Some operator splitting methods for convex optimization

Xinxin Li
Hong Kong Baptist University

Follow this and additional works at: https://repository.hkbu.edu.hk/etd_oa

Recommended Citation
Li, Xinxin, "Some operator splitting methods for convex optimization" (2014). Open Access Theses and Dissertations. 43.
https://repository.hkbu.edu.hk/etd_oa/43

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at HKBU Institutional Repository. It has been accepted for inclusion in Open Access Theses and Dissertations by an authorized administrator of HKBU Institutional Repository. For more information, please contact repository@hkbu.edu.hk.
Some Operator Splitting Methods for Convex Optimization

LI Xinxin

A thesis submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Principal Supervisor: Dr. YUAN Xiaoming

Hong Kong Baptist University

August 2014
DECLARATION

I hereby declare that this thesis represents my own work which has been done after registration for the degree of PhD at Hong Kong Baptist University, and has not been previously included in a thesis or dissertation submitted to this or any other institution for a degree, diploma or other qualifications.

Signature: ______________________

Date: August 2014
Abstract

Many applications arising in various areas can be well modeled as convex optimization models with separable objective functions and linear coupling constraints. Such areas include signal processing, image processing, statistical learning, wireless networks, etc. If these well-structured convex models are treated as generic models and their separable structures are ignored in algorithmic design, then it is hard to effectively exploit the favorable properties that the objective functions possibly have. Therefore, some operator splitting methods have regained much attention from different areas for solving convex optimization models with separable structures in different contexts.

In this thesis, some new operator splitting methods are proposed for convex optimization models with separable structures. We first propose combining the alternating direction method of multiplier with the logarithmic-quadratic proximal regularization for a separable monotone variational inequality with positive orthant constraints and propose a new operator splitting method. Then, we propose a proximal version of the strictly contractive Peaceman-Rachford splitting method, which was recently proposed for the convex minimization model with linear constraints and an objective function in form of the sum of two functions without coupled variables. After that, an operator splitting method suitable for parallel computation is proposed for a convex model whose objective function is the sum of three functions. For the new algorithms, we establish their convergence and estimate their convergence rates measured by the iteration complexity. We also apply the new algorithms to solve some applications arising in the image processing area; and report some preliminary numerical results. Last, we will discuss a particular video processing application and propose a series of new models for background extraction in different scenarios; to which some of the new methods are applicable.

Keywords: Convex optimization, Operator splitting method, Alternating direction method of multipliers, Peaceman-Rachford splitting method, Image processing
Acknowledgements

In the last four years, there have been many people who have encouraged, supported and helped me in my journey to pursue the PhD degree at Department of Mathematics, Hong Kong Baptist University. Without them, the thesis would never have been completed.

First and foremost, I would like to express my most gratitude and appreciation to my supervisor Dr. Xiaoming Yuan. I thank him for leading me into the amazing research world of Optimization. His endless supply of creative ideas, tireless enthusiasm on research and stringent academic attitude have greatly impressed me. He has set up an example for my academic life. I am very fortunate to have had the opportunity to work with him and grateful for his patient guidance, encouragement and research advice throughout my time of being a PhD student.

Secondly, I would gratefully thank Professor Michael Ng, particularly for his enlightening instructions and suggestions for my research on image processing.

I also thank Professor Walter Gander for his excellent and unique teaching that has showed me the great beauty and elegance of programming; and thus inspired my interest in coding.

I would like to show my heartfelt respect to Professor Bingsheng He, who is full of enthusiasm in research. I have learned a lot from numerous discussions with him. Great thanks also go to Professor Deren Han and Professor Min Li for the co-authorship of several papers and their insightful suggestions. I owe my sincere gratitude to Dr. Wenxing Zhang for his very helpful discussions and advices on my research, especially for his sharing of experience in coding which have helped me a lot for the research related to image processing. Also, I am much obliged to Dr. Caihua Chen for his valuable discussions.

I also thank all the friends I have met at Department of Mathematics, Hong Kong Baptist University. They are Ms. Lili Mo, Mr. Wenyi Tian, Mr. Chenyang Shen, Dr. Chi Pan Tam, Dr. Zhanwen Yang and others. It was their enlightening suggestions
and encouragements which have made me feel that I was not isolated in my research. Last but not the least, my thanks would go to my beloved family, for their constant support and unconditional love. I am especially indebted to my mother for everything she has done for me.
## Contents

Abstract ii

Acknowledgements iii

Table of Contents v

List of Figures ix

List of Tables xii

### Chapter 1 Introduction

1.1 Some convex programming models ....................... 1
1.2 Notations ............................................. 2
1.3 Convex model (P1) ..................................... 5
   1.3.1 Optimality condition for (P1) ....................... 5
   1.3.2 Augmented Lagrangian method ..................... 6
   1.3.3 A typical application: basis pursuit .............. 6
1.4 Convex model (P2) ..................................... 7
   1.4.1 Optimality condition for (P2) ....................... 7
   1.4.2 Some operator splitting methods for solving (P2) .... 8
      1.4.2.1 The alternating direction method of multipliers (ADMM) 8
      1.4.2.2 Generalized ADMM ............................. 10
      1.4.2.3 Linearized ADMM ............................. 10
      1.4.2.4 The Peaceman-Rachford splitting method (PRSM) .......... 12
      1.4.2.5 The strictly contractive PRSM ................. 13
1.4.3 A typical application: image restoration ........................................... 14
1.5 A variational inequality with positive orthants (VI+) .......................... 15
1.5.1 The ADMM for VI+ ................................................................. 15
1.5.2 Generalized ADMM for VI+ ................................................... 16
1.5.3 Quadratic-proximal regularized ADMM for VI+ ............................ 16
1.5.4 LQP-regularized ADMM for VI+ ............................................. 17
1.6 Convex model (P3) ........................................................................ 18
1.6.1 Optimality condition for (P3) ..................................................... 18
1.6.2 Some operator splitting methods for solving (P3) ......................... 19
1.6.2.1 The direct application of ADMM .......................................... 19
1.6.3 A typical application: image decomposition ............................... 21
1.7 Organization of the thesis .................................................................. 22

Chapter 2 A generalized ADMM with LQP regularization for a class of variational inequalities ................................................................. 24
2.1 Algorithm ..................................................................................... 25
2.2 Global convergence ...................................................................... 25
2.3 Convergence rate ......................................................................... 33
2.3.1 Convergence rate in an ergodic sense ....................................... 33
2.3.2 Convergence rate in a nonergodic sense ................................... 34

Chapter 3 A proximal strictly contractive PRSM for (P2) ............................... 36
3.1 Algorithm ..................................................................................... 36
3.2 Global convergence ...................................................................... 38
3.3 Convergence rate ......................................................................... 48
3.3.1 Convergence rate in an ergodic sense ....................................... 48
3.3.2 Convergence rate in a nonergodic sense ................................... 49
3.4 Applications to image processing .................................................. 51
3.4.1 A wavelet-based inpainting model ........................................... 52
3.4.1.1 Application background ...................................................... 53
3.4.1.2 Implementation of PSC-PRSM ............................................ 53
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2 Future work</td>
<td>126</td>
</tr>
<tr>
<td>References</td>
<td>128</td>
</tr>
<tr>
<td>Curriculum Vitae</td>
<td>144</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Image deconvolution</td>
<td>14</td>
</tr>
<tr>
<td>1.2</td>
<td>Image decomposition</td>
<td>21</td>
</tr>
<tr>
<td>3.1</td>
<td>Clean and degraded images of Peppers</td>
<td>56</td>
</tr>
<tr>
<td>3.2</td>
<td>The evolution of SNR (dB) w.r.t. computing time for Peppers</td>
<td>58</td>
</tr>
<tr>
<td>3.3</td>
<td>Restored images by the three test methods on Peppers</td>
<td>59</td>
</tr>
<tr>
<td>3.4</td>
<td>Clean and degraded images of Boat</td>
<td>62</td>
</tr>
<tr>
<td>3.5</td>
<td>The evolution of SNR (dB) w.r.t. computing time for Boat</td>
<td>62</td>
</tr>
<tr>
<td>3.6</td>
<td>Restored images by the three test methods on Boat</td>
<td>63</td>
</tr>
<tr>
<td>3.7</td>
<td>Shepp-Logan phantom, MRI and their FBP images</td>
<td>66</td>
</tr>
<tr>
<td>3.8</td>
<td>Comparison results for Shepp-Logan Phantom</td>
<td>68</td>
</tr>
<tr>
<td>3.9</td>
<td>Comparison results for Magnetic Resonance Image</td>
<td>69</td>
</tr>
<tr>
<td>3.10</td>
<td>Comparison results of test methods for Shepp-Logan Phantom</td>
<td>70</td>
</tr>
<tr>
<td>3.11</td>
<td>Comparison results of test methods for Magnetic Resonance Image</td>
<td>71</td>
</tr>
<tr>
<td>3.12</td>
<td>Images reconstructed by PSC-PRSM methods</td>
<td>72</td>
</tr>
<tr>
<td>3.13</td>
<td>Images reconstructed by test methods</td>
<td>73</td>
</tr>
<tr>
<td>4.1</td>
<td>The original images for the test of local effect</td>
<td>90</td>
</tr>
<tr>
<td>4.2</td>
<td>(a) result by our proposed method; (b) result by Ng &amp; Wang’s method with ( \mu = 10^{-5} ); (c) result by Ng &amp; Wang’s method with ( \mu = 10^{-2} )</td>
<td>90</td>
</tr>
<tr>
<td>4.3</td>
<td>Test examples: (a) the color wheel; (b) dark color wheel; (c) noisy paper; (d) a waving girl; (e) buildings; (f) people</td>
<td>91</td>
</tr>
</tbody>
</table>
4.4 From top to bottom: (a) and (b): the enhanced images; (c) and (d): the residual images with Figure 4.3(a); (e) and (f): the histogram distributions of S-CIELAB with Figure 4.3(a); (g) and (h): the spatial distribution of the errors with with Figure 4.3(a) (The areas where the difference are higher than 20 units are marked by green color).

4.5 From top to bottom: (a) and (b): the enhanced images; (c) and (d): the histogram distributions of S-CIELAB with Figure 4.3(c); (e) and (f): the spatial distribution of the errors with with Figure 4.3(c) (The areas where the difference are higher than 25 units are marked by green color).

4.6 From top to bottom: (a) and (b): the enhanced images; (c) and (d): the histogram distributions of S-CIELAB with Figure 4.3(d); (e) and (f): the spatial distribution of the errors with with Figure 4.3(d) (The areas where the difference are higher than 25 units are marked by green color).

4.7 From top to bottom: (a) and (b): the enhanced images; (c) and (d): the histogram distributions of S-CIELAB with Figure 4.3(e); (e) and (f): the spatial distribution of the errors with with Figure 4.3(e) (The areas where the difference are higher than 25 units are marked by green color).

5.1 Backgrounds of synthetic videos. Left: Cameraman. Right: Barbara.

5.2 Some frames of test videos.

5.3 Numerical comparisons of RPCA and MED on Cameraman.

5.4 Numerical comparisons of RPCA and MED on Barbara.

5.5 Numerical comparisons of RPCA and MED on Hall.

5.6 Numerical comparisons of RPCA and MED on Mall.
5.7 Evolutions of relative error w.r.t. iterations for Cameraman. . . . . 115
5.8 Evolutions of the errors w.r.t. the frames for synthetic videos. . . . 115
5.9 Numerical comparisons of RPCA-i and MED-i on noisy Cameraman. 116
5.10 Numerical comparisons of RPCA-i and MED-i on noisy Mall. . . . 117
5.11 Some frames of test degraded videos with motion blur and noise. . 118
5.12 Numerical comparisons of RPCA-ii and MED-ii on degraded Cameraman 119
5.13 Numerical comparisons of RPCA-ii and MED-ii on degraded Barbara. 120
5.14 Numerical comparisons of RPCA-ii and MED-ii on degraded Hall. . . 121
5.15 Numerical comparisons of RPCA-ii and MED-ii on degraded Mall. . . 121
5.16 Some frames of test corrupted real videos with blur and noise. . . . 121
5.17 Numerical comparisons of RPCA-ii and MED-ii on corrupted Hall. . 122
5.18 Numerical comparisons of RPCA-ii and MED-ii on corrupted Mall. . 123
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Comparison of computing time to achieve different SNR on Peppers.</td>
<td>59</td>
</tr>
<tr>
<td>3.2</td>
<td>Comparison of computing time to achieve different SNR on Boat.</td>
<td>62</td>
</tr>
<tr>
<td>3.3</td>
<td>Comparison of computing time to achieve different SNR on Phantom.</td>
<td>71</td>
</tr>
<tr>
<td>3.4</td>
<td>Comparison of computing time to achieve different SNR on MRI.</td>
<td>72</td>
</tr>
<tr>
<td>5.1</td>
<td>Computational results on background extraction for noise-free videos.</td>
<td>112</td>
</tr>
<tr>
<td>5.2</td>
<td>Parameters for models RPCA-i and MED-i.</td>
<td>114</td>
</tr>
<tr>
<td>5.3</td>
<td>Computational results for noisy videos (synthetic datasets).</td>
<td>116</td>
</tr>
<tr>
<td>5.4</td>
<td>Computational results for noisy videos (real datasets).</td>
<td>117</td>
</tr>
<tr>
<td>5.5</td>
<td>Parameters for models RPCA-ii and MED-ii.</td>
<td>119</td>
</tr>
<tr>
<td>5.6</td>
<td>Computational results for the motion blurred and noisy videos.</td>
<td>120</td>
</tr>
<tr>
<td>5.7</td>
<td>Computational results for the blurred and noisy videos.</td>
<td>122</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Some convex programming models

In this thesis, we focus on a class of convex programming problems with separable objective functions and linear coupling constraints. Depending on the extent of separability, we discuss three abstract models of this kind.

a). A generic convex minimization problem with linear constraints:

\[
\min \{ \theta(x) | Ax = b, \ x \in \mathcal{X} \}, \quad (P1)
\]

where \( A \in \mathbb{R}^{m \times n} \) is full column rank, \( b \in \mathbb{R}^m \), \( \mathcal{X} \subseteq \mathbb{R}^n \) is a closed convex nonempty set and \( \theta : \mathbb{R}^n \rightarrow \mathbb{R} \) is a closed convex proper function (possibly nonsmooth).

b). A particular separable case of \( (P1) \) where the objective function is separable into two individual functions without coupled variables. For this case, by decomposing the linear constraints into two parts accordingly, we consider the model

\[
\min \{ \theta_1(x_1) + \theta_2(x_2) | A_1 x_1 + A_2 x_2 = b, \ (x_1, x_2) \in (\mathcal{X}_1 \times \mathcal{X}_2) \}, \quad (P2)
\]

where \( A_1 \in \mathbb{R}^{m \times n_1} \), \( A_2 \in \mathbb{R}^{m \times n_2} \) are full column rank matrices, \( b \in \mathbb{R}^m \); \( \mathcal{X}_1 \subseteq \mathbb{R}^{n_1} \), \( \mathcal{X}_2 \subseteq \mathbb{R}^{n_2} \) are closed convex nonempty sets and \( \theta_1, \theta_2 \) are closed convex
proper functions. To exploit the separable structure fully and develop more customized algorithms, the philosophy of algorithmic design for the separable case \( \text{[P2]} \) is different from that for the generic case \( \text{[P1]} \).

c). A particular separable case of \( \text{[P1]} \) where the objective function is separable into three individual functions without coupled variables. Again, by rewriting the linear constraints in accordance with the separable objective function, we consider the model

\[
\min \left\{ \sum_{i=1}^{3} \theta_i(x_i) \mid \sum_{i=1}^{3} A_i x_i = b, \ x_i \in \mathcal{X}_i, \ i = 1, 2, 3 \right\},
\]

\( \text{(P3)} \)

where \( A_i \in \mathcal{R}^{m \times n_i} \ (i = 1, 2, 3) \) are full column rank matrices, \( b \in \mathcal{R}^m; \ \mathcal{X}_i \subseteq \mathcal{R}^{n_i} \ (i = 1, 2, 3) \) are closed convex nonempty sets and \( \theta_i: \mathcal{R}^{n_i} \rightarrow \mathcal{R} \ (i = 1, 2, 3) \) are closed convex proper functions. Note that \( \text{[P2]} \) is a special case of \( \text{[P3]} \).

But, we still consider \( \text{[P3]} \) individually because of its own wide applications in various fields and its unique speciality in algorithmic design.

In the remaining part of this chapter, we will first introduce some notations which are frequently used in the thesis. Then, we will give a brief summary for each considered convex model. The summary includes optimality conditions and some existing classical operator splitting methods, which are essential for subsequent analysis. We will also list some typical applications for each model, which serve to motivate our study of operator splitting methods. Finally, at the end of this chapter, we will show the outline in the rest of the thesis.

\[ \text{1.2 Notations} \]

- For any \( x \in \mathcal{R}^n \), let \( \theta : \mathcal{R}^n \rightarrow [-\infty, +\infty] \) be a function. The domain and epigraph of \( \theta \) are defined as \( \text{dom} \ \theta := \{ x \in \mathcal{R}^n \mid \theta(x) < +\infty \} \) and \( \text{epi} \ \theta := \{ (x, y) \in \mathcal{R}^n \times \mathcal{R} \mid \theta(x) \leq y \} \), respectively. The function \( \theta \) is said to be proper if \( \text{dom} \ \theta \) is nonempty and \( \theta \) is called lower semi-continuous (l.s.c) if \( \text{epi} \ \theta \) is closed in \( \mathcal{R}^n \times \mathcal{R} \).
• Let $f : \mathcal{R}^n \to \mathcal{R}$ be a convex closed proper function. $\xi \in \mathcal{R}^n$ is a subgradient of $f$ at $x$ if it satisfies the following inequality:

$$f(y) - f(x) \geq \langle \xi, y - x \rangle, \forall y \in \mathcal{R}^n.$$ (1.2.1)

The set of all subgradients of $f$ at $x$ is denoted by $\partial f(x)$, and referred to as the subdifferential of $f$ at $x$.

• Let $f : \mathcal{R}^n \to \mathcal{R}$ be a convex closed proper function. For any $x \in \mathcal{R}^n$, we define the proximal operator (also called the resolvent operator) of $f$ at $x$

$$\text{prox}_f(x) := (I + \partial f)^{-1}(x) = \arg \min \{f(y) + \frac{1}{2}\|y - x\|^2 \mid y \in \mathcal{R}^n\},$$ (1.2.2)

where $I$ is the identity operator.

• For a vector $x$, $x \in \mathcal{R}^n_+$ (Resp., $x \in \mathcal{R}^n_{++}$) means the elements in $x$ are all nonnegative (Resp., positive).

• The notation $\| \cdot \|_p$ refers to the standard definition of an $\ell_p$-norm, i.e., for any $x = (x_1, x_2, \ldots, x_n) \in \mathcal{R}^n$,

$$\|x\|_p := \left( \sum_{i=1}^{n} |x_i|^p \right)^{1/p}.$$ (1.2.3)

In particular $\| \cdot \|_2 = \| \cdot \|$.

• We also define another $p$-norm $\| | \cdot | \|_p$ on $\mathcal{R}^n \times \mathcal{R}^n$. Let $y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \in \mathcal{R}^n \times \mathcal{R}^n$, $|y|$ denotes a vector in $\mathcal{R}^n$ whose entries are given by $|y|_i := \sqrt{(y_1)_i^2 + (y_2)_i^2}$, $i = 1, 2, \ldots, n$. It follows that

$$\| |y| \|_p := \| (|y|) \|_p = \left( \sum_{i=1}^{n} |y|_i^p \right)^{1/p}.$$ (1.2.4)

• The discrete gradient operator is denoted as $\nabla := \begin{pmatrix} \nabla_x \\ \nabla_y \end{pmatrix}$, where $\nabla_x : \mathcal{R}^n \to \mathcal{R}^n$ and $\nabla_y : \mathcal{R}^n \to \mathcal{R}^n$ are the finite-difference operators in the horizontal and vertical directions respectively. It is normally assumed with Neumann (Resp.,
circulant) boundary conditions, under which it can be diagonalized by DCT (Resp., FFT), where DCT is the Discrete Cosine Transform and FFT is the Fast Fourier Transform.

• Similarly, the blurring matrix $H$ which is the matrix representation of a convolution operator (associated with a spatially invariant point spread function), can also be diagonalized by DCT (Resp., FFT) if reflective (Resp., circulant) boundary conditions are exploited, see [79].

• The divergence operator: $\text{div} := -\nabla^T = [-\nabla^T_x, \nabla^T_y]^T$. The Laplacian of $x \in \mathbb{R}^n$ is defined by $\Delta x := \text{div}(\nabla x)$.

• Let $C \subseteq \mathbb{R}^n$. $\delta_C$ denotes the indicator function of the set $C$, i.e.,

$$\delta_C(x) := \begin{cases} 0 & \text{if } x \in C, \\ \infty & \text{otherwise}. \end{cases}$$

(1.2.5)

• The proximal operator of $\| \cdot \|_1$ is the well-known soft-thresholding (also the so-called shrinkage operator, see [46]). For any $y^0 \in \mathbb{R}^n$, it is defined by

$$S_{\beta}(y^0) := \arg \min \{ \beta\|y\|_1 + \frac{1}{2}\|y - y^0\|^2 \mid y \in \mathbb{R}^n \}. \quad (1.2.6)$$

This operator has a closed-form expression,

$$S_{\beta}(y^0) = \text{sign}(y^0) \circ \max\{ \text{abs}(y^0) - \beta, 0 \}, \quad (1.2.7)$$

with abs(·) and sign(·) the absolute value and sign functions, respectively. “◦” in (1.2.7) stands for the componentwise scalar multiplication.

• For a matrix $G$, $G \succ 0$ (Resp., $\succeq 0$) means $G$ is positive definite (Resp., semi-definite).

• For any positive definite matrix $H$, we denote by $\| \cdot \|_H$ the $H$-norm, i.e., $\|x\|_H = \sqrt{x^T H x}$.

• For a matrix $X \in \mathbb{R}^{m \times n}$, the Frobenius norm of $X$ is denoted by $\|X\|_F$. The nuclear norm of $X$ is defined to be $\|X\|_* = \sum_{i=1}^{q} \sigma_i(X)$, where $\sigma_i(X), i = 1, \ldots, q$ are the singular values of $X$ and $q = \min\{m, n\}$. 
1.3 Convex model (P1)

In this section, we consider the model

\[
\min\{\theta(x) \mid Ax = b, \ x \in \mathcal{X}\}, \quad (P1)
\]

where \( A \in \mathbb{R}^{m \times n} \) is full column rank, \( b \in \mathbb{R}^m \), \( \mathcal{X} \subseteq \mathbb{R}^n \) is a closed convex nonempty set and \( \theta : \mathbb{R}^n \to \mathbb{R} \) is a closed convex proper function (possibly nonsmooth).

1.3.1 Optimality condition for (P1)

The Lagrangian function of (P1) is given by

\[
L(x, \lambda) = \theta(x) - \lambda^T (Ax - b), \quad (1.3.1)
\]

which is defined on \( \mathcal{X} \times \mathbb{R}^m \) and \( \lambda \) is the Lagrange multiplier. Solving (P1) is equivalent to finding a saddle point of the Lagrangian function (1.3.1), denoted by \((x^*, \lambda^*)\), which satisfies

\[
\begin{align*}
&x^* \in \mathcal{X}, \quad \theta(x) - \theta(x^*) + (x - x^*)^T (-A^T \lambda^*) \geq 0, \ \forall x \in \mathcal{X}, \\
&\lambda^* \in \mathbb{R}^m, \quad (\lambda - \lambda^*)^T (Ax^* - b) \geq 0, \ \forall \lambda \in \mathbb{R}^m.
\end{align*}
\]

By denoting

\[
w = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad F_1(w) = \begin{pmatrix} -A^T \lambda \\ Ax - b \end{pmatrix}, \quad \text{and} \quad \Omega_1 = \mathcal{X} \times \mathbb{R}^m, \quad (1.3.3)
\]

the optimality condition can be characterized as a variational inequality (VI):

\[
w^* \in \Omega_1, \quad \theta(x) - \theta(x^*) + (w - w^*)^T F_1(w^*) \geq 0, \ \forall w \in \Omega. \quad (1.3.4)
\]

We denote by VI(\( \Omega_1, F_1, \theta \)) the variational inequality problem (1.3.3)-(1.3.4). Since \( F_1 \) in (1.3.3) is an affine mapping with a skew-symmetric matrix, it is monotone. We denote by \( \Omega_1^* \) the solution set of VI(\( \Omega_1, F_1, \theta \)). If \( w^* = (x^*, \lambda^*) \in \Omega_1^* \), then \( x^* \) is a solution of (P1) (see e.g., [58]).
1.3.2 Augmented Lagrangian method

For solving the model (P1), a benchmark solver in the literature is the augmented Lagrangian method (ALM) which is also known as the method of multipliers in [89,126].

The augmented Lagrangian function of (P1) is given by

$$L_\beta(x,\lambda) = \theta(x) - \lambda^T(Ax - b) + \frac{\beta}{2}\|Ax - b\|^2,$$

(1.3.5)

where \(\lambda\) is the Lagrange multiplier and \(\beta > 0\) is a penalty parameter for the violation of the linear constraints.

The iterative scheme of ALM for (P1) is

$$\begin{align*}
x^{k+1} &= \text{arg min}\ \{L_\beta(x,\lambda^k) \mid x \in \mathcal{X}\}, \\
\lambda^{k+1} &= \lambda^k - \beta(Ax^{k+1} - b).
\end{align*}$$

(1.3.6)

ALM’s convergence can be assured under rather mild conditions including cases when \(\theta\) is not smooth. In [132], it was shown that the ALM (1.3.6) is exactly the application of proximal point algorithm [108,132] to the dual problem of (P1). The equivalence between the augmented Lagrangian and Bregman iteration with linear constraints was drawn in [153].

1.3.3 A typical application: basis pursuit

Basis pursuit plays a central role in modern statistical signal processing, particularly the theory of compressed sensing (CS), for which some of the original work was done by Candès et al [26] and Donoho [45]. The fundamental principle of CS is that if a signal is sparse under a chosen basis, the signal can be recovered through convex optimization with very few measurements. There are extensive references in this field, see [24] for a recent survey.

In CS, searching for the sparsest signal \(x^*\) in some basis \(\mathcal{B}\) that matches the sensed values \(b = \Phi x\) leads to consider

$$\min\{\|\Psi x\|_0 \mid \Phi x = b, \ x \in \mathcal{R}^n\},$$

(1.3.7)
where $\Psi$ is a transform matrix defined by $B$ and $\Phi \in \mathcal{R}^{m \times n}$ is an under-determined matrix with $m \ll n$. The combinatorial optimization problem (1.3.7) is NP-hard to solve and convexification of the objective function is introduced in the following basis pursuit [34] formulation
\[
\min \{ \| \Psi x \|_1 \mid \Phi x = b, \ x \in \mathcal{R}^n \}. \tag{1.3.8}
\]
Model (1.3.8) is thus a concrete application of the model (P1) with $\theta(x) := \| \Psi x \|_1$, $A := \Phi$, $b := b$ and $\mathcal{X} := \mathcal{R}^n$.

1.4 Convex model (P2)

In this section, we consider the model
\[
\min \{ \theta_1(x_1) + \theta_2(x_2) \mid A_1x_1 + A_2x_2 = b, \ (x_1, x_2) \in (\mathcal{X}_1 \times \mathcal{X}_2) \}, \tag{P2}
\]
where $A_1 \in \mathcal{R}^{m \times n_1}$, $A_2 \in \mathcal{R}^{m \times n_2}$ are full column rank matrices, $b \in \mathcal{R}^m$; $\mathcal{X}_1 \subseteq \mathcal{R}^{n_1}$, $\mathcal{X}_2 \subseteq \mathcal{R}^{n_2}$ are closed convex nonempty sets and $\theta_1, \theta_2$ are closed convex proper functions.

1.4.1 Optimality condition for (P2)

The Lagrangian function of (P2) is given by
\[
\mathcal{L}(x_1, x_2, \lambda) = \theta_1(x_1) + \theta_2(x_2) - \lambda^T(A_1x_1 + A_2x_2 - b), \tag{1.4.1}
\]
which is defined on $\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{R}^m$ and $\lambda$ is the Lagrange multiplier. Solving (P2) is equivalent to finding a saddle point of the Lagrangian function (1.4.1), denoted by $(x_1^*, x_2^*, \lambda^*)$, which satisfies
\[
\begin{cases}
  x_1^* \in \mathcal{X}_1, & \theta_1(x_1) - \theta_1(x_1^*) + (x_1 - x_1^*)^T(-A_1^T\lambda^*) \geq 0, \ \forall x_1 \in \mathcal{X}_1, \\
  x_2^* \in \mathcal{X}_2, & \theta_2(x_2) - \theta_2(x_2^*) + (x_2 - x_2^*)^T(-A_2^T\lambda^*) \geq 0, \ \forall x_2 \in \mathcal{X}_2, \\
  \lambda^* \in \mathcal{R}^m, & (\lambda - \lambda^*)^T(A_1x_1^* + A_2x_2^* - b) \geq 0, \ \forall \lambda \in \mathcal{R}^m.
\end{cases} \tag{1.4.2}
\]
By denoting
\[ u = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad w = \begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix}, \quad F_2(w) = \begin{pmatrix} -A_1^T \lambda \\ -A_2^T \lambda \\ A_1 x_1 + A_2 x_2 - b \end{pmatrix} \]

and
\[ \Omega_2 = X_1 \times X_2 \times \mathbb{R}^m, \quad \theta(u) = \theta_1(x_1) + \theta_2(x_2), \]
the optimality condition can be characterized as a monotone VI:

\[ u^* \in \Omega, \quad \theta(u) - \theta(u^*) + (w - w^*)^T F_2(w^*) \geq 0, \quad \forall w \in \Omega. \]

We denote by VI(\(\Omega_2, F_2, \theta\)) the variational inequality problem (1.4.3)-(1.4.5). Since \(F_2\) in (1.4.3) is an affine mapping with a skew-symmetric matrix, it is monotone. We denote by \(\Omega^*_2\) the solution set of VI(\(\Omega_2, F_2, \theta\)). If \(w^* = (x_1^*, x_2^*, \lambda^*) \in \Omega_2^*, \) then \((x_1^*, x_2^*)\) is a solution of \((P2)\).

### 1.4.2 Some operator splitting methods for solving \((P2)\)

For solving the model \((P2)\), it is not wise to ignore the separable structure and apply the generic-purpose ALM directly. Since the computation task of ALM largely lies on how we solve the unconstrained subproblem in (1.3.6), i.e., to solve \((x_1, x_2)\)-subproblem if ALM is applied on model \((P2)\). The idea of operator splitting methods is to decompose the joint minimization task into multiple easier and smaller subproblems such that the involved variables can be minimized separately in the alternative order. The benefit in doing so is that solving subproblems separately are usually much easier than directly solving \((x_1, x_2)\)-subproblem. In the following, we give a brief review of some classical operator splitting methods for solving model \((P2)\).

#### 1.4.2.1 The alternating direction method of multipliers (ADMM)

A simple but powerful algorithm in the literature is the alternating direction method of multipliers (ADMM). In fact, the ADMM had been developed so far in advance of the ready availability of large-scale optimization problems. It was first introduced
in mid-1970’s by French mathematicians Glowinski and Marrocco [68], Gabay and Mercier [65]. It turns out that the ADMM is equivalent, or closely related to many famous algorithms, such as the Douglas-Rachford splitting method in PDE literature [64], Springarn’s method of partial inverses [136], Dykstra’s alternating projection method [48], Bregman iterative algorithms for \( \ell_1 \) problems in signal processing [73], proximal methods [132], and many others. We refer to [23, 50, 67] for some review papers of ADMM.

The augmented Lagrangian function for model (P2) is

\[
L_\beta(x_1, x_2, \lambda) = \theta_1(x_1) + \theta_2(x_2) - \lambda^T(A_1 x_1 + A_2 x_2 - b) + \frac{\beta}{2} \|A_1 x_1 + A_2 x_2 - b\|_2^2, \tag{1.4.6}
\]

and the iterative scheme of the ADMM for (P2) reads as

\[
\begin{aligned}
&x_1^{k+1} = \arg\min \big\{ L_\beta(x_1, x_2^k, \lambda^k) \mid x_1 \in X_1 \big\}, \\
&x_2^{k+1} = \arg\min \big\{ L_\beta(x_1^{k+1}, x_2, \lambda^k) \mid x_2 \in X_2 \big\}, \\
&\lambda^{k+1} = \lambda^k - \beta(A_1 x_1^{k+1} + A_2 x_2^{k+1} - b),
\end{aligned}
\tag{1.4.7}
\]

where \( \lambda^k \in \mathbb{R}^m \) is the Lagrange multiplier and \( \beta > 0 \) is a penalty parameter. In [65, 68, 72], it was guaranteed that the sequence generated by the ADMM converges under the condition that \( \beta > 0 \).

The ADMM scheme [1.4.7] can be regarded as a splitting version of the ALM [1.3.6] whose subproblem at each iteration is decomposed into two in Gauss-Seidel fashion. This important feature makes it possible to exploit the properties of \( \theta_1 \) and \( \theta_2 \) individually; hence some subproblems easy enough to have closed-form solutions could be generated. It also explains the recent burst of the ADMM’s novel applications in various exciting and important areas including compressive sensing, image processing, machine learning, statistics, numerical linear algebra and many others, see [29, 33, 57, 87, 118, 119, 139, 149].

In particular, we point out that although there are many discussions on the convergence of the ADMM in the literature, the estimation of the ADMM’s convergence rate had remained a theoretical challenge for a few decades until the work [88]. In [88], a worst-case \( O(1/t) \) convergence rate \(^1\) of the ADMM scheme [1.4.7] measured by the iteration complexity was established in an ergodic sense.

\(^1\)A worst-case \( O(1/t) \) convergence rate means the accuracy to a solution under certain criteria
1.4.2.2 Generalized ADMM

The ADMM can also be interpreted as Douglas-Rachford splitting method (DRSM) [47] applied to the dual problem of (P2). The connection was first explored by Gabay in [64] and also discussed by Glowinski and Le Tallec in [71]. Then Eckstein and Bertsekas [53] found that DRSM is a special application of the proximal point algorithm (PPA) proposed in [108,132]. Therefore, they applied the relaxed PPA [74] to the dual of (P2) and derived the so-called generalized ADMM for model (P2):

\[
\begin{align*}
    x_1^{k+1} &= \arg\min \{ \theta_1(x_1) - x_1^T A_1^T \lambda^k + \frac{\beta}{2} \| A_1 x_1 + A_2 x_2^k - b \|^2 \mid x_1 \in X_1 \}, \\
    x_2^{k+1} &= \arg\min \{ \theta_2(x_2) - x_2^T A_2^T \lambda^k + \frac{\beta}{2} \| [\alpha A_1 x_1^{k+1} - (1 - \alpha)(A_2 x_2^k - b)] + A_2 x_2 - b \|^2 \mid x_2 \in X_2 \}, \\
    \lambda^{k+1} &= \lambda^k - \beta \{ [\alpha A_1 x_1^{k+1} - (1 - \alpha)(A_2 x_2^{k+1} - b)] + A_2 x_2^{k+1} - b \},
\end{align*}
\]

(1.4.8)

where the parameter \( \alpha \in (0, 2) \) is a relaxation factor. We can see that the generalized ADMM (1.4.8) is equally implementable as the ADMM (1.4.7). When \( \alpha = 1 \), the generalized ADMM reduces to the original ADMM (1.4.7). Empirically, the generalized ADMM (1.4.8) with an over-relaxed choice of \( \alpha \), especially \( \alpha \in [1.5, 1.8] \) has been shown in [49,51] to accelerate the convergence of the original ADMM (1.4.7). In [53], the authors also generalize the convergence theory of generalized ADMM (1.4.8) to allow for inexact minimization.

1.4.2.3 Linearized ADMM

Note that the ADMM is essentially a first-order method and the advantage of the ADMM’s efficiency can be ensured only when all resulting subproblems can be easily solved. Consequently, we are interested in simplifying or reformulating the resulting ADMM subproblems to obtain closed-form solutions if possible.

In general, if the functions \( \theta_1 \) and \( \theta_2 \) in model (P2) are generic without any specific properties, we need to consider how to solve the subproblems iteratively subject to certain inexactness criteria and ensure the convergence, see e.g [81,118]. But when \( \| x - x^0 \|^2 \) is of the order \( O(1/t) \) after \( t \) iterations of an iterative scheme; or equivalently, it requires at most \( O(1/\epsilon) \) iterations to achieve an approximate solution with an accuracy of \( \epsilon \).
\(\theta_1\) is special such as the \(\ell_1\)-norm or the nuclear norm function, it is interesting to discuss special strategies to solve the subproblems in (1.4.7) more efficiently. For such an example arising frequently in sparse or low-rank optimization models, we can linearize the quadratic-term in the \(x_1\)-subproblem of (1.4.7) and thus the linearized subproblem has a closed-form solution representable by the soft-thresholding operator (1.2.7).

Linearizing the quadratic term \(\beta_2 \|A_1 x_1 + A_2 x_2^k - b\|_2^2\) in the \(x_1\)-subproblem of (1.4.7) results in the following approximated subproblem:

\[
\min \left\{ \theta_1(x_1) - (\lambda^k)^T(A_1 x_1 + A_2 x_2^k - b) + (v^k)^T(x_1 - x_1^k) + \frac{\mu}{2} \|x_1 - x_1^k\|_2^2 \mid x_1 \in \mathbb{R}^{n_1} \right\},
\]

where \(v^k := \beta A_1^T(A_1 x_1^k + A_2 x_2^k - b)\) is the gradient of the quadratic term \(\frac{\beta}{2} \|A_1 x_1 + A_2 x_2^k - b\|_2^2\) at \(x_1 = x_1^k\) and \(\mu > 0\) is a proximal parameter. With a simple rearrangement in (1.4.9), we see that it amounts to

\[
\min \left\{ \theta_1(x_1) + \frac{\mu}{2} \|x - x^k + \frac{\beta}{\mu} A_1^T(A_1 x_1^k + A_2 x_2^k - b - \frac{\lambda^k}{\beta})\|_2^2 \mid x_1 \in \mathbb{R}^{n_1} \right\}.
\] (1.4.10)

We thus use the solution given by (1.4.10) as an approximate solution of the \(x_1^{k+1}\) in (1.4.7) and present the linearized ADMM

\[
\begin{align*}
x_1^{k+1} &= \arg \min \left\{ \theta_1(x_1) + \frac{\mu}{2} \|x_1 - x_1^k + \frac{\beta}{\mu} A_1^T(A_1 x_1^k + A_2 x_2^k - b - \frac{\lambda^k}{\beta})\|_2^2 \mid x_1 \in \mathcal{X}_1 \right\}, \\
x_2^{k+1} &= \arg \min \left\{ \theta_2(x_2) + \frac{\mu}{2} \|A_1 x_1^{k+1} + A_2 x_2 - b - \frac{1}{\beta} \lambda^k\|_2^2 \mid x_2 \in \mathcal{X}_2 \right\}, \\
\lambda^{k+1} &= \lambda^k - \beta(A_1 x_1^{k+1} + A_2 x_2^{k+1} - b),
\end{align*}
\] (1.4.11)

where \(\mu \geq \beta \|A_1^T A_1\|\) is required to ensure the convergence.

The linearized ADMM is equivalent with the classical split inexact Uzawa method [156] in literature. Obviously, the blend of the idea of linearization and the ADMM is the essence of the linearized ADMM. In fact, we regard the linearization on the quadratic term as a must for developing efficient algorithms when \(\theta_1\) is special in the sense that the resolvent operator (1.2.2) of \(\partial \theta_1\) has a closed-form representation.

Because of the ability of taking advantage of the properties of \(\theta_1\) effectively, the linearized ADMM is very efficient for solving a broad spectrum of applications such as image processing, machine learning, computer vision, [130, 148, 152, 156]. In [88], a
worst-case $O(1/t)$ convergence rate measured by the iteration complexity was established for both (1.4.7) and (1.4.11) simultaneously.

1.4.2.4 The Peaceman-Rachford splitting method (PRSM)

Recall that the ADMM can be interpreted as Douglas-Rachford splitting method (DRSM) [47] applied to the dual problem of (P2). In PDE literature, the Peaceman-Rachford splitting method (PRSM) [102,123] is also well-known for solving discretized heat equations like DRSM. But some of its theoretical aspects and numerical performance is different from DRSM, as commented in [64,70], “it is ‘less robust’ in that it converges under more restrictive assumptions than DRSM”, but, again remarked by [64,70], “if it does converge, then its rate of convergence is faster”. In fact, some better numerical performance of PRSM over DRSM has been witnessed by many researchers and we refer to [16,71] for some numerical verification of the efficiency of PRSM.

Inspired by the connection of the ADMM and DRSM, we apply PRSM to the dual of model (P2) and obtain the iterative scheme of primal PRSM for model (P2),

\[
\begin{align*}
    x_{1}^{k+1} &= \arg \min \left\{ \theta_{1}(x_{1}) + \frac{\beta}{2} \| A_{1} x_{1} + A_{2} x_{2}^{k} - b - \frac{1}{\beta} \lambda_{k}^{k+1} \|^2 \mid x_{1} \in X_{1} \right\}, \\
    \lambda_{k+\frac{1}{2}}^{k+1} &= \lambda_{k}^{k+1} - \beta \left( A_{1} x_{1}^{k+1} + A_{2} x_{2}^{k} - b \right), \\
    x_{2}^{k+1} &= \arg \min \left\{ \theta_{2}(x_{2}) + \frac{\beta}{2} \| A_{1} x_{1}^{k+1} + A_{2} x_{2} - b - \frac{1}{\beta} \lambda_{k+\frac{1}{2}}^{k+1} \|^2 \mid x_{2} \in X_{2} \right\}, \\
    \lambda_{k+1}^{k+1} &= \lambda_{k+\frac{1}{2}}^{k+1} - \beta \left( A_{1} x_{1}^{k+1} + A_{2} x_{2}^{k+1} - b \right).
\end{align*}
\]

(1.4.12)

The PRSM (1.4.12) scheme for (P2) is very similar to the ADMM (1.4.7) except that it updates the the Lagrange multiplier twice, once after each minimization of the augmented Lagrangian function.

Just like the PRSM for the dual of (P2), the convergence of the primal PRSM (1.4.12) cannot be guaranteed unless there are further assumptions on the model (P2), which is obviously different from ADMM. In [82], this difference was explained from the perspective of contraction property—the sequence generated by the ADMM (1.4.7) is strictly contractive with respect to the solution set of the model (P2) while that by
PRSM (1.4.12) is only contractive. In [41,52] some examples were given to show that there are cases where the sequence generated by PRSM indeed stays with a constant distance to the solution set of model (P2). Thus, with only the convexity assumption on the objective function of the model (P2), the convergence of PRSM is not guaranteed. In the recent work [82], a worst-case $O(1/t)$ convergence rate of PRSM in an ergodic sense is established meaningfully without any further assumptions on the model (P2).

1.4.2.5 The strictly contractive PRSM

The lack of strict contraction has inspired the authors of [82] to consider the following strictly contractive PRSM (SC-PRSM for abbreviation):

\[
\begin{align*}
x_1^{k+1} &= \arg\min \left\{ \theta_1(x_1) + \frac{\beta}{2} \| A_1 x_1 + A_2 x_2^k - b - \frac{1}{2} \lambda^k \|_2^2 \mid x_1 \in X_1 \right\}, \\
\lambda^{k+\frac{1}{2}} &= \lambda^k - \alpha \beta (A_1 x_1^{k+1} + A_2 x_2^k - b), \\
x_2^{k+1} &= \arg\min \left\{ \theta_2(x_2) + \frac{\beta}{2} \| A_1 x_1^{k+1} + A_2 x_2 - b - \frac{1}{2} \lambda^{k+\frac{1}{2}} \|_2^2 \mid x_2 \in X_2 \right\}, \\
\lambda^{k+1} &= \lambda^{k+\frac{1}{2}} - \alpha \beta (A_1 x_1^{k+1} + A_2 x_2^{k+1} - b),
\end{align*}
\]

(1.4.13)

where $\alpha \in (0,1)$ is an underdetermined relaxation factor. Note that the scheme (1.4.13) can be regarded as a modification of the original PRSM (1.4.12), where both the Lagrange-multiplier-updating steps are scaled by the factor $\alpha \in (0,1)$.

It was proved in [82] that the sequence generated by (1.4.13) is strictly contractive with respect to the solution set of (P2); thus the global convergence and worst-case $O(1/t)$ convergence rate measured by the iteration complexity in a nonergodic sense were established for (1.4.13) therein. Numerically, SC-PRSM is also shown to accelerate the original PRSM empirically when $\alpha$ is chosen to be close to 1 for arbitrarily fixed $\beta$. This can be intuitively understood: SC-PRSM is strictly contractive with respect to the solution set of model (P2) while the sequence of PRSM (1.4.12) is only contractive; therefore the former usually converges to the solution set faster than the latter in practice.

\footnote{The definition of a contractive sequence is referred to [21].}
1.4.3 A typical application: image restoration

Image restoration plays an important role in various areas of applied sciences such as medical and astronomical imaging, film restoration, image and video coding and many others [9, 28, 103, 115, 135]. Let \( u \in \mathcal{R}^n \) represent an \( n_1 \times n_2 \) image with \( n = n_1 \cdot n_2 \). Note that two- or higher-dimensional images are tackled by vectorizing them as one-dimensional vectors, e.g., in the lexicographic order. The image deconvolution problem is to restore the original image \( u \) from its degraded image, denoted by \( f \), see Figure 1.1.

![original image blurred image restored image](image)

Figure 1.1: Image deconvolution

The mathematical problem is

\[
f = Bu + n,
\]

where \( B : \mathcal{R}^n \to \mathcal{R}^n \) is a blurring matrix and \( n \) is the additive Gaussian noise. Due to the ill-posedness of the problem (1.4.14), certain regularization techniques are required.

A representative model for problem (1.4.14) is:

\[
\min \left\{ \frac{1}{2} \| Bu - f \|^2 + \tau \| \nabla u \|_1 \mid u \in \mathcal{R}^n \right\},
\]

where \( \tau > 0 \) is a regularization parameter. In (1.4.15), \( \| \nabla \cdot \|_1 \) is the total variation (TV) term [133], which is well-known for preserving the edges of the recovered image. By introducing the auxiliary variables \( v \in \mathcal{R}^{2n} \), we can reformulate (1.4.15) as

\[
\min \left\{ \frac{1}{2} \| Bu - f \|^2 + \tau \| v \|_1 \mid \nabla u - v = 0, (u, v) \in (\mathcal{R}^n \times \mathcal{R}^{2n}) \right\}.
\]
Model (1.4.16) is thus a concrete application of the model (P2) with \( \theta_1(x_1) := \frac{1}{2} \| Bu - f \|^2 \), \( \theta_2(x_2) := \tau \| v \|_1 \); \( A_1 := \nabla \), \( A_2 := -I \), \( b := 0 \); and \( X_1 = \mathcal{R}^n, X_2 = \mathcal{R}^{2n} \).

1.5 A variational inequality with positive orthants (VI+)

In this section, we consider a variational inequality problem with the block-separable structure: Find a vector \( u^* \in \Omega \), such that

\[
(u - u^*)^T T(u^*) \geq 0, \quad \forall u \in \Omega
\]  

(1.5.1)

with

\[
u := \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad T(u) := \begin{pmatrix} f(x_1) \\ g(x_2) \end{pmatrix}
\]  

(1.5.2)

and

\[
\Omega := \{ (x_1, x_2) \mid A_1 x_1 + A_2 x_2 = b, \ x_1 \in \mathcal{R}^{n_1}_+, \ x_2 \in \mathcal{R}^{n_2}_+ \},
\]  

(1.5.3)

where \( f : \mathcal{R}^{n_1}_+ \to \mathcal{R}^{n_1} \) and \( g : \mathcal{R}^{n_2}_+ \to \mathcal{R}^{n_2} \) are continuous and monotone operators. We denote this separable monotone variational inequality with positive orthants by VI+ and assume its solution set, denoted by \( \Omega^* \), is nonempty.

This VI+ (1.5.1)-(1.5.3) subsumes a class of convex programming problems with block-separable structures, and thus finds a broad spectrum of applications in various fields; see, e.g., [17, 63, 66, 71, 144].

1.5.1 The ADMM for VI+

For solving VI+ (1.5.1)- (1.5.3), the iterative scheme of the ADMM (1.4.7) generates the new iterate \( w^{k+1} := (x_1^{k+1}, x_2^{k+1}, \lambda^{k+1}) \in \mathcal{R}^{n_1} \times \mathcal{R}^{n_2} \times \mathcal{R}^m \) by the tasks

\[
0 \leq x_1^{k+1} \perp \left\{ f(x_1^{k+1}) - A_1^T [ \lambda^k - \beta (A_1 x_1^{k+1} + A_2 x_2^{k} - b)] \right\} \geq 0,
\]  

(1.5.4)

\[
0 \leq x_2^{k+1} \perp \left\{ g(x_2^{k+1}) - A_2^T [ \lambda^k - \beta (A_1 x_1^{k+1} + A_2 x_2^{k+1} - b)] \right\} \geq 0,
\]  

(1.5.5)

\[
\lambda^{k+1} := \lambda^k - \beta (A_1 x_1^{k+1} + A_2 x_2^{k+1} - b),
\]  

(1.5.6)
where $\beta > 0$ and $\lambda \in \mathbb{R}^m$ is the Lagrange multiplier associated with the linear constraint in $\Omega$. At each iteration, the ADMM (1.5.4)-(1.5.6) requires us to solve two monotone complementary problems.

### 1.5.2 Generalized ADMM for VI+

Similarly, as in Section 1.4.2.2, an immediate improvement on the ADMM (1.5.4)-(1.5.6) is the generalized ADMM proposed by Eckstein and Bertsekas in [53]. More specifically, applying the generalized ADMM (1.4.8) to solve VI+, we obtain the following iterative scheme:

\[
0 \leq x_{1}^{k+1} \perp \left\{ f(x_{1}^{k+1}) - A_{1}^{T}[\lambda^{k} - \beta(A_{1}x_{1}^{k+1} + A_{2}x_{2}^{k} - b)] \right\} \geq 0, \tag{1.5.7}
\]

\[
0 \leq x_{2}^{k+1} \perp \left\{ g(x_{2}^{k+1}) - A_{2}^{T}[\lambda^{k} - \beta(\alpha A_{1}x_{1}^{k+1} - (1 - \alpha)(A_{2}x_{2}^{k} - b) + A_{2}x_{2}^{k+1} - b)] \right\} \geq 0, \tag{1.5.8}
\]

\[
\lambda^{k+1} := \lambda^{k} - \beta[\alpha A_{1}x_{1}^{k+1} - (1 - \alpha)(A_{2}x_{2}^{k} - b) + A_{2}x_{2}^{k+1} - b], \tag{1.5.9}
\]

where the parameter $\alpha \in (0, 2)$ is a relaxation factor. Obviously, the generalized ADMM (1.5.7)-(1.5.9) includes the original ADMM (1.5.4)-(1.5.6) as a special case with $\alpha = 1$.

### 1.5.3 Quadratic-proximal regularized ADMM for VI+

Similarly, as in Section 1.4.2.3, a substantial improvement on the ADMM (1.5.4)-(1.5.6) is to combine with proximal regularization, that is, we can solve the following subproblems to generate a new iterate:

\[
0 \leq x_{1}^{k+1} \perp \left\{ f(x_{1}^{k+1}) - A_{1}^{T}[\lambda^{k} - \beta(A_{1}x_{1}^{k+1} + A_{2}x_{2}^{k} - b)] + R(x_{1}^{k+1} - x_{1}^{k}) \right\} \geq 0, \tag{1.5.10}
\]

\[
0 \leq x_{2}^{k+1} \perp \left\{ g(x_{2}^{k+1}) - A_{2}^{T}[\lambda^{k} - \beta(A_{1}x_{1}^{k+1} + A_{2}x_{2}^{k+1} - b)] + S(x_{2}^{k+1} - x_{2}^{k}) \right\} \geq 0, \tag{1.5.11}
\]

\[
\lambda^{k+1} := \lambda^{k} - \beta(A_{1}x_{1}^{k+1} + A_{2}x_{2}^{k+1} - b), \tag{1.5.12}
\]

where $R(x_{1}^{k+1} - x_{1}^{k})$ and $S(x_{2}^{k+1} - x_{2}^{k})$ are quadratic proximal regularization terms, and the symmetric positive definite matrices $R \in \mathbb{R}^{n_{1} \times n_{1}}$ and $S \in \mathbb{R}^{n_{2} \times n_{2}}$ are quadratic proximal parameters. Thus, at each iteration, the ADMM with quadratic proximal
regularization (1.5.10)-(1.5.12) requires us to solve two strongly monotone complementarity problems.

1.5.4 LQP-regularized ADMM for VI+

The implementations of the ADMM (1.5.4)-(1.5.6) rely on how efficiently the resulting complementarity problems could be solved. In [7, 154], the complementarity subproblems arising in the ADMM (1.5.4)-(1.5.6) were suggested to be regularized by the logarithmic-quadratic proximal (LQP) regularization proposed in [8]. The LQP regularization forces the solutions of the ADMM subproblems to be interior points of \( \mathbb{R}^{n_1}_+ \) and \( \mathbb{R}^{n_2}_+ \), respectively, thus the complementarity subproblems (1.5.4) and (1.5.5) reduce to two easier systems of nonlinear equations.

In the following, we briefly introduce the logarithm-quadratic proximal method (LQP). Let us take the \( x_1 \)-related ADMM subproblem in (1.5.4) as an example. Instead of using the quadratic proximal term \( R(x_1 - x^k_1) \) in (1.5.10), the LQP utilizes the nonquadratic proximal regularization term

\[
R[(x_1 - x^k_1) + \mu(x^k_1 - P^2_k x^{-1}_1)],
\]

where \( \mu \in (0, 1) \) is a given constant, \( P_k \in \mathbb{R}^{n_1 \times n_1} \) is a diagonal matrix whose \( j \)-th diagonal element is given by \( (x^k_1)_j \), \( x^{-1}_1 \in \mathbb{R}^{n_1} \) is a vector whose \( j \)-th element is \( 1/(x_1)_j \). Thus, with the LQP regularization, the complementarity problem (1.5.10) is substituted by

\[
0 \leq x_1^{k+1} \perp \{ f(x_1^{k+1}) - A^T_1 [\lambda^k - \beta(A_1 x_1^{k+1} + A_2 x^k_2 - b)]
\]

\[
+ R[(x_1^{k+1} - x^k_1) + \mu(x^k_1 - P^2_k (x_1^{k+1})^{-1})] \geq 0.
\]

As proved in [6,8], the LQP method guarantees that the new iterate \( x_1^{k+1} \) obtained by solving (1.5.14) (which has the unique solution) lies in the interior of \( \mathbb{R}^{n_1}_+ \), provided that the previous iterate \( x^k \) does. Hence, the complementarity problem (1.5.14) reduces to the following system of nonlinear equations:

\[
f(x_1^{k+1}) - A^T_1 [\lambda^k - \beta(A_1 x_1^{k+1} + A_2 x^k_2 - b)] + R[(x_1^{k+1} - x^k_1) + \mu(x^k_1 - P^2_k (x_1^{k+1})^{-1})] = 0. \]
More specifically, the iterative scheme of LQP-regularized ADMM for VI+ is as follows:

\[ f(x_k^{k+1}) - A_1^T [\lambda_k - \beta (A_1 x_1^{k+1} + A_2 x_2^k - b)] + R[(x_1^{k+1} - x_1^k) + \mu (x_1^k - P_2 x_1^{k+1} - 1)] = 0, \quad (1.5.16) \]

\[ g(x_2^{k+1}) - A_2^T [\lambda_k - \beta (A_1 x_1^{k+1} + A_2 x_2^k - b)] + S[(x_2^{k+1} - x_2^k) + \mu (x_2^k - Q_2 x_2^{k+1} - 1)] = 0 \]

\[ \lambda^{k+1} := \lambda_k - \beta (A_1 x_1^{k+1} + A_2 x_2^{k+1} - b), \quad (1.5.18) \]

where \( R \in \mathbb{R}^{n_1 \times n_1}, S \in \mathbb{R}^{n_2 \times n_2}; R(x_1^{k+1} - x_1^k) \) and \( S(x_2^{k+1} - x_2^k) \) are quadratic proximal regularization terms; \( \mu \in (0, 1) \) is a given constant; \( P_k \in \mathbb{R}^{n_1 \times n_1} \) and \( Q_k \in \mathbb{R}^{n_2 \times n_2} \) are diagonal matrices whose \( j \)-th diagonal element are given by \((x_1^k)_j\) and \((x_2^k)_j\), respectively; \((x_1^{k+1})^{-1} \in \mathbb{R}^{n_1}\) and \((x_2^{k+1})^{-1} \in \mathbb{R}^{n_2}\) are vectors whose \( j \)-th element are given by \(1/(x_1^{k+1})_j\) and \(1/(x_2^{k+1})_j\), respectively. Recently, a worst-case \( O(1/t) \) convergence rate of the ADMM with LQP regulariztion in an ergodic sense was shown in [140].

### 1.6 Convex model \((P3)\)

In this section, we consider the model

\[
\min \left\{ \sum_{i=1}^{3} \theta_i(x_i) \mid \sum_{i=1}^{3} A_i x_i = b, \ x_i \in \mathcal{X}_i, \ i = 1, 2, 3 \right\}, \quad (P3)
\]

where \( A_i \in \mathbb{R}^{m \times n_i} (i = 1, 2, 3) \) are full column rank matrices, \( b \in \mathbb{R}^{m}; \mathcal{X}_i \subseteq \mathbb{R}^{n_i} (i = 1, 2, 3) \) are closed convex nonempty sets and \( \theta_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R} (i = 1, 2, 3) \) are closed convex proper functions.

#### 1.6.1 Optimality condition for \((P3)\)

The Lagrangian function of \((P3)\) is given by

\[
\mathcal{L}(x_1, x_2, x_3, \lambda) = \theta_1(x_1) + \theta_2(x_2) + \theta_3(x_3) - \lambda^T (A_1 x_1 + A_2 x_2 + A_3 x_3 - b), \quad (1.6.1)
\]

which is defined on \( \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \times \mathbb{R}^m \) and \( \lambda \) is the Lagrange multiplier. Solving \((P3)\) is equivalent to finding a saddle point of the Lagrangian function \((1.6.1)\), denoted by
(x_1^*, x_2^*, x_3^*, \lambda^*)$, which satisfies

\[
\begin{align*}
x_1^* &\in \mathcal{X}_1, & \theta_1(x_1) - \theta_1(x_1^*) + (x_1 - x_1^*)^T(-A_1^T\lambda^*) \geq 0, \forall x_1 \in \mathcal{X}_1, \\
x_2^* &\in \mathcal{X}_2, & \theta_2(x_2) - \theta_2(x_2^*) + (x_2 - x_2^*)^T(-A_2^T\lambda^*) \geq 0, \forall x_2 \in \mathcal{X}_2, \\
x_3^* &\in \mathcal{X}_3, & \theta_3(x_3) - \theta_3(x_3^*) + (x_3 - x_3^*)^T(-A_3^T\lambda^*) \geq 0, \forall x_3 \in \mathcal{X}_3, \\
\lambda^* &\in \mathcal{R}^m, & (\lambda - \lambda^*)^T(Ax - b) \geq 0, \forall \lambda \in \mathcal{R}^m.
\end{align*}
\] (1.6.2)

By denoting

\[
x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad w = \begin{pmatrix} x \\ \lambda \end{pmatrix}, \quad F_3(w) = \begin{pmatrix} -A_1^T\lambda \\ -A_2^T\lambda \\ -A_3^T\lambda \\ A_1x_1 + A_2x_2 + A_3x_3 - b \end{pmatrix}
\] (1.6.3)

and

\[
\Omega_3 = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \times \mathcal{R}^m,
\] (1.6.4)

the optimality condition can be characterized as a monotone VI:

\[
w^* \in \Omega_3, \quad \theta(x) - \theta(x^*) + (w - w^*)^TF_3(w^*) \geq 0, \forall x \in \Omega_3.
\] (1.6.5)

We denote by VI(\Omega_3, F_3, \theta) the variational inequality problem (1.6.3)-(1.6.5). Since $F_3$ in (1.6.3) is an affine mapping with a skew-symmetric matrix, it is monotone. We denote by $\Omega_3^*$ the solution set of VI(\Omega_3, F_3, \theta). If $w^* = (x_1^*, x_2^*, x_3^*, \lambda^*) \in \Omega_3^*$, then $(x_1^*, x_2^*, x_3^*)$ is a solution of (\text{P}3).

### 1.6.2 Some operator splitting methods for solving (\text{P}3)

#### 1.6.2.1 The direct application of ADMM

The augmented Lagrangian function for model (\text{P}3) is

\[
\mathcal{L}_{\beta}(x_1, x_2, x_3, \lambda) = \sum_{i=1}^3 \theta_i(x_i) - \lambda^T \left( \sum_{i=1}^3 A_i x_i - b \right) + \frac{\beta}{2} \left\| \sum_{i=1}^3 A_i x_i - b \right\|^2.
\] (1.6.6)

Intuitively, the ideas of most methods discussed for model (\text{P}2) can be extended to solve (\text{P}3). For example, we extend the ADMM (1.4.7) in a straightforward way,
yielding the following iterative scheme (EADMM for short):

\[
\begin{align*}
    x_1^{k+1} &= \arg \min \left\{ \mathcal{L}_\beta(x_1, x_2, x_3^k) \mid x_1 \in \mathcal{X}_1 \right\}, \\
    x_2^{k+1} &= \arg \min \left\{ \mathcal{L}_\beta(x_1^{k+1}, x_2, x_3^k) \mid x_2 \in \mathcal{X}_2 \right\}, \\
    x_3^{k+1} &= \arg \min \left\{ \mathcal{L}_\beta(x_1^{k+1}, x_2^{k+1}, x_3) \mid x_3 \in \mathcal{X}_3 \right\}, \\
    \lambda^{k+1} &= \lambda^k - \beta(A_1x_1^{k+1} + A_2x_2^{k+1} + A_3x_3^{k+1} - b) .
\end{align*}
\]  

(1.6.7)

The EADMM scheme (1.6.7), which comes from the straightforward splitting of the ALM subproblem in alternating order, preserves the advantages of the original ADMM scheme (1.6.7) such that \( \theta_i \)'s properties can be exploited individually and thus the subproblems might be easy. Because of its good performance in numerical implementation, the EADMM has been preferred for model (P3) for a long time by many researchers in different areas, despite the absence of the EADMM’s convergence analysis. Until the recent work in [32], the authors present a counter example to show that the iterative sequence obtained by the EADMM fails to converge, which means that the EADMM is not necessarily convergent unless there are further assumptions on the model (P3).

Likely, most ideas of the other splitting methods for model (P2) cannot be extended in a straightforward manner to solve model (P3) either. However, there is a series of variant operator splitting methods developed for model (P3), see [78, 83–85], which all have theoretical guarantees on the convergence. These splitting algorithms are based on different algorithmic frameworks and each of them is particularly efficient for some specific applications. They all share the benefit of preserving the decomposition nature as (1.6.7). Under some special circumstances, such as the matrices in linear constraints are identity or sparse, the performance of some splitting algorithms can be comparable with the EADMM (1.6.7). For succinctness, we do not go into details of these splitting methods, and refer to the papers mentioned above for further analysis.

Finally, we give a remark that although our discussions on model (P1), (P2) and (P3) are concentrated on the vector case, similar analysis for the considered models and reviewed methods can also apply to the matrix case.
1.6.3 A typical application: image decomposition

Image decomposition plays an important role in realm of object recognition, biomedical engineering, astronomical imaging, etc \cite{3, 5, 15, 59, 110, 137}. The target image is required to be decomposed into two meaningful components, one is the geometrical part or sketchy approximation of image which is called \textit{cartoon}, and the other is the oscillating or small scale special patterns of image which is called \textit{texture}. Mathematically, the cartoon can be described by a piecewise smooth (or piecewise constant) function whilst the texture is commonly oscillating. Because of their different properties, it is more efficient and effective to separate them for processing.

For a given image $f \in \mathcal{R}^n$ tackled by vectorization, the image decomposition is to derive $u$ and $v$ such that $f = u + v$, where $u$ represents the cartoon and $v$ represents the texture, see Figure 1.2.

![Figure 1.2: Image decomposition](image)

In \cite{120}, the authors study an image decomposition model for images with corruptions, e.g. blurry and/or missing pixels:

$$\min \{\tau \|\nabla u\|_1 + \frac{1}{2} \|K(u + \text{div} g) - f\|_2^2 + \mu \|g\|_p \mid (u, g) \in (\mathcal{R}^n \times \mathcal{R}^n)\}, \quad (1.6.8)$$

with $p \geq 1$. Here, $K : \mathcal{R}^n \rightarrow \mathcal{R}^n$ is a linear operator; $\tau, \mu \geq 0$ are trade-offs to balance the image $f$ into the cartoon $u$ and the texture $\text{div} g$, respectively. Different choices of $K$ correspond to the observed images with different corruptions: (i) images with blurry, i.e., $K = B$ where $B$ is the blurring matrix; (ii) images with missing pixels, i.e., $K = S$ where $S$ is a binary diagonal matrix (also the so-called mask).
By introducing auxiliary variable $x$, $y$ and $z$, the model (1.6.8) amounts to

$$\begin{align*}
\min \quad & \tau \|x\|_1 + \|K y - f\|_2^2 + \mu \|z\|_p \\
\text{s.t.} \quad & x = \nabla u \\
& y = u + \text{div } g \\
& z = g.
\end{align*}$$

(1.6.9)

Model (1.6.9) is thus a concrete application of the model (P3) with $\theta_1(x_1) := 0, \theta_2(x_2) := 0, \theta_3(x_3) := \tau \|x\|_1 + \|K y - f\|_2^2 + \mu \|z\|_p$; $A_1 := (0, \text{div}^T, I)^T, A_2 := (\nabla^T, I, 0)^T, A_3 := -I, b := 0$ and $\mathcal{X}_1 = \mathcal{R}^n, \mathcal{X}_2 = \mathcal{R}^n, \mathcal{X}_3 = \mathcal{R}^{2n} \times \mathcal{R}^n \times \mathcal{R}^n$.

1.7 Organization of the thesis

The remaining parts of the thesis are organized as follows.

- In Chapter 2, for solving the structured variational inequality problem with positive orthants $\text{VI}^+$ (1.5.1)-(1.5.3), we consider combining the generalized ADMM (1.5.7)-(1.5.9) with the logarithmic-quadratic regularization (1.5.14). For the derived algorithm, we prove its global convergence and establish its worst-case convergence rates measured by the iteration complexity in both the ergodic and nonergodic senses.

- In Chapter 3, we focus on the model (P2) and propose a proximal version of SC-PRSM (1.4.13) in [82]. We establish the worst-case convergence rate measured by the iteration complexity in both the ergodic and nonergodic senses for the new algorithm. We also present some applications in image processing to demonstrate the efficiency of the new algorithm, and its superiority to some existing benchmarks in the literature.

- In Chapter 4, we focus on the model (P3) and propose a parallel operator splitting algorithm, where all subproblems can have closed-form solutions. Moreover, the variables are updated in a simultaneous way, which makes the algorithm need less CPU time than that of a sequential one in a parallel implementation. Under very mild conditions, we prove the global convergence of the
In Chapter 5, we focus on an application in video processing—the background extraction problem in surveillance video. We propose some median-filter based variational models for extracting static backgrounds from surveillance videos corrupted by noise, blur or both. The new models are constructed based on the fact that the matrix representation of a static background consists of identical columns; hence the idea of median filter is embedded in these models. These new models significantly differ from existing models originating from the robust principal component analysis (RPCA) in that no nuclear-norm term is involved; thus the computation of singular value decomposition (SVD) can be completely avoided when solving these new models iteratively. We show that these new models can be fit into the abstract models (P1) and (P2), thus, they can be easily solved by well-developed operator splitting methods such as the ADMM (1.4.7) and the EADMM (1.6.7), etc. We compare the new models with their PRCA-based counterparts via testing some synthetic and real videos. Our numerical results show that compared with RPCA-based models, these median filter based variational models can extract more accurate backgrounds when the background in a surveillance video is static and numerically they can be solved much more efficiently.

Finally, in Chapter 6, we present some concluding remarks and discuss several future research topics.
Chapter 2

A generalized ADMM with LQP regularization for a class of variational inequalities

In this chapter, we focus on the variational inequality problem with separable structures and positive orthants. Recall the VI+ which is to find $u^* \in \Omega$ such that

$$
(u - u^*)^T T(u^*) \geq 0, \quad \forall u \in \Omega,
$$

with

$$
u := \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad T(u) := \begin{pmatrix} f(x_1) \\ g(x_2) \end{pmatrix}
$$

and

$$
\Omega := \{(x_1, x_2) \mid A_1 x_1 + A_2 x_2 = b, \ x_1 \in \mathcal{R}^{n_1}_+, \ x_2 \in \mathcal{R}^{n_2}_+\},
$$

where $A_1 \in \mathbb{R}^{m \times n_1}$ and $A_2 \in \mathbb{R}^{m \times n_2}$ are given matrices; $b \in \mathbb{R}^m$ is a given vector; $f : \mathcal{R}^{n_1}_+ \rightarrow \mathcal{R}^{n_1}$ and $g : \mathcal{R}^{n_2}_+ \rightarrow \mathcal{R}^{n_2}$ are continuous and monotone operators.

In the following, we consider combining the generalized ADMM (1.5.7)-(1.5.9) with the logarithmic-quadratic regularization (1.5.14). For the derived algorithm, we prove its global convergence and establish its worst-case convergence rates measured by the iteration complexity in both the ergodic and nonergodic senses.
2.1 Algorithm

Combining the generalized ADMM (1.5.7)-(1.5.9) with the LQP regularization (1.5.14) in [8], we propose the following generalized ADMM with LQP regularization for solving VI+ (2.0.1)-(2.0.3):

\[ f(x_1^{k+1}) - A_1^T[\lambda^k - \beta(A_1x_1^{k+1} + A_2x_2^k - b)] + R[(x_1^{k+1} - x_1^k) + \mu(x_1^k - P_1^k(x_1^{k+1})^{-1})] = 0, \]

(2.1.1)

\[ g(x_2^{k+1}) - A_2^T[\lambda^k - \beta[\alpha A_1x_1^{k+1} - (1 - \alpha)(A_2x_2^k - b) + A_2x_2^{k+1} - b]] + S[(x_2^{k+1} - x_2^k) + \mu(x_2^k - Q_2^k(x_2^{k+1})^{-1})] = 0, \]

(2.1.2)

\[ \lambda^{k+1} := \lambda^k - \beta[\alpha A_1x_1^{k+1} - (1 - \alpha)(A_2x_2^k - b) + A_2x_2^{k+1} - b], \]

(2.1.3)

where \( \alpha \) is restricted in the interval \((0, 2)\) and \( \beta > 0 \). Obviously, the iterative scheme (2.1.1)-(2.1.3) enjoys both the advantages of the generalized ADMM and LQP.

2.2 Global convergence

In this section, we prove the global convergence of the generalized ADMM with LQP regularization (2.1.1)-(2.1.3).

First, it is easy to see that VI+ (2.0.1)-(2.0.3) can be rewritten as the following compact form: Finding \( w^* \in W \) such that

\[ (w - w^*)^T \Gamma(w^*) \geq 0, \quad \forall w \in W, \]

(2.2.1)

with

\[
\begin{pmatrix}
    x_1 \\
    x_2 \\
    \lambda
\end{pmatrix}, \quad \Gamma(w) := 
\begin{pmatrix}
    f(x_1) - A_1^T \lambda \\
    g(x_2) - A_2^T \lambda \\
    A_1 x_1 + A_2 x_2 - b
\end{pmatrix} \quad \text{and} \quad W := \mathcal{R}_+^{n_1} \times \mathcal{R}_+^{n_2} \times \mathcal{R}^m.
\]

(2.2.2)

We denote by VI+(\( W, \Gamma \)) the variational inequality problem (2.2.1)-(2.2.2); and by \( W^* \) the solution set of VI+(\( W, \Gamma \)). Note the mapping \( \Gamma(w) \) in (2.2.2) is monotone with respect to \( W \) under the monotonicity assumption of \( f \) and \( g \). Therefore, the solution set \( W^* \) of VI+(\( W, \Gamma \)) is closed and convex; see, e.g., [58]. In later analysis of convergence, we will focus on this reformulation.
Next, we recall a characterization of the solution set $W^*$ proved in [58], see (2.3.2) in pp. 159 of [58]:

$$W^* := \bigcap_{w \in W} \{ \bar{w} \in W \mid (w - \bar{w})^T \Gamma(w) \geq 0 \}.$$ 

Therefore, as Definition 1 in [116], we give the definition of an $\varepsilon$-approximation solution of the variational inequality problem.

**Definition 2.2.1** $\bar{w} \in W$ can be regarded as an $\varepsilon$-approximation solution of $VI^+(W, \Gamma)$ iff it satisfies

$$\sup_{w \in B_W(\bar{w})} \{(\bar{w} - w)^T \Gamma(w)\} \leq \varepsilon,$$

where $B_W(\bar{w}) := \{ w \in W \mid \|w - \bar{w}\| \leq 1 \}$. (2.2.3)

Based on this definition, the worst-case $\mathcal{O}(1/t)$ convergence rate of the generalized ADMM with LQP regularization in an ergodic sense will be established in the sense that we can find a $\bar{w} \in W$ such that

$$(\bar{w} - w)^T \Gamma(w) \leq \varepsilon, \quad \forall w \in B_W(\bar{w}),$$

with $\varepsilon = \mathcal{O}(1/t)$, after $t$ iterations.

The following lemma was proved in [154] and it was inspired by Proposition 1 in [8]. We need this lemma to analyze the convergence for the generalized ADMM with LQP regularization.

**Lemma 2.2.2** Let $P := \text{diag}(p_1, p_2, \cdots, p_t) \in \mathcal{R}^{t \times t}$ be a positive definite diagonal matrix, $q(u) \in \mathcal{R}^t$ be a monotone mapping of $u$ with respect to $\mathcal{R}^+_{++}$, and $\mu \in (0, 1)$. For a given $\bar{u} \in \mathcal{R}_+^{t+}$, we define $\bar{U} := \text{diag}(\bar{u}_1, \bar{u}_2, \cdots, \bar{u}_t)$. Then, the equation

$$q(u) + P[(u - \bar{u}) + \mu(\bar{u} - \bar{U}^2 u^{-1})] = 0$$

has the unique positive solution $u$. In addition, for this positive solution $u \in \mathcal{R}_+^{t+}$ and any $v \in \mathcal{R}_+^{t+}$, we have

$$(v - u)^T q(u) \geq \frac{1 + \mu}{2} \left(\|u - v\|_P^2 - \|\bar{u} - v\|_P^2\right) + \frac{1 - \mu}{2} \|\bar{u} - u\|_P^2.$$ 

(2.2.5)

**Remark 2.2.3** It follows from Lemma 2.2.2 that there exist unique $x_1^{k+1} \in \mathcal{R}_{++}^{n_1}$ and $x_2^{k+1} \in \mathcal{R}_{++}^{n_2}$ satisfying (2.1.1) and (2.1.2), respectively.
We define two matrices $G_\alpha$ and $M_\alpha$ as follows:

$$G_\alpha := \begin{pmatrix} (1 + \mu)R & 0 & 0 \\ 0 & (1 + \mu)S + \frac{\beta}{\alpha}A_2^T A_2 & \frac{1-\alpha}{\alpha} A_2^T \\ 0 & \frac{1-\alpha}{\alpha} A_2 & \frac{1}{\alpha^\beta} I \end{pmatrix}$$ \hspace{1cm} (2.2.6)

and

$$M_\alpha := \begin{pmatrix} (1 - \mu)R & 0 & 0 \\ 0 & (1 - \mu)S & 0 \\ 0 & 0 & \frac{2-\alpha}{\beta} I \end{pmatrix}$$ \hspace{1cm} (2.2.7)

where $\mu \in (0, 1)$ and $\alpha \in (0, 2)$. Obviously, the matrices $G_\alpha$ and $M_\alpha$ are positive definite if $R$ and $S$ are positive definite. These matrices will help us present the convergence analysis in a more compact notation.

Then, as in [88, 140], let us define an auxiliary variable:

$$\bar{w}^k := \begin{pmatrix} \bar{x}_1^k \\ \bar{x}_2^k \\ \bar{\lambda}^k \end{pmatrix} = \begin{pmatrix} x_1^{k+1} \\ x_2^{k+1} \\ \lambda^k - \beta (A_1 x_1^{k+1} + A_2 x_2^k - b) \end{pmatrix}. \hspace{1cm} (2.2.8)$$

The sequence $\{\bar{w}^k\}$ will be used frequently in later analysis.

To prove the global convergence for the generalized ADMM with LQP regularization, we first present some lemmas.

**Lemma 2.2.4** Let the sequence $\{w^k\}$ be generated by the generalized ADMM with LQP regularization [2.1.1] – [2.1.3], $\alpha \in (0, 2)$, and the accompanying sequence $\{\bar{w}^k\}$ be defined by (2.2.8). Then, for any $(x_2, \lambda) \in \mathcal{R}^{n_2} \times \mathcal{R}^m$, we have

$$2(\lambda - \bar{\lambda}^k)^T (A_1 \bar{x}_1^k + A_2 \bar{x}_2^k - b) + \frac{2\beta}{\alpha} (x_2 - \bar{x}_2^k)^T A_2^T A_2 (x_2^k - \bar{x}_2^k)$$

$$= \frac{\beta}{\alpha} (\|x_2 - \bar{x}_2\|^2 A_2^T A_2 - \|x_2 - \bar{x}_2^k\|^2 A_2^T A_2) + \frac{1}{\alpha\beta} (\|\lambda - \lambda^{k+1}\|^2 - \|\lambda - \lambda^k\|^2)$$

$$+ \frac{2 - \alpha}{\beta} \|\lambda^k - \bar{\lambda}^k\|^2 + \frac{2(1 - \alpha)}{\alpha} (\lambda - \lambda^k)^T (A_2 x_2^k - A_2 \bar{x}_2^k).$$

**Proof:** Using (2.1.3) and (2.2.8), we have

$$\lambda^{k+1} = \lambda^k - \beta \left[ \alpha (A_1 \bar{x}_1^k + A_2 \bar{x}_2^k - b) + (\alpha - 1)(A_2 x_2^k - A_2 \bar{x}_2^k) \right], \hspace{1cm} (2.2.9)$$
from which we get

\[ A_1 \bar{x}_1^k + A_2 \bar{x}_2^k - b = \frac{1}{\alpha \beta} (\lambda^k - \lambda^{k+1}) + \frac{1 - \alpha}{\alpha} (A_2 \bar{x}_2^k - A_2 \bar{x}_2^k). \]

Then, we obtain

\[
2(\lambda - \bar{\lambda})^T (A_1 \bar{x}_1^k + A_2 \bar{x}_2^k - b) \\
= \frac{2}{\alpha \beta} (\lambda - \bar{\lambda})^T (\lambda^k - \lambda^{k+1}) + \frac{2(1 - \alpha)}{\alpha} (\lambda - \bar{\lambda})^T (A_2 \bar{x}_2^k - A_2 \bar{x}_2^k) \\
= \frac{2}{\alpha \beta} (\lambda - \lambda^{k+1})^T (\lambda^k - \lambda^{k+1}) + \frac{2}{\alpha \beta} (\lambda^{k+1} - \bar{\lambda})^T (\lambda^k - \lambda^{k+1}) \\
+ \frac{2(1 - \alpha)}{\alpha} (\lambda^k - \bar{\lambda})^T (A_2 \bar{x}_2^k - A_2 \bar{x}_2^k) + \frac{2(1 - \alpha)}{\alpha} (\lambda^k - \lambda^{k+1})^T (A_2 \bar{x}_2^k - A_2 \bar{x}_2^k). \tag{2.2.10}
\]

For the first item in (2.2.10), we have the following identity

\[
\frac{2}{\alpha \beta} (\lambda - \lambda^{k+1})^T (\lambda^{k+1} - \lambda^k) = \frac{1}{\alpha \beta} \|\lambda - \lambda^{k+1}\|^2 - \lambda^{k+1} \lambda^k + \frac{1}{\alpha \beta} \|\lambda^{k+1} - \lambda^k\|^2. \tag{2.2.11}
\]

Then, it follows from (2.1.3) and (2.2.8) that

\[
\lambda^{k+1} = \lambda^k - \beta \left[ \alpha (A_1 \bar{x}_1^k + A_2 \bar{x}_2^k - b) + (A_2 \bar{x}_2^k - A_2 \bar{x}_2^k) \right] = \lambda^k - \alpha (\lambda^k - \bar{\lambda}) + \beta (A_2 \bar{x}_2^k - A_2 \bar{x}_2^k).
\]

Therefore, we obtain

\[
\lambda^{k+1} - \bar{\lambda} = (1 - \alpha) (\lambda^k - \bar{\lambda}) + \beta (A_2 \bar{x}_2^k - A_2 \bar{x}_2^k), \tag{2.2.12}
\]

and

\[
\lambda^k - \lambda^{k+1} = \alpha (\lambda^k - \bar{\lambda}) + \beta (A_2 \bar{x}_2^k - A_2 \bar{x}_2^k). \tag{2.2.13}
\]

Using the above two equations and by simple manipulations, we get

\[
\frac{2}{\alpha \beta} (\lambda^{k+1} - \bar{\lambda})^T (\lambda^k - \lambda^{k+1}) + \frac{2(1 - \alpha)}{\alpha} (\lambda^{k+1} - \bar{\lambda})^T (A_2 \bar{x}_2^k - A_2 \bar{x}_2^k) \\
= \frac{2}{\alpha \beta} (1 - \alpha) (\lambda^k - \bar{\lambda}) + \beta (A_2 \bar{x}_2^k - A_2 \bar{x}_2^k)[\alpha (\lambda^k - \bar{\lambda}) + \beta (A_2 \bar{x}_2^k - A_2 \bar{x}_2^k)] \\
+ \frac{2(1 - \alpha)}{\alpha} (\lambda^k - \bar{\lambda})^T (A_2 \bar{x}_2^k - A_2 \bar{x}_2^k) \\
= \frac{2(1 - \alpha)}{\beta} \|\lambda^k - \bar{\lambda}\|^2 - \frac{2 \alpha}{\alpha} \|x_2^k - \bar{x}_2^k\|^2 + 2 \|A_2 \bar{x}_2^k - A_2 \bar{x}_2^k\|^2. \tag{2.2.14}
\]

It follows from (2.2.13) that

\[
\frac{1}{\beta} \|\lambda^k - \lambda^{k+1}\|^2 = \frac{\alpha^2}{\beta} \|\lambda^k - \bar{\lambda}\|^2 + \beta \|x_2^k - \bar{x}_2^k\|^2 A_2^T A_2 - 2 \alpha (\lambda^k - \bar{\lambda})^T (A_2 \bar{x}_2^k - A_2 \bar{x}_2^k),
\]
and thus
\[2(\lambda^k - \tilde{\lambda}^k)^T (A_2 x_2^k - A_2 \tilde{x}_2^k) = \frac{\alpha}{\beta} \|\lambda^k - \tilde{\lambda}^k\|^2 + \frac{\beta}{\alpha} \|x_2^k - \tilde{x}_2^k\|^2_{A_2^T A_2} - \frac{1}{\alpha \beta} \|\lambda^k - \lambda^{k+1}\|^2.\]

From the above equality and (2.2.14), we obtain
\[
\frac{2}{\alpha \beta} (\lambda^{k+1} - \tilde{\lambda}^{k+1})^T (\lambda^k - \tilde{\lambda}^k) + \frac{2(1 - \alpha)}{\alpha} (\lambda^k - \tilde{\lambda}^k)^T (A_2 x_2^k - A_2 \tilde{x}_2^k)
= \frac{2 - \alpha}{\beta} \|\lambda^k - \tilde{\lambda}^k\|^2 - \frac{\beta}{\alpha} \|x_2^k - \tilde{x}_2^k\|^2_{A_2^T A_2} - \frac{1}{\alpha \beta} \|\lambda^k - \lambda^{k+1}\|^2.
\]

Substituting this and (2.2.11) into (2.2.10), we have
\[
2(\lambda - \tilde{\lambda})^T (A_1 \bar{x}_1^k + A_2 \bar{x}_2^k - b)
= \frac{1}{\alpha \beta} (\|\lambda - \lambda^{k+1}\|^2 - \|\lambda - \lambda^k\|^2 + \frac{2 - \alpha}{\beta} \|\lambda^k - \tilde{\lambda}^k\|^2 - \frac{1}{\alpha \beta} \|x_2^k - \tilde{x}_2^k\|^2_{A_2^T A_2}
+ \frac{2(1 - \alpha)}{\alpha} (\lambda - \tilde{\lambda})^T (A_2 x_2^k - A_2 \bar{x}_2^k).
\]

The following is an identity
\[
\frac{2}{\alpha \beta} (x_2^k - \bar{x}_2^k)^T A_2^T A_2 (x_2^k - \bar{x}_2^k) = \frac{\beta}{\alpha} \|x_2^k - \bar{x}_2^k\|^2_{A_2^T A_2} - \frac{\beta}{\alpha} \|x_2^k - \bar{x}_2^k\|^2_{A_2^T A_2} - \frac{\beta}{\alpha} \|x_2^k - \bar{x}_2^k\|^2_{A_2^T A_2}.
\]

Adding the above two equalities, the assertion is proved. \(\square\)

**Lemma 2.2.5** Let the sequence \(\{w^k\}\) be generated by the generalized ADMM with LQP regularization (2.1.1)-(2.1.3), \(\alpha \in (0, 2)\), and the accompanying sequence \(\{\tilde{w}^k\}\) be defined by (2.2.8). Then, for any \(w := (x_1, x_2, \lambda) \in \mathcal{W}\), we have
\[
(w - \tilde{w})^T \Gamma(\tilde{w}) \geq \frac{1}{2} (\|w^{k+1} - w\|^2_{G_\alpha} - \|w^k - w\|^2_{G_\alpha}) + \frac{1}{2} \|w^k - \tilde{w}^k\|^2_{M_\alpha}, \tag{2.2.15}
\]
where \(G_\alpha\) and \(M_\alpha\) are defined by (2.2.6) and (2.2.7), respectively.

**Proof:** Note that \(x_1^k = x_1^{k+1}\) and \(x_2^k = x_2^{k+1}\). Applying Lemma 2.2.2 to (2.1.1) by setting \(P = R, \bar{u} = x_1^k, u = \bar{x}_1^k, q(u) = f(\bar{x}_1^k) - A_1^T [\lambda^k - \beta (A_1 \bar{x}_1^k + A_2 x_2^k - b)] \tag{2.2.8}\) and \(v = x_1^k, u = \bar{x}_1^k, q(u) = f(\bar{x}_1^k) - A_1^T \bar{\lambda}^k\) and \(v = x_1^k\) in (2.2.3), for any \(x_1 \in \mathcal{R}^n_+\) we have
\[
(x_1 - \bar{x}_1^k)^T [f(\bar{x}_1^k) - A_1^T \bar{\lambda}^k] \geq \frac{1 + \mu}{2} (\|x_1^{k+1} - x_1\|^2_R - \|x_1^k - x_1\|^2_R) + \frac{1 - \mu}{2} \|x_1^k - \bar{x}_1^k\|^2_R. \tag{2.2.16}
\]
Similarly, applying Lemma 2.2.2 to (2.1.2) and using (2.1.3) and (2.2.8), for any $x_2 \in \mathcal{R}_+^{m_2}$ we get

$$(x_2 - \bar{x}_2^k)^T [g(\bar{x}_2^k) - A_2^T \lambda^{k+1}] \geq \frac{1 + \mu}{2} (||x_2^{k+1} - x_2||_S^2 - ||x_2^k - x_2||_S^2) + \frac{1 - \mu}{2} ||x_2^k - \bar{x}_2^k||_S^2.\tag{2.2.17}$$

Using (2.2.12), $\lambda^{k+1}$ can be written as

$$\lambda^{k+1} = \bar{\lambda}^k - (1 - \alpha) (\bar{\lambda}^k - \lambda^k) - \beta A_2 (\bar{x}_2^k - x_2^k).\tag{2.2.18}$$

Substituting this into (2.2.17) and using the notation $\bar{w}^k$, for any $x_2 \in \mathcal{R}_+^{m_2}$ we obtain

$$(x_2 - \bar{x}_2^k)^T [g(\bar{x}_2^k) - A_2^T \lambda^k] \geq \beta (x_2 - \bar{x}_2^k)^T A_2^T A_2 (x_2 - \bar{x}_2^k) + (1 - \alpha) (x_2 - \bar{x}_2^k)^T A_2^T (\lambda^k - \bar{\lambda}^k) + \frac{1 + \mu}{2} (||x_2^{k+1} - x_2||_S^2 - ||x_2^k - x_2||_S^2) + \frac{1 - \mu}{2} ||x_2^k - \bar{x}_2^k||_S^2.\tag{2.2.19}$$

Using Lemma 2.2.4 and $\bar{x}_2^k = x_2^{k+1}$, we have

$$(\lambda - \bar{\lambda}^k)^T (\bar{A}_1 \bar{x}_2^k + A_2 \bar{x}_2^k - b) = \frac{\beta}{\alpha} (x_2 - \bar{x}_2^k)^T A_2^T A_2 (\bar{x}_2^k - x_2^k) + \frac{\beta}{2\alpha} (||x_2 - x_2^{k+1}||_{A_2^TA_2}^2 - ||x_2 - x_2^k||_{A_2^TA_2}^2) + \frac{1 - \alpha}{2\beta} \|\lambda^k - \bar{\lambda}^k\|^2 + \frac{1 - \alpha}{\alpha} (x_2 - \bar{x}_2^k)^T A_2^T (\lambda - \lambda^k).$$

Combining (2.2.16), (2.2.19) and the above equation together, we get

$$(w - \bar{w}^k)^T G(w^k) \geq \frac{1}{2} \left[ (||x_2^{k+1} - x_2||^2_{(1+\mu)R} - ||x_2^k - x_2||^2_{(1+\mu)R}) + \frac{1}{\alpha \beta} (||\lambda^{k+1} - \lambda||^2 - ||\lambda^k - \lambda||^2) 
+ (||x_2^{k+1} - x_2||^2_{\frac{\mu}{\beta} A_2 (1+\mu) S} - ||x_2^k - x_2||^2_{\frac{\mu}{\beta} A_2 (1+\mu) S}) \right] 
+ \frac{1 - \alpha}{\alpha} \left[ \beta (x_2 - \bar{x}_2^k)^T A_2^2 (x_2^k - x_2^k) + \alpha (x_2 - \bar{x}_2^k)^T A_2^2 (\lambda^k - \bar{\lambda}^k) + (x_2 - \bar{x}_2^k)^T A_2^2 (\lambda - \lambda^k) \right] 
+ \frac{1}{2} \left[ ||x_2^k - \bar{x}_2^k||^2_{(1-\mu)R} + ||x_2^k - \bar{x}_2^k||^2_{(1-\mu)S} + \frac{2 - \alpha}{\beta} ||\lambda^k - \bar{\lambda}^k||^2 \right]. \tag{2.2.20}$$
From (2.2.18) and by simple manipulations, we obtain
\[
\beta(x_2 - \bar{x}_2^k)^T A_2^T A_2(x_2^k - x_2^k) + \alpha(x_2 - \bar{x}_2^k)^T A_2^T (\lambda^k - \bar{\lambda}^k) + (x_2^k - \bar{x}_2^k) A_2^T (\lambda - \lambda^k) \\
= \beta(x_2 - \bar{x}_2^k)^T A_2^T A_2(x_2^k - x_2^k) + (x_2 - \bar{x}_2^k)^T A_2^T [\lambda - \bar{\lambda}^k + (1 - \alpha)(\bar{\lambda}^k - \lambda^k) - (\lambda - \lambda^k)] \\
+ (x_2^k - \bar{x}_2^k) A_2^T (\lambda - \lambda^k) \\
= (x_2 - \bar{x}_2^k)^T A_2^T [\lambda - \bar{\lambda}^k + (1 - \alpha)(\bar{\lambda}^k - \lambda^k) + \beta A_2(x_2^k - x_2^k)] - (x_2 - \bar{x}_2^k)^T A_2^T (\lambda - \lambda^k) \\
= (x_2 - \bar{x}_2^k)^T A_2^T (\lambda - \lambda^{k+1}) - (x_2 - \bar{x}_2^k)^T A_2^T (\lambda - \lambda^k).
\]
Substituting the above equality into (2.2.20) and using the notation \(G_\alpha\) and \(M_\alpha\), we get (2.2.15) immediately. The proof is completed.

The following result shows the contraction of the sequence generated by the generalized ADMM with LQP regularization (2.1.1)-(2.1.3), based on which its global convergence can be established easily.

**Lemma 2.2.6** Let the sequence \(\{w^k\}\) be generated by the generalized ADMM with LQP regularization (2.1.1)-(2.1.3), \(\alpha \in (0, 2)\), and the accompanying sequence \(\{\bar{w}^k\}\) be defined by (2.2.8). Then for any \(w^* \in W^*\), we have
\[
\|w^{k+1} - w^*\|^2_{G_\alpha} \leq \|w^k - w^*\|^2_{G_\alpha} - \|w^k - \bar{w}^k\|^2_{M_\alpha},
\]
where \(G_\alpha\) and \(M_\alpha\) are defined by (2.2.6) and (2.2.7), respectively.

**Proof:** Setting \(w = w^*\) in (2.2.15), we get
\[
2(w^* - \bar{w}^k)^T \Gamma(\bar{w}^k) \geq \|w^{k+1} - w^*\|^2_{G_\alpha} - \|w^k - w^*\|^2_{G_\alpha} + \|w^k - \bar{w}^k\|^2_{M_\alpha}.
\]
On the other hand, since \(\Gamma\) is monotone, \(\bar{w}^k \in W\), and \(w^* \in W^*\), we have
\[
0 \geq (w^* - \bar{w}^k)^T \Gamma(w^*) \geq (w^* - \bar{w}^k)^T \Gamma(w^k).
\]
From the above two inequalities, the assertion (2.2.21) is proved.

Using the lemmas already proved, we are ready to prove the global convergence for the generalized ADMM with LQP regularization.

**Theorem 2.2.7** The sequence \(\{w^k\}\) generated by the generalized ADMM with LQP regularization (2.1.1)-(2.1.3) converges to some \(w^\infty\) which is a solution of \(\text{VI}^+(W, \Gamma)\).
Proof: It follows from (2.2.21) that \( \{ w^k \} \) is bounded and
\[
\lim_{k \to \infty} \| w^k - \bar{w}^k \|_{M_\alpha} = 0.
\] (2.2.22)
Thus, we have that \( \{ \bar{w}^k \} \) is also bounded and it has at least one cluster point. Let \( w^\infty \) be a cluster point of \( \{ \bar{w}^k \} \) and the subsequences \( \{ \bar{w}^{kJ} \} \) and \( \{ w^{kJ} \} \) both converge to \( w^\infty \). Then it follows from (2.2.6), (2.2.8) and (2.2.12) that
\[
\lim_{k \to \infty} \| w^{k+1} - \bar{w}^k \|_{G_\alpha} = \lim_{k \to \infty} \frac{1}{\alpha \beta} \| \lambda^{k+1} - \bar{\lambda}^k \|^2 = \lim_{k \to \infty} \frac{1}{\alpha \beta} \| (1 - \alpha) (\lambda^k - \bar{\lambda}^k) + \beta A_2 (x^k_2 - \bar{x}^k_2) \|^2 = 0.
\]

Therefore, using (2.2.22) and the above formula, we obtain
\[
\lim_{k \to \infty} \| w^{k+1} - w^k \|_{G_\alpha} = 0.
\]
From (2.2.15), (2.2.22) and the above formula, we get
\[
\liminf_{k \to \infty} (w - \bar{w}^k)^T \Gamma(\bar{w}^k) \geq 0, \quad \forall w \in \mathcal{W}.
\]
Then we have
\[
\liminf_{j \to \infty} (w - \bar{w}^{kJ})^T \Gamma(\bar{w}^{kJ}) \geq 0, \quad \forall w \in \mathcal{W},
\]
and consequently
\[
(w - w^\infty)^T \Gamma(w^\infty) \geq 0, \quad \forall w \in \mathcal{W},
\]
since the mapping \( \Gamma \) is continuous and \( \bar{w}^{kJ} \to w^\infty \) \((j \to \infty)\). This means \( w^\infty \) is a solution point of \( \text{VI}+(\mathcal{W}, \Gamma) \). Note that the inequality (2.2.21) is true for all solution points of \( \text{VI}+(\mathcal{W}, \Gamma) \), hence we have
\[
\| w^{k+1} - w^\infty \|_{G_\alpha} \leq \| w^{l} - w^\infty \|_{G_\alpha}, \quad \forall k \geq 0, \forall l \leq k.
\] (2.2.23)
Since \( w^{kJ} \to w^\infty \) \((j \to \infty)\), for any given \( \varepsilon > 0 \), there exists a \( j_0 > 0 \) such that
\[
\| w^{kJ_0} - w^\infty \|_{G_\alpha} \leq \varepsilon.
\] (2.2.24)
Therefore, for any \( k \geq k_{j_0} \), it follows from (2.2.23) and (2.2.24) that
\[
\| w^{k+1} - w^\infty \|_{G_\alpha} \leq \| w^{kJ_0} - w^\infty \|_{G_\alpha} \leq \varepsilon.
\]
This implies that the sequence \( \{ w^k \} \) converges to a point \( w^\infty \) in \( \mathcal{W}^* \). \( \square \)
2.3 Convergence rate

2.3.1 Convergence rate in an ergodic sense

Now, we establish a worst-case $O(1/t)$ convergence rate in an ergodic sense for the generalized ADMM with LQP regularization.

**Theorem 2.3.1** For any integer $t > 0$, there is a $\tilde{w}_t \in W$ which is a convex combination of the iterates $\bar{w}^0, \bar{w}^1, \ldots, \bar{w}^t$ defined by (2.2.8). Then, for any $w \in W$, we have

$$
(\bar{w}_t - w)^T \Gamma(w) \leq \frac{1}{2(t + 1)} \|w^0 - w\|_{G_\alpha}^2,
$$

(2.3.1)

where $\bar{w}_t := (\sum_{k=0}^{t} \bar{w}^k)/(t + 1)$ and $G_\alpha$ is defined by (2.2.6).

**Proof:** From (2.2.15), we have

$$(w - \bar{w}^k)^T \Gamma(\bar{w}^k) + \frac{1}{2}\|w^k - w\|^2_{G_\alpha} \geq \frac{1}{2}\|w^{k+1} - w\|^2_{G_\alpha}, \quad \forall w \in W. \quad (2.3.2)$$

Since $\Gamma$ is monotone, from the above inequality, we have

$$(w - \bar{w}^k)^T \Gamma(w) + \frac{1}{2}\|w^k - w\|^2_{G_\alpha} \geq \frac{1}{2}\|w^{k+1} - w\|^2_{G_\alpha}, \quad \forall w \in W. \quad (2.3.2)$$

Summing the inequality (2.3.2) over $k = 0, 1, \ldots, t$, we obtain

$$
\left[(t + 1)w - (\sum_{k=0}^{t} \bar{w}^k)\right]^T \Gamma(w) + \frac{1}{2}\|w^0 - w\|^2_{G_\alpha} \geq \frac{1}{2}\|w^{t+1} - w\|^2_{G_\alpha} \geq 0, \quad \forall w \in W.
$$

Since $\sum_{k=0}^{t} 1/(t + 1) = 1$, $\bar{w}_t$ is a convex combination of $\bar{w}^0, \bar{w}^1, \ldots, \bar{w}^t$ and thus $\bar{w}_t \in W$. Using the notation of $\bar{w}_t$, we derive

$$(w - \bar{w}_t)^T \Gamma(w) + \frac{1}{2(t + 1)}\|w^0 - w\|^2_{G_\alpha} \geq 0, \quad \forall w \in W. \quad (2.3.1)$$

The assertion (2.3.1) follows from the above inequality immediately. \qed

**Remark 2.3.2** It follows from Lemma 2.2.6 that the sequence $\{w^k\}$ generated by the generalized ADMM with LQP regularization is bounded. According to (2.2.22), the sequence $\{\bar{w}^k\}$ defined by (2.2.8) is also bounded. Therefore, there exists a constant $D > 0$ such that

$$
\|w^k\|_{G_\alpha} \leq D \quad \text{and} \quad \|\bar{w}^k\|_{G_\alpha} \leq D, \quad \forall k \geq 0.
$$
Recall that $\bar{w}_t$ is the average of $\{\bar{w}_0, \bar{w}_1, \cdots, \bar{w}_t\}$. Thus, we have $\|\bar{w}_t\|_{G_\alpha} \leq D$. Denote $B_W(\bar{w}_t) := \{w \in W \mid \|w - \bar{w}_t\|_{G_\alpha} \leq 1\}.$

For any $w \in B_W(\bar{w}_t)$, we get

\[
(\bar{w}_t - w)^T\Gamma(w) \leq \frac{1}{2(t+1)}\|w^0 - w\|^2_{G_\alpha} \\
\leq \frac{1}{2(t+1)}\left(\|w^0 - \bar{w}_t\|_{G_\alpha} + \|\bar{w}_t - w\|_{G_\alpha}\right)^2 \\
\leq \frac{1}{2(t+1)}\left(\|w^0\|_{G_\alpha} + \|\bar{w}_t\|_{G_\alpha} + \|\bar{w}_t - w\|_{G_\alpha}\right)^2 \\
\leq \frac{(2D + 1)^2}{2(t+1)}.
\]

Thus, for any given $\varepsilon > 0$, after at most $t := \left\lceil \frac{(2D + 1)^2}{2\varepsilon} - 1 \right\rceil$ iterations, we have

\[
(\bar{w}_t - w)^T\Gamma(w) \leq \varepsilon, \quad \forall w \in B_W(\bar{w}_t),
\]

which means $\bar{w}_t$ is an approximate solution of $VI+(W, \Gamma)$ with an accuracy of $O(1/t)$. That is, a worst-case $O(1/t)$ convergence rate of the generalized ADMM with LQP regularization in an ergodic sense is established.

### 2.3.2 Convergence rate in a nonergodic sense

Now, we derive a worst-case $O(1/t)$ convergence rate in a nonergodic sense for the generalized ADMM with LQP regularization.

**Theorem 2.3.3** Let the sequence $\{w^k\}$ be generated by the generalized ADMM with LQP regularization (2.1.1)-(2.1.3), $\alpha \in (0, 2)$, and the accompanying sequence $\{\bar{w}^k\}$ be defined by (2.2.8). Then, for any $w^* \in W^*$ and integer $t \geq 0$, we have

\[
\min_{k \in \{0, \cdots, t\}} \|w^k - \bar{w}^k\|^2_{M_\alpha} \leq \frac{1}{t+1}\|w^0 - w^*\|^2_{G_\alpha}, \tag{2.3.3}
\]

where $G_\alpha$ and $M_\alpha$ are defined by (2.2.6) and (2.2.7), respectively.

**Proof:** It follows from Lemma [2.2.6] that

\[
\|w^k - \bar{w}^k\|^2_{M_\alpha} \leq \|w^k - w^*\|^2_{G_\alpha} - \|w^{k+1} - w^*\|^2_{G_\alpha}, \quad \forall w^* \in W^*, \quad k \geq 0.
\]
Summing the above inequality over \( k = 0, 1, \ldots, t \), we obtain

\[
\sum_{k=0}^{t} \| w^k - \bar{w}^k \|^2_{M_\alpha} \leq \| w^0 - w^* \|^2_{G_\alpha}.
\]

The assertion (2.3.3) follows from the above inequality immediately. \( \square \)

**Remark 2.3.4** It follows from (2.2.15) that

\[
(w - \bar{w}^t)^T \Gamma(\bar{w}^t) \geq \frac{1}{2} (w^{t+1} - w^t)^T G_\alpha (w^{t+1} + w^t - 2w) + \frac{1}{2} \| w^t - \bar{w}^t \|^2_{M_\alpha}.
\]

Using (2.2.8) and (2.2.13), we have

\[
w^{t+1} - w^t = \begin{pmatrix} x_1^{t+1} - x_1^t \\ x_2^{t+1} - x_2^t \\ \lambda^{t+1} - \lambda^t \end{pmatrix} = \begin{pmatrix} \bar{x}_1^t - x_1^t \\ \bar{x}_2^t - x_2^t \\ \alpha (\bar{\lambda}^t - \lambda^t) + \beta (A_2 \bar{x}_2 - A_2 \bar{x}_2^t) \end{pmatrix}.
\]

If \( \| w^t - \bar{w}^t \|_{M_\alpha} = 0 \), from the above two formulae, we have \( \| w^t - w^{t+1} \|_{M_\alpha} = 0 \), and thus

\[
(w - \bar{w}^t)^T \Gamma(\bar{w}^t) \geq 0, \quad \forall w \in \mathcal{W},
\]

which means that \( \bar{w}^t \) is a solution of VI\((\mathcal{W}, \Gamma)\) according to (2.2.1). Therefore, \( \| w^t - \bar{w}^t \|_{M_\alpha} \) can be viewed as an error measurement in term of the distance to the solution set of VI\((\mathcal{W}, \Gamma)\) for the \((t + 1)\)-th iteration of the scheme (2.1.1)-(2.1.3).

It is thus reasonable to seek an upper bound of \( \min_{k \in \{0, \ldots, t\}} \| w^k - \bar{w}^k \|^2_{M_\alpha} \) in term of the quantity \( O(1/t) \) for the purpose of investigating the convergence rate of the generalized ADMM with LQP regularization (2.1.1)-(2.1.3). Note that \( \mathcal{W}^* \) is convex and closed.

Let \( d := \inf \{ \| w^0 - w^* \|_{G_\alpha} \mid w^* \in \mathcal{W}^* \} \). Then, for any given \( \varepsilon > 0 \), Theorem 2.3.3 shows that the generalized ADMM with LQP regularization (2.1.1)-(2.1.3) needs at most \( \lceil d^2 / \varepsilon - 1 \rceil \) iterations to ensure that \( \min_{k \in \{0, \ldots, t\}} \| w^k - \bar{w}^k \|^2_{M_\alpha} \leq \varepsilon \). A worst-case \( O(1/t) \) convergence rate in a nonergodic sense for the generalized ADMM with LQP regularization (2.1.1)-(2.1.3) is thus established by Theorem 2.3.3.
Chapter 3

A proximal strictly contractive PRSM for \((P2)\)

In this chapter, we focus on the model \((P2)\) and discuss a proximal version of the SC-PRSM (1.4.13) where one of its subproblems at each iteration is regularized by a proximal regularization term. With the proximal matrix chosen properly, the regularized subproblem could be much easier than the original one. This feature of the proposed method is particularly efficient for some sparse and low-rank optimization models — we verified it by some applications arising in the area of image processing. In addition to its significant importance in numerical implementation, we also show that the proximal version of SC-PRSM enjoys the same theoretical results as the original SC-PRSM, e.g., the global convergence and the measurable worst-case convergence rate estimated by the iteration complexity. Thus, the proposed method can be regarded as a specific version of the prototype SC-PRSM algorithm in [82] that is potentially useful for a range of application arising in different areas such as image processing, machine learning, computer vision, etc.

3.1 Algorithm

Inspired by the same reason as linearized ADMM (1.4.11), it is necessary to discuss how to solve the subproblems in (1.4.13) more efficiently for some interesting concrete
applications of the abstract model (P2). Like the linearized ADMM (1.4.11), our main purpose is for the scenario when $\theta_1(x_1)$ is special (e.g., the resolvent operator of $\partial \theta_1$ has a closed-form representation) and it deserves to exploit its properties effectively in algorithmic design. We thus propose the following proximal version of the SC-PRSM (1.4.13):

$$
\begin{align*}
    x_1^{k+1} &= \operatorname{arg min} \left\{ \theta_1(x_1) + \frac{\alpha}{2} \| A_1 x_1 + A_2 x_2^k - b - \frac{1}{\beta} \lambda^k \|_G^2 + \frac{1}{2} \| x_1 - x_1^k \|_G^2 \mid x_1 \in X_1 \right\}, \\
    \lambda^{k+\frac{1}{2}} &= \lambda^k - r \beta (A_1 x_1^{k+1} + A_2 x_2^k - b), \\
    x_2^{k+1} &= \operatorname{arg min} \left\{ \theta_2(x_2) + \frac{\alpha}{2} \| A_1 x_1^{k+1} + A_2 x_2 - b - \frac{1}{\beta} \lambda^{k+\frac{1}{2}} \|_G^2 \mid x_2 \in X_2 \right\}, \\
    \lambda^{k+1} &= \lambda^{k+\frac{1}{2}} - s \beta (A_1 x_1^{k+1} + A_2 x_2^{k+1} - b),
\end{align*}
$$

(3.1.1)

where $G \in \mathbb{R}^{n_1 \times n_1}$ is a positive semi-definite matrix; $r$ and $s$ are scalars in $[0, 1]$. Note that this scheme differs from the well-known predictor corrector proximal multiplier method in [35].

Regarding the iterative scheme (3.1.1), we would make some explanations.

- First, the most interesting choice for $G$ is still $G = \mu \cdot I_{n_1 \times n_1} - \beta A_1^T A_1$ with the requirement $\mu \geq \beta \| A_1^T A_1 \|$. With this choice, the quadratic term $\frac{\alpha}{2} \| A_1 x_1 + A_2 x_2^k - b - \frac{1}{\beta} \lambda^k \|_G^2$ is linearized and the linearized $x_1$-subproblem in (3.1.1) reduces to estimating the resolvent operator of $\partial \theta_1(x_1)$. This means the linearized $x_1$-subproblem has a closed-form solution if the resolvent operator of $\partial \theta_1(x_1)$ has a closed-form representation. Moreover, we only require the positive semidefiniteness of $G$ and abuse slightly the notation $\| x_1 \|_G^2 := x_1^T G x_1$. This notation makes the scheme (3.1.1) include more existing methods as special cases.

- Second, we introduce two different factors $r$ and $s$ for different Lagrange-multiplier-updating steps in (3.1.1), rather than one as the original SC-PRSM (1.4.13), and slightly extend their ranges to $[0, 1]$ from $(0, 1)$. This consideration provides us more freedom in parameter-tuning and thus potentially leads to better numerical performance; we refer to [82] for the importance of the relaxation factor in accelerating SC-PRSM numerically. Nevertheless, with the more relaxed choice of the factors $r$ and $s$, some existing methods are included.
as special cases of the scheme (3.1.1) — it reduces to the ADMM (1.4.7) with \( r = 0, s = 1 \) and \( G = 0_{n_1 \times n_1} \); the linearized ADMM (1.4.11) with \( r = 0, s = 1 \) and \( G = \mu \cdot I_{n_1 \times n_1} - \beta A_1^T A_1 \); the generalized ADMM in [53] with \( r = \alpha - 1 \) with \( \alpha \in (1, 2), s = 1 \) and \( G = 0_{n_1 \times n_1} \); the original PRSM (1.4.12) with \( r = s = 1 \) and \( G = 0_{n_1 \times n_1} \); the SC-PRSM (1.4.13) with \( r = s \in (0, 1) \) and \( G = 0_{n_1 \times n_1} \). Note that the scheme (3.1.1) reduces to the alternating minimization scheme [145] widely used in image processing with \( r = s = 0 \) and \( G = 0_{n_1 \times n_1} \); but under general convex settings there is no affirmative convergence for this case. We thus do not include the case where \( r = s = 0 \) in our discussion. In other words, we always assume \( r + s > 0 \) in upcoming analysis.

### 3.2 Global convergence

In this section, we prove the global convergence for the scheme (3.1.1). This can be done essentially by proving a strict contraction property of the sequence \( \{w^k\} \) generated by the iterative scheme (3.1.1). Our convergence rate analysis for the scheme (3.1.1) is also crucially based on the contraction property of its iterative sequence.

First recall that in Section 1.4.1, we give a variational characterization of model (P2), by showing that the problem (P2) amounts to the variational inequality VI(\( \Omega_2, F_2, \theta \)) (1.4.3)-(1.4.5).

The following theorem provides a characterization of the solution set \( \Omega_2^* \), which is very useful for the convergence rate analysis for the proposed scheme (3.1.1).

**Theorem 3.2.1** Let \( \Omega_2^* \) be the set of \( w^* \) satisfying VI(\( \Omega_2, F_2, \theta \)) (1.4.3)-(1.4.5). Then \( \Omega_2^* \) is a closed convex set, and it can be characterized as

\[
\Omega_2^* = \bigcap_{w \in \Omega_2} \{ \tilde{w} \in \Omega_2 \mid \theta(u) - \theta(\tilde{u}) + (w - \tilde{w})^T F_2(w) \geq 0 \}. \tag{3.2.1}
\]

**Proof:** See [58 Theorem 2.3.5] or [88].

Theorem 3.2.1 indicates that \( \tilde{w} \in \Omega_2 \) is an approximate solution of VI(\( \Omega_2, F_2, \theta \)) with
an accuracy of $O(1/t)$ if it satisfies

$$\theta(\tilde{w}) - \theta(u) + (\tilde{w} - w)^T F_2(w) \leq \epsilon, \ \forall w \in \Omega_2$$

with $\epsilon = O(1/t)$. This characterization makes it possible to analyze the convergence rate of some splitting methods by the variational inequality approach instead of the conventional approach based on the estimation of functional values.

In order to present our analysis in a compact notation, we introduce several matrices

$$H := \begin{pmatrix} G & 0 & 0 \\ 0 & \frac{(r+s-rs)\beta}{r+s} A_2^T A_2 & -\frac{r-s}{r+s} A_2^T \\ 0 & -\frac{r}{r+s} A_2 & \frac{1}{\beta(r+s)} I_m \end{pmatrix}, \quad M := \begin{pmatrix} I_{n_1} & 0 & 0 \\ 0 & I_{n_2} & 0 \\ 0 & -s\beta A_2 & (r+s)I_m \end{pmatrix},$$

$$Q := \begin{pmatrix} G & 0 & 0 \\ 0 & \beta A_2^T A_2 & -rA_2^T \\ 0 & -A_2 & \frac{1}{\beta} I_m \end{pmatrix},$$

where $r \in [0, 1]$, $s \in [0, 1]$, $r+s > 0$ and $G \succeq 0$ are assumed throughout this chapter.

These matrices will help us present our analysis in succinct notation. For the matrices defined above, the following lemma summarizes some relationships among them. We need these properties in our analysis later.

**Lemma 3.2.2** If $r \in [0, 1]$, $s \in [0, 1]$, $r+s > 0$ and $G \succeq 0$, then the matrices $H$, $M$ and $Q$ defined in (3.2.3) satisfy

(i) $HM = Q$;

(ii) $H \succeq 0$;

(iii) $Q^T + Q - M^T HM \succeq \frac{1-s}{r+s+2} M^T HM$.

**Proof:** Item (i) holds evidently. As for Item (ii), for any $w = (x_1, x_2, \lambda)$,

$$w^T H w = ||x_1||_2^2 + \beta \frac{r+s-rs}{r+s} ||A_2 x_2||^2 + \frac{1}{\beta(r+s)} ||\lambda||^2 - \frac{2r}{r+s} x_2^T A_2^T \lambda$$

$$= ||x_1||_2^2 + \frac{\beta}{r+s} (r+s-rs) ||A_2 x_2||^2 + \frac{1}{\beta^2} ||\lambda||^2 - \frac{2r}{\beta} x_2^T A_2^T \lambda$$

$$= ||x_1||_2^2 + \frac{\beta}{r+s} (r+s)(1-r) ||A_2 x_2||^2 + ||rA_2 x_2 - \frac{1}{\beta} \lambda||^2$$

$$\geq 0.$$
Therefore, $H$ is positive semi-definite for $r, s \in [0, 1]$. Pertaining to Item (iii), we can deduce that

\[(r + s + 2)(Q^T + Q - M^T H M) - (1 - s)M^T Q = \begin{pmatrix} (r + 2s + 1)G & 0 & 0 \\ 0 & (1 - s)(r + 1)\beta A_2^T A_2 & -2(1 - s)A_2^T \\ 0 & -2(1 - s)A_2 & 4-(r+s)(r+1)I_m \end{pmatrix}.\]

Since $r \in [0, 1]$, $s \in [0, 1]$ and $r + s > 0$, we have the submatrix in the right-hand side of (3.2.4)

\[\begin{pmatrix} (1 - s)(r + 1)\beta A_2^T A_2 & -2(1 - s)A_2^T \\ -2(1 - s)A_2 & 4-(r+s)(r+1)I_m \end{pmatrix} \succeq 0,
\]

which, together with $G \succeq 0$, indicates that $(r + s + 2)(Q^T + Q - M^T H M) - (1 - s)M^T Q \succeq 0$; or equivalently

\[Q^T + Q - M^T H M \succeq \frac{1 - s}{r + s + 2} M^T H M.\]

The proof is complete. \[\Box\]

**Remark 3.2.3** Although $H$ and $Q^T + Q - M^T H M$ are positive semi-definite as $r, s \in [0, 1]$, we use the notations:

\[\|w\|^2_H = w^T H w; \|w\|^2_{Q^T + Q - M^T H M} = w^T (Q^T + Q - M^T H M) w, \forall w \in \Omega_2.\]

For the notational simplification, we first need to define an auxiliary sequence $\{\tilde{w}^k\}$ as follows:

\[\tilde{w}^k = \begin{pmatrix} \tilde{x}_1^k := x_1^{k+1} \\ \tilde{x}_2^k := x_2^{k+1} \\ \tilde{\lambda}^k := \lambda^k - \beta(A_1x_1^{k+1} + A_2x_2^k - b) \end{pmatrix}, \tag{3.2.5}\]

where the sequence $\{(x_1^k, x_2^k, \lambda^k)\}$ is generated by the scheme (3.1.1). Then, it follows from the iterative scheme (3.1.1) and the notation (3.2.5) that

\[\lambda^{k+\frac{1}{2}} = \lambda^k - r(\lambda^k - \tilde{\lambda}^k), \tag{3.2.6}\]
and

\[
\lambda^{k+1} = \lambda^{k+\frac{1}{2}} - s\beta(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k - b) \tag{3.2.7}
\]
\[
= \lambda^k - r(\lambda^k - \tilde{\lambda}^k) - s(\beta(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k - b) - \beta A_2(x_2^k - \tilde{x}_2^k))
\]
\[
= \lambda^k - ((r + s)(\lambda^k - \tilde{\lambda}^k) - s\beta A_2(x_2^k - \tilde{x}_2^k)).
\]

Consequently, substituting these equations into (3.1.1), we have the following expressions:

\[
\begin{align*}
\tilde{x}_1^k &= \arg\min \left\{ \theta_1(x_1) + \frac{s}{2}\|A_1x_1 + A_2x_2 - b - \frac{\lambda^k}{\beta}\|^2 + \frac{1}{2}\|x_1 - x_1^k\|^2 \mid x_1 \in X_1 \right\}; \\
\tilde{x}_2^k &= \arg\min \left\{ \theta_2(x_2) + \frac{s}{2}\|A_1\tilde{x}_1^k + A_2x_2 - b - \frac{1}{\beta}[\lambda^k - r(\lambda^k - \tilde{\lambda}^k)][\|\lambda^k - \tilde{\lambda}^k]\|^2 \mid x_2 \in X_1 \right\}.
\end{align*}
\tag{3.2.8}
\]

The following relationship which is an immediate result from (3.2.5) and (3.2.7) is very useful in our analysis:

\[
\begin{pmatrix}
\begin{pmatrix} x_1^{k+1} \\ x_2^{k+1} \\ \lambda^{k+1} \end{pmatrix} \\
\begin{pmatrix} x_1^k \\ x_2^k \\ \lambda^k \end{pmatrix}
\end{pmatrix} =
\begin{pmatrix}
I_{n_1} & 0 & 0 \\
0 & I_{n_2} & 0 \\
0 & -s\beta A_2 & (r + s)I_m
\end{pmatrix}
\begin{pmatrix}
\begin{pmatrix} x_1^{k-1} \\ x_2^{k-1} \\ \lambda^{k-1} \end{pmatrix} \\
\begin{pmatrix} x_1^k \\ x_2^k \\ \lambda^k \end{pmatrix}
\end{pmatrix}.
\]

With the matrix \(M\) defined in (3.2.3), the above relationship can be written in a compact form:

\[
w^{k+1} = w^k - M(w^k - \tilde{w}^k). \tag{3.2.9}
\]

We will use the equation (3.2.9) frequently in our analysis.

Because of the characterization (3.2.2) of \(\Omega_2^s\), we are interested in finding an upper bound of \(\theta(\tilde{u}) - \theta(u) + (\tilde{w} - w)^TF_2(w)\) \((\forall w \in \Omega_2)\) for the \(k\)-th iterate and then refining this upper bound into a term that is suitable for recursive operation. With such a bound, it is possible to estimate the worst-case convergence rate measured by the iteration complexity for (P2). We first prove two lemmas.

**Lemma 3.2.4** For given \(w^k \in \Omega_2\), let \(w^{k+1}\) be generated by the iterative scheme (3.1.1) and \(\tilde{w}^k\) be defined in (3.2.5). Then, it holds that \(\tilde{w}^k \in \Omega_2\) and

\[
\theta(\tilde{w}^k) - \theta(u) + (\tilde{w}^k - w)^TF_2(\tilde{w}^k) \leq -(w - \tilde{w}^k)^TQ(w^k - \tilde{w}^k), \quad \forall w \in \Omega_2, \tag{3.2.10}
\]

where the matrix \(Q\) is defined in (3.2.3).
Proof: The first-order optimality condition of the $x_1$-subproblem in (3.2.8) and the fact $x_1^{k+1} = \tilde{x}_1^k$ in (3.2.5) mean
\[ \theta_1(x_1) - \theta_1(\tilde{x}_1^k) + (x_1 - \tilde{x}_1^k)^T\{ A_1^T(\beta(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k - b) - \lambda^k) + G(\tilde{x}_1^k - x_1^k)\} \geq 0, \quad \forall x_1 \in \mathcal{X}_1. \] (3.2.11)

Note the definition of $\tilde{\lambda}^k$ in (3.2.5). We thus have
\[ \theta_1(x_1) - \theta_1(\tilde{x}_1^k) + (x_1 - \tilde{x}_1^k)^T\{ -A_1^T\tilde{\lambda}^k + G(\tilde{x}_1^k - x_1^k)\} \geq 0, \quad \forall x_1 \in \mathcal{X}_1, \quad (3.2.12) \]
and also have
\[ \beta(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k - b) - \beta A_2(\tilde{x}_2^k - x_2^k) + (\tilde{\lambda}^k - \lambda^k) = 0. \] (3.2.13)

Analogously, it follows from the first-order optimality condition of the $x_2$-subproblem in (3.2.8) that
\[ \theta_2(x_2) - \theta_2(\tilde{x}_2^k) + (x_2 - \tilde{x}_2^k)^T\{ A_2^T\{ \beta(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k - b) - (\lambda^k - r(\lambda^k - \tilde{\lambda}^k))\} \geq 0, \quad \forall x_2 \in \mathcal{X}_2. \] (3.2.14)

Because of (3.2.13), we have
\[ \beta(A_1\tilde{x}_1^k + A_2\tilde{x}_2^k - b) - (\lambda^k - r(\lambda^k - \tilde{\lambda}^k)) = -(\tilde{\lambda}^k - \lambda^k) + \beta A_2(\tilde{x}_2^k - x_2^k) - (\lambda^k - r(\lambda^k - \tilde{\lambda}^k)) = -\tilde{\lambda}^k + \beta A_2(\tilde{x}_2^k - x_2^k) - r(\lambda^k - \tilde{\lambda}^k). \]

Thus, (3.2.14) implies
\[ \theta_2(x_2) - \theta_2(\tilde{x}_2^k) + (x_2 - \tilde{x}_2^k)^T\{ -A_2^T\tilde{\lambda}^k + \beta A_2^T A_2(\tilde{x}_2^k - x_2^k) - r A_2^T (\tilde{\lambda}^k - \lambda^k) \} \geq 0, \quad \forall x_2 \in \mathcal{X}_2. \] (3.2.15)

Now, combining (3.2.12), (3.2.13) and (3.2.15) together, we have $\tilde{u}^k = (\tilde{x}_1^k, \tilde{x}_2^k, \tilde{\lambda}^k) \in \Omega_2$ and
\[
\begin{align*}
\theta(u) - \theta(\tilde{u}^k) + & \left( x_1 - \tilde{x}_1^k \right)^T \left( \begin{array}{c}
-A_1^T \tilde{\lambda}^k \\
-A_2^T \tilde{\lambda}^k \\
-A_1^T \tilde{\lambda}^k \\
A_1 \tilde{x}_1^k + A_2 \tilde{x}_2^k - b \\
G(\tilde{x}_1^k - x_1^k) \\
\beta A_2 (\tilde{x}_2^k - x_2^k) - r A_2^T (\tilde{\lambda}^k - \lambda^k) \\
-A_2 (\tilde{x}_2^k - x_2^k) + \frac{1}{\beta} (\tilde{\lambda}^k - \lambda^k)
\end{array} \right) \geq 0, \quad \forall \omega \in \Omega_2.
\end{align*}
\]
Recall the definitions of $w$ and $F_2$ in (3.2.8); and $Q$ in (3.2.9). The above inequality can be written as (3.2.10). The proof is complete. 

In the next lemma, we further refine the bound $(w - \tilde{w}^k)^TQ(w^k - \tilde{w}^k)$ found in (3.2.10) into the sum of some quadratic terms. This refined bound consisting of quadratic terms is convenient for the manipulation over the whole sequence $\{w^k\}$ recursively and thus for establishing the convergence rate of $\{w^k\}$.

**Lemma 3.2.5** Let $\{w^k\}$ be the sequence generated by the scheme (3.1.1) and the sequence $\{\tilde{w}^k\}$ be defined in (3.2.5), then $\tilde{w}^k \in \Omega$ and

$$
\theta(u) - \theta(\tilde{w}^k) + (w - \tilde{w}^k)^TF_2(w) \geq \frac{1}{2}\left(\|w - w^{k+1}\|_H^2 - \|w - w^k\|_H^2\right)
+ \frac{1}{2}\|w^k - \tilde{w}^k\|_{Q+Q-MTHM}^2, \quad \forall w \in \Omega,
$$

(3.2.16)

where the matrices $H$, $M$ and $Q$ are defined in (3.2.3).

**Proof:** It follows from the Item $(i)$ in Lemma 3.2.2 and (3.2.9) that

$$(w - \tilde{w}^k)^TQ(w^k - \tilde{w}^k) = (w - \tilde{w}^k)^TH(w^k - w^{k+1}), \quad \forall w \in \Omega.
$$

(3.2.17)

Setting $a := w$, $b := \tilde{w}^k$, $c := w^k$ and $d := w^{k+1}$ in the following identity

$$
2(a - b)^TH(c - d) = \|a - d\|_H^2 - \|a - c\|_H^2 + \|c - b\|_H^2 - \|d - b\|_H^2,
$$

we obtain

$$
2(w - \tilde{w}^k)^TQ(w^k - \tilde{w}^k) = \|w - w^{k+1}\|_H^2 - \|w - w^k\|_H^2 + \|w^k - \tilde{w}^k\|_H^2 - \|w^{k+1} - \tilde{w}^k\|_H^2,
$$

(3.2.18)

where the equality is due to Item $(i)$ of Lemma 3.2.2 and (3.2.9). We thus have

$$
\|w^k - \tilde{w}^k\|_H^2 - \|w^{k+1} - \tilde{w}^k\|_H^2 = \|w^k - \tilde{w}^k\|_H^2 - \|(w^k - \tilde{w}^k) - (w^k - w^{k+1})\|_H^2
= \|w^k - \tilde{w}^k\|_H^2 - \|(w^k - \tilde{w}^k) - M(w^k - \tilde{w}^k)\|_H^2
= 2(w^k - \tilde{w}^k)^THM(w^k - \tilde{w}^k) - (w^k - \tilde{w}^k)^TM^THM(w^k - \tilde{w}^k)
= (w^k - \tilde{w}^k)^T(Q^T + Q - MTM)(w^k - \tilde{w}^k).
$$

43
where the second equality is because of (3.2.9) and the last equality comes from Item (i) of Lemma 3.2.2. Substituting the above equality into (3.2.18), we obtain the equation

\[ 2(w - \tilde{w}^k)^T Q(w - w^{k+1})_H^2 = \| w - w^k \|_H^2 + \| w^k - \tilde{w}^k \|_Q^2 + \| w^k - \tilde{w}^k \|_Q^2 + Q^T M \nabla H, \quad \forall w \in \Omega. \]  

(3.2.19)

From the monotonicity of \( F_2(w) \), we have

\[ (w - \tilde{w}^k)^T (F_2(w) - F_2(\tilde{w}^k)) \geq 0, \quad \forall w \in \Omega. \]

Then, the assertion (3.2.16) is an immediate conclusion from (3.2.10) and (3.2.19).

The proof is complete.

Before we prove the contractiveness of the sequence \( \{w^k\} \), let us first prove one more lemma.

**Lemma 3.2.6** Let the sequence \( \{w^k\} \) be generated by (3.1.1) with \( s = 1 \). Then

\[ (x^2_k - x^2_{k+1})^T A^T_2 (\lambda^k - \lambda^{k+1}) \geq 0. \]  

(3.2.20)

**Proof:** It follows from the optimal condition of the \( x_2 \)-subproblem of (3.1.1) that

\[ \theta_2(x_2) - \theta_2(x_2^{k+1}) + (x_2 - x_2^{k+1})^T \beta A^T_2 (A_1 x_1^{k+1} + A_2 x_2^{k+1} - b - \frac{1}{\beta} \lambda^{k+1}) \geq 0, \quad \forall x_2 \in X_2. \]  

(3.2.21)

Since \( \lambda^{k+1} = \lambda^{k+\frac{1}{2}} - \beta (A_1 x_1^{k+1} + A_2 x_2^{k+1} - b) \), we can rewrite (3.2.21) as

\[ \theta_2(x_2) - \theta_2(x_2^{k+1}) - (x_2 - x_2^{k+1})^T A^T_2 \lambda^{k+1} \geq 0, \quad \forall x_2 \in X_2. \]  

(3.2.22)

Analogous to (3.2.22), we have

\[ \theta_2(x_2) - \theta_2(x_2^{k+1}) - (x_2 - x_2^{k+1})^T A^T_2 \lambda^k \geq 0, \quad \forall x_2 \in X_2. \]  

(3.2.23)

Setting \( x_2 := x_2^k \) in (3.2.22), \( x_2 := x_2^{k+1} \) in (3.2.23), and adding them together, we get

\[ (x_2^k - x_2^{k+1})^T A^T_2 (\lambda^k - \lambda^{k+1}) \geq 0. \]

The proof is complete.

Now we start to prove that the sequence \( \{w^k\} \) generated by (3.1.1) is strictly contractive with respect to the solution set \( \Omega^*_2 \).
Theorem 3.2.7 Let \( \{w^k\} \) be the sequence generated by (3.1.1) with \( r, s \in [0, 1] \) and \( 0 < r + s < 2 \). Then, we have the following conclusions.

1) The sequence \( \{w^k\} \) is strictly contractive with respect to the solution set \( \Omega^* \):
\[
\|w^{k+1} - w^*\|_H^2 \leq \|w^k - w^*\|_H^2 - \|w^k - \tilde{w}^k\|_Q^{T+Q-M^T_H}.
\] (3.2.24)

2) The sequence \( \{w^k\} \) is bounded.

3) We have
\[
\sum_{k=1}^{\infty} \|w^k - w^{k+1}\|_H^2 < \infty,
\] (3.2.25)
where the matrix \( H \) is defined in (3.2.3).

Proof: To prove the first assertion, we set \( w := w^* \) in (3.2.16) and get
\[
\theta(w^*) - \theta(\tilde{w}^k) + (w^* - \tilde{w}^k)^T F_2(w^*) 
\geq \frac{1}{2} \left( \|w^* - w^{k+1}\|_H^2 - \|w^* - w^k\|_H^2 \right) + \frac{1}{2} \|w^k - \tilde{w}^k\|_Q^{T+Q-M^T_H}.
\]
By (3.4.3), the above inequality means
\[
\|w^{k+1} - w^*\|_H^2 \leq \|w^k - w^*\|_H^2 - \|w^k - \tilde{w}^k\|_Q^{T+Q-M^T_H}.
\]
For the second assertion, we first note that (3.2.24) implies the boundedness of \( \{Hw^k\} \) since \( H \succeq 0 \). Then we immediately obtain that the sequences \( \{Gx^k\} \) and \( \{Su^k\} \) are also bounded, where
\[
S = \begin{pmatrix}
(1 - \frac{r s}{r + s}) \beta A_2^T A_2 & -\frac{r}{r + s} A_2^T \\
-\frac{r}{r + s} A_2 & \frac{1}{\beta(r + s)} I_m
\end{pmatrix}
\] (3.2.26)
is a submatrix of \( H \). Since \( r, s \in [0, 1], 0 < r + s < 2 \) and \( A_2 \) is assumed to be full column rank, we have \( S \succeq 0 \). Thus the sequences \( \{x^k_2\} \) and \( \{\lambda^k\} \) are both bounded. From the scheme (3.1.1), it is easy to obtain
\[
(r + s) \beta A_1 x_1^{k+1} = \lambda^k - \lambda^{k+1} - r \beta A_2 x_2^k - s \beta A_2 x_2^{k+1} + (r + s) \beta b,
\] (3.2.27)
which implies the boundedness of \( \{A_1 x_1^k\} \). Since \( A_1 \) is also assumed to be full column rank, we have that \( \{x_1^k\} \) is bounded.

To prove the assertion (3.2.25), we discuss two cases: (a) \( s = 1 \) and (b) \( s \neq 1 \).
(a) When $s = 1$, it follows from \((3.2.24)\) that
\[
\sum_{k=1}^{\infty} \| w^k - \tilde{w}^k \|_{Q^T + Q - M^T H M}^2 < \infty, \tag{3.2.28}
\]
and
\[
\| w^k - \tilde{w}^k \|_{Q^T + Q - M^T H M}^2 = \| x^k - x^{k+1} \|_G^2 + \frac{1 - r}{\beta} \| \lambda^k - \tilde{\lambda}^k \|_2^2
\]
\[
= \| x^k - x^{k+1} \|_G^2 + \frac{1 - r}{\beta(r + 1)^2} \| \lambda^k - \lambda^{k+1} + \beta B(y^k - y^{k+1}) \|_2^2,
\]
where the first equality is because of \((3.2.3)\) and the second equality comes from \((3.2.7)\). Substituting the above equality into \((3.2.28)\), we immediately get
\[
\sum_{k=1}^{\infty} \| x^{k+1} - x^k \|_G^2 < \infty, \tag{3.2.29}
\]
and
\[
\frac{1 - r}{\beta(r + 1)^2} \sum_{k=1}^{\infty} \| \lambda^k - \lambda^{k+1} + \beta A_2(x^k - x^{k+1}) \|_2^2 < \infty. \tag{3.2.30}
\]
Denote $\rho = \frac{1 - r}{\beta(r + 1)^2}$. Note that $\rho > 0$ since $0 < r < 1$. From \((3.2.30)\), we have
\[
\sum_{k=1}^{\infty} \| \lambda^k - \lambda^{k+1} \|_{2I_m}^2 + \| x_2^k - x_2^{k+1} \|_{2\beta^2 B^T B}^2 + 2\rho \beta (x^k_2 - x^{k+1}_2)^T A_2^T (\lambda^k - \lambda^{k+1}) < \infty.
\]
It follows from the above inequality and \((3.2.20)\) that
\[
\sum_{k=1}^{\infty} \| \lambda^k - \lambda^{k+1} \|_2^2 < \infty, \quad \text{and} \quad \sum_{k=1}^{\infty} \| x_2^k - x_2^{k+1} \|_{2 \beta A_2^T A_2}^2 < \infty. \tag{3.2.31}
\]
Combining \((3.2.20)\), \((3.2.29)\) and \((3.2.31)\), the assertion \((3.2.25)\) holds.

(b) When $s \neq 1$, by \((3.2.24)\) and Item (iii) in \((3.2.2)\), it holds that
\[
\| w^{k+1} - w^* \|_H^2 \leq \| w^k - w^* \|_H^2 - \frac{1 - s}{r + s + 2} \| w^k - w^{k+1} \|_H^2, \ \forall w^* \in \Omega,
\]
which immediately implies \((3.2.25)\).

We complete the proof since in both cases the assertion \((3.2.25)\) holds. \qed

Theorem 3.2.7 is useful for proving the convergence and establishing the worst-case convergence rate in a non-ergodic sense for the scheme \((3.1.1)\), as shown in the following theorem.
Theorem 3.2.8 The sequence \( \{w^k = (x_1^k, x_2^k, \lambda^k)\} \) generated by the scheme (3.1.1) converges to a KKT point of \( \{P\} \) with \( r, s \in [0, 1] \) and \( 0 < r + s < 2 \).

Proof: Since the sequence \( \{w^k\} \) is bounded, there must exist a cluster point \( w^\infty \) and we denote the subsequence which converges to \( w^\infty \) by \( \{w^k_{j}\} \). Recall (3.2.5). We have \( \tilde{x}_{1}^{k_{j}} \to x^\infty_{1} \) and \( \tilde{x}_{2}^{k_{j}} \to x^\infty_{2} \). Moreover, from (3.2.25), we know \( \|w^k - w^{k+1}\|_{H} \to 0 \). Therefore, \( \lambda^k - \lambda^{k+1} \to 0 \) and \( x_2^k - x_2^{k+1} \to 0 \) due to the positive definiteness of \( S \) (3.2.26). By taking the limit over \( k_{j} \to \infty \) in (3.2.24), we get \( \tilde{\lambda}^{k_{j}} \to \lambda^\infty \). Thus, we conclude that \( w^\infty \) is also a cluster point of the sequence \( \{\tilde{w}^k\} \). Substituting (3.2.17) into (3.2.10), we have

\[
\theta(u) - \theta(\tilde{w}^k) + (w - \tilde{w}^k)^T F_2(\tilde{w}^k) \geq (w - \tilde{w}^k)^T H(w^k - w^{k+1}), \quad \forall w \in \Omega. \tag{3.2.32}
\]

Note that \( H(w^k - w^{k+1}) \to 0 \) for any \( k \to \infty \). By taking the limit over \( k_{j} \to \infty \) in (3.2.32), we have that the cluster point \( w^\infty \) is also a solution point of VI(\( \Omega, F, \theta \)). Therefore, the inequality (3.2.24) is also valid if \( w^* \) is replaced by \( w^\infty \). Thus, \( \{\|w^k - w^\infty\|_{H}\} \) is a monotonically non-increasing sequence and we further have

\[
\lim_{k \to \infty} H(w^k - w^\infty) = 0,
\]

since \( w^\infty \) is a cluster point of \( \{w^k\} \). By the definition of \( H \) and the positive definiteness of \( S \), it is not difficult to show

\[
\lim_{k \to \infty} x_2^k = x_2^\infty, \quad \lim_{k \to \infty} \lambda^k = \lambda^\infty.
\]

By (3.2.27), using the fact \( b = A_1 x_1^\infty + A_2 x_2^\infty \), we know that

\[
\lim_{k \to \infty} A_1 x_1^k = A_1 x_1^\infty.
\]

Since \( A_1 \) is full column rank, we have

\[
\lim_{k \to \infty} x_1^k = x_1^\infty.
\]

The global convergence of scheme (3.1.1) is thus established. \( \square \)
3.3 Convergence rate

3.3.1 Convergence rate in an ergodic sense

In this section, we establish a worst-case $O(1/t)$ convergence rate in an ergodic sense for the scheme (3.1.1). That is, we show that we can find an approximate solution of $VI(\Omega_2, F_2, \theta)$ with an accuracy of $O(1/t)$ based on the first $t$ iterations of the scheme (3.1.1). The result is summarized in the following theorem.

**Theorem 3.3.1** Let $\{w^k\}$ be the sequence generated by the scheme (3.1.1) and $\{\tilde{w}^k\}$ be the sequence defined in (3.2.5). Let $\tilde{w}_t$ be defined as

$$\tilde{w}_t := \frac{1}{t+1} \sum_{k=0}^{t} \tilde{w}^k. \quad (3.3.1)$$

Then, for any integer $t > 0$, it holds that $\tilde{w}_t \in \Omega_2$ and

$$\theta(\tilde{u}_t) - \theta(u) + (\tilde{w}_t - w)^T F_2(w) \leq \frac{1}{2(t+1)} \|w - w^0\|_H^2, \quad \forall w \in \Omega_2. \quad (3.3.2)$$

**Proof:** It follows from (3.2.5) that $\tilde{w}^k \in \Omega_2$ for all $k \geq 0$. By the convexity of $\mathcal{X}_1$ and $\mathcal{X}_2$, (3.3.1) implies $\tilde{w}_t \in \Omega_2$. Summing (3.2.16) over $k = 0, 1, \ldots, t$, it yields

$$(t+1) \theta(u) - \sum_{k=0}^{t} \theta(\tilde{u}^k) + \left( (t+1)w - \sum_{k=0}^{t} \tilde{w}^k \right)^T F_2(w) + \frac{1}{2} \|w - w^0\|_H^2 \geq 0, \quad \forall w \in \Omega_2. \quad (3.3.3)$$

From the definition of $\tilde{w}_t$, we have

$$\frac{1}{t+1} \sum_{k=0}^{t} \theta(\tilde{u}^k) - \theta(u) + (\tilde{w}_t - w)^T F_2(w) \leq \frac{1}{2(t+1)} \|w - w^0\|_H^2, \quad \forall w \in \Omega_2. \quad (3.3.3)$$

Since $\theta(u)$ is convex and $\tilde{u}_t = \frac{1}{t+1} \sum_{k=0}^{t} \tilde{u}^k$, we have

$$\theta(\tilde{u}_t) \leq \frac{1}{t+1} \sum_{k=0}^{t} \theta(\tilde{u}^k).$$

Substituting the above inequality into (3.3.3), the proof is complete. \qed

Let $w^0 = (x_1^0, x_2^0, \lambda^0)$ be the initial iterate. For a given compact set $\mathcal{D} \subset \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{R}^m$, let $d = \sup\{\|w - w^0\|_H \mid w \in \mathcal{D}\}$. Then, after $t$ iterations of the scheme (3.1.1), the point $\tilde{w}_t \in \Omega_2$ defined in (3.3.1) satisfies

$$\sup_{w \in \mathcal{D}} \{\theta(\tilde{u}_t) - \theta(u) + (\tilde{w}_t - w)^T F_2(w)\} \leq \frac{d^2}{2(t+1)}, \quad (3.3.4)$$

which means $\tilde{w}_t$ is an approximate solution of $VI(\Omega_2, F_2, \theta)$ with an accuracy of $O(1/t)$ (Recall the characterization of $\Omega^*_2$ in Theorem 3.2.1).
3.3.2 Convergence rate in a nonergodic sense

In this section, we establish a worst-case $O(1/t)$ convergence rate in a nonergodic sense for the sequence $\{w^k\}$ generated by the scheme (3.1.1).

In the next two lemmas, we will prove the monotonicity of the sequence $\|w^k - w^{k+1}\|_H^2$. It is critical for a nonergodic convergence rate analysis.

Lemma 3.3.2 Let the sequence $\{w^k\}$ be generated by the scheme (3.1.1) and the sequence $\tilde{w}^k$ be defined in (3.2.5). Then, we have

$$\tilde{w}^k - \tilde{w}^{k+1} \geq (\tilde{w}^k - \tilde{w}^{k+1})^T Q (w^k - \tilde{w}^k),$$

where the matrix $Q$ is defined in (3.2.3).

Proof: Setting $w := \tilde{w}^{k+1}$ in (3.2.10), we have

$$\theta(\tilde{w}^{k+1}) - \theta(\tilde{w}^k) + (\tilde{w}^{k+1} - \tilde{w}^k)^T F_2(\tilde{w}^k) \geq (\tilde{w}^{k+1} - \tilde{w}^k)^T Q (w^k - \tilde{w}^k).$$

Note that (3.2.10) is also true for $k := k + 1$. Thus, we have

$$\theta(w) - \theta(\tilde{w}^{k+1}) + (w - \tilde{w}^{k+1})^T F_2(\tilde{w}^{k+1}) \geq (w - \tilde{w}^{k+1})^T Q (w^{k+1} - \tilde{w}^{k+1}), \quad \forall w \in \Omega_2.$$

Setting $w := \tilde{w}^k$ in the above inequality, we obtain

$$\theta(\tilde{w}^k) - \theta(\tilde{w}^{k+1}) + (\tilde{w}^{k+1} - \tilde{w}^{k+1})^T F_2(\tilde{w}^{k+1}) \geq (\tilde{w}^k - \tilde{w}^{k+1})^T Q (w^{k+1} - \tilde{w}^{k+1}).$$

Adding (3.3.6)-(3.3.7) and using the monotonicity of $F_2$, we complete the proof immediately.

Lemma 3.3.3 Let the sequence $\{w^k\}$ be generated by the scheme (3.1.1) and the sequence $\tilde{w}^k$ be defined in (3.2.5). Then

$$\|w^{k+1} - w^{k+2}\|_H^2 \leq \|w^k - w^{k+1}\|_H^2,$$

where the matrix $H$ is defined in (3.2.3).
Proof: Adding the identity

\[
\{(w^k - w^{k+1}) - (\tilde{w}^k - \tilde{w}^{k+1})\}^T Q \{(w^k - w^{k+1}) - (\tilde{w}^k - \tilde{w}^{k+1})\}
\]

\[
= \frac{1}{2} \|(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\|_Q^2.
\]

to both sides of (3.3.5), we get

\[
(w^k - w^{k+1})^T Q \{(w^k - w^{k+1}) - (\tilde{w}^k - \tilde{w}^{k+1})\} \geq \frac{1}{2} \|(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\|_Q^2.
\]

(3.3.9)

By (3.2.9) and Lemma 3.2.2, (3.3.9) can be written as

\[
(w^k - \tilde{w}^k)^T M^T HM \{(w^k - w^{k+1}) - (\tilde{w}^k - \tilde{w}^{k+1})\} \geq \frac{1}{2} \|(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\|_Q^2.
\]

(3.3.10)

Setting \(a := M(w^k - \tilde{w}^k)\) and \(b := M(w^{k+1} - \tilde{w}^{k+1})\) in the identity

\[
\|a\|^2_H - \|b\|^2_H = 2a^T H (a - b) - \|a - b\|^2_H,
\]

we obtain

\[
\|M(w^k - \tilde{w}^k)\|^2_H - \|M(w^{k+1} - \tilde{w}^{k+1})\|^2_H = 2(w^k - \tilde{w}^k)^T M^T HM \{(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\}
\]

\[
- \|M(w^k - \tilde{w}^k) - M(w^{k+1} - \tilde{w}^{k+1})\|^2_H.
\]

Since the inequality (3.3.10) holds, we have

\[
\|M(w^k - \tilde{w}^k)\|^2_H - \|M(w^{k+1} - \tilde{w}^{k+1})\|^2_H \geq \|(w^k - \tilde{w}^k) - (w^{k+1} - \tilde{w}^{k+1})\|_Q^2 \geq 0.
\]

By (3.2.9), we immediately have

\[
\|w^{k+1} - w^{k+2}\|^2_H \leq \|w^k - w^{k+1}\|^2_H.
\]

(3.3.11)

The lemma is proved.

Now, we can establish a worst-case \(O(1/t)\) convergence rate in a nonergodic sense for the scheme (3.1.1).

Theorem 3.3.4 Let the sequence \(\{w^k\}\) be generated by the scheme (3.1.1) with \(r, s \in [0, 1]\) and \(0 < r + s < 2\). It holds

\[
\|w^k - w^{k+1}\|^2_H = \mathcal{O}(1/k), \quad k \to \infty.
\]

(3.3.12)
Proof: Setting $d = \sum_{k=1}^{\infty} \|w^k - w^{k+1}\|_H^2$ in (3.2.25), we have

$$\sum_{i=0}^{k} \|w^i - w^{i+1}\|_H^2 \leq d. \quad (3.3.13)$$

By Lemma 3.3.3, the sequence $\{\|w^k - w^{k+1}\|_H^2\}$ is monotonically non-increasing. Therefore, we have

$$(k + 1)\|w^i - w^{i+1}\|_H^2 \leq \sum_{i=0}^{k} \|w^i - w^{i+1}\|_H^2. \quad (3.3.14)$$

Thus, the assertion (3.3.12) follows from (3.3.13) and (3.3.14) immediately. \hfill \square

3.4 Applications to image processing

In this section, we apply the proposed proximal version of the SC-PRSM (3.1.1) (PSC-PRSM for abbreviation) to solve some concrete applications of the abstract model (P2) arising in image processing; and report the numerical results. Our purpose is to show how to choose an appropriate proximal matrix for the PSC-PRSM (3.1.1) for a given application of (P2) such that the resulting subproblems are much alleviated than the subproblems of the original SC-PRSM (1.4.13); the necessity of combining a proximal regularization term with (1.4.13) and accelerating it numerically (which is the main motivation of proposing the PSC-PRSM) is thus illustrated. These special cases include:

- Case 1, the $x_1$-subproblem in (1.4.13) does not have a closed-form solution but the function $\theta_1$ is special in the sense that the resolvent operator of $\partial \theta_1$ is given explicitly (such as the mentioned $l_1$-norm or nuclear-norm term has a closed form solution); and

- Case 2, the $x_1$-subproblem in (1.4.13) does have a closed-form solution but it is too computationally expensive.

We will particularly compare the PSC-PRSM (3.1.1) with the original SC-PRSM (1.4.13), and demonstrate how the proximal regularization term in (3.1.1) can be
numerically beneficial for some applications of the model. Note that the necessity of embedding the linearization technique into splitting methods to generate easier subproblems has been well illustrated in the ADMM literature, see, e.g., \cite{130,148,152,156}. Here we reiterate this philosophy of algorithmic design in the PRSM literature, by comparing the PSC-PRSM and SC-PRSM for a wavelet-based image inpainting model (see (3.4.1)). For Case 2, we test the computerized tomography problem (see (3.4.13)) arising in medical image processing and show how efficient the PSC-PRSM is.

We need to make it clear how to specify the choices of parameters for the PSC-PRSM (3.1.1) in our experiments. Recall that the parameters $r$ and $s$ are both underdetermined relaxation factors, and they play the same role as $\alpha$ in the original SC-PRSM (1.4.13). Since how to choose $\alpha$ has been addressed in detail in \cite{82}, we here do not discuss the detail of how to determine $r$ and $s$ for the PSC-PRSM. We just follow the experience in \cite{82} and choose values close to 1 for them. For the penalty parameter $\beta$, it is commonly known that operator splitting type methods including the ADMM and PRSM type methods are sensitive to its value and it is generally required to tune it for a special implementation of such a method. Since it is not the emphasis of this chapter, we here do not report the sensitivity of the PSC-PRSM to the choice of $\beta$. Instead, we just tune it and fix it throughout the test of an application.

All codes in our numerical experiments were written by Matlab 7.9; and were run on a desktop computer with an Intel CPU (2.66GHz) and 3.5GB RAM running Windows XP.

3.4.1 A wavelet-based inpainting model

We first take a wavelet-based image inpainting model as an illustrative example of the mentioned Case 1 and verify numerically the efficiency of the PSC-PRSM (3.1.1).
3.4.1.1 Application background

A brief introduction to wavelet-based image processing is as follows. For more details, see e.g. [138]. Let \( x \in \mathcal{R}^l \) represent an \( l_1 \times l_2 \) digital image with \( l = l_1 \cdot l_2 \). Note that a two-dimensional image can be represented by vectorizing it as a one-dimensional vector in certain order, e.g. the lexicographic order. Let \( W \in \mathcal{R}^{l \times n} \) be a wavelet dictionary, i.e., each column of \( W \) be the element of a wavelet frame (an orthogonal basis or an overcomplete dictionary). For an image \( x \) possessing a sparse representation under the dictionary \( W \), it can be written as \( x = W\eta \) with \( \eta \in \mathcal{R}^n \) being a sparse vector. A wavelet-based image processing problem is to reconstruct the clean image \( x = W\eta \) from an observation \( b \in \mathcal{R}^l \) (e.g., with missing pixels or noise). We consider the wavelet-based image inpainting model in [55]:

\[
\min \left\{ \|\eta\|_1 \mid \|A\eta - b\| \leq \epsilon \right\}.
\] (3.4.1)

In (3.4.1), \( A = SBW \) where the matrix \( S \in \mathcal{R}^{l \times l} \) is a diagonal matrix whose diagonal elements are either 0 or 1. The locations of 0 and 1 correspond to missing and known pixels of the image \( W\eta \), respectively; the matrix \( B \in \mathcal{R}^{l \times l} \) is a blurring matrix and the positive scalar \( \epsilon \) is the noise level.

3.4.1.2 Implementation of PSC-PRSM

Note that the model (3.4.1) can be written as

\[
\min \left\{ \|\eta\|_1 \mid A\eta - b = y; \|y\| \leq \epsilon \right\},
\] (3.4.2)

where \( y \in \mathcal{R}^l \) is an auxiliary variable. Let \( C = \{y : \|y\| \leq \epsilon\} \) and \( \delta_C(\cdot) \) be the indicator function of the convex set \( C \). Then, the model (3.4.2) is a special case of the abstract model (122) with \( x_1 := \eta, \theta_1(x_1) := \|\eta\|_1, \theta_2(x_2) := \delta_C(y), A_1 := A \) and \( A_2 := -I \); thus the proposed PSC-PRSM (3.1.1) is applicable. It is easy to see that
the resulting iterative scheme when (3.1.1) is applied to (3.4.2) is

\[
\begin{align*}
\eta^{k+1} &= \arg \min \left\{ \| \eta \|_1 + \frac{\beta}{2} \| A\eta - y^k - b - \frac{1}{\beta} \lambda^k \|_2^2 + \frac{1}{2} \| \eta - \eta^k \|_2^2 \right\}, \\
\lambda^{k+\frac{1}{2}} &= \lambda^k - r\beta (A\eta^{k+1} - y^k - b), \\
y^{k+1} &= \arg \min \left\{ \| Ay^{k+1} - y - b - \frac{1}{\beta} \lambda^{k+\frac{1}{2}} \|_2^2 \mid \| y \| \leq \epsilon \right\}, \\
\lambda^{k+1} &= \lambda^{k+\frac{1}{2}} - s\beta (A\eta^{k+1} - y^{k+1} - b).
\end{align*}
\] (3.4.3)

Let us explain how to solve the subproblems in (3.4.3).

- Note that the objective function of the \(\eta\)-subproblem in (3.4.3) consists of an \(l_1\)-norm term and two quadratic terms. In order to take advantage of the fact that the resolvent operator of \(\| \cdot \|_1\) has a closed-form representation, just like the idea of the linearized ADMM (1.4.11) in [130, 148, 152, 156], an obvious choice of \(G\) is

\[
G = \mu I - \beta A^T A,
\] (3.4.4)

in order to make the \(\eta\)-subproblem in (3.4.3) easier. With this choice, the \(\eta\)-subproblem in (3.4.3) reduces to

\[
\eta^{k+1} = \arg \min \left\{ \| \eta \|_1 + \frac{\mu}{2} \| \eta - \eta^k + \frac{\beta}{\mu} A^T (A\eta^k - y^k - b - \frac{1}{\beta} \lambda^k) \|_2^2 \right\},
\] (3.4.5)

whose solution is given explicitly by the soft-thresholding operator (1.2.7)

\[
\eta^{k+1} = S_{\frac{1}{\mu}} \left( \eta^k - \frac{\beta}{\mu} A^T (A\eta^k - y^k - b - \frac{1}{\beta} \lambda^k) \right),
\] (3.4.6)

- The solution of the \(y\)-subproblem in (3.4.3) is given explicitly by

\[
y^{k+1} = (Ay^{k+1} - b - \lambda^{k+\frac{1}{2}} / \beta) \cdot \max \left( \epsilon / \| Ay^{k+1} - b - \lambda^{k+\frac{1}{2}} / \beta \|, 1 \right).\] (3.4.7)

Finally, let us explain how to choose \(\mu\) when determining the matrix \(G\) in (3.4.3). Technically, the convergence of the PSC-PRSM has been proved under the condition that \(G\) is positive semi-definite, which can be guaranteed sufficiently when \(\mu \geq \beta \| A_1^T A_1 \|\). Since the dictionary \(W\) has the property \(WW^T = I\) and the mask \(S\) satisfies \(\| S \| = 1\), we have \(\| A_1^T A_1 \| = \| W^T B^T S^T S B W \| = 1\). In other words, \(\mu \geq \beta\) is sufficient to ensure the positive semi-definiteness of \(G\) and thus the convergence of
the sequences generated by (3.4.3). This condition requires that the value of \( \mu \) should be sufficiently large so as to theoretically ensure the convergence of (3.4.3). But a larger value of \( \mu \) also means the proximal term has a heavier weight in the objective of the \( \eta \)-subproblem in (3.4.3); thus forces the new iterate \( \eta^{k+1} \) to be closer to the last iterate \( \eta^k \) — slower convergence thus might occur due to the “too-small-step-size” phenomenon if \( \mu \) is too large. Therefore, the principle of choosing \( \mu \) to determine the proximal matrix \( G \) via (3.4.4) is to choose a value as small as possible while obeying the requirement \( \mu \geq \beta \|W^T B^T S^T SBW\| \) — sometimes it is even more effective to relax this sufficient requirement appropriately in computation for some iterations. According to our numerical experiments, we have observed that this practical strategy can accelerate the PSC-PRSM significantly. For succinctness, we just specify our tuned strategy of choosing this parameter for the tested example, but do not report the detail of PSC-PRSM’s sensitivity to \( \mu \).

3.4.1.3 Comparison with SC-PRSM

We reiterate that the proposed PSC-PRSM is a proximal version of the SC-PRSM in [82], and it is able to linearize the quadratic term of the \( x_1 \)-subproblem in (1.4.13) and thus generate an easier subproblem with a closed-form solution for some special cases of \( \theta_1 \). As we have mentioned, this can be regarded as a special strategy of solving the \( x_1 \)-subproblem in (1.4.13) approximately — only one iteration is executed for this subproblem. Thus, to show the efficiency of the PSC-PRSM, it is necessary to compare it with the original SC-PRSM whose \( x_1 \)-subproblem is solved iteratively by some methods at each iteration.

In this subsection, we test the model (3.4.1) to demonstrate this comparison. Note that the iterative scheme of SC-PRSM for solving (3.4.2) is

\[
\begin{align*}
\eta^{k+1} &= \arg \min \left\{ \| \eta \|_1 + \frac{\beta}{2} \| A\eta - y^k - b - \frac{1}{\beta} \lambda^k \|_2 \right\}, \\
\lambda^{k+\frac{1}{2}} &= \lambda^k - \alpha \beta (A\eta^{k+1} - y^k - b), \\
y^{k+1} &= \arg \min \left\{ \| A\eta^{k+1} - y - b - \frac{1}{\beta} \lambda^{k+\frac{1}{2}} \|_2 \mid \| y \| \leq \epsilon \right\}, \\
\lambda^{k+1} &= \lambda^{k+\frac{1}{2}} - \alpha \beta (A\eta^{k+1} - y^{k+1} - b).
\end{align*}
\]
We adopt two very efficient solvers for solving the $\eta$-subproblem in (3.4.8): the “FISTA” in [11] and “GPSR” in [60]. In the following, “SC-PRSM+FISTA” and “SC-PRSM+GPSR” denote the implementation of SC-PRSM (3.4.8) where its $\eta$-subproblem at each iteration is solved iteratively by FISTA and GPSR, respectively. To probe the possibility of accelerating SC-PRSM by changing the accuracy of solving the $\eta$-subproblem, we test different accuracies ($10^{-1}, 10^{-2}, 10^{-3}$) of the $\eta$-subproblem’s solutions obtained approximately by FISTA and GPSR. A maximal number of 50 is set for the implementation of FISTA and GPSR.

We test the $256 \times 256$ image Peppers.png, as shown in Figure 3.1. The dictionary $W$ is chosen as the inverse Haar wavelet transform with a level of 6 (see e.g., [1,11]). The clean Peppers image is first blurred by the out-of-focus convolution using the MATLAB command “fspecial(‘disk’,3)”. Then some pixels labeled as texts are missing; and finally, the Gaussian white noise with a standard variation 0.01 is added to the degraded image. The scalar $\epsilon$ in (3.4.1) is set as the spectral norm of the additive Gaussian noise, i.e., $\epsilon = 2.568$. The degraded image is also shown in Figure 3.1.

The quality of a restored image is measured by the signal-to-noise ratio (SNR) in

---

1. We wrote the code on our own due to the lack of its package for the model (3.4.1).
3. In our numerical experiments, the stopping criteria were satisfied by less than 50 inner iterations for all the tested scenarios.
decibel (dB)

\[ \text{SNR} := 20 \log_{10} \frac{\|x\|}{\|\bar{x} - x\|}, \]

where \( \bar{x} \) is a restored image and \( x \) is a clean image.

The initial value is taken as \((\eta^0, \lambda^0, y^0) = (0, 0, 0)\) for all methods to be tested.

Parameters for the methods to be compared are given below.

- For PSC-PRSM, \( r = 0.95, s = 0.8, \beta = 1, \mu \) is chosen by the following dynamical strategy:
  \[
  \mu^{k+1} = \begin{cases} 
  \mu^k, & \text{if } \text{mod}(k, 2) \neq 0; \\
  \max \{\mu^k \times 0.5, 0.98\}, & \text{if } \text{mod}(k, 2) = 0 \text{ and } k \leq 100; \\
  \min \{\mu^k \times 1.01, 1.2\}, & \text{if } \text{mod}(k, 2) = 0 \text{ and } k > 100,
  \end{cases}
  \]
  with \( \mu^0 = 1.5\|A^T A\| = 1.54 \).

- For SC-PRSM+FISTA, \( \alpha = 0.9, \beta = 1 \). The \( \eta \)-subproblem in (3.4.8) is solved by FISTA with the Lipschitz constant as \( \|A^T A\| = 1 \).

- For SC-PRSM+GPSR, \( \alpha = 0.9, \beta = 1 \). The \( \eta \)-subproblem in (3.4.8) is solved by GPSR\(^5\).

To implement FISTA and GPSR for SC-PRSM+FISTA and SC-PRSM+GPSR, respectively, we set the initial iterate as the newly obtained iterate \( \eta^k \) from the outer loop.

In Figure 3.2, we plot the evolutions of SNR values with respect to computing time in seconds (up to 200 seconds) for PSC-PRSM, SC-PRSM+FISTA and SC-PRSM+GPSR. The different accuracies for implementing FISTA and GPSR are given in the parentheses. According to this figure, PSC-PRSM is advantageous, as it achieves the highest SNR values in the shortest time. This verifies the motivation of accelerating SC-PRSM by generating an easier proxy of its subproblem (via

\(^4\)In our numerical experiments, we only observed 180 times out of 1,000 iterations that \( \mu < 1 \).

\(^5\)We use the monotone version of GPSR and take its parameter values set by defaulted in the original package of GPSR.
a proximal regularization), rather than solving it iteratively. The efficiency of the PSC-PRSM is thus illustrated. This figure also shows that SC-PRSM+FISTA is also very competitive with PSC-PRSM, especially when the $\eta$-subproblem of SC-PRSM is solved with lower accuracy (e.g. $10^{-1}, 10^{-2}$). For SC-PRSM+GPSR, under different accuracies of GPSR, they give similar performance and are much slower than PSC-PRSM in terms of time.

In Table 3.1 we list the computing time in seconds required by each method to achieve some given SNR values. For SC-PRSM+FISTA and SC-PRSM+GPSR, we only list their respective fastest cases in our experiments. The images restored by executing these three methods for 200 seconds are shown in Figure 3.3.

![Figure 3.2: The evolution of SNR (dB) w.r.t. computing time for Peppers.](image)

### 3.4.1.4 Comparison with other benchmarks

In the literature, there are many well-developed packages that are applicable to the model (3.4.1). For example, NESTA in [12], CSALSA in [112] and SPGL1 in [146][147]. In particular, CSALSA is essentially an application of the ADMM (1.4.7) with the
Table 3.1: Comparison of computing time to achieve different SNR on Peppers.

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>22</th>
<th>26</th>
<th>30</th>
<th>34</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSC-PRSM</td>
<td>0.81</td>
<td>2.49</td>
<td>6.03</td>
<td>29.62</td>
</tr>
<tr>
<td>SC-PRSM+FISTA(10^{-1})</td>
<td>0.96</td>
<td>3.13</td>
<td>8.03</td>
<td>35.30</td>
</tr>
<tr>
<td>SC-PRSM+GPSR(10^{-1})</td>
<td>1.50</td>
<td>6.65</td>
<td>21.80</td>
<td>129.03</td>
</tr>
</tbody>
</table>

Figure 3.3: Restored images by the three test methods on Peppers.
condition that \((A^T A + \alpha I)\) or \((A_1 A_1^T + \alpha I)\) should be invertible for any \(\alpha > 0\). Note that this condition is not satisfied by the model \(3.4.1\), unless \(S\) is an identity matrix, i.e., \(A = BW\) in \(3.4.1\), where \(B\) is the matrix representation of the convolution operator and \(W\) is a wavelet dictionary. Also, the superiority of CSALSA to NESTA was shown in [1]. We thus compare the proposed PSC-PRSM with the ADMM \(1.4.7\) and the SPGL1 in [146, 147].

To implement the ADMM \(1.4.7\) for \(3.4.1\), we follow CSALSA in [1, 2] and reformulate \(3.4.1\) as

\[
\min \{ \|z\|_1 \mid z = \eta; y = A\eta - b; \|y\| \leq \epsilon \},
\]

which corresponds to the model \((P2)\) with \(x_1 := (z^T, y^T)^T, x_2 := \eta, \theta_1(x_1) := \|z\|_1 + \delta_C(y), \theta_2(x_2) := 0, A_1 := \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}, A_2 := \begin{pmatrix} -I \\ -A \end{pmatrix}, \) and \(b := \begin{pmatrix} 0 \\ -b \end{pmatrix}\). Thus the ADMM \(1.4.7\) is applicable; and its iterative scheme reads as

\[
\begin{align*}
  z^{k+1} &= \arg \min \left\{ \|z\|_1 + \frac{\beta_1}{2} \|z - \eta^k - \frac{1}{\beta_1} \lambda_1^k\|_2^2 \right\}, \\
  y^{k+1} &= \arg \min \left\{ \|y - A\eta^k + b - \frac{1}{\beta_2} \lambda_2^k\|_2^2 \mid \|y\| \leq \epsilon \right\}, \\
  \eta^{k+1} &= \arg \min \left\{ \|z^{k+1} - \eta - \frac{1}{\beta_1} \lambda_1^k\|_2^2 + \|y^{k+1} - A\eta + b - \frac{1}{\beta_2} \lambda_2^k\|_2^2 \right\}, \\
  \lambda_1^{k+1} &= \lambda_1^k - \beta_1(z^{k+1} - \eta^{k+1}), \\
  \lambda_2^{k+1} &= \lambda_2^k - \beta_2(y^{k+1} - A\eta^{k+1} + b),
\end{align*}
\]

with \(\beta_1 > 0\) and \(\beta_2 > 0\) being the penalty parameters.

In \(3.4.10\), the \(z\)-subproblem can be solved via shrinkage operation, see \((1.2.7)\)

\[
z^{k+1} = S_{1/\beta_1} (\eta^k + \lambda_1^k/\beta_1); \tag{3.4.11}
\]

the \(y\) subproblem can be solved by

\[
y^{k+1} = (A\eta^k - b + \lambda_2^k/\beta_2) \cdot \max \left( \epsilon / \|A\eta^k - b + \lambda_2^k/\beta_2\|, 1 \right). \tag{3.4.12}
\]

For the \(\eta\)-subproblem whose closed-form solution does not exist, since \(A^T A + I\) is not invertible, we employ the preconditioned conjugate gradient (PCG) method to solve

\[\text{SPGL1}\] uses the spectral projected gradient algorithm developed in [18]; and its code was downloaded at [http://www.cs.ubc.ca/~mpf/spgl1/download.html](http://www.cs.ubc.ca/~mpf/spgl1/download.html)
it iteratively. We again set different accuracies ($10^{-1}, 10^{-2}, 10^{-4}, 10^{-6}$) in the PCG’s stopping criterion and compare all the cases. The initial iterate for implementing PCG is set to be the last iterate $\eta_k^k$ from the outer loop (3.4.10). A maximum number of 30 is set for the implementation of PCG.\footnote{In our experiments, the stopping criteria were satisfied by less than 30 inner iterations for all tested scenarios.}

We test the 256 $\times$ 256 image Boat.png, as shown in Figure 3.4. The dictionary $W$ is chosen as the inverse Haar wavelet transform with a level of 6 (see e.g., [11]). The Boat image is first blurred by the out-of-focus convolution using the MATLAB command \texttt{fspecial(‘disk’,5)}. Then some pixels labeled as texts are missing, and finally the Gaussian white noise with a standard variation 0.05 is added to the degraded image (thus the scalar $\epsilon$ in (3.4.1) is set as 12.8401). The degraded image is also displayed in Figure 3.4.

The parameters of the methods to be compared are described as follows.

- For the PSC-PRSM, the parameter setting is the same as that in Section 3.4.1.3.
- For ADMM (3.4.10), $\beta_1 = 1$ and $\beta_2 = 1$.
- For the SPGL1, we take the default parameter settings suggested in its package released by the authors.

In Figure 3.5 we plot the evolutions of the SNR values with respect to the computing time in seconds (up to 400 seconds). The PSC-PRSM can achieve the same level of SNR values as SPGL1 and ADMM, but in a much faster speed, especially in the first 100 seconds. According to this figure, we found that the ADMM’s subproblems should be solved in an appropriate accuracy — neither a too loose (e.g., $10^{-1}$) nor a too high (e.g., $10^{-6}$) accuracy should be employed for solving the ADMM’s subproblems.

In Table 3.2 we list the respective computing time in seconds for the tested methods to achieve some given SNR values. The images restored by implementing each tested method for 400 seconds are displayed in Figure 3.6.
Figure 3.4: Clean and degraded images of Boat.

Figure 3.5: The evolution of SNR (dB) w.r.t. computing time for Boat.

Table 3.2: Comparison of computing time to achieve different SNR on Boat.

<table>
<thead>
<tr>
<th>SNR</th>
<th>10</th>
<th>14</th>
<th>18</th>
<th>22</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSC-PRSM</td>
<td>0.13</td>
<td>0.37</td>
<td>1.63</td>
<td>13.25</td>
</tr>
<tr>
<td>ADMM($10^{-2}$)</td>
<td>1.19</td>
<td>2.07</td>
<td>7.21</td>
<td>62.55</td>
</tr>
<tr>
<td>SPGL1</td>
<td>0.21</td>
<td>0.85</td>
<td>8.28</td>
<td>53.22</td>
</tr>
</tbody>
</table>
3.4.2 The computational tomography problem

In this subsection, we take the computational tomography problem arising in medical image processing as an illustrative example of the mentioned Case 2 and verify numerically the efficiency of the PSC-PRSM (3.1.1).

3.4.2.1 Application background

Here we give a brief introduction to the computerized tomography problem. The goal of a computerized tomography system is to reconstruct the internal structure of the body as two-dimensional cross-sectional images. Computational tomography works by sending X-ray beams through the patient’s body from many directions and reconstructing the body’s slice images from the measured projections, see [94] for more background. The measured projections can be represented by the Radon transform of the body slice. Let $u \in \mathbb{R}^n$ denote the unknown body slice image, and $R \in \mathbb{R}^{m \times n}$ represent the Radon transform matrix. Then, the measured projection procedure can be described by $b = Ru + \epsilon$, where $\epsilon$ represents the noise. The technique of Filtered Back Projection (FBP), see [31], is one of the most widely used algorithms in clinics for reconstructing the body slice image $u$. It essentially involves solving the inverse Radon transform of the measured projections, i.e., $R^{-1}b$. However, in presence of noise $\epsilon$, the reconstruction by FBP will be perturbed by the magnified noise since the condition number of Radon transform is very large, see [79].

Figure 3.6: Restored images by the three test methods on Boat.
the total variational (TV) technique in [133] that is good in reconstructing image’s sharp edges, a computational tomography imaging model (see [20]) is

\[
\min \{ \tau \| \nabla u \|_1 + \frac{1}{2} \| Ru - b \|_2^2 \},
\]

(3.4.13)

where \( \| \nabla u \| \) is the total variation (TV) norm of \( u \). We assume that the \( \nabla^T \) adopts the circulant boundary condition; and \( \tau > 0 \) is a regularization parameter.

### 3.4.2.2 The implementation of PSC-PRSM

Note that the model (3.4.13) can be reformulated as

\[
\min \{ \tau \| v \|_1 + \frac{1}{2} \| Ru - b \|_2^2 \mid \nabla u - v = 0 \},
\]

(3.4.14)

where \( v \in \mathbb{R}^{2n} \) is an auxiliary variable. Thus, it is a special case of the model (P2) with \( x_1 := u, x_2 := v, \theta_1(x_1) := \frac{1}{2} \| Ru - b \|_2^2, \theta_2(x_2) := \tau \| v \|_1, A := \nabla, B := -I_{2n}, b := 0 \). The SC-PRSM (1.4.13) and PSC-PRSM (3.1.1) are thus both applicable.

Let us first see the implementation of the SC-PRSM (1.4.13) to (3.4.14). More specifically, its iterative scheme is

\[
\begin{aligned}
\lambda_k^{+1} &= \lambda_k - \alpha \beta (\nabla u_k^{+1} - v_k) \\
v_k^{+1} &= \arg \min \left\{ \tau \| v \|_1 + \frac{1}{2} \| \nabla u_k^{+1} - v - \frac{1}{\beta} \lambda_k^{+1} \|_2^2 \right\}, \\
\lambda_k^{+1} &= \lambda_k^{+2} - \alpha \beta (\nabla u_k^{+1} - v_k^{+1}).
\end{aligned}
\]

(3.4.15)

Then, how to solve the \( u \)- and \( v \)-subproblems in (3.4.15) can be summarized as follows.

- The solution of the \( u \)-subproblem in (3.4.15) is given by the system of linear equations

\[
(R^T R + \beta \nabla^T \nabla)u^{k+1} = R^T b + \beta \nabla^T (v_k^{+1} - \frac{1}{\beta} \lambda_k^{+1}).
\]

(3.4.16)

- The solution of the \( v \)-subproblem in (3.4.15) is given explicitly by the shrinkage operator [1.2.7]

\[
v_k^{+1} = S_{\tau/\beta}(\nabla u_k^{+1} - \lambda_k^{+2}/\beta).
\]
The coefficient matrix of the system of linear equations (3.4.16) is symmetric and positive definite. Thus, theoretically, \( u^{k+1} \) can be solved directly by using some direct solvers, e.g. the Cholesky decomposition. However, because the Radon matrix is generic without any specific structure and it is often of high dimension for medical imaging models, see e.g. [44], direct solvers for this system of linear equations are usually very expensive. We will verify this fact by applying the Cholesky decomposition to (3.4.16). Alternatively, we can solve (3.4.16) approximately by some iterative method to shorten the computing time. In the following, for comparison purpose, we will test the scenarios where (3.4.16) is solved by the Barzilai-Borwein (BB) gradient method in [10] and the preconditioned conjugate gradient (PCG) method. We will denote by “SC-PRSM+Chol”, “SC-PRSM+BB” and “SC-PRSM+PCG” the SC-PRSM (3.4.15) when its \( u \)-subproblem is solved by the Cholesky decomposition, BB and PCG methods, respectively. Again, different accuracies \( 10^{-3}, 10^{-4}, 10^{-5} \) will be tested for implementing the BB and PCG methods and they are specified in the parentheses.

Then, we list the iterative scheme when the PSC-PRSM (3.1.1) is applied to (3.4.14):

\[
\begin{align*}
\lambda^{k+1} &= \lambda^k - r\beta(\nabla u^{k+1} - v^k), \\
v^{k+1} &= \arg\min \left\{ \tau \|v\|_1 + \frac{\beta}{2} \|\nabla u^{k+1} - v - \frac{1}{\beta} \lambda^{k+1} \|_2^2 \right\}, \\
\lambda^{k+1} &= \lambda^{k+1} - s\beta(\nabla u^{k+1} - v^{k+1}).
\end{align*}
\]

(3.4.17)

In fact, with an appropriate choice of \( G \), the \( u \)-subproblem in (3.4.17) can be alleviated significantly. Below are two such choices.

- Let \( G := \mu I - RT R - \beta \nabla T \nabla \) with \( \mu \geq \| R^T R + \beta \nabla T \nabla \|. \) Then, the solution of the \( u \)-subproblem in (3.4.17) is given by

\[
u^{k+1} = u^k - \frac{1}{\mu} R^T (Ru^k - b) - \frac{1}{\mu} \nabla^T (\beta \nabla u^k - \beta v^k - \lambda^k).
\]

(3.4.18)

Note that the system (3.4.18) is much easier than the system (3.4.16) since \( u^{k+1} \) can be explicitly calculated instead of solving a system of linear equations. We
follow the notation in [156] and call (3.4.18) an explicit solution. The PSC-
PRSM with this choice of $G$ is denoted by “PSC-PRSM-EXP” accordingly.

- Let $G := \mu I - R^T R$ with $\mu \geq \|R^T R\|$. Then, the solution of the $u$-subproblem in (3.4.17) is given by

$$
(\mu I + \beta \nabla^T \nabla)u^{k+1} = (\mu u^k - R^T (Ru^k - b) + \nabla^T (\lambda^k + \beta v^k)).
$$

(3.4.19)

Note that under the circulant boundary condition, the coefficient matrix of (3.4.19) can be diagonalized by the Fast Fourier Transform (FFT), see e.g. [79].

Thus, it can be solved very efficiently. We follow the notation in [156] and call (3.4.19) a semi-implicit solution because the function of the matrix $G$ is to linearize the quadratic term $\|Ru - b\|^2$ in the objective function of the $u$-
subproblem in (3.4.17). The PSC-PRSM with this choice of $G$ is denoted by

“PSC-PRSM-IMP” accordingly.

We test the $128 \times 128$ Shepp-Logan phantom and $128 \times 128$ magnetic resonance images. Each image has 50 uniformly oriented projections corrupted by the Gaussian white noise with the standard variance 1. We list the original images and the FBP images in Figure 3.7.

Figure 3.7: Shepp-Logan phantom, MRI and their FBP images.

3.4.2.3 Comparison with SC-PRSM

We first compare the PSC-PRSM with SC-PRSM for the computational tomography model (3.4.13). More specifically, we compare “PSC-PRSM-EXP” and “PSC-PRSM-IMP” with “SC-PRSM+Chol”, “SC-PRSM+BB” and “SC-PRSM+PCG”.
The quality of a restored image measured by the improved signal to noise ratio (ISNR) in unit of dB is defined as

\[
\text{ISNR} := 10 \log_{10} \left( \frac{\| u^0 - u \|^2}{\| u - \hat{u} \|^2} \right),
\]

where \( u \in \mathbb{R}^n \) is the original image, \( u^0 \) is the distorted image and \( \hat{u} \) is the restored image.

The scalar \( \tau = 0.06 \) in the model (3.4.13). The initial iterates are taken as the FBP images for all methods to be tested. Parameters of the methods to be tested are listed as below.

- For PSC-PRSM-EXP and PSC-PRSM-IMP, we choose \( \beta = 0.1 \), \( r = 0.95 \) and \( s = 0.9 \) in (3.4.17). The parameter \( \mu \) in (3.4.19) and (3.4.18) is adjusted by

\[
\mu^{k+1} = \begin{cases} 
\mu^k, & \text{if } \text{mod}(k, 20) \neq 0; \\
\max \{ \mu^k \times 0.6, 0.8 \}, & \text{if } \text{mod}(k, 20) = 0 \text{ and } k \leq 100; \\
\min \{ \mu^k \times 1.01, 5 \}, & \text{if } \text{mod}(k, 20) = 0 \text{ and } k > 100,
\end{cases}
\]

where \( \mu^0 = 1.7 \).

- For SC-PRSM+Chol, SC-PRSM+PCG and SC-PRSM+BB, we choose \( \beta = 0.1 \) and \( \alpha = 0.9 \) in (3.4.15). The maximum iteration number of PCG and BB is set to be 50. In practice, we find it is a safe bound to ensure the subproblem is solved with the specific accuracy (e.g. \( 10^{-3}, 10^{-4}, 10^{-5} \)) by PCG and BB methods. The initial point of PCG and BB is set to be the last iterate \( u^k \) from the outer loop (3.4.15).

In Figures 3.8 and 3.9, we plot the evolution of the ISNR value and the objective function value with respect to the computing time for the Shepp-Logan phantom and magnetic resonance images, respectively. Since the SC-PRSM+Chol is significantly slower than the other methods (see Tables 3.3 and 3.4), we do not include its curves in Figures 3.8 and 3.9 for the convenience of displaying other curves clearly. From

\[8\]To ensure the positive semi-definiteness of \( G \) i.e. \((\mu I - R^T R - \beta \nabla T \nabla)\) and \((\mu I - R^T R)\), the initial value of \( \mu \) is set as \( \mu = 1.1 \times \| R^T R + \beta \nabla T \nabla \| = 1.1 \times 1.58 = 1.738 \approx 1.7 \). There are 1,600 times out of 2,000 iterations that \( \mu \) does not satisfy \( \mu \geq 1.7 \).
these two figures, we see that compared with the SC-PRSM type methods, both PSC-PRSM-EXP and PSC-PRSM-IMP are more efficient.

(a) Evolution of ISNR with respect to the computing time. (b) Evolution of objective function value with respect to the computing time.

Figure 3.8: Comparison results for Shepp-Logan Phantom.

3.4.2.4 Comparison with ADMM

We now compare the PSC-PRSM with ADMM for the computational tomography model \((3.4.13)\). First, we list the iterative scheme when ADMM is applied to \((3.4.14)\):

\[
\begin{align*}
    u^{k+1} & = \arg \min \left\{ \frac{1}{2} \| Ru - b \|_2^2 + \frac{\beta}{2} \| \nabla u - v^k - \frac{1}{\beta} \lambda^k \|_2^2 \right\}, \\
    v^{k+1} & = \arg \min \left\{ \tau \| v \|_1 + \frac{\beta}{2} \| \nabla u^{k+1} - v - \frac{1}{\beta} \lambda^k \|_2^2 \right\}, \\
    \lambda^{k+1} & = \lambda^k + \delta - \beta (\nabla u^{k+1} - v^{k+1}).
\end{align*}
\]

Like SC-PRSM, we need some iterative methods to approximately solve the \(u\)-subproblem. In the following, for comparison purpose, we will test the scenarios where the \(u\)-subproblem in \((3.4.21)\) is solved by the Barzilai-Borwein (BB) gradient method in \([10]\) and the preconditioned conjugate gradient (PCG) method. Accordingly, they are denoted by “ADMM+BB” and “ADMM+PCG”, respectively. Again, different accura-
The scalar \( \tau = 0.06 \) in the model (3.4.13). The initial points are taken as the FBP image, i.e. \( R^{-1}b \), for all methods to be tested, where \( R^{-1} \) is the inverse Radon transform and \( b \) is the measured projection data. The parameters settings for the test methods are listed below.

- For the PSC-PRSM-EXP and PSC-PRSM-IMP, the parameter setting is the same as that in Section 3.4.2.3.
- For ADMM+PCG and ADMM+BB, we choose \( \beta = 0.1 \) in (3.4.21). The maximum iteration number of PCG and BB is set to be 50. Again, here the maximum iteration number is a safe bound to ensure the subproblem is solved with the specific accuracy \( (10^{-3}, 10^{-4}, 10^{-5}) \) by PCG and BB methods.

In Figures 3.10 and 3.11 we plot the evolution of the ISNR value and the objective function value with respect to the computing time for the Shepp-Logan phantom and magnetic resonance images, respectively. From these two figures, we see that

(a) Evolution of ISNR with respect to the computing time.

(b) Evolution of objective function value with respect to the computing time.

Figure 3.9: Comparison results for Magnetic Resonance Image.
compared with the SC-PRSM type methods, both PSC-PRSM-EXP and PSC-PRSM-IMP are more efficient.

In Tables 3.3 and 3.4, we report the computing time in seconds for all the methods under test to achieve some given ISNR values. The reconstructed Shepp-Logan phantom image (at the 100th second) and magnetic resonance image (at the 40th second) by all test methods are shown in Figure 3.12 and 3.13.

Figure 3.10: Comparison results of test methods for Shepp-Logan Phantom.
(a) Evolution of ISNR with respect to the computing time.

(b) Evolution of objective function value with respect to the computing time.

Figure 3.11: Comparison results of test methods for Magnetic Resonance Image.

Table 3.3: Comparison of computing time to achieve different SNR on Phantom.

<table>
<thead>
<tr>
<th>ISNR</th>
<th>8</th>
<th>12</th>
<th>16</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSC-PRSM-EXP</td>
<td>0.43</td>
<td>1.87</td>
<td>4.40</td>
<td>19.32</td>
</tr>
<tr>
<td>PSC-PRSM-IMP</td>
<td>0.65</td>
<td>3.64</td>
<td>9.58</td>
<td>48.78</td>
</tr>
<tr>
<td>SC-PRSM+Chol</td>
<td>33.65</td>
<td>244.66</td>
<td>765.19</td>
<td>2605.1</td>
</tr>
<tr>
<td>SC-PRSM+BB(10^{-4})</td>
<td>2.11</td>
<td>7.83</td>
<td>19.14</td>
<td>103.51</td>
</tr>
<tr>
<td>SC-PRSM+PCG(10^{-4})</td>
<td>0.69</td>
<td>3.43</td>
<td>8.91</td>
<td>44.24</td>
</tr>
<tr>
<td>ADMM+BB(10^{-4})</td>
<td>3.19</td>
<td>15.96</td>
<td>36.73</td>
<td>96.95</td>
</tr>
<tr>
<td>ADMM+PCG(10^{-4})</td>
<td>0.88</td>
<td>4.42</td>
<td>12.23</td>
<td>65.00</td>
</tr>
</tbody>
</table>
Table 3.4: Comparison of computing time to achieve different SNR on MRI.

<table>
<thead>
<tr>
<th>Method</th>
<th>ISNR 11</th>
<th>ISNR 12</th>
<th>ISNR 13</th>
<th>ISNR 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSC-PRSM+EXP</td>
<td>0.44</td>
<td>0.68</td>
<td>1.12</td>
<td>2.36</td>
</tr>
<tr>
<td>PSC-PRSM+IMP</td>
<td>0.64</td>
<td>0.99</td>
<td>1.84</td>
<td>4.21</td>
</tr>
<tr>
<td>SC-PRSM+Chol</td>
<td>32.60</td>
<td>51.83</td>
<td>97.17</td>
<td>232.97</td>
</tr>
<tr>
<td>SC-PRSM+BB(10^{-4})</td>
<td>3.52</td>
<td>4.87</td>
<td>7.23</td>
<td>13.01</td>
</tr>
<tr>
<td>SC-PRSM+PCG(10^{-4})</td>
<td>1.07</td>
<td>1.54</td>
<td>2.42</td>
<td>4.83</td>
</tr>
<tr>
<td>ADMM+BB(10^{-4})</td>
<td>5.66</td>
<td>8.97</td>
<td>16.57</td>
<td>29.87</td>
</tr>
<tr>
<td>ADMM+PCG(10^{-4})</td>
<td>1.51</td>
<td>2.21</td>
<td>3.86</td>
<td>7.79</td>
</tr>
</tbody>
</table>

Figure 3.12: Images reconstructed by PSC-PRSM methods.
Figure 3.13: Images reconstructed by test methods.
Chapter 4

A parallel operator splitting algorithm for (P3)

In this chapter, we focus on the model (P3) and propose a parallel operator splitting algorithm, where all subproblems can have closed-form solutions if the resolvent operators (4.2.2) of \( \partial \theta_i \), \( i = 1, 2, 3 \) can be easily evaluated. Moreover, the variables are updated in a simultaneous way, which makes the algorithm need less CPU time than that of a sequential one in a parallel implementation. Under very mild conditions, we prove the global convergence of the proposed algorithm. Finally, we apply this algorithm on the Retinex problem in imaging sciences and report some promising numerical results.

4.1 Algorithm

For succinctness of the presentation in subsequent analysis, we first introduce some notations here.

We denote \( x := (x_1^T, x_2^T, x_3^T)^T \), \( A := (A_1, A_2, A_3) \) and \( \mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3 \). Then model (P3) can be rewritten as

\[
\min \left\{ \sum_{i=1}^{3} \theta_i(x_i) \mid Ax = b, \ x \in \mathcal{X} \right\}. \quad (4.1.1)
\]

For a given \( x \in \mathcal{X} \), let \( \xi_i \in \partial \theta_i(x_i) \), \( i = 1, 2, 3 \) and denote \( \xi := (\xi_1^T, \xi_2^T, \xi_3^T)^T \). Let
λ ∈ ℜ^m be a Lagrangian multiplier associated with the linear constraint in (4.1.1). For any given β > 0, we define

\[ E_β(x, λ) = \begin{pmatrix} e_β(x, λ) \\ β(Ax - b) \end{pmatrix} := \begin{pmatrix} x - P_X\{x - β[ξ - A^Tλ]\} \\ β(Ax - b) \end{pmatrix}, \] (4.1.2)

where for a given nonempty closed convex subset Ω of ℜ^n, \( P_Ω \) denotes the orthogonal projection of a vector in ℜ^n onto Ω,

\[ P_Ω\{x\} := \operatorname{arg\,min}\{\|x - y\|_2 \mid x \in Ω\}. \]

A well known property (see e.g., [58]) of the projection operator \( P_Ω\{·\} \) is

\[ (w - P_Ω\{w\})^T(w' - P_Ω\{w\}) ≤ 0, \quad ∀w ∈ ℜ^n, \; ∀w' ∈ Ω. \] (4.1.3)

We have the following lemma.

**Lemma 4.1.1** \( x ∈ X \) is a solution of (P3) if and only if there is \( λ ∈ ℜ^m \), and there are subgradients \( ξ_1 ∈ ∂θ_1(x_1), ξ_2 ∈ ∂θ_2(x_2) \) and \( ξ_3 ∈ ∂θ_3(x_3) \), such that \( (x, λ) \) is a zero point of \( E_β(x, λ) \), for any arbitrary but given \( β > 0 \).

**Proof:** The first-order optimality condition for (P3) is that there exists some \( λ ∈ ℜ^m \) such that

\[ \begin{cases} 0 ∈ ∂θ_1(x_1) - A_1^Tλ + N_{Ω_1}(x_1), \\
0 ∈ ∂θ_2(x_2) - A_2^Tλ + N_{Ω_2}(x_2), \\
0 ∈ ∂θ_3(x_3) - A_3^Tλ + N_{Ω_3}(x_3), \\
0 = A_1x_1 + A_2x_2 + A_3x_3 - b, \end{cases} \] (4.1.4)

where \( N_{Ω} \) is the normal cone operator to \( Ω \) defined as

\[ N_{Ω}(x) := \begin{cases} \{w \mid (y - x)^T w ≤ 0, \; ∀y ∈ Ω\}, & \text{if} \; x ∈ Ω, \\
0, & \text{otherwise}. \end{cases} \]

The assertion then follows immediately by combining (4.1.4) and (4.1.3).

We are now ready to present our algorithm for solving model (P3).

**Algorithm 4.2** A Parallel Operator Splitting Algorithm for (P3).
S0. Choose an arbitrary initial point \((x^0, \lambda^0) \in X \times R^m, \xi_i^0 \in \partial \theta_i(x_i^0), i = 1, 2, 3, \epsilon > 0, \gamma \in (0, 2), \text{ and } \beta_0 > 0. \) Set \(k := 0.\)

S1. Set
\[
\tilde{\lambda}^k = \lambda^k - \beta_k (Ax^k - b)
\]
Compute
\[
e^k = e_{\beta_k}(x^k, \tilde{\lambda}^k)
\]
using (4.1.2); and set
\[
\alpha_k := \frac{\varphi_k}{\psi_k},
\]
where \(\varphi_k\) and \(\psi_k\) are calculated by (4.2.8) and (4.2.9), respectively.

S2. For each \(i = 1, 2, 3,\) solve the following \((x_i^{k+1}, \xi_i^{k+1})\)-subproblems simultaneously:
\[
x_i^{k+1} = \arg \min \{\theta_i(x_i) + \frac{1}{2\beta_k} \|x_i - (x_i^k + \beta_k \xi_i^k - \gamma \alpha_k e_i^k)\|^2\},
\]
and
\[
\xi_i^{k+1} = \xi_i^k + \frac{1}{\beta_k} (x_i^k - x_i^{k+1} - \gamma \alpha_k e_i^k).
\]

S3. Update the multiplier \(\lambda^{k+1}\) via
\[
\lambda^{k+1} = \lambda^k - \gamma \alpha_k \beta_k (A(x^k - e^k) - b)
\]
and the parameter \(\beta_{k+1}\) via
\[
\beta_{k+1} = \rho_k \beta_k, \quad \text{where } \rho_k \in (0, 1].
\]

S4. If
\[
\|Ax^{k+1} - b\| + \|e^k\| \leq \epsilon,
\]
then stop; Otherwise, go to the next step. Set \(k := k + 1,\) and go to S1.

In the Algorithm 4.2 at \(k\)-th iteration,
\[
\varphi_k = \|e^k\|^2 + \|\tilde{\lambda}^k - \lambda^k\|^2 - \beta_k (\lambda^k - \tilde{\lambda}^k)^T Ae^k
\]
and
\[
\psi_k = \|e^k\|^2 + \|\tilde{\lambda}^k - \lambda^k\|^2 + \beta_k^2 \|A(x^k - e^k) - b\|^2.
\]
Remark 4.2.1 Note that at Step S2, the individual $x_i$-subproblems are fully implemented in a parallel manner; as a consequence, the CPU time required is approximately equal to the time consumed in solving the most difficult subproblems.

Remark 4.2.2 It follows from the updating schemes for $\xi$ that

$$\xi^k_1 \in \partial \theta_1(x^k_1), \ \xi^k_2 \in \partial \theta_2(x^k_2), \text{ and } \xi^k_3 \in \partial \theta_3(x^k_3), \text{ for all } k \geq 0.$$

Hence, from Lemma 4.1.1 we assert that if

$$\|Ax^{k+1} - b\| + \|e^k\| = 0, \quad (4.2.10)$$

then $x^k$ is a solution of $(P_3)$. As a consequence, if the left hand side of $(4.2.10)$ is small enough, we can consider $x^k$ as an approximate solution of $(P_3)$, which justifies the stopping criterion of the algorithm.

The parameter $\alpha_k$ plays the role of the “step size”. The following lemma shows that the step size is uniformly bounded away from zero.

Lemma 4.2.3 Suppose that $\{\beta_k\} \subseteq [\beta_{\text{min}}, \beta_{\text{max}}] \subset (0, \frac{2}{\sqrt{3}c})$, where $c := \max\{\|A_1\|, \|A_2\|, \|A_3\|\}$. Then there exists some positive number $\alpha_{\text{min}} > 0$, such that $\alpha_k \geq \alpha_{\text{min}}$ for all $k \geq 0$.

Proof. To prove the assertion, we need to bound the numerator $\varphi_k$ from below and the denominator $\psi_k$ from above. For any two vectors $a, b \in \mathbb{R}^n$ and for any $\tau > 0$, the following inequality holds

$$2a^Tb \leq \tau \|a\|^2 + \frac{1}{\tau} \|b\|^2.$$

Consequently, for any $\tau > 0,

$$2\beta_k(\lambda^k - \bar{\lambda}^k)^T(Ae^k) \leq 3\tau \|\lambda^k - \bar{\lambda}^k\|^2 + \frac{1}{\tau} (\|\beta_k Ae^k\|^2) \leq 3\tau \|\lambda^k - \bar{\lambda}^k\|^2 + \frac{\beta^2_k}{\tau} (\|A\|^2 \|e^k\|^2) \leq 3\tau \|\lambda^k - \bar{\lambda}^k\|^2 + \frac{\beta^2_k c^2}{\tau} (\|e^k\|^2). \quad (4.2.11)$$
Inserting (4.2.11) into (4.2.8), we obtain
\[ \varphi_k \geq \left( 1 - \frac{3\tau}{2} \right) \|\lambda^k - \bar{\lambda}^k\|^2 + \left( 1 - \frac{\beta_k^2 c^2}{2\tau} \right) (\|e^k\|^2). \] (4.2.12)

We now consider the denominator \( \psi_k \). Also from the Cauchy-Schwarz inequality, we obtain
\[ \|(A(x^k - e^k) - b)\|^2 \leq 4 \left( \|Ax^k - b\|^2 + \|Ae^k\|^2 \right). \]

Hence, it follows from (4.2.9) that
\[ \psi_k \leq (\|e^k\|^2 + \|\lambda^k - \bar{\lambda}^k\|^2) + 4 \beta_k^2 \left( \|Ax^k - b\|^2 + \|Ae^k\|^2 \right) \leq \Delta (\|e^k\|^2 + \|\lambda^k - \bar{\lambda}^k\|^2) \] (4.2.13)

where
\[ \Delta := \max \{ 5, \ 1 + 4\beta_{\text{max}}^2 \|A_1\|^2, \ 1 + 4\beta_{\text{max}}^2 \|A_2\|^2, \ 1 + 4\beta_{\text{max}}^2 \|A_3\|^2 \}. \]

Since (4.2.12) holds for any \( \tau > 0 \), it holds for \( \tau = \beta_k c / \sqrt{3} \), resulting in
\[ \varphi_k \geq \left( 1 - \frac{\sqrt{3}\beta_k c}{2} \right) (\|\lambda^k - \bar{\lambda}^k\|^2 + \|e^k\|^2) \geq \left( 1 - \frac{\sqrt{3}\beta_{\text{max}} c}{2} \right) (\|\lambda^k - \bar{\lambda}^k\|^2 + \|e^k\|^2). \] (4.2.14)

Combining (4.2.13) and (4.2.14), we get
\[ \alpha_k \geq \left( 1 - \frac{\sqrt{3}\beta_{\text{max}} c}{2} \right) / \Delta =: \alpha_{\text{min}} > 0. \] (4.2.15)

This completes the proof. \( \square \)

4.3 Global convergence

We now analyze the convergence of the proposed Algorithm 4.2 and first present the following lemma.

**Lemma 4.3.1** For any \((x^*, \lambda^*) \in \Omega_3^*\), where \( \Omega_3^* \) is the solution set of VI\((\Omega_3, F_3, \theta)\) (1.6.3)-(1.6.3). The sequence \( \{ (x^k, \lambda^k) \} \) generated by the Algorithm 4.2 satisfies
\[ \varphi_k \leq (x^k - x^* + \beta_k (\xi^k - \xi^*))^T e^k + (\lambda^k - \lambda^*)^T ((\lambda^k - \bar{\lambda}^k) - \beta_k A e^k) \] (4.3.1)
Proof: Since \((x^*, \lambda^*)\) is a solution of (4.1.1), it follows from Lemma 4.1.1 that

\[
\begin{align*}
0 &\in \partial \theta_1(x_1^*) - A_1^T \lambda^* + N_{X_1}(x_1^*), \\
0 &\in \partial \theta_2(x_2^*) - A_2^T \lambda^* + N_{X_2}(x_2^*), \\
0 &\in \partial \theta_3(x_3^*) - A_3^T \lambda^* + N_{X_3}(x_3^*), \\
0 &\in A_1 x_1^* + A_2 x_2^* + A_3 x_3^* - b.
\end{align*}
\]

(4.3.2)

Then, from the definition of the normal cone, there exist \(\xi_1^* \in \partial \theta_1(x_1^*)\), \(\xi_2^* \in \partial \theta_2(x_2^*)\), and \(\xi_3^* \in \partial \theta_3(x_3^*)\), such that

\[\frac{3}{3} (\xi^* - A^T \lambda^*) \geq 0, \quad \forall x' \in X. \quad (4.3.3)\]

Setting

\[x' = P_X[x^k - \beta_k(\xi^k - A^T \bar{\lambda}^k)] = x^k - e^k\]

in (4.3.3), we have

\[\frac{3}{3} (\frac{3}{3} - e^k - x^*)^T (\xi^* - A^T \lambda^*) \geq 0. \quad (4.3.4)\]

On the other hand, setting \(w := (x^k - \beta_k(\xi^k - A^T \bar{\lambda}^k))\), and \(w' := x^*\) in (4.3.3), we obtain

\[\frac{3}{3} (\frac{3}{3} - e^k - x^*)^T (e^k - \beta_k(\xi^k - A^T \bar{\lambda}^k)) \geq 0. \quad (4.3.5)\]

Adding (4.3.4) and (4.3.5) yields

\[\frac{3}{3} (\frac{3}{3} - e^k - x^*)^T (e^k - \beta_k(\xi^k - \xi^*) + \beta_k A^T (\bar{\lambda}^k - \lambda^*)) \geq 0\]

i.e.,

\[
\frac{3}{3} (\frac{3}{3} - x^* + \beta_k(\xi^k - \xi^*))^T e^k \\
\geq ||e^k||^2 + \beta_k (x^k - x^*)^T (\xi^k - \xi^*) - \beta_k [A(x^k - e^k) - b]^T (\bar{\lambda}^k - \lambda^*) \\
\geq ||e^k||^2 - \beta_k [A(x^k - e^k) - b]^T (\bar{\lambda}^k - \lambda^*)
\]

(4.3.6)

where in the first inequality we use the fact that \(Ax^* = b\), and the second inequality follows from the monotonicity of the mappings \(\partial \theta_1\), \(\partial \theta_2\) and \(\partial \theta_3\). The relation (4.2.1)
implies that
\[
\beta_k [A x^k - A e^k - b]^T (\lambda^k - \lambda^*)
= \beta_k [A x^k - A e^k - b]^T (\lambda^k - \lambda^*) + \beta_k [A x^k - A e^k - b]^T (\bar{\lambda}^k - \lambda^k)
= \beta_k [A x^k - A e^k - b]^T (\lambda^k - \lambda^*) - \|\bar{\lambda}^k - \lambda^k\|^2 + \beta_k (A e^k)^T (\lambda^k - \bar{\lambda}^k)
= [(\lambda^k - \bar{\lambda}^k) - \beta_k A e^k]^T (\lambda^k - \lambda^*) - \|\bar{\lambda}^k - \lambda^k\|^2 + \beta_k (A e^k)^T (\lambda^k - \bar{\lambda}^k)
\]

Inserting (4.3.7) into (4.3.8), we obtain the assertion immediately from (4.2.8). \qed

Lemma 4.3.2 For any \((x^*, \lambda^*) \in \Omega_3^*\), it holds
\[
\left\| x^{k+1} - x^* + \beta_{k+1}(\xi^{k+1} - \xi^*) \right\|_{\lambda^{k+1} - \lambda^*}^2 \leq \left\| x^k - x^* + \beta_k(\xi^k - \xi^*) \right\|_{\lambda^k - \lambda^*}^2 - \gamma(2 - \gamma)\alpha_k\varphi_k. \quad (4.3.8)
\]

Proof: The schemes in updating \(\xi_1, \xi_2\) and \(\xi_3\), i.e., (4.2.5), lead to the following relationships
\[
x^{k+1} + \beta_k\xi^{k+1} = x^k + \beta_k\xi^k - \gamma\alpha_k e^k,
\]
which together with (4.2.6) implies that
\[
\left\| x^{k+1} - x^* + \beta_k(\xi^{k+1} - \xi^*) \right\|_{\lambda^{k+1} - \lambda^*}^2 = \left\| x^k - x^* + \beta_k(\xi^k - \xi^*) \right\|_{\lambda^k - \lambda^*}^2 + \gamma^2\alpha_k^2\|e^k\|^2 + \gamma^2\alpha_k^2\beta_k^2\|A(x^k - e^k) - b\|^2
\]
\[
-2\gamma\alpha_k(x^k - x^* + \beta_k(\xi^k - \xi^*))^T e^k - 2\gamma\alpha_k\beta_k(\lambda^k - \lambda^*)^T (A(x^k - e^k) - b)
\]
\[
\leq \left\| x^k - x^* + \beta_k(\xi^k - \xi^*) \right\|_{\lambda^k - \lambda^*}^2 + \gamma^2\alpha_k^2\|e^k\|^2 + \gamma^2\alpha_k^2\beta_k^2\|A(x^k - e^k) - b\|^2 - 2\gamma\alpha_k\varphi_k
\]
\[
= \left\| x^k - x^* + \beta_k(\xi^k - \xi^*) \right\|_{\lambda^k - \lambda^*}^2 - \gamma(2 - \gamma)\alpha_k\varphi_k, \quad (4.3.9)
\]
where the first inequality follows from Lemma 4.3.1 and the last equality from the definition of \(\varphi_k\).
From the monotonicity of the mappings $\partial \theta_1$, $\partial \theta_2$, and $\partial \theta_3$, and the nonincreasing property of $\{\beta_k\}$, we have

$$\left\| x^{k+1} - x^* + \beta_{k+1}(\xi^{k+1} - \xi^*) \right\|_{\lambda^{k+1} - \lambda^*}^2 \leq \left\| x^{k+1} - x^* + \beta_k(\xi^{k+1} - \xi^*) \right\|_{\lambda^{k+1} - \lambda^*}^2. \quad (4.3.10)$$

Combining (4.3.9) and (4.3.10), we complete this proof. $\square$

**Theorem 4.3.3** Suppose that $\{\beta_k\} \subseteq [\beta_{\min}, \beta_{\max}] \subset (0, \frac{2}{\sqrt{3c}})$ is a nonincreasing sequence of parameters, the iterative sequence $\{(x^k, \lambda^k)\}$ generated by Algorithm 4.2 converges to a solution of (P3).

**Proof:** Combining (4.3.9), (4.2.15), and (4.3.10), we have

$$\left\| x^{k+1} - x^* + \beta_{k+1}(\xi^{k+1} - \xi^*) \right\|_{\lambda^{k+1} - \lambda^*}^2 \leq \left\| x^k - x^* + \beta_k(\xi^k - \xi^*) \right\|_{\lambda^k - \lambda^*}^2 - \gamma(2 - \gamma)\alpha_{\min}\varphi_k. \quad (4.3.11)$$

Since $\gamma \in (0, 2), \alpha_{\min} > 0$, and $\varphi_k \geq 0$, we obtain

$$\left\| x^{k+1} - x^* + \beta_{k+1}(\xi^{k+1} - \xi^*) \right\|_{\lambda^{k+1} - \lambda^*}^2 \leq \left\| x^k - x^* + \beta_k(\xi^k - \xi^*) \right\|_{\lambda^k - \lambda^*}^2, \quad (4.3.12)$$

which means that the generated sequence $\{(x^k, \lambda^k; \xi^k)\}$ is bounded. Rearranging terms in (4.3.12), it follows that

$$\gamma(2 - \gamma)\alpha_{\min}\varphi_k \leq \left\| x^k - x^* + \beta_k(\xi^k - \xi^*) \right\|_{\lambda^k - \lambda^*}^2 - \left\| x^{k+1} - x^* + \beta_{k+1}(\xi^{k+1} - \xi^*) \right\|_{\lambda^{k+1} - \lambda^*}^2.$$

Summing both sides of the above inequality for all $k \geq 0$, we have

$$\sum_{k=0}^{\infty} \gamma(2 - \gamma)\alpha_{\min}\varphi_k \leq \left\| x^0 - x^* + \beta_0(\xi^0 - \xi^*) \right\|_{\lambda^0 - \lambda^*}^2 < \infty,$$

81
which, together with the assumption that $\gamma \in (0, 2)$ and the fact that $\alpha_{\text{min}} > 0$, implies that

$$\lim_{k \to \infty} \varphi_k = 0.$$ 

Hence, it follows from (4.2.14) that

$$\lim_{k \to \infty} \|\lambda^k - \bar{\lambda}^k\|^2 = \lim_{k \to \infty} \|e^k\|^2 = 0,$$

and further

$$\lim_{k \to \infty} \|E_\beta(x^k, \lambda^k)\|^2 = 0.$$

For any fixed $(x, \lambda)$, $\|E_\beta(x, \lambda)\|$ is an increasing function of $\beta$ [53], we have

$$\lim_{k \to \infty} \|E_{\beta_{\text{min}}}(x^k, \lambda^k)\|^2 = 0. \quad (4.3.13)$$

Since the sequence $\{(x^k, \lambda^k)\}$ is bounded, it has at least one cluster point. Let $(x^\infty, \lambda^\infty)$ in $\mathcal{X} \times \mathcal{R}^m$ be an arbitrary cluster point and let $\{(x^{kj}, \lambda^{kj})\}$ be the corresponding subsequence converging to it. Then, taking limit along this subsequence in (4.3.13), and using the upper-semicontinuity of the partial subdifferential operators, we have

$$\|E_{\beta_{\text{min}}}(x^\infty, \lambda^\infty)\|^2 = \|E_{\beta_{\text{min}}} (\lim_{j \to \infty} x^{kj}, \lim_{j \to \infty} \lambda^{kj})\|^2 = 0,$$

which means that $(x^\infty)$ is a solution of the optimization problem (P3), and $\lambda^\infty$ is the optimal Lagrangian multiplier associated to the linear constraint. Since $(x^*, \lambda^*)$ is an arbitrary solution, we can replace it with $(x^\infty, \lambda^\infty)$ in (4.3.12) and get

$$\left\| x^{k+1} - x^\infty + \beta_{k+1}(\xi^{k+1} - \xi^\infty) \right\|_{\lambda^{k+1} - \lambda^\infty} \leq \left\| x^k - x^\infty + \beta_k(\xi^k - \xi^\infty) \right\|_{\lambda^k - \lambda^\infty},$$

and therefore the whole sequence $\{(x^k, \lambda^k)\}$ converges to $(x^\infty, \lambda^\infty)$. This completes the proof. \hfill \Box

### 4.4 Application to Retinex

In digital image processing, the cameras records information based on the light reflected by objectives. Thus, the digital image information acquired by the cameras is
affected by the intensity of light. On the other hand, human beings can automatically
discount the variation of the illumination. This feature, called color constancy, en-
ures that the perceived color remains constant under varying illumination condition.
To simulate the human visual system, Land and McCann [99] proposed Retinex
theory. The word “Retinex” is a blend of “retina” and “cortex”, because both the
eye and brain are involved in the system. Since its appearance, Retinex theory has
inspired a wide range of implementation, improvements and discussions, see [13, 14,
19, 40, 41, 44, 61, 90, 92, 93, 94, 107, 109, 113, 114, 117, 127, 127].

4.4.1 A new model

As in [117], we construct and discuss our model based on a single channel image, just
for the purpose of simplicity.

Let $S$ be the image defined on $\Omega$. The image intensity is proportional to the reflectance
function $R$ and the illumination function $L$:

$$ S = L \cdot R. \quad (4.4.1) $$

The reflectance $R$ can be perceived by HVS while the illumination $L$ is automatically
discounted by HVS. In practice, it is assumed that $0 < R < 1$ (reflectivity) and
$0 < L < \infty$ (illumination effect). This assumption implies that

$$ L > S > 0. $$

Usually, the product form (4.4.1) is converted into the sum form [117]

$$ s = l - r, $$

where

$$ s = \log(S), \quad l = \log(L), \quad \text{and} \quad r = -\log(R). $$

Here we set a minus before $\log(R)$ for the purpose of ensuring $r > 0$, since we assume
that $0 < R < 1$. The task is then to separate $l$ and $r$, from a given $s$. Define

$$ E(r, l) = \int_{\Omega} |Dr| + \int_{\Omega} \frac{\alpha}{2} |\nabla l|^2 dx, \quad (4.4.2) $$
where $\alpha$ is a positive regularization parameter. Based on the assumptions from Retinex theory, the functions of the two terms in (4.4.2) are

- The total variation (TV) on $r$ — to ensure that the reflect function is piecewise constant;
- The $\ell_2$ norm on $\nabla l$ — to ensure that illumination function is spatially smooth.

Using (4.4.2) as energy function, Ma and Osher [106] suggests the following minimization model

$$(r, l) = \arg \min_{(r, l) \in \Lambda} E(r, l),$$

where the feasible set $\Lambda$ is defined by

$$\Lambda = \{(r, l) \mid (r, l) \in BV(\Omega) \times W^{1,2}(\Omega), r \geq 0, l \geq s\}.$$  

Introducing a data fidelity term, Ng and Wang [117] defined a new energy function

$$E'(r, l) = \int_\Omega |Dr|^2 dx + \alpha \int_\Omega \nabla l^2 dx + \beta \int_\Omega (l - r - s)^2 dx + \mu \int_\Omega l^2 dx,$$

where $\alpha, \beta$ and $\mu$ are positive numbers for regularization. Ng and Wang proposed to solve the optimization problem (4.4.3) with the energy function (4.4.5) and feasible set (4.4.4):

$$(r, l) = \arg \min_{(r, l) \in \Lambda} E'(r, l).$$

Note that in their function (4.4.5), the term $\int_\Omega (l - r - s)^2 dx$ is used for fidelity, while the term $\int_\Omega l^2 dx$ is only for regularization. In other words, this term is only to ensure the existence of solutions for (4.4.3) with the energy function (4.4.5).

In this paper, we propose to minimize the energy functional defined by (4.4.3), and consider the feasible region defined by

$$\Gamma = \Psi \cap Y$$

where

$$\Psi = \{(r, l) \mid (r, l) \in BV(\Omega) \times W^{1,2}(\Omega), \bar{r} \geq r \geq 0, \bar{l} \geq l \geq s\}.$$
\( \Upsilon = \{ (r, l) \mid (r, l) \in BV(\Omega) \times W^{1,2}(\Omega), \int_{\Omega} (l - r - s)^2 dx \leq \sigma^2 \} \). (4.4.9)

The parameter \( \sigma^2 \) in (4.4.9) is an estimate of the variance of the noise in the image \( s \); both \( \bar{l} \) and \( \bar{r} \) in (4.4.8) are two positive parameters. That is, our model considered in this paper is

\[
\min_{(r, l) \in \Gamma} \int_{\Omega} |Dr| + \frac{\alpha}{2} |\nabla l|^2 dx. \tag{4.4.10}
\]

Comparing our model with the model of Ng and Wang \([117]\), we can find that

1. Both models contain the TV term \( \int_{\Omega} |Dr| \) and \( \int_{\Omega} |\nabla l|^2 dx \), which are based on the Retinex theory;

2. When \( \bar{l} = \bar{r} = \infty \), our model is

\[
\min_{(r, l) \in \Lambda} \int_{\Omega} |Dr| + \frac{\alpha}{2} |\nabla l|^2 dx, \text{ s.t. } \int_{\Omega} (l - r - s)^2 dx \leq \sigma^2. \tag{4.4.11}
\]

which yields the same solution as the optimization problem

\[
\min_{(r, l) \in \Lambda} \int_{\Omega} |Dr| + \frac{\alpha}{2} |\nabla l|^2 dx + \frac{\beta}{2} \int_{\Omega} (l - r - s)^2 dx, \tag{4.4.12}
\]

for a suitable choice of the Lagrange multiplier \( \beta \) (see \([27]\)). The only difference between (4.4.12) and Ng and Wang’s model (4.4.5) is the term \( \frac{\mu}{2} \int_{\Omega} l^2 dx \). In the numerical experiments of \([117]\), a smaller \( \mu \) is shown to result in better visual effects and they argued that essentially, \( \mu = 0 \) is the best choice.

The discrete model corresponding to (4.4.10) is

\[
\min \|\nabla r\|_1 + \frac{\alpha}{2} \|\nabla l\|^2, \text{ s.t. } \|l - r - s\| \leq \sigma^2, \ r \in R, \ l \in L, \tag{4.4.13}
\]

where

\[
R := \{ r \in \mathcal{R}^n \mid 0 \leq r \leq \bar{r} \} \quad \text{and} \quad L := \{ l \in \mathcal{R}^n \mid s \leq l \leq \bar{l} \},
\]

and \( \bar{r} \in \mathcal{R}^n_+ \), \( \bar{l} \in \mathcal{R}^n_+ \) are two given vectors. Note that for succinctness, the notations \( r, l, s \) are still used as their discrete representations.

We now prove the existence of solutions for model (4.4.13).

**Lemma 4.4.1** For any given \( s \in \mathcal{R}^n \), the problem (4.4.13) has at least one solution.

**Proof:** It is obvious the objective function is continuous and the feasible region is compact. Hence, the existence of solution follows immediately. \(\square\)
4.4.2 Numerical implementation

Recall that by introducing auxiliary variable $z = (t, u) \in \mathbb{R}^{2n} \times \mathbb{R}^n$, model (4.4.13) is equivalently transformed to

$$\min_{r,l,(t,u)} \| l \|_1 + \frac{\alpha}{2} \| \nabla l \|^2 \quad \text{s.t.} \quad \begin{pmatrix} \nabla \\ I \end{pmatrix} r + \begin{pmatrix} 0 \\ -I_n \end{pmatrix} l + \begin{pmatrix} -I_{2n} & 0 \\ 0 & I_n \end{pmatrix} \begin{pmatrix} t \\ u \end{pmatrix} = 0,$$

$$r \in [0, \bar{r}], \quad l \in [s, \bar{l}], \quad (t, u) \in \{(t, u) \mid \|u - s\| \leq \sigma\}.$$ (4.4.14)

Model (4.4.14) is thus a concrete application of model (P3) with $x_1 = r$, $x_2 = l$, $x_3 = (t, u)$,

$$A_1 = \begin{pmatrix} \nabla \\ I_n \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 \\ -I_n \end{pmatrix}, \quad A_3 = \begin{pmatrix} -I_{2n} & 0 \\ 0 & I_n \end{pmatrix}, \quad b = 0,$$ (4.4.15)

and

$$X_1 = [0, \bar{r}], \quad X_2 = [s, \bar{l}], \quad X_3 = \{(t, u) \mid \|u - s\| \leq \sigma\}.$$ (4.4.16)
Applying the Algorithm 4.2 to model (4.4.14), we have

\[
\begin{align*}
\lambda^k_t &= \lambda^k_t - \beta_k(\nabla r^k - t^k), \\
\lambda^k_u &= \lambda^k_u - \beta_k(r^k - t^k + u^k), \\
e^k_r &= r^k - P_{[0,1]}\{r^k - \beta_k(\xi_f^k - \nabla^T\lambda^k_t)\}, \\
e^k_t &= t^k - P_{[s,\bar{s}]}\{t^k - \beta_k(\xi_g^k + \lambda^k_u)\}, \\
e^k_i &= \beta_k(\xi_g^k + \lambda^k_u), \\
e^k_u &= u^k - P_{[|u|-s,|u|+s]}\{u^k - \beta_k(\xi_u^k - \lambda^k_u)\}, \\
\alpha_k &= \gamma \frac{\phi_k}{\psi_k}, \text{where } \phi_k \text{ and } \psi_k \text{ are calculated by (4.4.18) and (4.4.19)}, \\
r^{k+1} &= \arg\min \{\frac{1}{2\beta_k}\|r - (r^k + \beta_k \xi_f^k - \alpha_k e^k_r)\|^2\}, \\
\xi_{f}^{k+1} &= \xi_f^k + \frac{1}{\beta_k}(r^k - r^{k+1} - \alpha_k e^k_r), \\
l^{k+1} &= \arg\min \{\frac{1}{2\beta_k}\|l - (l^k + \beta_k \xi_g^k - \alpha_k e^k_t)\|^2\}, \\
\xi_{g}^{k+1} &= \xi_g^k + \frac{1}{\beta_k}(l^k - l^{k+1} - \alpha_k e^k_t), \\
u^{k+1} &= \arg\min \{\frac{1}{2\beta_k}\|u - (u^k + \beta_k \xi_u^k - \alpha_k e^k_u)\|^2\}, \\
\xi_{u}^{k+1} &= \xi_u^k + \frac{1}{\beta_k}(u^k - u^{k+1} - \alpha_k e^k_u), \\
\lambda^k_{t+1} &= \lambda^k_t + \alpha_k \beta_k (\nabla (e^k_r - r^k) - (e^k_t - t^k)), \\
\lambda^k_{u+1} &= \lambda^k_u + \alpha_k \beta_k ((e^k_r - r^k) - (e^k_t - t^k) + (e^k_u - u^k)), \\
\beta_{k+1} &= \rho_k \beta_k,
\end{align*}
\]

where

\[
\phi_k = (\|e^k_t\|^2 + \|e^k_r\|^2 + \|e^k_i\|^2 + \|e^k_u\|^2 + \|\lambda^k_t - \lambda^k_{t-1}\|^2 + \|\lambda^k_u - \lambda^k_{u-1}\|^2) \\
- \beta_k(\lambda^k_t - \lambda^k_{t-1})^T(\nabla e^k_t - e^k_i) - \beta_k(\lambda^k_u - \lambda^k_{u-1})^T(e^k_t - e^k_i + e^k_u),
\]

and

\[
\psi_k = (\|e^k_t\|^2 + \|e^k_r\|^2 + \|e^k_i\|^2 + \|e^k_u\|^2 + \|\lambda^k_t - \lambda^k_{t-1}\|^2 + \|\lambda^k_u - \lambda^k_{u-1}\|^2) \\
+ \beta_k^2\|\nabla (r^k - e^k_r) - (t^k - e^k_t)\|^2 + \beta_k^2(\|r^k - e^k_r\|^2 + (t^k - e^k_t) + (u^k - e^k_u))^2.
\]

All the subproblems in (4.4.17) render closed-form solutions:
• The vectors \( e_r, e_l, e_t, e_u \) can be uniquely obtained by simple orthogonal projections.

• \( r \)-subproblem in (4.4.17) has solution

\[
r^{k+1} = r^k + \beta \xi^k - \alpha_k e^k_r.
\]  

(4.4.20)

• \( t \)-subproblem in (4.4.17) is equivalent to the linear system

\[
(\alpha \nabla^T \nabla + \frac{1}{\beta}) t^{k+1} = \frac{1}{\beta_k} (t^k + \beta_k \xi^k_t - \alpha_k e^k_t)
\]  

(4.4.21)

and it can be diagonalized by FFT since since circulant boundary condition is exploited to the gradient operator \( \nabla \), see [79].

• \( t \)-subproblem in (4.4.17) corresponds to the soft-thresholding operation, see (1.2.7)

\[
t^{k+1} = S_{\beta_k} (t^k + \beta_k \xi^k_t - \alpha_k e^k_t).
\]  

(4.4.22)

• \( u \)-subproblem in (4.4.17) has solution

\[
u^{k+1} = u^k + \beta_k \xi^k_u - \alpha_k e^k_u.
\]  

(4.4.23)

Remark 4.4.2 We note that the \( l^k, t^k, u^k \) is not required by the \( r^{k+1} \)-subproblem in (4.4.17). Similar facts also holds for \( l^{k+1}, t^{k+1}, u^{k+1} \). Thus, the \( r, l, t, u \)-subproblems are fully eligible for a parallel computation.

4.4.3 Numerical results

In this section, we apply the Algorithm [4.2] to several test images and present the numerical results to illustrate the effectiveness of the proposed model (4.4.13) and Algorithm [4.2]. We will also implement the Algorithm 3.1 in [117] and compare the numerical results. In the following parts, we will denote Algorithm 3.1 in [117] by “Ng & Wang’s method”. All codes in our numerical experiments are written by Matlab 7.9; and are run on a desktop computer with an Intel CPU (2.66GHz) and 3.5GB RAM running Windows XP.
For both algorithms, we use the HSV Retinex model for color images. The procedure is as follows: First, we map the RGB image into the HSV space. Then we perform color balance by linearly stretching the range of the V-channel, i.e. the intensity layer of the input image to $[0, 255]$. After that, we apply the Algorithm 4.2 to the V-channel of the color image. We note that the reflectance image obtained from Retinex is usually an overenhanced image. Therefore, we add a Gamma correction operation \cite{[117]} to the illumination function. Suppose $L = \exp(l)$ is the illumination function obtained from Retinex algorithm, and $S = \exp(s)$ is the initial image; then the reflection function will be given by $R = S/L$. The Gamma correction of $L$ with an adjusting parameter $\gamma$ is defined as follows:

$$L' = W \left( \frac{L}{W} \right)^{1/\gamma}. \quad (4.4.24)$$

The usual adjusting parameter $\gamma$ is set to be 2.2 in the tests. Here $W$ is the white value (it is equal to 255 in an 8-bit image and it is also equal to 255 in the value channel of an HSV image), and the final result is then given as follows:

$$S' = L' \cdot R. \quad (4.4.25)$$

At the end, we obtain the final image by transforming it back to a RGB image with $S'$ in the V-channel.

To make the comparison between two models, we first specify the parameters. We fix $\alpha = 2, \sigma = 1, \bar{r} = s, \bar{l} = 1$ in model (4.4.13) and choose $\beta = \frac{2}{3\sqrt{3}}, \gamma = 1, \rho = 0.95$ in Algorithm 4.2 for perceptual better performance. Note that here $\beta = \frac{2}{3\sqrt{c}}$ with $c = \max\{\|A_1\|, \|A_2\|, \|A_3\|\}$. For model (4.4.5), We use the parameters suggested in \cite{[117]} in the tests: $\alpha = 1, \beta = 0.1, \mu = 10^{-5}$. In the Ng & Wang’s method, We takes $\lambda = 1, \epsilon_l = \epsilon_r = \epsilon_w = 10^{-3}$ as mentioned in \cite{[117]}. For the stopping criteria of both algorithms, in each iteration, we calculate $\|t^{k+1} - t^k\|/\|t^{k+1}\|$ and stop the iteration if it is less than the tolerance $\epsilon = 10^{-3}$.

First of all, we test a group of figures, Figure 4.1, to show the local effect of the two models. Figure 4.2 illustrates that the proposed model can strongly retain small details and does not any disorder around the edge while the results by Ng & Wang’s
model has this kind of effect and the results are also sensitive to the parameter $\mu$, which further shows that the proposition of the new model (4.4.13) is meaningful and necessary.

![Figure 4.1: The original images for the test of local effect.](image)

Figure 4.1: The original images for the test of local effect.

![Figure 4.2: (a) result by our proposed method; (b) result by Ng & Wang’s method with $\mu = 10^{-5}$; (c) result by Ng & Wang’s method with $\mu = 10^{-2}$.](image)

Figure 4.2: (a) result by our proposed method; (b) result by Ng & Wang’s method with $\mu = 10^{-5}$; (c) result by Ng & Wang’s method with $\mu = 10^{-2}$.

In the following, we present a series of examples using HSV Retinex. The figure
examples are shown in Figure 4.3.

Figure 4.3: Test examples: (a) the color wheel; (b) dark color wheel; (c) noisy paper; (d) a waving girl; (e) buildings; (f) people.

As a first example, we use the color wheel image in Figure 4.3(a) as the original image to illustrate the good performance of the proposed Algorithm 4.2 over the Ng & Wang’s method in [117]. We adjust the brightness of the original image with uniformly dark lightness in Figure 4.3(b). In Figure 4.4, we show the comparison results of the two algorithms. The residual parts in Figure 4.4(c) and (d) show that our proposed model preserves most of the color intensity after the enhancement, and it is better than the residual part by Ng & Wang’s method. In Figure 4.4, we also give a comparison by using the S-CIELAB color metric [157], which includes a spatial processing step and is useful and efficient for measuring color reproduction errors of digital images. We show the S-CIELAB errors between Figure 4.3(a) and Figure 4.4(a) (this is Ng & Wang’s method), and between Figure 4.3(a) and Figure 4.4(b) (this is the proposed method). In this test, the S-CIELAB error values have 4.43% of the image exceeding 20 units and there are 3802 pixels whose errors are larger than 20 in the Algorithm 4.2 while there are 11.46% and 9829 pixels in the Ng &
Figure 4.4: From top to bottom: (a) and (b): the enhanced images; (c) and (d): the residual images with Figure 4.3(a); (e) and (f): the histogram distributions of S-CIELAB with Figure 4.3(a); (g) and (h): the spatial distribution of the errors with Figure 4.3(a) (The areas where the difference are higher than 20 units are marked by green color).
Wang’s method. As we have the ground truth image in Figure 4.3(a), it is clear by the comparison that the proposed method provides a very good enhanced image. Finally, we remark that the computational time required for processing Figure 4.3(b) by the proposed Algorithm 4.2 and Ng & Wang’s method in [117] are 0.77 and 6.01 seconds, respectively, and thus the proposed method is quite efficient.

Figure 4.5: From top to bottom: (a) and (b): the enhanced images; (c) and (d): the histogram distributions of S-CIELAB with Figure 4.3(c); (e) and (f): the spatial distribution of the errors with with Figure 4.3(c) (The areas where the difference are higher than 25 units are marked by green color).
In the second example in Figure 4.3(c), the image is a piece of paper with texts and plots covered in shadows. By the color constancy assumption, we are supposed to get a piecewise constant background after computation. We compare the results of Ng & Wang’s method and our proposed method in Figure 4.5. In the resulting image Figure 4.5(b) by our proposed model (4.4.13), the background clutter and the shades are less obvious than of model (4.4.5); Figure 4.5(c) has more low contrasted details than Figure 4.5(a). Again, here we show the S-SCIELAB errors between Figure 4.3(c) and Figure 4.5(a) (this is Ng & Wang’s method), and between Figure 4.3(c) and Figure 4.5(b) (this is the proposed method). In this test, the S-CIELAB error values have 0.63% of the image exceeding 20 units in the proposed method, and there are 549 pixels whose errors are larger than 25 in the Algorithm 4.2, while there are 17.23% and 15088 pixels in the Ng & Wang’s method. Finally, we remark that the computational time required for processing Figure 4.3(c) by the proposed Algorithm 4.2 and Ng & Wang’s method in [117] are 0.30 and 2.59 seconds, respectively, and thus the proposed method is quite efficient.

In the third example in Figure 4.3(d), the image is the moment of a girl waving (the right part is under a shadow). The original figure is of low contrast and covered by shadow in a backlighting condition. After computation, we are supposed to remove the shadow from the single image. We compare the results of Ng & Wang’s method and our proposed method in Figure 4.6. In the resulting image Figure 4.6(b) by our proposed model (4.4.13), one can observe that the shadow in the resulting image of the proposed model is less obvious and some low contrast parts are improved better by using the model (4.4.13) than the model (4.4.5). The improved details include the texture of the hair and the hand rail. Again, here we show the S-CIELAB errors between Figure 4.3(d) and Figure 4.6(a) (this is Ng & Wang’s method), and between Figure 4.3(d) and Figure 4.6(b) (this is the proposed method). In this test, the S-CIELAB error values have 52.96% of the image exceeding 25 units in the proposed method and there are 73997 pixels whose errors are larger than 25 in the Algorithm 4.2, while there are 54.72% and 75060 pixels in the Ng & Wang’s method. Finally, we remark that the computational time required for processing Figure 4.3(d) by the
Figure 4.6: From top to bottom: (a) and (b): the enhanced images; (c) and (d): the histogram distributions of S-CIELAB with Figure 4.3(d); (e) and (f): the spatial distribution of the errors with Figure 4.3(d) (The areas where the difference are higher than 25 units are marked by green color).
The proposed Algorithm 4.2 and Ng & Wang’s method in [117] are 6.18 and 13.58 seconds, respectively and thus the proposed method is quite efficient.

![Ng & Wang’s method](image1)

![Algorithm 4.2](image2)

Figure 4.7: From top to bottom: (a) and (b): the enhanced images; (c) and (d): the histogram distributions of S-CIELAB with Figure 4.3(e); (e) and (f): the spatial distribution of the errors with with Figure 4.3(e) (The areas where the difference are higher than 25 units are marked by green color).

The forth example is the underexposed buildings in outdoor scenes. The results of the enhanced images are shown in Figure 4.7(b). We can see that the contrast of the enhanced image in Figure 4.7(b) is higher than that in Figure 4.7(a), and the
details merged in the darkness are recurred. The improved details include the area around the roof and the car. Again, here we show the S-SCIELAB errors between Figure 4.3(e) and Figure 4.7(a) (this is Ng & Wang’s method), and between Figure 4.3(e) and Figure 4.7(b) (this is the proposed method). In this test, the S-SCIELAB error values have 24.34% of the image exceeding 25 units in the proposed method and there are 39913 pixels whose errors are larger than 25 in the Algorithm 4.2, while there are 48.17% and 79002 pixels in the Ng & Wang’s method. Finally, we remark that the computational time required for processing Figure 4.3(e) by the proposed Algorithm 4.2 and Algorithm 3.1 in [117] are 1.19 and 11.41 seconds, respectively, thus the proposed method is quite efficient.

The fifth example is the image of several persons. The images enhanced by the proposed algorithm are shown in Figure 4.3(f). We can see that the enhanced results in Figure 4.8(a) are of low contrast compared with the proposed Algorithm 4.2 in the resulting image Figure 4.8(b). The shadow in the resulting image Figure 4.8(b) of the new proposed model is weaker than that of the model (4.4.5). Again, here we show the S-SCIELAB errors between Figure 4.3(f) and Figure 4.8(a) (this is Ng & Wang’s method), and between Figure 4.3(f) and Figure 4.8(b) (this is the proposed method). In this test, the S-SCIELAB error values have 20.75% of the image exceeding 15 units in the proposed method and there are 19349 pixels whose errors are larger than 15 in the Algorithm 4.2, while there are 22.07% and 20580 pixels in the Ng & Wang’s method. Finally, we remark that the computational time required for processing Figure 4.3(f) by the proposed Algorithm 4.2 and Ng & Wang’s method in [117] are 2.02 and 6.41 seconds, respectively, and thus the proposed method is quite efficient.
Figure 4.8: From top to bottom: (a) and (b): the enhanced images; (c) and (d): the histogram distributions of S-CIELAB with Figure 4.3(f); (e) and (f): the spatial distribution of the errors with Figure 4.3(f) (The areas where the difference are higher than 15 units are marked by green color).
Chapter 5

Median filter based variational models for background extraction

In this chapter, we propose some variational models for extracting static backgrounds from surveillance videos corrupted by noise, blur or both. The new models are constructed based on the fact that the matrix representation of a static background consists of identical columns; hence the idea of median filter is embedded in these models. These new models significantly differ from existing models originating from the robust principal component analysis (RPCA) in that no nuclear-norm term is involved; thus the computation of singular value decomposition can be completely avoided when solving these new models iteratively. This is an important feature since usually the dimensionality of a surveillance video model is large and so the involved SVD (which is inevitable for RPCA-based models) is very expensive computationally. We show that these new models can be fit into the abstract models (\(P1\) and \(P2\)), thus, they can be easily solved by well-developed operator splitting methods such as ADMM (1.4.7) and EADMM (1.6.7), etc. We compare the new models with their PRCA-based counterparts via testing some synthetic and real videos. Our numerical results show that compared with RPCA-based models, these median filter based variational models can extract more accurate backgrounds when the background in a surveillance video is static and numerically they can be solved much more efficiently.
5.1 Background and motivation

In video surveillance systems, fixed cameras are typically used to monitor activities at outdoor or indoor sites. Since the cameras are static and fixed, the detection of moving objects can be achieved by comparing each new frame with a representation of the scene background. This background modeling process usually forms the first step in an automated visual surveillance system. Background extraction results can be used for further image processing and analysis such as motion detection, object tracking, classification and video content processing systems, etc., see for instance [38, 43, 143]. For this reason, background extraction from surveillance video is usually required to be as fast and as simple as possible.

In the literature, there are many articles devoted to extracting backgrounds from surveillance videos. For example, the frame difference approach [111], the median filter method [42, 158], and the running Gaussian average method [125, 150]. These approaches offer acceptable accuracy while achieving a high frame rate and having limited memory requirement. Some other approaches include the kernel density estimation KDE approach [112], the Kalman filter method [131], the mixture of Gaussians method [62], the co-occurrence of image variations approach [134], and the eigen-backgrounds approach [121]. These different approaches share the same idea in estimating the background from the temporal sequence of frames, and criteria for measuring the effectiveness of these approaches include the accuracy of extracted background, the requirement of memory, the complexity of computation, etc.

Recently, there are much research based on subspace learning models for background extraction. Various models based on the principal component analysis (PCA) have been developed for background extraction by data dimension reduction, see e.g. [22, 25, 30, 142]. Those methods typically render promising results for background extraction problems. In particular, the robust principal component analysis (RPCA) model proposed in [25] has become a benchmark model for background extraction. The rationale of using RPCA is that when we stack each video frame as a column of a matrix $M$, the background component can be viewed as approximately lying
in a low-rank subspace $L$ and the foreground objects such as “moving persons” or “abnormalities” can be considered as a sparse outlier $S$. Here, we assume that the video consists of $n$ frames and each frame is a vector in $\mathcal{R}^m$. Therefore, extracting the background and foreground objects of the video $M$ amounts to the following matrix decomposition problem:

$$M = L + S,$$

(5.1.1)

where $M, L, S \in \mathcal{R}^{m \times n}$. The authors of \cite{25} prove rigourously that under some suitable assumptions, both the low-rank matrix $L$ and the sparse matrix $S$ can be recovered exactly (in statistical sense) by solving the following convex optimization model:

$$\min_{L, S \in \mathcal{R}^{m \times n}} \{ \|L\|_* + \tau \|S\|_1 \mid L + S = M \},$$

(5.1.2)

where $\| \cdot \|_*$ is the nuclear norm, $\| \cdot \|_1$ is the $\ell_1$ norm in componentwise (defined as the sum of absolute values of all entries), and $\tau > 0$ is a weighting parameter. The nuclear-norm function is the convex envelop of the rank function of a matrix and it is used to induce the low-rank component of $M$ (i.e., the background of a video); the $\ell_1$-norm function is the convex envelope of the cardinality function of a matrix and it is used to induce the sparse component of $M$ (i.e., the foreground of a video).

When there is noise in a surveillance video, we can consider an unconstrained version of the RPCA model (5.1.1)

$$\min_{L, S \in \mathcal{R}^{m \times n}} \{ \|L\|_* + \tau \|S\|_1 + \frac{\mu}{2} \|M - L - S\|_F^2 \},$$

(5.1.3)

where $\mu > 0$ is a penalty parameter and the Frobenius norm involved term is to accommodate the noise in observation, see e.g. \cite{101,122,141,152}. It has been demonstrated in \cite{122} that the $\| \cdot \|_F$ term in (5.1.3) can be employed to deal with some anomalies which cannot be classified as the background or foreground of a video, such as turbulence or Gaussian noise. Naturally, when the surveillance video is corrupted by both noise and blur, the model (5.1.3) can be generalized to the following model

$$\min_{L, S \in \mathcal{R}^{m \times n}} \{ \|L\|_* + \tau \|S\|_1 + \frac{\mu}{2} \|M - H(L + S)\|_F^2 \},$$

(5.1.4)
where $H$ is the matrix representation of a blurring operator (here, the blur is assumed to occur frame-wisely), see e.g. [79].

We restrict our discussion to the scenario where the background of a surveillance video is static. This scenario often occurs in urban surveillance environments such as banks, shopping malls, airports and train stations, see e.g. [129]. In addition, extracting a static background is useful for further background subtraction techniques which classify the type of a given pixel via subtraction operation or pixel classification (where a pixel is classified as a background or a moving object), see e.g. [56, 62, 76, 91, 143]. Moreover, effective models for extracting static backgrounds are also helpful for detecting varying backgrounds with shadings, shadows, ghosts, and more severely high-frequency illumination changes, see e.g. [129]. Therefore, research on static background extraction has attracted wide attention from various authors and inspired some efficient techniques. For example, the probability-based background extraction algorithm in [36], block-based processing algorithm in [39], the local image flow algorithm in [77], the adaptive smoothness method in [105], the Markov random field framework in [129] and the energy minimization method in [37, 151]. In the literature, the median filter technique which uses the median value over all frames of a given video as the static background matrix is among the simplest and commonly used, see e.g. [42, 43, 104, 128, 158].

Our main motivation is the fact that for a surveillance video with a static background, the background matrix should consist of identical columns, instead of being a general low-rank matrix. The background matrix thus should be represented by $\mathbf{u}1^T$ where $\mathbf{u} \in \mathcal{R}^m$ and $1$ denotes the vector in $\mathcal{R}^n$ whose entries are all 1. Accordingly, the matrix representation of a surveillance video $M$ should be decomposed as

$$M = \mathbf{u}1^T + S$$

in order to extract its static background and foreground. With this argument, we consider the following model for extracting the static background from a surveillance video:

$$\min_{\mathbf{u} \in \mathcal{R}^n} \|M - \mathbf{u}1^T\|_1.$$
Note the closed-form solution of (5.1.6) is given by

$$\text{Median}\{M_{i,1}, M_{i,2}, \cdots , M_{i,n}\}, \; i = 1, 2, \cdots , m,$$

where “Median” is defined as the median value of all entries in the set, and $M_{i,j}$ refers to the $(i,j)$-th entry of $M$. In other words, by (5.1.6) the background is extracted as the median at each pixel location of all frames of a surveillance video. This shares the same idea of the temporal median filter approach. The median filter approach can extract static backgrounds accurately and it is easy to implement.

For the surveillance video of outdoor scenes, noise and blur often occur. We thus propose median filter based denoising/deblurring models for static background extraction from surveillance video which is corrupted by noise, blur or both. In fact, by adopting the median filter approach which uses $u_1^T$ to denote the background matrix, we can consider the model

$$\min_{S \in \mathbb{R}^{m \times n}, u \in \mathbb{R}^m} \left\{ \|S\|_1 + \frac{\mu}{2} \|M - H(u_1^T + S)\|_F^2 \right\},$$

(5.1.8)

where $\mu > 0$ is a parameter and $H$ is again the matrix representation of a regular blurring operator in the literature as (5.1.4). Obviously, the model (5.1.8) includes the noisy case without blur as a special case with $H = I$:

$$\min_{S \in \mathbb{R}^{m \times n}, u \in \mathbb{R}^m} \left\{ \|S\|_1 + \frac{\mu}{2} \|M - u_1^T - S\|_F^2 \right\}.$$  

(5.1.9)

We will compare the effectiveness of background extraction between the models (5.1.2) and (5.1.5) for the noiseless and blurless scenario, (5.1.3) and (5.1.9) for the noisy but blurless scenario, and (5.1.4) and (5.1.8) for both noisy and blurred scenario of background extraction from surveillance video. Via these comparisons, we show the necessity of considering the proposed median filter based variational models for extracting static backgrounds from various corrupted surveillance videos.

As we have mentioned, the proposed median filter based variational models also enjoy the advantage over RPCA-based models that no SVD is required (because no nuclear-norm is involved) when numerical algorithms are implemented to solve them iteratively. Since usually the dimensionality of the matrix representation of a
surveillance video is very high, meaning a SVD of this matrix being very expensive, this represents an important numerical advantage of the proposed median filter based variational models. Moreover, the proposed median filter based variational models also have the advantage that fewer parameters are involved, see e.g. the comparison between (5.1.3) and (5.1.9).

5.2 Numerical implementation

Note that the model (5.1.2) is a special case of model (P2); and Models (5.1.3), (5.1.4), (5.1.8) and (5.1.9) are special cases of model (P3). In this section, we elaborate on the detail when ADMM (1.4.7) and its extended version EADMM (1.6.7) are applied to solve the mentioned RPCA-based and median filter based models.

First, applying ADMM (1.4.7) to RPCA model (5.1.2), we have the following scheme

\[
\begin{align*}
L^{k+1} &= \arg \min \left\{ \frac{1}{2} \| L \|_* + \frac{\beta}{2} \| L + S^k - M - \frac{1}{\beta} \Lambda^k \|_F^2 \right\}, \\
S^{k+1} &= \arg \min \left\{ \tau \| S \|_1 + \frac{\beta}{2} \| L^{k+1} + S - M - \frac{1}{\beta} \Lambda^k \|_F^2 \right\}, \\
\Lambda^{k+1} &= \Lambda^k - \beta \left( L^{k+1} + S^{k+1} - M \right).
\end{align*}
\]

(5.2.1)

Both the \(L\) and \(S\)-subproblems in (5.2.1) have closed-form solutions as explained below.

- The solution of the \(L\)-subproblem in (5.2.1) is given by

\[ L^{k+1} = D_{1/\beta} \left( M + \frac{\Lambda^k}{\beta} - S^k \right). \]

For any \(c > 0\) and \(T \in \mathcal{R}^{m \times n}\), the operator \(D_c\) is defined as

\[ D_c(T) := U \text{diag} \left( S_c(\Sigma) \right) V^T, \]

(5.2.2)

where \(U \in \mathcal{R}^{m \times m}\), \(V \in \mathcal{R}^{n \times n}\) and \(\Sigma \in \mathcal{R}^{m \times n}\) are obtained by the SVD of \(T\), i.e., \(T = U \Sigma V^T\); and the operator \(S_c(\cdot)\) is the soft-thresholding operation, see e.g. in [46] or (1.2.7).

- The solution of the \(S\)-subproblem in (5.2.1) is given by

\[ S^{k+1} = S_{\tau/\beta} \left( M + \frac{\Lambda^k}{\beta} - L^{k+1} \right). \]
Then, we explain the detail of applying EADMM (1.6.7) to the model (5.1.4) (The model (5.1.3) is a special case of (5.1.4); thus its detail is omitted). By introducing an auxiliary variable $K$, the model (5.1.4) can be rewritten as

$$\begin{align*}
\min & \quad \|L\|_* + \tau \|S\|_1 + \frac{\mu}{2} \|M - HK\|_F^2 \\
\text{s.t.} & \quad K = L + S,
\end{align*}$$

(5.2.3)

which is a special case of (P3) and thus EADMM (1.6.7) is applicable. We specify the iterative scheme of EADMM (1.6.7) for solving (5.1.4) as follows.

$$\begin{align*}
L^{k+1} &= \arg \min \left\{ \|L\|_* + \frac{\beta}{2} \|L + S^k - K^k - \frac{1}{\beta} \Lambda^k\|_F^2 \right\}, \\
S^{k+1} &= \arg \min \left\{ \tau \|S\|_1 + \frac{\beta}{2} \|L^{k+1} + S - K^k - \frac{1}{\beta} \Lambda^k\|_F^2 \right\}, \\
K^{k+1} &= \arg \min \left\{ \frac{\mu}{2} \|M - HK\|_F^2 + \frac{\beta}{2} \|L^{k+1} + S^{k+1} - K^k - \frac{1}{\beta} \Lambda^k\|_F^2 \right\}, \\
\Lambda^{k+1} &= \Lambda^k - \beta \left( L^{k+1} + S^{k+1} - K^k \right).
\end{align*}$$

(5.2.4)

In (5.2.4), $L$, $S$ and $K$-subproblems all have closed-form solutions as explained below.

- The solution of the $L$-subproblem in (5.2.4) is given by
  $$L^{k+1} = \mathcal{D}_{1/\beta}(K^k + \Lambda^k/\beta - S^k),$$
  where $\mathcal{D}_{1/\beta}$ is defined in (5.2.2).

- The solution of the $S$-subproblem in (5.2.4) is given by
  $$S^{k+1} = \mathcal{S}_{\tau/\beta}(K^k + \Lambda^k/\beta - L^{k+1}),$$
  where $\mathcal{S}_{\tau/\beta}$ is the soft-thresholding operation as defined in (1.2.7).

- $K$-subproblem in (5.2.4) is equivalent to the linear system
  $$\left( \mu H^T H + \beta I \right) K = \mu H^T M + \beta (L^{k+1} + S^{k+1} - \Lambda^k/\beta),$$
  (5.2.5)

which can be easily handled by FFT if periodic boundary condition is exploited to blurring matrix $H$, or by DCT if reflective boundary condition is exploited, see [79].
Now, we show how to apply EADMM (1.6.7) to solve the model in (5.1.8) (The model (5.1.9) is a special case of (5.1.8); thus its detail is omitted). By introducing an auxiliary variable

\[ U \in \Omega = \{ u_1^T \mid u \in \mathbb{R}^m, 1 \in \mathbb{R}^n \}, \]

the model (5.1.8) can be rewritten as

\[
\begin{align*}
\min & \quad \|S\|_1 + \frac{\beta}{2} \|M - HK\|_F^2 \\
\text{s.t.} & \quad K = U + S \\
& \quad U \in \Omega,
\end{align*}
\]

which is a special case of (P3). Thus, applying EADMM (1.6.7) to (5.2.6), we have

\[
\begin{aligned}
S^{k+1} &= \arg \min \left\{ \|S\|_1 + \frac{\beta}{2} \|S + u^k 1^T - K - \frac{1}{\beta} \Lambda^k\|_F^2 \right\}, \\
K^{k+1} &= \arg \min \left\{ \frac{\mu}{2} \|M - HK\|_F^2 + \frac{\beta}{2} \|S^{k+1} + u^k 1^T - K - \frac{1}{\beta} \Lambda^k\|_F^2 \right\}, \\
u^{k+1} &= \arg \left\{ \|u 1^T + S^{k+1} - K^{k+1} - \frac{1}{\beta} \Lambda^k\|_F^2 \right\}, \\
\Lambda^{k+1} &= \Lambda^k - \beta \left( u^{k+1} 1^T + S^{k+1} - K^{k+1} \right).
\end{aligned}
\]

- The solution of the S-subproblem in (5.2.7) is given by the soft-thresholding operation

\[ S^{k+1} = S_1/\beta \left( K^k + \frac{\Lambda^k}{\beta} - u^k 1^T \right). \]

- K-subproblem in (5.2.7) is equivalent to the linear system

\[ (\mu H^T H + \beta I)K = \mu H^T M + \beta (u^k 1^T + S^{k+1} - \Lambda^k/\beta), \]

which also has a closed-form solution. Note FFT (respectively, DCT) can be utilized if periodic (respectively, reflective) boundary condition is exploited to the blurring matrix \( H \), see [79] or (5.2.5).

- The \( u \)-subproblem has the solution

\[ u^{k+1} = \text{Mean} \left( K^{k+1} + \frac{\Lambda^k}{\beta} - S^{k+1} \right), \]

where the operator \( \text{Mean} : \mathbb{R}^{m \times n} \to \mathbb{R}^m \) is defined as

\[ (\text{Mean}(P))_i := \frac{1}{n} \sum_{j=1}^{n} P_{i,j}, \quad i = 1, \ldots, m, \forall P \in \mathbb{R}^{m \times n}. \]
Remark 5.2.1 Technically, the model (5.2.6) also fits the applicable scope of the original ADMM (1.4.7) in [65, 68]; but hard subproblems occur in the direct application of ADMM. Nevertheless, for some special case of (5.2.6) such as $H = I$, the application of the original ADMM is still easily implementable. Specifically, by defining
\[
Z := \begin{bmatrix} K \\ U \end{bmatrix},
\]
the model in (5.2.6) with $H = I$ can be rewritten as
\[
\begin{align*}
\min & \quad \|S\|_1 + \frac{\mu}{2} \|M - [I, 0] Z\|_F^2 \\
\text{s.t.} & \quad [I, -I] Z - S = 0.
\end{align*}
\] (5.2.8)
Consequently, the original ADMM in [65, 68] is applicable to (5.2.8). We list the resulting $S$- and $Z$-subproblems as follows

- The $S$-subproblem amounts to
\[
S^{k+1} = \arg \min \{\|S\|_1 + \frac{\beta}{2} \|K^k - U^k - S - \Lambda^k/\beta\|_F^2\}, \tag{5.2.9}
\]
whose closed-form solution is given by
\[
S^{k+1} = S_1/\beta(U^k + \Lambda^k/\beta - K^k).
\]

- The $Z$-subproblem is
\[
(K^{k+1}, U^{k+1}) = \arg \min \{\frac{\mu}{2} \|M - K\|_F^2 + \frac{\beta}{2} \|K - U - S^{k+1} - \Lambda^k/\beta\|_F^2\}. \tag{5.2.10}
\]
It follows from the first-order optimality condition of (5.2.10) that $(K, U)$-subproblem is equivalent to the following linear equations
\[
\begin{cases}
(\mu + \beta)K - \beta U = \mu M + \beta(S^{k+1} + \frac{1}{\beta} \Lambda^k), \\
(K - U - S^{k+1} - \frac{1}{\beta} \Lambda^k)1 = 0.
\end{cases}
\] (5.2.11)
Consequently, $K^{k+1}$ and $u^{k+1}$ can be solved orderly by
\[
\begin{cases}
K^{k+1} [((\mu + \beta) I + \frac{\beta}{m} 11^T] = \mu M + \left(1 - \frac{1}{m}\right) (\beta S^{k+1} + \Lambda^k), \\
u^{k+1} = \text{mean}\left(K^{k+1} - \frac{1}{\beta} \Lambda^k - S^{k+1}\right).
\end{cases}
\] (5.2.12)
The first linear system in (5.2.12) can be solved by PCG because its coefficient matrix is positive definite or can be solved by using the Sherman-Morrison-Woodbury formula [75].

But we still apply Algorithm 2 to solve the model in (5.2.6) because we also consider the case with a generic $H$.

5.3 Numerical results

In this section, we report some numerical results to verify the effectiveness of the proposed models. Our numerical results consist of the following parts:

1. Compare the RPCA model (5.1.2) (notation: RPCA) with the model (5.1.6) (notation: MED) for noiseless and blurless videos.

2. Compare the modified RPCA model (5.1.3) (notation: RPCA-i) with model (5.1.9) (notation: MED-i) for noisy but blurless videos.

3. Compare the modified RPCA model (5.1.4) (notation: RPCA-ii) with the model (5.1.8) (notation: MED-ii) for both noisy and blurred videos.

We test both synthetic videos and real videos for each case. For synthetic videos, we first choose some images which are commonly used in image reconstruction literature as backgrounds and then add some moving objects as foregrounds. Therefore, for the synthetic videos, the ground-truths of both backgrounds and foregrounds are known; and the ranks of the background matrices of these synthetic videos are exactly 1. For real videos, we test some examples available at the web site http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html. The ground-truths of backgrounds and foregrounds of these real-life videos are unknown. More specifically, the videos to be tested are listed below.

(i) **Cameraman video** — A synthetic gray scale video with eighty 256-by-256 frames; The background is the 256-by-256 ‘Cameraman.tif’ image, see the left of Figure 5.1.
(ii) **Barbara video** — A synthetic gray scale video with eighty 512-by-512 frames; The background is the 512-by-512 ‘Barbara.png’ image, see the right of Figure 5.1.

(iii) **Hall video** — A gray scale real surveillance video in the hall of an airport with two hundred 144-by-176 frames.

(iv) **Mall video** — A gray scale real surveillance video in the mall of a shopping center with one hundred 256-by-320 frames.

We show some frames of the test videos in Figure 5.2.

Figure 5.1: Backgrounds of synthetic videos. Left: Cameraman. Right: Barbara.

![Cameraman video](image1) ![Barbara video](image2)

![Hall video](image3) ![Mall video](image4)

Figure 5.2: Some frames of test videos.

In our numerical experiments, all pixel values of the test videos are re-scaled into $[0, 1]$. All the codes for implementing Algorithms 1.4.7 and 1.6.7 are written by Matlab 7.1 and run on a computer cluster equipped with 6 Core Xeon X5670 2.93GHz CPU and 64G memory.
We first give two remarks on the implementation details. 1): To reduce the computing time of SVD when ADMM (1.4.7) is applied to (5.1.2) and EADMM (1.6.7) to (5.1.3) and (5.1.4), we use the package PROPACK in [100] for only a partial SVD when solving the L-subproblem at each iteration. The strategy of choosing the number of singular values in PROPACK at each iteration is set as follows: set \(sv_0 = 100\) and update \(sv_k\) via

\[
sv_{k+1} = \begin{cases} 
sv_k + 1, & \text{if } sv_k < sv_k; \\
\min\{sv_k + \text{round}(0.05d), d\}, & \text{if } sv_k = sv_k,
\end{cases}
\]

where \(d = \min\{m, n\}\) and \(sv_k\) is the number of singular values that is larger than a given threshold. 2): In the ADMM (1.4.7) for solving the model (5.1.2), we choose the initial values as \((S^0, \Lambda^0) = (0_{m \times n}, 0_{m \times n})\); and in the EADMM (1.6.7) for solving the model (5.1.3) and (5.1.4), we choose the initial values as \((S^0, K^0, \Lambda^0) = (0_{m \times n}, M, 0_{m \times n})\). In the EADMM (1.6.7) for solving the median filter based models in (5.1.9) and (5.1.8), we set \((K^0, U^0, \Lambda^0) = (M, u^01^T, 0_{m \times n})\), where the \(u^0\) is obtained from (5.1.7).

### 5.3.1 Comparison between MED and RPCA

In this subsection, we compare the RPCA model (5.1.2) and the median filter based (5.1.6) for noiseless and blurless videos. Model (5.1.6) has a closed-form solution given by (5.1.7). For the model (5.1.2), we adopt ADMM (1.4.7) to solve it as in [155]. We first discuss how to choose parameters in both models. As aforementioned, a substantial advantage of our model (5.1.6) compared to the RPCA-based (5.1.2) is that no tuning parameter is required in (5.1.6); while there is one parameter \(\tau\) in the model (5.1.2) and another parameter \(\beta\) when ADMM (1.4.7) is implemented. We here choose \(\tau = 1/\sqrt{\max\{m, n\}}\) as in [25] and \(\beta = 0.01|\Omega|/\|M\|_1\) as suggested in [141], where \(|\Omega|\) is the cardinality of set \(\Omega\).

For the Cameraman and Barbara videos, since the ground-truths are known, the stopping criterion to implement ADMM is set as \(\text{RelErr} < 10^{-7}\), where

\[
\text{RelErr} := \max\left\{\frac{\|L^k - L^*\|_F}{\|L^*\|_F}, \frac{\|S^k - S^*\|_F}{\|S^*\|_F}\right\}.
\]
In (5.3.2), $L^*$ and $S^*$ are the ground-truths of the background and foreground matrices, respectively. As for the Hall and Mall videos without ground-truths, the iteration is terminated when $Tol < 10^{-2}$, in which the Tol is defined as

$$Tol := \max \left\{ \frac{\|L - \hat{L}\|_F}{\|L\|_F + 1}, \frac{\|S - \hat{S}\|_F}{\|S\|_F + 1} \right\},$$

where $\hat{L}$ and $\hat{S}$ are the obtained background and foreground when the iteration is terminated.

In Table 5.1, we report the numerical results for these videos. In this table, “Itr” and “Time” represent the number of iterations and computing time in seconds, respectively; “$\|S\|_0$” denotes the number of non-zero entries of $S$; and “RelErr” is the error defined in (5.3.2). Figure 5.3-5.6 show the extracted backgrounds. For the synthetic videos, both the models (5.1.2) and (5.1.6) can extract the static background exactly, i.e., the rank of the extracted backgrounds is exactly equal to 1; and for the tested real videos, the model (5.1.6) is able to extract static backgrounds while the RPCA model (5.1.2) extract inaccurate backgrounds with higher ranks. So the median filter based model (5.1.6) is more accurate than the RPCA-based model (5.1.2) in extracting static backgrounds. In terms of computational time, the model (5.1.6) has a closed-form solution and thus no iteration is required. Therefore, solving (5.1.6) is significantly faster than solving the RPCA model (5.1.2) in order to extract static backgrounds from noiseless and blurless videos.

### 5.3.2 Comparison between MED-i and RPCA-i

In this subsection, we compare the models (5.1.3) with (5.1.9) for noisy but blurless videos. Both (5.1.3) and (5.1.9) are special cases of (P3). For the details of the iterative schemes to solve models (5.1.3) and (5.1.9), we refer to (5.2.4) and (5.2.7), respectively.

We degrade each frame of the original videos by zero-mean Gaussian noise with standard variance of 0.01.

There are two parameters $\mu$ and $\tau$ in (5.1.3), and one parameter $\mu$ in (5.1.9). Moreover, the implementation of EADMM (1.6.7) requires to specify a parameter $\beta$. These
Table 5.1: Computational results on background extraction for the noise-free videos. (* refers to the machine precision.)

| Video  | Model | Itr | Time (seconds) | Rank | ||S||₀ | RelErr(Tol) |
|--------|-------|-----|----------------|------|-------|-------------|
| Cameraman | RPCA | 25  | 24.65          | 1    | 346926 | 9.27e-08    |
| Cameraman | MED  | 1   | 0.52           | 1    | 346926 | 4.91e-17    |
| RPCA  | 25  | 95.65 | 1 | 430857 | 4.55e-08 |
| Barbara | MED  | 1   | 1.7            | 1    | 430587 | 5.07e-17    |
| RPCA  | 36  | 44.24 | 40 | 1202312 | 9.17e-03 |
| Hall  | MED  | 1   | 0.18           | 1    | 4205311 | 0*          |
| RPCA  | 31  | 55.15 | 16 | 3692679 | 9.64e-03 |
| Mall  | MED  | 1   | 0.9            | 1    | 5437819 | 0*          |

Figure 5.3: Numerical comparisons of RPCA and MED on Cameraman video.
Figure 5.4: Numerical comparisons of RPCA and MED on Barbara video.

<table>
<thead>
<tr>
<th>Background</th>
<th>Model RPCA</th>
<th>Model MED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreground</td>
<td><img src="image1.png" alt="Image of Barbara video" /></td>
<td><img src="image2.png" alt="Image of Barbara video" /></td>
</tr>
</tbody>
</table>

Figure 5.5: Numerical comparisons of RPCA and MED on Hall video.

<table>
<thead>
<tr>
<th>Background</th>
<th>Model RPCA</th>
<th>Model MED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreground</td>
<td><img src="image1.png" alt="Image of Hall video" /></td>
<td><img src="image2.png" alt="Image of Hall video" /></td>
</tr>
</tbody>
</table>

Figure 5.6: Numerical comparisons of RPCA and MED on Mall video.

<table>
<thead>
<tr>
<th>Background</th>
<th>Model RPCA</th>
<th>Model MED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreground</td>
<td><img src="image1.png" alt="Image of Mall video" /></td>
<td><img src="image2.png" alt="Image of Mall video" /></td>
</tr>
</tbody>
</table>
parameters should be tuned to extract high-quality backgrounds via EADMM (1.6.7). We use the Cameraman video as the test video to tune these parameters. For models (5.1.3) and (5.1.9), we first fix $\beta$ and test some choices of $(\tau, \mu)$, and choose the pair achieving the lowest relative error (defined in (5.3.2)). Then we fix the tuned value of $(\tau, \mu)$ and test some choices of $\beta$, and choose the best one in the sense that the lowest relative error is achieved. Experiments results are illustrated in Fig 5.7 and the tuned values are listed in Table 5.2. For all the test videos in comparison of (5.1.3) and (5.1.9), we use the parameter values in Table 5.2.

Table 5.2: Parameters for models (5.1.3) and (5.1.9).

<table>
<thead>
<tr>
<th>Model (5.1.3)</th>
<th>Model (5.1.9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau$</td>
<td>$\mu$</td>
</tr>
<tr>
<td>Cameraman</td>
<td>0.002</td>
</tr>
<tr>
<td>Barbara</td>
<td>0.002</td>
</tr>
<tr>
<td>Hall</td>
<td>0.006</td>
</tr>
<tr>
<td>Mall</td>
<td>0.003</td>
</tr>
</tbody>
</table>

For the Cameraman and Barbara videos, the stopping criterion to implement EADMM (1.6.7) is set as $\text{RelErr} < 2 \times 10^{-2}$, where $L^k = u^k u^k^T$ for model (5.1.9) and hereafter. In Table 5.3 we report the numerical results for these two synthetic videos. It can be seen that both the models (5.1.3) and (5.1.9) can extract the clean background — i.e., the rank of the extracted background is exactly 1. From this table, we see that because of the absence of nuclear norm in (5.1.9), it requires much less time than the model (5.1.3) to extract the background from a video. We note in Table 5.3 that the errors of the extracted backgrounds and foregrounds by $\text{MED-i}$ and $\text{RPCA-i}$ are about the same. In Figure 5.9 we further demonstrate that the extracted backgrounds and foregrounds by $\text{RPCA-i}$ and $\text{MED-i}$ for the noisy Cameraman video are about the same. In Figure 5.8 we compare the errors of each frame of the extracted background and foreground to the ground-truths by different models. The “Error” therein is the...
Figure 5.7: Evolutions of relative error with respect to iterations for the Cameraman video. (a): RPCA $- i$ with fixed $\beta = 0.01$. (b): RPCA $- i$ with fixed $\tau = 0.5/\sqrt{m}$ and $\mu = 0.1$. (c): MED $- i$ with fixed $\beta = 1$. (d): MED $- i$ with fixed $\mu = 10^2$.

Figure 5.8: Evolutions of the errors with respect to the frames. (a) and (b): noisy Cameraman video’s background and foreground; (c) and (d): noisy Barbara video’s background and foreground.
change of error with definition as

\[ \text{Error} := \|x - x^*\|_2, \]

where \( x \) is the vector form of each frame.

Table 5.3: Computational results on background extraction for noisy videos (synthetic datasets).

<table>
<thead>
<tr>
<th>Video</th>
<th>Model</th>
<th>Itr</th>
<th>Time (seconds)</th>
<th>Rank</th>
<th>( |S|_0 )</th>
<th>RelErr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cameraman</td>
<td>RPCA-i</td>
<td>19</td>
<td>20.06</td>
<td>1</td>
<td>456363</td>
<td>2.33e-02</td>
</tr>
<tr>
<td>Cameraman</td>
<td>MED-i</td>
<td>15</td>
<td>6.04</td>
<td>1</td>
<td>346926</td>
<td>1.56e-02</td>
</tr>
<tr>
<td>Barbara</td>
<td>RPCA-i</td>
<td>20</td>
<td>77.88</td>
<td>1</td>
<td>431511</td>
<td>4.19e-02</td>
</tr>
<tr>
<td>Barbara</td>
<td>MED-i</td>
<td>13</td>
<td>20.93</td>
<td>1</td>
<td>430857</td>
<td>1.49e-02</td>
</tr>
</tbody>
</table>

Figure 5.9: Numerical comparisons of RPCA-i and MED-i on noisy Cameraman video.

For the Hall and Mall videos without ground-truths, we terminate EADMM when the extracted foregrounds by both models possess almost the same number of
nonzero elements. Table 5.4 reports the numerical results for these real-life videos. We see that for the test real-life videos, the median filter based model (5.1.9) is able to extract static backgrounds while the RPCA model (5.1.3) extracts backgrounds with higher ranks. The errors of the extracted backgrounds and foregrounds by \texttt{MED-i} and \texttt{RPCA-i} are basically equal. Also, because of the absence of nuclear norm in (5.1.9), the implementation of EADMM (1.6.7) is SVD-free and thus it is much faster than its implementation to (5.1.3). In Figure 5.10, we show that the differences of the extracted backgrounds (or foregrounds) between \texttt{RPCA-i} and \texttt{MED-i} for the Mall video are very little.

Table 5.4: Computational results on background extraction for the noisy videos (real datasets).

<table>
<thead>
<tr>
<th>Video</th>
<th>Model</th>
<th>Itr</th>
<th>Time</th>
<th>Rank</th>
<th>(|S|_0)</th>
<th>Tol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>(seconds)</td>
<td>(L)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPCA-i</td>
<td>19</td>
<td>27.02</td>
<td>10</td>
<td>367773</td>
<td>2.79e-03</td>
<td></td>
</tr>
<tr>
<td>Hall</td>
<td>MED-i</td>
<td>19</td>
<td>9.59</td>
<td>1</td>
<td>368383</td>
<td>1.28e-03</td>
</tr>
<tr>
<td>RPCA-i</td>
<td>25</td>
<td>43.59</td>
<td>4</td>
<td>778256</td>
<td>7.91e-04</td>
<td></td>
</tr>
<tr>
<td>Mall</td>
<td>MED-i</td>
<td>17</td>
<td>11.46</td>
<td>1</td>
<td>782861</td>
<td>8.17e-04</td>
</tr>
</tbody>
</table>

Figure 5.10: Numerical comparisons of RPCA-i and MED-i on noisy Mall.
5.3.3 Comparison between MED-ii and RPCA-ii

In this subsection, we compare the models (5.1.4) with (5.1.8) for noisy and blurred videos. Both (5.1.3) and (5.1.9) are special cases of (P3). For details of the iterative schemes to solve models (5.1.4) and (5.1.8), we refer to (5.2.4) and (5.2.7), respectively.

We degrade each frame of the original videos in Figure 5.11 with the motion blur and additive Gaussian white noise. The motion blur is generated by the command \texttt{fspecial('motion',9,0)} in MATLAB and the Gaussian white noise with standard variance of 0.005 is added to the blurred videos. Some frames of the corrupted synthetic and real videos are shown in Figure 5.11.

![Corrupted Videos](image)

Figure 5.11: Some frames of test degraded videos with motion blur and noise.

Likewise, the selected parameters for the test models are listed in Table 5.5. The stopping criterion of EADMM (1.6.7) for the synthetic videos is set as RelErr $< 6.5 \times 10^{-2}$ and Tol $< 10^{-2}$ for the real videos. Table 5.6 reports the numerical results of the test videos by both models. Figure 5.12-5.15 illustrate the separated backgrounds and foregrounds of the four datasets, respectively. Furthermore, we take more realistic scenarios (e.g. Gaussian blur, out-of-focus blur) into consideration. We degrade each frame of the Hall and Mall videos in Figure 5.2 with the Gaussian blur and out-of-focus blur, respectively. The Gaussian blur and out-of-focus blur are generated by the command \texttt{fspecial('gaussian',9,1)} and \texttt{fspecial('disk',5)}.
in MATLAB, respectively. The Gaussian white noise with standard variance of 0.005 is added to both blurred videos. Some frames of the corrupted real videos are shown in Figure 5.16. We still use the same group of parameters as in Table 5.5. Table 5.7 reports the numerical results. Figure 5.17-5.18 show the separated background and foreground of the real datasets. These results show that our proposed models are competitive with the benchmark RPCA models, because the background extraction by our model costs much less time to achieve the same level of accuracy.

Table 5.5: Parameters for models (5.1.4) and (5.1.8).

<table>
<thead>
<tr>
<th>Video</th>
<th>Model (5.1.4)</th>
<th>Model (5.1.8)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\tau$</td>
<td>$\mu$</td>
</tr>
<tr>
<td>Cameraman</td>
<td>0.0039</td>
<td>4.096</td>
</tr>
<tr>
<td>Barbara</td>
<td>0.002</td>
<td>2.56</td>
</tr>
<tr>
<td>Hall</td>
<td>$\frac{1}{\sqrt{m}}$</td>
<td>$0.1\sqrt{m}$</td>
</tr>
<tr>
<td>Mall</td>
<td>$\frac{1}{\sqrt{m}}$</td>
<td>$0.1\sqrt{m}$</td>
</tr>
</tbody>
</table>

Figure 5.12: Numerical comparisons of RPCA-ii and MED-ii on motion blurred and noisy Cameraman video.
Table 5.6: Computational results on background extraction for the motion blurred and noisy videos.

<table>
<thead>
<tr>
<th>Video</th>
<th>Model</th>
<th>Itr</th>
<th>Time (seconds)</th>
<th>Rank</th>
<th>$|S|_0$</th>
<th>RelErr</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPCA-ii</td>
<td>19</td>
<td>31.56</td>
<td>1</td>
<td>599191</td>
<td>6.49e-02</td>
<td></td>
</tr>
<tr>
<td>Cameraman</td>
<td>MED-ii</td>
<td>15</td>
<td>9.54</td>
<td>1</td>
<td>380997</td>
<td>6.50e-02</td>
</tr>
<tr>
<td>RPCA-ii</td>
<td>13</td>
<td>80.69</td>
<td>1</td>
<td>501506</td>
<td>6.47e-02</td>
<td></td>
</tr>
<tr>
<td>Barbara</td>
<td>MED-ii</td>
<td>11</td>
<td>25.78</td>
<td>1</td>
<td>601655</td>
<td>6.46e-02</td>
</tr>
<tr>
<td>RPCA-ii</td>
<td>22</td>
<td>41.76</td>
<td>18</td>
<td>1670409</td>
<td>9.90e-03</td>
<td></td>
</tr>
<tr>
<td>Hall</td>
<td>MED-ii</td>
<td>24</td>
<td>15.52</td>
<td>1</td>
<td>2095552</td>
<td>9.05e-03</td>
</tr>
<tr>
<td>RPCA-ii</td>
<td>16</td>
<td>46.67</td>
<td>11</td>
<td>5157749</td>
<td>9.35e-03</td>
<td></td>
</tr>
<tr>
<td>Mall</td>
<td>MED-ii</td>
<td>15</td>
<td>14.63</td>
<td>1</td>
<td>3489347</td>
<td>9.78e-03</td>
</tr>
</tbody>
</table>

Figure 5.13: Numerical comparisons of RPCA-ii and MED-ii on motion blurred and noisy Barbara video.
Figure 5.14: Numerical comparisons of RPCA-ii and MED-ii on motion blurred and noisy Hall video.

Figure 5.15: Numerical comparisons of RPCA-ii and MED-ii on motion blurred and noisy Mall video.

Figure 5.16: Some frames of blurred and noisy videos: Hall video with Gaussian blur and Mall video with out-of-focus blur.
Table 5.7: Computational results on background extraction for the blurred and noisy videos.

| Video | Model  | Itr | Time  | Rank | ||$S||_0  | RelErr |
|-------|--------|-----|-------|------|-----------|---------|
|       | RPCA-ii | 28  | 43.44 | 19   | 1032646   | 9.97e-03|
| Hall  | MED-ii  | 31  | 20.23 | 1    | 1685781   | 9.96e-03|
|       | RPCA-ii | 16  | 46.67 | 4    | 2083446   | 8.87e-03|
| Mall  | MED-ii  | 24  | 22.72 | 1    | 968898    | 9.36e-03|

Figure 5.17: Numerical comparisons of RPCA-ii and MED-ii on Gaussian blurred and noisy Hall video.
5.4 Concluding Remarks

We have proposed some median filter based variational models for static background extraction from surveillance video with noise, blur or both. In these models, the static background is represented by a matrix consisting of identical columns; thus solving these models iteratively requires no singular value decomposition. We have tested some video datasets; and the numerical results show that these new models can extract more accurate backgrounds than existing models based on the RPCA. Moreover, our numerical experiments show that the proposed median filter based variational models can be solved efficiently by the well-developed operator splitting methods in optimization literature such as the ADMM (1.4.7).

From Figure 5.12-5.15 we can see that the extracted backgrounds and foregrounds still have some artifacts. To overcome this demerit, we can further integrate the well-known TV term $\|\nabla \cdot \|_1$ in our models so as to preserve sharp edges of videos. For example, we can revise the model in (5.1.8) to

$$\min_{S \in \mathbb{R}^{m \times n}, u \in \mathbb{R}^m} \|S\|_1 + \frac{\mu}{2}\|M - H(u1^T + S)\|_F^2 + \eta \|\nabla(u1^T + S)\|_1,$$  (5.4.1)

where $\eta > 0$ is a regularization parameter and the TV term here is supposed to be applied to each frame. Our preliminary numerical experiments show that when the noise level is high (e.g., with Gaussian noise of standard variance 0.1), the model
in (5.3.1) is more robust than the model in (5.1.8) to extract the background and foreground with few artifacts. For succinctness, we omit the detail of median filter based variational model with total variation regularization.
Chapter 6

Conclusions and future work

6.1 Conclusions

In this thesis, we studied three kinds of abstract models for separable convex programming problems, which have wide applications in the real world. For the considered convex models, we proposed and analyzed some operator splitting methods. Our principles of designing such operator splitting methods focus especially on when it is possible to minimize the functionals by solving a sequence of simple convex minimization problems with explicit formulas for their solutions. We also apply the proposed algorithms in some imaging and video processing problems to demonstrate their efficiency, and superiority to some existing benchmarks in the literature. Specifically, we summarize the main work in the thesis.

6.1.1 Algorithm Proposition

- For the separable monotone variational inequality with positive orthants VI+ (2.0.1)-(2.0.3), we derive a new algorithm by combining the generalized ADMM in [53] with the logarithmic-quadratic proximal regularization proposed in [8].

- For model (P2), we present a proximal version of SC-PRSM (1.4.13). The resulting subproblem can be easy enough to have a closed-form solution for most sparsity-driven applications.
• For model (P3), we propose a parallel splitting algorithm, which is easily implementable in the sense that all the subproblems could yield closed-form solutions.

6.1.2 Convergence Analysis

• For the combination of the generalized ADMM with the logarithmic-quadratic proximal regularization (2.1.1)-(2.1.3), we prove the global convergence and establish the worst-case $O(1/t)$ convergence rates in both the ergodic and non-ergodic senses.

• For the proposed PSC-PRSM (3.1.1), we prove the global convergence and establish the worst-case $O(1/t)$ convergence rates measured by the iteration complexity in both the ergodic and nonergodic senses.

• For the proposed parallel operator splitting algorithm (4.2), we prove its global convergence.

6.1.3 Applications

• Image Restoration We apply our proposed PSC-PRSM (3.1.1) to the wavelet-based image inpainting and the computerized tomography problem arising in medical image processing area.

• Retinex We apply the proposed operator splitting algorithm (4.2) to solve a constrained TV model for Retinex in image enhancement realm.

• Video processing We propose some median filter based variational models for extracting static backgrounds from surveillance videos, and investigate several practical situations when the surveillance video is corrupted by noise, blur or both.

6.2 Future work

Finally, we show some interesting and meaningful topics for future research.
1. Solving the linear equality constrained optimization problem is equivalent to finding the zero point (root) of a maximal monotone operator from dual perspective. The essence of the operator splitting methods thus can be interpreted as decomposing the maximal monotone operator into the sum of two (or more) maximal monotone operators, whose resolvent are easier to evaluated than the resolvent of the original one. One of the maximal monotone operator is typically the form of the composition of a maximal monotone operator with a linear transformation and its adjoint. Inspired by this, we can extend the operator splitting methods to find the root for the sum of two (or more) general maximal monotone operators.

2. In Chapter 3, for the proposed algorithm PSC-PRSM, we emphasize that our main purpose is for the scenario where $\theta_1(x_1)$ is special (e.g., the resolvent operator of $\partial \theta_1$ has a closed-form representation), since this feature is particularly efficient for some sparse and low-rank optimization models. Thus it deserves to exploit its properties effectively in the algorithmic design. However, for a generic case where both $\theta_1$ and $\theta_2$ have no particular properties, the subproblems have to be solved iteratively subject to a general inexactness criterion, as the analysis in [148], we leave as a future work. Moreover, it is also a research topic worthy of consideration to analyze the convergence theory for the combination of SC-PRSM with logarithmic-quadratic proximal regularization in [7, 8].

3. In Chapter 5, we have proposed some median filter based variational models for static background extraction from surveillance video with noise, blur or both. For future work, it is interesting to consider extending the proposed median filter based variational models to the scenarios with time-varying backgrounds. Besides, it is also practically important to develop real-time operator splitting algorithms.
Bibliography


140


[152] J. F. Yang and X. M. Yuan, Linearized augmented Lagrangian and alternating

for \( \ell_1 \)-minimization with applications to compressed sensing, SIAM J. Imaging

[154] X. M. Yuan and M. Li, An LQP-based decomposition method for solving a

[155] X. M. Yuan and J. F. Yang, Sparse and low-rank matrix decomposition via al-

[156] X. Q. Zhang, M. Burger and S. Osher, A unified primal-dual algorithm frame-


[158] Q. Zhou and J. Aggarwal, Tracking and classifying moving objects from videos,
Proc. IEEE Workshop on Performance Evaluation of Tracking and Surveillance,
Curriculum Vitae

Academic qualifications of the thesis author, Ms. LI Xinxin:

- Received the degree of Bachelor of Science in Mathematics from Jilin University, June 2010.

August 2014