solution

$$\mathbf{v}^{(k+1)} = \frac{\eta}{\eta+\alpha} (\mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{e}^{(k)}) . \quad (\text{B.3})$$

Again, using the same trick as before, we find that the dual variable \mathbf{e} is also redundant, yielding the identity

$$\alpha\mathbf{v}^{(k+1)} + \eta\mathbf{e}^{(k+1)} = \mathbf{0} \quad (\text{B.4})$$

if we initialize \mathbf{e} as $\mathbf{e}^{(0)} = -\alpha\eta^{-1}\mathbf{v}^{(0)}$. Substituting (B.4) into (B.2), the ADMM iterates (B.2) simplify to:

$$\begin{cases} \mathbf{x}^{(k+1)} = (\rho\mathbf{A}'\mathbf{A} + \eta\mathbf{C}'\mathbf{C})^{-1} (\mathbf{A}'\mathbf{y} + (\rho - 1)\mathbf{A}'\mathbf{u}^{(k)} + (\eta - \alpha)\mathbf{C}'\mathbf{v}^{(k)}) \\ \mathbf{u}^{(k+1)} = \frac{\rho}{\rho+1}\mathbf{A}\mathbf{x}^{(k+1)} + \frac{1}{\rho+1}\mathbf{u}^{(k)} \\ \mathbf{v}^{(k+1)} = \frac{\eta}{\eta+\alpha}\mathbf{C}\mathbf{x}^{(k+1)} + \frac{\alpha}{\eta+\alpha}\mathbf{v}^{(k)}. \end{cases} \quad (\text{B.5})$$

To further simplify (B.5), let's denote

$$\begin{cases} \mathbf{s} \triangleq (\rho\mathbf{A}'\mathbf{A} + \eta\mathbf{C}'\mathbf{C})^{-1} \mathbf{A}'\mathbf{y} \\ \mathbf{P} \triangleq (\rho - 1) (\rho\mathbf{A}'\mathbf{A} + \eta\mathbf{C}'\mathbf{C})^{-1} \mathbf{A}' \\ \mathbf{Q} \triangleq (\eta - \alpha) (\rho\mathbf{A}'\mathbf{A} + \eta\mathbf{C}'\mathbf{C})^{-1} \mathbf{C}' . \end{cases} \quad (\text{B.6})$$

It follows that

$$\begin{cases} \mathbf{x}^{(k+1)} = \mathbf{s} + \mathbf{P}\mathbf{u}^{(k)} + \mathbf{Q}\mathbf{v}^{(k)} \\ \mathbf{u}^{(k+1)} = \frac{\rho}{\rho+1}\mathbf{A}(\mathbf{s} + \mathbf{P}\mathbf{u}^{(k)} + \mathbf{Q}\mathbf{v}^{(k)}) + \frac{1}{\rho+1}\mathbf{u}^{(k)} \\ \mathbf{v}^{(k+1)} = \frac{\eta}{\eta+\alpha}\mathbf{C}(\mathbf{s} + \mathbf{P}\mathbf{u}^{(k)} + \mathbf{Q}\mathbf{v}^{(k)}) + \frac{\alpha}{\eta+\alpha}\mathbf{v}^{(k)}, \end{cases} \quad (\text{B.7})$$

and we have the transition equation of the split variables:

$$\begin{bmatrix} \mathbf{u}^{(k+1)} \\ \mathbf{v}^{(k+1)} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\rho}{\rho+1}\mathbf{A}\mathbf{P} + \frac{1}{\rho+1}\mathbf{I}_n & \frac{\rho}{\rho+1}\mathbf{A}\mathbf{Q} \\ \frac{\eta}{\eta+\alpha}\mathbf{C}\mathbf{P} & \frac{\eta}{\eta+\alpha}\mathbf{C}\mathbf{Q} + \frac{\alpha}{\eta+\alpha}\mathbf{I}_m \end{bmatrix}}_{\mathbf{G}} \begin{bmatrix} \mathbf{u}^{(k)} \\ \mathbf{v}^{(k)} \end{bmatrix} + \begin{bmatrix} \frac{\rho}{\rho+1}\mathbf{A}\mathbf{s} \\ \frac{\eta}{\eta+\alpha}\mathbf{C}\mathbf{s} \end{bmatrix}. \quad (\text{B.8})$$

Since we already know that the two-split ADMM algorithm (B.2) is convergent if \mathbf{A} and \mathbf{C} have disjoint null spaces (except for $\{\mathbf{0}\}$) [44], the split variables in (B.2) should converge linearly with rate of convergence $\varrho(\mathbf{G})$, where $\varrho(\cdot)$ denotes the spectral radius of a matrix. However, what we really care about is the convergence rate of \mathbf{x} . To find the convergence rate of \mathbf{x} , consider

$$\begin{aligned} & \mathbf{x}^{(k+1)} - \mathbf{s} \\ &= \begin{bmatrix} \mathbf{P} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(k)} \\ \mathbf{v}^{(k)} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{P} & \mathbf{Q} \end{bmatrix} \left(\begin{bmatrix} \frac{\rho}{\rho+1}\mathbf{A}\mathbf{P} + \frac{1}{\rho+1}\mathbf{I}_n & \frac{\rho}{\rho+1}\mathbf{A}\mathbf{Q} \\ \frac{\eta}{\eta+\alpha}\mathbf{C}\mathbf{P} & \frac{\eta}{\eta+\alpha}\mathbf{C}\mathbf{Q} + \frac{\alpha}{\eta+\alpha}\mathbf{I}_m \end{bmatrix} \begin{bmatrix} \mathbf{u}^{(k-1)} \\ \mathbf{v}^{(k-1)} \end{bmatrix} + \begin{bmatrix} \frac{\rho}{\rho+1}\mathbf{A}\mathbf{s} \\ \frac{\eta}{\eta+\alpha}\mathbf{C}\mathbf{s} \end{bmatrix} \right) \\ &= \left(\frac{\rho}{\rho+1}\mathbf{P}\mathbf{A} + \frac{1}{\rho+1}\mathbf{I}_n + \frac{\eta}{\eta+\alpha}\mathbf{Q}\mathbf{C} \right) \mathbf{P}\mathbf{u}^{(k-1)} \\ & \quad + \left(\frac{\rho}{\rho+1}\mathbf{P}\mathbf{A} + \frac{\eta}{\eta+\alpha}\mathbf{Q}\mathbf{C} + \frac{\alpha}{\eta+\alpha}\mathbf{I}_n \right) \mathbf{Q}\mathbf{v}^{(k-1)} + \left(\frac{\rho}{\rho+1}\mathbf{P}\mathbf{A} + \frac{\eta}{\eta+\alpha}\mathbf{Q}\mathbf{C} \right) \mathbf{s}. \quad (\text{B.9}) \end{aligned}$$

Unfortunately, (B.9) is not a transition equation of \mathbf{x} (or $\mathbf{x} - \mathbf{s}$), so we cannot find the linear convergence rate of \mathbf{x} in general, except for three cases: (1) $\rho = 1$, (2) $\eta = \alpha$, and (3) $\rho = \eta/\alpha$.

B.1.1 Case I

When $\rho = 1$, \mathbf{P} becomes a zero matrix, and the split variable \mathbf{u} is redundant. In fact, the two-split ADMM algorithm (B.2) reduces to the SB method when $\rho = 1$ as

shown in Section A.2. In this case, we have

$$\mathbf{v}^{(k+1)} = \left(\frac{\eta}{\eta+\alpha} \mathbf{C}\mathbf{Q} + \frac{\alpha}{\eta+\alpha} \mathbf{I}_m \right) \mathbf{v}^{(k)} + \frac{\eta}{\eta+\alpha} \mathbf{C}\mathbf{s} \quad (\text{B.10})$$

and

$$\begin{aligned} & \mathbf{x}^{(k+1)} - \mathbf{s} \\ &= \left(\frac{\eta}{\eta+\alpha} \mathbf{Q}\mathbf{C} + \frac{\alpha}{\eta+\alpha} \mathbf{I}_n \right) \mathbf{Q}\mathbf{v}^{(k-1)} + \frac{\eta}{\eta+\alpha} \mathbf{Q}\mathbf{C}\mathbf{s} \\ &= \underbrace{\left(\frac{\eta}{\eta+\alpha} \mathbf{Q}\mathbf{C} + \frac{\alpha}{\eta+\alpha} \mathbf{I}_n \right)}_{\mathbf{H}_1} (\mathbf{x}^{(k)} - \mathbf{s}) + \frac{\eta}{\eta+\alpha} \mathbf{Q}\mathbf{C}\mathbf{s}. \end{aligned} \quad (\text{B.11})$$

Because the two-split ADMM algorithm (B.2) is convergent, it follows that \mathbf{x} converges linearly to the solution $\hat{\mathbf{x}}$ with rate $\varrho(\mathbf{H}_1)$. Now, applying our BCCB approximations of $\mathbf{A}'\mathbf{A}$ and $\mathbf{C}'\mathbf{C}$, we can approximate the transition matrix \mathbf{H}_1 as

$$\begin{aligned} \mathbf{H}_1 &= \frac{\eta}{\eta+\alpha} ((\eta - \alpha) (\mathbf{A}'\mathbf{A} + \eta\mathbf{C}'\mathbf{C})^{-1} \mathbf{C}') \mathbf{C} + \frac{\alpha}{\eta+\alpha} \mathbf{I}_n \\ &\approx \mathbf{U} \text{diag} \left\{ \frac{\eta}{\eta+\alpha} \frac{(\eta-\alpha)\omega_i}{\lambda_i + \eta\omega_i} + \frac{\alpha}{\eta+\alpha} \right\} \mathbf{U}' \\ &= \mathbf{U} \text{diag} \left\{ \frac{\eta}{\eta+\alpha} \frac{\alpha\lambda_i + \eta^2\omega_i}{\eta\lambda_i + \eta^2\omega_i} \right\} \mathbf{U}' \\ &= \mathbf{U} \text{diag} \left\{ s_1(\delta_i) \triangleq \frac{\eta}{\eta+\alpha} \frac{\alpha + \eta^2\delta_i}{\eta + \eta^2\delta_i} \right\} \mathbf{U}', \end{aligned} \quad (\text{B.12})$$

where $\delta_i \triangleq \omega_i/\lambda_i \geq 0$ is the ratio of the spectra of $\mathbf{C}'\mathbf{C}$ and $\mathbf{A}'\mathbf{A}$. Note that for any non-negative δ , $s'_1(\delta) = \frac{\eta}{\eta+\alpha} \frac{\eta-\alpha}{(\eta\delta+1)^2}$ is greater zero if $\eta > \alpha$, and it is less than zero if $\eta < \alpha$. When $\eta > \alpha$, $\varrho(\mathbf{H}_1)$, i.e., $\max_i s_1(\delta_i)$, is determined by δ_{\max} , and we can find the optimal AL penalty parameter

$$\eta^* = \arg \min_{\eta} \left\{ \frac{\eta}{\eta+\alpha} \frac{\alpha + \eta^2\delta_{\max}}{\eta + \eta^2\delta_{\max}} \right\} = \sqrt{\alpha/\delta_{\max}}. \quad (\text{B.13})$$

Note that (B.13) holds only if $\eta > \alpha$; therefore, $\eta^* = \sqrt{\alpha/\delta_{\max}}$ only if $\delta_{\max} < \alpha^{-1}$. Similarly, when $\eta < \alpha$, $\varrho(\mathbf{H}_1)$ is determined by δ_{\min} . Follow the same procedure, we have $\eta^* = \sqrt{\alpha/\delta_{\min}}$ only if $\delta_{\min} > \alpha^{-1}$. Finally, for the case that $\delta_{\min} < \alpha^{-1} < \delta_{\max}$, $\eta^* = \alpha$ because both $\eta^* > \alpha$ and $\eta^* < \alpha$ lead to a contradiction to the condition $\delta_{\min} < \alpha^{-1} < \delta_{\max}$. Summarizing, the optimal AL penalty parameter of the SB method for solving the quadratically regularized image restoration problem (B.1) is

$$\eta^* = \sqrt{\alpha/\gamma}, \quad (\text{B.14})$$

where $\gamma \triangleq \text{median}\{\delta_{\min}, \delta_{\max}, \alpha^{-1}\}$.

B.1.2 Case II

When $\eta = \alpha$, \mathbf{Q} becomes a zero matrix, and the split variable \mathbf{v} is redundant. Furthermore, we can easily prove that when $\eta = \alpha$, the two-split ADMM algorithm (B.2) reduces to the alternating direction AL method that solves the constrained minimization problem:

$$(\hat{\mathbf{x}}, \hat{\mathbf{u}}) \in \arg \min_{\mathbf{x}, \mathbf{u}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{u}\|_2^2 + \frac{\alpha}{2} \|\mathbf{C}\mathbf{x}\|_2^2 \right\} \text{ s.t. } \mathbf{u} = \mathbf{A}\mathbf{x} \quad (\text{B.15})$$

that is also equivalent to (B.1). In this case, we have

$$\mathbf{u}^{(k+1)} = \left(\frac{\rho}{\rho+1} \mathbf{A}\mathbf{P} + \frac{1}{\rho+1} \mathbf{I}_n \right) \mathbf{u}^{(k)} + \frac{\rho}{\rho+1} \mathbf{A}\mathbf{s} \quad (\text{B.16})$$

and

$$\begin{aligned} & \mathbf{x}^{(k+1)} - \mathbf{s} \\ &= \left(\frac{\rho}{\rho+1} \mathbf{P}\mathbf{A} + \frac{1}{\rho+1} \mathbf{I}_n \right) \mathbf{P}\mathbf{u}^{(k-1)} + \frac{\rho}{\rho+1} \mathbf{P}\mathbf{A}\mathbf{s} \\ &= \underbrace{\left(\frac{\rho}{\rho+1} \mathbf{P}\mathbf{A} + \frac{1}{\rho+1} \mathbf{I}_n \right)}_{\mathbf{H}_2} (\mathbf{x}^{(k)} - \mathbf{s}) + \frac{\rho}{\rho+1} \mathbf{P}\mathbf{A}\mathbf{s}. \end{aligned} \quad (\text{B.17})$$

Follow the same trick, we can approximate the transition matrix \mathbf{H}_2 as

$$\begin{aligned} \mathbf{H}_2 &= \frac{\rho}{\rho+1} \left((\rho-1) (\rho\mathbf{A}'\mathbf{A} + \alpha\mathbf{C}'\mathbf{C})^{-1} \mathbf{A}' \right) \mathbf{A} + \frac{1}{\rho+1} \mathbf{I}_n \\ &\approx \mathbf{U} \text{diag} \left\{ \frac{\rho}{\rho+1} \frac{(\rho-1)\lambda_i}{\rho\lambda_i + \alpha\omega_i} + \frac{1}{\rho+1} \right\} \mathbf{U}' \\ &= \mathbf{U} \text{diag} \left\{ \frac{\rho}{\rho+1} \frac{\rho^2\lambda_i + \alpha\omega_i}{\rho^2\lambda_i + \alpha\rho\omega_i} \right\} \mathbf{U}' \\ &= \mathbf{U} \text{diag} \left\{ s_2(\delta_i) \triangleq \frac{\rho}{\rho+1} \frac{\rho^2 + \alpha\delta_i}{\rho^2 + \alpha\rho\delta_i} \right\} \mathbf{U}', \end{aligned} \quad (\text{B.18})$$

and the optimal AL penalty parameter ρ^* will be

$$\rho^* = \sqrt{\alpha\gamma}. \quad (\text{B.19})$$

B.1.3 Case III

Finally, when $\rho = \eta/\alpha$, we have the identity $\frac{1}{\rho+1} = \frac{\alpha}{\eta+\alpha}$ and we rewrite (B.9) as

$$\begin{aligned}
& \mathbf{x}^{(k+1)} - \mathbf{s} \\
&= \left(\frac{\eta}{\eta+\alpha} \mathbf{P}\mathbf{A} + \frac{\eta}{\eta+\alpha} \mathbf{Q}\mathbf{C} + \frac{\alpha}{\eta+\alpha} \mathbf{I}_n \right) (\mathbf{P}\mathbf{u}^{(k-1)} + \mathbf{Q}\mathbf{v}^{(k-1)}) + \frac{\eta}{\eta+\alpha} (\mathbf{P}\mathbf{A} + \mathbf{Q}\mathbf{C}) \mathbf{s} \\
&= \underbrace{\left(\frac{\eta}{\eta+\alpha} \mathbf{P}\mathbf{A} + \frac{\eta}{\eta+\alpha} \mathbf{Q}\mathbf{C} + \frac{\alpha}{\eta+\alpha} \mathbf{I}_n \right)}_{\mathbf{H}_3} (\mathbf{x}^{(k)} - \mathbf{s}) + \frac{\eta}{\eta+\alpha} (\mathbf{P}\mathbf{A} + \mathbf{Q}\mathbf{C}) \mathbf{s}. \tag{B.20}
\end{aligned}$$

The transition matrix \mathbf{H}_3 is approximately

$$\begin{aligned}
\mathbf{H}_3 &= \frac{\eta}{\eta+\alpha} \left(\frac{\eta}{\alpha} \mathbf{A}'\mathbf{A} + \eta \mathbf{C}'\mathbf{C} \right)^{-1} \left(\left(\frac{\eta}{\alpha} - 1 \right) \mathbf{A}'\mathbf{A} + (\eta - \alpha) \mathbf{C}'\mathbf{C} \right) + \frac{\alpha}{\eta+\alpha} \mathbf{I}_n \\
&\approx \mathbf{U} \text{diag} \left\{ \frac{\eta}{\eta+\alpha} \left(\frac{(\eta-\alpha)\lambda_i + \alpha(\eta-\alpha)\omega_i}{\eta\lambda_i + \alpha\eta\omega_i} \right) + \frac{\alpha}{\eta+\alpha} \right\} \mathbf{U}' \\
&= \mathbf{U} \text{diag} \left\{ \frac{\eta}{\eta+\alpha} \right\} \mathbf{U}' \\
&= \mathbf{U} \text{diag} \left\{ s_3(\delta_i) \triangleq \frac{\eta}{\eta+\alpha} \right\} \mathbf{U}'. \tag{B.21}
\end{aligned}$$

Surprisingly, \mathbf{H}_3 has a uniform spectrum, and $\varrho(\mathbf{H}_3) = \eta/(\eta + \alpha)$. Theoretically, we can achieve arbitrarily fast asymptotic convergence rate in this quadratic case by choosing

$$\eta^* \approx 0. \tag{B.22}$$

However, a smaller AL penalty parameter leads to a larger step size. When η is too small, we might encounter overshoots at the beginning and oscillation as the algorithm proceeds. Therefore, in practice, η^* cannot be arbitrarily small.

B.2 Parameter selection for image restoration problems

This section considers parameter selection of ADMM algorithms for image restoration in practical situations, where $\mathbf{A}'\mathbf{A}$ is a non-invertible low-pass filter, $\mathbf{C}'\mathbf{C}$ is a non-invertible high-pass filter, some frequency band is non-zero only for $\mathbf{A}'\mathbf{A}$ (such as the DC component), and some frequency band is non-zero only for $\mathbf{C}'\mathbf{C}$ (such as the extremely high frequency component). In this case, δ_i has an extremely huge dynamic range, i.e., $\delta_{\min} \approx 0$ and $\delta_{\max} \approx \infty$. Therefore, for most cases, the optimal AL penalty parameter η^* of the SB method (B.14) will be α . Furthermore, the optimal AL penalty parameter ρ^* of the two-split ADMM algorithm when $\eta = \alpha$ (Case II) is one, which then reverts to Case I, i.e., the SB method. Hence, once η is chosen to be

the optimal η of the SB method, the optimal two-split ADMM algorithm is the SB method itself!

Now, consider the case that η is suboptimal, i.e., $\eta \neq \alpha$. When $\eta > \alpha$, $\delta_{\max} \approx \infty$ determines the asymptotic convergence rate. In this case,

$$\varrho(\mathbf{H}_1) \approx \frac{\eta}{\eta + \alpha} = \varrho(\mathbf{H}_3) , \quad (\text{B.23})$$

which means that the SB method is no better than the two-split ADMM algorithm (B.2) with $\rho = \eta/\alpha$ when η is over-estimated. However, in practice, the SB method appears to converge a little bit faster because most frequency components in the SB method have convergence rate less than $\varrho(\mathbf{H}_1) \approx \varrho(\mathbf{H}_3)$. On the other hand, when $\eta < \alpha$, $\delta_{\min} \approx 0$ determines the convergence rate. In this case,

$$\varrho(\mathbf{H}_1) \approx \frac{\alpha}{\eta + \alpha} > \frac{\eta}{\eta + \alpha} = \varrho(\mathbf{H}_3) , \quad (\text{B.24})$$

which means that the SB method is slower than the two-split ADMM algorithm (B.2) with $\rho = \eta/\alpha$ when η is under-estimated. In sum, $\varrho(\mathbf{H}_1) \gtrsim \varrho(\mathbf{H}_3)$ for any η . That is, the two-split ADMM algorithm (B.2) with $\rho = \eta/\alpha$ is less sensitive to the choice of η due to the additional split and converges faster than the SB method especially for small η in most cases.

This analysis of the two-split ADMM algorithm might seem to be useless because we assume that we can solve the inner least-squares problem exactly and efficiently in our analysis, while the minimization problem (B.1) itself is a least-squares problem. In fact, if we initialize \mathbf{d} and \mathbf{e} in (B.2) properly as mentioned before, the two-split ADMM algorithm should solve the minimization problem in one iteration if we set $(\rho, \eta) = (1, \alpha)$, which happens to be the optimal SB method, as in (B.5). This does not contradict the non-zero (1/2) asymptotic convergence rate we showed in (B.12), (B.18), and (B.21) because the \mathbf{x} -update just solves the original minimization problem fortuitously. The other split variables still follow the asymptotic convergence rate we derived before. The goal of this analysis was to show that ADMM algorithms can sometimes converge faster than the SB method, and the simple analysis might give some intuition about the parameter tuning for practical problems.

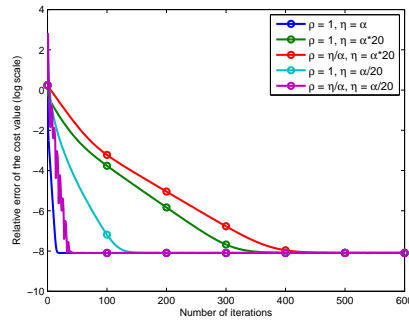
B.3 Numerical experiments

In this section, we verify the convergence rate result and parameter selection discussed in the previous section using an image restoration problem with a quadratic

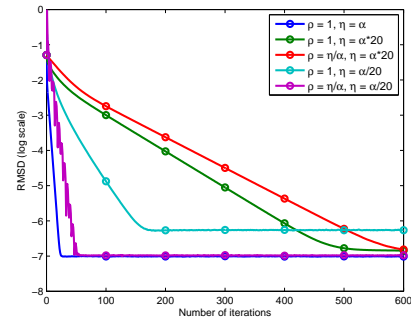


Figure B.1: An image restoration problem instance: the true image (left), the noisy blurred image (middle), and the converged reference reconstruction (right).

regularizer. Figure B.1 shows an image restoration problem instance: the true image (left), the noisy blurred image (middle), and the converged reference reconstruction (right). We use a quadratic roughness penalty as the regularizer where the regularization parameter α is chosen to be 2^{-4} for better noise-resolution tradeoff. Note that since a masked finite difference matrix (in horizontal and vertical directions) is used, we cannot solve the \mathbf{x} -update in (B.2) efficiently using FFT. Instead, we solve it using PCG with an appropriate circulant preconditioner for three iterations. The inexact updates might affect the convergence rate but not very significantly thanks to the circulant preconditioner. Figure B.2 shows the convergence rate curves (the relative error of cost value and RMS difference) of the two-split ADMM algorithm (B.2) with different parameter settings. As can be seen in Figure B.2, all reconstructed images with different parameter settings converge to the solution with minimum cost value (up to the machine epsilon of the single-precision floating-point arithmetic). When $(\alpha, \eta) = (1, \alpha)$, the two-split ADMM algorithm, i.e., the optimal SB method, achieves the fastest convergence rate with no ripple. As mentioned before, with a proper initialization, the two-split ADMM algorithm with this parameter setting should converge immediately; the non-zero convergence rate comes from the inexact updates. When η is over-estimated ($\eta = \alpha \times 20$), the SB method ($\rho = 1$) and the two-split ADMM algorithm with $\rho = \eta/\alpha = 20$ exhibit similar slow convergence rate. When η is under-estimated ($\eta = \alpha/20$), the SB method ($\rho = 1$) is much slower than the two-split ADMM algorithm with $\rho = \eta/\alpha = 1/20$. One might expect the ADMM algorithm with these parameters to converge with the same asymptotic convergence rate as the fastest two-split ADMM algorithm because η is very small, but in fact it suffers from strong overshoots and oscillation due to the large step size as mentioned in Section B.1.3.



(a)



(b)

Figure B.2: The convergence rate curves with different parameter settings: (a) the relative error of cost value and (b) the RMS difference between the reconstructed image and the reference reconstruction as a function of the number of iterations.

APPENDIX C

Convergence analyses of the inexact linearized AL method

C.1 Introduction

Consider a general composite convex optimization problem:

$$\hat{\mathbf{x}} \in \arg \min_{\mathbf{x}} \{g(\mathbf{Ax}) + h(\mathbf{x})\} \quad (\text{C.1})$$

and its equivalent constrained minimization problem:

$$(\hat{\mathbf{x}}, \hat{\mathbf{u}}) \in \arg \min_{\mathbf{x}, \mathbf{u}} \{g(\mathbf{u}) + h(\mathbf{x})\} \text{ s.t. } \mathbf{u} = \mathbf{Ax}, \quad (\text{C.2})$$

where both g and h are closed and proper convex functions. The inexact linearized AL methods that solve (C.2) are as follows:

$$\begin{cases} \left\| \mathbf{x}^{(k+1)} - \arg \min_{\mathbf{x}} \phi_k(\mathbf{x}) \right\| \leq \delta_k \\ \mathbf{u}^{(k+1)} \in \arg \min_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{\rho}{2} \left\| \mathbf{Ax}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)} \right\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{Ax}^{(k+1)} + \mathbf{u}^{(k+1)}, \end{cases} \quad (\text{C.3})$$

and

$$\begin{cases} \left| \phi_k(\mathbf{x}^{(k+1)}) - \min_{\mathbf{x}} \phi_k(\mathbf{x}) \right| \leq \varepsilon_k \\ \mathbf{u}^{(k+1)} \in \arg \min_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{\rho}{2} \left\| \mathbf{Ax}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)} \right\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{Ax}^{(k+1)} + \mathbf{u}^{(k+1)}, \end{cases} \quad (\text{C.4})$$

This appendix is based on the supplementary material of [38].

where

$$\phi_k(\mathbf{x}) \triangleq h(\mathbf{x}) + \check{\theta}_k(\mathbf{x}; \mathbf{x}^{(k)}), \quad (\text{C.5})$$

and

$$\check{\theta}_k(\mathbf{x}; \mathbf{x}^{(k)}) \triangleq \theta_k(\mathbf{x}^{(k)}) + \langle \nabla \theta_k(\mathbf{x}^{(k)}), \mathbf{x} - \mathbf{x}^{(k)} \rangle + \frac{\rho L}{2} \|\mathbf{x} - \mathbf{x}^{(k)}\|_2^2 \quad (\text{C.6})$$

is the separable quadratic surrogate (SQS) function of

$$\theta_k(\mathbf{x}) \triangleq \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}\|_2^2 \quad (\text{C.7})$$

with $L > \|\mathbf{A}\|_2^2 = \lambda_{\max}(\mathbf{A}'\mathbf{A})$, $\{\delta_k\}_{k=0}^{\infty}$ and $\{\varepsilon_k\}_{k=0}^{\infty}$ are two non-negative sequences, \mathbf{d} is the scaled Lagrange multiplier of the split variable \mathbf{u} , and $\rho > 0$ is the corresponding AL penalty parameter. Furthermore, in [38], we also showed that the inexact linearized AL method is equivalent to the inexact version of the Chambolle-Pock first-order primal-dual algorithm (CPPDA) [50]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \text{prox}_{\sigma h}(\mathbf{x}^{(k)} - \sigma \mathbf{A}'\bar{\mathbf{z}}^{(k)}) \\ \mathbf{z}^{(k+1)} \in \text{prox}_{\tau g^*}(\mathbf{z}^{(k)} + \tau \mathbf{A}\mathbf{x}^{(k+1)}) \\ \bar{\mathbf{z}}^{(k+1)} = \mathbf{z}^{(k+1)} + (\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)}) \end{cases} \quad (\text{C.8})$$

that solves the minimax problem:

$$(\hat{\mathbf{z}}, \hat{\mathbf{x}}) \in \arg \min_{\mathbf{z}} \max_{\mathbf{x}} \left\{ \Omega(\mathbf{z}, \mathbf{x}) \triangleq \langle -\mathbf{A}'\mathbf{z}, \mathbf{x} \rangle + g^*(\mathbf{z}) - h(\mathbf{x}) \right\} \quad (\text{C.9})$$

with $\mathbf{z} = -\tau \mathbf{d}$, $\sigma = \rho^{-1}t$, $\tau = \rho$, and $t \triangleq 1/L$, where prox_f denotes the proximal mapping of f defined as:

$$\text{prox}_f(\mathbf{z}) \triangleq \arg \min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 \right\}, \quad (\text{C.10})$$

and f^* denotes the convex conjugate of a function f . Note that $g^{**} = g$ and $h^{**} = h$ since both g and h are closed, proper, and convex.

C.2 Proof of Theorem 1

Theorem C.1. *Consider a constrained composite convex optimization problem (C.2) where both g and h are closed and proper convex functions. Let $\rho > 0$ and $\{\delta_k\}_{k=0}^{\infty}$*

denote a non-negative sequence such that

$$\sum_{k=0}^{\infty} \delta_k < \infty. \quad (\text{C.11})$$

If (C.2) has a solution $(\hat{\mathbf{x}}, \hat{\mathbf{u}})$, then the sequence of updates $\{(\mathbf{x}^{(k)}, \mathbf{u}^{(k)})\}_{k=0}^{\infty}$ generated by the inexact linearized AL method (C.3) converges to $(\hat{\mathbf{x}}, \hat{\mathbf{u}})$; otherwise, at least one of the sequences $\{(\mathbf{x}^{(k)}, \mathbf{u}^{(k)})\}_{k=0}^{\infty}$ or $\{\mathbf{d}^{(k)}\}_{k=0}^{\infty}$ diverges.

Proof. To prove this theorem, we first consider the exact linearized AL method:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg \min_{\mathbf{x}} \left\{ h(\mathbf{x}) + \check{\theta}_k(\mathbf{x}; \mathbf{x}^{(k)}) \right\} \\ \mathbf{u}^{(k+1)} \in \arg \min_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)}. \end{cases} \quad (\text{C.12})$$

Note that

$$\begin{aligned} & \check{\theta}_k(\mathbf{x}; \mathbf{x}^{(k)}) \\ &= \theta_k(\mathbf{x}^{(k)}) + \langle \nabla \theta_k(\mathbf{x}^{(k)}), \mathbf{x} - \mathbf{x}^{(k)} \rangle + \frac{\rho L}{2} \|\mathbf{x} - \mathbf{x}^{(k)}\|_2^2 \\ &= \theta_k(\mathbf{x}^{(k)}) + \langle \nabla \theta_k(\mathbf{x}^{(k)}), \mathbf{x} - \mathbf{x}^{(k)} \rangle + \frac{\rho}{2} \|\mathbf{x} - \mathbf{x}^{(k)}\|_{\mathbf{A}'\mathbf{A}}^2 + \frac{\rho}{2} \|\mathbf{x} - \mathbf{x}^{(k)}\|_{\mathbf{LI} - \mathbf{A}'\mathbf{A}}^2 \\ &= \theta_k(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{x} - \mathbf{x}^{(k)}\|_{\mathbf{G}}^2, \end{aligned} \quad (\text{C.13})$$

where $\mathbf{G} \triangleq \mathbf{LI} - \mathbf{A}'\mathbf{A} \succ 0$. Therefore, the exact linearized AL method can also be written as

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg \min_{\mathbf{x}} \left\{ h(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}\|_2^2 + \frac{\rho}{2} \|\mathbf{x} - \mathbf{x}^{(k)}\|_{\mathbf{G}}^2 \right\} \\ \mathbf{u}^{(k+1)} \in \arg \min_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)}. \end{cases} \quad (\text{C.14})$$

Now, consider another constrained minimization problem that is also equivalent to (C.1) but uses two split variables:

$$(\hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{v}}) \in \arg \min_{\mathbf{x}, \mathbf{u}, \mathbf{v}} \{g(\mathbf{u}) + h(\mathbf{x})\} \text{ s.t. } \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A} \\ \mathbf{G}^{1/2} \end{bmatrix}}_{\mathbf{S}} \mathbf{x}. \quad (\text{C.15})$$

The corresponding augmented Lagrangian and ADMM iterates [45] are

$$\begin{aligned} \mathcal{L}_A(\mathbf{x}, \mathbf{u}, \mathbf{d}, \mathbf{v}, \mathbf{e}; \rho, \eta) \\ \triangleq g(\mathbf{u}) + h(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \mathbf{u} - \mathbf{d}\|_2^2 + \frac{\eta}{2} \|\mathbf{G}^{1/2}\mathbf{x} - \mathbf{v} - \mathbf{e}\|_2^2 \end{aligned} \quad (\text{C.16})$$

and

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg \min_{\mathbf{x}} \left\{ h(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}\|_2^2 + \frac{\eta}{2} \|\mathbf{G}^{1/2}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)}\|_2^2 \right\} \\ \mathbf{u}^{(k+1)} \in \arg \min_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)} \\ \mathbf{v}^{(k+1)} = \mathbf{G}^{1/2}\mathbf{x}^{(k+1)} - \mathbf{e}^{(k)} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{G}^{1/2}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)}, \end{cases} \quad (\text{C.17})$$

where \mathbf{e} is the scaled Lagrange multiplier of the split variable \mathbf{v} , and $\eta > 0$ is the corresponding AL penalty parameter. Note that since \mathbf{G} is positive definite, \mathbf{S} defined in (C.15) has full column rank. Hence, the ADMM iterates (C.17) are convergent [44, Theorem 8]. Solving the last two iterates in (C.17) yields identities

$$\begin{cases} \mathbf{v}^{(k+1)} = \mathbf{G}^{1/2}\mathbf{x}^{(k+1)} \\ \mathbf{e}^{(k+1)} = \mathbf{0} \end{cases} \quad (\text{C.18})$$

if we initialize \mathbf{e} as $\mathbf{e}^{(0)} = \mathbf{0}$. Substituting (C.18) into (C.17), we have the equivalent ADMM iterates:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg \min_{\mathbf{x}} \left\{ h(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}\|_2^2 + \frac{\eta}{2} \|\mathbf{G}^{1/2}\mathbf{x} - \mathbf{G}^{1/2}\mathbf{x}^{(k)}\|_2^2 \right\} \\ \mathbf{u}^{(k+1)} \in \arg \min_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)}. \end{cases} \quad (\text{C.19})$$

When $\eta = \rho$, the equivalent ADMM iterates (C.19) reduce to (C.14). Therefore, the linearized AL method is a convergent ADMM! Finally, by using [44, Theorem 8], the linearized AL method is convergent if the error of \mathbf{x} -update is summable. That is, the inexact linearized AL method (C.3) is convergent if the non-negative sequence $\{\delta_k\}_{k=0}^{\infty}$ satisfies $\sum_{k=0}^{\infty} \delta_k < \infty$. \blacksquare

C.3 Proof of Theorem 2

Theorem C.2. *Consider a minimax problem (C.9) where both g and h are closed and proper convex functions. Suppose it has a saddle-point $(\hat{\mathbf{z}}, \hat{\mathbf{x}})$. Note that since the minimization problem (C.1) happens to be the dual problem of (C.9), $\hat{\mathbf{x}}$ is also a solution of (C.1). Let $\rho > 0$ and $\{\varepsilon_k\}_{k=0}^{\infty}$ denote a non-negative sequence such that*

$$\sum_{k=0}^{\infty} \sqrt{\varepsilon_k} < \infty. \quad (\text{C.20})$$

Then, the sequence of updates $\{(-\rho \mathbf{d}^{(k)}, \mathbf{x}^{(k)})\}_{k=0}^{\infty}$ generated by the inexact linearized AL method (C.4) is a bounded sequence that converges to $(\hat{\mathbf{z}}, \hat{\mathbf{x}})$, and the primal-dual gap of $(\mathbf{z}_k, \mathbf{x}_k)$ has the following bound:

$$\Omega(\mathbf{z}_k, \hat{\mathbf{x}}) - \Omega(\hat{\mathbf{z}}, \mathbf{x}_k) \leq \frac{(C + 2A_k + \sqrt{B_k})^2}{k}, \quad (\text{C.21})$$

where $\mathbf{z}_k \triangleq \frac{1}{k} \sum_{j=1}^k (-\rho \mathbf{d}^{(j)})$, $\mathbf{x}_k \triangleq \frac{1}{k} \sum_{j=1}^k \mathbf{x}^{(j)}$,

$$C \triangleq \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\rho^{-1}t}} + \frac{\|(-\rho \mathbf{d}^{(0)}) - \hat{\mathbf{z}}\|_2}{\sqrt{2\rho}}, \quad (\text{C.22})$$

$$A_k \triangleq \sum_{j=1}^k \sqrt{\frac{\varepsilon_{j-1}}{(1 - t \|\mathbf{A}\|_2^2) \rho^{-1}t}}, \quad (\text{C.23})$$

and

$$B_k \triangleq \sum_{j=1}^k \varepsilon_{j-1}. \quad (\text{C.24})$$

Proof. As mentioned before, the inexact linearized AL method is the inexact version of CPPDA with a specific choice of σ and τ and a substitution $\mathbf{z} = -\tau \mathbf{d}$. Here, we just prove the convergence of the inexact CPPDA by extending the analysis in [50], and the inexact linearized AL method is simply a special case of the inexact CPPDA. However, since the proximal mapping in the \mathbf{x} -update of the inexact CPPDA is solved inexactly, the existing analysis is not applicable. To solve this problem, we adopt the error analysis technique developed in [30]. We first define the inexact proximal mapping

$$\mathbf{u} \stackrel{\varepsilon}{\approx} \text{prox}_{\phi}(\mathbf{v}) \quad (\text{C.25})$$

to be the mapping that satisfies

$$\phi(\mathbf{u}) + \frac{1}{2} \|\mathbf{u} - \mathbf{v}\|_2^2 \leq \varepsilon + \min_{\bar{\mathbf{u}}} \left\{ \phi(\bar{\mathbf{u}}) + \frac{1}{2} \|\bar{\mathbf{u}} - \mathbf{v}\|_2^2 \right\}. \quad (\text{C.26})$$

Therefore, the inexact CPPDA is defined as

$$\begin{cases} \mathbf{x}^{(k+1)} \stackrel{\varepsilon_k}{\approx} \text{prox}_{\sigma h}(\mathbf{x}^{(k)} - \sigma \mathbf{A}' \bar{\mathbf{z}}^{(k)}) \\ \mathbf{z}^{(k+1)} \in \text{prox}_{\tau g^*}(\mathbf{z}^{(k)} + \tau \mathbf{A} \mathbf{x}^{(k+1)}) \\ \bar{\mathbf{z}}^{(k+1)} = \mathbf{z}^{(k+1)} + (\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)}) \end{cases} \quad (\text{C.27})$$

with $\sigma\tau \|\mathbf{A}\|_2^2 < 1$. One can verify that with $\mathbf{z} = -\tau \mathbf{d}$, $\sigma = \rho^{-1}t$, and $\tau = \rho$, the inexact CPPDA (C.27) is equivalent to the inexact linearized AL method (C.4). Schmidt *et al.* showed that

$$\mathbf{u} \stackrel{\varepsilon}{\approx} \text{prox}_{\phi}(\mathbf{v}) \Leftrightarrow \mathbf{v} - \mathbf{u} - \mathbf{f} \in \partial_{\varepsilon} \phi(\mathbf{u}) \quad (\text{C.28})$$

with $\|\mathbf{f}\|_2 \leq \sqrt{2\varepsilon}$, and for any $\mathbf{s} \in \partial_{\varepsilon} \phi(\mathbf{u})$,

$$\phi(\mathbf{w}) \geq \phi(\mathbf{u}) + \mathbf{s}'(\mathbf{w} - \mathbf{u}) - \varepsilon \quad (\text{C.29})$$

for all \mathbf{w} , where $\partial_{\varepsilon} \phi(\mathbf{u})$ denotes the ε -subdifferential of ϕ at \mathbf{u} [30, Lemma 2]. When $\varepsilon = 0$, (C.28) and (C.29) reduce to the standard optimality condition of a proximal mapping and the definition of subgradient, respectively. At the j th iteration, $j = 0, \dots, k-1$, the updates generated by the inexact CPPDA (C.27) satisfy

$$\begin{cases} (\mathbf{x}^{(j)} - \sigma \mathbf{A}' \bar{\mathbf{z}}^{(j)}) - \mathbf{x}^{(j+1)} - \mathbf{f}^{(j)} \in \partial_{\varepsilon_j}(\sigma h)(\mathbf{x}^{(j+1)}) \\ (\mathbf{z}^{(j)} + \tau \mathbf{A} \mathbf{x}^{(j+1)}) - \mathbf{z}^{(j+1)} \in \partial(\tau g^*)(\mathbf{z}^{(j+1)}). \end{cases} \quad (\text{C.30})$$

In other words,

$$\frac{\mathbf{x}^{(j)} - \mathbf{x}^{(j+1)}}{\sigma} - \mathbf{A}' \bar{\mathbf{z}}^{(j)} - \frac{\mathbf{f}^{(j)}}{\sigma} \in \partial_{\varepsilon_j} h(\mathbf{x}^{(j+1)}) \quad (\text{C.31})$$

and

$$\frac{\mathbf{z}^{(j)} - \mathbf{z}^{(j+1)}}{\tau} + \mathbf{A} \mathbf{x}^{(j+1)} \in \partial g^*(\mathbf{z}^{(j+1)}), \quad (\text{C.32})$$

where $\|\mathbf{f}^{(j)}\|_2 \leq \sqrt{2\varepsilon_j}$. From (C.31), we have

$$\begin{aligned}
h(\mathbf{x}) &\geq h(\mathbf{x}^{(j+1)}) + \langle \partial_{\varepsilon_j} h(\mathbf{x}^{(j+1)}), \mathbf{x} - \mathbf{x}^{(j+1)} \rangle - \varepsilon_j \\
&= h(\mathbf{x}^{(j+1)}) + \left\langle \frac{\mathbf{x}^{(j)} - \mathbf{x}^{(j+1)}}{\sigma}, \mathbf{x} - \mathbf{x}^{(j+1)} \right\rangle - \langle \mathbf{A}' \bar{\mathbf{z}}^{(j)}, \mathbf{x} - \mathbf{x}^{(j+1)} \rangle \\
&\quad - \left\langle \frac{\mathbf{f}^{(j)}}{\sigma}, \mathbf{x} - \mathbf{x}^{(j+1)} \right\rangle - \varepsilon_j \\
&= h(\mathbf{x}^{(j+1)}) + \frac{1}{2\sigma} (\|\mathbf{x}^{(j+1)} - \mathbf{x}\|_2^2 + \|\mathbf{x}^{(j+1)} - \mathbf{x}^{(j)}\|_2^2 - \|\mathbf{x}^{(j)} - \mathbf{x}\|_2^2) \\
&\quad + \langle -\mathbf{A}'(\bar{\mathbf{z}}^{(j)} - \mathbf{z}^{(j+1)}), \mathbf{x} - \mathbf{x}^{(j+1)} \rangle + \langle -\mathbf{A}'\mathbf{z}^{(j+1)}, \mathbf{x} - \mathbf{x}^{(j+1)} \rangle \\
&\quad - \left\langle \frac{\mathbf{f}^{(j)}}{\sigma}, \mathbf{x} - \mathbf{x}^{(j+1)} \right\rangle - \varepsilon_j \\
&\geq h(\mathbf{x}^{(j+1)}) + \frac{1}{2\sigma} (\|\mathbf{x}^{(j+1)} - \mathbf{x}\|_2^2 + \|\mathbf{x}^{(j+1)} - \mathbf{x}^{(j)}\|_2^2 - \|\mathbf{x}^{(j)} - \mathbf{x}\|_2^2) \\
&\quad + \langle -\mathbf{A}'(\bar{\mathbf{z}}^{(j)} - \mathbf{z}^{(j+1)}), \mathbf{x} - \mathbf{x}^{(j+1)} \rangle + \langle -\mathbf{A}'\mathbf{z}^{(j+1)}, \mathbf{x} - \mathbf{x}^{(j+1)} \rangle \\
&\quad - \frac{1}{\sigma} \|\mathbf{f}^{(j)}\|_2 \|\mathbf{x} - \mathbf{x}^{(j+1)}\|_2 - \varepsilon_j \\
&\geq h(\mathbf{x}^{(j+1)}) + \frac{1}{2\sigma} (\|\mathbf{x}^{(j+1)} - \mathbf{x}\|_2^2 + \|\mathbf{x}^{(j+1)} - \mathbf{x}^{(j)}\|_2^2 - \|\mathbf{x}^{(j)} - \mathbf{x}\|_2^2) \\
&\quad + \langle -\mathbf{A}'(\bar{\mathbf{z}}^{(j)} - \mathbf{z}^{(j+1)}), \mathbf{x} - \mathbf{x}^{(j+1)} \rangle + \langle -\mathbf{A}'\mathbf{z}^{(j+1)}, \mathbf{x} - \mathbf{x}^{(j+1)} \rangle \\
&\quad - \frac{\sqrt{2\varepsilon_j}}{\sigma} \|\mathbf{x} - \mathbf{x}^{(j+1)}\|_2 - \varepsilon_j \tag{C.33}
\end{aligned}$$

for any $\mathbf{x} \in \text{Dom } h$. From (C.32), we have

$$\begin{aligned}
g^*(\mathbf{z}) &\geq g^*(\mathbf{z}^{(j+1)}) + \langle \partial g^*(\mathbf{z}^{(j+1)}), \mathbf{z} - \mathbf{z}^{(j+1)} \rangle \\
&= g^*(\mathbf{z}^{(j+1)}) + \left\langle \frac{\mathbf{z}^{(j)} - \mathbf{z}^{(j+1)}}{\tau}, \mathbf{z} - \mathbf{z}^{(j+1)} \right\rangle + \langle \mathbf{A}\mathbf{x}^{(j+1)}, \mathbf{z} - \mathbf{z}^{(j+1)} \rangle \\
&= g^*(\mathbf{z}^{(j+1)}) + \frac{1}{2\tau} (\|\mathbf{z}^{(j+1)} - \mathbf{z}\|_2^2 + \|\mathbf{z}^{(j+1)} - \mathbf{z}^{(j)}\|_2^2 - \|\mathbf{z}^{(j)} - \mathbf{z}\|_2^2) \\
&\quad - \langle -\mathbf{A}'(\mathbf{z} - \mathbf{z}^{(j+1)}), \mathbf{x}^{(j+1)} \rangle \tag{C.34}
\end{aligned}$$

for any $\mathbf{z} \in \text{Dom } g^*$. Summing (C.33) and (C.34), it follows:

$$\begin{aligned}
\frac{\|\mathbf{x}^{(j)} - \mathbf{x}\|_2^2}{2\sigma} + \frac{\|\mathbf{z}^{(j)} - \mathbf{z}\|_2^2}{2\tau} &\geq (\Omega(\mathbf{z}^{(j+1)}, \mathbf{x}) - \Omega(\mathbf{z}, \mathbf{x}^{(j+1)})) \\
&\quad + \frac{\|\mathbf{x}^{(j+1)} - \mathbf{x}\|_2^2}{2\sigma} + \frac{\|\mathbf{z}^{(j+1)} - \mathbf{z}\|_2^2}{2\tau} + \frac{\|\mathbf{x}^{(j+1)} - \mathbf{x}^{(j)}\|_2^2}{2\sigma} + \frac{\|\mathbf{z}^{(j+1)} - \mathbf{z}^{(j)}\|_2^2}{2\tau} \\
&\quad + \langle -\mathbf{A}'(\bar{\mathbf{z}}^{(j)} - \mathbf{z}^{(j+1)}), \mathbf{x} - \mathbf{x}^{(j+1)} \rangle - \frac{\sqrt{2\varepsilon_j}}{\sigma} \|\mathbf{x} - \mathbf{x}^{(j+1)}\|_2 - \varepsilon_j. \tag{C.35}
\end{aligned}$$

Furthermore,

$$\begin{aligned}
& \langle -\mathbf{A}'(\bar{\mathbf{z}}^{(j)} - \mathbf{z}^{(j+1)}), \mathbf{x} - \mathbf{x}^{(j+1)} \rangle \\
&= \langle -\mathbf{A}'(\mathbf{z}^{(j+1)} - 2\mathbf{z}^{(j)} + \mathbf{z}^{(j-1)}), \mathbf{x}^{(j+1)} - \mathbf{x} \rangle \\
&= \langle -\mathbf{A}'(\mathbf{z}^{(j+1)} - \mathbf{z}^{(j)}), \mathbf{x}^{(j+1)} - \mathbf{x} \rangle - \langle -\mathbf{A}'(\mathbf{z}^{(j)} - \mathbf{z}^{(j-1)}), \mathbf{x}^{(j)} - \mathbf{x} \rangle \\
&\quad - \langle -\mathbf{A}'(\mathbf{z}^{(j)} - \mathbf{z}^{(j-1)}), \mathbf{x}^{(j+1)} - \mathbf{x}^{(j)} \rangle \\
&\geq \langle -\mathbf{A}'(\mathbf{z}^{(j+1)} - \mathbf{z}^{(j)}), \mathbf{x}^{(j+1)} - \mathbf{x} \rangle - \langle -\mathbf{A}'(\mathbf{z}^{(j)} - \mathbf{z}^{(j-1)}), \mathbf{x}^{(j)} - \mathbf{x} \rangle \\
&\quad - \|\mathbf{A}\|_2 \|\mathbf{z}^{(j)} - \mathbf{z}^{(j-1)}\|_2 \|\mathbf{x}^{(j+1)} - \mathbf{x}^{(j)}\|_2 \\
&\geq \langle -\mathbf{A}'(\mathbf{z}^{(j+1)} - \mathbf{z}^{(j)}), \mathbf{x}^{(j+1)} - \mathbf{x} \rangle - \langle -\mathbf{A}'(\mathbf{z}^{(j)} - \mathbf{z}^{(j-1)}), \mathbf{x}^{(j)} - \mathbf{x} \rangle \\
&\quad - \|\mathbf{A}\|_2 \left(\frac{\sqrt{\sigma/\tau}}{2} \|\mathbf{z}^{(j)} - \mathbf{z}^{(j-1)}\|_2^2 + \frac{1}{2\sqrt{\sigma/\tau}} \|\mathbf{x}^{(j+1)} - \mathbf{x}^{(j)}\|_2^2 \right) \tag{C.36}
\end{aligned}$$

$$\begin{aligned}
&\geq \langle -\mathbf{A}'(\mathbf{z}^{(j+1)} - \mathbf{z}^{(j)}), \mathbf{x}^{(j+1)} - \mathbf{x} \rangle - \langle -\mathbf{A}'(\mathbf{z}^{(j)} - \mathbf{z}^{(j-1)}), \mathbf{x}^{(j)} - \mathbf{x} \rangle \\
&\quad - \sqrt{\sigma\tau} \|\mathbf{A}\|_2 \left(\frac{\|\mathbf{z}^{(j)} - \mathbf{z}^{(j-1)}\|_2^2}{2\tau} + \frac{\|\mathbf{x}^{(j+1)} - \mathbf{x}^{(j)}\|_2^2}{2\sigma} \right), \tag{C.37}
\end{aligned}$$

where (C.36) is due to Young's inequality. Plugging (C.37) into (C.35), it follows that for any (\mathbf{z}, \mathbf{x}) ,

$$\begin{aligned}
& \frac{\|\mathbf{x}^{(j)} - \mathbf{x}\|_2^2}{2\sigma} + \frac{\|\mathbf{z}^{(j)} - \mathbf{z}\|_2^2}{2\tau} \geq (\Omega(\mathbf{z}^{(j+1)}, \mathbf{x}) - \Omega(\mathbf{z}, \mathbf{x}^{(j+1)})) \\
&\quad + \frac{\|\mathbf{x}^{(j+1)} - \mathbf{x}\|_2^2}{2\sigma} + \frac{\|\mathbf{z}^{(j+1)} - \mathbf{z}\|_2^2}{2\tau} + (1 - \sqrt{\sigma\tau} \|\mathbf{A}\|_2) \frac{\|\mathbf{x}^{(j+1)} - \mathbf{x}^{(j)}\|_2^2}{2\sigma} \\
&\quad + \frac{\|\mathbf{z}^{(j+1)} - \mathbf{z}^{(j)}\|_2^2}{2\tau} - \sqrt{\sigma\tau} \|\mathbf{A}\|_2 \frac{\|\mathbf{z}^{(j)} - \mathbf{z}^{(j-1)}\|_2^2}{2\tau} \\
&\quad + \langle -\mathbf{A}'(\mathbf{z}^{(j+1)} - \mathbf{z}^{(j)}), \mathbf{x}^{(j+1)} - \mathbf{x} \rangle - \langle -\mathbf{A}'(\mathbf{z}^{(j)} - \mathbf{z}^{(j-1)}), \mathbf{x}^{(j)} - \mathbf{x} \rangle \\
&\quad - \frac{\sqrt{2\varepsilon_j}}{\sigma} \|\mathbf{x} - \mathbf{x}^{(j+1)}\|_2 - \varepsilon_j. \tag{C.38}
\end{aligned}$$

Suppose $\mathbf{z}^{(-1)} = \mathbf{z}^{(0)}$, i.e., $\bar{\mathbf{z}}^{(0)} = \mathbf{z}^{(0)}$. Summing up (C.38) from $j = 0, \dots, k-1$ and using

$$\langle -\mathbf{A}'(\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}), \mathbf{x}^{(k)} - \mathbf{x} \rangle \leq \frac{\|\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\|_2^2}{2\tau} + \sigma\tau \|\mathbf{A}\|_2^2 \frac{\|\mathbf{x}^{(k)} - \mathbf{x}\|_2^2}{2\sigma} \tag{C.39}$$

as before, we have

$$\begin{aligned}
& \sum_{j=1}^k (\Omega(\mathbf{z}^{(j)}, \mathbf{x}) - \Omega(\mathbf{z}, \mathbf{x}^{(j)})) + (1 - \sigma\tau \|\mathbf{A}\|_2^2) \frac{\|\mathbf{x}^{(k)} - \mathbf{x}\|_2^2}{2\sigma} + \frac{\|\mathbf{z}^{(k)} - \mathbf{z}\|_2^2}{2\tau} \\
& + (1 - \sqrt{\sigma\tau} \|\mathbf{A}\|_2) \sum_{j=1}^k \frac{\|\mathbf{x}^{(j)} - \mathbf{x}^{(j-1)}\|_2^2}{2\sigma} + (1 - \sqrt{\sigma\tau} \|\mathbf{A}\|_2) \sum_{j=1}^{k-1} \frac{\|\mathbf{z}^{(j)} - \mathbf{z}^{(j-1)}\|_2^2}{2\tau} \\
& \leq \frac{\|\mathbf{x}^{(0)} - \mathbf{x}\|_2^2}{2\sigma} + \frac{\|\mathbf{z}^{(0)} - \mathbf{z}\|_2^2}{2\tau} + \sum_{j=1}^k \varepsilon_{j-1} + \sum_{j=1}^k 2\sqrt{\frac{\varepsilon_{j-1}}{\sigma}} \frac{\|\mathbf{x}^{(j)} - \mathbf{x}\|_2}{\sqrt{2\sigma}}. \quad (\text{C.40})
\end{aligned}$$

Since $\sigma\tau \|\mathbf{A}\|_2^2 < 1$, we have $1 - \sigma\tau \|\mathbf{A}\|_2^2 > 0$ and $1 - \sqrt{\sigma\tau} \|\mathbf{A}\|_2 > 0$. If we choose $(\mathbf{z}, \mathbf{x}) = (\hat{\mathbf{z}}, \hat{\mathbf{x}})$, the first term on the left-hand side of (C.40) is the sum of k non-negative primal-dual gaps, and all terms in (C.40) are greater than or equal to zero. Let $D \triangleq 1 - \sigma\tau \|\mathbf{A}\|_2^2 > 0$. We have three inequalities:

$$\begin{aligned}
D \cdot \frac{\|\mathbf{x}^{(k)} - \hat{\mathbf{x}}\|_2^2}{2\sigma} \\
\leq \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2^2}{2\sigma} + \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2^2}{2\tau} + \sum_{j=1}^k \varepsilon_{j-1} + \sum_{j=1}^k 2\sqrt{\frac{\varepsilon_{j-1}}{\sigma}} \frac{\|\mathbf{x}^{(j)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}}, \quad (\text{C.41})
\end{aligned}$$

$$\begin{aligned}
D \cdot \left(\frac{\|\mathbf{x}^{(k)} - \hat{\mathbf{x}}\|_2^2}{2\sigma} + \frac{\|\mathbf{z}^{(k)} - \hat{\mathbf{z}}\|_2^2}{2\tau} \right) \\
\leq \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2^2}{2\sigma} + \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2^2}{2\tau} + \sum_{j=1}^k \varepsilon_{j-1} + \sum_{j=1}^k 2\sqrt{\frac{\varepsilon_{j-1}}{\sigma}} \frac{\|\mathbf{x}^{(j)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}}, \quad (\text{C.42})
\end{aligned}$$

and

$$\begin{aligned}
\sum_{j=1}^k (\Omega(\mathbf{z}^{(j)}, \hat{\mathbf{x}}) - \Omega(\hat{\mathbf{z}}, \mathbf{x}^{(j)})) \\
\leq \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2^2}{2\sigma} + \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2^2}{2\tau} + \sum_{j=1}^k \varepsilon_{j-1} + \sum_{j=1}^k 2\sqrt{\frac{\varepsilon_{j-1}}{\sigma}} \frac{\|\mathbf{x}^{(j)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}}. \quad (\text{C.43})
\end{aligned}$$

All these inequality has a common right-hand-side. To continue the proof, we have

to bound $\|\mathbf{x}^{(j)} - \hat{\mathbf{x}}\|_2 / \sqrt{2\sigma}$ first. Dividing D from both sides of (C.41), we have

$$\begin{aligned} \left(\frac{\|\mathbf{x}^{(k)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}} \right)^2 &\leq \left(\frac{1}{D} \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2^2}{2\sigma} + \frac{1}{D} \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2^2}{2\tau} + \sum_{j=1}^k \frac{\varepsilon_{j-1}}{D} \right) \\ &\quad + \sum_{j=1}^k 2 \left(\frac{1}{D} \sqrt{\frac{\varepsilon_{j-1}}{\sigma}} \right) \frac{\|\mathbf{x}^{(j)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}}. \end{aligned} \quad (\text{C.44})$$

Let

$$S_k \triangleq \frac{1}{D} \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2^2}{2\sigma} + \frac{1}{D} \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2^2}{2\tau} + \sum_{j=1}^k \frac{\varepsilon_{j-1}}{D}, \quad (\text{C.45})$$

$$\lambda_j \triangleq 2 \left(\frac{1}{D} \sqrt{\frac{\varepsilon_{j-1}}{\sigma}} \right), \quad (\text{C.46})$$

and

$$u_j \triangleq \frac{\|\mathbf{x}^{(j)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}}. \quad (\text{C.47})$$

We have $u_k^2 \leq S_k + \sum_{j=1}^k \lambda_j u_j$ from (C.44) with $\{S_k\}_{k=0}^\infty$ an increasing sequence, $S_0 \geq u_0^2$ (note that $0 < D < 1$ because $0 < \sigma\tau \|\mathbf{A}\|_2^2 < 1$), and $\lambda_j \geq 0$ for all j . According to [30, Lemma 1], it follows that

$$\frac{\|\mathbf{x}^{(k)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}} \leq \tilde{A}_k + \left(\frac{1}{D} \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2^2}{2\sigma} + \frac{1}{D} \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2^2}{2\tau} + \tilde{B}_k + \tilde{A}_k^2 \right)^{1/2}, \quad (\text{C.48})$$

where

$$\tilde{A}_k \triangleq \sum_{j=1}^k \frac{1}{D} \sqrt{\frac{\varepsilon_{j-1}}{\sigma}}, \quad (\text{C.49})$$

and

$$\tilde{B}_k \triangleq \sum_{j=1}^k \frac{\varepsilon_{j-1}}{D}. \quad (\text{C.50})$$

Since \tilde{A}_j and \tilde{B}_j are increasing sequences of j , for $j \leq k$, we have

$$\begin{aligned}
\frac{\|\mathbf{x}^{(j)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}} &\leq \tilde{A}_j + \left(\frac{1}{D} \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2^2}{2\sigma} + \frac{1}{D} \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2^2}{2\tau} + \tilde{B}_j + \tilde{A}_j^2 \right)^{1/2} \\
&\leq \tilde{A}_k + \left(\frac{1}{D} \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2^2}{2\sigma} + \frac{1}{D} \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2^2}{2\tau} + \tilde{B}_k + \tilde{A}_k^2 \right)^{1/2} \\
&\leq \tilde{A}_k + \left(\frac{1}{\sqrt{D}} \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}} + \frac{1}{\sqrt{D}} \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2}{\sqrt{2\tau}} + \sqrt{\tilde{B}_k + \tilde{A}_k} \right). \quad (\text{C.51})
\end{aligned}$$

Now, we can bound the right-hand-side of (C.41), (C.42), and (C.43) as

$$\begin{aligned}
&\frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2^2}{2\sigma} + \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2^2}{2\tau} + \sum_{j=1}^k \varepsilon_{j-1} + \sum_{j=1}^k 2\sqrt{\frac{\varepsilon_{j-1}}{\sigma}} \frac{\|\mathbf{x}^{(j)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}} \\
&\leq \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2^2}{2\sigma} + \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2^2}{2\tau} \\
&\quad + \sum_{j=1}^k \varepsilon_{j-1} + \sum_{j=1}^k 2\sqrt{\frac{\varepsilon_{j-1}}{\sigma}} \left(2\tilde{A}_k + \frac{1}{\sqrt{D}} \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}} + \frac{1}{\sqrt{D}} \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2}{\sqrt{2\tau}} + \sqrt{\tilde{B}_k} \right) \\
&= \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2^2}{2\sigma} + \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2^2}{2\tau} \\
&\quad + \tilde{B}_k D + 2\tilde{A}_k D \left(2\tilde{A}_k + \frac{1}{\sqrt{D}} \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}} + \frac{1}{\sqrt{D}} \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2}{\sqrt{2\tau}} + \sqrt{\tilde{B}_k} \right) \\
&\leq \left(\frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}} + \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2}{\sqrt{2\tau}} + 2\tilde{A}_k \sqrt{D} + \sqrt{\tilde{B}_k D} \right)^2 \\
&= \left(\frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}} + \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2}{\sqrt{2\tau}} + 2A_k + \sqrt{B_k} \right)^2 \quad (\text{C.52})
\end{aligned}$$

$$\leq \left(\frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}} + \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2}{\sqrt{2\tau}} + 2A_\infty + \sqrt{B_\infty} \right)^2 \quad (\text{C.53})$$

if $\{\sqrt{\varepsilon_k}\}_{k=0}^\infty$ is absolutely summable (and therefore, $\{\varepsilon_k\}_{k=0}^\infty$ is also absolutely summable), where

$$A_k \triangleq \tilde{A}_k \sqrt{D} = \sum_{j=1}^k \sqrt{\frac{\varepsilon_{j-1}}{(1-\sigma\tau\|\mathbf{A}\|_2^2)^\sigma}}, \quad (\text{C.54})$$

and

$$B_k \triangleq \tilde{B}_k D = \sum_{j=1}^k \varepsilon_{j-1}. \quad (\text{C.55})$$

Hence, from (C.42), we have

$$\frac{\|\mathbf{x}^{(k)} - \hat{\mathbf{x}}\|_2^2}{2\sigma} + \frac{\|\mathbf{z}^{(k)} - \hat{\mathbf{z}}\|_2^2}{2\tau} \leq \frac{1}{D} \left(\frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}} + \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2}{\sqrt{2\tau}} + 2A_\infty + \sqrt{B_\infty} \right)^2 < \infty. \quad (\text{C.56})$$

This implies that the sequence of updates $\{(\mathbf{z}^{(k)}, \mathbf{x}^{(k)})\}_{k=0}^\infty$ generated by the inexact CPPDA (C.27) is a bounded sequence. Let

$$C \triangleq \frac{\|\mathbf{x}^{(0)} - \hat{\mathbf{x}}\|_2}{\sqrt{2\sigma}} + \frac{\|\mathbf{z}^{(0)} - \hat{\mathbf{z}}\|_2}{\sqrt{2\tau}}. \quad (\text{C.57})$$

From (C.43) and the convexity of h and g^* , we have

$$\begin{aligned} \Omega(\mathbf{z}_k, \hat{\mathbf{x}}) - \Omega(\hat{\mathbf{z}}, \mathbf{x}_k) &\leq \frac{1}{k} \sum_{j=1}^k (\Omega(\mathbf{z}^{(j)}, \hat{\mathbf{x}}) - \Omega(\hat{\mathbf{z}}, \mathbf{x}^{(j)})) \\ &\leq \frac{(C + 2A_k + \sqrt{B_k})^2}{k} \end{aligned} \quad (\text{C.58})$$

$$\leq \frac{(C + 2A_\infty + \sqrt{B_\infty})^2}{k}, \quad (\text{C.59})$$

where $\mathbf{z}_k \triangleq \frac{1}{k} \sum_{j=1}^k \mathbf{z}^{(j)}$, and $\mathbf{x}_k \triangleq \frac{1}{k} \sum_{j=1}^k \mathbf{x}^{(j)}$. That is, the primal-dual gap of $(\mathbf{z}_k, \mathbf{x}_k)$ converges to zero with rate $O(1/k)$. Following the procedure in [50, Section 3.1], we can further show that the sequence of updates $\{(\mathbf{z}^{(k)}, \mathbf{x}^{(k)})\}_{k=0}^\infty$ generated by the inexact CPPDA (C.27) converges to a saddle-point of (C.9) if the dimension of \mathbf{x} and \mathbf{z} is finite. \blacksquare

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] J. A. Fessler. Penalized weighted least-squares image reconstruction for positron emission tomography. *IEEE Trans. Med. Imag.*, 13(2):290–300, June 1994.
- [2] J. A. Fessler. Statistical image reconstruction methods for transmission tomography. In M. Sonka and J. Michael Fitzpatrick, editors, *Handbook of Medical Imaging, Volume 2. Medical Image Processing and Analysis*, pages 1–70. SPIE, Bellingham, 2000.
- [3] J-B. Thibault, K. Sauer, C. Bouman, and J. Hsieh. A three-dimensional statistical approach to improved image quality for multi-slice helical CT. *Med. Phys.*, 34(11):4526–44, November 2007.
- [4] B. De Man and J. A. Fessler. Statistical iterative reconstruction for X-ray computed tomography. In Ming Jiang Yair Censor and Ge Wang, editors, *Biomedical Mathematics: Promising Directions in Imaging, Therapy Planning and Inverse Problems*, pages 113–40. Medical Physics Publishing, Madison, WI, 2010. ISBN: 9781930524484.
- [5] Ionizing radiation exposure of the population of the United States. Technical Report 160, National Council on Radiation Protection and Measurements (NCRP), 2009.
- [6] J. de Mey. Implementing ultra-low dose CT with Veo at University Hospital, Brussels. *GE Healthcare CT Clarity*, pages 21–3, November 2011.
- [7] B. Vandeghinste, B. Goossens, J. D. Beenhouwer, A. Pizurica, W. Philips, S. Vandenberghe, and S. Staelens. Split-Bregman-based sparse-view CT reconstruction. In *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med.*, pages 431–4, 2011.
- [8] S. Ramani and J. A. Fessler. A splitting-based iterative algorithm for accelerated statistical X-ray CT reconstruction. *IEEE Trans. Med. Imag.*, 31(3):677–88, March 2012.
- [9] M. G. McGaffin, S. Ramani, and J. A. Fessler. Reduced memory augmented Lagrangian algorithm for 3D iterative X-ray CT image reconstruction. In *Proc. SPIE 8313 Medical Imaging 2012: Phys. Med. Im.*, page 831327, 2012.
- [10] H. Erdoğan and J. A. Fessler. Ordered subsets algorithms for transmission tomography. *Phys. Med. Biol.*, 44(11):2835–51, November 1999.
- [11] G. N. Hounsfield. Computerized transverse axial scanning (tomography): Part I. Description of system. *Brit. J. Radiology*, 46(552):1016–22, December 1973.
- [12] Y. Long, J. A. Fessler, and J. M. Balter. 3D forward and back-projection for X-ray CT using separable footprints. *IEEE Trans. Med. Imag.*, 29(11):1839–50, November 2010.
- [13] J. Radon. On the determination of functions from their integrals along certain manifold. *Berichte Sächs. Akad. Wiss. (Leipzig)*, 69:262–78, 1917. Über die Bestimmung von Funktionen durch ihre Intergralwerte Langs gewisser Mannigfaltigkeiten.

- [14] L. A. Feldkamp, L. C. Davis, and J. W. Kress. Practical cone beam algorithm. *J. Opt. Soc. Am. A*, 1(6):612–9, June 1984.
- [15] J. A. Fessler and W. L. Rogers. Spatial resolution properties of penalized-likelihood image reconstruction methods: Space-invariant tomographs. *IEEE Trans. Im. Proc.*, 5(9):1346–58, September 1996.
- [16] J. A. Fessler. Hybrid Poisson/polynomial objective functions for tomographic image reconstruction from transmission scans. *IEEE Trans. Im. Proc.*, 4(10):1439–50, October 1995.
- [17] D. R. Hunter and K. Lange. A tutorial on MM algorithms. *American Statistician*, 58(1):30–7, February 2004.
- [18] P. J. Huber. *Robust statistics*. Wiley, New York, 1981.
- [19] S. Ahn and J. A. Fessler. Globally convergent image reconstruction for emission tomography using relaxed ordered subsets algorithms. *IEEE Trans. Med. Imag.*, 22(5):613–26, May 2003.
- [20] S. Ahn, J. A. Fessler, D. Blatt, and A. O. Hero. Convergent incremental optimization transfer algorithms: Application to tomography. *IEEE Trans. Med. Imag.*, 25(3):283–96, March 2006.
- [21] D. P. Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey, 2010. August 2010 (revised December 2010) Report LIDS 2848.
- [22] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Stat.*, 22(3):400–7, September 1951.
- [23] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Dokl. Akad. Nauk. USSR*, 269(3):543–7, 1983.
- [24] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–52, May 2005.
- [25] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.
- [26] D. Kim, S. Ramani, and J. A. Fessler. Ordered subsets with momentum for accelerated X-ray CT image reconstruction. In *Proc. IEEE Conf. Acoust. Speech Sig. Proc.*, pages 920–3, 2013.
- [27] D. Kim, S. Ramani, and J. A. Fessler. Accelerating X-ray CT ordered subsets image reconstruction with Nesterov’s first-order methods. In *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med.*, pages 22–5, 2013.
- [28] D. Kim and J. A. Fessler. Ordered subsets acceleration using relaxed momentum for X-ray CT image reconstruction. In *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, 2013. To appear.
- [29] O. Devolder. Stochastic first order methods in smooth convex optimization, 2011.
- [30] M. Schmidt, N. Le Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Adv. in Neural Info. Proc. Sys.*, pages 1458–66, 2011.
- [31] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal Stat. Soc. Ser. B*, 58(1):267–88, 1996.
- [32] T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci.*, 2(2):323–43, 2009.

- [33] B. Vandeghinste, B. Goossens, R. Van Holen, C. Vanhove, A. Pizurica, S. Vandenberghe, and S. Staelens. Combined shearlet and TV regularization in sparse-view CT reconstruction. In *Proc. 2nd Intl. Mtg. on image formation in X-ray CT*, pages 37–40, 2012.
- [34] H. Nien and J. A. Fessler. Combining augmented Lagrangian method with ordered subsets for X-ray CT reconstruction. In *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med.*, pages 280–3, 2013.
- [35] H. Nien and J. A. Fessler. Accelerating ordered-subsets X-ray CT image reconstruction using the linearized augmented Lagrangian framework. In *Proc. SPIE 9033 Medical Imaging 2014: Phys. Med. Im.*, page 903332, 2014.
- [36] H. Nien and J. A. Fessler. Fast splitting-based ordered-subsets X-ray CT image reconstruction. In *Proc. 3rd Intl. Mtg. on image formation in X-ray CT*, 2014. To appear.
- [37] H. Nien and J. A. Fessler. Accelerating model-based X-ray CT image reconstruction using variable splitting methods with ordered subsets. In *SIAM Conf. Imaging Sci., Abstract Book*, 2014. Invited talk.
- [38] H. Nien and J. A. Fessler. Fast X-ray CT image reconstruction using the linearized augmented Lagrangian method with ordered subsets. *arXiv: 1402.4381*, 2014. Submitted to *IEEE Trans. Med. Imag.*.
- [39] Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low-rank representation. In *Adv. in Neural Info. Proc. Sys.*, pages 612–20, 2011.
- [40] X. Wang and X. Yuan. The linearized alternating direction method of multipliers for Dantzig selector. *SIAM J. Sci. Comput.*, 34(5):A2792–A2811, 2012.
- [41] J. Yang and X. Yuan. Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. *Math. Comp.*, 82(281):301–29, January 2013.
- [42] Y. Xiao, S. Wu, and D. Li. Splitting and linearizing augmented Lagrangian algorithm for subspace recovery from corrupted observations. *Adv. Comput. Math.*, 38(4):837–58, May 2013.
- [43] W. Deng and W. Yin. On the global and linear convergence of the generalized alternating direction method of multipliers, 2012.
- [44] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, April 1992.
- [45] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo. An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Trans. Im. Proc.*, 20(3):681–95, March 2011.
- [46] E. Esser, X. Zhang, and T. Chan. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Imaging Sci.*, 3(4):1015–46, 2010.
- [47] X. Zhang, M. Burger, X. Bresson, and S. Osher. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM J. Imaging Sci.*, 3(3):253–76, 2010.
- [48] X. Zhang, M. Burger, and S. Osher. A unified primal-dual algorithm framework based on Bregman iteration. *Journal of Scientific Computing*, 46(1):20–46, 2011.

- [49] D. Kim, D. Pal, J-B. Thibault, and J. A. Fessler. Accelerating ordered subsets image reconstruction for X-ray CT using spatially non-uniform optimization transfer. *IEEE Trans. Med. Imag.*, 32(11):1965–78, November 2013.
- [50] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Im. Vision*, 40(1):120–45, 2011.
- [51] H. Ouyang, N. He, L. Tran, and A. G. Gray. Stochastic alternating direction method of multipliers. In S. Dasgupta and D. Mcallester, editors, *Proc. Intl. Conf. Machine Learning*, volume 28, pages 80–8. JMLR Workshop and Conference Proceedings, 2013.
- [52] I. Daubechies, M. Defrise, and C. D. Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, 57(11):1413–57, November 2004.
- [53] Y. Ouyang, Y. Chen, G. Lan, and E. Pasiliao Jr. An accelerated linearized alternating direction method of multipliers. *arXiv: 1401.6607*, 2014.
- [54] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, 2 edition, 1999.
- [55] B. O’Donoghue and E. Candès. Adaptive restart for accelerated gradient schemes. *Found. Comput. Math.*, 13, July 2013.
- [56] S. Kontogiorgis and R. R. Meyer. A variable-penalty alternating directions method for convex optimization. *Mathematical Programming*, 83(1–3):29–53, 1998.
- [57] G. T. Herman and L. B. Meyer. Algebraic reconstruction techniques can be made computationally efficient. *IEEE Trans. Med. Imag.*, 12(3):600–9, September 1993.
- [58] N. Le Roux, M. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for strongly-convex optimization with finite training sets. In *Adv. in Neural Info. Proc. Sys.*, pages 2672–80, 2012.
- [59] M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient, 2013.
- [60] G-H. Chen, J. Tang, and S. Leng. Prior image constrained compressed sensing (PICCS): A method to accurately reconstruct dynamic CT images from highly undersampled projection data sets. *Med. Phys.*, 35(2):660–3, February 2008.
- [61] R. C. Fair. On the robust estimation of econometric models. *Ann. Econ. Social Measurement*, 2:667–77, October 1974.
- [62] H. Nien and J. A. Fessler. Splitting-based statistical X-ray CT image reconstruction with blind gain correction. In *Proc. SPIE 8668 Medical Imaging 2013: Phys. Med. Im.*, page 86681J, 2013.
- [63] J-B. Thibault, Z. Yu, K. Sauer, C. Bouman, and J. Hsieh. Correction of gain fluctuations in iterative tomographic image reconstruction. In *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med.*, pages 112–5, 2007.
- [64] J. A. Fessler and S. D. Booth. Conjugate-gradient preconditioning methods for shift-variant PET image reconstruction. *IEEE Trans. Im. Proc.*, 8(5):688–99, May 1999.
- [65] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, pages 3–20. MIT Press, Cambridge, MA, 1991.
- [66] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. In *SIG-GRAPH*, pages 39–46, 1995.

- [67] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH*, pages 31–42, 1996.
- [68] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *SIGGRAPH*, pages 43–54, 1996.
- [69] I. Daubechies, B. Han, A. Ron, and Z. Shen. Framelets: MRA-based constructions of wavelet frames. *Appl. Comput. Harmon. Anal.*, 14(1):1–46, January 2003.
- [70] E. Candes, L. Dermanet, D. L. Donoho, and L. Ying. Fast discrete curvelet transforms. *SIAM J. Multiscale Model. Simul.*, 5(3):861–99, 2006.
- [71] J. Chai, S. Chan, H. Shum, and X. Tong. Plenoptic sampling. In *SIGGRAPH*, pages 307–18, 2000.
- [72] A. Isaksen, L. McMillan, and S. J. Gortler. Dynamically reparameterized light fields. In *SIGGRAPH*, pages 297–306, 2000.
- [73] M. Zwicker, W. Matusik, F. Durand, and H. Pfister. Antialiasing for automultiscopic 3D displays. In *Proceedings of the Eurographics Symposium on Rendering Techniques*, pages 73–82, 2006.
- [74] A. Levin, W. T. Freeman, and F. Durand. Understanding camera trade-offs through a Bayesian analysis of light field projections. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 88–101, 2008.
- [75] S. Kuthirummal, H. Nagahara, C. Zhou, and S. K. Nayar. Flexible depth of field photography. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(1):58–71, 2011.
- [76] A. Levin and F. Durand. Linear view synthesis using a dimensionality gap light field prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 13–8, 2010.
- [77] H. Nien and J. A. Fessler. A convergence proof of the split Bregman method for regularized least-squares problems. *arXiv: 1402.4371*, 2014.
- [78] J. Cai, S. Osher, and Z. Shen. Split Bregman methods and frame based image restoration. *SIAM J. Multiscale Model. Simul.*, 8(2):337–69, 2009.
- [79] M. Nikolova and M. K. Ng. Analysis of half-quadratic minimization methods for signal and image recovery. *SIAM J. Sci. Comp.*, 27(3):937–66, 2005.
- [80] E. Esser. Applications of Lagrangian-based alternating direction methods and connections to split Bregman, 2009. CAM report 9:31.
- [81] R. Glowinski and A. Marrocco. Sur l'approximation par elements nis d'ordre un, et la resolution par penalisation-dualite d'une classe de problemes de dirichlet nonlineaires, rev. francaise d'auto. *Inf. Rech. Oper.*, R-2:41–76, 1975.
- [82] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite-element approximations. *Comput. Math. Appl.*, 2(1):17–40, 1976.
- [83] S. Setzer. Split Bregman algorithm, Douglas-Rachford splitting and frame shrinkage. In X-C Tai, K Morken, M Lysaker, and K-A Lie, editors, *Scale Space and Variational Methods in Computer Vision*, pages 464–76. Springer, 2009. LNCS 5567.
- [84] J. Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Amer. Math. Soc.*, 82(2):421–39, July 1956.

- [85] P. L. Combettes and J-C. Pesquet. A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE J. Sel. Top. Sig. Proc.*, 1(4):564–74, December 2007.
- [86] I. Y. Chun and T. M. Talavage. Efficient compressed sensing statistical X-ray/CT reconstruction from fewer measurements. In *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med*, pages 30–3, 2013.
- [87] S. F. Cauley, Y. Xi, B. Bilgic, J. Xia, E. Adalsteinsson, V. Balakrishnan, L. L. Wald, and K. Setsompop. Hierarchically semiseparable (HSS) compressed sensing for MRI in linear-time, 2013.