Fast numerical methods for image processing based on partial differential equations

Duan, Yuping

2012

Duan, Y. P. (2012). Fast numerical methods for image processing based on partial differential equations. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/47834>

<https://doi.org/10.32657/10356/47834>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**FAST NUMERICAL METHODS FOR IMAGE
PROCESSING BASED ON PARTIAL DIFFERENTIAL
EQUATIONS**

DUAN YUPING

SCHOOL OF PHYSICAL AND MATHEMATICAL SCIENCES

2012

Fast Numerical Methods for Image Processing based on Partial Differential Equations

Duan Yuping

School of Physical and Mathematical Sciences

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2012

Acknowledgment

First and foremost, I would like to thank my supervisor, Prof. Xue-Cheng Tai for his personal guidance and encouraging engagement in supervising my Ph.D study. I thank him for the continuing coaching and valuable discussion regarding my research. I have learnt numerous advanced mathematical techniques and earnest research attitude from him. My appreciation is further extended to Prof. Li-Lian Wang and Prof. Viet Ha Hoang for their seminars related to partial differential equations. I am also grateful of Prof. Desheng Wang and Prof. Songheng Chan for their interesting lectures and for kindly presenting my qualification examination.

I sincerely appreciate NTU for awarding me the Research Scholarship. I would like to thank the management and leadership of Prof. San Ling, Prof. Yeow Meng Chee, Prof. Bernhard Schmidt and Prof. Chaoping Xing for SPMS to give us a wonderful environment to develop ourselves. I am also appreciative of the help from Ms. Lin Jiaying, for her great help on my dissertation issues and many other matters.

Special thanks go to those people who substantially contribute to this thesis. Those people are Dr. Chunlin Wu, Dr. Yu Wang and Juyong Zhang. I would like to thank them for the excellent collaborations. Many thanks to Dr. Chunlin Wu for the numerous valuable discussions about the algorithms and proofs presented in this thesis. I would like to thank Dr. Yu Wang for lots of helpful discussions and inspiration of the curvature project in the work.

I am also thankful for the valuable discussion and suggestions I received from the postdoctoral researchers and graduate students in our group, which include Huibing Chang, Dr. Haixia Liang, Dr. Zhifeng Pang, Dr. Andy C. Yau, Xiaodan Zhao and Dr. Jinghui Zhou. Many thanks to my friends and colleagues at NTU. It is a pleasure memory to study with all of you. Those people are Jie Chen, Ying Gu, Michael Daniel Villena Samson, Juan Shi, Min Wan, Shenling Wang, Yu Wang, Jing Zhang, Liangfeng Zhang and Xiaodan Zhao.

Finally, I am deeply grateful to my parents for giving me the opportunity to do my studies and their support in everything I did.

Acknowledgement	1
Summary	6
1 Introduction	11
1.1 Background and Motivation	12
1.2 Outline of the Dissertation	17
2 Preliminaries	21
2.1 Function Spaces	21
2.2 Convexity	25
2.3 Gâteaux Derivative	27
3 Fundamental Variational Models	29
3.1 The ROF Model	29
3.2 The Vectorial TV Model	31
3.3 The TV- L^1 model	31
3.4 Euler's Elastica	33
3.5 The Mumford-Shah Model	35
3.5.1 Level set function	35
3.5.2 The Chan-Vese model	36
3.6 A Brief Survey of Existing Algorithms of the ROF Model	37

3.6.1	Computing the Euler-Lagrange equation	37
3.6.2	Explicit time marching scheme	38
3.6.3	Linearization of the Euler-Lagrange equation	39
3.6.4	Dual approaches	39
3.6.5	Domain decomposition and multigrid methods	41
3.6.6	Bregman methods	43
3.6.7	Augmented Lagrangian method	45
3.6.8	Graph cuts methods	47
4	DDM with Graph Cuts Algorithms for TV Minimization	51
4.1	Graph Cuts Algorithm for Total Variation	51
4.1.1	Discretization of energy functional	52
4.1.2	Minimization by graph cuts	53
4.1.3	Graph construction	56
4.1.4	Total variation minimization algorithm	58
4.2	Domain Decomposition Method	60
4.3	Graph Cuts Algorithms with Domain Decomposition	62
4.3.1	Overlapping domain decomposition algorithm	63
4.3.2	Time and memory costs	65
4.4	Implementation of the Algorithms	66
4.5	Numerical Results	68
4.5.1	Test of the parameters	69
4.5.2	Performance of DD based algorithm and memory consumption	73
5	DDM with Graph cuts Algorithms for Image Segmentation	79
5.1	Introduction	79
5.2	Mumford-Shah Model with PCLSM	81
5.2.1	Mumford-Shah model	81
5.2.2	Piecewise constant level set method	81
5.2.3	The minimization problem	82
5.3	Graph Cuts for Multiphase Mumford-Shah Model	84
5.3.1	Discretization of energy functional	84

5.3.2	Graph construction	85
5.3.3	An iterative segmentation scheme	87
5.4	Graph Cuts Algorithms with Domain Decomposition Method	88
5.4.1	Non-overlapping domain decomposition method	88
5.4.2	Overlapping domain decomposition method	89
5.5	Numerical Experiments	91
5.5.1	Tests of DD based algorithm on 2D experiments	92
5.5.2	3D experiments	97
6	A Fixed-point ALM for TV Minimization Problems	107
6.1	Introduction	107
6.2	The Fixed-point Formulation	109
6.3	Convergence Analysis	113
6.4	First-order Augmented Lagrangian Algorithms	117
6.5	Acceleration of the Fixed-point Algorithm	120
6.6	Applications	121
6.6.1	Notations and preliminary remarks	121
6.6.2	The ROF model	123
6.6.3	The vectorial TV model	127
6.6.4	The high order model	130
6.6.5	The TV- L^1 model	134
7	Augmented Lagrangian Method for Euler's Elastica	141
7.1	Introduction	141
7.2	Augmented Lagrangian Method for Euler's Elastica Model	143
7.3	The Existing Algorithm	144
7.4	The Proposed Algorithm	146
7.5	Numerical Solutions for Subproblems	148
7.5.1	Notations	148
7.5.2	Subproblems	150
7.6	Numerical Examples	156

8	Image Restoration and Segmentation on Triangulated Surfaces	163
8.1	Introduction	164
8.2	Notation	165
8.3	TV based Restoration and Segmentation on Triangulated Surfaces . .	166
8.3.1	Basic spaces and differential mappings	166
8.3.2	TV restoration model (ROF) on triangulated surfaces	168
8.3.3	Vectorial TV restoration model on triangulated surfaces	168
8.3.4	Multi-region segmentation on triangulated surfaces	170
8.4	ALM for TV Restoration on Triangulated Surfaces	171
8.4.1	Solving the u -sub problem	173
8.4.2	Solving the p -sub problem	173
8.5	ALM for Vectorial TV Restoration on Triangulated Surfaces	174
8.5.1	Solving the $\mathbf{u}$ -sub problem	176
8.5.2	Solving the $\mathbf{p}$ -sub problem	176
8.6	ALM for Multi-region Segmentation on Triangulated Surfaces	177
8.6.1	Solving the $\mathbf{u}$ -sub problem	179
8.6.2	Solving the $\mathbf{p}$ -sub problem	180
8.6.3	Solving the $\mathbf{z}$ -sub problem	180
8.7	Numerical Examples	182
8.7.1	Test platform, stopping condition, and the computation of SNR	182
8.7.2	Examples	182
9	Conclusion and Outlook	191
9.1	Conclusion	191
9.2	Future works	193
	Bibliography	198
	List of Publications	216

Total variation methods were first introduced by Rudin, Osher and Fatemi in 1992 for solving image denoising problems. Since then, total variation (TV) has played the most important role in image processing due to its property in removing noise while preserving sharp edges. However, how to efficiently compute the solution of the ROF model, as well as other variational models, is still a challenging task.

The main purpose of this dissertation is to construct and analyze fast numerical methods for variational models in image processing based on partial differential equations (PDEs). Two kinds of fast numerical techniques – domain decomposition method (DDM) and augmented Lagrangian method (ALM) are discussed and applied for solving the variational models. DDMs, which are efficient tools for handling PDEs, have been implemented to accelerate the numerical solvers for the Rudin-Osher-Fatemi (ROF) model. ALMs, being a certain class of algorithms for solving constrained optimization problems, have also been successfully used in solving the ROF model.

Firstly, we apply DDM to the discrete $TV-L^1$ model and ROF model for image denoising problems. The proposed algorithm benefits from fast solvers by the graph cuts method. We compare the time complexity and memory usage between the proposed domain decomposition algorithm and the original graph cuts algorithm theoretically and numerically. Then, we extend the DDM to the Mumford-Shah model for image segmentation problems. Our method is applied to the discrete piecewise

constant level set formulation of the Mumford-Shah model and solved using the graph cuts algorithm. The proposed methods yield iterative segmentation tools for both 2D and 3D segmentation tasks.

Secondly, we follow the framework of the ALM for general convex problems arising from image processing. A fixed-point method is explored to the original augmented Lagrangian formulation of the convex problems. The proofs for the convergence of the proposed algorithms are provided under some mild assumptions. We evaluate the performance of our algorithm using the real and synthetic imagery on the well-known models, such as the ROF model, the vectorial TV model, the 4th-order model and the TV- L^1 model.

Traditional numerical methods for minimizing the energy functional of Euler's elastica are complicated and computationally complex. A novel algorithm for solving Euler's elastica is achieved based on ALM in this dissertation. We use a fixed-point approach to tackle one nonlinear constraint arising from the model. The proposed algorithm is very efficient in terms of low computational cost per iteration, because our method solves Euler's elastica using only cheap arithmetic operators.

Finally, we propose an efficient algorithm for TV based image restoration and segmentation problems over triangulated surfaces using the ALM. Inspired by the success of the operator splitting method and ALM in 2D planar image processing, we successfully extend the method to TV and vectorial TV based image restoration and segmentation on triangulated surfaces. We consider the applications of the TV based image restoration and TV based multi-region image segmentation on both grey and color images in the numerical examples.

a.e.	almost every(where)
e.g.	exempli gratia
etc.	et cetera
i.e.	id est
l.s.c.	lower semi-continuous
s.t.	such that
w.r.t	with respect to
ALM	Augmented Lagrangian Method
AMA	Alternating Minimization Algorithm
BV	Bounded Variation
CGM	Chan-Golub-Mulet
DDM	Domain Decomposition Method
EE	Euler's Elastica
FFT	Fast Fourier Transform
FP	Fixed Point
MGM	Multigrid Method
Max-Flow/Min-Cut	Maximum Flow/Minimum Cut
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
PDE	Partial Differential Equation
PSC/SSC	Parallel/Successive Subspace Correction
PSNR	Peak Signal to Noise Ratio
ROF	Rudin-Osher-Fatemi
SNR	Signal to Noise Ratio
TV/VTV	Total Variation/Vectorial Total Variation

CHAPTER 1

Introduction

Contents

1.1 Background and Motivation	12
1.2 Outline of the Dissertation	17

Image processing is an important branch of computer vision, which includes many subfields such as denoising, segmentation and inpainting, etc.. This is a highly challenging issue, because images may be very diverse and the way we perceive them varies a lot according to individuals. This dissertation will focus on two specific branches of image processing called image denoising and image segmentation. Such problems are usually formulated as the minimization of the energies in mathematical language. To solve these energy minimization problems numerically, there are mainly continuous and discrete approaches. The continuous approach pursues a function which takes values in a continuous set while the discrete approach seeks a variable at each discrete pixel in the discrete image domain. Although both kinds of methods have been successfully studied for image denoising and segmentation, they still remain most active areas of research in image processing and computer vision. This dissertation serves as some attempts to develop efficient algorithms based on the existing continuous and discrete approaches for well-known variational models in image denoising and segmentation.

1.1 Background and Motivation

Image denoising is perhaps the most fundamental task in image restoration. Noise is an unwanted by-product occurring naturally during the acquisition and transition of the image. Let $f(x, y)$ denote the observed image intensity at a pixel position (x, y) and $u_0(x, y)$ be the true image intensity. Using the additive noise, the image $f(x, y)$ can be written as

$$f(x, y) = u_0(x, y) + \eta(x, y), \quad (1.1)$$

where $\eta(x, y)$ is the additive noise. There are mainly two kinds of noise concerned in our work:

- **Gaussian noise:** In Gaussian noise models, the noise density follows a Gaussian normal distribution. Figure 1.1 (b) shows the image (a) corrupted by Gaussian noise with mean zero and variance 0.1.
- **Salt & pepper noise:** In the salt & pepper noise models, the noisy pixels take only the maximum and the minimum values. Figure 1.1 (c) shows the image (b) polluted with 0.2 Salt & pepper noise.

The denoising problem is to extract u_0 from the noisy image f .

Figure 1.1: Examples of the image (a) corrupted by additive Gaussian noise in (b) and salt & pepper noise in (c).

Image segmentation is problems of partitioning the image domain into different regions and boundaries based on the image intensity f . The result of the segmentation

is a set of regions which cover the whole image domain Ω without any overlap. Each region corresponds to an object in the image. A general image segmentation problem can be formulated as:

Find a partition of Ω satisfying

$$\Omega = \Omega_0 \cup \Omega_1 \cup \cdots \cup \Omega_n, \quad \Omega_i \cap \Omega_j = \emptyset, \quad \forall i \neq j. \quad (1.2)$$

The number of regions n can either be fixed in advance or unknown. Figure 1.2 shows an ideal segmentation example of a noisy image with the “background” Ω_0 and three components Ω_1 , Ω_2 and Ω_3 .

Figure 1.2: A four regions segmentation example.

Such image denoising and segmentation problem can be computed by a minimization of some energy functional. Energy minimization aims to reconstruct the data, which not only approximates the observed data but also is highly correlated, also called regular. Therefore, the standard form of the energy functional is composed of a data fitting term and a regularization term. We use the data fitting term to force the solution to fit the observed data and the regularization term to favor a regular solution.

Let us use image denoising as an example. According to the maximum likelihood principle, we can find an approximation of u by solving a least-square problem as follows

$$\min_u \int_{\Omega} (u - f)^2 d\Omega. \quad (1.3)$$

The Tikhonov model is proposed to recover u from the noisy image f by adding a regularization term to the energy (1.3) in [1, 2] as follows

$$\min_u \left\{ \frac{1}{2} \int_{\Omega} |\nabla u|^2 d\Omega + \frac{\mu}{2} \int_{\Omega} (u - f)^2 d\Omega \right\}. \quad (1.4)$$

The notations in (1.3) and (1.4) are explained as follows:

- Ω is the image domain, which is bounded in $\mathbb{R}^n$. Usually Ω is chosen as a rectangle in $\mathbb{R}^2$.
- The function $f: \Omega \rightarrow \mathbb{R}$ is the observed image.
- μ is a relative weight parameter to be selected.
- $|\cdot|$ is the Euclidean norm on $\mathbb{R}^2$, i.e.,

$$|\nabla u| := |(u_x, u_y)| = \sqrt{u_x^2 + u_y^2}.$$

Due to the L^2 -norm of the gradient in (1.4), the Tikhonov model removes the noise but unfortunately penalizes the gradients too much at the edges. In [3], Rudin, Osher and Fatemi proposed a model by using the L^1 -norm of the gradient to replace the L^2 -norm in (1.4) as the regularization term. The L^1 -norm of the gradient, which is called the total variation, not only removes noise but also preserves sharp discontinuities in the image.

To solve the energy minimization problem (1.4) numerically, we can adopt either continuous methods or discrete optimization methods. Generally speaking, continuous methods regard the image domain as continuous and seek a solution to minimize the energy in a continuous setting. On the contrary, the discrete methods esteem the image domain as discrete and look for a variable at each discrete pixel within the image domain which also minimizes the energy.

In the light of continuous approaches, when variational problems are continuous and differentiable, it is possible to compute the functionals by solving their associated Euler-Lagrange equations, which are usually partial differential equations (PDEs). As

the powerful tools to model and simulate many dynamic, PDEs play an important role in describing the equilibrium phenomena, such as diffusion, advection, transport and others.

By using the continuous approach, the Tikhonov model (1.4) amounts to solving its associated Euler-Lagrange equation together with a boundary condition as follows

$$\mu(u - f) - \Delta u = 0, \quad x \in \Omega; \quad \frac{\partial u}{\partial n} = 0 \quad \text{on } \partial\Omega, \quad (1.5)$$

where Δu is the Laplacian of u . The equation (1.5) gives us a necessary condition to describe the energy functional (1.4) at stationary points. To compute u numerically, we can discretize (1.5) and solve the system.

In view of discrete approaches, instead of solving the Euler-Lagrange equation numerically, the energy functional is discretized directly and computed by combinatorial and integer optimization techniques. We can regard the discrete energy minimization problem as a pixel-labeling problem. The goal is to find a labeling for each pixel which minimizes the energy functional. The standard form of the discrete energy functional is composed of two terms as follows

$$\min_u \left\{ \sum_{p \in \mathcal{P}} D_p(u_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(u_p, u_q) \right\}. \quad (1.6)$$

Here, the notations in (1.6) are explained as follows:

- $\mathcal{P}$ is a set of pixels.
- $\mathcal{N}$ is a set of all pairs of neighboring pixels.
- $D_p(\cdot)$ is a function derived from the observed data that measures the cost of assigning the label to the pixel p , and usually corresponds to the data fidelity term.
- $V_{p,q}(\cdot, \cdot)$ measures the cost of assigning label to the adjacent pixels p , q , and usually corresponds to the regularization term.

The graph cuts method, as the most successful and popular method for solving discrete energy minimization problems, has been greatly studied during the last two decades. In this class of approaches, the discrete energy minimization is formulated as the problem of finding the maximum flow as well as the minimum cut over the graph.

More specifically, we contrast advantages of continuous and discrete approaches for solving the energy minimization problems as follows:

- (i.) One merit of discrete methods is that the global minimizer can be found as long as the discrete energy can be stated as a max-flow / min-cut problem. On the other hand, the solution from continuous methods may be trapped into a local minima when the energy functional is non-convex.
- (ii.) Compared to discrete methods, continuous methods discretize the Euler-Lagrange equation instead of the energy functional itself. Thus, continuous methods are more accurate in representing the geometrical entities without grid bias or metrication artifacts.

Although both continuous and discrete methods give us powerful techniques for energy minimization problems, there is still great need to develop fast and efficient algorithms in both settings. The main motivations of this dissertation are two-fold:

- (i.) For discrete approaches, there are usually high memory requirements of the graph cuts algorithms in dealing with the energy minimization problems, especially for large-scale problems with 2D and 3D data sets. Simultaneously, the computational costs also increase sharply.
- (ii.) For continuous approaches, the lack of differentiability of TV makes the minimization of the variational models computationally challenging. Moreover, there is a need for algorithms that can simplify the variational models by making use of the operator splitting technique. Such algorithms solve the constrained optimization models consisting of convex functionals, which is subject to a number of convex constraints.

Regarding to the above problems, our endeavor is to elaborate efficient algorithms for both discrete and continuous approaches. Roughly speaking, this dissertation focuses on contributions in the following aspects:

- ① We improve the efficiency of the existing graph cuts algorithms for discrete variational models for image denoising and segmentation problems by making use of the domain decomposition method.
- ② We use techniques related to the augmented Lagrangian method to solve variational models for image denoising applications. Operator splitting and decomposition methods are used to generate convex functions composed with linear operators. In addition, we take advantage of the fixed-point-based approach to deal with the associated PDEs.
- ③ The same idea of operator splitting and augmented Lagrangian method in 2D planar image processing is extended to TV and vectorial TV based image restoration and segmentation on triangulated surfaces.

1.2 Outline of the Dissertation

In this dissertation, we try to develop efficient algorithms to solve the variational models for image denoising and segmentation problems. We outline the contributions of each chapter as follows:

In Chapter 2, we have a brief review on the function spaces and mathematical tools at the first place of the dissertation.

In Chapter 3, we introduce the variational models, which include the ROF model, the vectorial TV model, the TV- L^1 model, Euler's elastica and the Mumford-Shah model. Furthermore, we focus on reviewing different kinds of numerical algorithms for solving the ROF model, which is the foundation and background of the dissertation.

In Chapter 4, we try to accelerate the discrete algorithm for solving the TV minimization problems by applying the domain decomposition method.

- For both the TV- L^1 model and the ROF model, a novel algorithm is proposed, which is based on the graph cuts algorithm.
- It is shown theoretically and experimentally that the domain decomposition method can improve both time and memory usage.
- Our algorithm is tested by impulsive noise removal problems for the TV- L^1 model while examined by Gaussian noise removal problems for the ROF model.

In Chapter 5, we present an algorithm by combining the domain decomposition method with the graph cuts algorithm for solving the multiphase Mumford-Shah model.

- A new method to minimize the multiphase Mumford-Shah model with piecewise constant level set representation is proposed.
- A domain decomposition method is applied to the Mumford-Shah model and graph cuts algorithm is utilized to minimize the energy functional.
- The proposed algorithm greatly reduces the memory costs and enables us to solve large size problems effectively.
- Its numerical performance is very robust due to the monotonicity property of the algorithm. It should be noted that the algorithm can segment a 3D MRI image with around $4 \times 10^8 (512 \times 512 \times 512 \times 3)$ voxels in just a few minutes and the result is comparable with that of traditional variational methods.

In Chapter 6, we propose a fixed-point-based algorithm to the augmented Lagrangian formulation of general convex optimization problems.

- For general convex minimization problems, an efficient algorithm is proposed based on the augmented Lagrangian formulation and the fixed-point approach.
- The alternating minimization algorithm in [4] can be derived from the proposed algorithm easily.

- The proofs for the convergence of the fixed-point augmented Lagrangian method and alternating minimization algorithm are provided under some mild assumptions.
- Two kinds of first-order augmented Lagrangian schemes are presented and their connections to the first-order primal-dual algorithm in [5] are established.
- An acceleration technique is used in the proposed algorithm to achieve better convergence.
- A number of applications of general convex problems are established to test the proposed algorithm, such as the ROF model, the vectorial TV model, the high order model and the TV- L^1 model.

In Chapter 7, we propose a fast algorithm for minimizing Euler's elastica energy, in which the Euler's elastica functional is reformulated as a constrained minimization problem.

- An algorithm based on ALM for handling the energy minimization problem associated with Euler's elastica is proposed and solved by operator splitting techniques.
- To tackle the nonlinear constraints arising in the model, a novel fixed-point-based approach is used so that each of the sub-problems either is linear problem or has a closed form solution.
- Numerical examples demonstrate the efficiency and properties of the proposed method.

In Chapter 8, we discuss the image restoration and segmentation over the triangulated surfaces based on the augmented Lagrangian formulation.

- A new algorithm based on ALM is proposed for TV and vectorial TV based image restoration on mesh surface.
- A new algorithm based on ALM is presented for vectorial TV based multi-region image segmentation on mesh surface.

- The proposed algorithms benefit from fast solvers for sparse linear systems and closed form solutions to subproblems.
- Experiments on both gray and color images demonstrate the efficiency of our algorithms.

In Chapter 9, we give the conclusion of the topics studied and discussed in the dissertation and some possible research directions for future investigation.

CHAPTER 2

Preliminaries

Contents

2.1	Function Spaces	21
2.2	Convexity	25
2.3	Gâteaux Derivative	27

In order to study the energy minimization problems, we review the function spaces and mathematical tools in this chapter. We refer to [6, 7, 8] for more complete informations.

2.1 Function Spaces

Vector space

A vector space V is a set that is closed under finite vector addition and scalar multiplication. For a general vector space, the scalars are members of a field F , in which case V is called a vector space over F . In order for V to be a vector space, the following conditions must hold for $\forall u, v, w \in V$ and $\forall \alpha, \beta \in F$:

1. Commutativity: $u + v = v + u$;

2. Associativity: $(u + v) + w = u + (v + w)$;
3. Additive identity: $0 + u = u + 0 = u, \forall u$;
4. Existence of additive inverse: $u + (-u) = 0, \forall u, \exists -u$;
5. Associativity of scalar multiplication: $\alpha(\beta u) = (\alpha\beta)u$;
6. Distributivity of scalar sums: $(\alpha + \beta)u = \alpha u + \beta u$;
7. Distributivity of vector sums: $\alpha(u + v) = \alpha u + \alpha v$;
8. Scalar multiplication identity: $1u = u$.

Euclidean space

For each positive integer n , let $\mathbb{R}^n$ be the set of all n -tuples of real numbers, $x = (x_1, x_2, \dots, x_n)$. We define the inner product of $x, y \in \mathbb{R}^n$ by

$$\langle x, y \rangle := \sum_{i=1}^n x_i y_i,$$

and the norm by

$$|x| := \langle x, x \rangle^{\frac{1}{2}} = \left(\sum_{i=1}^n x_i^2 \right)^{\frac{1}{2}}.$$

The vector space $\mathbb{R}^n$ with the defined inner product and norm is called Euclidean n -space.

Banach space

Definition 2.1. *Given a real vector space X , a norm on X is a function $\|\cdot\| : X \rightarrow \mathbb{R}$ such that*

1. $\|u + v\| \leq \|u\| + \|v\|, \quad \forall u, v \in X$;
2. $\|\alpha u\| = |\alpha| \|u\|, \quad \forall u \in X \text{ and } \forall \alpha \in \mathbb{R}$;
3. $\|u\| \geq 0, \quad \forall u \in X$;

4. $\|u\| = 0$ if and only if $u = 0$.

The space $(X, \|\cdot\|)$ is called a normed vector space. A *Banach space* is a normed vector space which is complete with respect to the metric:

$$d(u, v) = \|u - v\|, \quad \forall u, v \in X.$$

Hilbert space

Definition 2.2. Let X be a real vector space. An inner product on X is a function $\langle u, v \rangle : X \times X \rightarrow \mathbb{R}$ such that

1. $\langle u, v \rangle = \langle v, u \rangle, \quad \forall u, v \in X;$
2. $\langle \alpha u + \beta v, w \rangle = \alpha \langle u, w \rangle + \beta \langle v, w \rangle, \quad \forall u, v, w \in X \text{ and } \forall \alpha, \beta \in \mathbb{R};$
3. $\langle u, u \rangle \geq 0, \quad \forall u \in X;$
4. $\langle u, u \rangle = 0$ if and only if $u = 0$.

The inner product $\langle \cdot, \cdot \rangle$ induces a norm on X , given by

$$\|u\| = \sqrt{\langle u, u \rangle}, \quad \forall u \in X.$$

The metric on X can be defined by

$$d(u, v) = \|u - v\|.$$

A *Hilbert space* is a Banach space endowed with an inner product.

Space of all measurable functions

Let Ω be a nonempty open set in $\mathbb{R}^n$ and let u be a measurable function on Ω .

Definition 2.3. For $1 \leq p \leq \infty$, let

$$L^p(\Omega) = \{u : u \text{ is measurable on } \Omega \text{ and } \|u\|_{L^p(\Omega)} < \infty\},$$

where for $1 \leq p < \infty$,

$$\|u\|_{L^p(\Omega)} := \left(\int_{\Omega} |u(x)|^p dx \right)^{\frac{1}{p}},$$

and

$$\|u\|_{L^\infty(\Omega)} := \operatorname{ess\,sup}_{x \in \Omega} |u(x)|.$$

Space of continuous functions

Let Ω be a nonempty open set in $\mathbb{R}^n$. The continuous function spaces are defined as

- $C^k(\Omega)$ is the set of all real functions: $u : \Omega \rightarrow \mathbb{R}$ that have continuous partial derivatives of order $m = 0, 1, \dots, k$,

$$C^k(\Omega) = \{u : \Omega \rightarrow \mathbb{R} : u \text{ is } k\text{-times continuously differentiable}\}.$$

- $C_0^k(\Omega)$ is the set of all functions $u \in C^k(\Omega)$ with compact support. For the continuous function $u(x)$ defined on $\mathbb{R}^n$, the support set is the closure of the set of points where $u(x)$ is non-zero, that is,

$$\operatorname{supp} u = \overline{\{x \in \mathbb{R}^n : u(x) \neq 0\}}.$$

We say u has compact support if $\operatorname{supp} u$ is bounded.

Space of functions with bounded variation

Total variation is an important concept in image processing and computer vision. We have the following definition for total variation:

Definition 2.4. Let Ω be a bound open domain in $\mathbb{R}^n$ and u be a function in $L^1(\Omega)$.

We set

$$TV(u) = \|u\|_{TV(\Omega)} = \sup \left\{ \int_{\Omega} u \nabla \cdot \xi d\Omega : \xi = (\xi_1, \dots, \xi_n) \in C_0^1(\Omega, \mathbb{R}^n), |\xi|_\infty \leq 1 \right\}, \quad (2.1)$$

where $|\xi|_\infty = \sup_x \sqrt{\sum_{i=1}^n \xi_i^2(x)}$.

A function for which the TV norm yields a finite value, is said to have bounded variation (BV). Explicitly, the space of functions with bounded variation is defined as

$$BV(u) = \left\{ u \in L^1(\Omega) : \|u\|_{TV(\Omega)} < \infty \right\}.$$

$BV(\Omega)$ is a Banach space endowed with the norm

$$\|u\|_{BV(\Omega)} = \|u\|_{L^1(\Omega)} + \|u\|_{TV(\Omega)}.$$

In [9], Federer and Fleming applied an integral formula for TV, the so-called co-area formula. It establishes the relationship between the total variation of u and the perimeter of the level set $\{x : u(x) > \gamma\}$ for all values of γ . We define $E(\gamma)$ to be the lower level set of u at level γ as

$$E(\gamma) = \{(x, y) \in \Omega : u(x, y) < \gamma\}.$$

Let $\text{Per}(E(\gamma))$ be the parameter, i.e., the length of $\partial E(\gamma)$. The co-area formula gives us:

Theorem 2.5. *Let $u \in BV(\Omega)$, then for a.e. $\gamma \in \mathbb{R}$, the set $\{x : u(x) > \gamma\}$ is a finite-perimeter set in Ω , and one has*

$$TV(u) = \int_{\Omega} |\nabla u| d\Omega = \int_{-\infty}^{\infty} \text{Per}(E(\gamma)) d\gamma. \quad (2.2)$$

Intuitively speaking, the co-area formula shows that the TV measures the accumulated surfaces of all its level sets including both the jumps and the lengths of all these level sets.

2.2 Convexity

We aim to find the global minimizer of the minimization problem. An important issue is whether the functional is convex or not. When the functional is convex, any local minimizer is a global minimizer. On the contrary, when the functional is not

convex, it is difficult to say whether the minimizer is local or global. To study the convexity, we start by defining the convex set:

Let X be a finite-dimensional, Euclidean space (or a Hilbert space). A subset $C \subset X$ is said to be a convex set if the straight line segment connecting any two points in C lies entirely inside C . For any two points $x, y \in C$, we have

$$tx + (1 - t)y \in C, \quad \text{for all } t \in [0, 1].$$

We further introduce the definitions of convex, proper and lower semi-continuous for the functions:

Definition 2.6. We say a function $f : X \rightarrow [-\infty, \infty]$ is

- convex if for any $x, y \in X \subset \mathbb{R}^n$,

$$f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y), \quad \text{for all } t \in [0, 1]; \quad (2.3)$$

- proper if f is not identically $-\infty$ or ∞ ;
- lower semi-continuous (l.s.c.) if and only if for any $x \in X$ and any sequence $\{x_n\}_{n=1}^{\infty}$ converging to x , there holds

$$\liminf_{n \rightarrow \infty} f(x_n) = f(x). \quad (2.4)$$

Let X be a vector space. The dual space X^* is defined as the set of all linear functionals on X , i.e.,

$$X^* = \{\phi : X \rightarrow \mathbb{R}, \text{ s.t. } \phi \text{ is linear}\}.$$

The Legendre-Fenchel conjugate of the functional f is defined by:

Definition 2.7. The Legendre-Fenchel conjugate of a functional f is defined for each $y \in X^*$ as

$$f^*(y) = \sup_{x \in X} \{\langle x, y \rangle - f(x)\}.$$

The second conjugates of X and f are expressed as:

$$X^{**} = (X^*)^* \quad \text{and} \quad f^{**} = (f^*)^*.$$

In the finite dimensional Hilbert space setting, we have $X^{**} = X$ and $f^{**} = f$. Consequently, the conjugate of f^* is given in the dual representation

$$f(x) = \sup_{y \in X^*} \{ \langle x, y \rangle - f^*(y) \}.$$

And the dual representation of the Euclidean norm can be obtained as follows; see [8]

$$|x| = \sup_{|y| \leq 1} \langle x, y \rangle. \quad (2.5)$$

For the non-differentiable convex functions, we define the subgradient as follows

Definition 2.8. *The subgradient of a convex function $f : X \rightarrow \mathbb{R}$ at $x \in X$ is defined as:*

$$\partial f(x) = \{v \in X^* : f(y) \geq f(x) + \langle v, y - x \rangle, \quad \forall y \in X\}. \quad (2.6)$$

We can identify the minimizers by making use of the subgradient through the following result:

Lemma 2.9. *Let $f : X \rightarrow \mathbb{R}$ be convex, then $x \in \arg \min_{x \in X} f(x)$ if and only if*

$$0 \in \partial f(x).$$

2.3 Gâteaux Derivative

Let $E(u)$ be an energy functional defined on a Hilbert space X .

Definition 2.10. *The Gâteaux derivative of E at u in the direction of v is defined by*

$$E'(u)v = \lim_{\epsilon \rightarrow 0} \frac{E(u + \epsilon v) - E(u)}{\epsilon}.$$

If there exists $\tilde{u} \in X^*$ (the dual space of X) such that

$$E'(u)v = \tilde{u}(v) := \langle \tilde{u}, v \rangle, \quad \forall v \in X,$$

then E is Gâteaux differentiable at u and the derivative value at u is $\tilde{u}$.

We call a point $u \in X$ a critical point of E , if

$$E'(u) = 0.$$

The above condition is a necessary condition for u being the local minimizer of E . It is also called the Euler-Lagrange equation for E .

There is an important property of Gâteaux differentiable given as follows:

Theorem 2.11. *Suppose that the energy functional $E : X \rightarrow \mathbb{R}$ is Gâteaux differentiable and suppose that u is the minimizer of E , i.e.,*

$$E(u) = \inf_{v \in X} E(v).$$

Then we have

$$E'(u)v = 0, \quad \forall v \in X.$$

CHAPTER 3

Fundamental Variational Models

Contents

3.1	The ROF Model	29
3.2	The Vectorial TV Model	31
3.3	The TV-L^1 model	31
3.4	Euler's Elastica	33
3.5	The Mumford-Shah Model	35
3.6	A Brief Survey of Existing Algorithms of the ROF Model	37

3.1 The ROF Model

In [3], Rudin, Osher and Fatemi (ROF model) firstly introduced the total variation (TV) for solving image denoising problems. The ROF model is proposed to minimize the total variation subject to a constraint as follows

$$\min_u \int_{\Omega} |\nabla u| d\Omega, \quad \text{s.t., } \|u - f\|_2^2 = |\Omega| \sigma^2, \quad (3.1)$$

where $|\Omega|$ is the area of the domain Ω , σ^2 is an estimate of the variance of the Gaussian noise in the observed image f and $\|\cdot\|_2$ denotes the L^2 norm.

The connection between the definition of total variation in (3.1) and (2.1) is established using the dual representation of the Euclidean norm in (3.1) and applying integration by parts as follows; see [8]

$$\begin{aligned} \int_{\Omega} |\nabla u| d\Omega &= \int_{\Omega} \sup_{|\xi| \leq 1} \langle \nabla u, \xi \rangle d\Omega \\ &= \sup_{|\xi| \leq 1} \left[\int_{\partial\Omega} u(\xi \cdot n) dS - \int_{\Omega} u \nabla \cdot \xi d\Omega \right], \end{aligned} \quad (3.2)$$

where $\partial\Omega$ denotes the boundary of Ω and n denotes the outward unit normal to $\partial\Omega$. The term on the boundary vanishes if ξ is compactly supported in Ω . Since the constraint $|\xi| \leq 1$ in the above equality is equivalent to $|\xi| \leq 1$, we can drop the minus sign in the term with $\nabla \cdot \xi$ and come up with the definition of total variation in (2.1).

To formulate the ROF model (3.1) into a convex problem, the author used the inequality constraint $\|u - f\|_2^2 \leq |\Omega|\sigma^2$ to replace the original equality constraint $\|u - f\|_2^2 = |\Omega|\sigma^2$ in [10]. The constrained minimization problem (3.1) is then reformulated to be an unconstrained model as follows

$$\min_u \left\{ E(u) = \int_{\Omega} |\nabla u| d\Omega + \frac{\mu}{2} \int_{\Omega} (u - f)^2 d\Omega \right\}. \quad (3.3)$$

By this technique, the non-convex ROF model (3.1) is successfully approximated by the convex optimization problem (3.3), which is mostly discussed. Due to the nonlinearity and non-differentiability of TV term in (3.3), to solve the ROF model is still numerically challenging. Even so, during the last two decades, a lot of effort has been contributed to design fast solvers for the ROF model in [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]. Figure 3.1 gives an example of the ROF model for image denoising with the additive Gaussian noise.

Figure 3.1: Example of image denoising from the ROF model.

3.2 The Vectorial TV Model

In [22, 23, 24, 25], the authors extended the scalar ROF model for the vector-valued denoising problems. The total variation norm for the vectorial data is defined in [22], which is called vectorial total variation (VTV). Let us denote in general an M -channel image by $\mathbf{u} = (u_1, u_2, \dots, u_M)$. If $M = 3$, it is usual color models, such as RGB.

With the vectorial total variation, we can formulate the total variation based model for recovering the vector-valued data $\mathbf{f}$ by

$$\min_{\mathbf{u}} \left\{ \int_{\Omega} |\nabla \mathbf{u}| d\Omega + \frac{\mu}{2} \int_{\Omega} (\mathbf{u} - \mathbf{f})^2 d\Omega \right\}, \quad (3.4)$$

where

$$|\nabla \mathbf{u}| = \sqrt{\sum_{i=1}^M |\nabla u_i|^2}. \quad (3.5)$$

Figure 3.2 illustrates the denoising performance of the vectorial TV model (3.4) for the RGB color image.

3.3 The TV- L^1 model

Another basic and successful image restoration model is the TV- L^1 model, which can be obtained by replacing the L^2 data fitting term in (3.3) with the L^1 data fitting

Figure 3.2: Example of image restoration from the vectorial TV model.

term. The so-called TV- L^1 model is given as the following minimization problem

$$\min_u \left\{ \int_{\Omega} |\nabla u| d\Omega + \mu \|u - f\|_1 \right\}, \quad (3.6)$$

where $\|\cdot\|_1$ denotes the L^1 norm.

Compared to the ROF model, the TV- L^1 model (3.6) offers us the following properties: First, the TV- L^1 model is more effective than the ROF model in removing impulsive noises (e.g., salt & pepper noises) in [26, 27]. Secondly, the TV- L^1 model is contrast invariant. The homogeneity exists for both the TV norm and L^1 -norm in (3.6). Therefore, for given constant $t > 0$, when u is a solution to (3.6) for a certain input image f , tu is the solution to (3.6) for input image tf .

On the other hand, the TV- L^1 model (3.6) is not a strictly convex functional any more. This means that there is no unique global minimizer guaranteed for the model. Similarly, the TV- L^1 model also suffers the difficulties numerically due to its nonlinearity and non-differentiability. Studies have been successfully done for developing fast numerical solvers for the TV- L^1 model in [28, 29, 30, 26, 27]. We provide an example of the TV- L^1 model for removing the salt & pepper noise in Figure 3.3.

Due to its non-smoothness, a typical way to deal with the energy minimization problem (3.6) is to find a convex approximation of the TV- L^1 model. Such an idea was first proposed in [31] as the following optimization problem

$$\min_{u,z} \left\{ \int_{\Omega} |\nabla u| d\Omega + \mu \int_{\Omega} |z - f| d\Omega + \frac{\eta}{2} \int_{\Omega} (u - z)^2 d\Omega \right\}. \quad (3.7)$$

Figure 3.3: Example of image restoration from the TV- L^1 model.

When the parameter η is large enough, we almost have $u = z$ over the image domain. The general approach in solving (3.7) is to decompose it into two minimization subproblems regarding to the variable u and z , respectively:

- we search for u : when z is fixed

$$\min_u \left\{ \int_{\Omega} |\nabla u| d\Omega + \frac{\eta}{2} \int_{\Omega} (u - z)^2 d\Omega \right\}. \quad (3.8)$$

- we search for z : when u is fixed

$$\min_z \left\{ \frac{\eta}{2} \int_{\Omega} (u - z)^2 d\Omega + \mu \int_{\Omega} |z - f| d\Omega \right\}. \quad (3.9)$$

It is clear that the u -sub problem is exactly the ROF model in the above system. And, there is the closed form solution for the z -sub problem, which could be solved without much effort.

3.4 Euler's Elastica

In spite of many advantageous properties, TV-based methods have a common disadvantage: piecewise constant images are favored over piecewise smooth images, which is the so-called staircasing effect. To overcome this drawback, some high order models

[32, 33, 34, 35] are proposed to yield smoother results. As one of them, Euler's elastica model, which is defined based on the curvature of the level curves of images. It was introduced into computer vision in [36] and further studied in [37, 38, 39, 33, 35].

Euler's elastica energy can be described by the curvature κ of a smooth curve Γ as the following

$$\int_{\Gamma} \left(a + b|\kappa|^{\tau}(s) \right) ds, \quad (3.10)$$

where s is the arc length and a and b are two positive parameters. In the functional (3.10), the first term minimizes the total length and the second term minimizes the power of total curvature. The power τ can be set to either $\tau = 1$ as in [35], or $\tau = 2$ as in [33].

The Euler's elastica of all the level curves of an image u can be expressed as

$$\int_{l=0}^L \int_{\gamma_l: u=l} (a + b|\kappa|^{\tau}(s)) ds dl, \quad (3.11)$$

where γ_l is the level curve with $u = l$. Note that the curvature κ is defined as a function of u

$$\kappa(u) = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right). \quad (3.12)$$

Substituting the above equation into (3.11) and using the co-area formula yields

$$\int_{\Omega} \left(a + b \left| \nabla \cdot \frac{\nabla u}{|\nabla u|} \right|^{\tau} \right) |\nabla u|. \quad (3.13)$$

For image denoising applications, the elastica energy (3.13) can be used as a regularization term. Together with the data fitting term, we can formulate the minimization problem to approximate the noisy image f by Euler's elastica energy as follows

$$\min_u \left\{ \int_{\Omega} \left(a + b \left| \nabla \cdot \frac{\nabla u}{|\nabla u|} \right|^{\tau} \right) |\nabla u| + \frac{\mu}{t} \int_{\Omega} |u - f|^t \right\}, \quad (3.14)$$

the choice of t is determined by the type of noise in f : e.g., $t = 1$ for the salt & pepper noise and $t = 2$ for the Gaussian white noise.

3.5 The Mumford-Shah Model

Mumford and Shah proposed a segmentation model in [40], which is based on the piecewise smooth approximation of the intensity function f , given as follows

$$\min_{u, \Gamma} \left\{ G(u, \Gamma) = \int_{\Omega} (u - f)^2 d\Omega + \alpha \int_{\Omega \setminus \Gamma} |\nabla u|^2 d\Omega + \beta \text{length}(\Gamma) \right\}. \quad (3.15)$$

In the model (3.15), Γ is a set that contains edges and α, β are tuning parameters. The first term in (3.15) measures the fidelity of the approximation u to the input image f . The second term ensures that u is smooth except across the edges Γ . The last term in (3.15) measures the length of the edge set Γ .

3.5.1 Level set function

Osher and Sethian proposed the level set method as an efficient implicit representation for evolving curves and surfaces, in which the interface is expressed as a level set of a function, called the level set function. The level set function can be used to compute the geometric characteristics and the motion of the front.

The level set function ϕ of the closed boundary Γ of the subset $\Omega_{in} \subset \Omega$ is defined as follows

$$\begin{cases} \phi(x) > 0, & \text{for } x \in \Omega_{in}, \\ \phi(x) < 0, & \text{for } x \in \Omega \setminus \Omega_{in}, \\ \phi(x) = 0, & \text{for } x \in \partial\Omega_{in}. \end{cases} \quad (3.16)$$

A typical choice of the level set function ϕ is given by the sign distance function as follows

$$\phi(x) = \begin{cases} \text{dist}(x, \Gamma), & \text{for } x \in \Omega_{in}, \\ -\text{dist}(x, \Gamma), & \text{for } x \in \Omega \setminus \Omega_{in}, \\ 0, & \text{for } x \in \partial\Omega_{in}. \end{cases} \quad (3.17)$$

Such a representation preserves the geometrical quantities, properties and motions of the boundary.

3.5.2 The Chan-Vese model

Chan and Vese proposed an active contour approach based on the Mumford-Shah model in [41]. The idea of the active contour model is to find a contour which separates the image domain into two disjoint regions, Ω_{in} and $\Omega \setminus \Omega_{in}$. The Chan-Vese model is proposed as the following energy minimization problem

$$\min_{\Gamma, c_1, c_2} \left\{ \int_{\Omega_{in}} (f - c_1)^2 d\Omega + \int_{\Omega \setminus \Omega_{in}} (f - c_2)^2 d\Omega + \beta \text{length}(\Gamma) \right\}, \quad (3.18)$$

where c_1 and c_2 are two constants that approximate the image intensities in Ω_{in} and $\Omega \setminus \Omega_{in}$, respectively.

Based on the level set method, the Chan-Vese model is formulated by representing the region Ω_{in} and $\Omega \setminus \Omega_{in}$ using the Heaviside function as follows

$$\min_{\phi, c_1, c_2} \left\{ \int_{\Omega} \{ \mathcal{H}(\phi)(f - c_1)^2 + (1 - \mathcal{H}(\phi))(f - c_2)^2 \} d\Omega + \beta |\nabla \mathcal{H}(\phi)| \right\}, \quad (3.19)$$

where $\mathcal{H}(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ is the Heaviside function: $\mathcal{H}(x) = 0$ if $x < 0$ and $\mathcal{H}(x) = 1$ if $x \geq 0$.

The minimization problem (3.19) is solved by the gradient descent of its Euler-Lagrange equation. In order to compute the associated Euler-Lagrange equation for the unknown ϕ , we consider slightly regularized versions of the function H , denoted by H_ϵ , as $\epsilon \rightarrow 0$. Therefore, we have

$$\phi_t = \delta_\epsilon(\phi) \left\{ - \{ (f - c_1)^2 - (f - c_2)^2 \} + \beta \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right\}, \quad (3.20)$$

where $\delta_\epsilon(\phi) = H'_\epsilon(\phi)$ is the Dirac delta function.

The two constants c_1 and c_2 are updated according to

$$c_1 = \frac{\int_{\Omega} \mathcal{H}(\phi) f d\Omega}{\int_{\Omega} \mathcal{H}(\phi) d\Omega} \quad \text{and} \quad c_2 = \frac{\int_{\Omega} (1 - \mathcal{H}(\phi)) f d\Omega}{\int_{\Omega} (1 - \mathcal{H}(\phi)) d\Omega}. \quad (3.21)$$

The idea of the level set based methods in solving the two-phase Mumford-Shah model has been extended to multiphase segmentation problems in [42, 43, 44, 45, 46].

In Figure 3.4, we give an example of two-phase segmentation which is obtained from the Mumford-Shah model.

Figure 3.4: Example of image segmentation from the Mumford-Shah model.

3.6 A Brief Survey of Existing Algorithms of the ROF Model

Because of the fundamental role of the ROF model in the world of image processing, we give a brief survey of the existing numerical methods in solving the ROF model. These methods are also very important and are the foundation of the contributions in this dissertation.

Due to the non-differentiability of the L^1 norm at zero, it is a challenging task to compute the solution of the ROF model. There are several kinds of methods, listed in Table 3.1, which have been proved particularly efficient in solving the ROF model (3.3).

3.6.1 Computing the Euler-Lagrange equation

A direct way for solving the ROF model is to evolve a PDE, whose steady state corresponds to the local minima of (3.3). The PDE is obtained by the Euler-Lagrange

Table 3.1: A brief survey of existing algorithms to solve ROF model.

List of Algorithms	References
Explicit time marching	[47], [3]
Linearization of the Euler-Lagrange equations	[48], [49], [50]
Dual approaches	[11], [12]
Domain decomposition and Multigrid methods	[51], [52], [53], [48]
Bregman iteration	[54], [15], [14], [55]
Augmented Lagrangian method	[56], [16]
Graph cuts methods	[57], [58], [59], [60]

equation of (3.3) together with a boundary condition

$$\begin{cases} -\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \mu(u - f) = 0 & \text{in } \Omega, \\ \frac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega. \end{cases} \quad (3.22)$$

Because of the term $|\nabla u|$ in the denominator, the PDE (3.22) is highly nonlinear and even undefined at $\nabla u = 0$. To overcome the singularity, we use a regularized term $|\nabla u|_\epsilon = \sqrt{|\nabla u|^2 + \epsilon}$ to replace the norm in the denominator. For a small ϵ , the equation (3.22) is still nearly degenerate. On the contrary, the solution from (3.22) loses its ability to preserve the sharp edges for a large ϵ .

3.6.2 Explicit time marching scheme

In [3], the gradient descent is used to the Euler-Lagrange equation of the ROF model (3.3). The solution is then solved by a parabolic equation with time as an evolution parameter, which is

$$\frac{\partial u}{\partial t} = - \left[-\nabla \cdot \left(\frac{\nabla u}{|\nabla u|_\epsilon} \right) + \mu(u - f) \right], \quad (3.23)$$

with the same boundary condition as (3.22). The PDE (3.23) is computed by the following explicit-time marching scheme until a steady state is reached

$$u^{k+1} = u^k - dt \left[-\nabla \cdot \left(\frac{\nabla u^k}{|\nabla u^k|_\epsilon} \right) + \mu(u^k - f) \right].$$

Due to the conditional stability of the explicit time-marching scheme, we need to choose the time step dt sufficiently small to guarantee the convergence, especially in the flat regions where $|\nabla u| \approx 0$. Therefore, this approach is very reliable, but converges considerably slowly.

3.6.3 Linearization of the Euler-Lagrange equation

Vogel and Oman proposed a fixed-point algorithm for solving the ROF model in [49]. The authors linearized the Euler-Lagrange equation (3.22) by taking the non-linear term in the denominator using the previous iteration. Therefore, we solve the following linear equation

$$-\nabla \cdot \left(\frac{\nabla u^{k+1}}{|\nabla u^k|_\epsilon} \right) + \mu(u^{k+1} - f) = 0. \quad (3.24)$$

The above linear system can be solved by any linear iteration method (e.g., Jacobi, Gauss-Seidel, SOR). However, this linearized approach still greatly depends on the choice of the regularization parameter ϵ . For small ϵ the system converges slowly while for large ϵ edges become blurred.

3.6.4 Dual approaches

The dual approaches [11, 12, 21], which solve the minimization problem (3.3) based on the dual formulation of the ROF model, avoid the degeneration ($\nabla u \approx 0$). Here we mainly review the CGM dual method in [11] and Chambolles dual method in [12].

I. The CGM dual method

In [11], Chan et al proposed a primal-dual method for the ROF model. They defined a new variable as follows

$$w = \frac{\nabla u}{|\nabla u|}. \quad (3.25)$$

By applying w to the Euler-Lagrange equation (3.22), we can reformulate the ROF model as the following primal-dual system

$$\begin{cases} -\nabla \cdot w + \mu(u - f) = 0, \\ \nabla u - w|\nabla u| = 0, \end{cases} \quad (3.26)$$

where u and w are called primal and dual variable, respectively.

Newton's linearization technique is adopted for both the primal and dual variable as the numerical solver for the linear system. Although the regularized TV norm (with parameter ϵ) has to be used in solving the primal variable, the primal-dual Newton's method is proven to be efficient and ϵ can be chosen very close to zero.

II. Chambolle's dual method

Another work based on a dual formulation is proposed by Chambolle in [12]. In this approach, the author substituted the dual form of TV (2.1) into the ROF model (3.3) and achieved the following optimization problem

$$\min_u \max_{|\xi| \leq 1} \left\{ \int_{\Omega} u \nabla \cdot \xi d\Omega + \frac{\mu}{2} \int_{\Omega} (u - f)^2 d\Omega \right\}. \quad (3.27)$$

Since the functional (3.27) is convex and lower semi-continuous w.r.t. u , one is allowed to interchange the min and the max. The variable u can be solved explicitly as

$$u = f - \frac{1}{\mu} \nabla \cdot \xi. \quad (3.28)$$

By substituting (3.28) into the minimization problem (3.27), we arrive at the dual ROF model

$$\min_{|\xi| \leq 1} \left\{ \int_{\Omega} \xi \cdot \nabla f d\Omega + \frac{1}{2\mu} \int_{\Omega} (\nabla \cdot \xi)^2 d\Omega \right\}, \quad (3.29)$$

or, equivalently,

$$\min_{|\xi| \leq 1} \left\{ \frac{1}{2} \|\nabla \cdot \xi - \mu f\|_2^2 \right\}. \quad (3.30)$$

In [12], the constrained problem (3.30) is computed by the Lagrangian method and the semi-implicit gradient descent algorithm is used as the numerical solver.

3.6.5 Domain decomposition and multigrid methods

The Multigrid Method (MGM) is one of the most powerful numerical methods in solving linear and nonlinear elliptic problems [61, 62, 63]. The basic idea of the multigrid method is that low frequency residual errors of the system can be well approximated on a coarse formulation of the original problem.

The linear multigrid method is applied to the linearized Euler-Lagrange equation in [53, 48, 64]. In each inner iteration, we apply the linear multigrid method to the following equation

$$-\nabla \cdot \left(\frac{\nabla u^{k+1}}{|\nabla u^k|_\epsilon} \right) + \mu(u^{k+1} - f) = 0.$$

In [65, 66, 67], the nonlinear multigrid method is used to the non-linear Euler-Lagrange equation of the ROF model while the same idea is used for the solving the regularized Chambolle's dual formulation of the ROF model in [68]. For both cases, the nonlinear multigrid method successfully provides us fast numerical solvers for the ROF model.

Other approaches in [69, 65, 70] propose the multilevel algorithms and domain decomposition based subspace correction algorithms for solving the ROF model. Here, we give a brief review of the space decomposition method in the framework of Tai and Xu [61].

Space decomposition refers to a method that decomposes the space V into a sum of closed subspaces, i.e., there are closed subspaces $V_i \subset V$, $i = 1, 2, \dots, m$ such that

$$V = V_1 + V_2 + \dots + V_m, \quad i = 1, \dots, m. \quad (3.31)$$

This means that for any $u \in V$, there exists $u_i \in V_i$ such that $u = \sum_{i=1}^m u_i$.

Based on the method in [71, 72], instead of solving the energy minimization problem (3.3), we solve a number of subspace minimization problems over the subspaces repeatedly

$$\min_{e_i \in V_i} E(u^0 + e_i), \quad (3.32)$$

where u^0 denotes a current approximation and e_i denotes the correction on each subspace V_i .

Following [71, 72], there are mainly two kinds of subspace correction methods in solving the energy minimization problem based on (3.32), namely the successive subspace correction (SSC) method and the parallel subspace correction (PSC) method.

Algorithm 1 SSC

1. Choose an initial value $u^0 \in V$.
2. For $k \geq 0$, if $u^k \in V$ is defined, find $u^{k+i/m} = u^{k+(i-1)/m} + e_i^k$ with $e_i^k \in V_i$ sequentially for $i = 1, 2, \dots, m$ such that

$$E(u^{k+(i-1)/m} + e_i^k) \leq E(u^{k+(i-1)/m} + v_i), \quad \forall v_i \in V_i. \quad (3.33)$$

3. Go to the next iteration of k .
-

Algorithm 2 PSC

1. Choose an initial value $u^0 \in V$ and relaxation parameters $\varpi_i > 0$ such that $\sum_{i=1}^m \varpi_i \leq 1$.
2. For $k \geq 0$, if $u^k \in V$ is defined, then find $e_i^k \in V_i$ in parallel for $i=1, \dots, m$ such that

$$E(u^k + e_i^k) \leq E(u^k + v_i), \quad \forall v_i \in V_i. \quad (3.34)$$

3. Set

$$u^{k+1} = u^k + \sum_{i=1}^m \varpi_i e_i^k,$$

and go to the next iteration of k .

Both the DDM and the MGM can be interpreted as space decomposition algorithms and the relation between the DDM and the MGM was studied and provided in [65]. Furthermore, it has been proved in [61] that the iterative solutions from both the SSC Algorithm and PSC Algorithm converge to the true solution for general convex functionals.

When applying the DDM to the ROF model, we can use either the PSC algorithm or the SSC algorithm to solve its Euler-Lagrange equation (3.22). For example, given an initial value u^0 , the SSC algorithm solves the ROF model (3.3) by the following scheme

$$\begin{cases} -\nabla \cdot \left(\frac{\nabla u^{k+\frac{j}{m}}}{\sqrt{|\nabla u^{k+\frac{j-1}{m}}|^2 + \epsilon}} \right) + \mu(u^{k+\frac{j}{m}} - f) = 0 & \text{in } \Omega_j, \\ \frac{\partial u^{k+\frac{j}{m}}}{\partial n} = 0 & \text{on } \partial\Omega_j \cap \partial\Omega, \\ u^{k+\frac{j}{m}} = u^{k+\frac{j-1}{m}} & \text{on } \partial\Omega_j \setminus \partial\Omega. \end{cases} \quad (3.35)$$

Outside Ω_j , there is $u^{k+\frac{j}{m}} = u^{k+\frac{j-1}{m}}$.

3.6.6 Bregman methods

I. Bregman method

The Bregman method (or Bregman iterative regularization method) was first introduced in [54] and further analyzed in [73, 14, 19, 55, 15, 74, 75]. The basic idea of Bregman iteration is to transform a constrained problem to a series of unconstrained problems. In each unconstrained problem, the object function is defined by the Bregman distance of a convex function in [76].

The Bregman distance of a convex functional $F(u)$ is defined as the following quantity

$$D_F^g(u, v) = F(u) - F(v) - \langle g, u - v \rangle, \quad (3.36)$$

where $g \in \partial F(v)$. Bregman distance is not a distance in the usual sense because it is not in general symmetric. It measures closeness in the sense that $D_F^g(u, v) \geq 0$, and

$D_F^g(u, v) \geq D_F^g(w, v)$ for w on the line segment between u and v .

In [54], the authors proposed an iterative regularization procedure to improve the ROF model (3.3) by making use of the Bregman distance as follows

$$\begin{cases} u^{k+1} = \min_u \{D_{TV}^{g^k}(u, u^k) + \frac{\vartheta}{2}\|u - f\|_2^2\}, \\ g^{k+1} = g^k - \vartheta(u^{k+1} - f), \end{cases} \quad (3.37)$$

where TV denotes $TV(u)$.

Alternatively, by applying the Bregman iteration to the constrained ROF model (3.1), we can get an equivalent scheme to (3.37) as follows

$$\begin{cases} u^{k+1} = \min_u \{TV(u) + \frac{\vartheta}{2}\|f + g^k - u\|_2^2\}, \\ g^{k+1} = g^k + f - u^{k+1}. \end{cases} \quad (3.38)$$

Compared to other traditional continuous methods, Bregman iteration converges very quickly when applied to certain types of objective functions, especially for minimization problems containing an L^1 -regularization term.

II. Split Bregman method

In [15, 19], the operator splitting method is used to the ROF model and the new generated minimization problem is solved by the Bregman iteration, the so-called “Split Bregman” method. This technique has a very broad use in the class of L^1 -regularized problems.

In the first place, the ROF model (3.3) is rewritten into the constrained optimization problem by introducing a new variable d as follows

$$\begin{aligned} \min_{u, d} \quad & \left\{ E_{Breg}(d, u) = \|d\|_1 + \frac{\mu}{2}\|u - f\|_2^2 \right\}, \\ \text{s.t.,} \quad & d = \nabla u. \end{aligned} \quad (3.39)$$

We replace the constrained problem (3.39) by the following unconstrained problem

by adding the L^2 penalty term

$$\min_{u, d} \left\{ \|d\|_1 + \frac{\mu}{2} \|u - f\|_2^2 + \frac{\vartheta}{2} \|d - \nabla u\|_2^2 \right\}. \quad (3.40)$$

The Bregman method is then applied to solve (3.40) and generates the given iterative system

$$\left\{ \begin{aligned} (u^{k+1}, d^{k+1}) &= \min_{u, d} \{ D_{E_{Breg}}^{(g_u^k, g_d^k)}(u, u^k, d, d^k) + \frac{\vartheta}{2} \|d - \nabla u\|_2^2 \} \\ &= \min_{u, d} \{ E_{Breg}(u, d) - \langle g_u^k, u - u^k \rangle - \langle g_d^k, d - d^k \rangle + \frac{\vartheta}{2} \|d - \nabla u\|_2^2 \}, \\ g_u^{k+1} &= g_u^k - \vartheta \nabla \cdot (\nabla u^{k+1} - d^{k+1}), \\ g_d^{k+1} &= g_d^k - \vartheta (d^{k+1} - \nabla u^{k+1}). \end{aligned} \right. \quad (3.41)$$

The scheme (3.41) can be interpreted as iteratively “adding back the residual” and simplified as follows

$$\left\{ \begin{aligned} (u^{k+1}, d^{k+1}) &= \min_{u, d} \{ E_{Breg}(u, d) + \frac{\vartheta}{2} \|d - \nabla u - y^k\|_2^2 \}, \\ y^{k+1} &= y^k + (\nabla u^{k+1} - d^{k+1}). \end{aligned} \right. \quad (3.42)$$

Therefore, the L^1 -regularized problem is transformed into a sequence of unconstrained optimization problems and Bregman updates. The system generated by the split Bregman method (3.42) can be solved efficiently by fast iterative optimization methods, such as Newton method or Gauss-Seidel method. In addition, Fast Fourier Transform (FFT) was first applied to the alternating minimization algorithms generated by Bregman methods in [14, 55]. Both kinds of approaches give us fast numerical solvers for solving the ROF model.

3.6.7 Augmented Lagrangian method

The augmented Lagrangian method (ALM) was first discussed and studied in [77, 78, 79]. It was further developed by Rockafellar and Glowinski and Le Tallec in [80, 81]. The ALM has recently been successfully applied to the ROF model in [16, 17]. The

authors introduced a new variable and reformulated the ROF model using ALM. The constrained optimization problem is expressed as follows

$$\begin{aligned} \min_{u,p} \quad & \left\{ \int_{\Omega} |p| d\Omega + \frac{\mu}{2} \|u - f\|_2^2 \right\}, \\ \text{s.t.}, \quad & p = \nabla u. \end{aligned} \quad (3.43)$$

To enforce the constraint, we define the augmented Lagrangian functional for the constrained optimization problem (3.43) as follows

$$\mathcal{L}_{ROF}(u, p; \lambda) = \int_{\Omega} |p| d\Omega + \frac{\mu}{2} \|u - f\|_2^2 + \langle \lambda, p - \nabla u \rangle + \frac{r}{2} \|p - \nabla u\|_2^2, \quad (3.44)$$

where λ is the Lagrange multiplier, r is a positive constant and the inner product is $\langle \lambda, p - \nabla u \rangle = \int_{\Omega} \lambda(p - \nabla u) d\Omega$.

Using the augmented Lagrangian method for solving (3.43), we consider the following saddle-point problem

$$\begin{aligned} \text{Find } & (u^*, p^*; \lambda^*), \\ \text{s.t.}, \quad & \mathcal{L}_{ROF}(u^*, p^*; \lambda) \leq \mathcal{L}_{ROF}(u^*, p^*; \lambda^*) \leq \mathcal{L}_{ROF}(u, p; \lambda^*), \quad \forall (u, p; \lambda). \end{aligned} \quad (3.45)$$

The relation between the saddle-point problem (3.45) and the solution to (3.3) is stated in the following theorem

Theorem 3.1. *u is a solution of (3.3) if and only if there exist p and λ such that $(u, p; \lambda)$ is a solution of (3.45).*

The Lagrangian functional (3.44) is solved by the operator splitting and decomposition method in [17]. There are two sub-problems emerging from (3.44) regarding to the variable u and p , i.e.,

– The u -sub problem: for given p

$$\min_u \left\{ \frac{\mu}{2} \|u - f\|_2^2 - \langle \lambda, \nabla u \rangle + \frac{r}{2} \|p - \nabla u\|_2^2 \right\}. \quad (3.46)$$

- The p -sub problem: for given u

$$\min_p \left\{ \int_{\Omega} |p| d\Omega + \langle \lambda, p \rangle + \frac{r}{2} \|p - \nabla u\|_2^2 \right\}. \quad (3.47)$$

Both sub-problems (3.46) and (3.47) can be efficiently solved. Various techniques can be applied to find the optimality condition of the above system, such as Newton's and quasi Newton's linearization. In [17], the u -sub problem (3.46) is solved by FFT while there is closed form solution for the p -sub problem (3.47).

Moreover, the equivalences among the CGM dual method, Chambolle's dual method, the Bregman method, the split Bregman method and the augmented Lagrangian method are studied and established in [19, 17, 74].

3.6.8 Graph cuts methods

Graph cuts is an efficient technique to find the minimizers of the discrete energy minimization problems arising from optimization. It was first introduced to minimize the energy functions in computer vision in [82]. Since then, a great number of graph cuts based methods have been studied in [83, 57, 84, 85, 59, 86, 87, 88, 58] to generate fast methods to minimize a discrete variant of the perimeter and an additional fidelity term.

I. Background on graphs

A directed weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ refers to a collection of vertices $\mathcal{V}$ and a collection of directed edges $\mathcal{E}$ that connect pairs of vertices. There are two special vertices in the graph, i.e., the source s and the sink t . For each edge $(a, b) \in \mathcal{E}$, there is a capacity $c(a, b)$ which represents the maximum amount of flow that can pass through the edge (a, b) . A cut on the graph $\mathcal{G}$ partitions the vertices into two disjoint groups: source set S and sink set T , such that $s \in S$ and $t \in T$. The cost of the cut is the sum of capacities of all edges that go from S to T

$$c(S, T) = \sum_{a \in S, b \in T, (a, b) \in \mathcal{E}} c(a, b).$$

The min-cut algorithm aims to find the cut with the smallest cost, which is defined as follows:

Definition 3.2. *The min-cut problem is to find the partition which minimizes*

$$\min_{(S,T)} \sum_{a \in S, b \in T, (a,b) \in \mathcal{E}} c(a,b), \quad (3.48)$$

subject to

$$S \cap T = \emptyset, \quad S \cup T = \mathcal{V}, \quad s \in S, \quad t \in T.$$

For each $a \in \mathcal{V}$ the neighborhood system $N^+(a)$ is defined as all $b \in \mathcal{V}$ such that $(a,b) \in \mathcal{E}$ and $N^-(a)$ is defined as all $b \in \mathcal{V}$ such that $(b,a) \in \mathcal{E}$. For each $(a,b) \in \mathcal{E}$ define the “flow” variable $f(a,b) \in \mathbb{R}$. And there are the following two constraints on $f(a,b)$ given as:

- The weights $c(a,b)$ are upper bounds on the flow $f(a,b)$ for all $(a,b) \in \mathcal{E}$, i.e.,

$$f(a,b) \leq c(a,b), \quad \forall (a,b) \in \mathcal{E}. \quad (3.49)$$

- There is a conservation law for each vertex except s and t , i.e.,

$$\sum_{b \in N^+(a)} f(a,b) - \sum_{b \in N^-(a)} f(b,a) = 0, \quad \forall a \in \mathcal{V} \setminus \{s, t\}. \quad (3.50)$$

The min-cut problem and the max-flow problem are dual to each other. Therefore, we can solve the much easier max-flow problem to find the minimum cuts. We can formulate the max-flow problem as follows:

Definition 3.3. *The max-flow problem is to find the flow f which maximizes*

$$\max_f \sum_{a \in \mathcal{V}} f(s,a), \quad (3.51)$$

subject to (3.49) and (3.50).

To solve the min-cut/max-flow problem, there are mainly two groups of algorithms: Goldberg-Tarjan style “push-relabel” methods [89] and Ford-Fulkerson style “augmenting paths” [90]. In Figure 3.5, we provide an example of the directed capacitated graph in [84] to illustrate the min-cut algorithm.

Figure 3.5: Example of a directed capacitated graph. Edge costs are reflected by the thickness of the edges.

II. Graph cuts algorithm for the ROF model

Graph cuts algorithm has been used in solving the discrete ROF model and the discrete TV- L^1 model (see [58, 59, 86, 87] and many others). In these kinds of approaches, to make the total variation term be graph representable, the isotropic TV, given as

$$TV(u) = \int_{\Omega} \sqrt{u_x^2 + u_y^2} d\Omega,$$

is replaced by the anisotropic TV, which is defined by

$$TV_{ani}(u) = \int_{\Omega} (|u_x| + |u_y|) d\Omega.$$

By substituting the anisotropic TV into (3.3), the ROF model is expressed as the following energy minimization problem

$$\min_u E_{ani}(u) = \int_{\Omega} (|u_x| + |u_y|) d\Omega + \frac{\mu}{2} \|u - f\|^2. \quad (3.52)$$

The energy functional (3.52) is discretized into binary problem and decomposed into leveled energies by using the discrete co-area formula. Each level energy is a binary problem which can be solved efficiently by the graph cuts methods. More details for implementation the graph cuts algorithm in solving the ROF model are discussed in Chapter 4.

CHAPTER 4

DDM with Graph Cuts Algorithms for TV Minimization

Contents

4.1	Graph Cuts Algorithm for Total Variation	51
4.2	Domain Decomposition Method	60
4.3	Graph Cuts Algorithms with Domain Decomposition . .	62
4.4	Implementation of the Algorithms	66
4.5	Numerical Results	68

4.1 Graph Cuts Algorithm for Total Variation

We have introduced both the TV- L^1 model and ROF model for image denoising in Chapter 3. In this chapter, we further discuss the numerical solutions for these two total variation (TV) based models. At the first place, we recall the two models as follows:

$$\text{TV} - L^1 \text{ model : } \min_{u \in BV(\Omega)} TV(u) + \mu \|u - f\|_1, \quad (4.1)$$

$$\text{ROF model : } \min_{u \in BV(\Omega)} TV(u) + \mu \|u - f\|_2^2. \quad (4.2)$$

We adopt the discrete forms of the TV- L^1 model (4.1) and ROF model (4.2) presented in [86, 87, 91] and use the graph cuts algorithm in [84] to solve both discrete total variation minimization problems. We propose a new algorithm for total variation minimizations by combining the graph cuts algorithm with the Domain Decomposition Method (DDM). Numerical experiments show that the proposed domain decomposition based algorithm approximates the global minimizer well and works efficiently for large-scale and high resolution images.

4.1.1 Discretization of energy functional

We shall present the TV- L^1 model (4.1) and ROF model (4.2) in the discrete setting. Let $\mathcal{P}$ denote the index set of the grid points with mesh size $\delta = 1$, i.e.,

$$\mathcal{P} = \{(i, j) \mid i = 1, \dots, M, j = 1, \dots, N\}. \quad (4.3)$$

For the TV term in (4.1) and (4.2), there are two ways to discretize it, i.e., isotropic and anisotropic discretization. Since the isotropic discretization is not graph representable, we adopt the anisotropic discretization to represent TV term in this work. The anisotropic discretization of TV term depends on the neighborhoods involved.

For each grid point $p = (i, j) \in \mathcal{P}$, define the neighborhood systems,

$$\mathcal{N}_p^4 = \{(i \pm 1, j), (i, j \pm 1)\} \cap \mathcal{P}, \quad (4.4)$$

$$\mathcal{N}_p^8 = \{(i \pm 1, j), (i, j \pm 1), (i \pm 1, j \pm 1)\} \cap \mathcal{P}. \quad (4.5)$$

Based on the neighborhood systems above, the anisotropic discretization of TV(u) can be expressed as follows (c.f., [86, 92])

$$TV^4(u) = \sum_{(i,j) \in \mathcal{P}} |u_{i+1,j} - u_{i,j}| + |u_{i,j+1} - u_{i,j}|, \quad (4.6)$$

$$TV^8(u) = TV^4(u) + \frac{1}{\sqrt{2}} \sum_{(i,j) \in \mathcal{P}} |u_{i+1,j+1} - u_{i,j}| + |u_{i+1,j-1} - u_{i,j}|. \quad (4.7)$$

For each pair of neighboring grid points p and q , the term $|u_p - u_q|$ in (4.6) and

(4.7) has an equivalent representation $|u_p - u_q| = (u_p - u_q)^+ + (u_q - u_p)^+$, where $x^+ = \max\{x, 0\}$. Therefore, we can define another formula of $\text{TV}(u)$ as

$$\text{TV}^\kappa(u) = \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{N}_p^\kappa} w_{p,q} (u_p - u_q)^+, \quad (4.8)$$

where κ indicates what kind of neighborhoods is used and u_p denotes the value of the image u at the site $p = (i, j) \in \mathcal{P}$. The weights $w_{p,q}$ are chosen according to the cut metric described in [93].

The data fitting terms in the $\text{TV}-L^1$ model (4.1) and ROF model (4.2) can be discretized directly. Therefore, the discrete versions of (4.1) and (4.2) can be expressed as the following energy problems, respectively,

$$E^1(u; \mu, f) = \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{N}_p^\kappa} w_{p,q} (u_p - u_q)^+ + \mu \sum_{p \in \mathcal{P}} |u_p - f_p|, \quad (4.9)$$

$$E^2(u; \mu, f) = \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{N}_p^\kappa} w_{p,q} (u_p - u_q)^+ + \mu \sum_{p \in \mathcal{P}} |u_p - f_p|^2. \quad (4.10)$$

4.1.2 Minimization by graph cuts

To implement the graph cuts algorithm to the discrete $\text{TV}-L^1$ model (4.9) and ROF model (4.10), we start by rewriting both the TV term and the data fitting terms by the binary variables to make them be graph representable in [94]. Since the level set functions of u are greatly used for the discretizations of both the TV term and fidelity terms, we first give some basic properties of the level set functions of u .

An image u can be decomposed into its level set functions using the decomposition principle in [95]. To introduce the decomposition principle, we import the super-level indicator of u : $\mathbf{1}_{\{u \geq \gamma\}}$, for all grey-scale levels γ . For every $p \in \mathcal{P}$, the definition of the indicator $\mathbf{1}_{\{u_p \geq \gamma\}}$ is given as

$$\begin{aligned} \mathbf{1}_{\{u_p \geq \gamma\}} &= 1 & \text{for } u_p &\geq \gamma, \\ \mathbf{1}_{\{u_p \geq \gamma\}} &= 0 & \text{for } u_p &< \gamma. \end{aligned} \quad (4.11)$$

The decomposition principle states that the image u can be decomposed into its level set functions $\{\mathbf{1}_{\{u \geq \gamma\}}\}$. Provided u labels a finite number of grey levels $\gamma = \{0, 1, \dots, l_{max}\}$, where γ is the level control parameter and $l_{max} = \max_p \{u_p\}$, we have the following discrete formula of u

$$u = \sum_{\gamma=0}^{l_{max}} \mathbf{1}_{\{u \geq \gamma\}}. \quad (4.12)$$

Using the decomposition principle in [95], the image u also can be recovered from its level set functions $\{\mathbf{1}_{\{u \geq \gamma\}}\}$ from

$$u_p = \max \{ \gamma \mid \mathbf{1}_{\{u_p \geq \gamma\}} = 1, \gamma \in \{0, \dots, l_{max}\}, \forall p \in \mathcal{P} \}.$$

Therefore, we can regard the corresponding level set family $\{\mathbf{1}_{\{u \geq \gamma\}}\}$ as an equivalent representation of u .

We can use the indicators $\mathbf{1}_{\{u_p \geq \gamma\}}$ and $\mathbf{1}_{\{u_q \geq \gamma\}}$ to replace the term u_p and u_q in the discrete TV formula (4.8) and rewrite the term $(u_p - u_q)^+$ as follows

$$(u_p - u_q)^+ = \sum_{\gamma=0}^{l_{max}} (\mathbf{1}_{\{u_p \geq \gamma\}} - \mathbf{1}_{\{u_q \geq \gamma\}})^+. \quad (4.13)$$

Therefore, for any grid point $p \in \mathcal{P}$ and $q \in \mathcal{N}_p^\kappa$, the TV term can be expressed in terms of binary variables $\mathbf{1}_{\{u_p \geq \gamma\}}$ and $\mathbf{1}_{\{u_q \geq \gamma\}}$ as follows

$$TV^\kappa(u) = \sum_{\gamma=0}^{l_{max}} TV^\kappa(\mathbf{1}_{\{u \geq \gamma\}}), \quad (4.14)$$

where

$$TV^\kappa(\mathbf{1}_{\{u \geq \gamma\}}) = \sum_{p \in \mathcal{P}} \sum_{q \in \mathcal{N}_p^\kappa} w_{p,q} (\mathbf{1}_{\{u_p \geq \gamma\}} - \mathbf{1}_{\{u_q \geq \gamma\}})^+. \quad (4.15)$$

Similarly, we express the L^1 fidelity term and L^2 fidelity term in terms of indicators $\mathbf{1}_{\{u \geq \gamma\}}$ and $\mathbf{1}_{\{f \geq \gamma\}}$. Based on the properties of the indicator stated in [91], we obtain

the discrete form of the fidelity terms as follows

$$\sum_{p \in \mathcal{P}} |u_p - f_p| = \sum_{\gamma=0}^{l_{max}} \sum_{p \in \mathcal{P}} \left((1 - \mathbf{1}_{\{f_p \geq \gamma\}}) \mathbf{1}_{\{u_p \geq \gamma\}} + \mathbf{1}_{\{f_p \geq \gamma\}} \mathbf{1}_{\{u_p < \gamma\}} \right), \quad (4.16)$$

$$\sum_{p \in \mathcal{P}} |u_p - f_p|^2 = \sum_{\gamma=0}^{l_{max}} 2 \sum_{p \in \mathcal{P}} \left(\left(\gamma - \frac{1}{2} \right) \mathbf{1}_{\{u_p \geq \gamma\}} + f_p \mathbf{1}_{\{u_p < \gamma\}} \right) + C_1, \quad (4.17)$$

where C_1 is a constant that does not depend on u . More details of the deduction can be found in [91].

So far, we have expressed the TV term as well as the L^1 and L^2 fidelity terms by the binary variables. For the convenience, associated with each level γ , let $U_\gamma = \mathbf{1}_{\{u \geq \gamma\}}$, so that $U_\gamma \in \{0, 1\}^{M \times N}$. Besides, we use $U_\gamma(p)$ to denote the value of U_γ at the site $p \in \mathcal{P}$. Therefore, the discrete TV- L^1 model (4.9) and discrete ROF model (4.10) can be rewritten by combining $TV^\kappa(u)$, for a particular choice of κ , and the data fitting terms

$$E^1(u; \mu, f) = \sum_{\gamma=0}^{l_{max}} \left(TV^\kappa(U_\gamma) + \mu \sum_{p \in \mathcal{P}} \left((1 - \mathbf{1}_{\{f_p \geq \gamma\}}) U_\gamma(p) + \mathbf{1}_{\{f_p \geq \gamma\}} (1 - U_\gamma(p)) \right) \right), \quad (4.18)$$

$$E^2(u; \mu, f) = \sum_{\gamma=0}^{l_{max}} \left(TV^\kappa(U_\gamma) + 2\mu \sum_{p \in \mathcal{P}} \left(\left(\gamma - \frac{1}{2} \right) U_\gamma(p) + f_p (1 - U_\gamma(p)) \right) \right) + C_2, \quad (4.19)$$

where $TV^\kappa(U_\gamma)$ is defined by (4.15) with $\mathbf{1}_{\{u \geq \gamma\}}$ replaced by U_γ and C_2 is a constant independent of u .

In order to minimize (4.18) and (4.19), one would like to decompose $E^1(u; \mu, f)$ and $E^2(u; \mu, f)$ into leveled energies over the grey-scale levels. For a fixed level γ , we consider the following two discrete geometry energy minimization problems that correspond to the problems of minimizing the discrete TV- L^1 energy (4.18) and ROF

energy (4.19) respectively

$$\min_{U_\gamma \in \{0,1\}^{M \times N}} E_d^1(U_\gamma; \mu, \gamma, f) = TV^\kappa(U_\gamma) + \mu \sum_{p \in \mathcal{P}} \left((1 - \mathbf{1}_{\{f_p \geq \gamma\}}) U_\gamma(p) + \mathbf{1}_{\{f_p \geq \gamma\}} (1 - U_\gamma(p)) \right), \quad (4.20)$$

$$\min_{U_\gamma \in \{0,1\}^{M \times N}} E_d^2(U_\gamma; \mu, \gamma, f) = TV^\kappa(U_\gamma) + 2\mu \sum_{p \in \mathcal{P}} \left(\left(\gamma - \frac{1}{2} \right) U_\gamma(p) + f_p (1 - U_\gamma(p)) \right). \quad (4.21)$$

We minimize all $E_d^1(U_\gamma; \mu, \gamma, f)$ or $E_d^2(U_\gamma; \mu, \gamma, f)$ independently and get a family of minimizers $\{U_\gamma^*\}$ for $\gamma \in \{0, 1, \dots, l_{max}\}$, where U_γ^* is the minimizer of (4.20) or (4.21) on the level γ . The topographic map relies on simple inclusions of $\{U_\gamma^*\}$, which was proven for discrete levels and continuous levels in [96, 86]. For our case, as the upper level set functions in [96], these $\{U_\gamma^*\}$ are decreasing, i.e.,

$$U_{\gamma_1}^* \supseteq U_{\gamma_2}^*, \quad \forall \gamma_1 \leq \gamma_2. \quad (4.22)$$

Based on all the minimizers $\{U_\gamma^* : \gamma = 0, \dots, l_{max}\}$, one can obtain the minimizer u of the discrete TV- L^1 energy (4.18) and the ROF energy (4.19). Since (4.22) holds, one can regard the family $\{U_\gamma^*\}$ as the level set functions of u . Then the equivalence between the minimizer u and $\{U_\gamma^* : \gamma = 0, \dots, l_{max}\}$ can be established as

$$u_p = \max \left\{ \gamma \mid U_\gamma^*(p) = 1, \gamma \in \{0, \dots, l_{max}\}, \forall p \in \mathcal{P} \right\}. \quad (4.23)$$

Such relationship is established for both (4.18) and (4.19), which is provided and proven in [86].

4.1.3 Graph construction

For the numerical solver, we use the graph cuts method to solve the leveled energy (4.20) and (4.21) on each level. For the implementation, we construct a graph containing $M \times N$ vertices together with the source s and sink t using the method in

[94]. More specifically, we have $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and

$$\mathcal{V} = \{v_p \mid p \in \mathcal{P}\} \cup \{s\} \cup \{t\}. \quad (4.24)$$

There are two kinds of edges in each graph $\mathcal{E}$: $\mathcal{E}_D$ corresponds to the data fitting term and $\mathcal{E}_R$ corresponds to the TV term. They are defined, respectively, as

$$\mathcal{E}_D = \cup_{p \in \mathcal{P}} \{(s, v_p) \cup (v_p, t)\}, \quad (4.25)$$

$$\mathcal{E}_R = \{(v_p, v_q) \mid p \in \mathcal{P}, q \in \mathcal{N}_p^\kappa\}. \quad (4.26)$$

Using the leveled TV- L^1 energy (4.20) for example, we give the capacities to $\mathcal{E}_D$ and $\mathcal{E}_R$ as follows, where $y = \mathbf{1}_{\{f_p \geq \gamma\}}$,

$$\mathcal{E}_D : \quad c(s, v_p) = \mu y, \quad c(v_p, t) = \mu(1 - y), \quad \forall p \in \mathcal{P}, \quad (4.27)$$

$$\mathcal{E}_R : \quad c(v_p, v_q) = w_{p,q}, \quad \forall p \in \mathcal{P}, \quad \forall q \in \mathcal{N}_p^\kappa. \quad (4.28)$$

In Figure 4.1, we present how to add the two kinds of edges according to the above capacities to construct the graph for the leveled TV- L^1 model.

Figure 4.1: The graph representation of the TV- L^1 model. (a) represents the edges of data fitting term and (b) represents the edges of TV term.

For the leveled ROF energy (4.21), we only have to change the capacities of the

edges in $\mathcal{E}_D$

$$\mathcal{E}_D : \quad c(s, v_p) = 2\mu f_p, \quad c(v_p, t) = 2\mu(\gamma - \frac{1}{2}), \quad \forall p \in \mathcal{P}. \quad (4.29)$$

To construct the graph for the leveled ROF energy, we only have to change the capacities of edges corresponding to the data fitting term in Figure 4.1 (a).

Since there is a relationship (4.22) among all the levels, we solve the discrete minimization problem (4.20) and (4.21) from l_{max} to 0. On the first level l_{max} , we construct a graph by adding all edges according to (4.27)-(4.29). After using the max-flow / min-cut algorithm on the corresponding graph of the first level, the parametric algorithm generates a min-cut $(S^{l_{max}}, T^{l_{max}})$. It is obvious that the relationship $T^{l_{max}} \subseteq T^{l_{max}-1} \subseteq \dots \subseteq T^0$ holds over all the levels. Therefore, we argue that the nodes in $T^{l_{max}} \setminus \{t\}$ do not take effect in determining the min-cut (S^k, T^k) for any $k < l_{max}$. Here, we create an $M \times N$ matrix *Isused* to mark the nodes in the graph. The initial value of all the elements in *Isused* is 0 and when the node has been cut into T^k on some level k , we change the corresponding element in *Isused* to be 1. We update *Isused* in this way on every level. When we compute the energy minimization problem on level k , we do not need to include the nodes with *Isused* = 1, as well as all edges from or to this set of nodes, in the max-flow calculations. To do this, we disconnect the nodes set *Isused* = 0 and *Isused* = 1 from each other by freezing the flow in all edges which go through these two sets and only consider the sub-network corresponding to the node set with *Isused* = 0 and its reduced edges.

There is a simple algorithm provided to explain the method of constructing the graph based on *Isused*; see Algorithm 3. In the algorithm, we let *Isused*(p) denote the value of *Isused* at the site $p \in \mathcal{P}$. For convenience, we use ROF model (4.19) as an example in the explanation of all algorithms here and there.

4.1.4 Total variation minimization algorithm

A simple graph cuts based total variation minimization algorithm can be used to solve the discrete TV- L^1 model (4.18) and ROF model (4.19). The details for the ROF model are given in Algorithm 4.

Algorithm 3 Graph construction algorithm

For $\forall p \in \mathcal{P}$,

- If $Isused(p) = 0$
Give the capacities to the edges of p in $\mathcal{E}_D$ according to (4.29) and the edges in $\mathcal{E}_R$ according to (4.28).

- Else
Freeze the capacities to the edges of p by giving the capacities of the edges in $\mathcal{E}_R$ as

$$c(s, p) = 0, \quad c(p, t) = \infty,$$

and no edges between p and q in $\mathcal{E}_D$ iff $Isused(q) = 0, \forall q \in \mathcal{N}_p^\kappa$.

Algorithm 4 Graph cuts algorithm for total variation minimization

Set $Isused(p) = 0, \forall p \in \mathcal{P}$.For $l = l_{max}, \dots, 1, 0$

1. Construct the graph using $Isused$ based on Graph construction Algorithm and use the graph cuts algorithm to solve the energy minimization problem (4.21) on level l . The cut result U_l^* is estimated from

$$U_l^* = \arg \min_{U_l} E_d^2(U_l; \mu, l, f). \quad (4.30)$$

2. Update $Isused$ by

$$Isused(p) = 1 \quad \text{if} \quad U_l^*(p) = 1, \quad \forall p \in \mathcal{P}.$$

3. Update $l \leftarrow l - 1$.

Estimate u according to (4.23).

We can use different graph cuts algorithms in (4.30) of Algorithm 4 to solve the minimization problem, such as the augmenting path algorithm described in [92, 84] and the push-relabel algorithm provided in [91].

4.2 Domain Decomposition Method

It is well-known that domain decomposition method is a powerful technique for solving the partial differential equations [97, 98, 61]. DDM solves a boundary value problem by splitting the problem into smaller boundary value problems on subdomains. Due to the superiority for parallel computer environments, DDM has been applied for image processing problems [61, 99].

As what was done in [61], we partition the image domain Ω into four kinds of subdomains, Ω_1 , Ω_2 , Ω_3 and Ω_4 . It is obvious that each subdomain is a union of the sub problems. We use Figure 4.2 to illustrate the decomposition of the domain Ω with 25 blocks.

Ω_1	Ω_2	Ω_1	Ω_2	Ω_1
Ω_3	Ω_4	Ω_3	Ω_4	Ω_3
Ω_1	Ω_2	Ω_1	Ω_2	Ω_1
Ω_3	Ω_4	Ω_3	Ω_4	Ω_3
Ω_1	Ω_2	Ω_1	Ω_2	Ω_1

Figure 4.2: An example of domain decomposition with 25 subdomains

There are two kinds of decomposition methods depending on whether the subdomains overlap or not, namely the overlapping and non-overlapping domain decomposition. We use Figure 4.3 to dispatch four kinds of subdomains in the overlapping domain decomposition approach corresponding to the partition in Figure 4.2.

Figure 4.3: Four kinds of subdomains in the overlapping domain decomposition

In the Figure 4.3, we let $\{D_i\}_{i=1}^4$ denote the non-overlapping domain decomposition with subdomains of solid boundaries and $\{\Omega_i\}_{i=1}^4$ be the overlapping domain decomposition with subdomains of dashed boundaries, which are shown in Figure 4.3. Here, each Ω_i is an enlarged subdomain with $\Omega_i = D_i^\delta$ consisting of grid points p with $\text{dist}(p, D_i) \leq \delta$. We regard δ as the overlapping size of the overlapping domain decomposition. Large overlapping size can give faster convergence. However, it also leads to the increment of memory and computational cost in each subdomain. Therefore, we have to choose a proper overlapping size in the implementation, which will be discussed in Section 4. We mainly introduce the overlapping domain decomposition in our work. However, we also test the non-overlapping domain decomposition approach when δ is zero in the numerical experiments.

More precisely, we decompose the image domain Ω into the overlapping subdomains Ω_i , $i = 1, \dots, 4$,

$$\Omega = \bigcup_{i=1}^4 \Omega_i, \quad \Omega_i \cap \Omega_j \neq \emptyset, \quad i \neq j. \quad (4.31)$$

To explain the theory clearly, we need to introduce some notations. We have $\Omega_i = \Omega_i^0 \cup \partial\Omega_i$. Let Ω_i^0 denote the interior grid points of Ω_i , i.e., $\Omega_i^0 = D_i$ and $\partial\Omega_i$ denote the grid points on the boundary of Ω_i , i.e., $\partial\Omega_i = \Omega_i \setminus D_i$. Correspondingly, $\mathcal{P}_i^0$ is the index set for Ω_i^0 and $\partial\mathcal{P}_i$ is the index set for $\partial\Omega_i$.

Based on the overlapping domain decomposition in Figure 4.3, the solution space

is also decomposed into subspaces. Let $V = BV^{DIS}(\Omega)$, where $BV^{DIS}(\Omega)$ is the space of bounded variation defined in the discrete sense. We have

$$V = \sum_{i=1}^4 V_i, \quad V_i = BV_0^{DIS}(\Omega_i), \quad (4.32)$$

where $BV_0^{DIS}(\Omega_i)$ is the subspace of $BV^{DIS}(\Omega_i)$ with zero traces on the “interior” boundaries $\partial\Omega_i \setminus \partial\Omega$.

We explicitly define $BV_0^{DIS}(\Omega_i)$ as follows

$$BV_0^{DIS}(\Omega_i) = \{ v \mid v|_{\partial\Omega_i \setminus \partial\Omega} = 0, \forall v \in BV^{DIS}(\Omega_i) \}.$$

For ease of illustration, we keep using the ROF model (4.2) for example in the following sections. Therefore, we can simply use $E(u)$ to denote the discrete ROF model (4.19) and $E_d(U)$ to denote the corresponding leveled energy of the ROF model (4.21). When applying DDM to the TV- L^1 model (4.1), one only has to adopt the TV- L^1 energy (4.18) and (4.20) to replace (4.19) and (4.21) respectively.

4.3 Graph Cuts Algorithms with Domain Decomposition

Using graph cuts based algorithms to solve the minimization problem (4.21), we have to construct a graph and use the graph cuts algorithm on each level from l_{max} to 0, which will lead the increment of the computational time. On the other hand, when the images become large, Algorithm 4 requires a lot of memory to construct the graph. These become crucial problems for high resolution and large-scale images. In these cases, we consider to apply a domain decomposition method to Algorithm 4 to reduce the computational cost and memory usage.

4.3.1 Overlapping domain decomposition algorithm

Recall the successive subspace correction (SSC) Algorithm 1 discussed in Chapter 4, we will implement the domain decomposition based SSC approach to the discrete ROF model (4.19).

We decompose the image domain Ω into four kinds of subdomains as shown in Figure 4.3. Therefore, given the initial value $u^0 = f$, the SSC Algorithm requires us to compute u^{k+1} from

$$\begin{cases} E(u^{k+\frac{i-1}{4}} + e_i^k) \leq E(u^{k+\frac{i-1}{4}} + v_i), & \forall v_i \in V_i, \\ u^{k+\frac{i}{4}} = u^{k+\frac{i-1}{4}} + e_i^k, & i = 1, 2, 3, 4. \end{cases}$$

Here $e_i^k \in V_i$ is the correction of the subproblem over Ω_i .

Thus, $u^{k+\frac{i}{4}}$ is the minimizer of the associated discrete energy functional (4.19) in Ω_i for $i = 1, \dots, 4$. We can further define the energy functional for $u^{k+\frac{i}{4}}$ over the subdomain Ω_i as follows

$$E(u^{k+\frac{i}{4}}; \mu, f) = \sum_{\gamma=0}^{l_{max}} E_d(\tilde{U}_\gamma^i; \mu, \gamma, f),$$

with $E_d(\tilde{U}_\gamma^i; \mu, \gamma, f) = E_d(\tilde{U}_\gamma^i|_{\Omega_i}; \mu, \gamma, f) + E_d(\tilde{U}_\gamma^i|_{\Omega \setminus \Omega_i}; \mu, \gamma, f),$

where $\tilde{U}_\gamma^i$ is obtained by $\tilde{U}_\gamma^i = \mathbf{1}_{\{u^{k+\frac{i}{4}} \geq \gamma\}}$ for $\gamma \in \{0, 1, \dots, l_{max}\}$ and $\tilde{U}_\gamma^i|_{\Omega_i}$ denotes the value of $\tilde{U}_\gamma^i$ inside Ω_i .

Since we only care about the solution of $\tilde{U}_\gamma^i$ in Ω_i , we minimize the following energy functional from $\gamma = l_{max}$ to $\gamma = 0$

$$\begin{aligned} & E_d(\tilde{U}_\gamma^i|_{\Omega_i}; \mu, \gamma, f) \\ &= \sum_{p \in \mathcal{P}_i^0} \sum_{q \in \mathcal{N}_p^s \cap \mathcal{P}_i} w_{p,q} (\tilde{U}_\gamma^i(p) - \tilde{U}_\gamma^i(q))^+ + 2\mu \sum_{p \in \mathcal{P}_i^0} \left(\left(\gamma - \frac{1}{2} \right) \tilde{U}_\gamma^i(p) + f_p (1 - \tilde{U}_\gamma^i(p)) \right). \end{aligned}$$

Besides, we have the boundary condition for $u^{k+\frac{i}{4}}$ as follows

$$u^{k+\frac{i}{4}} = u^{k+\frac{i-1}{4}} \quad \text{on} \quad \partial\Omega_i \setminus \partial\Omega.$$

Outside Ω_i , we have $u^{k+\frac{i}{4}} = u^{k+\frac{i-1}{4}}$. Thus, there is no need to solve $u^{k+\frac{i}{4}}$ outside Ω_i .

This algorithm is carried out from Ω_1 to Ω_4 one after another, which is the classic SSC method. The specific graph cuts based SSC method for ROF model (4.19) can be described as Algorithm 5.

Algorithm 5 Domain decomposition algorithm for total variation minimization

For $k = 0, 1, \dots$,

1. While $i = 1, \dots, 4$
 For $l = l_{max}, \dots, 1, 0$
 - i. Construct the graph using *Isused* based on Graph construction Algorithm and use the graph cuts algorithm to solve the energy minimization problem in Ω_i on level l . The cut result $U_l^*|_{\Omega_i}$ is estimated from

$$U_l^*|_{\Omega_i} = \arg \min_{\tilde{U}_l^i|_{\Omega_i}} E_d(\tilde{U}_l^i|_{\Omega_i}; \mu, l, f). \quad (4.33)$$

- ii. Update *Isused* by

$$Isused(p) = 1 \quad \text{if} \quad U_l^*|_{\Omega_i}(p) = 1, \quad \forall p \in \mathcal{P}_i^0.$$

2. According to (4.23), we have

$$u^{k+\frac{i}{4}} = \begin{cases} \max\{l \mid U_l^*|_{\Omega_i}(p) = 1, l \in \{0, 1, \dots, l_{max}\}\}, & \text{in } \Omega_i, \\ u^{k+\frac{i-1}{4}}, & \text{in } \Omega \setminus \Omega_i. \end{cases}$$

Go to the next iteration if not convergent.

For the minimization problem (4.33), we use graph cuts algorithm to find the value of U_l in Ω_i . We know Ω_i contains many disjoint subdomains, i.e., $\Omega_i = \bigcup_j \Omega_{i,j}$. As the subproblems over $\Omega_{i,j}$ are independent of each other, we can use the graph cuts algorithm to solve the subdomain problems simultaneously. If we have parallel computation environments, these subdomain problems can be solved in parallel. In our implementations, we just solve the problems one by one. Even so, the computational cost is reduced compared to solving the minimization problem (4.19) by Algorithm 4 in the whole domain.

In the non-overlapping domain decomposition approach, that is the case $\delta = 0$,

there is no regularization edges between adjacent subdomains. For a grid point $p \in \mathcal{P}_i$ on the boundary of Ω_i , the subdomain problem only includes the regularization terms related to $q \in \mathcal{N}_p^\kappa \cap \mathcal{P}_i$, i.e., the subdomain problem only regularizes the grid points inside the subdomain. For the neighboring grid point $q \notin \mathcal{N}_p^\kappa \cap \mathcal{P}_i$, we just ignore this kind of the regularization terms.

In the overlapping domain decomposition approach, the value is $u^{k+\frac{i}{4}} = u^{k+\frac{i-1}{4}}$ in $\Omega \setminus \Omega_i$ and thus has no need for computation. The value of $u^{k+\frac{i}{4}}$ inside Ω_i needs to be calculated by the graph cuts algorithm. For a grid point $p \in \mathcal{P}_i^0$, $\mathcal{N}_p^\kappa$ may be outside Ω_i^0 . However, this does not cause any problem for solving (4.33) as the value of $u^{k+\frac{i}{4}}$ outside Ω_i^0 is already known. This will take care of the regularization terms between the subdomains. More details of the implementation of the Algorithm 5 can be found in Section 3.4.

For this algorithm, we have (c.f., [61])

$$E(u^{k+1}) \leq E(u^{k+\frac{3}{4}}) \leq E(u^{k+\frac{1}{2}}) \leq E(u^{k+\frac{1}{4}}) \leq E(u^k) \leq \dots \leq E(u^1) \leq E(u^0). \quad (4.34)$$

This guarantees the monotonicity of the cost functional and thus gives a robust algorithm.

4.3.2 Time and memory costs

In this work, we use the augmenting path algorithm introduced in [84] as the numerical solver of (4.30) in Algorithm 4 and (4.33) in Algorithm 5. The worst time complexity of the augmenting path algorithm is $O(mn^2|C|)$, with m being the number of edges, n being the number of the nodes in the graph and $|C|$ denoting the trivial upper bound on the number of augments. In our setting, to implement Algorithm 4, we have to construct l_{max} graphs and use $|C|_{max}$ to denote the maximal upper bound among the graphs. There are total $2(M(N-1) + N(M-1) + MN)$ edges and MN nodes in each graph. These are the same for both the TV- L^1 model and ROF model. Thus, the worst time complexity of Algorithm 4 is $O\left(l_{max}(3MN - M - N)(MN)^2|C|_{max}\right)$.

On the other hand, to implement Algorithm 5, we first decompose the image domain into 4 kinds of subdomains as shown in Figure 4.3 and the sub problems over

the same kind of subdomains can be computed in parallel. Suppose we decompose the image domain into subdomains with uniform size, we use M_i and N_i to denote the size of the subdomain, $|C_i|_{max}$ to denote the maximal upper bound on the number of augments in the subdomain and k to be the iteration number of Algorithm 5. Therefore, the worst time complexity of Algorithm 5 over the image domain Ω is $O(4kl_{max}(3M_iN_i - M_i - N_i)(M_iN_i)^2|C_i|_{max})$.

For the memory cost, both the nodes and the edges have to be stored when implementing Algorithm 4 and Algorithm 5. Thus, the memory cost of Algorithm 4 is the total number of nodes and edges in the graph, that is $O(7MN - 2M - 2N)$. Consequently, the memory cost of Algorithm 5 is the largest sum of nodes and edges in the subdomain, which is $O(7M_iN_i - 2M_i - 2N_i)$. If the image domain is decomposed into N_s uniform size subdomains, the memory cost of Algorithm 5 is theoretically $1/N_s$ of Algorithm 4.

Finally, we conclude the worst time and memory costs from the Algorithm 4 and Algorithm 5 in Table 4.1.

Table 4.1: The time and space costs of Algorithm 4 and Algorithm 5.

	Time Cost	Memory Cost
Algorithm 4	$O(l_{max}(3MN - M - N)(MN)^2 C _{max})$	$O(7MN - 2M - 2N)$
Algorithm 5	$O(4kl_{max}(3M_iN_i - M_i - N_i)(M_iN_i)^2 C_i _{max})$	$O(7M_iN_i - 2M_i - 2N_i)$

4.4 Implementation of the Algorithms

For the implementation of Algorithm 4, we use the graph construction algorithm to construct the corresponding graph on each level $\gamma \in \{l_{max}, \dots, 1, 0\}$. We freeze the grid points that already find the optimal value by defining the capacity to the edge connecting to source s as 0 and the capacity to the edge connecting to sink t as ∞ and ignore the regularization edges. Once the graph is constructed, we use the augmenting path algorithm to find the minimum cut U_γ . After the same process

through all the levels, we obtain the minimizer over the image domain according to the equation (4.23).

For the non-overlapping DD based Algorithm 5, the implementation is quite easy. The subdomain Ω_i contains many disjoint subproblems, i.e., $\Omega_i = \bigcup_j \Omega_{i,j}$. We regard each subdomain $\Omega_{i,j}$ as a self-contained problem and use the augmenting path algorithm to get the minimum cut. Here, we use nearly the same method as Algorithm 4. The only difference is that we need to construct and use the graph cuts algorithm over each subdomain $\Omega_{i,j}$ instead of the whole domain Ω .

For the implementation of the overlapping DD based Algorithm 5, we take some extra care to the overlapping grid points. For a given $p \in \mathcal{P}_i^0$, $\mathcal{N}_p^\kappa$ may be outside Ω_i^0 and these values are known and needed for the minimization of the subproblem. If we take $\kappa = 4$, then $\mathcal{N}_p^4$ is always within $\mathcal{P}_i = \mathcal{P}_i^0 \cup \partial\mathcal{P}_i$ for any $p \in \mathcal{P}_i^0$. When we implement the graph cuts algorithm to each subdomain, we construct the graph in the same way as in Algorithm 4 based on the matrix I_{used} . For each subdomain problem, we construct the graph for the subdomain $\Omega_{i,j} \subset \Omega_i$ to include the interior and boundary grid points on one level, i.e., the subdomain graph is

$$\begin{aligned}\mathcal{V}^{i,j} &= \{v_p \mid p \in \mathcal{P}_{i,j}\}, \\ \mathcal{E}^{i,j} &= \mathcal{E}_D^{i,j} \cup \mathcal{E}_R^{i,j}, \\ \mathcal{E}_D^{i,j} &= \bigcup_{p \in \mathcal{P}_{i,j}} \{(s, v_p) \cup (v_p, t)\}, \\ \mathcal{E}_R^{i,j} &= \{(v_p, v_q) \mid p \in \mathcal{P}_{i,j}^0, q \in \mathcal{N}_p^\kappa\}.\end{aligned}$$

In the above notations, $\mathcal{P}_{i,j}$ and $\mathcal{P}_{i,j}^0$ are self-explainable. For the overlapping domain decomposition Algorithm 5, the capacities of the edges for interior grid points are defined as in (4.27)-(4.29). The boundary value of $u^{k+\frac{i}{4}}$ is known as $u^{k+\frac{i}{4}} = u^{k+\frac{i-1}{4}}$ in $\Omega \setminus \Omega_i$. We only compute the value $u^{k+\frac{i}{4}}$ using Algorithm 4 in the interior of Ω_i . To keep the boundary values unchanged, the capacity for the edges in $\mathcal{E}_D^{i,j}$ for any $p \in \partial\mathcal{P}_{i,j}$ should be defined to be ∞ or 0. This is the only extra “care” that we need to take for the implementation of Algorithm 4.

4.5 Numerical Results

In this section, we implement our domain decomposition methods on synthetic and real data. We develop our codes in C++ environment. All numerical experiments are performed on an IBM server with 2.93 GHz, 8 Intel (R) Xeon (R) Quad-Core CPU and 128 RAM. We use MATLAB to load and save images.

We present various numerical results to demonstrate the efficiency of the proposed DD based Algorithm 5. We use PSNR as a criteria for the quality of restoration, which is usually expressed in terms of the logarithmic decibel scale

$$\text{PSNR}(u, w) = 10 \log_{10} \frac{255^2}{\frac{1}{MN} \sum_{i,j} (u_{i,j} - w_{i,j})^2},$$

where $u_{i,j}$ and $w_{i,j}$ denote the pixel values of the restored and the clean images respectively. We use the relative error between two consecutive iterations

$$\frac{\sum_{i,j} (u^k(i, j) - u^{k-1}(i, j))^2}{\sum_{i,j} (u^k(i, j))^2} < TOL,$$

as the stopping rule for Algorithm 5, TOL is the prescribed tolerance.

At the beginning, we test the impact of the parameter μ , the convergence of DD based Algorithm 5 and the influences of the overlapping size δ and subdomain size d to Algorithm 5. Furthermore, we apply the DD based Algorithm 5 to test the TV- L^1 model for solving impulsive noise removal problems and the ROF model for solving Gaussian noise removal problems. During the experiments, we compare the visual quality, computational cost, PSNR, numerical energy and memory usage between Algorithm 4 and DD based Algorithm 5. We test the TV- L^1 model and ROF model involving 4 neighbors or 8 neighbors for different examples. For all the experiments, the initial value of u^0 for the DD based Algorithm 5 is chosen as $u^0 = f$. The parameter μ varies with cases.

4.5.1 Test of the parameters

There are several important parameters for our DD based Algorithm 5: the parameter μ , the error tolerance TOL , the overlapping size δ (pixels) and the subdomain size d (pixels). We choose two images Peppers (512×512) and Cameraman (512×512) and add Gaussian white noise of mean 0 and variation 0.02 to both images. We apply both Algorithm 4 and Algorithm 5 to the noisy images to identify the best choices of μ , TOL , δ and d in the first place. These tests are all from the ROF model (4.2). The similar conclusions exist for the TV- L^1 model (4.1).

I. Test of μ

Both the TV- L^1 model and ROF model are very sensitive to the parameter μ . To get the best restoration results from the given images, we should first identify a suitable μ for the models. We fix the error tolerance $TOL = 10^{-4}$ and the overlapping size $\delta = 1$ for the overlapping DD based Algorithm 5. We plot the PSNR against μ obtained by Algorithm 4 and Algorithm 5 with different subdomain size d in Figure 4.4. For both algorithms, we have the same conclusion that a good choice of μ is between 0.01 and 0.03 and the largest PSNR is achieved around $\mu = 0.02$. Therefore, we fix $\mu = 0.02$ for both images Peppers and Cameraman in the following tests.

Figure 4.4: The impact of μ to PSNR: Peppers (left) and Cameraman (right).

II. Test of the error tolerance TOL

In [86], it is proved that the discrete ROF model can gain the global minimizer of (4.2). We hope the proposed DD based Algorithm 5 can approximate the results from Algorithm 4 with both the visual quality and PSNR. Therefore, we track the relative error TOL , PSNR and the computational time corresponding to the different iterations in Table 4.2. From the results, we observe that one iteration ($TOL = 10^{-1}$) or two iterations ($TOL = 10^{-4}$) for Algorithm 5 is enough to generate the results with the comparable PSNR as from Algorithm 4. In the following tests, we record the results from Algorithm 5 with both the error tolerance $TOL = 10^{-1}$ and $TOL = 10^{-4}$.

Table 4.2: The relative error of Algorithm 5 for Peppers and Cameraman. For the comparison, the PSNR and computational time of Algorithm 4 are provided in the table.

Algorithm 5	Peppers			Cameraman		
Iteration No	TOL	PSNR(dB)	Time(s)	TOL	PSNR(dB)	Time(s)
1	5.8×10^{-2}	27.82	18.89	5.4×10^{-2}	27.91	18.62
2	3.91×10^{-5}	27.85	37.55	3.21×10^{-5}	27.93	37.41
3	1.76×10^{-7}	27.85	56.61	1.47×10^{-7}	27.93	55.83
Algorithm 4		27.88	46.96		27.93	48.43

III. Test of the overlapping size δ

For the overlapping DD based Algorithm 5, the overlapping size δ affects the minimization problem on the subdomains. To illustrate the impact of the overlapping size, we tabulate the PSNR and computational cost of Algorithm 5 in Table 4.3. The Algorithm 5 is implemented with the fixed subdomain size $d = 64$, but different overlapping size δ from $\delta = 0$ to $\delta = 6$. From the results shown in Table 4.3, we see that the computational cost of the DD based Algorithm 5 keeps increasing by enlarging the overlapping size. It means that larger overlapping size leads to additional computational costs. On the other hand, the PSNR is improved by adding the overlapping pixels to the subdomains, but it almost remains the same for all different overlapping sizes. Therefore, it is advisable to choose δ to be 1 or 2 to balance a good trade-off between convergence rate and quality of restoration. And we fix the overlapping size $\delta = 1$ for all the experiments in this work.

Table 4.3: Test of dependence of the overlapping sizes. In the experiments, we fix the subdomain size $d = 64$.

Image	μ	d	δ	$TOL = 10^{-1}$		$TOL = 10^{-4}$	
				PSNR(dB)	Time(s)	PSNR(dB)	Time(s)
Peppers	0.02	64	0	27.76	18.35	27.76	36.91
			1	27.82	19.19	27.85	38.17
			2	27.82	20.03	27.85	40.05
			3	27.82	21.05	27.85	41.89
			4	27.82	22.47	27.85	43.29
			5	27.82	26.98	27.85	49.2
			6	27.82	27.55	27.85	50.22
Camera -man	0.02	64	0	27.81	18.04	27.81	36.16
			1	27.91	18.69	27.93	37.27
			2	27.91	19.91	27.93	39.3
			3	27.91	20.68	27.93	40.9
			4	27.91	21.6	27.93	43
			5	27.91	26.02	27.93	49.27
			6	27.91	27.34	27.93	50.58

IV. Test of the subdomain size d

Another important parameter for the DD based Algorithm 5 is the subdomain size d . Therefore, we examine the influence of subdomain size to the overall performance of the DD based Algorithm 5. We fix the overlapping size $\delta = 1$ and tabulate the computational cost and PSNR with different subdomain size d from $d = 256$ to $d = 8$ in Table 4.4. The experiment explains that the computational cost of Algorithm 5 first decreases and then increases while the subdomain size keeps decreasing. It also shows that a good choice of d is roughly $1/8$ or $1/16$ of the input image size. We set the subdomain size for the 512×512 images Peppers and Cameraman to be $d = 64$ in this work.

After these tests, we provide the restored images from the ROF model with the optimal choices of the intrinsic parameters. Figure 4.5 and Figure 4.6 are devoted to Peppers and Cameraman respectively with $\mu = 0.02$, $TOL = 10^{-4}$, $\delta = 1$ and $d = 64$.

(a) Restored images from Algorithm 4, Non-overlapping DD based Algorithm 5 and overlapping DD based Algorithm 5

(b) Difference images between the restored images and noisy images

Figure 4.5: Numerical results from Pepper.

(a) Restored images from Algorithm 4, Non-overlapping DD based Algorithm 5 and overlapping DD based Algorithm 5

(b) Difference images between the restored images and noisy images

Figure 4.6: Numerical results from Cameraman.

Table 4.4: Test of dependence of the subdomain sizes. In the experiments, we choose the overlapping size $\delta = 1$.

Image	μ	δ	d	$TOL = 10^{-1}$		$TOL = 10^{-4}$	
				PSNR(dB)	Time(s)	PSNR(dB)	Time(s)
Peppers	0.02	1	256	27.87	27.87	27.87	54.73
			128	27.85	23.54	27.86	44.24
			64	27.82	19.29	27.85	38.11
			32	27.73	19.82	27.81	40.2
			16	27.52	22.2	27.70	64.71
			8	27.06	27.14	27.49	78.52
Camera -man	0.02	1	256	27.93	27.7	27.93	54.66
			128	27.93	23.36	27.94	47.2
			64	27.91	19.06	27.93	38.24
			32	27.82	20.06	27.92	59.17
			16	27.64	21.73	27.90	64.88
			8	27.01	27.04	27.79	78.9

4.5.2 Performance of DD based algorithm and memory consumption

We apply both the TV- L^1 model (4.1) and ROF model (4.2) with 4-neighborhood to the image Lena (1024×1024). We add different levels of Gaussian white noise with mean zero and variance $\sigma = 0.01$, $\sigma = 0.02$, $\sigma = 0.03$ and $\sigma = 0.05$ to Lena to test the proposed algorithms for solving the ROF model. It has been demonstrated in [100, 27, 101] that the TV- L^1 model is particularly suitable for handling non-Gaussian additive noises such as impulsive noise. Therefore, we add different levels of salt & pepper noise 10%, 20%, 30% and 50% to Lena to test our algorithms for solving the TV- L^1 model. During the experiment, image domain is always decomposed into 16×16 subdomains ($d = 64$), the overlapping size is chosen $\delta = 1$ and the iteration is fixed for both the TV- L^1 model and ROF model to be 1. The computational costs of the TV- L^1 model and ROF model are provided in Table 4.5. Besides, we display the recovered results from the TV- L^1 model in Figure 4.7.

To further compare Algorithm 4 and Algorithm 5, we list the PSNR and minimal energy in the Lena experiment from the TV- L^1 model and ROF model in Table 4.6 and Table 4.7 respectively. Here we also test the DD based Algorithm 5 in the case of

Figure 4.7: The restoration results from the TV- L^1 model of Lena with salt & pepper noise 10%, 20%, 30% and 50%.

Table 4.5: The computational time(s) of the TV- L^1 model and ROF model on Lena with different noise. In the table, we use Lena0.1, Lena0.2, Lena0.3 and Lena0.5 to indicate what kind of salt & pepper noise is added to Lena while we use Lena0.01, Lena0.02, Lena0.03 and Lena0.05 to indicate what kind of Gaussian noise is added to Lena.

Prob	TV- L^1			Prob	ROF		
	μ	ALG 4	ALG 5		μ	ALG 4	ALG 5
Lena0.1	1	155.89	78.12	Lena0.01	0.02	151.29	73.07
Lena0.2	1	160.8	80.13	Lena0.02	0.02	154.93	74.8
Lena0.3	1	165.76	83.99	Lena0.03	0.01	161.24	77.38
Lena0.5	0.8	178.17	94.35	Lena0.05	0.01	161.51	80.16

$\delta = 0$, which is denoted as the non-overlapping DD based Algorithm 5 in this work. From Table 4.6, we see that PSNR from the DD based Algorithm 5 approximates the value from Algorithm 4 and the overlapping DD based Algorithm 5 always gives better PSNR than the non-overlapping DD based Algorithm 5. Besides, Table 4.7 tells that the energy from the DD based Algorithm 5 is convergent and approximates the minimal energy obtained by Algorithm 4, especially the overlapping DD based Algorithm 5.

Table 4.6: The PSNR(dB) comparison among Algorithm 4, non-overlapping Algorithm 5 and overlapping Algorithm 5 of the TV- L^1 model and ROF model on Lena.

Prob	TV- L^1			Prob	ROF		
	ALG 4	NON ALG 5	ALG 5		ALG 4	NON ALG 5	ALG 5
Lena0.1	38.10	37.86	38.34	Lena0.01	32.71	32.58	32.68
Lena0.2	36.75	36.53	36.86	Lena0.02	31.28	30.96	31.07
Lena0.3	35.24	35.04	35.16	Lena0.03	30.06	29.96	30.08
Lena0.5	31.54	31.24	31.53	Lena0.05	29.30	29.11	29.16

Table 4.7: The minimal energy of the TV- L^1 model and ROF model on Lena. In the table, the energy of the TV- L^1 model is the quoted number multiplied by 10^7 and the energy of the ROF model is the quoted number multiplied by 10^9 .

Prob	TV- L^1			Prob	ROF		
	ALG 4	NON ALG 5	ALG 5		ALG 4	NON ALG 5	ALG 5
Lena0.1	1.848	1.855	1.850	Lena0.01	10.205	10.205	10.206
Lena0.2	3.159	3.167	3.162	Lena0.02	10.243	10.245	10.244
Lena0.3	4.456	4.464	4.460	Lena0.03	5.1343	5.1354	5.1353
Lena0.5	5.675	5.685	5.682	Lena0.05	5.1361	5.1376	5.1375

Memory consumption

We have theoretically compared the memory cost between Algorithm 4 and Algorithm 5 in Section 3.4. In this subsection, we compare the memory usages of the proposed algorithms for solving the TV- L^1 model and ROF model in practice. We use the image Lena with 10% salt & pepper noise to test the TV- L^1 model and Lena with Gaussian noise $\sigma = 0.01$ to test the ROF model. We decompose the image domain into subdomains with size from $d = 512$ to $d = 16$ and record the largest memory consumption required by Algorithm 4, non-overlapping and overlapping DD based Algorithm 5 in Table 4.8. The results demonstrate that the DD based Algorithm 5 saves a large quantity of memory compared to Algorithm 4, and the overlapping DD based Algorithm 5 always requires larger memory than non-overlapping Algorithm 5. Besides, from the data in Table 4.8, it is also shown that 16×16 subdomains are the best choice to implement DD based Algorithm 5 to image with size 1024×1024 . Altogether, the proposed algorithm is a fast and robust method in particular for images of large sizes.

Finally, we give an example to illustrate the superiority of the overlapping DD method compared to the non-overlapping DD method. The experiment is taken on the image Cartoon (128×128) corrupted with additive Gaussian noise with mean zero and variance 0.01. We apply the ROF model to the noisy image and use the 8 neighbors in the neighborhood for all the algorithms. In this experiment, we choose the same

Table 4.8: The memory consumptions of the TV- L^1 model and ROF model on Lena. In the table, the unit of the number is KB.

Subdomain d	TV- L^1		ROF	
	NON ALG 5	ALG 5	NON ALG 5	ALG 5
1024	291,056	291,056	288,760	288,760
512	62,612	120,428	59,164	115,580
256	30,332	51,932	26,672	51,612
128	20,116	26,208	16,332	26,440
64	19,260	25,096	15,156	20,948
32	18,500	20,984	14,404	20,052
16	18,268	20,008	14,180	20,084

value $\mu = 0.02$ for all the algorithms and decompose the image into 4×4 subdomains for DD based algorithms. The overlapping size is $\delta = 1$ for the overlapping DD based Algorithm 5. Through the results shown in Figure 4.8, we can easily see that the overlapping decomposition method is better than the non-overlapping decomposition method. For the same parameter, overlapping Algorithm 5 gives the better and more continuous result than the non-overlapping Algorithm 5.

Figure 4.8: The comparison between non-overlapping DD based Algorithm 5 and overlapping DD based Algorithm 5.

CHAPTER 5

DDM with Graph cuts Algorithms for Image Segmentation

Contents

5.1	Introduction	79
5.2	Mumford-Shah Model with PCLSM	81
5.3	Graph Cuts for Multiphase Mumford-Shah Model . . .	84
5.4	Graph Cuts Algorithms with Domain Decomposition Method	88
5.5	Numerical Experiments	91

5.1 Introduction

Segmentation is one of the fundamental problems in image processing and computer vision tasks. The result of image segmentation is a set of contours extracted from the image, or a set of regions that collectively cover the entire image. Mumford-Shah model (3.15) proposed in [40] is an effective tool for region based image segmentation. This model is robust to noise and can segment objects without edges. However, the minimization problem is difficult to solve numerically.

The level set method (3.17) proposed in [102, 103] is an important numerical technique for tracking interfaces and shapes. The advantage of the level set method is that one can perform numerical computations involving curves and surfaces without having to parameterize these objects. The level set method represents the interfaces by embedding them in a higher dimensional function ϕ , referred to as the level set function. Traditionally the evolving interfaces are represented by the zero level set of ϕ , which is usually given as a signed distance function. The level set method has been applied for image processing problems and was first introduced to solve the Mumford-Shah functional by Chan and Vese in [41, 42]. Chan and Vese model achieves great success in image segmentation due to its advantages in obtaining large convergence range and handling the topological changes. A lot further work of Chan and Vese model was done in [104, 105, 42, 43]. Some variants of the level set method, the so-called “Piecewise Constant Level Set Method” (PCLSM), were proposed in [44, 106, 107]. This PCLSM can identify several interfaces by one single level set function, which makes it more suitable to solve the Mumford-Shah model with multiphase cases.

Traditionally, methods based on gradient descent are often used for solving the Mumford-Shah models, see [44, 106, 107]. These methods are normally slow and difficult to find global minimizers. Recently, many work has been done on applying graph cuts algorithms for image segmentation [57, 84, 94, 108]. They are proven to be more efficient for solving this kind of energy minimization problems. The connection between graph cuts and variational problems has been established in [93, 59, 109]. For Mumford-Shah segmentation, some work using graph cuts optimization for two-phase model has been done in [110] and [111]. For multiphase Mumford-Shah model, the methods of [57, 112, 59] can be adopted for solving the corresponding energy minimization problems. In this work, we shall follow the approach given in [113]. In [113], the authors used the level set formulation of Mumford-Shah model [103] and adopted the graph construction method in [59, 85] to multiphase Mumford-Shah model. However, when the images become large and the number of phases increases, especially for 3D segmentation cases, both computational cost and memory usage increase greatly. In this work we try to find some remedies for these difficulties and

show that we could get some algorithms which have quite high efficiency as well as low memory usage. We propose a method combining the Domain Decomposition Method (DDM) with graph cuts algorithms for the Mumford-Shah model for 2D examples and further extend the method to 3D data.

5.2 Mumford-Shah Model with PCLSM

5.2.1 Mumford-Shah model

We have presented the general Mumford-Shah model (3.15) in Chapter 3. In this chapter, we consider the multiphase situation with n interfaces Γ_i involved. We search an approximation image u by minimizing the following energy functional

$$G(u, \Gamma_i) = \int_{\Omega} (u - f)^2 d\Omega + \alpha \int_{\Omega \setminus \bigcup_i \Gamma_i} |\nabla u|^2 d\Omega + \sum_{i=1}^n \beta \int_{\Gamma_i} ds. \quad (5.1)$$

where α and β are nonnegative constants and $\int_{\Gamma_i} ds$ is the length of the boundary of interfaces Γ_i . The most popular way to solve this minimization problem is to apply the level set method [41], especially the piecewise constant level set Mumford-Shah model. For such case, the second term vanishes in the minimization functional.

5.2.2 Piecewise constant level set method

In [44, 106, 107], the piecewise constant level set method (PCLSM) was proposed and applied to the Mumford-Shah model. The main idea of PCLSM is to seek a partition of the domain Ω into n subdomains Ω_i , $i = 1, 2, \dots, n$. The essential idea is to use a piecewise constant level set function ϕ to identify the subdomains

$$\phi = i \quad \text{in } \Omega_i. \quad (5.2)$$

Once the function ϕ is identified, we can construct the corresponding characteristic

functions for each subdomain Ω_i as

$$\psi_i = \frac{1}{v_i} \prod_{\substack{j=1 \\ j \neq i}}^n (\phi - j), \quad \text{with} \quad v_i = \prod_{\substack{k=1 \\ k \neq i}}^n (i - k). \quad (5.3)$$

If ϕ is defined as in (5.2), we have $\psi_i(x) = 1$ for $x \in \Omega_i$, otherwise we have $\psi_i(x) = 0$. Based on these characteristic functions, we can extract the geometrical information of the boundaries of the subdomains $\{\Omega_i\}_{i=1}^n$. For example, the length of the interfaces surrounding each subdomain Ω_i , $i = 1, 2, \dots, n$, should be

$$\text{Length}(\partial\Omega_i) = \int_{\Omega} |\nabla \psi_i| d\Omega. \quad (5.4)$$

For some given values $c_i, i = 1, 2, \dots, n$, define

$$u = \sum_{i=1}^n c_i \psi_i. \quad (5.5)$$

We have $u = c_i$ in the corresponding subdomain Ω_i , if ϕ satisfies (5.2). In the next subsection, we shall use this idea for image segmentation with the Mumford-Shah model.

5.2.3 The minimization problem

We assume u is a piecewise constant function as given in (5.5) and ϕ is the corresponding level set function (5.2). The multiphase piecewise constant Mumford-Shah model is to solve the following minimization problem

$$\min_{\mathbf{c} \in \mathbb{R}^n, \phi \in \{1, 2, \dots, n\}} G(\mathbf{c}, \phi), \quad G(\mathbf{c}, \phi) = \int_{\Omega} (u - f)^2 d\Omega + \beta \sum_{i=1}^n \int_{\Omega} |\nabla \psi_i| d\Omega. \quad (5.6)$$

We use total variation (TV) of the characteristic function to replace the last term of the Mumford-Shah functional, measuring the length of the interfaces. Such an approach has also been used in other segmentation models in [114, 115]. The

piecewise constant level set function ϕ can be expressed in terms of its characteristic functions by

$$\phi = \sum_{i=1}^n i\psi_i(\phi).$$

The gradient of ϕ and $\psi_i(\phi)$ are respectively

$$\nabla\phi = \sum_{i=1}^n i\nabla\psi_i(\phi) \quad \text{and} \quad \nabla\psi_i = \psi'_i(\phi)\nabla\phi,$$

and resultantly we have

$$|\nabla\phi| \leq \sum_{i=1}^n |i||\nabla\psi_i(\phi)| \quad \text{and} \quad \sum_{i=1}^n |\nabla\psi_i(\phi)| \leq \sum_{i=1}^n |\psi'_i(\phi)||\nabla\phi|.$$

Therefore, the following relationship exists

$$\rho_1(n) \int_{\Omega} |\nabla\phi| d\Omega \leq \sum_{i=1}^n \int_{\Omega} |\nabla\psi_i(\phi)| d\Omega \leq \rho_2(n) \int_{\Omega} |\nabla\phi| d\Omega, \quad (5.7)$$

where $\rho_1(n)$ and $\rho_2(n)$ only depend on n .

Unless “symmetry” is a crucial issue for the segmentation problem, we replace the regularization term in $G(\mathbf{c}, \phi)$ by an approximative functional and solve the following minimization problem

$$\min_{\mathbf{c} \in \mathbb{R}^n, \phi \in \{1, 2, \dots, n\}} G(\mathbf{c}, \phi), \quad G(\mathbf{c}, \phi) = \int_{\Omega} (u - f)^2 d\Omega + \beta \int_{\Omega} |\nabla\phi| d\Omega. \quad (5.8)$$

This functional is the Mumford-Shah model we used in the paper. In [44, 106, 107], the constrained optimization problem (5.8) was solved by finding the saddle point of the corresponding augmented Lagrangian functional. In these methods, some iterative numerical methods are used to solve the corresponding Euler-Lagrange equations, such as gradient decent time marching scheme. In the next section, we shall construct a graph and solve the minimization problem (5.8) by the graph cuts algorithms as in [59, 113].

5.3 Graph Cuts for Multiphase Mumford-Shah Model

Instead of solving the Euler-Lagrange equation, graph cuts algorithms have been proposed to solve the minimization problem (5.8). We give a review of this algorithm in the following.

5.3.1 Discretization of energy functional

We have introduced the discrete setting for the total variation models in Chapter 4. Here, the same set of pixel $\mathcal{P}$ and neighborhood system are used to discretize the Mumford-Shah model for 2D images. Therefore, we can write the discrete Mumford-Shah model (5.8) for 2D images directly using the same notations as

$$G_d(\mathbf{c}, \phi) = \sum_{p \in \mathcal{P}} |u_p - f_p|^2 + \beta \sum_{p \in \mathcal{P}, q \in \mathcal{N}_p^\kappa} w_{p,q} |\phi_p - \phi_q|. \quad (5.9)$$

Above, $\mathcal{N}_p^\kappa$, $\kappa = 4$ or 8 is the neighborhood and $w_{p,q}$ is the corresponding weight for the discretized TV-term as in (4.6) and (4.7), see also [113]. f_p is the intensity value of f at $p \in \mathcal{P}$ and u_p is related to ϕ_p as in (5.5) and (5.3). We assume that the value of $c_i, i = 1, 2, \dots, n$ are known. For boundary points p , $\mathcal{N}_p^\kappa$ has less neighboring points.

Here, we focus on introducing the discrete form of the Mumford-Shah model for 3D images. Let $\mathcal{P}^{3D}$ denote the index set of the pixels, i.e.,

$$\mathcal{P}^{3D} = \{(i, j, k) \mid i = 1, \dots, N_x, j = 1, \dots, N_y, k = 1, \dots, N_z\}. \quad (5.10)$$

For 3D segmentation problems, we can adopt the neighborhood involving either 6 neighbors or 26 neighbors at site $p \in \mathcal{P}^{3D}$, which are defined respectively as

$$\mathcal{N}_p^6 = \{(i \pm 1, j, k), (i, j \pm 1, k), (i, j, k \pm 1)\} \cap \mathcal{P}, \quad (5.11)$$

$$\begin{aligned} \mathcal{N}_p^{26} = \{(i \pm 1, j, k), (i, j \pm 1, k), (i, j, k \pm 1), (i \pm 1, j \pm 1, k), \\ (i \pm 1, j, k \pm 1), (i, j \pm 1, k \pm 1), (i \pm 1, j \pm 1, k \pm 1)\} \cap \mathcal{P}. \end{aligned} \quad (5.12)$$

For example, if we use the neighborhood involving 6 neighbors to discretize the

regularization term, we will get the following discrete total variation

$$TV^{3D,6}(\phi) = \sum_{i,j,k} \left(|\phi_{i+1,j,k} - \phi_{i,j,k}| + |\phi_{i,j+1,k} - \phi_{i,j,k}| + |\phi_{i,j,k+1} - \phi_{i,j,k}| \right),$$

together with the discrete data fitting term, the Mumford-Shah model (5.8) is then defined in the discrete setting for 3D images.

To sum up, for both 2D and 3D images, the Mumford-Shah model (5.8) can be transformed into discrete forms which are graph representable. Therefore, discrete Mumford-Shah model can be efficiently solved by the max-flow/min-cut algorithm.

5.3.2 Graph construction

Recently, a special graph has been constructed in [59, 113] to solve the multiphase Mumford-Shah model. In this subsection, we briefly review the essential ideas followed the instruction in [113].

Figure 5.1: Explanation of the multiphase graph cuts.

To use graph cuts algorithm for the multiphase segmentation problems, we have to introduce an extra dimension, i.e., we construct graph with one dimension higher than the original image. For a 2D image of size $M \times N$, we construct a graph in 3D

containing $M \times N \times (n - 1)$ vertices. Specifically, we have $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and

$$\mathcal{V} = \{v_{p,l} \mid (p, l) \in \mathbb{R}^2 \times \mathbb{R} \mid p \in \mathcal{P}, l \in \{1, \dots, n - 1\}\}. \quad (5.13)$$

The edges $\mathcal{E}$ are divided into two groups: $\mathcal{E}_D$ corresponding to the data fitting term in (5.9) and $\mathcal{E}_R$ corresponding to the TV term in (5.9). They are defined, respectively, as

$$\mathcal{E}_D = \cup_{p \in \mathcal{P}} \{(s, v_{p,1}) \cup_{l=1}^{n-2} (v_{p,l}, v_{p,l+1}) \cup (v_{p,n-1}, t)\}, \quad (5.14)$$

$$\mathcal{E}_R = \{(v_{p,l}, v_{q,l}) \mid p \in \mathcal{P}, q \in \mathcal{N}_p^\kappa, l \in \{1, \dots, n - 1\}\}. \quad (5.15)$$

In Figure 5.1, the graph for a 1D signal with 4-phase segmentation is shown. The edges in $\mathcal{E}_D$ are illustrated as the vertical arrows while the edges in $\mathcal{E}_R$ are illustrated as the horizontal arrows in Figure 5.1. A cut is called admissible if it only serves one vertical edge for each $p \in \mathcal{P}$ (c.f., [113]). In order to exclude non-admissible cut, we introduce an artificial constant $\omega > 0$ and define the capacity of the edges as

$$c(s, v_{p,1}) = |f_p - c_1|^2 + \frac{\omega}{MN}, \quad \forall p \in \mathcal{P}, \quad (5.16)$$

$$c(v_{p,l}, v_{p,l+1}) = |f_p - c_{l+1}|^2 + \frac{\omega}{MN}, \quad \forall p \in \mathcal{P}, \quad \forall l \in \{1, \dots, n - 2\}, \quad (5.17)$$

$$c(v_{p,n}, t) = |f_p - c_n|^2 + \frac{\omega}{MN}, \quad \forall p \in \mathcal{P}, \quad (5.18)$$

$$c(v_{p,l}, v_{q,l}) = \beta w_{p,q}, \quad \forall p \in \mathcal{P}, \quad \forall q \in \mathcal{N}_p^\kappa, \quad \forall l \in \{1, \dots, n - 1\}. \quad (5.19)$$

In the above, β is the regularization parameter, $w_{p,q}$ is the weight for the discretization of the TV norm and $\mathcal{N}_p^\kappa$ is the set containing the neighbors of $p \in \mathcal{P}$ used in the discretization.

After adding all edges to the graph, we can solve the minimization by using the max-flow/min-cut algorithm. We emphasize that the segmentation problem is transferred from the size of $M \times N$ to the size of $M \times N \times (n - 1)$.

5.3.3 An iterative segmentation scheme

In the last section, we show that graph cuts algorithms can be used to solve the Mumford-Shah minimization problem when the values of $\mathbf{c}$ are known. For minimization problem (5.8), we also need to estimate the $\mathbf{c}$ value and the following algorithm – Algorithm 6, is rather robust and converges fast.

Algorithm 6 Graph cuts segmentation algorithm

Choose initial values for $\mathbf{c}^0$, set $k = 0$.

while ($\|\mathbf{c}^k - \mathbf{c}^{k-1}\|_\infty > TOL$)

1. Use graph cuts to compute ϕ^{k+1} from

$$\phi^{k+1} = \arg \min_{\tilde{\phi}} G_d(\mathbf{c}^k, \tilde{\phi}). \quad (5.20)$$

2. Compute the characteristic functions $\{\psi_j^{k+1}\}_{j=1}^n$ from ϕ^{k+1} , c.f., (5.3).

3. Update $\mathbf{c}^{k+1}$ by

$$c_j^{k+1} = \frac{\sum_{p \in \mathcal{P}} f_p \psi_{j,p}^{k+1}}{\sum_{p \in \mathcal{P}} \psi_{j,p}^{k+1}}, \quad j = 1, \dots, n. \quad (5.21)$$

4. Update $k \leftarrow k + 1$.
-

In Algorithm 6, the initial values $\mathbf{c}^0$ are computed very efficiently by the ISODATA algorithm; see [116]. We control the convergence of $\mathbf{c}$ using the maximum norm, which is defined as

$$\|\mathbf{c}\|_\infty := \max\{|c_1|, \dots, |c_n|\}.$$

For segmentation problems, the above iterative procedure normally converges in about 5-6 iterations. Compared with traditional gradient decent methods, it is normally 500 times faster for relatively large size 2D images we have tested; see [113]. However, when the image size is very large, the memory requirement and computational cost become a challenging problem.

5.4 Graph Cuts Algorithms with Domain Decomposition Method

As we discussed, when the images become large, the computational and memory cost of the multiphase graph cuts algorithm increases greatly. This causes problems for some data set with very large size, especially for 3D applications. Therefore, we consider to use DDM to overcome these difficulties.

5.4.1 Non-overlapping domain decomposition method

We assume image domain Ω is decomposed into 4 kinds of non-overlapping subdomains as shown in Figure 4.2. The subdomains intersect only on their interfaces, which is the case overlapping size $\delta = 0$ in Figure 4.3. More precisely, we express Ω as a union of non-overlapping subdomains Ω_i , $i = 1, \dots, 4$,

$$\Omega = \bigcup_{i=1}^4 \Omega_i, \quad \Omega_i \cap \Omega_j = \emptyset, \quad i \neq j.$$

We denote $\mathcal{P}_i \subset \mathcal{P}$, $i = 1, 2, 3, 4$, the index sets for the grid points of the subdomains. Corresponding to each kind of subdomains Ω_i , $i = 1, 2, 3, 4$, we define the following energy functional

$$G_d^i(\mathbf{c}, \phi) = \sum_{p \in \mathcal{P}_i} |u_p - f_p|^2 + \beta \sum_{p \in \mathcal{P}_i, q \in \mathcal{N}_p^* \cap \mathcal{P}_i} w_{p,q} |\phi_p - \phi_q|. \quad (5.22)$$

Thus, the non-overlapping algorithm solves the subproblems over Ω_i from $i = 1$ to $i = 4$ to cover the whole domain instead of computing Mumford-Shah model (5.8) over Ω . We give the non-overlapping domain decomposition based algorithm for the Mumford-Shah model as Algorithm 7.

Here and later, we denote $\phi|_{\Omega_i}$ be the value of ϕ in Ω_i . For minimization problem (5.23), we only need to use graph cuts algorithms to find the values of ϕ^{k+1} in Ω_i . Each Ω_i contains many disjoint subdomains, i.e., $\Omega_i = \bigcup_j \Omega_{i,j}$. As the subproblems over $\Omega_{i,j}$ are independent of each other, we can use the graph cuts algorithms to

Algorithm 7 Non-overlapping Domain Decomposition Method

Choose initial values for $\mathbf{c}^0$, set $k = 0$.

While ($\|\mathbf{c}^k - \mathbf{c}^{k-1}\|_\infty > TOL$)

1. For $i=1, 2, 3, 4$, let $\phi^{k+\frac{i}{4}} = \phi^{k+\frac{i-1}{4}}$ in $\Omega \setminus \Omega_i$ and use a graph cuts algorithm to compute $\phi^{k+\frac{i}{4}}|_{\Omega_i}$ from

$$\phi^{k+\frac{i}{4}}|_{\Omega_i} = \arg \min_{\tilde{\phi}} G_d^i(\mathbf{c}^k, \tilde{\phi}). \quad (5.23)$$

2. Compute the characteristic functions $\{\psi_j^{k+1}\}_{j=1}^n$ from ϕ^{k+1} , c.f. (5.3).
3. Update $\mathbf{c}^{k+1}$ according to the following discrete formula for $\mathbf{c}$

$$c_j^{k+1} = \frac{\sum_{p \in \mathcal{P}} f_p \psi_{j,p}^{k+1}}{\sum_{p \in \mathcal{P}} \psi_{j,p}^{k+1}}, \quad j = 1, \dots, n. \quad (5.24)$$

4. Update $k \leftarrow k + 1$.

solve the subdomain problems simultaneously. If we have parallel computers, these subdomain problems can be solved in parallel. In our implementations, we just solve the subproblems one by one. Even so, the computational cost is reduced compared to solving by the graph cuts algorithm in the whole domain.

For a given $p \in \mathcal{P}_i$ on the boundary $\partial\Omega_i$ of Ω_i , the subdomain energy functional G_d^i only includes regularization terms related to $q \in \mathcal{N}_p^\kappa \cap \mathcal{P}_i$, i.e., the subdomain problems only regularize with points inside the subdomain. Thus, this will cause some errors compared with Algorithm 6. Due to the reason that the $c_j, j = 1, 2, \dots, n$ are computed globally, it shows that the non-overlapping algorithm has always been able to find a good segmentation despite this error.

5.4.2 Overlapping domain decomposition method

Unlike the non-overlapping case, the subdomains overlap with each other in overlapping DDM. Therefore, overlapping DDM corresponds to the case with $\delta \geq 1$ in

Figure 4.3. More specifically, the image domain Ω is decomposed into the overlapping subdomains Ω_i , $i = 1, \dots, 4$, as follows

$$\Omega = \bigcup_{i=1}^4 \Omega_i, \quad \Omega_i \cap \Omega_j \neq \emptyset, \quad i \neq j.$$

Simultaneously, the index sets $\mathcal{P}_i$ also have overlaps. To explain the algorithm clearly, we need to introduce some notations. We use Ω_i^0 to denote the interior grid points of Ω_i and $\partial\Omega_i$ to denote the boundary grid points of Ω_i . Correspondingly, $\mathcal{P}_i^0$ is the index set for Ω_i^0 and $\partial\mathcal{P}_i$ is the index set for $\partial\Omega_i$.

In subdomain Ω_i , $i = 1, \dots, 4$, we formulate the energy problem as follows

$$G_d^i(\mathbf{c}, \phi) = \sum_{p \in \mathcal{P}_i^0} |u_p - f_p|^2 + \beta \sum_{p \in \mathcal{P}_i^0, q \in \mathcal{N}_p^\kappa} w_{p,q} |\phi_p - \phi_q|. \quad (5.25)$$

Therefore, the overlapping domain decomposition based algorithm for solving the Mumford-Shah model (5.8) is given as Algorithm 8.

However, as the subdomains overlap with each other, solving (5.26) is quite different from solving (5.23). The value of $\phi^{k+\frac{i}{4}}$ is equal to $\phi^{k+\frac{i-1}{4}}$ in $\Omega \setminus \Omega_i^0$ and thus has no need for computation. The value of $\phi^{k+\frac{i}{4}}$ in Ω_i^0 needs to be solved through (5.26). For a point $p \in \mathcal{P}_i^0$, $\mathcal{N}_p^\kappa$ may be outside Ω_i^0 . However, this does not cause any problem for solving (5.26) as the value outside Ω_i^0 is already known. This will take care of the regularization between the subdomains. The implementation for (5.26) is very similar to the situation in Chapter 4 so that we will not common to much here.

For this algorithm, we have, (c.f., [61])

$$\begin{aligned} G_d(\mathbf{c}^{k+1}, \phi^{k+1}) &\leq G_d(\mathbf{c}^k, \phi^{k+1}) \leq G_d(\mathbf{c}^k, \phi^{k+3/4}) \leq G_d(\mathbf{c}^k, \phi^{k+1/2}) \\ &\leq G_d(\mathbf{c}^k, \phi^{k+1/4}) \leq G_d(\mathbf{c}^k, \phi^k). \end{aligned}$$

This guarantees the monotonicity of the cost functional and thus gives a robust algorithm.

Algorithm 8 Overlapping Domain Decomposition Method

Choose initial values for $\mathbf{c}^0$ and ϕ^0 . Set $k = 0$.

While ($\|\mathbf{c}^k - \mathbf{c}^{k-1}\|_\infty > TOL$)

1. For $i=1, 2, 3, 4$, let $\phi^{k+\frac{i}{4}} = \phi^{k+\frac{i-1}{4}}$ in $\Omega \setminus \Omega_i^0$ and use a graph cuts algorithm to compute $\phi^{k+\frac{i}{4}}|_{\Omega_i^0}$ from

$$\phi^{k+\frac{i}{4}}|_{\Omega_i^0} = \arg \min_{\tilde{\phi}} G_d^i(\mathbf{c}^k, \tilde{\phi}). \quad (5.26)$$

2. Compute the characteristic functions $\{\psi_j^{k+1}\}_{j=1}^n$ from ϕ^{k+1} , c.f. (5.3).
3. Update $\mathbf{c}^{k+1}$ according to the following discrete formula for $\mathbf{c}$

$$c_j^{k+1} = \frac{\sum_{p \in \mathcal{P}} f_p \psi_{j,p}^{k+1}}{\sum_{p \in \mathcal{P}} \psi_{j,p}^{k+1}}, \quad j = 1, \dots, n. \quad (5.27)$$

4. Update $k \leftarrow k + 1$.

5.5 Numerical Experiments

In the following, we implement our domain decomposition based algorithms on 2D synthetic and real data and 3D real data respectively. We develop our codes in C++ using the augmenting path algorithm proposed in [84]. All 2D numerical experiments are performed on a HP xw4600 Workstation with an Intel(R) Core(TM) 2 Duo CPU E6750 @ 2.66 GHz, 2.67 GHz and 2.00 GB of RAM and 3D experiments are developed in LINUX environment.

In our implementations, we take $TOL = 0.1$, the phase number $n = 4$ and $\omega = \kappa(n-1)\beta$. The neighborhood κ is either $\kappa = 4$ or $\kappa = 8$ for 2D examples and $\kappa = 6$ or $\kappa = 26$ for 3D examples. The value of β varies with the examples. For Algorithm 8, we always take $\phi^0 = 0$ and use ISODATA algorithm of [116] to get the initial values for $\mathbf{c}$. Besides, the size of overlapping is one pixel unless specified otherwise.

5.5.1 Tests of DD based algorithm on 2D experiments

First of all, we set up a series of experiments to demonstrate the effectiveness of our Domain Decomposition (DD) based algorithms: Algorithm 7 and Algorithm 8. These experiments are developed on four 1024×1024 images: Lena, Lake, Tree and Clock. We apply four phase DD based algorithms together with Algorithm 6 to these four images. For all the algorithms, we choose the same neighborhood involving 4 connectivities and set $\beta = 500$. For DD based algorithms, the image domain is decomposed into 16×16 subdomains. The segmentation results of each algorithm are displayed in Figure 5.2 and the corresponding computational time is tabulated in Table 5.2.

In the following, we first test the sensitivity of the subdomain size d to Algorithm 7 and Algorithm 8. We examine the computational time and approximation error (i.e., difference between the solution over the whole domain and the solution from DDM) of both DD based algorithms while the subdomain number is increased from 4 to 128 subdomains. Then we compare the memory consumption and minimal energy between DD based algorithms and Algorithm 6 for these four images. For these tests, we fix $\beta = 500$ for all algorithms.

I. Sensitivity to the subdomain size

The subdomain size d greatly affects the performance of Algorithm 7 and Algorithm 8. Therefore, we test the computational cost of DD based algorithms with regard to different decompositions of the image domain. Precisely, the image domain is decomposed into 2^2 , 2^3 , 2^4 , 2^5 , 2^6 and 2^7 subdomains with subdomain size $d = 512$, $d = 256$, $d = 128$, $d = 64$, $d = 32$ and $d = 16$ respectively. We implement DD based Algorithm 7 and Algorithm 8 to the image domain with all these decompositions. The computational time of non-overlapping Algorithm 7 and overlapping Algorithm 8 are plotted in Figure 5.3(a) and Figure 5.3(b) respectively. Through the results, we see that DD based algorithms can improve the computational efficiency compared to Algorithm 6.

Meanwhile, we evaluate an error estimation between DD based algorithms and Algorithm 6 with different decompositions. Continuing the experiment, we let the

Figure 5.2: The comparison results of Lena, Lake, Tree and Clock with 16×16 sub-domains. From left to right: given image, restored images by Algorithm 6, Algorithm 7 and Algorithm 8 respectively.

(a) Computational cost from Algorithm 7 (b) Computational cost from Algorithm 8

Figure 5.3: Computational costs of non-overlapping Algorithm 7 and overlapping Algorithm 8 with different subdomain sizes.

images be decomposed into $2^2, 2^3, 2^4, 2^5, 2^6$ and 2^7 subdomains and use the same parameters. The error is computed by comparing the segmentation results of Algorithm 7 or Algorithm 8 with the results of Algorithm 6 from the following formula

$$\frac{\sum_{p \in \mathcal{P}} \chi(\phi_p, \phi_p^0)}{M \times N}, \quad (5.28)$$

where ϕ_p^0 is from Algorithm 6 and ϕ_p is either from non-overlapping Algorithm 7 or from overlapping Algorithm 8. If $\phi_p = \phi_p^0$, we have $\chi(\phi_p, \phi_p^0) = 0$, otherwise we have $\chi(\phi_p, \phi_p^0) = 1$. We estimate the error between Algorithm 7 and Algorithm 6 and show the result in Figure 5.4(a) while we exhibit the similar result between Algorithm 8 and Algorithm 6 in Figure 5.4(b). From the results, it is clear that the error of non-overlapping algorithm is always somewhat larger than the corresponding error of overlapping algorithm. Since the error of overlapping Algorithm 8 keeps less than or around 0.2% when the images are decomposed from 4 to 128 subdomains, we can say that overlapping domain decomposition based algorithm is quite a stable method for numerical application.

II. Comparison of memory consumption

Figure 5.4: Approximation error of non-overlapping Algorithm 7 and overlapping Algorithm 8 with different subdomain sizes.

Table 5.1: The memory consumption of Lena, Lake, Tree and Clock. In the table, the unit of the number is kiloBytes.

Problem	Subdomain Number	Algorithm 6	Algorithm 7	Algorithm 8
Lena	16×16	587,262	28,560	31,960
Lake	16×16	551,202	28,560	31,404
Tree	16×16	541,124	28,560	31,888
Clock	16×16	552,272	29,040	31,904

Next, we analyze the memory consumptions of DD based algorithms and Algorithm 6. As the beginning, the images are decomposed into 16×16 subdomains. We record the largest memory demanded by each algorithm of each image in the experiment and tabulate them in Table 5.1. Therefore, from the table we can see, for a 1024×1024 image, Algorithm 6 requires a memory around $600MB$ while DD based algorithms only need less than 10% of this amount. It is proved that DD based algorithms can greatly decrease the memory requirements compared to Algorithm 6. Thus, non-overlapping Algorithm 7 and overlapping Algorithm 8 can make segmentation problem (5.8) with large data size be solvable on computers with small memory.

Table 5.2: Computational time in seconds for 2D experiments: Lena, Lake, Tree, Clock and MR.

Prob	Size	Neighbor	Subdomain No	Algorithm 6	Algorithm 7	Algorithm 8
Lena	1024×1024	4	16×16	51.641	38.797	31.688
Lake	1024×1024	4	16×16	57.45	44.657	42.875
Tree	1024×1024	4	16×16	29.859	22.594	25.86
Clock	1024×1024	4	16×16	74.203	56.438	63.344
MR	670×530	8	10×10	22.844	15.562	14.813

III. Comparison of minimal energy

Moreover, we try to illustrate that the minimal energy obtained by DD based algorithms approximates the minimal energy of Algorithm 6, which is recognized as the global minimizer over the image domain. We continue to decompose the image domain into 16×16 subdomains as before. The energies of DD based algorithms are calculated using the cut results over the entire domain. For Algorithm 7, we add all the weights of the cut edges in each subdomain while we add the weights of the internal cut edges in each subdomain for Algorithm 8. We display the comparison results of each image in Figure 5.5. In the figure, the energy of Algorithm 6 is marked by “*” and the energies of non-overlapping and overlapping decomposition are denoted by “+” and “o” respectively. Through the experiment results, we see that the energy of DD based algorithms is convergent and the difference between the energy of DD based algorithms and the energy of Algorithm 6 is acceptable to us.

One more 2D experiment, we implement our DD based algorithms to a real brain MR image of high resolution. We use four phase image segmentation approach to extract the 4 different classes of the brain image. We segment the MR image with TV^8 norm and display the results in Figure 5.6 and the computational time in Table 5.2. We can see that our decomposition methods can get almost the same result as by Algorithm 6 visually. In the meantime, the decomposition methods improve more than $\frac{1}{4}$ of Algorithm 6’s computational time.

Figure 5.5: The energy comparison between Algorithm 6 (black curve), Algorithm 7 (blue curve) and Algorithm 8 (red curve).

5.5.2 3D experiments

In this subsection, we test our non-overlapping Algorithm 7 and overlapping Algorithm 8 to five 3D real data with name and size as follows: MRI($250 \times 250 \times 120$), Blow($512 \times 512 \times 512$), CGQ($512 \times 512 \times 512$), Hailuo($500 \times 500 \times 150$) and Huluo($512 \times 512 \times 512$).

At the beginning, we implement our multiphase graph cuts algorithms to a 3D MRI data. In this test, we apply four phase algorithms to the data and use $1000(10 \times 10 \times 10)$ subdomains for both Algorithm 7 and Algorithm 8. For the neighborhood, we adopt both 6 and 26 connectivities for this 3D data respectively in the experiment. We display the computational time of each algorithm with two different connectivities in Table 5.5 and the surface results in Figure 5.7. Meantime, for different connectivities, we give the comparison results of a chosen slice of the data from each algorithm in

Figure 5.6: The comparison results of MR.

Figure 5.8. From the detail of the results in Figure 5.8, it is easy to find out that overlapping Algorithm 8 approximates the result of Algorithm 6 better than non-overlapping Algorithm 7. During the experiment, we also increase the subdomain number to be $20 \times 20 \times 20$ to see the effect of DD in improving the computational cost compared to Algorithm 6. The computational time is also shown in Table 5.5.

Secondly, we apply six phase algorithms to segment the data CGQ. This CGQ is a real machine component with noise. Since this data is too large for Algorithm 6 to handle, we extract a $256 \times 256 \times 256$ data from initial data to compare the computational time between Algorithm 6 and DD based algorithms. For DD based Algorithm 7 and Algorithm 8, we use $32 \times 32 \times 32$ subdomains for both data of size either $512 \times 512 \times 512$ or $256 \times 256 \times 256$. We choose one slice from the results of Algorithm 6 and overlapping Algorithm 8 with data size $256 \times 256 \times 256$ and show these phase results one by one in Figure 5.9. Meanwhile, we display some surface results from overlapping Algorithm 8 in Figure 5.10 and the computational time of each algorithm in Table 5.5. For the initial $512 \times 512 \times 512$ data, we fix the iteration being 11 for both Algorithm 7 and Algorithm 8 and record the corresponding computational time in Table 5.5.

For the data Blow, we implement three phase segmentation algorithms and adopt the neighborhood with 6 connectivity. Similarly, we extract a $256 \times 256 \times 256$ data from initial Blow data for the comparison of computational cost between Algorithm 6 and DD based algorithms. For Algorithm 7 and Algorithm 8, we use $32 \times 32 \times 32$ subdomains for Blow data of size either $512 \times 512 \times 512$ or $256 \times 256 \times 256$. The

(a) Grey Matter

(b) White Matter

Figure 5.7: The surface display of reconstructed grey and white matter. All the figures come from the overlapping Algorithm 8.

numerical results of Blow are provided in Figure 5.11 and the computational time of different algorithms is shown in Table 5.5.

Besides, we develop a series of tests on Blow to show the advantages of our DD based algorithms. Firstly, we give a comparison of the computational time with different sizes of data extracted from Blow to illustrate the necessity of DDM. We tabulate the computational time from Algorithm 6 with different sizes of data in Figure 5.12(a) and use Figure 5.12(b) to manifest that DD can greatly reduce the computational cost when solving the segmentation problem. Then we compute the approximation error between DD based algorithms and Algorithm 6 and list both computational time and approximation error of non-overlapping Algorithm 7 and overlapping Algorithm 8 in Table 5.3. From the numbers in Table 5.3, they show that

Figure 5.8: The comparison results of MRI. Row one are comparison results with 6 connectivity and Row two are with 26 connectivity.

our DD based algorithms require less time and keep a quite small error compared to Algorithm 6, especially overlapping Algorithm 8 which always keeps the error under 0.2%. Furthermore, we tabulate the largest physical memory used in the experiment regarding to these extracted data from Blow in Table 5.4. Corresponding to 2D experiment results, both DD based algorithms save considerable quantity of memory and make large size data be solvable for us.

For the test of Hailuo and Huluo, we apply two phase segmentation algorithms and adopt the neighborhood involving 6 pixels. We use $20 \times 20 \times 20$ subdomains for Hailuo experiment and $32 \times 32 \times 32$ subdomains for Huluo experiment when we implement both non-overlapping Algorithm 7 and overlapping Algorithm 8. The numerical results of Algorithm 8 of these two data are supplied in Figure 5.13 and the computational time of each algorithm to both data is displayed in Table 5.5. These two phase experiments also demonstrate that DD can make large size data be segmented and save much more computational cost in the meantime.

Figure 5.9: The CGQ results of slice nr.: 150. Row one from left to right are results of phase 1 to phase 6 of Algorithm 6. Row two from left to right are results of phase 1 to phase 6 of overlapping Algorithm 8.

Table 5.3: Computational costs and error estimation of Algorithm 7 and Algorithm 8 for Blow experiment. The corresponding computational time of Algorithm 6 are listed in Figure 5.12(a).

Problem Size	β	Subdomain Number	Algorithm 7		Algorithm 8	
			time	error	time	error
$256 \times 256 \times 256$	250	$32 \times 32 \times 32$	143.54	0.25%	202.03	0.053%
$128 \times 128 \times 128$	250	$16 \times 16 \times 16$	16.01	0.32%	22.84	0.16%
$64 \times 64 \times 64$	250	$8 \times 8 \times 8$	2.8	0.33%	5.7	0.039%
$32 \times 32 \times 32$	250	$4 \times 4 \times 4$	0.39	0.41%	0.77	0.042%
$16 \times 16 \times 16$	250	$2 \times 2 \times 2$	0.05	0.097%	0.06	0%
$8 \times 8 \times 8$	250	$2 \times 2 \times 2$	0.002	0.059%	0.004	0%

Figure 5.10: The surface results of 6 phase experiment: CGQ from overlapping Algorithm 8. All the results are from extracted data with proportion 1:2 to initial image.

Table 5.4: Memory consumption of Algorithm 6, Algorithm 7 and Algorithm 8 for Blow experiment. The numbers are physical memory that a task has used (in kilo-Bytes) in experiment.

Problem Size	β	Subdomain Number	Algorithm 6	Algorithm 7	Algorithm 8
$512 \times 512 \times 512$	500	$32 \times 32 \times 32$	—	4,723,000	2,847,972
$256 \times 256 \times 256$	250	$32 \times 32 \times 32$	10,724,676	723,276	391,836
$128 \times 128 \times 128$	250	$16 \times 16 \times 16$	1,322,256	91,704	50,956
$64 \times 64 \times 64$	250	$8 \times 8 \times 8$	188,912	15,840	8,504
$32 \times 32 \times 32$	250	$4 \times 4 \times 4$	21,760	3,500	3,240
$16 \times 16 \times 16$	250	$2 \times 2 \times 2$	3,676	1,728	1,944
$8 \times 8 \times 8$	250	$2 \times 2 \times 2$	1,428	1,192	1,376

Figure 5.11: The surface results of 3 phase experiment: Blow from overlapping Algorithm 8. All the results are from extracted data with proportion 1:2 to initial image.

Image Size	CPU Time
$256 \times 256 \times 256$	581.21
$128 \times 128 \times 128$	90.93
$64 \times 64 \times 64$	13.54
$32 \times 32 \times 32$	1.21
$16 \times 16 \times 16$	0.07
$8 \times 8 \times 8$	0.004

(a)

(b)

Figure 5.12: Computational costs in seconds for different resolution. In (b), numbers on x-coordinate denote the image sizes in table (a).

(a) Hailuo

(b) Huluo

Figure 5.13: The surface results of 2 phase examples from overlapping Algorithm 8. The results are from extracted data with proportion 1:2 to initial image.

Table 5.5: Computational time in seconds for 3D experiments. – denotes problem can not be handled by the method.

Problem	Data Size	β	Neighborhood	Phase No.	Subdomain No.	ALG 6	ALG 7	ALG 8
MRI	$250 \times 250 \times 120$	500	6	4	$10 \times 10 \times 10$	713.28	501.96	674.57
MRI	$250 \times 250 \times 120$	300	26	4	$10 \times 10 \times 10$	4897.16	2172.37	2629.56
MRI	$240 \times 240 \times 120$	500	6	4	$20 \times 20 \times 20$	879.07	313.08	317.25
CGQ	$256 \times 256 \times 256$	500000	6	6	$32 \times 32 \times 32$	6634.12	1793.87	1809.92
CGQ	$512 \times 512 \times 512$	300000	6	6	$32 \times 32 \times 32$	–	10277.2	17401.5
Blow	$256 \times 256 \times 256$	250	6	3	$32 \times 32 \times 32$	621.59	143.54	202.03
Blow	$512 \times 512 \times 512$	500	6	3	$32 \times 32 \times 32$	–	934.4	1280.57
Hailuo	$500 \times 500 \times 300$	500	6	2	$20 \times 20 \times 20$	619.01	102.94	219.04
Huluo	$256 \times 256 \times 256$	2000	6	2	$32 \times 32 \times 32$	245.09	49.53	98.04
Huluo	$512 \times 512 \times 512$	2000	6	2	$32 \times 32 \times 32$	–	400.9	780.29

CHAPTER 6

A Fixed-point ALM for TV Minimization Problems

Contents

6.1	Introduction	107
6.2	The Fixed-point Formulation	109
6.3	Convergence Analysis	113
6.4	First-order Augmented Lagrangian Algorithms	117
6.5	Acceleration of the Fixed-point Algorithm	120
6.6	Applications	121

6.1 Introduction

We have reviewed in Chapter 3 that there are many kinds of numerical methods to solve the ROF model (3.3). Although dual formulations, Bregman techniques and augmented Lagrangian methods give a minimizer from the ROF model (3.1) numerically fast, linear systems still need to be solved either by iterative methods or FFT.

Recently, some first-order methods for total variation minimizations have been studied in [117, 5, 118], that is, optimization methods are based on function values

and gradient evaluations. In these approaches, when dealing with (3.1), the dominant computation effort is a relative cheap matrix-vector multiplication involving A and A^T instead of pursuing the solution of a linear system. In [119], the authors have brought up an efficient algorithm for (3.1) based on its Bregman formulation, which also gets rid of handling with any linear system. From the same point of view, we propose novel algorithms for solving the ROF model (3.1) as well as the following general convex minimization problem based on the augmented Lagrangian formulation

$$\min_u \left\{ J(Au) + H(u) \right\}. \quad (6.1)$$

Let X, Y be two finite-dimensional real vector spaces equipped with an inner product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \| = \langle \cdot, \cdot \rangle^{\frac{1}{2}}$. In (6.1), the map $A : X \rightarrow Y$ is a continuous linear operator with induced norm

$$\|A\| = \max \{ \|Ax\| : x \in X \text{ with } \|x\| \leq 1 \},$$

and $J(\cdot)$ and $H(\cdot)$ satisfy the following assumptions

- $J : Y \rightarrow [0, +\infty]$ is a continuous convex function which is possibly non-smooth,
- $H : X \rightarrow [0, +\infty]$ is a strongly convex function with the modulus $o > 0$ such that

$$\langle \partial H(x) - \partial H(y), x - y \rangle \geq o \|x - y\|^2 \quad \forall x, y \in X,$$

where $\partial H(x)$ denotes the subdifferential of H at the point $x \in X$, which is the set of subgradients of H at x .

We apply a fixed-point based method to the augmented Lagrangian formulation of (6.1). A direct relationship between the proposed fixed-point augmented Lagrangian algorithm and the alternating minimization algorithm in [4] is established in this paper. In addition, we develop two kinds of first-order augmented Lagrangian algorithms for (6.1). The connections among the proposed first-order augmented Lagrangian algorithms, FPALM and AMA are discussed and established.

6.2 The Fixed-point Formulation

To take advantage of the separable structure in model (6.1), we reformulate it into the constrained minimization problem by introducing an auxiliary variable p for Au . Therefore, the convex problem (6.1) is reformulated into a constrained problem as follows

$$\begin{aligned} \min_{u,p} \quad & \left\{ J(p) + H(u) \right\}, \\ \text{s.t.}, \quad & p = Au. \end{aligned} \tag{6.2}$$

We use the augmented Lagrangian method to solve (6.2) and define the corresponding augmented Lagrangian functional of (6.2) as follows

$$\mathcal{L}(u, p; \lambda) = J(p) + H(u) + \langle \lambda, Au - p \rangle + \frac{r}{2} \|Au - p\|_2^2, \tag{6.3}$$

where λ is the Lagrange multiplier and r is a positive constant.

We aim to find the saddle-point of the augmented Lagrangian functional (6.3). It is difficult to compute the saddle-point problem because variable u and p are coupled together in (6.3). Usually one separates the variable u and p and then uses an alternative minimization procedure [120, 13, 17] to solve (6.3). Therefore, we separate (6.3) to be the following two sub-problems:

- The u -sub problem: for given p

$$\min_u \left\{ H(u) + \langle \lambda, Au \rangle + \frac{r}{2} \|Au - p\|_2^2 \right\}. \tag{6.4}$$

- The p -sub problem: for given u

$$\min_p \left\{ J(p) - \langle \lambda, p \rangle + \frac{r}{2} \|Au - p\|_2^2 \right\}. \tag{6.5}$$

For the u -sub problem (6.4), the optimality condition gives us a linear PDE

$$\partial H(u) + A^T \lambda + r A^T (Au - p) = 0. \tag{6.6}$$

It allows us to solve (6.6) by using a wide range of linear iterative methods. Due to the block circulant property of $A^T A$, to best of our knowledge, the FFT can be applied to (6.6) as long as the periodic boundary condition is utilized in the linear system.

Motivated by the desire to search a simpler algorithm for solving (6.4), we make our effort to use the matrix-vector multiplication to reduce the computational cost coming along with solving the linear system. In the proposed approach, the optimal condition for u in the $(k+1)^{th}$ iteration is given by

$$\partial H(u) + A^T \lambda^{k+1} + r A^T (Au^k - p^k) = 0. \quad (6.7)$$

In order to obtain (6.7), we shall apply a fixed-point based approach to the quadratic penalty in (6.4). We also remove the factor $\frac{1}{2}$ to get rid of some factor 2 in the sequel. Strictly speaking, the penalty term in the u -sub problem (6.4) is replaced by an approximation as follows

$$\frac{r}{2} \|Au - p^k\|_2^2 \approx r \langle Au - p^k, Au^k - p^k \rangle.$$

Thus, instead of solving (6.4), the minimizer u^{k+1} is computed by the following minimization problem

$$\min_u \left\{ H(u) + \langle \lambda^{k+1}, Au \rangle + r \langle Au - p^k, Au^k - p^k \rangle \right\}. \quad (6.8)$$

For the p -sub problem (6.5), we can rewrite it to be

$$\min_p \left\{ J(p) + \frac{r}{2} \|p - (Au + \frac{\lambda}{r})\|_2^2 \right\}. \quad (6.9)$$

Therefore, we can find the closed form solution for the variable p , if J is the L^1 -norm, which means

$$J(p) = \int_{\Omega} |p| d\Omega.$$

If we use Uzawa's algorithm to solve the saddle-point system, we propose the

following fixed-point augmented Lagrangian algorithm for (6.1). The algorithm iteratively and alternatively computes the variable u and p according to (6.8) and (6.9), respectively: Set $\lambda^0 = 0$, $p^0 = 0$, $u^0 = f$,

Algorithm 9 Fixed-point Augmented Lagrangian Algorithm

1. Update λ^{k+1} from:

$$\lambda^{k+1} = \lambda^k + r(Au^k - p^k).$$

2. Solve u^{k+1} from:

$$u^{k+1} = \arg \min_u \left\{ H(u) + \langle \lambda^{k+1}, Au \rangle + r \langle Au - p^k, Au^k - p^k \rangle \right\}.$$

3. Solve p^{k+1} from:

$$p^{k+1} = \arg \min_p \left\{ J(p) - \langle \lambda^{k+1}, p \rangle + \frac{r}{2} \|Au^{k+1} - p\|_2^2 \right\}.$$

In Algorithm 9, when solving the u variable, if we relax the Lagrange multiplier λ^{k+1} to λ^k , the minimization problem for the u -sub problem becomes

$$\min_u \left\{ H(u) + \langle \lambda^k, Au \rangle + r \langle Au, Au^k - p^k \rangle \right\}.$$

Together with the following equation of updating the Lagrange multiplier

$$r(Au^k - p^k) = \lambda^{k+1} - \lambda^k, \quad (6.10)$$

the minimization of the variable u in the $(k+1)^{th}$ iteration can be further simplified as follows

$$\min_u \left\{ H(u) + \langle \lambda^{k+1}, Au \rangle \right\}. \quad (6.11)$$

Using the minimization (6.11) to replace the u -sub problem in Algorithm 9, we get the exact Tseng's alternating minimization algorithm in [4]. We describe the AMA for (6.1) as the following alternative algorithm: Set $\lambda^0 = 0$, $p^0 = 0$, $u^0 = f$,

In Algorithm 10, if the gradient descent is applied to u -sub problem, the solution

Algorithm 10 Alternating Minimization Algorithm

1. Update λ^{k+1} from:

$$\lambda^{k+1} = \lambda^k + r(Au^k - p^k).$$

2. Solve u^{k+1} from:

$$u^{k+1} = \arg \min_u \left\{ H(u) + \langle \lambda^{k+1}, Au \rangle \right\}.$$

3. Solve p^{k+1} from:

$$p^{k+1} = \arg \min_p \left\{ J(p) - \langle \lambda^{k+1}, p \rangle + \frac{r}{2} \|Au^{k+1} - p\|_2^2 \right\}.$$

of u in the k^{th} iteration can be achieved as follows

$$u^k = (I - t\partial H)u^{k-1} - tA^T\lambda^k, \quad (6.12)$$

where $t > 0$ is a small enough number.

By substituting (6.12) into the formula to update the Lagrange multiplier, we obtain a new formula for λ^{k+1}

$$\lambda^{k+1} = \lambda^k + r(Au^k - p^k) = \lambda^k + r\left(A((I - t\partial H)u^{k-1} - tA^T\lambda^k) - p^k\right).$$

In the implicit formulation, we arrive at the following formula to compute u in the k^{th} iteration

$$u^k = (I - t\partial H)u^{k-1} - tA^T\lambda^{k+1} = (I - t\partial H)u^{k-1} - tA^T(\lambda^k + r(Au^k - p^k)), \quad (6.13)$$

which is the gradient descent for (6.4). Therefore, combining the u -sub problem (6.13) with the p -sub problem (6.5) gives us the exact augmented Lagrangian method (ALM) for the general convex problem (6.1). We summarize the ALM for solving (6.1) in Algorithm 11 as follows: Set $\lambda^0 = 0$, $p^0 = 0$, $u^0 = f$,

Algorithm 11 Augmented Lagrangian Method

1. Update λ^{k+1} from:

$$\lambda^{k+1} = \lambda^k + r(Au^k - p^k).$$

2. Compute p^{k+1} from:

$$p^{k+1} = \arg \min_p \left\{ J(p) - \langle \lambda^{k+1}, p \rangle + \frac{r}{2} \|Au^k - p\|_2^2 \right\}.$$

3. Compute u^{k+1} from:

$$u^{k+1} = \arg \min_u \left\{ H(u) + \langle \lambda^{k+1}, Au \rangle + \frac{r}{2} \|Au - p^{k+1}\|_2^2 \right\}.$$

6.3 Convergence Analysis

In this section we establish the convergence of Algorithm 9 and Algorithm 10. We adopt the similar method in [17, 121, 122] to prove the convergence. We define a particular norm of u with a positive definite matrix L as follows

$$\|u\|_L := \sqrt{\langle Lu, u \rangle}.$$

Theorem 6.1. *Let the sequence $(u^k, p^k; \lambda^k)$ be generated by Algorithm 9. Assume $L_1 = oI - 2rA^T A$ is positive definite, then*

- $\lim_{k \rightarrow \infty} u^k = u^*.$

Proof. Let $(u^*, p^*; \lambda^*)$ be the saddle-point of the Lagrangian functional (6.3). We have

$$p^* = Au^*.$$

We define the errors of $(u^{k+1}, p^{k+1}; \lambda^{k+1})$ as follows

$$u_e^{k+1} := u^{k+1} - u^*; \quad p_e^{k+1} := p^{k+1} - p^*; \quad \lambda_e^{k+1} := \lambda^{k+1} - \lambda^*.$$

We define two variables $t^k \in \partial H(u^k)$ and $s^k \in \partial J(p^k)$. Therefore, the errors for t^k and s^k can be obtained as $t_e^k = t^k - t^*$ and $s_e^k = s^k - s^*$ where $t^* \in \partial H(u^*)$ and

$s^* \in \partial J(p^*)$.

Using the optimality condition, we have the following sequence $(u^{k+1}, p^{k+1}; \lambda^{k+2})$ generated by Algorithm 9 in terms of errors satisfying

$$\begin{cases} t_e^{k+1} &= rA^T(p_e^k - Au_e^k - \frac{1}{r}\lambda_e^{k+1}), \\ s_e^{k+1} &= r(Au_e^{k+1} + \frac{1}{r}\lambda_e^{k+1} - rp_e^{k+1}), \\ \lambda_e^{k+2} &= \lambda_e^{k+1} + r(Au_e^{k+1} - p_e^{k+1}). \end{cases} \quad (6.14)$$

By taking the inner products of the first two equalities in (6.14) with u_e^{k+1} and p_e^{k+1} , we have the following equations

$$\begin{aligned} \langle t_e^{k+1}, u_e^{k+1} \rangle &= r \langle Au_e^{k+1}, p_e^k - Au_e^k - \frac{1}{r}\lambda_e^{k+1} \rangle, \\ \langle s_e^{k+1}, p_e^{k+1} \rangle &= r \langle p_e^{k+1}, Au_e^{k+1} + \frac{1}{r}\lambda_e^{k+1} - p_e^{k+1} \rangle, \end{aligned}$$

Similarly, by taking the inner product of the last equality in (6.14) with λ_e^{k+1} and using the relation (6.10), we obtain the following equation

$$\frac{1}{2r}(\|\lambda_e^{k+2}\|_2^2 - \|\lambda_e^{k+1}\|_2^2) - \frac{r}{2}\|Au_e^{k+1} - p_e^{k+1}\|_2^2 = \langle \lambda_e^{k+1}, Au_e^{k+1} - p_e^{k+1} \rangle.$$

Summing over these three equalities, we obtain

$$\begin{aligned} &\langle t_e^{k+1}, u_e^{k+1} \rangle + \langle s_e^{k+1}, p_e^{k+1} \rangle + \frac{1}{2r}\|\lambda_e^{k+2}\|_2^2 - \frac{1}{2r}\|\lambda_e^{k+1}\|_2^2 \\ &= r \langle Au_e^{k+1}, p_e^k \rangle - r \langle Au_e^{k+1}, Au_e^k \rangle + r \langle p_e^{k+1}, Au_e^{k+1} \rangle - r \langle p_e^{k+1}, p_e^{k+1} \rangle \\ &\quad + \frac{r}{2}\|Au_e^{k+1} - p_e^{k+1}\|_2^2 \\ &= -\frac{r}{2}\|Au_e^{k+1} - p_e^k\|_2^2 - \frac{r}{2}\|Au_e^{k+1} + Au_e^k\|_2^2 + \frac{3r}{2}\|Au_e^{k+1}\|_2^2 + \frac{r}{2}\|Au_e^k\|_2^2 \\ &\quad - \frac{r}{2}\|p_e^{k+1}\|_2^2 + \frac{r}{2}\|p_e^k\|_2^2. \end{aligned} \quad (6.15)$$

By the convexity of the functional $J(\cdot)$, we have

$$\langle s_e^{k+1}, p_e^{k+1} \rangle \geq 0. \quad (6.16)$$

Since $H(\cdot)$ is strongly convex with modulus o , we have

$$\langle t_e^{k+1}, u_e^{k+1} \rangle \geq o \|u_e^{k+1}\|_2^2. \quad (6.17)$$

By taking the above relationships of s_e^{k+1} (6.16) and t_e^{k+1} (6.17) back to the equality (6.15), we obtain the following inequality

$$\begin{aligned} & \langle (oI - 2rA^T A)u_e^{k+1}, u_e^{k+1} \rangle + \frac{r}{2} \|Au_e^{k+1}\|_2^2 + \frac{r}{2} \|p_e^{k+1}\|_2^2 + \frac{1}{2r} \|\lambda_e^{k+2}\|_2^2 \\ & \leq \frac{r}{2} \|Au_e^k\|_2^2 + \frac{r}{2} \|p_e^k\|_2^2 + \frac{1}{2r} \|\lambda_e^{k+1}\|_2^2. \end{aligned}$$

Let $L_1 = oI - 2rA^T A$. The above inequality becomes

$$\|u^{k+1} - u^*\|_{L_1}^2 + \frac{r}{2} \|Au_e^{k+1}\|_2^2 + \frac{r}{2} \|p_e^{k+1}\|_2^2 + \frac{1}{2r} \|\lambda_e^{k+2}\|_2^2 \leq \frac{r}{2} \|Au_e^k\|_2^2 + \frac{r}{2} \|p_e^k\|_2^2 + \frac{1}{2r} \|\lambda_e^{k+1}\|_2^2. \quad (6.18)$$

As long as L_1 is positive definite, the sequence $(u^k, p^k; \lambda^k)$ is bounded. Summing (6.18) over k from 0 to ∞ , we can derive

$$\sum_{k=0}^{\infty} (\|u^{k+1} - u^*\|_{L_1}^2) < \infty.$$

Thus we have

$$\lim_{k \rightarrow \infty} (u^k - u^*) = 0,$$

which completes the proof. ■

Theorem 6.2. *Let the sequence $(u^k, p^k; \lambda^k)$ be generated by Algorithm 10. Assume $L_2 = 2oI - rA^T A$ is positive definite, then*

- $\lim_{k \rightarrow \infty} u^k = u^*.$
- $\lim_{k \rightarrow \infty} p^k = Au^*.$

Proof. The proof is similar to the above one. We use the same kind of notations for $(u_e^k, p_e^k; \lambda_e^k)$ and $t_e^k := t^k - t^*$, $s_e^k := s^k - s^*$, where $t^k \in \partial H(u^k)$, $s^k \in \partial J(p^k)$, $t^* \in \partial H(u^*)$ and $s^* \in \partial J(p^*)$.

We have the following sequence $(u^{k+1}, p^{k+1}; \lambda^{k+2})$ generated by Algorithm 10 in terms of errors satisfying

$$\begin{cases} t_e^{k+1} &= rA^T(p_e^k - Au_e^k - \frac{1}{r}\lambda_e^k), \\ s_e^{k+1} + rp_e^{k+1} &= r(Au_e^{k+1} + \frac{1}{r}\lambda_e^{k+1}), \\ \lambda_e^{k+2} &= \lambda_e^{k+1} + r(Au_e^{k+1} - p_e^{k+1}). \end{cases} \quad (6.19)$$

By taking the inner product of the above equalities with u_e^{k+1} , p_e^{k+1} and λ_e^{k+1} respectively, we then obtain

$$\begin{aligned} \langle t_e^{k+1}, u_e^{k+1} \rangle &= r \langle Au_e^{k+1}, p_e^k - Au_e^k - \frac{1}{r}\lambda_e^k \rangle, \\ \langle s_e^{k+1}, p_e^{k+1} \rangle + r \|p_e^{k+1}\|_2^2 &= r \langle p_e^{k+1}, Au_e^{k+1} + \frac{1}{r}\lambda_e^{k+1} \rangle, \\ \frac{1}{2r} (\|\lambda_e^{k+2}\|_2^2 - \|\lambda_e^{k+1}\|_2^2) - \frac{r}{2} \|Au_e^{k+1} - p_e^{k+1}\|_2^2 &= \langle \lambda_e^{k+1}, Au_e^{k+1} - p_e^{k+1} \rangle. \end{aligned}$$

Summing over these three equalities, we have

$$\begin{aligned} &\langle t_e^{k+1}, u_e^{k+1} \rangle + \langle s_e^{k+1}, p_e^{k+1} \rangle + r \|p_e^{k+1}\|_2^2 + \frac{1}{2r} \|\lambda_e^{k+2}\|_2^2 \\ &= r \langle Au_e^{k+1}, p_e^k \rangle - r \langle Au_e^{k+1}, Au_e^k \rangle - \langle Au_e^{k+1}, \lambda_e^k \rangle + r \langle Au_e^{k+1}, p_e^{k+1} \rangle \\ &\quad + \frac{r}{2} \|Au_e^{k+1} - p_e^{k+1}\|_2^2 + \langle \lambda_e^{k+1}, Au_e^{k+1} \rangle + \frac{1}{2r} \|\lambda_e^{k+1}\|_2^2 \\ &= \frac{r}{2} \|Au_e^{k+1}\|_2^2 + \frac{r}{2} \|p_e^{k+1}\|_2^2 + r \langle Au_e^{k+1}, p_e^k \rangle - r \langle Au_e^{k+1}, Au_e^k \rangle \\ &\quad + \langle Au_e^{k+1}, \lambda_e^{k+1} - \lambda_e^k \rangle + \frac{1}{2r} \|\lambda_e^{k+1}\|_2^2 \\ &= \frac{r}{2} \|Au_e^{k+1}\|_2^2 + \frac{r}{2} \|p_e^{k+1}\|_2^2 + \frac{1}{2r} \|\lambda_e^{k+1}\|_2^2. \end{aligned}$$

Therefore, we obtain the following inequality by taking the relationships (6.16) and (6.17) to the above equality

$$\langle (2oI - rA^T A)u_e^{k+1}, u_e^{k+1} \rangle + r \|p_e^{k+1}\|_2^2 + \frac{1}{r} \|\lambda_e^{k+2}\|_2^2 \leq \frac{1}{r} \|\lambda_e^{k+1}\|_2^2.$$

Let $L_2 = 2oI - rA^T A$. We deduce that

$$\|u^{k+1} - u^*\|_{L_2}^2 + r\|p^{k+1} - Au^*\|_2^2 + \frac{1}{r}\|\lambda^{k+2} - \lambda^*\|_2^2 \leq \frac{1}{r}\|\lambda^{k+1} - \lambda^*\|_2^2. \quad (6.20)$$

Hence, $\lim_{k \rightarrow \infty} \|\lambda^k - \lambda^*\|_2^2$ exists. Furthermore, it follows from (6.20) that

$$\|u^{k+1} - u^*\|_{L_2}^2 + r\|p^{k+1} - Au^*\|_2^2 \leq \frac{1}{r}\|\lambda^{k+1} - \lambda^*\|_2^2 - \frac{1}{r}\|\lambda^{k+2} - \lambda^*\|_2^2. \quad (6.21)$$

Summing (6.21) from 0 to ∞ , we have

$$\sum_{k=0}^{\infty} (\|u^{k+1} - u^*\|_{L_2}^2) + \sum_{k=0}^{\infty} r\|p^{k+1} - Au^*\|_2^2 < \infty.$$

Since L_2 is positive definite, we conclude that

$$\begin{aligned} \lim_{k \rightarrow \infty} (u^k - u^*) &= 0, \\ \lim_{k \rightarrow \infty} (p^k - Au^*) &= 0, \end{aligned}$$

which complete the proof. ■

6.4 First-order Augmented Lagrangian Algorithms

In this section, we construct two kinds of first-order augmented Lagrangian schemes. We show that Algorithm 9 and Algorithm 10 are two special cases of the proposed first-order schemes. Furthermore, we establish a connection between the first-order augmented Lagrangian algorithms and the first-order primal-dual algorithm in [5].

In [5], authors define the primal-dual formulation for (6.1) as follows

$$\min_u \max_{\lambda} \left\{ \langle \lambda, Au \rangle - J^*(\lambda) + H(u) \right\}, \quad (6.22)$$

where J^* is the convex conjugate of J and defined by

$$J^*(p) = \sup_p \left\{ \langle \lambda, p \rangle - J(p) \right\}.$$

Let (u^*, λ^*) be a solution of the saddle-point system (6.22). Then (u^*, λ^*) has to satisfy the following optimality conditions

$$\begin{aligned} -A^T \lambda^* &\in \partial H(u^*), \\ Au^* &\in \partial J^*(\lambda^*). \end{aligned} \tag{6.23}$$

For the augmented Lagrangian formulation (6.3) of the general convex problem (6.1), we have the optimality conditions for the saddle-point $(u^*, p^*; \lambda^*)$

$$\begin{aligned} -A^T(\lambda^* + r(Au^* - p^*)) &\in \partial H(u^*), \\ \lambda^* + r(Au^* - p^*) &\in \partial J(p^*). \end{aligned}$$

Since there is $Au^* = p^*$, we have

$$\begin{aligned} -A^T \lambda^* &\in \partial H(u^*), \\ \lambda^* &\in \partial J(Au^*). \end{aligned} \tag{6.24}$$

Therefore, (6.3) is equivalent to the primal-dual formulation (6.22). We refer to [5] for more information. We assume that J and H are “simple”, in the sense that their resolvent operator defined through

$$x = (I + \tau \partial H)^{-1}(y) = \arg \min_x \left\{ \frac{\|x - y\|^2}{2\tau} + H(x) \right\},$$

where τ is a parameter, has a closed-form representation (or can be efficiently solved up to a high precision, e.g. using a Newton method in low dimension).

In [5], the authors used the resolvent operator and proposed the following scheme for (6.22)

$$\begin{cases} \lambda^{k+1} = (I + r \partial J^*)^{-1}(\lambda^k + r A \bar{u}^k), \\ u^{k+1} = (I + \tau \partial H)^{-1}(u^k - \tau A^T \lambda^{k+1}), \\ \bar{u}^{k+1} = u^{k+1} + \theta(u^{k+1} - u^k), \end{cases} \tag{6.25}$$

where $\theta \in [0, 1]$ is a parameter.

Similarly, if we adopt the same idea to both the u -sub problem (6.8) and w -sub

problem (6.9), we can construct a novel first-order scheme for (6.1) based on its augmented Lagrangian formulation. For the u -sub problem, we have

$$\begin{aligned} u^{k+1} &= (I + \tau \partial H)^{-1}(u^k - \tau(A^T \lambda^{k+1} + rA^T(Au^k - p^k))) \\ &= (I + \tau \partial H)^{-1}(u^k - \tau(A^T(2\lambda^{k+1} - \lambda^k))), \end{aligned} \quad (6.26)$$

and for the p -sub problem, we have

$$p^{k+1} = (I + \delta \partial J)^{-1}(w^k + \delta(\lambda^{k+1} + rAu^{k+1})),$$

where δ is a parameter.

Therefore, we can express the corresponding first-order augmented Lagrangian algorithm for (6.3) in the following way

$$\begin{cases} \lambda^{k+1} = \lambda^k + r(Au^k - p^k), \\ \bar{\lambda}^{k+1} = 2\lambda^{k+1} - \lambda^k, \\ u^{k+1} = (I + \tau \partial H)^{-1}(u^k - \tau A^T \bar{\lambda}^{k+1}), \\ p^{k+1} = (I + \delta \partial J)^{-1}(p^k + \delta(\lambda^{k+1} + rAu^{k+1})). \end{cases} \quad (6.27)$$

If $\tau = 1$ and $\delta = \frac{1}{r}$ in the above scheme, we see that it becomes the proposed Fixed-point Augmented Lagrangian Algorithm 9.

Furthermore, if we replace λ^{k+1} by λ^k in the u -sub problem (6.26), we have

$$\begin{aligned} u^{k+1} &= (I + \tau \partial H)^{-1}(u^k - \tau(A^T \lambda^k + rA^T(p^k - Au^k))) \\ &= (I + \tau \partial H)^{-1}(u^k - \tau A^T \lambda^{k+1}). \end{aligned}$$

Thus, the first-order augmented Lagrangian scheme (6.27) can be modified as follows

$$\begin{cases} \lambda^{k+1} = \lambda^k + r(Au^k - p^k), \\ u^{k+1} = (I + \tau \partial H)^{-1}(u^k - \tau A^T \lambda^{k+1}), \\ p^{k+1} = (I + \delta \partial J)^{-1}(p^k + \delta(\lambda^{k+1} + rAu^{k+1})). \end{cases} \quad (6.28)$$

If $\tau = 1$ and $\delta = \frac{1}{r}$ in the above scheme, it is equivalent to the Alternating Minimization Algorithm 10.

Furthermore, we eliminate p in (6.28) by the Moreau decomposition in [123, 118, 124], which means

$$\begin{aligned}\lambda^{k+1} &= \lambda^k + rAu^k - r \arg \min_p \left\{ J(p) - \langle \lambda^k, p \rangle + \frac{r}{2} \|p - Au^k\|_2^2 \right\} \\ &= r \left(\frac{\lambda^k + rAu^k}{r} - \arg \min_p \left\{ J(p) + \frac{r}{2} \left\| p - \frac{\lambda^k + rAu^k}{r} \right\|_2^2 \right\} \right) \\ &= \arg \min_{\lambda} \left\{ J^*(\lambda) + \frac{1}{2r} \|\lambda - (\lambda^k + rAu^k)\|_2^2 \right\}.\end{aligned}$$

Together with the u -sub problem, we get the following primal-dual kind of scheme

$$\begin{cases} \lambda^{k+1} = (I + r\partial J^*)^{-1}(\lambda^k + rAu^k), \\ u^{k+1} = (I + \tau\partial H)^{-1}(u^k - \tau A^T \lambda^{k+1}). \end{cases}$$

This scheme, corresponding to the case $\theta = 0$ in the first-order primal-dual algorithm (6.25) is the semi-implicit classical Arrow-Hurwicz algorithm [77].

6.5 Acceleration of the Fixed-point Algorithm

In [125], a gradient method with an $O(1/k^2)$ complexity was proposed for solving the convex programming problems. The idea was recently applied to the shrinkage-thresholding algorithm in [126]. Encouraged by the same idea, we compute the Lagrange multiplier λ^{k+1} by a specific linear combination of $\{\lambda^k, \lambda^{k-1}\}$ rather than update it by λ^k . The specific approach to compute λ^{k+1} can be described as follows

Algorithm 12 Acceleration on Lagrange multiplier

1. Compute q^{k+1} from: $q^{k+1} = \lambda^k + r(Au^k - p^k)$.
 2. Compute t_{k+1} by: $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$.
 3. Update λ^{k+1} from: $\lambda^{k+1} = q^{k+1} + \frac{t_k - 1}{t_{k+1}}(q^{k+1} - q^k)$.
-

The acceleration technique will be applied to both Algorithm 9 and Algorithm 10 in this section. We will demonstrate that Algorithm 9 and Algorithm 10 with acceleration Algorithm 12 provide us algorithms with better convergence rates in the next section.

6.6 Applications

In this section, we consider the applications of the proposed algorithms to image denoising models, including the ROF model, the vectorial TV model, a high order model and the TV- L^1 model. We will examine the efficiency of Algorithm 9, Algorithm 10 and the fast fixed-point augmented Lagrangian algorithm (FFPALM) with the Lagrange multiplier updated by Algorithm 12. We mainly use the time trials and SNR for the comparisons. All the algorithms are implemented in MATLAB environment on a 2.4 GHz CPU and 4GB memory.

6.6.1 Notations and preliminary remarks

We give some notations at the beginning. Without loss of generality, we represent a grey image as an $N \times N$ matrix, named as

$$u : \{1, \dots, N\} \times \{1, \dots, N\},$$

where $N \geq 2$. We define two inner product vector spaces

$$V = \mathbb{R}^{N \times N} \quad \text{and} \quad Q = V \times V.$$

We denote the usual inner product and Euclidean norm of V as $(\cdot, \cdot)_V$ and $\|\cdot\|_V$, respectively. We also equip the space Q with inner product $(\cdot, \cdot)_Q$ and norm $\|\cdot\|_Q$, which are defined as follows. For $p = (p_1, p_2) \in Q$ and $q = (q_1, q_2) \in Q$,

$$\langle p, q \rangle_Q = \langle p_1, q_1 \rangle_V + \langle p_2, q_2 \rangle_V \quad \text{and} \quad \|p\|_Q = \sqrt{\langle p, p \rangle_Q}.$$

The discrete backward and forward differential operators for $u \in V$ are defined with the periodic boundary condition as follows

$$\partial_x^- u(i, j) = \begin{cases} u(i, j) - u(i-1, j) & 1 < i \leq N, \\ u(1, j) - u(N, j) & i = 1. \end{cases}$$

$$\partial_y^- u(i, j) = \begin{cases} u(i, j) - u(i, j-1) & 1 < j \leq N, \\ u(i, 1) - u(i, N) & j = 1. \end{cases}$$

$$\partial_x^+ u(i, j) = \begin{cases} u(i+1, j) - u(i, j) & 1 \leq i < N, \\ u(1, j) - u(N, j) & i = N. \end{cases}$$

$$\partial_y^+ u(i, j) = \begin{cases} u(i, j+1) - u(i, j) & 1 \leq j < N, \\ u(i, 1) - u(i, N) & j = N. \end{cases}$$

Based on the forward differential operators, we give the gradient operator as a mapping $\nabla : V \rightarrow Q$. For $u \in V$, the discrete gradient operator ∇u is defined as follows

$$\nabla u(i, j) := (\partial_x^+ u(i, j), \partial_y^+ u(i, j)).$$

The corresponding discrete backward adjoint operator $\text{div} : Q \rightarrow V$ is defined for $p = (p_1, p_2) \in Q$ as

$$\text{div} p(i, j) := \partial_x^- p_1(i, j) + \partial_y^- p_2(i, j).$$

We also define the Laplace operator $\Delta : V \rightarrow V$ for $u \in V$ by the following mapping

$$\Delta u(i, j) := \partial_x^- \partial_x^+ u(i, j) + \partial_y^- \partial_y^+ u(i, j).$$

When studying high order models, we need to introduce second order difference operators. We denote the discrete Hessian of $u \in V$ as

$$Hu = \begin{pmatrix} \partial_x^- \partial_x^+ u & \partial_x^+ \partial_y^+ u \\ \partial_y^+ \partial_x^+ u & \partial_y^- \partial_y^+ u \end{pmatrix} \in Q_2,$$

with

$$Hu(i, j) = \begin{pmatrix} \partial_x^- \partial_x^+ u(i, j) & \partial_x^+ \partial_y^+ u(i, j) \\ \partial_y^+ \partial_x^+ u(i, j) & \partial_y^- \partial_y^+ u(i, j) \end{pmatrix},$$

where

$$Q_2 = V \times V \times V \times V.$$

By regarding the Hessian as an operator $H: V \rightarrow Q_2$, we find its adjoint operator $H^*: Q_2 \rightarrow V$ as

$$H^*p = \partial_x^+ \partial_x^- p_{11} + \partial_y^- \partial_x^- p_{12} + \partial_x^- \partial_y^- p_{21} + \partial_y^+ \partial_y^- p_{22},$$

for

$$p = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix} \in Q_2.$$

6.6.2 The ROF model

I. The algorithms

One of the most successful applications of the general convex problem (6.1) with the quadratic fidelity term is the ROF model. We can get the ROF model by assuming $A = \nabla$. Therefore, we are interested in solving the following constrained problem

$$\begin{aligned} \min_{u, p} \quad & \left\{ \|p\|_1 + \frac{\mu}{2} \|u - f\|_2^2 \right\}, \\ \text{s.t.,} \quad & p = \nabla u. \end{aligned} \tag{6.29}$$

We define the augmented Lagrangian functional of the constrained optimization problem (6.29) as follows

$$\mathcal{L}_{ROF}(u, p; \lambda) = \|p\|_1 + \frac{\mu}{2} \|u - f\|_2^2 + \langle \lambda, \nabla u - p \rangle + \frac{r}{2} \|\nabla u - p\|_2^2. \tag{6.30}$$

To find the saddle-point of (6.30), we separate the variable u and p and then

use an alternative minimization procedure. We split (6.30) into the following two sub-problems:

- The u -sub problem: for a given p

$$\min_u \left\{ \frac{\mu}{2} \|u - f\|_2^2 + \langle \lambda, \nabla u \rangle + \frac{r}{2} \|\nabla u - p\|_2^2 \right\}. \quad (6.31)$$

- The p -sub problem: for a given u

$$\min_p \left\{ \|p\|_1 - \langle \lambda, p \rangle + \frac{r}{2} \|\nabla u - p\|_2^2 \right\}. \quad (6.32)$$

We can apply either the fixed-point augmented Lagrangian Algorithm 9 or the alternating minimization Algorithm 10 to the saddle-point system (6.30). The main difference between these two approaches lies in the u -sub problem while we solve the p -sub problem in the exactly same way. For the p -sub problem, we explicitly compute the optimal value of p using the point-wise shrinkage operator. In the $(k + 1)^{th}$ iteration, we simply compute p^{k+1} from

$$p^{k+1} = \text{shrink}(\nabla u^{k+1} + \frac{\lambda^{k+1}}{r}, \frac{1}{r}),$$

with

$$\text{shrink}(q, 1/r) := \begin{cases} q - 1/r, & \text{for } q > 1/r, \\ 0, & \text{for } -1/r \leq q \leq 1/r, \\ q + 1/r, & \text{for } q < -1/r. \end{cases}$$

where

$$q = \nabla u^{k+1} + \frac{\lambda^{k+1}}{r}.$$

For the u -sub problem, if we apply Algorithm 9, the solution of the u -sub problem is

$$u^{k+1} = f + \frac{1}{\mu} \text{div} \lambda^{k+1} + \frac{r}{\mu} (\Delta u^k - \text{div} p^k).$$

We can show that the above scheme is convergent when μ and r satisfy that

$\mu I + 2r\Delta$ is positive definite. We find that $\mu I + 2r\Delta$ is a real symmetric linear operator on $\mathbb{R}^{N \times N}$. Suppose $\eta \in \mathbb{R}$ is an eigenvalue of the operator $\mu I + 2r\Delta$. We can show $\eta > 0$, provided $0 < r/\mu < 1/16$. Then we have $\mu I + 2r\Delta$ is positive definite as long as $0 < r/\mu < 1/16$.

On the other hand, if we use Algorithm 10, the solution of the u -sub problem becomes

$$u^{k+1} = f + \frac{1}{\mu} \operatorname{div} \lambda^{k+1}.$$

In this case, the scheme can be shown to be convergent when μ and r satisfy that $2\mu I + r\Delta$ is positive definite. Thus, there exists $0 < r/\mu < 1/4$.

II. Numerical examples

In the following, we demonstrate the numerical performance of the proposed Algorithm 9 and Algorithm 10. For all of the tested algorithms, we used the following stopping condition: Given a tolerance TOL , the iteration runs until

$$\frac{\sum_{i,j} (u^k(i,j) - u^{k-1}(i,j))^2}{\sum_{i,j} (u^{k-1}(i,j))^2} \leq TOL. \quad (6.33)$$

We also define the numerical energy of (6.29) by

$$E(u^k) = \sum_{i,j} |\nabla u^k(i,j)| + \frac{r}{2} \sum_{i,j} (u^k(i,j) - f(i,j))^2. \quad (6.34)$$

In addition, we use the signal-to-noise ratio (SNR) to estimate the difference between restored image u and original image u_c for image denoising applications

$$SNR(u^k, u_c) = 10 \log_{10} \left(\frac{\sum_{i,j} (u^k(i,j) - a_1)^2}{\sum_{i,j} (|u^k(i,j) - u_c(i,j)| - a_2)^2} \right), \quad (6.35)$$

where a_1 and a_2 are the average of u^k and $u^k - u_c$, respectively.

We choose one grey image: Lena (512×512) to test the denoising capacity of the

Table 6.1: The computation summary of the ROF model on Lena.

	$\sigma=25$			$\sigma=35$		
	CPU (s)	ITER	SNR (dB)	CPU (s)	ITER	SNR (dB)
FPALM	2.76	36	14.87	4.11	57	12.87
AMA	0.71	13	14.75	2.40	41	12.77
FFPALM	2.40	30	14.79	2.31	31	12.68
FAMA	1.46	21	14.78	1.82	30	12.63
ALM	2.81	15	14.78	2.45	13	12.69

ROF model using different noise level of Gaussian noise with mean zero and variance $\sigma = 25$ and $\sigma = 35$. We implement Algorithm 9, Algorithm 10 and Algorithm 11 and set the error tolerance be $TOL = 10^{-5}$ for all the algorithms. For performance evaluation, we use the following parameters settings

1. FPALM: Fixed-point augmented Lagrangian Algorithm 9 with $r = 0.004$, $\mu = 0.05$ for $\sigma = 25$ and $r = 0.002$, $\mu = 0.025$ for $\sigma = 35$.
2. AMA: Alternating minimization Algorithm 10 with $r = 0.01$, $\mu = 0.05$ for $\sigma = 25$ and $r = 0.006$, $\mu = 0.025$ for $\sigma = 35$.
3. FFPALM: Fast fixed-point augmented Lagrangian algorithm with $r = 0.003$, $\mu = 0.05$ for $\sigma = 25$ and $r = 0.001$, $\mu = 0.025$ for $\sigma = 35$.
4. FAMA: Fast alternating minimization algorithm with $r = 0.008$, $\mu = 0.05$ for $\sigma = 25$ and $r = 0.004$, $\mu = 0.025$ for $\sigma = 35$.
5. ALM: Augmented Lagrangian Algorithm 11 with FFT applied for solving the u -sub problem as in Wu and Tai framework [17] with $r = 0.1$, $\mu = 0.05$ for $\sigma = 25$ and $r = 0.05$, $\mu = 0.025$ for $\sigma = 35$.

We list the computational time, iteration number and SNR from each algorithm of Lena in Table 6.1. The selected visual results of Lena with variance $\sigma = 35$ are listed in Figure 6.1.

We further compare the performance among Algorithm 9, Algorithm 10 and Algorithm 11 and all the comparison results given are from Lena with variance $\sigma = 35$.

Figure 6.1: The ROF model of Lena. Row one from left to right: noise image, recovered image by ALM and FPALM. Row two from left to right: recovered image by AMA, FFPALM and FAMA.

We plot the figures of the SNR and numerical energy of the FFPALM, FAMA and ALM in Figure 6.2 (a) and (b). We also estimate the relative error (6.33) of the three algorithms and plot the results in Figure 6.2 (c).

To demonstrate the efficiency of the acceleration technique to the overall performance of both FPALM and AMA, we track the root mean squared error in each algorithm. We first run a well performing method for a very long time (≈ 100000 iterations) in order to compute a “ground truth” solution. Then, we apply each algorithm to Lena ($\sigma = 35$) and compute the error to the pre-determined ground truth solution within 10000 iterations. The plots of RMSE are displayed in Figure 6.3. One can see that the acceleration technique can improve the proposed algorithms to obtain better convergence rates.

6.6.3 The vectorial TV model

I. The algorithms

In [22], Blomgren and Chan defined the vectorial total variation (VTV) norm. Based on the VTV norm, the vectorial TV model was further studied in [127, 23]. Recently,

Figure 6.2: Plots of SNR, numerical energy (6.34) and relative error (6.33) of Lena with $\sigma = 35$.

Figure 6.3: Plots of convergence for Algorithm 9 (FPALM) and Algorithm 10 (AMA) of Lena with $\sigma = 35$.

a dual formulation of the vectorial ROF model was proposed in [24] by extending Chambolle's dual method for the ROF model in [12].

Let us denote in general an M -channel image by $\mathbf{u} = (u_1, u_2, \dots, u_M)$, where $u_m \in V$, $\forall m = 1, 2, \dots, M$. The intensity at pixel (i, j) is multi-valued, $\mathbf{u}(i, j) = (u_1(i, j), u_2(i, j), \dots, u_M(i, j))$. If $M = 3$, it is usual color models such as RGB.

For convenience of description, we introduce the following notations

$$\mathbf{V} = V \times V \times \dots \times V,$$

$$\mathbf{Q} = Q \times Q \times \dots \times Q.$$

Hence, the gradient of $\mathbf{u}$ is $\nabla \mathbf{u} = (\nabla u_1, \nabla u_2, \dots, \nabla u_M)$.

We consider the following vector-valued image restoration problem

$$\min_{\mathbf{u}} \left\{ \|\nabla \mathbf{u}\|_1 + \frac{\mu}{2} \|\mathbf{u} - \mathbf{f}\|_{\mathbf{V}}^2 \right\}. \quad (6.36)$$

By introducing a new variable $\mathbf{p} = (p_1, p_2, \dots, p_M)$ to (6.36), the minimization problem (6.36) is reformulated into the following constrained minimization problem

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{p}} \left\{ \|\mathbf{p}\|_1 + \frac{\mu}{2} \|\mathbf{u} - \mathbf{f}\|_{\mathbf{V}}^2 \right\}, \\ \text{s.t., } \mathbf{p} = \nabla \mathbf{u}. \end{aligned} \quad (6.37)$$

Therefore, the corresponding augmented Lagrangian functional of (6.37) is defined as

$$\mathcal{L}_{VTV}(\mathbf{p}, \mathbf{u}; \lambda) = \|\mathbf{p}\|_1 + \frac{\mu}{2} \|\mathbf{u} - \mathbf{f}\|_{\mathbf{V}}^2 + \langle \lambda, \nabla \mathbf{u} - \mathbf{p} \rangle + \frac{\mathbf{r}}{2} \|\mathbf{p} - \nabla \mathbf{u}\|_{\mathbf{V}}^2. \quad (6.38)$$

Similar to the ROF model, we use the operator splitting method to solve the saddle-point problem. We split the augmented Lagrangian functional (6.38) into two sub-problems regarding to the variable $\mathbf{p}$ and $\mathbf{u}$. There is the closed form solution for the $\mathbf{p}$ -sub problem. For the $\mathbf{u}$ -sub problem, if we use Algorithm 9, we solve the

$\mathbf{u}$ -sub problem in the $(k+1)^{th}$ iteration from

$$\mathbf{u}^{k+1} = \mathbf{f} + \frac{1}{\mu} \operatorname{div} \boldsymbol{\lambda}^{k+1} + \frac{r}{\mu} (\Delta \mathbf{u}^k - \operatorname{div} \mathbf{p}^k).$$

On the other hand, if we apply Algorithm 10, we have the solution of $\mathbf{u}^{k+1}$ as

$$\mathbf{u}^{k+1} = \mathbf{f} + \frac{1}{\mu} \operatorname{div} \boldsymbol{\lambda}^{k+1}.$$

II. Numerical examples

In the following, we provide the numerical results of the proposed algorithms on two color images: Lena(512×512) and Rose(250×303) added by the Gaussian noise with mean zero and variance $\sigma = 25$. For both images, we set the stopping condition as $TOL = 10^{-4}$ for all the test algorithms. We implement the following algorithms and parameters setting for the comparison

1. FPALM: Fixed-point augmented Lagrangian Algorithm 9 with $r = 0.0035$ and $\mu = 0.05$.
2. AMA: Alternating minimization Algorithm 10 with $r = 0.01$ and $\mu = 0.05$.
3. FFPALM: Fast fixed-point augmented Lagrangian algorithm with $r = 0.003$ and $\mu = 0.05$.
4. FAMA: Fast alternating minimization with $r = 0.008$ and $\mu = 0.05$.
5. ALM: Augmented Lagrangian Algorithm 3 which is proposed in [17] with $r = 0.1$ and $\mu = 0.05$.

We tabulate the computational time, iteration number and SNR of all algorithms in Table 6.2. Besides, we display the numerical results of Rose in Figure 6.4.

6.6.4 The high order model

I. The algorithms

Table 6.2: The computation summary for the experiment of the vectorial TV model.

	Lena			Rose		
	CPU(s)	ITER	SNR(dB)	CPU(s)	ITER	SNR(dB)
FPALM	3.25	14	16.20	1.06	14	16.61
AMA	1.37	8	16.59	0.52	9	16.51
FFPALM	3.17	13	16.45	1.14	15	16.33
FAMA	1.93	10	16.56	0.75	11	16.27
ALM	7.81	10	16.27	2.27	10	16.35

We know that the ROF model tends to converge to a piecewise constant image, which will cause the staircase effect to the results; see [128, 32]. High order models can overcome this problem in [10, 32, 34, 129]. Here comes another well-known application of the general convex optimization problem (6.1), the 4th-order model proposed in [34]. One can get the high order model from (6.1) by assuming $A = H$, where H denotes the Hessian operator together with the quadratic fidelity term. Therefore, we consider the following image restoration problem

$$\min_u \left\{ \|Hu\|_1 + \frac{\mu}{2} \|u - f\|_2^2 \right\}. \quad (6.39)$$

We can reformulate (6.39) into a constrained optimization problem by introducing a new variable $p = Hu$ as follows

$$\begin{aligned} \min_{u,p} \quad & \left\{ \|p\|_1 + \frac{\mu}{2} \|u - f\|_2^2 \right\}, \\ \text{s.t.}, \quad & p = Hu. \end{aligned} \quad (6.40)$$

To solve the constrained optimization problem, we use the augmented Lagrangian method and define the augmented Lagrangian functional for (6.40) as follows

$$\mathcal{L}_{Ht}(u, p; \lambda) = \|p\|_1 + \frac{\mu}{2} \|u - f\|_2^2 + \langle \lambda, Hu - p \rangle + \frac{r}{2} \|Hu - p\|_2^2. \quad (6.41)$$

We separate the variable u and p in the saddle-point problem (6.41). Therefore, two sub-problems emerge as follows:

Figure 6.4: The vectorial TV model of Rose. Row one from left to right: noise image, recovered image by ALM and FPALM. Row two from left to right: recovered image by AMA, FFPALM and FAMA.

- The u -sub problem: for a given p

$$\min_u \left\{ \frac{\mu}{2} \|u - f\|_2^2 + \langle \lambda, Hu \rangle + \frac{r}{2} \|Hu - p\|_2^2 \right\}. \quad (6.42)$$

- The p -sub problem: for a given u

$$\min_p \left\{ \|p\|_1 - \langle \lambda, p \rangle + \frac{r}{2} \|Hu - p\|_2^2 \right\}. \quad (6.43)$$

We can use both the fixed-point augmented Lagrangian Algorithm 9 and the alternating minimization Algorithm 10 to solve the alternative minimization system. For the p -sub problem, we explicitly compute the optimal value of p in the $(k+1)^{th}$ iteration by the shrinkage operation, which is

$$p^{k+1} = \text{shrink}\left(Hu^{k+1} + \frac{\lambda^{k+1}}{r}, \frac{1}{r}\right).$$

If we apply Algorithm 9 to (6.42), we solve the solution of the u -sub problem in the $(k + 1)^{th}$ iteration by

$$u^{k+1} = f - \frac{1}{\mu} H^* \lambda^{k+1} - \frac{r}{\mu} H^* (Hu^k - p^k).$$

For this case, the above scheme can be proved convergent as long as μ and r satisfying $\mu I + 2rH^*H$ is positive definite. Therefore, we have $0 < r/\mu < 1/128$.

If we apply Algorithm 10 to (6.42), we obtain the solution u^{k+1} by

$$u^{k+1} = f - \frac{1}{\mu} H^* \lambda^{k+1}.$$

For this case, the above scheme can be shown to be convergent as long as μ and r satisfying $2\mu I + rH^*H$ is positive definite. Thus, there exists $0 < r/\mu < 1/32$.

II. Numerical examples

In the following, we develop the numerical results of the high order model (6.39) with the proposed algorithms. We choose the image Tomato (512×512) as an example and add the Gaussian noise with mean zero and variance $\sigma = 10$ and $\sigma = 25$. We choose different stopping criterions for the test algorithms to derive the results with similar visual quality. We apply the acceleration Algorithm 12 to both Algorithm 9 and Algorithm 10 in the experiment to reduce the numerical expense. We implement the following algorithms and parameters setting for the comparison

1. FFPALM: Fast fixed-point augmented Lagrangian algorithm with $r = 0.0009$, $\mu = 0.05$ and $TOL = 10^{-7}$.
2. FAMA: Fast alternating minimization algorithm with $r = 0.001$, $\mu = 0.05$ and $TOL = 10^{-7}$.
3. ALM: Augmented Lagrangian Algorithm 3 which is proposed in [17] with $r = 0.1$, $\mu = 0.05$ and $TOL = 10^{-9}$.

We tabulate the computation cost, number of iteration and SNR of all the algorithms above in Table 6.3. We also display the visual results of Tomato added by Gaussian noise with mean zero and variance $\sigma = 10$ in Figure 6.5.

Table 6.3: The computation summary for the high order model on Tomato.

	$\sigma=10$			$\sigma=25$		
	CPU(s)	ITER	SNR(dB)	CPU(s)	ITER	SNR(dB)
FFPALM	16.38	182	24.79	19.28	216	22.95
FAMA	17.80	193	24.73	20.63	226	22.92
ALM	30.12	106	24.76	31.07	113	22.99

Figure 6.5: The high order model of Tomato. Row one from left to right: original image, noise image and recovered image by ROF model. Row two from left to right: recovered image by FFPALM, FAMA and ALM.

6.6.5 The TV- L^1 model

Another basic and successful image restoration model is the TV- L^1 model, which is obtained by replacing the squared L^2 fidelity term in the ROF model (3.3) with the L^1 fidelity term. Compared to the ROF model, the TV- L^1 model is more efficient in removing the impulsive noise (e.g., salt & pepper noise) in [26, 27]. Therefore, the TV- L^1 model is quoted as follows

$$\min_u \left\{ \|\nabla u\|_1 + \mu \|u - f\|_1 \right\}. \quad (6.44)$$

The TV- L^1 model is difficult to compute due to the nonlinearity and non-differentiability of both the TV term and data fitting term. There are many traditional methods to solve the TV- L^1 model in [130, 20, 74]. In this chapter, we use the alternating direction method to solve the TV- L^1 model (6.44) by Algorithm 9 and Algorithm 10. To get the constrained problem, we can either introduce one auxiliary variable as in [5] or two auxiliary variables as in [56], which will be discussed separately in the following sections.

I. Two sub-problems algorithms

In the first case, we reformulate the minimization problem (6.44) into the constrained optimization problem by introducing $p = \nabla u$, which gives

$$\begin{aligned} \min_{u,p} \quad & \left\{ \|p\|_1 + \mu \|u - f\|_1 \right\}, \\ \text{s.t.}, \quad & p = \nabla u. \end{aligned} \quad (6.45)$$

The augmented Lagrangian functional of the constrained minimization problem (6.45) is defined as

$$\mathcal{L}_{TV-L^1}(u, p; \lambda) = \|p\|_1 + \mu \|u - f\|_1 + \langle \lambda, \nabla u - p \rangle + \frac{r}{2} \|\nabla u - p\|_2^2. \quad (6.46)$$

To solve the above saddle-point problem, we separate it into two sub-problems and apply an alternative minimization method to the sub-problems. The two sub-problems are given as follows:

- The u -sub problem: for a given p

$$\min_u \left\{ \mu \|u - f\|_1 + \langle \lambda, \nabla u \rangle + \frac{r}{2} \|\nabla u - p\|_2^2 \right\}. \quad (6.47)$$

- The p -sub problem: for a given u

$$\min_p \left\{ \|p\|_1 - \langle \lambda, p \rangle + \frac{r}{2} \|\nabla u - p\|_2^2 \right\}. \quad (6.48)$$

Sub-problems (6.47) and (6.48) can be efficiently solved. For (6.47), the optimality condition gives the following equation

$$\partial(\mu\|u - f\|_1) - \operatorname{div}\lambda - r\operatorname{div}(\nabla u - p) = 0.$$

In the $(k + 1)^{th}$ iteration, if we use the fixed-point augmented Lagrangian Algorithm 9 to the u -sub problem, we have

$$\partial(\mu\|u - f\|_1) = \operatorname{div}\lambda^{k+1} + r\Delta u^k - r\operatorname{div}p^k. \quad (6.49)$$

On the other hand, if we apply the alternating minimization Algorithm 10 to the u -sub problem, then we have

$$\partial(\mu\|u - f\|_1) = \operatorname{div}\lambda^{k+1}. \quad (6.50)$$

Furthermore, the gradient descent can be applied to solve (6.49) and (6.50), which gives

$$u^{k+1} = (I + \zeta\partial H)^{-1}\tilde{u},$$

where $\zeta > 0$ is a constant and $H(u) = \mu\|u - f\|_1$ and

$$\tilde{u} = \begin{cases} u^k - \zeta(\operatorname{div}\lambda^{k+1} + r\Delta u^k - r\operatorname{div}p^k), & \text{for (6.49),} \\ u^k - \zeta\operatorname{div}\lambda^{k+1}, & \text{for (6.50).} \end{cases}$$

For both cases, the solution of the u -sub problem can be obtained by the point-wise shrinkage operator as follows

$$u = \operatorname{shrink}(\tilde{u} - f, \zeta\mu) + f.$$

Therefore, we have

$$u(i, j) = \begin{cases} \tilde{u}(i, j) - \zeta\mu, & \text{for } \tilde{u}(i, j) - f(i, j) > \zeta\mu, \\ f(i, j), & \text{for } |\tilde{u}(i, j) - f(i, j)| \leq \zeta\mu, \\ \tilde{u}(i, j) + \zeta\mu, & \text{for } \tilde{u}(i, j) - f(i, j) < -\zeta\mu. \end{cases}$$

Therefore, we get the following fixed-point augmented Lagrangian algorithm for the TV- L^1 model as Algorithm 13: Set $\lambda^0 = 0$, $p^0 = 0$, $u^0 = f$,

Algorithm 13 Two sub-problems based fixed-point ALM for the TV- L^1 model

1. Update λ^{k+1} from: $\lambda^{k+1} = \lambda^k + r(\nabla u^k - p^k)$.
 2. Solve u^{k+1} from: $u^{k+1} = \text{shrink}(\tilde{u} - f, \zeta\mu) + f$.
 3. Solve p^{k+1} from: $p^{k+1} = \text{shrink}(\nabla u^{k+1} + \frac{\lambda^{k+1}}{r}, \frac{1}{r})$.
-

II. Three sub-problems algorithms

In this case, we introduce two variable $p = \nabla u$ and $z = u$ and reformulate the TV- L^1 model (6.44) to be the following constrained optimization problem

$$\begin{aligned} \min_{p, z} \quad & \left\{ \|p\|_1 + \mu \|z - f\|_1 \right\}, \\ \text{s.t.,} \quad & p = \nabla u, \quad z = u. \end{aligned} \quad (6.51)$$

Thus, the corresponding augmented Lagrangian formulation of the constrained minimization problem (6.51) becomes

$$\begin{aligned} \mathcal{L}_{TV-L^1}(p, z, u; \lambda_p, \lambda_z) \\ = \|p\|_1 + \mu \|z - f\|_1 + \langle \lambda_p, \nabla u - p \rangle + \langle \lambda_z, u - z \rangle + \frac{r_p}{2} \|\nabla u - p\|_2^2 + \frac{r_z}{2} \|u - z\|_2^2. \end{aligned} \quad (6.52)$$

To solve the above saddle-point problem, we separate the functional into three sub-problems and apply the alternative minimization method to the system. The corresponding three sub-problems are given as follows:

- The u -sub problem: for given z and p

$$\min_u \left\{ \langle \lambda_p, \nabla u \rangle + \langle \lambda_z, u \rangle + \frac{r_p}{2} \|\nabla u - p\|_2^2 + \frac{r_z}{2} \|u - z\|_2^2 \right\}. \quad (6.53)$$

- The z -sub problem: for given u and p

$$\min_z \left\{ \mu \|z - f\|_1 - \langle \lambda_z, z \rangle + \frac{r_z}{2} \|u - z\|_2^2 \right\}. \quad (6.54)$$

- The p -sub problem: for given u and z

$$\min_p \left\{ \|p\|_1 - \langle \lambda_p, p \rangle + \frac{r_p}{2} \|\nabla u - p\|_2^2 \right\}. \quad (6.55)$$

Both sub-problems (6.54) and (6.55) can be solved by the shrinkage operator. For the u -sub problem, the Euler-Lagrange equation of (6.53) is

$$-\operatorname{div} \lambda_p + \lambda_z - r_p \operatorname{div}(\nabla u - p) + r_z(u - z) = 0. \quad (6.56)$$

In the $(k+1)^{th}$ iteration, we apply the fixed-point method to (6.56) and rewrite it as follows

$$-\operatorname{div} \lambda_p^{k+1} + \lambda_z^{k+1} - r_p \operatorname{div}(\nabla u^k - p^k) + r_z(u^{k+1} - z^k) = 0. \quad (6.57)$$

Furthermore, we relax the term $\operatorname{div} \lambda_p^{k+1}$ to $\operatorname{div} \lambda_p^k$ in (6.57). We then solve the solution of the u -sub problem in the $(k+1)^{th}$ iteration from

$$u^{k+1} = z^k + \frac{1}{r_z} \operatorname{div} \lambda_p^{k+1} - \frac{1}{r_z} \lambda_z^{k+1}. \quad (6.58)$$

Therefore, we have another fixed-point augmented Lagrangian algorithm for the TV- L^1 model as Algorithm 14: Set $\lambda_p^0 = 0$, $\lambda_z^0 = 0$, $p^0 = 0$, $u^0 = f$,

III. Numerical examples

In the following, we give the numerical results to demonstrate the performance of the proposed algorithms for the TV- L^1 model. We choose the image Cameraman (512×512) and add the salt & pepper noise: 20% and 50% to the image. Since the TV- L^1 model poses a non-smooth optimization problem, the acceleration Algorithm 12 can be applied to neither Algorithm 13 nor Algorithm 14. For performance evaluation, we use the following algorithms and parameters

Algorithm 14 Three sub-problems based fixed-point ALM for the TV- L^1 model

1. Update λ_p^{k+1} and λ_z^{k+1} from: $\lambda_p^{k+1} = \lambda_p^k + r_p(\nabla u^k - p^k)$, $\lambda_z^{k+1} = \lambda_z^k + r_z(u^k - z^k)$.
 2. Solve u^{k+1} from: $u^{k+1} = z^k + \frac{1}{r_z} \operatorname{div} \lambda_p^{k+1} - \frac{1}{r_z} \lambda_z^{k+1}$.
 3. Solve z^{k+1} from: $z^{k+1} = \operatorname{shrink}(u^{k+1} - f + \frac{\lambda_z^{k+1}}{r_z}, \frac{1}{r_z}) + f$.
 4. Solve p^{k+1} from: $p^{k+1} = \operatorname{shrink}(\nabla u^{k+1} + \frac{\lambda_p^{k+1}}{r_p}, \frac{1}{r_p})$.
-

Table 6.4: The computation summary for the experiment of the TV- L^1 model on Cameraman with 20% and 50% pepper & salt noise.

	20%			50%		
	CPU(s)	ITER	SNR(dB)	CPU(s)	ITER	SNR(dB)
Algorithm 5	3.14	59	16.65	3.80	82	13.17
Algorithm 6	3.62	60	16.58	5.37	88	13.18
ALM	4.42	19	16.67	6.62	31	13.20

1. Algorithm 13: Two sub-problems based fixed-point augmented Lagrangian method with $\mu = 1$, $r = 0.05$, $\zeta = 5$ and $TOL = 5 \times 10^{-5}$.
2. Algorithm 14: Three sub-problems based fixed-point augmented Lagrangian method with $\mu = 1$, $r_w = 0.03$, $r_z = 0.2$ and $TOL = 5 \times 10^{-5}$.
3. ALM: Augmented Lagrangian method which is proposed in Wu-Zhang-Tai formulation [56] with $\mu = 1$, $r_w = 0.1$, $r_z = 0.1$ and $TOL = 5 \times 10^{-6}$.

We display the computational time, iteration and SNR of each algorithm in Table 6.4 and the selected visual results of Cameraman with noise 50% in Figure 6.6. We plot the SNR and numerical energy from the Algorithm 13, Algorithm 14 and ALM in Figure 6.7.

Figure 6.6: The TV- L^1 model of Cameraman. Row One: noise image and recovered image by ALM. Row Two: recovered image by Algorithm 13 and Algorithm 14.

Figure 6.7: Plots of SNR and numerical energy of Cameraman.

CHAPTER 7

Augmented Lagrangian Method for Euler's Elastica

Contents

7.1	Introduction	141
7.2	Augmented Lagrangian Method for Euler's Elastica Model	143
7.3	The Existing Algorithm	144
7.4	The Proposed Algorithm	146
7.5	Numerical Solutions for Subproblems	148
7.6	Numerical Examples	156

7.1 Introduction

Euler's elastica is an important image processing model, which was first introduced in [36]. In this chapter, we concern the following Euler's elastica minimization problem for image denoising

$$\min_u \left\{ \int_{\Omega} \left(a + b \left(\nabla \cdot \frac{\nabla u}{|\nabla u|} \right)^2 \right) |\nabla u| + \frac{\mu}{t} \int_{\Omega} |u - f|^t \right\}. \quad (7.1)$$

In the above energy functional, we set the power of the mean curvature, τ in (3.14), to be $\tau = 2$, but the techniques developed below can be extended to the case $\tau = 1$ without many efforts. In (7.1), we choose t according to the type of noise in f : e.g., $t = 1$ for the salt & pepper noise and $t = 2$ for the Gaussian white noise. Both cases are included in this chapter.

The numerical solutions [33, 35] of Euler's elastica functional (3.14) are normally complex and time consuming due to the high nonlinearity of the partial differential equations (PDEs). In [131, 132], graph-cuts methods are applied to the high order models and Euler's elastica. Recently, the augmented Lagrangian method (ALM) [133] has been implemented to efficiently solve the variational models in image processing; see [16, 17]. Along the similar idea, [134] applies the ALM to the Euler's elastica model (3.14), in which great efficiency is achieved by solving the subproblems emerging from the augmented Lagrangian functional with Fast Fourier Transform (FFT). In their approach, a quadratic penalty term in the augmented Lagrangian functional is relaxed to the first order and a frozen coefficient method with FFT is used to solve the coupled PDEs with the variable coefficient. The dependence of the FFT gives rise to a limitation to apply the algorithm [134] to more general problems such as the problems defined on non-flat geometries [135]. To overcome this hurdle, [136] proposes an algorithm for the elastica energy by replacing FFT with the cheap arithmetic operation.

In this chapter, we use the operator splitting technique to propose a novel augmented Lagrangian algorithm for Euler's elastica model (3.14). We decompose the saddle-point problem into subproblems to simplify the structure. Instead of disposing of the quadratic term in the subproblem related to the normal vector, we apply a fixed-point method to find a closed form solution, which is different from [136]. Furthermore, in contrast to [134], we introduce a new variable into the constrained problem to avoid solving the PDEs with variable coefficients. In the proposed formulation, all subproblems are either linear which can be solved efficiently by the iterative solver, or having closed-form solutions. Therefore, the new method is computationally efficient in terms of both memory requests and computational costs.

7.2 Augmented Lagrangian Method for Euler's Elastica Model

In this section, we propose an augmented Lagrangian formulation for the Euler's elastica energy (7.1). First, we introduce two extra variables $\mathbf{p}$ and $\mathbf{n}$ to (7.1) and cast it into the following constrained minimization problem

$$\begin{aligned} \min_{u, \mathbf{p}, \mathbf{n}} \quad & \left\{ \int_{\Omega} \left(a + b(\nabla \cdot \mathbf{n})^2 \right) |\mathbf{p}| d\Omega + \frac{\mu}{t} \int_{\Omega} |u - f|^t d\Omega \right\}, \\ \text{s.t.,} \quad & \mathbf{p} = \nabla u; \quad \mathbf{n} = \frac{\mathbf{p}}{|\mathbf{p}|}. \end{aligned} \quad (7.2)$$

The second constraint in (7.2) is equivalent to $\mathbf{p} = |\mathbf{p}|\mathbf{n}$ if $\mathbf{p} \neq \mathbf{0}$. Therefore, the corresponding augmented Lagrangian functional for the constrained optimization problem (7.2) is defined as follows

$$\begin{aligned} \mathcal{L}(u, \mathbf{p}, \mathbf{n}; \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) = & \int_{\Omega} \left(a + b(\nabla \cdot \mathbf{n})^2 \right) |\mathbf{p}| d\Omega + \frac{\mu}{t} \int_{\Omega} |u - f|^t d\Omega \\ & + \int_{\Omega} (\mathbf{p} - |\mathbf{p}|\mathbf{n}) \cdot \boldsymbol{\lambda}_1 d\Omega + \frac{r_1}{2} \int_{\Omega} (\mathbf{p} - |\mathbf{p}|\mathbf{n})^2 d\Omega \\ & + \int_{\Omega} (\mathbf{p} - \nabla u) \cdot \boldsymbol{\lambda}_2 d\Omega + \frac{r_2}{2} \int_{\Omega} (\mathbf{p} - \nabla u)^2 d\Omega, \end{aligned} \quad (7.3)$$

where $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are the Lagrange multipliers and r_1 and r_2 are positive penalty parameters. We aim to seek a saddle-point of the augmented Lagrangian functional (7.3), which satisfies

$$\begin{aligned} \text{Find } & (u^*, \mathbf{p}^*, \mathbf{n}^*; \boldsymbol{\lambda}_1^*, \boldsymbol{\lambda}_2^*), \\ \text{s.t., } & \mathcal{L}(u^*, \mathbf{p}^*, \mathbf{n}^*; \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2) \leq \mathcal{L}(u^*, \mathbf{p}^*, \mathbf{n}^*; \boldsymbol{\lambda}_1^*, \boldsymbol{\lambda}_2^*) \leq \mathcal{L}(u, \mathbf{p}, \mathbf{n}; \boldsymbol{\lambda}_1^*, \boldsymbol{\lambda}_2^*), \quad \forall (u, \mathbf{p}, \mathbf{n}; \boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2). \end{aligned} \quad (7.4)$$

Based on the above constrained formulation, we develop the novel scheme for the elastica energy (7.1) in this chapter. We first review the existing algorithm in [134] for solving the minimization problem (7.2). We further modify the augmented Lagrangian functional (7.3) by introducing another new variable. The advantages of

the proposed algorithm compared to the existing algorithm in [134] are illustrated as well.

7.3 The Existing Algorithm

In this section, we give a brief review of the augmented Lagrangian method applied to Euler's elastica in the Tai-Hahn-Chung (THC) formulation [134]. It is difficult to solve the augmented Lagrangian functional (7.3) efficiently because of the non-differentiable quadratic term involved $|\mathbf{p}|$. To overcome the difficulty, the authors introduce one more variable into the constrained problem (7.2), which satisfies

$$\mathbf{m} = \mathbf{n} \quad \text{and} \quad |\mathbf{m}| \leq 1.$$

The constrained optimization problem (7.2) is reformatted as the following structure

$$\begin{aligned} \min_{u, \mathbf{p}, \mathbf{n}, \mathbf{m}} \quad & \left\{ \int_{\Omega} \left(a + b(\nabla \cdot \mathbf{n})^2 \right) |\mathbf{p}| d\Omega + \frac{\mu}{t} \int_{\Omega} |u - f|^t d\Omega \right\}, \\ \text{s.t.}, \quad & \mathbf{p} = \nabla u; \quad \mathbf{n} = \mathbf{m}; \quad |\mathbf{p}| = \mathbf{m} \cdot \mathbf{p}; \quad |\mathbf{m}| \leq 1. \end{aligned} \quad (7.5)$$

By the constraint $|\mathbf{m}| \leq 1$, there exists the relationship $|\mathbf{p}| - \mathbf{m} \cdot \mathbf{p} \geq 0$, a.e. in Ω . Consequently, the quadratic penalty term $\int_{\Omega} (|\mathbf{p}| - \mathbf{m} \cdot \mathbf{p})^2$ in (7.3) can be relaxed to a first order penalty term. The augmented Lagrangian functional for the constrained problem (7.5) is defined as follows

$$\begin{aligned} \mathcal{L}(u, \mathbf{p}, \mathbf{n}, \mathbf{m}; \lambda_1, \lambda_2, \lambda_3) = & \int_{\Omega} \left(a + b(\nabla \cdot \mathbf{n})^2 \right) |\mathbf{p}| d\Omega + \frac{\mu}{t} \int_{\Omega} |u - f|^t d\Omega \\ & + \int_{\Omega} (|\mathbf{p}| - \mathbf{m} \cdot \mathbf{p}) \lambda_1 d\Omega + r_1 \int_{\Omega} (|\mathbf{p}| - \mathbf{m} \cdot \mathbf{p}) d\Omega + \int_{\Omega} (\mathbf{p} - \nabla u) \cdot \lambda_2 d\Omega \\ & + \frac{r_2}{2} \int_{\Omega} (\mathbf{p} - \nabla u)^2 d\Omega + \int_{\Omega} (\mathbf{n} - \mathbf{m}) \cdot \lambda_3 d\Omega + \frac{r_3}{2} \int_{\Omega} (\mathbf{n} - \mathbf{m})^2 + \delta_{\mathcal{R}}(\mathbf{m}), \end{aligned} \quad (7.6)$$

where the set $\mathcal{R} := \{ \mathbf{m} \in \mathbf{L}^2(\Omega) \mid |\mathbf{m}| \leq 1 \text{ a.e. in } \Omega \}$ and $\delta_{\mathcal{R}}$ is the indicator

function defined by

$$\delta_{\mathcal{R}}(\mathbf{m}) := \begin{cases} 0, & \mathbf{m} \in \mathcal{R}, \\ +\infty, & \text{otherwise.} \end{cases}$$

In [134], the energy functional (7.6) is decomposed into a number of subproblems as follows

$$\epsilon_1(u) = \frac{\mu}{t} \int_{\Omega} |u - f|^t d\Omega - \int_{\Omega} \nabla u \cdot \boldsymbol{\lambda}_2 d\Omega + \frac{r_2}{2} \int_{\Omega} (\mathbf{p} - \nabla u)^2 d\Omega, \quad (7.7)$$

$$\epsilon_2(\mathbf{p}) = \int_{\Omega} (a + b(\nabla \cdot \mathbf{n})^2) |\mathbf{p}| d\Omega + \int_{\Omega} (|\mathbf{p}| - \mathbf{m} \cdot \mathbf{p}) \lambda_1 d\Omega \quad (7.8)$$

$$\begin{aligned} & + r_1 \int_{\Omega} (|\mathbf{p}| - \mathbf{m} \cdot \mathbf{p}) d\Omega + \int_{\Omega} \mathbf{p} \cdot \boldsymbol{\lambda}_2 d\Omega + \frac{r_2}{2} \int_{\Omega} (\mathbf{p} - \nabla u)^2 d\Omega, \\ \epsilon_3(\mathbf{n}) &= \int_{\Omega} b |\mathbf{p}| (\nabla \cdot \mathbf{n})^2 d\Omega + \int_{\Omega} \mathbf{n} \cdot \boldsymbol{\lambda}_3 d\Omega + \frac{r_3}{2} \int_{\Omega} (\mathbf{n} - \mathbf{m})^2 d\Omega, \end{aligned} \quad (7.9)$$

$$\epsilon_4(\mathbf{m}) = -(\lambda_1 + r_1) \int_{\Omega} \mathbf{p} \cdot \mathbf{m} d\Omega - \int_{\Omega} \mathbf{m} \cdot \boldsymbol{\lambda}_3 d\Omega + \frac{r_3}{2} \int_{\Omega} (\mathbf{n} - \mathbf{m})^2 d\Omega + \delta_{\mathcal{R}}(\mathbf{m}). \quad (7.10)$$

The above subproblems are solved alternatively in one iteration in the algorithm THC. For the u -sub problem (7.7), since its Euler-Lagrange equation is a linear PDE, it can be efficiently solved by the FFT. There are closed form solutions for both the $\mathbf{p}$ -sub problem (7.8) and $\mathbf{m}$ -sub problem (7.10) due to their good structures. In the algorithm THC, the most difficult and time-consuming part lies in the $\mathbf{n}$ -sub problem (7.9).

The Euler-Lagrange equation of $\mathbf{n}$ -sub problem (7.9) gives us the coupled PDE as follows

$$-2\nabla(b|\mathbf{p}|\nabla \cdot \mathbf{n}) + r_3(\mathbf{n} - \mathbf{m}) + \boldsymbol{\lambda}_3 = 0. \quad (7.11)$$

The coefficient of $\nabla \cdot \mathbf{n}$ is a variable in (7.11), which makes it difficult to handle. Aimed to use the FFT, a frozen coefficient method is applied to the coupled PDEs in [134]. However, there are some drawbacks caused by this method. First, the method needs the inner iterations. Since the coefficient of $\nabla \cdot \mathbf{n}$ is fixed artificially in each iteration inner iteration, it may require more iterations to achieve the convergence. Second, to solve the coupled PDEs (7.11), two FFT algorithms are necessary in an

inner iteration. Considering FFT used in u -sub problem, three times of FFT are involved at least in one outer iteration of the algorithm in [134]. The above points will increase the computational costs of the algorithm in [134]. In addition, there exist cases that the domain or boundary condition assigned to the minimization problem (3.14) is not applicable to the FFT. Therefore, we consider to find a better way to solve the augmented Lagrangian functional (7.3).

7.4 The Proposed Algorithm

In this section, we propose a more efficient algorithm to solve the augmented Lagrangian functional (7.3). Unlike the algorithm in [134], we keep the two constraints in the optimization problem (7.2) and introduce one more variable to denote the mean curvature, which is

$$h = \nabla \cdot \mathbf{n}.$$

We use the variable h to remove the variable coefficient in the n -sub problem (7.9) in Tai-Hahn-Chung formulation [134]. Therefore, the Euler's elastica model (3.14) is reformulated as the following constrained optimization problem in our work

$$\begin{aligned} \min_{u, \mathbf{p}, \mathbf{n}, h} \quad & \left\{ \int_{\Omega} (a + bh^2) |\mathbf{p}| d\Omega + \frac{\mu}{t} \int_{\Omega} |u - f|^t d\Omega \right\}, \\ \text{s.t.,} \quad & \mathbf{p} = \nabla u; \quad \mathbf{p} = |\mathbf{p}| \mathbf{n}; \quad h = \nabla \cdot \mathbf{n}. \end{aligned} \quad (7.12)$$

We use the augmented Lagrangian method in [16] to rewrite the constrained problem (7.12) into an unconstrained minimization problem. We define the augmented Lagrangian functional for (7.12) as follows

$$\begin{aligned} \mathcal{L}(u, \mathbf{p}, \mathbf{n}, h; \lambda_1, \lambda_2, \lambda_3) = & \int_{\Omega} (a + bh^2) |\mathbf{p}| d\Omega + \frac{\mu}{t} \int_{\Omega} |u - f|^t d\Omega + \int_{\Omega} (\mathbf{p} - |\mathbf{p}| \mathbf{n}) \cdot \lambda_1 d\Omega \\ & + \frac{r_1}{2} \int_{\Omega} (\mathbf{p} - |\mathbf{p}| \mathbf{n})^2 d\Omega + \int_{\Omega} (\mathbf{p} - \nabla u) \cdot \lambda_2 d\Omega + \frac{r_2}{2} \int_{\Omega} (\mathbf{p} - \nabla u)^2 d\Omega \\ & + \int_{\Omega} (h - \nabla \cdot \mathbf{n}) \lambda_3 d\Omega + \frac{r_3}{2} \int_{\Omega} (h - \nabla \cdot \mathbf{n})^2 d\Omega. \end{aligned} \quad (7.13)$$

We apply an iterative algorithm to solve the saddle-point problem corresponding to the augmented Lagrangian functional (7.13); see Algorithm 15.

Algorithm 15 Augmented Lagrangian Method for the Euler's elastica model

1. Initialization: $u^0, \mathbf{p}^0, \mathbf{n}^0, h^0$ and $\boldsymbol{\lambda}_1^0, \boldsymbol{\lambda}_2^0, \lambda_3^0$.
2. For $k = 0, 1, 2, \dots$, compute $(u^k, \mathbf{p}^k, \mathbf{n}^k, h^k)$ as an approximate minimizer of the augmented Lagrangian functional with the Lagrange multiplier $\boldsymbol{\lambda}_1^{k-1}, \boldsymbol{\lambda}_2^{k-1}$ and λ_3^{k-1} , i.e.,

$$(u^k, \mathbf{p}^k, \mathbf{n}^k, h^k) \approx \arg \min \mathcal{L}(u, \mathbf{p}, \mathbf{n}, h; \boldsymbol{\lambda}_1^{k-1}, \boldsymbol{\lambda}_2^{k-1}, \lambda_3^{k-1}). \quad (7.14)$$

3. Update the Lagrange multipliers:

$$\boldsymbol{\lambda}_1^k = \boldsymbol{\lambda}_1^{k-1} + r_1(\mathbf{p}^k - |\mathbf{p}^k| \mathbf{n}^k), \quad (7.15)$$

$$\boldsymbol{\lambda}_2^k = \boldsymbol{\lambda}_2^{k-1} + r_2(\mathbf{p}^k - \nabla u^k), \quad (7.16)$$

$$\lambda_3^k = \lambda_3^{k-1} + r_3(h^k - \nabla \cdot \mathbf{n}^k). \quad (7.17)$$

Since the variable $u, \mathbf{p}, \mathbf{n}, h$ are coupled together in the minimization problem (7.14), it is difficult to solve all the variables simultaneously. Therefore, we use the decomposition technique to separate the problem (7.14) into a number of sub minimization problems. The energy functionals corresponding to each variable are given separately as follows

$$\mathcal{E}_1(u) = \frac{\mu}{t} \int_{\Omega} |u - f|^t d\Omega - \int_{\Omega} \nabla u \cdot \boldsymbol{\lambda}_2 d\Omega + \frac{r_2}{2} \int_{\Omega} (\mathbf{p} - \nabla u)^2 d\Omega, \quad (7.18)$$

$$\begin{aligned} \mathcal{E}_2(\mathbf{p}) = & \int_{\Omega} (a + bh^2) |\mathbf{p}| d\Omega + \int_{\Omega} (\mathbf{p} - |\mathbf{p}| \mathbf{n}) \cdot \boldsymbol{\lambda}_1 d\Omega + \frac{r_1}{2} \int_{\Omega} (\mathbf{p} - |\mathbf{p}| \mathbf{n})^2 d\Omega \\ & + \int_{\Omega} \mathbf{p} \cdot \boldsymbol{\lambda}_2 d\Omega + \frac{r_2}{2} \int_{\Omega} (\mathbf{p} - \nabla u)^2 d\Omega, \end{aligned} \quad (7.19)$$

$$\mathcal{E}_3(\mathbf{n}) = - \int_{\Omega} |\mathbf{p}| \mathbf{n} \cdot \boldsymbol{\lambda}_1 d\Omega + \frac{r_1}{2} \int_{\Omega} (\mathbf{p} - |\mathbf{p}| \mathbf{n})^2 d\Omega - \int_{\Omega} \lambda_3 \nabla \cdot \mathbf{n} d\Omega + \frac{r_3}{2} \int_{\Omega} (h - \nabla \cdot \mathbf{n})^2 d\Omega, \quad (7.20)$$

$$\mathcal{E}_4(h) = \int_{\Omega} b |\mathbf{p}| h^2 d\Omega + \int_{\Omega} \lambda_3 h d\Omega + \frac{r_3}{2} \int_{\Omega} (h - \nabla \cdot \mathbf{n})^2 d\Omega. \quad (7.21)$$

In our work, all subproblems (7.18) to (7.21) are efficiently solved. We iteratively and alternatively solve the minimizer from each the above subproblem by using Algorithm 16.

Algorithm 16 Augmented Lagrangian Method for the Euler's elastica model

1. Initialization: $\tilde{u}^0 = u^{k-1}$, $\tilde{\mathbf{p}}^0 = \mathbf{p}^{k-1}$, $\tilde{\mathbf{n}}^0 = \mathbf{n}^{k-1}$ and $\tilde{h}^0 = h^{k-1}$.
 2. For $l = 0, 1, 2, \dots, L$ and fixed Lagrange multipliers $\lambda_1 = \lambda_1^{k-1}$, $\lambda_2 = \lambda_2^{k-1}$ and $\lambda_3 = \lambda_3^{k-1}$: Compute $\tilde{u}^l$ from (7.18), $\tilde{\mathbf{p}}^l$ from (7.19), $\tilde{\mathbf{n}}^l$ from (7.20) and $\tilde{h}^l$ from (7.21).
 3. $(u^k, \mathbf{p}^k, \mathbf{n}^k, h^k) = (\tilde{u}^L, \tilde{\mathbf{p}}^L, \tilde{\mathbf{n}}^L, \tilde{h}^L)$.
-

Here L can be chosen using some convergence test techniques. In this chapter, we simply set $L = 1$. We will discuss the specific solution to each subproblem in the forthcoming section.

7.5 Numerical Solutions for Subproblems

In this section, we explain how to find the minimizer of each subproblem. We use a staggered grid system as in Figure 7.1 to solve the energy functional minimization (7.18) to (7.21) and update the Lagrange multipliers from (7.15) to (7.17).

7.5.1 Notations

Similar to the last chapter, we define an image u as a function

$$u : \{1, \dots, M\} \times \{1, \dots, N\},$$

where $M, N \geq 2$. We denote the Euclidean space $\mathbb{R}^{M \times N}$ as V and define another inner product vector space: $Q = V \times V$.

We use the discrete backward and forward differential operators for $u \in V$, which are defined with periodic boundary condition as given in Chapter 6.

Figure 7.1: Grid definition. The rule of indexing variables in the augmented Lagrangian functional (7.3): u , h and λ_3 are defined on $\bullet$ -nodes. The first and second component of $\mathbf{p}$, $\mathbf{n}$, λ_1 and λ_2 are defined on $\circ$ -nodes and $\square$ -nodes, respectively.

Consequently, the discrete forward and backward gradient operator $\nabla : V \rightarrow Q$ are given as follows:

$$\nabla u(i, j) := (\partial_x^+ u(i, j), \partial_y^+ u(i, j)).$$

And the corresponding discrete backward and forward adjoint operator $\text{div} : Q \rightarrow V$ of $-\nabla$ is obtained:

$$\text{div} \mathbf{p}(i, j) := \partial_x^- p_1(i, j) + \partial_y^- p_2(i, j).$$

When a variable defined on $\circ$ -nodes (or $\square$ -nodes) needs to be evaluated at $(i, j) \in \square$ -nodes (or $\circ$ -nodes), for example $\mathbf{p} = (p_1, p_2)$, we use the average operators:

$$\begin{aligned} \mathcal{A}_{i,j}^{\square}(p_1) &= \frac{p_1(i, j+1) + p_1(i-1, j+1) + p_1(i, j) + p_1(i-1, j)}{4}, \\ \mathcal{A}_{i,j}^{\circ}(p_2) &= \frac{p_2(i+1, j) + p_2(i, j) + p_2(i+1, j-1) + p_2(i, j-1)}{4}, \end{aligned}$$

where p_1 and p_2 are defined on $\circ$ -nodes and $\square$ -nodes respectively. We need to define a special operator to measure the magnitude of $\mathbf{p}$ at $(i, j) \in \bullet$ -nodes as follows:

$$|\mathcal{A}_{i,j}^{\bullet}(\mathbf{p})| = \left(\left(\frac{p_1(i, j) + p_1(i-1, j)}{2} \right)^2 + \left(\frac{p_2(i, j) + p_2(i, j-1)}{2} \right)^2 \right)^{\frac{1}{2}}.$$

We compute the divergence of $\mathbf{p}$ at $(i, j) \in \bullet$ -nodes using the following operator:

$$\operatorname{div}_{i,j}^{\bullet}(\mathbf{p}) = p_1(i, j) - p_1(i-1, j) + p_2(i, j) - p_2(i, j-1).$$

7.5.2 Subproblems

In the following, for each subproblem, we denote the fixed Lagrange multipliers in the previous $(k-1)^{th}$ iteration as $\lambda_1 = \lambda_1^{k-1}$, $\lambda_2 = \lambda_2^{k-1}$ and $\lambda_3 = \lambda_3^{k-1}$.

The u -sub Problem

For the u -sub problem (7.18), the choice of t in our paper is either $t = 1$ or $t = 2$ for different applications. We discuss the solution to the minimization problem related to u w.r.t. the value of t as follows:

– Case 1: $t = 1$

For this case, we introduce a new variable to the u -sub problem and rewrite it as a constrained problem in the following way

$$\begin{aligned} \min_{u,v} \quad & \left\{ \mu \int_{\Omega} |v - f| d\Omega - \int_{\Omega} \nabla u \cdot \lambda_2 d\Omega + \frac{r_2}{2} \int_{\Omega} (\mathbf{p} - \nabla u)^2 d\Omega \right\}, \\ \text{s.t., } & v = u. \end{aligned} \tag{7.22}$$

We apply the augmented Lagrangian method to (7.22) and gain the following functional

$$\begin{aligned} \mathcal{L}(u, v; \lambda_4) = & \mu \int_{\Omega} |v - f| d\Omega - \int_{\Omega} \nabla u \cdot \lambda_2 d\Omega + \frac{r_2}{2} \int_{\Omega} (\mathbf{p} - \nabla u)^2 d\Omega \\ & + \int_{\Omega} \lambda_4 (v - u) d\Omega + \frac{r_4}{2} \int_{\Omega} (v - u)^2 d\Omega. \end{aligned} \tag{7.23}$$

We then use an iterative algorithm to solve the saddle-point problem (7.23); see Algorithm 17.

Algorithm 17 Augmented Lagrangian Method for u-sub Problem

1. Initialization: u^0 and v^0 .
2. For $k = 0, 1, 2, \dots$, compute (u^k, v^k) as an approximate minimizer of the augmented Lagrangian functional with the Lagrange multiplier λ_4^{k-1} , i.e.,

$$(u^k, v^k) \approx \arg \min \mathcal{L}(u, v; \lambda_4^{k-1}). \quad (7.24)$$

3. Update the Lagrange multipliers:

$$\lambda_4^k = \lambda_4^{k-1} + r_4(v^k - u^k). \quad (7.25)$$

To solve both variable u and v in (7.24), we separate it into the following two subproblems:

$$\min_u - \left\{ \int_{\Omega} \nabla u \cdot \boldsymbol{\lambda}_2 d\Omega + \frac{r_2}{2} \int_{\Omega} (\mathbf{p} - \nabla u)^2 d\Omega - \int_{\Omega} \lambda_4 u d\Omega + \frac{r_4}{2} \int_{\Omega} (v - u)^2 d\Omega \right\}, \quad (7.26)$$

for a given v , and

$$\min_v \left\{ \mu \int_{\Omega} |v - f| d\Omega + \int_{\Omega} \lambda_4 v d\Omega + \frac{r_4}{2} \int_{\Omega} (v - u)^2 d\Omega \right\}, \quad (7.27)$$

for a given u .

The Euler-Lagrange equation of (7.26) gives us a linear PDE

$$(r_4 - r_2 \Delta)u = \lambda_4 + r_4 v - \nabla \cdot \boldsymbol{\lambda}_2 - r_2 \nabla \cdot \mathbf{p}. \quad (7.28)$$

The PDE problem (7.28) can be solved efficiently by a wide range of linear iterative methods, such as Jacobi method, Gauss-Seidel method. In this chapter, we choose to use one sweep of the Gauss-Seidel method as the solver, which is enough to approximate the solution.

On the other hand, we consider the solution to the v -sub problem. Let $w = u - \frac{\lambda_4}{r_4}$. We can rewrite (7.27) as follows

$$\min_v \left\{ \mu \int_{\Omega} |v - f| + \frac{r_4}{2} \int_{\Omega} (v - w)^2 \right\}. \quad (7.29)$$

We can explicitly compute the optimal value of v in (7.29) using the shrinkage operator. For a grid point $(i, j) \in \Omega$, we simply compute

$$v(i, j) = \max \left\{ 0, 1 - \frac{\mu}{r_4 |w(i, j) - f(i, j)|} \right\} (w(i, j) - f(i, j)) + f(i, j).$$

– Case 2: $t = 2$

For this case, the minimization problem related to u can be simplified as

$$\min_u \left\{ \frac{\mu}{2} \int_{\Omega} (u - f)^2 - \int_{\Omega} \nabla u \cdot \boldsymbol{\lambda}_2 + \frac{r_2}{2} \int_{\Omega} (\mathbf{p} - \nabla u)^2 \right\}. \quad (7.30)$$

Therefore, the optimality condition of (7.26) gives us a linear equation of u , which is

$$(\mu - r_2 \Delta)u = \mu f - \nabla \cdot \boldsymbol{\lambda}_2 - r_2 \nabla \cdot \mathbf{p}. \quad (7.31)$$

Similarly, the PDE problem (7.31) can be solved efficiently by one sweep of the Gauss-Seidel method.

The $\mathbf{p}$ -sub Problem

For the $\mathbf{p}$ -sub problem, it is difficult to solve the Euler-Lagrange equation of (7.19) due to the non-differentiability element $|\mathbf{p}|$ in the quadratic term. To avoid this situation, we consider to apply a fixed-point formulation to the constraint $\mathbf{p} = |\mathbf{p}|\mathbf{n}$ in the k^{th} iteration, which gives

$$\mathbf{p} = |\mathbf{p}^{k-1}|\mathbf{n}.$$

To get rid of the nonlinearity and non-differentiability term, we use $\mathbf{p} - |\mathbf{p}^{k-1}|\mathbf{n}$ to replace $\mathbf{p} - |\mathbf{p}|\mathbf{n}$ in the quadratic penalty term in (7.19). Therefore, we reformulate

the energy functional of the variable $\mathbf{p}$ as follows

$$\mathcal{E}_2(\mathbf{p}) = \int_{\Omega} (a + bh^2 - \mathbf{n} \cdot \boldsymbol{\lambda}_1) |\mathbf{p}| d\Omega + \frac{r_1 + r_2}{2} \int_{\Omega} \left(\mathbf{p} - \frac{r_1 |\mathbf{p}^{k-1}| \mathbf{n} + r_2 \nabla u - \boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2}{r_1 + r_2} \right)^2 d\Omega.$$

For the simplicity, let

$$c = a + bh^2 - \mathbf{n} \cdot \boldsymbol{\lambda}_1 \quad \text{and} \quad \mathbf{q} = r_1 |\mathbf{p}^{k-1}| \mathbf{n} + r_2 \nabla u - \boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2.$$

Here, c can be either positive or negative. For each case, there is the closed form solution for solving $\mathbf{p}$ -sub problem. If c is positive, we have the following closed form solution for $\mathbf{p}$

$$\mathbf{p}(i, j) = \frac{1}{r_1 + r_2} \max \left\{ 0, 1 - \frac{c}{|\mathbf{q}(i, j)|} \right\} \mathbf{q}(i, j).$$

And if c is negative, the solution for $\mathbf{p}$ is

$$\mathbf{p}(i, j) = \frac{1}{r_1 + r_2} \left(1 - \frac{c}{|\mathbf{q}(i, j)|} \right) \mathbf{q}(i, j),$$

which belongs to the case when c is positive.

Therefore, for each case the closed form solution for $\mathbf{p}$ in (7.19) can be summarized as follows,

$$\mathbf{p}(i, j) = \frac{1}{r_1 + r_2} \max \left\{ 0, 1 - \frac{c}{|\mathbf{q}(i, j)|} \right\} \mathbf{q}(i, j). \quad (7.32)$$

According to the rule of indexing variables in Figure 7.1, the first and second component of $\mathbf{p}$, $\mathbf{n}$, $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are defined on $\circ$ -nodes and $\square$ -nodes, respectively. The discretization of c and $\mathbf{q}$ at $(i, j) \in \circ$ -node is obtained as follows:

$$\begin{aligned} c(i, j) &= a + b \left(\frac{h(i+1, j) + h(i, j)}{2} \right)^2 - \lambda_{11}(i, j) n_1(i, j) - \mathcal{A}_{i,j}^{\circ}(\lambda_{12}) \mathcal{A}_{i,j}^{\circ}(n_2), \\ q_1(i, j) &= r_2 \left(u(i+1, j) - u(i, j) \right) + r_1 |\mathbf{p}^{k-1}| n_1(i, j) - \lambda_{11}(i, j) - \lambda_{21}(i, j), \\ q_2(i, j) &= \frac{r_2}{2} \left(\frac{u(i+1, j+1) + u(i, j+1)}{2} - \frac{u(i+1, j-1) + u(i, j-1)}{2} \right) \\ &\quad + r_1 |\mathbf{p}^{k-1}| \left[\mathcal{A}_{i,j}^{\circ}(n_2) - \mathcal{A}_{i,j}^{\circ}(\lambda_{12}) - \mathcal{A}_{i,j}^{\circ}(\lambda_{22}) \right]. \end{aligned}$$

Similarly, a discretization of c and $\mathbf{q}$ at $(i, j) \in \square$ -node is obtained as follows:

$$\begin{aligned} c(i, j) &= a + b \left(\frac{h(i, j+1) + h(i, j)}{2} \right)^2 - \mathcal{A}_{i,j}^\square(\lambda_{11}) \mathcal{A}_{i,j}^\square(n_1) - \lambda_{12}(i, j) n_2(i, j), \\ q_1(i, j) &= \frac{r_2}{2} \left(\frac{u(i+1, j+1) + u(i+1, j)}{2} - \frac{u(i-1, j+1) + u(i-1, j)}{2} \right) \\ &\quad + r_1 |\mathbf{p}|^{k-1} |\mathcal{A}_{i,j}^\square(n_1) - \mathcal{A}_{i,j}^\square(\lambda_{11}) - \mathcal{A}_{i,j}^\square(\lambda_{21})|, \\ q_2(i, j) &= r_2 \left(u(i, j+1) - u(i, j) \right) + r_1 |\mathbf{p}|^{k-1} |n_2(i, j) - \lambda_{12}(i, j) - \lambda_{22}(i, j)|. \end{aligned}$$

The $\mathbf{n}$ -sub Problem

For the $\mathbf{n}$ -sub problem, the Euler-Lagrange equation for the energy (7.20) is the following linear coupled PDEs

$$-r_3 \nabla(\nabla \cdot \mathbf{n}) + r_1 |\mathbf{p}|^2 \mathbf{n} = r_1 \mathbf{p} |\mathbf{p}| + \boldsymbol{\lambda}_1 |\mathbf{p}| - r_3 \nabla h - \nabla \lambda_3.$$

Pay attention that the operator $\nabla(\nabla \cdot)$ is singular. Since it is possible that $|\mathbf{p}| = 0$, we add a quadratic penalty term to (7.20) to avoid the singularity. Therefore, we rewrite $\mathcal{E}_3(\mathbf{n})$ as follows

$$\begin{aligned} \mathcal{E}_3(\mathbf{n}) &= - \int_{\Omega} \boldsymbol{\lambda}_1 \cdot |\mathbf{p}| \mathbf{n} d\Omega + \frac{r_1}{2} \int_{\Omega} (\mathbf{p} - |\mathbf{p}| \mathbf{n})^2 d\Omega - \int_{\Omega} \lambda_3 \nabla \cdot \mathbf{n} d\Omega \\ &\quad + \frac{r_3}{2} \int_{\Omega} (h - \nabla \cdot \mathbf{n})^2 d\Omega + \frac{\gamma}{2} \int_{\Omega} (\mathbf{n} - \mathbf{n}^{k-1})^2 d\Omega. \end{aligned} \quad (7.33)$$

where γ is a positive constant.

We have the following optimality condition for the $\mathbf{n}$ -sub problem (7.33) by its Euler-Lagrange equation

$$\left(\gamma + r_1 |\mathbf{p}|^2 - r_3 \nabla \operatorname{div} \right) \mathbf{n} = \gamma \mathbf{n}^{k-1} + r_1 \mathbf{p} |\mathbf{p}| + \boldsymbol{\lambda}_1 |\mathbf{p}| - r_3 \nabla h - \nabla \lambda_3. \quad (7.34)$$

The equation (7.34) is coupled PDEs of the variable $\mathbf{n} = (n_1, n_2)$. When we compute the component n_1 , we use the n_2 in previous iteration

$$(\gamma + r_1 |\mathbf{p}|^2 - r_3 \partial_x^2) n_1 = \gamma n_1^{k-1} + r_3 \partial_x \partial_y n_2 + r_1 p_1 |\mathbf{p}| + \lambda_{11} |\mathbf{p}| - r_3 \partial_x h - \partial_x \lambda_3,$$

and vice versa, when solve n_2 , we use the n_1 in previous iteration

$$(\gamma + r_1|\mathbf{p}|^2 - r_3\partial_y^2)n_2 = \gamma n_2^{k-1} + r_3\partial_y\partial_x n_1 + r_1 p_2|\mathbf{p}| + \lambda_{12}|\mathbf{p}| - r_3\partial_y h - \partial_y \lambda_3.$$

Similarly to the u -sub problem, the above linear PDEs can be efficiently solved by the linear iterative methods. We still apply the one sweep Gauss-Seidel iteration to solve the $\mathbf{n}$ -sub problem in our work.

The h -sub Problem

For the h -sub problem, the Euler-Lagrange equation of the functional (7.21) gives

$$(2b|\mathbf{p}| + r_3)h = r_3\nabla \cdot \mathbf{n} - \lambda_3. \quad (7.35)$$

There is the closed form solution for h , which is

$$h(i, j) = \frac{r_3 \text{div}_{i,j}^\bullet(\mathbf{n}) - \lambda_3(i, j)}{2b|\mathcal{A}_{i,j}^\bullet(\mathbf{p}) + r_3}.$$

Update Lagrangian multipliers

We update Lagrangian Multipliers λ_1 , λ_2 , λ_3 and λ_4 using the staggered grid as shown Figure 7.1. The discretized forms of equations are written as:

$$\begin{aligned} \lambda_{11}^k &= \lambda_{11}^{k-1} + r_1(p_1^k - |\mathbf{p}^k|n_1^k) && \text{at } \circ \text{-nodes} \\ \lambda_{12}^k &= \lambda_{12}^{k-1} + r_1(p_2^k - |\mathbf{p}^k|n_2^k) && \text{at } \square \text{-nodes} \\ \lambda_{21}^k &= \lambda_{21}^{k-1} + r_2(p_1^k - \partial_1^+ u^k) && \text{at } \circ \text{-nodes} \\ \lambda_{22}^k &= \lambda_{22}^{k-1} + r_2(p_2^k - \partial_2^+ u^k) && \text{at } \square \text{-nodes} \\ \lambda_3^k &= \lambda_3^{k-1} + r_3(h^k - \text{div}_{i,j}^\bullet(\mathbf{n})) && \text{at } \bullet \text{-nodes} \\ \lambda_4^k &= \lambda_4^{k-1} + r_4(v^k - u^k) && \text{at } \bullet \text{-nodes} \end{aligned}$$

We summarize the solutions to subproblems of both the existing algorithm in [134] and the proposed algorithm in Table 7.1.

Table 7.1: Summary of methods for solving subproblems between the existing and proposed proposed algorithms.

Subproblem	Our proposed algorithm	The existing algorithm in [134]
v-sub problem	closed-form solution	closed-form solution
u-sub problem	GS of linear PDE	FFT of linear PDE
p -sub problem	closed-form solution	closed-form solution
n -sub problem	GS of coupled linear PDEs	FFT of coupled linear PDEs
h/m-sub problem	first order equation	closed-form solution

7.6 Numerical Examples

In this section, we present the numerical results to demonstrate the efficiency of the proposed algorithm. We explore the numerical results of the proposed algorithm for image denoising applications. We consider denoising examples with Gaussian white noise and Salt & pepper noise. In all the experiments, we set $\gamma = 0.01$. The system is a 2.4 GHz CPU and 4GB memory.

During the iterations, we define the relative error of the solution $\{u^k | k = 1, 2, \dots\}$ and we stop the iteration when the relative error is less than the given error tolerance, that is

$$\frac{\sum_{i,j} |u^k(i,j) - u^{k-1}(i,j)|}{\sum_{i,j} |u^{k-1}|} \leq \epsilon, \quad (7.36)$$

We monitor the relative residuals of variables in (7.14), which are defined as

$$\begin{aligned} R_1^k &= \frac{\sum_{i,j} |\mathbf{p}^k(i,j) - |\mathbf{p}^k(i,j)|\mathbf{n}^k(i,j)|}{|\Omega|}; & R_2^k &= \frac{\sum_{i,j} |\mathbf{p}^k(i,j) - \nabla u^k(i,j)|}{|\Omega|}; \\ R_3^k &= \frac{\sum_{i,j} |h(i,j) - \nabla \cdot \mathbf{n}(i,j)|}{|\Omega|}; & R_4^k &= \frac{\sum_{i,j} |v(i,j) - u(i,j)|}{|\Omega|}. \end{aligned} \quad (7.37)$$

Table 7.2: The computational time (sec) of our proposed algorithm is compared with the method (THC) in [134] for the results in Figure 7.2. The computational time is measured in seconds.

Images	Size	Our proposed algorithm			The existing algorithm (THC)		
		SNR	Iteration	Time	SNR	Iteration	Time
Test	60×60	25.55	138	1.45	25.35	102	24.17
Shapes	100×100	26.17	110	3.02	26.32	135	94.77

Correspondingly, the relative errors of the Lagrange multipliers are obtained by

$$\begin{aligned}
 L_1^k &= \frac{\sum_{i,j} |\lambda_1^k(i,j) - \lambda_1^{k-1}(i,j)|}{\sum_{i,j} |\lambda_1^{k-1}(i,j)|}; L_2^k = \frac{\sum_{i,j} |\lambda_2^k(i,j) - \lambda_2^{k-1}(i,j)|}{\sum_{i,j} |\lambda_2^{k-1}(i,j)|}; \\
 L_3^k &= \frac{\sum_{i,j} |\lambda_3^k(i,j) - \lambda_3^{k-1}(i,j)|}{\sum_{i,j} |\lambda_3^{k-1}(i,j)|}; L_4^k = \frac{\sum_{i,j} |\lambda_4^k(i,j) - \lambda_4^{k-1}(i,j)|}{\sum_{i,j} |\lambda_4^{k-1}(i,j)|}.
 \end{aligned} \tag{7.38}$$

In addition, we define the numerical energy of the Euler's elastica model as follows

$$E^k = \sum_{i,j} \left(a + b(h^k(i,j))^2 \right) |\mathbf{p}^k(i,j)| + \frac{\mu}{2} \sum_{i,j} \left(v^k(i,j) - f(i,j) \right)^2. \tag{7.39}$$

In the first place, we compare the proposed algorithm with the augmented Lagrangian method proposed (Tai-Hahn-Chung (THC)) in [134]. Our algorithm is implemented in C++ and the algorithm THC is implemented in Matlab environment. For a fair comparison, we set $a = 1$, $b = 10$, $\mu = 100$ and $r_1 = 2$ for both algorithms and stop the iteration if the relative error of the solution $\{u^k | k = 1, 2, \dots\}$ satisfies the same error tolerance. We choose two test images: Test (Row one) and Shapes (Row two) and add Gaussian white noise with mean zero and standard deviation 10 to them. The results from the elastica energy (3.14) are shown in Figure 7.2. We use $\epsilon = 4 \times 10^{-4}$ for example Test and $\epsilon = 2 \times 10^{-4}$ for example Shapes. We display the computational time, SNR and iteration number in the experiment in Table 7.2.

In Table 7.2, it shows that our algorithm may require more iterations to satisfy the

Figure 7.2: Denoising comparison results. For Test (Row one), we set $r_2 = 500$, $r_3 = 200$ in proposed algorithm and $r_2 = 2$, $r_3 = 600$ in THC. For Shapes (Row two), we set $r_2 = 500$, $r_3 = 10$ in proposed algorithm and $r_2 = 50$, $r_3 = 100$ in THC.

error tolerance than the algorithm THC, like the example Test (Row one). However, the sub problem of the variable n requires inner iterations for the convergence in the algorithm THC. Therefore, our algorithm can still save computational cost in these situations. In Figure 7.3, plots of relative error, numerical energy and residuals versus iteration numbers are shown for Test and Shapes in Table 7.2. From this test, we can see that the relative error in u^k (7.36), numerical energy (7.39) and residuals (7.37) of the proposed algorithm converge more stable compared with the algorithm THC.

In Figure 7.4 and Figure 7.5, we show the denoising results of our proposed algorithm for synthetic images and real images, respectively. We keep to add the Gaussian white noise with mean zero and the standard deviation 10 to the test images. In the figures, the noisy images are shown in the left column and the restored images are shown in the right column. In Figure 7.4 and 7.5, we use the L^2 data fitting term, $t = 2$, to recover the images corrupted by Gaussian noise. We summarize the size of images, SNR, number of outer iteration k and computational time in Table 7.3 for

the test images in Figure 7.4 and 7.5.

Figure 7.3: Plots of (7.36), (7.39) and (7.37) values versus iteration numbers for examples shown in Figure 7.2. In each pair, the left one is from our proposed algorithm and the right one is from THC.

In Figure 7.6, we display the plots of the relative error in u^k (7.36), numerical energy (7.39), relative residuals (7.37), relative errors of Lagrange multipliers (7.38) and SNR (6.35) versus the outer iterations k . Here, all the graphs in Figure 7.6 are plotted up to 10^3 iterations.

In Figure 7.7, we show the numerical results for image denoising applications with images corrupted by the salt & pepper noise. We add the salt & pepper noise with a noise density 0.4 to all the test images. For this case, we use the L^1 data fitting term, $t = 1$, in the functional (3.14). The size of images, SNR, number of outer iteration k and computational time are listed in Table 7.4 for the test images in Figure 7.7.

Figure 7.4: Euler's elastica based image denoising for Gaussian noise. We set $a = 1$, $b = 2$, $\mu = 100$, $r_1 = 2$, $r_2 = 200$ and $r_3 = 20$. The tolerance is $1.2 \cdot 10^{-4}$ for (a), $1 \cdot 10^{-4}$ for (b), (c) and 3.5×10^{-4} for (d).

Figure 7.5: Euler's elastica based image denoising for Gaussian noise. We set $a = 1$, $b = 2$, $\mu = 180$, $r_1 = 2$, $r_2 = 200$ and $r_3 = 20$. The tolerance is $3.5 \cdot 10^{-4}$.

Table 7.3: The size of images and the SNR for images in Figure 7.4 and 7.5 are shown. The computational time is measured in seconds.

Images	Size	SNR	# of iteration	Time (sec)
Figure 7.4(a)	60×60	21.54	92	0.99
Figure 7.4(b)	100×100	26.20	104	2.92
Figure 7.4(c)	100×100	23.12	83	2.33
Figure 7.4(d)	128×128	25.59	93	4.27
Figure 7.5(a)	256×256	18.81	67	12.10
Figure 7.5(b)	332×216	17.40	93	18.14

(a) Example in Figure 7.4 (d)

(b) Example in Figure 7.5 (a)

Figure 7.6: Plots of (7.36), (7.39), (7.37) and (7.38) values versus iteration numbers for examples shown in Figure 7.4 (d) and Figure 7.5 (a).

(a)

(b)

Figure 7.7: Euler's elastica based image denoising for salt & pepper noise. We set $a = 5$, $b = 20$, $\mu = 5$, $r_1 = 1$, $r_2 = 10$, $r_3 = 1$ and $r_4 = 20$ for both (a) and (b). The tolerance is $1.7 \cdot 10^{-3}$ for (a) and $1.8 \cdot 10^{-3}$ for (b).

Table 7.4: The size of images and the SNR for images in Figure 7.7 are shown. The computational time is measured in seconds.

Images	Size	SNR	# of iteration	Time (sec)
Figure 7.7(a)	420×400	19.53	50	51.87
Figure 7.7(b)	700×512	19.51	54	26.16

CHAPTER 8

Image Restoration and Segmentation on Triangulated Surfaces

Contents

8.1	Introduction	164
8.2	Notation	165
8.3	TV based Restoration and Segmentation on Triangulated Surfaces	166
8.4	ALM for TV Restoration on Triangulated Surfaces . . .	171
8.5	ALM for Vectorial TV Restoration on Triangulated Surfaces	174
8.6	ALM for Multi-region Segmentation on Triangulated Surfaces	177
8.7	Numerical Examples	182

We have discussed the augmented Lagrangian formulation of image denoising and segmentation models for 2D planar images. In this chapter, we extend ALM to image restoration and segmentation on triangulated surface. In view of independence of the contents in this chapter, we use a brand new system of notations, which maybe overlap with previous chapters.

8.1 Introduction

Since the pioneering work of Rudin, Osher, and Fatemi [3], TV [3] and vectorial TV models [25, 22, 41, 24] have been demonstrated very successful in image restoration. The success of TV and vectorial TV relies on its good edge preserving property, which suits most images where the edges are sparse. For years, these restoration models are usually solved by gradient descent method, which is quite slow due to the non-smoothness of the objective functionals. Recently, various convex optimization techniques have been proposed to efficiently solve these models [11, 137, 12, 21, 14, 18, 13, 19, 15, 16, 17, 56, 122, 138], some of which are closely related to iterative shrinkage-thresholding algorithms [126, 139, 140, 141].

In addition to image restoration, TV, or vectorial TV, plays an important role in convexifying variational image segmentation models. Although classical variational segmentation models and their gradient descent minimization methods have had a great success, such as snakes [142], geodesic active contours [143, 144], the Chan-Vese method [41] and level set [145] or piecewise constant level set [106] approximation and implementations [42, 17, 146, 44] of the Mumford-Shah model [40], they suffer from the local minima problem due to the non-convexity of the energy functionals. Very recently, various convexification techniques have been proposed to reformulate non-convex segmentation models to convex ones [147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159] and even more general extensions [160, 161], yielding fast and global minimization algorithms. These convex reformulations and extensions are all based on TV or vectorial TV regularizers.

Due to these successes, TV based restoration and segmentation models have been extended to data processing on triangulated manifolds [162, 163, 164] via gradient descent method. When this work was nearly finished, we got to know the technical report [135] released, which is, to the best of our knowledge, the only paper discussing fast convex optimization techniques for these problems. The authors proposed to use split Bregman iteration for TV denoising of gray images and Chan-Vese segmentation (which is a single-phase model). The split Bregman iteration was first introduced in [15] and then was found to be equivalent to the augmented Lagrangian method; see

e.g., [19, 16, 17] and references therein. In this chapter, we would like to present this approach for vectorial TV based image restoration and segmentation problems on triangular mesh surfaces by using the language of augmented Lagrangian method. For the sakes of completeness and readability, we will include the details of both TV and vectorial TV based restorations and segmentations applied to gray and color images, although the former has been discussed in [135] in terms of split Bregman iteration. In particular, we will first give several Hilbert spaces to describe TV and vectorial TV based variational models in the discrete setting. Secondly, ALM applied to TV and vectorial TV image restoration on mesh surfaces will be presented, leading to efficient algorithms for both gray and color image restoration. Thirdly, we discuss ALM for vectorial TV based multi-region (multi-phase) image segmentation, which also works for both gray and color images. For each problem, our algorithms benefit from fast solvers for sparse linear systems and closed form solutions to sub-problems.

The chapter is organized as follows. In section 2, we introduce some notation. In section 3, we first define several Hilbert spaces with differential mappings, and then present the TV and vectorial TV based restoration and segmentation models on triangular mesh surfaces. ALM for TV restoration of gray images, vectorial TV restoration of color images, and multi-region segmentation will be discussed in section 4, 5, 6, respectively. In section 7, we give some numerical experiments.

8.2 Notation

Assume M is a triangulated surface of arbitrary topology in $\mathbb{R}^3$ with no degenerate triangles. The set of vertices, the set of edges, and the set of face triangles of M are denoted as $\{v_i : i = 0, 1, \dots, N_v - 1\}$, $\{e_i : i = 0, 1, \dots, N_e - 1\}$, and $\{\tau_i : i = 0, 1, \dots, N_\tau - 1\}$, respectively, where N_v , N_e , and N_τ are the numbers of vertices, edges, and triangles. We explicitly denote an edge e whose endpoints are v_i, v_j as $[v_i, v_j]$. Similarly a triangle τ whose vertices are v_i, v_j, v_k is denoted as $[v_i, v_j, v_k]$. If v is an endpoint of an edge e , then we denote it as $v \prec e$. Similarly, e is an edge of a triangle τ is denoted as $e \prec \tau$; v is a vertex of a triangle τ is denoted as $v \prec \tau$ (see [165]). We denote the barycenters of triangle τ , edge e , and vertex v by $BC(\tau)$,

$BC(e)$, and $BC(v)$, respectively. Let $N_1(i)$ be the 1-neighborhood of vertex v_i . It is the set of indices of vertices that are connected to v_i . Let $D_1(i)$ be the 1-disk of the vertex v_i . $D_1(i)$ is the set of triangles with v_i being one of their vertices. It should be pointed out that the 1-disk of a boundary vertex is topologically just a half disk.

For each vertex v_i , we define a piecewise linear basis ϕ_i such that $\phi_i(v_j) = \delta_{ij}$, $i, j = 0, 1, \dots, N_v - 1$, where δ_{ij} is the Kronecker delta. $\{\phi_i : i = 0, 1, \dots, N_v - 1\}$ has the following properties:

1. local support: $\text{supp}\phi_i = D_1(i)$;
2. nonnegativity: $\phi_i \geq 0, i = 0, 1, \dots, N_v - 1$;
3. partition of unity: $\sum_{0 \leq i \leq N_v - 1} \phi_i \equiv 1$.

Most data on M are given by assigning values at the vertices of M . We can use $\{\phi_i : i = 0, 1, \dots, N_v - 1\}$ to build piecewise linear functions on M for these data. Suppose u reaches value u_i at vertex v_i , $i = 0, 1, \dots, N_v - 1$. Then $u(x) = \sum_{0 \leq i \leq N_v - 1} u_i \phi_i(x)$ for any $x \in M$. Similarly, piecewise linear vector-valued functions $(u_1(x), u_2(x), \dots, u_M(x))$ on M can be defined. In some applications, we also have piecewise constant function (vector) over M , that is, a single value (vector) is assigned to each triangle of M . See [163] and the references therein.

8.3 TV based Restoration and Segmentation on Triangulated Surfaces

In this section, we first define two basic Hilbert spaces on triangular mesh surfaces and differential mappings between them, and then present several TV and vectorial TV based image restoration and segmentation models.

8.3.1 Basic spaces and differential mappings

Given a triangular mesh surface M , we now define two Hilbert spaces. The linear space $V_M = \mathbb{R}^{N_v}$ is a set, whose elements are given by values at the vertices of M . For

instance, $u = (u_0, u_1, \dots, u_{N_v-1}) \in V_M$ means that u takes value u_i at vertex v_i for all i . By the basis $\{\phi_i : i = 0, 1, \dots, N_v - 1\}$, V_M is isomorphic to the space of all piecewise linear functions on M . We denote, by Q_M , the set of piecewise constant vector fields on M , i.e., $Q_M = \prod_{\tau \subset M} T\tau$, where $T\tau$ is the tangent vector space of the triangle τ . Note $Q_M \subset \mathbb{R}^{3 \times N_\tau}$. For $p \in Q_M$, p restricted in τ is a constant 3-dimensional vector lying on the tangent space of τ ; see [163].

We can equip the spaces V_M and Q_M with inner products and norms. Motivated by Eqn. (4.2) in [163], we define, for $u^1, u^2, u \in V_M$,

$$((u^1, u^2))_{V_M} = \sum_{0 \leq i \leq N_v-1} u_i^1 u_i^2 s_i, \quad \|u\|_{V_M} = \sqrt{((u, u))_{V_M}}, \quad (8.1)$$

where $s_i = \sum_{\tau \in D_1(i)} \frac{1}{3} s_\tau$ with s_τ as the area of the triangle τ . s_i is actually the area of the control cell of the vertex v_i based on barycentric dual mesh; see [162, 163] and references therein. For $p^1, p^2, p \in Q_M$, we have

$$((p^1, p^2))_{Q_M} = \sum_{\tau} (p_\tau^1, p_\tau^2) s_\tau, \quad \|p\|_{Q_M} = \sqrt{((p, p))_{Q_M}}, \quad (8.2)$$

where (p_τ^1, p_τ^2) is the conventional inner product in $\mathbb{R}^3$ (with induced norm denoted by $|p_\tau|$) and s_τ is the area of the triangle τ . We mention that, throughout this chapter, $((,))$ and $\| \cdot \|$ are used to denote inner products and norms of data defined on the whole surface, while $(,)$ and $\| \cdot \|$ are used for those of data restricted on vertices or triangles.

The gradient operator on M is a mapping $\nabla_M : V_M \rightarrow Q_M$. Given $u \in V_M$, which corresponds to a piecewise linear function on M , its gradient is as follows

$$\nabla_M u = \sum_{0 \leq i \leq N_v-1} u_i \nabla_M \phi_i, \quad (8.3)$$

where $\nabla_M \phi_i$ is a piecewise constant vector field on M [163].

Using the above inner products in V_M and Q_M , it is easy to derive the adjoint operator of $-\nabla_M$, say, the divergence operator $\text{div} : Q_M \rightarrow V_M$. For $p \in Q_M$, $\text{div} p$ is

given by

$$(\operatorname{div} p)_{v_i} = -\frac{1}{s_i} \sum_{\tau, v_i \prec \tau} s_\tau (p_\tau, (\nabla_M \phi_i)_\tau), \quad \forall i. \quad (8.4)$$

Note this is a special case of the divergence formulae in [163] where $X \in Q_M$.

8.3.2 TV restoration model (ROF) on triangulated surfaces

Assume $f \in V_M$ is an observed noisy gray (single-valued) image defined on M . The total variation restoration model (ROF) aims at solving

$$\min_{u \in V_M} \left\{ E_{\text{tv}}(u) = R_{\text{tv}}(\nabla_M u) + \frac{\mu}{2} \|u - f\|_{V_M}^2 \right\}, \quad (8.5)$$

where

$$R_{\text{tv}}(\nabla_M u) = \sum_{\tau} |(\nabla_M u)_\tau| s_\tau,$$

is the total variation of u , and $\mu > 0$ is a fidelity parameter. Under the assumption of no degenerate triangles on M , the objective functional in (8.5) is coercive, proper, continuous, and strictly convex. Hence, the restoration problem (8.5) has a unique solution.

8.3.3 Vectorial TV restoration model on triangulated surfaces

The TV model (8.5) can be extended to vectorial TV model for color (multi-valued) image restoration. For the convenience of description, we introduce the following notation (see [17] for similar notation extensions in 2D planar domain)

$$\begin{aligned} \mathbf{V}_M &= \underbrace{V_M \times V_M \times \cdots \times V_M}_{\mathfrak{M}}, \\ \mathbf{Q}_M &= \underbrace{Q_M \times Q_M \times \cdots \times Q_M}_{\mathfrak{M}}. \end{aligned}$$

Hence an $\mathfrak{M}$ -channel image $\mathbf{u} = (u_1, u_2, \dots, u_{\mathfrak{M}})$ is an element of $\mathbf{V}_M$, and its gradient $\nabla_M \mathbf{u} = (\nabla_M u_1, \nabla_M u_2, \dots, \nabla_M u_{\mathfrak{M}})$ is an element of $\mathbf{Q}_M$. The inner products

and norms in $\mathbf{V}_M$ and $\mathbf{Q}_M$ are as follows

$$\begin{aligned} ((\mathbf{u}^1, \mathbf{u}^2))_{\mathbf{V}_M} &= \sum_{1 \leq \mathbf{m} \leq \mathfrak{M}} ((u_{\mathbf{m}}^1, u_{\mathbf{m}}^2))_{V_M}, \quad \|\mathbf{u}\|_{\mathbf{V}_M} = \sqrt{((\mathbf{u}, \mathbf{u}))_{\mathbf{V}_M}}; \\ ((\mathbf{p}^1, \mathbf{p}^2))_{\mathbf{Q}_M} &= \sum_{1 \leq \mathbf{m} \leq \mathfrak{M}} ((p_{\mathbf{m}}^1, p_{\mathbf{m}}^2))_{Q_M}, \quad \|\mathbf{p}\|_{\mathbf{Q}_M} = \sqrt{((\mathbf{p}, \mathbf{p}))_{\mathbf{Q}_M}}. \end{aligned}$$

When restricting at a vertex v_i (and a triangle τ), we can also define the following vertex-by-vertex (and triangle-by-triangle) inner products and norms

$$\begin{aligned} (\mathbf{u}_i^1, \mathbf{u}_i^2) &= \sum_{1 \leq \mathbf{m} \leq \mathfrak{M}} (u_{\mathbf{m}}^1)_i (u_{\mathbf{m}}^2)_i, \quad |\mathbf{u}_i| = \sqrt{(\mathbf{u}_i, \mathbf{u}_i)}; \\ (\mathbf{p}_\tau^1, \mathbf{p}_\tau^2) &= \sum_{1 \leq \mathbf{m} \leq \mathfrak{M}} ((p_{\mathbf{m}}^1)_\tau, (p_{\mathbf{m}}^2)_\tau), \quad |\mathbf{p}_\tau| = \sqrt{(\mathbf{p}_\tau, \mathbf{p}_\tau)}. \end{aligned}$$

We consider the following vector-valued image restoration problem

$$\min_{\mathbf{u} \in \mathbf{V}_M} \left\{ E_{\text{vtr}}(\mathbf{u}) = R_{\text{vtr}}(\nabla_M \mathbf{u}) + \frac{\mu}{2} \|\mathbf{u} - \mathbf{f}\|_{\mathbf{V}_M}^2 \right\}, \quad (8.6)$$

where

$$R_{\text{vtr}}(\nabla_M \mathbf{u}) = \text{TV}(\mathbf{u}) = \sum_{\tau} \sqrt{\sum_{1 \leq \mathbf{m} \leq \mathfrak{M}} |(\nabla_M u_{\mathbf{m}})_\tau|^2 s_\tau}. \quad (8.7)$$

is the vectorial TV semi-norm (see [25, 149] for the representation in 2D image domains), and $\mathbf{f} = (f_1, f_2, \dots, f_{\mathfrak{M}}) \in \mathbf{V}_M$ is an observed noisy image. Similarly, the restoration problem (8.6) has a unique minimizer under the assumption of no degenerate triangles on M .

Remark. More general forms of the restoration models (8.5) and (8.6) can be considered, which contain a blur kernel K in the fidelity terms, as in the models in 2D planar image restoration. However, we mention that on general surfaces the blur theory has not been established yet. In non-flat spaces, a blur operator cannot be described as a simple convolution. Algorithms for deblurring on surfaces may contain many complicated calculations.

8.3.4 Multi-region segmentation on triangulated surfaces

In this subsection, we present the multi-region segmentation model based on vectorial TV regularization. This model has been proposed in 2D planar image segmentation recently in [150, 154] and extended to data segmentation on triangulated manifolds in [164].

We suppose that the $\mathfrak{N}$ -channel (either single or multiple channels) data $\mathbf{d}$ on M are to be partitioned to $\mathfrak{M}$ segments. Note that here the number of channels $\mathfrak{N}$ and that of segments $\mathfrak{M}$ are totally independent of each other. In most image data, the segments are assumed to be piecewise constant (for clean data) or approximately piecewise constant (for noisy data). We denote the mean values of the data on these segments as $\varsigma = (\varsigma_1, \dots, \varsigma_{\mathfrak{m}}, \dots, \varsigma_{\mathfrak{M}})$ (each $\varsigma_{\mathfrak{m}}$ is an $\mathfrak{N}$ -dimensional vector). In this chapter we assume ς is given, since ς can be estimated in advance in many applications.

In multi-region segmentation, we are to solve

$$\min_{\mathbf{u} \in C} \left\{ G_{\text{mrs}}(\mathbf{u}) = ((\mathbf{u}, \mathbf{s}))_{\mathbf{V}_M} + \beta R_{\text{mrs}}(\nabla_M \mathbf{u}) \right\}, \quad (8.8)$$

where $R_{\text{mrs}}(\nabla_M \mathbf{u}) = \text{TV}(\mathbf{u})$; $C = \{\mathbf{u} = (u_1, \dots, u_{\mathfrak{m}}, \dots, u_{\mathfrak{M}}) \in \mathbf{V}_M : (u_{\mathfrak{m}})_i \geq 0, \sum_{1 \leq \mathfrak{m} \leq \mathfrak{M}} (u_{\mathfrak{m}})_i = 1, \forall i\}$ is the constrained set; $\mathbf{s} = (s_1, \dots, s_{\mathfrak{m}}, \dots, s_{\mathfrak{M}}) \in \mathbf{V}_M$ and $(s_{\mathfrak{m}})_i$ indicates the affinity of the data at vertex v_i with class $\mathfrak{m}$; β is a positive parameter. A natural expression of $(s_{\mathfrak{m}})_i$ is the squared error between the data $\mathbf{d}$ at vertex v_i and the mean value $\varsigma_{\mathfrak{m}}$

$$(s_{\mathfrak{m}})_i = (\mathbf{d}_i - \varsigma_{\mathfrak{m}}, \mathbf{d}_i - \varsigma_{\mathfrak{m}}), \quad (8.9)$$

which will be considered in this chapter. We mention that other affinities with prescribed probability density functions can be similarly treated.

Since $G_{\text{mrs}}(\mathbf{u})$ is continuous and convex and C is convex and closed, the minimization problem (8.8) has at least one solution.

Remark. The model (8.8) is a convex relaxation [147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 160, 161] of the original multi-region segmentation (labeling) problem. The solution is thus not exactly “binary”. A “binarization step” is needed

to convert the solution of (8.8) to class numbers. In this chapter we use the improved binarization technique proposed in [161].

Remark. In the case that the mean values of the segments are unknown and difficult to estimate in advance, we need to solve the following minimization problem

$$\min_{\mathbf{u} \in C, \varsigma} \left\{ G_{\text{mrs}}(\mathbf{u}, \varsigma) = ((\mathbf{u}, \mathbf{s}(\varsigma)))_{\mathbf{V}_M} + \beta R_{\text{mrs}}(\nabla_M \mathbf{u}) \right\}, \quad (8.10)$$

where $R_{\text{mrs}}(\nabla_M \mathbf{u})$, C , β are the same as defined above; and $\mathbf{s}(\varsigma)$ is also defined by (8.9) except that here ς is unknown and thus written as a variable of $\mathbf{s}$. We mention that the problem (8.10) is non-convex and thus has local minima.

8.4 ALM for TV Restoration on Triangulated Surfaces

Due to the nonlinearity and non-differentiability of the TV model (8.5), the gradient descent method implemented in [162] is slow. Recently the augmented Lagrangian method has been proven to be very efficient for TV restoration in planar image processing [16, 17, 56]. In the following, we present the method on triangular mesh surfaces.

We first introduce a new variable $p \in Q_M$ and reformulate the problem (8.5) to be the following equality constrained problem

$$\begin{aligned} \min_{u \in V_M, p \in Q_M} & \left\{ R_{\text{tv}}(p) + \frac{\mu}{2} \|u - f\|_{V_M}^2 \right\}, \\ \text{s.t.,} & \quad p = \nabla_M u. \end{aligned} \quad (8.11)$$

Problems (8.5) and (8.11) are clearly equivalent to each other.

To solve (8.11), we define the augmented Lagrangian functional

$$\mathcal{L}_{\text{tv}}(u, p; \lambda) = R_{\text{tv}}(p) + \frac{\mu}{2} \|u - f\|_{V_M}^2 + ((\lambda, p - \nabla_M u))_{Q_M} + \frac{r}{2} \|p - \nabla_M u\|_{Q_M}^2, \quad (8.12)$$

with Lagrange multiplier $\lambda \in Q_M$ and positive constant r , and then consider the

following saddle-point problem

$$\begin{aligned} \text{Find } & (u^*, p^*; \lambda^*) \in V_M \times Q_M \times Q_M, \\ \text{s.t., } & \mathcal{L}_{\text{tv}}(u^*, p^*; \lambda) \leq \mathcal{L}_{\text{tv}}(u^*, p^*; \lambda^*) \leq \mathcal{L}_{\text{tv}}(u, p; \lambda^*), \forall (u, p; \lambda) \in V_M \times Q_M \times Q_M. \end{aligned} \quad (8.13)$$

Similar to [17] and references therein, it can be shown that the saddle-point problem (8.13) has at least one solution and all the saddle-points $(u^*, p^*; \lambda^*)$ s have the same u^* , which is the unique solution of the original problem (8.5).

Algorithm 18 Augmented Lagrangian method for the TV restoration model

1. *Initialization:* $\lambda^0 = 0, u^{-1} = 0, p^{-1} = 0$;
2. *For* $k=0, 1, 2, \dots$: compute (u^k, p^k) as a (approximate) minimizer of the augmented Lagrangian functional with the Lagrange multiplier λ^k , *i.e.*,

$$(u^k, p^k) \approx \arg \min_{(u, p) \in V_M \times Q_M} \mathcal{L}_{\text{tv}}(u, p; \lambda^k), \quad (8.14)$$

where $\mathcal{L}_{\text{tv}}(u, p; \lambda^k)$ is as defined in (8.12); and update

$$\lambda^{k+1} = \lambda^k + r(p^k - \nabla_M u^k). \quad (8.15)$$

We now present an iterative algorithm to solve the saddle-point problem. See Algorithm 18. We are now left with the minimization problem (8.14) to address. Since u, p are coupled together, it is in general difficult to compute the minimizers u^k and p^k simultaneously. Instead one usually separates the variables u and p and then applies an alternative minimization procedure [14, 15, 16, 17], through which one can only obtain the minimizer approximately in practice (as an inner iteration of the whole algorithm, this alternative minimization procedure cannot run to infinite steps); see the symbol $\approx$ in (8.14). Fortunately, we do not need an exact solution to (8.14) to get the solution to the saddle-point problem.

We separate (8.14) into the following two sub-problems:

- The u -sub problem: for a given p ,

$$\min_{u \in V_M} \left\{ \frac{\mu}{2} \|u - f\|_{V_M}^2 - ((\lambda^k, \nabla_M u))_{Q_M} + \frac{r}{2} \|p - \nabla_M u\|_{Q_M}^2 \right\}. \quad (8.16)$$

- The p -sub problem: for a given u ,

$$\min_{p \in Q_M} \left\{ R_{\text{tv}}(p) + ((\lambda^k, p))_{Q_M} + \frac{r}{2} \|p - \nabla_M u\|_{Q_M}^2 \right\}. \quad (8.17)$$

These two sub-problems can be efficiently solved.

8.4.1 Solving the u -sub problem

The u -sub problem (8.16) is a quadratic programming, whose optimality condition is as follows

$$\mu(u - f) + \text{div}_M \lambda^k + r \text{div}_M p - r \text{div}_M \nabla_M u = 0, \quad (8.18)$$

which is a sparse linear system and can be solved by various well-developed numerical packages.

8.4.2 Solving the p -sub problem

The p -sub problem is decomposable and thus can be solved triangle-by-triangle. For each triangle τ , we need to solve

$$\min_{p_\tau} \left\{ |p_\tau| + (\lambda_\tau^k, p_\tau) + \frac{r}{2} |p_\tau - (\nabla_M u)_\tau|^2 \right\}. \quad (8.19)$$

By a similar geometric argumentation as in [56], (8.19) has the following closed form solution

$$p_\tau = \begin{cases} (1 - \frac{1}{r} \frac{1}{|w_\tau|}) w_\tau, & |w_\tau| > \frac{1}{r}, \\ 0, & |w_\tau| \leq \frac{1}{r}, \end{cases} \quad (8.20)$$

where

$$w = \nabla_M u - \frac{\lambda^k}{r}. \quad (8.21)$$

It can be seen that p_τ obtained via (8.20) is in the tangent vector space $T\tau$ and thus $p \in Q_M$, as long as $\lambda_\tau^k \in T\tau, \forall \tau$, which is a natural consequence from the viewpoint of Hilbert spaces defined previously.

After knowing how to solve the sub-problems, we apply an alternative minimization to solve the problem (8.14). See Algorithm 19. It iteratively and alternatively computes the variable u and p according to (8.18) and (8.20). Here L can be chosen using some convergence test techniques. As observed in [15, 16, 17] and also our numerical experiments, $L = 1$ is a good choice and thus applied in this chapter.

Algorithm 19 Augmented Lagrangian method for the TV restoration model – solve the minimization problem (8.14)

- *Initialization:* $u^{k,0} = u^{k-1}, p^{k,0} = p^{k-1}$;
 - *For* $l = 0, 1, 2, \dots, L - 1$: Compute $u^{k,l+1}$ from (8.18) for $p = p^{k,l}$; and then compute $p^{k,l+1}$ from (8.20) for $u = u^{k,l+1}$;
 - $u^k = u^{k,L}, p^k = p^{k,L}$.
-

By the zero initialization of Algorithm 18 and induction, it is guaranteed that $u^k \in V_M, p^k \in Q_M, \forall k$.

8.5 ALM for Vectorial TV Restoration on Triangulated Surfaces

The vectorial TV model (8.6) is also non-differentiable and therefore hard to solve. In this section we present the augmented Lagrangian method for this model, which is extended from the previous section.

By introducing a new variable $\mathbf{p} \in \mathbf{Q}_M$, we reformulate the problem (8.6) to be the following equality constrained problem

$$\begin{aligned} \min_{\mathbf{u} \in \mathbf{V}_M, \mathbf{p} \in \mathbf{Q}_M} & \left\{ R_{\text{vTV}}(\mathbf{p}) + \frac{\mu}{2} \|\mathbf{u} - \mathbf{f}\|_{\mathbf{V}_M}^2 \right\}, \\ \text{s.t.,} & \quad \mathbf{p} = \nabla_M \mathbf{u}. \end{aligned} \tag{8.22}$$

We then define, for (8.22), the following augmented Lagrangian functional

$$\mathcal{L}_{\text{vtv}}(\mathbf{u}, \mathbf{p}; \lambda) = R_{\text{vtv}}(\mathbf{p}) + \frac{\mu}{2} \|\mathbf{u} - \mathbf{f}\|_{\mathbf{V}_M}^2 + ((\lambda, \mathbf{p} - \nabla_M \mathbf{u}))_{\mathbf{Q}_M} + \frac{r}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{\mathbf{Q}_M}^2, \quad (8.23)$$

with Lagrange multiplier $\lambda \in \mathbf{Q}_M$ and positive constant r , and consider the saddle-point problem

$$\begin{aligned} &\text{Find } (\mathbf{u}^*, \mathbf{p}^*; \lambda^*) \in \mathbf{V}_M \times \mathbf{Q}_M \times \mathbf{Q}_M, \\ &\text{s.t., } \mathcal{L}_{\text{vtv}}(\mathbf{u}^*, \mathbf{p}^*; \lambda) \leq \mathcal{L}_{\text{vtv}}(\mathbf{u}^*, \mathbf{p}^*; \lambda^*) \leq \mathcal{L}_{\text{vtv}}(\mathbf{u}, \mathbf{p}; \lambda^*), \forall (\mathbf{u}, \mathbf{p}; \lambda) \in \mathbf{V}_M \times \mathbf{Q}_M \times \mathbf{Q}_M. \end{aligned} \quad (8.24)$$

It can be shown that the saddle-point problem (8.24) has at least one solution and all the saddle-points $(\mathbf{u}^*, \mathbf{p}^*; \lambda^*)$'s have the same $\mathbf{u}^*$, which is the unique solution of the original problem (8.6).

Algorithm 20 Augmented Lagrangian method for the vectorial TV model

1. *Initialization:* $\lambda^0 = 0, \mathbf{u}^{-1} = 0, \mathbf{p}^{-1} = 0$;
2. *For* $k=0, 1, 2, \dots$: compute $(\mathbf{u}^k, \mathbf{p}^k)$ as a (approximate) minimizer of the augmented Lagrangian functional with the Lagrange multiplier λ^k , *i.e.*,

$$(\mathbf{u}^k, \mathbf{p}^k) \approx \arg \min_{(\mathbf{u}, \mathbf{p}) \in \mathbf{V}_M \times \mathbf{Q}_M} \mathcal{L}_{\text{vtv}}(\mathbf{u}, \mathbf{p}; \lambda^k), \quad (8.25)$$

where $\mathcal{L}_{\text{vtv}}(\mathbf{u}, \mathbf{p}; \lambda^k)$ is as defined in (8.23); and update

$$\lambda^{k+1} = \lambda^k + r(\mathbf{p}^k - \nabla_M \mathbf{u}^k). \quad (8.26)$$

We use Algorithm 20 to solve the saddle-point problem, where the minimization problem (8.25) is left to address. Since $\mathbf{u}, \mathbf{p}$ are coupled together in (8.25), we separate the variable $\mathbf{u}$ and $\mathbf{p}$ and then apply an alternative minimization procedure, as in the method for the TV model (8.5).

In particular, (8.25) is separated into the following two sub-problems:

- The $\mathbf{u}$ -sub problem: for a given $\mathbf{p}$,

$$\min_{\mathbf{u} \in \mathbf{V}_M} \left\{ \frac{\mu}{2} \|\mathbf{u} - \mathbf{f}\|_{\mathbf{V}_M}^2 - ((\lambda^k, \nabla_M \mathbf{u}))_{\mathbf{Q}_M} + \frac{r}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{\mathbf{Q}_M}^2 \right\}. \quad (8.27)$$

– The $\mathbf{p}$ –sub problem: for a given $\mathbf{u}$,

$$\min_{\mathbf{p} \in \mathbf{Q}_M} \left\{ R_{\text{vtr}}(\mathbf{p}) + ((\lambda^k, \mathbf{p}))_{\mathbf{Q}_M} + \frac{r}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{\mathbf{Q}_M}^2 \right\}. \quad (8.28)$$

8.5.1 Solving the $\mathbf{u}$ –sub problem

The $\mathbf{u}$ –sub problem (8.27) is a quadratic programming, whose optimality condition is as follows

$$\mu(\mathbf{u} - \mathbf{f}) + \text{div}_M \lambda^k + r \text{div}_M \mathbf{p} - r \text{div}_M \nabla_M \mathbf{u} = 0, \quad (8.29)$$

which is, in channel wise, as follows

$$\mu(u_m - f_m) + \text{div}_M \lambda_m^k + r \text{div}_M p_m - r \text{div}_M \nabla_M u_m = 0, 1 \leq m \leq \mathfrak{M}. \quad (8.30)$$

Therefore one only needs to solve $\mathfrak{M}$ sparse linear systems with the same coefficient matrix. Some well-developed numerical packages can be applied.

8.5.2 Solving the $\mathbf{p}$ –sub problem

The $\mathbf{p}$ –sub problem is decomposable and can be solved triangle-by-triangle. For each triangle τ , we need to solve

$$\min_{\mathbf{p}_\tau} \left\{ |\mathbf{p}_\tau| + (\lambda_\tau^k, \mathbf{p}_\tau) + \frac{r}{2} |\mathbf{p}_\tau - (\nabla_M \mathbf{u})_\tau|^2 \right\}, \quad (8.31)$$

which has the following closed form solution

$$\mathbf{p}_\tau = \begin{cases} (1 - \frac{1}{r} \frac{1}{|\mathbf{w}_\tau|}) \mathbf{w}_\tau, & |\mathbf{w}_\tau| > \frac{1}{r}, \\ 0, & |\mathbf{w}_\tau| \leq \frac{1}{r}, \end{cases} \quad (8.32)$$

where

$$\mathbf{w} = \nabla_M \mathbf{u} - \frac{\lambda^k}{r}. \quad (8.33)$$

It is seen that $\mathbf{p}_\tau$ obtained via (8.32) is in the tangent vector space $T\tau$ and thus $\mathbf{p} \in \mathbf{Q}_M$, as long as $\lambda_\tau^k \in T\tau, \forall \tau$. This is also a natural consequence from the

definition of $\mathbf{Q}_M$.

We then apply an alternative minimization to solve the problem (8.25). See Algorithm 21, which iteratively and alternatively computes the variable $\mathbf{u}$ and $\mathbf{p}$ according to (8.29) and (8.32). Here L can be chosen using some convergence test techniques. In this chapter we simply set $L = 1$.

Algorithm 21 Augmented Lagrangian method for the vectorial TV model – solve the minimization problem (8.25)

- *Initialization:* $\mathbf{u}^{k,0} = \mathbf{u}^{k-1}, \mathbf{p}^{k,0} = \mathbf{p}^{k-1}$;
 - *For* $l = 0, 1, 2, \dots, L - 1$: Compute $\mathbf{u}^{k,l+1}$ from (8.29) for $\mathbf{p} = \mathbf{p}^{k,l}$; and then compute $\mathbf{p}^{k,l+1}$ from (8.32) for $\mathbf{u} = \mathbf{u}^{k,l+1}$;
 - $\mathbf{u}^k = \mathbf{u}^{k,L}, \mathbf{p}^k = \mathbf{p}^{k,L}$.
-

By the zero initialization of Algorithm 20 and induction, it is guaranteed that $\mathbf{u}^k \in \mathbf{V}_M, \mathbf{p}^k \in \mathbf{Q}_M, \forall k$.

8.6 ALM for Multi-region Segmentation on Triangulated Surfaces

Due to the involvement of the vectorial TV semi-norm, the multi-region segmentation model (8.8) is non-differentiable and thus hard to solve. In this section we present to use the augmented Lagrangian method to solve it.

By defining

$$\chi_C(\mathbf{u}) = \begin{cases} 0, & \mathbf{u} \in C, \\ +\infty, & \mathbf{u} \notin C, \end{cases}$$

the problem (8.8) can be verified to be equivalent to the following minimization problem

$$\min_{\mathbf{u} \in \mathbf{V}_M} \left\{ \overline{G}_{\text{mrs}}(\mathbf{u}) = ((\mathbf{u}, \mathbf{s}))_{\mathbf{V}_M} + \beta R_{\text{mrs}}(\nabla_M \mathbf{u}) + \chi_C(\mathbf{u}) \right\}, \quad (8.34)$$

where the objective functional $\overline{G}_{\text{mrs}}(\mathbf{u})$ is convex, lower semi-continuous, and proper, as well as coercive over $\mathbf{V}_M$, by Cauchy-Schwartz inequality applied to $((\mathbf{u}, \mathbf{s}))_{\mathbf{V}_M}$.

The problem (8.34) can be further reformulated to

$$\begin{aligned} \min_{\mathbf{u} \in \mathbf{V}_M, \mathbf{p} \in \mathbf{Q}_M, \mathbf{z} \in \mathbf{V}_M} & \left\{ ((\mathbf{z}, \mathbf{s}))_{\mathbf{V}_M} + \beta R_{\text{mrs}}(\mathbf{p}) + \chi_C(\mathbf{z}) \right\}, \\ \text{s.t.,} & \quad \mathbf{p} = \nabla_M \mathbf{u}, \quad \mathbf{z} = \mathbf{u}, \end{aligned} \quad (8.35)$$

by introducing two auxiliary variable $\mathbf{p} \in \mathbf{Q}_M$ and $\mathbf{z} \in \mathbf{V}_M$. We remark that the introduction of $\mathbf{z}$ avoids a difficult inequality constrained quadratic programming problem (especially for large meshes) and will greatly simplify the calculation.

To solve (8.35), we define the following augmented Lagrangian functional

$$\begin{aligned} \mathcal{L}_{\text{mrs}}(\mathbf{u}, \mathbf{p}, \mathbf{z}; \lambda_{\mathbf{p}}, \lambda_{\mathbf{z}}) = & ((\mathbf{z}, \mathbf{s}))_{\mathbf{V}_M} + \beta R_{\text{mrs}}(\mathbf{p}) + \chi_C(\mathbf{z}) + ((\lambda_{\mathbf{p}}, \mathbf{p} - \nabla_M \mathbf{u}))_{\mathbf{Q}_M} \\ & + ((\lambda_{\mathbf{z}}, \mathbf{z} - \mathbf{u}))_{\mathbf{V}_M} + \frac{r_{\mathbf{p}}}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{\mathbf{Q}_M}^2 + \frac{r_{\mathbf{z}}}{2} \|\mathbf{z} - \mathbf{u}\|_{\mathbf{V}_M}^2, \end{aligned} \quad (8.36)$$

with Lagrange multipliers $\lambda_{\mathbf{p}} \in \mathbf{Q}_M$, $\lambda_{\mathbf{z}} \in \mathbf{V}_M$ and positive constants $r_{\mathbf{p}}, r_{\mathbf{z}}$, and then consider the following saddle-point problem

$$\begin{aligned} \text{Find } & (\mathbf{u}^*, \mathbf{p}^*, \mathbf{z}^*; \lambda_{\mathbf{p}}^*, \lambda_{\mathbf{z}}^*) \in \mathbf{V}_M \times \mathbf{Q}_M \times \mathbf{V}_M \times \mathbf{Q}_M \times \mathbf{V}_M, \\ \text{s.t., } & \mathcal{L}_{\text{mrs}}(\mathbf{u}^*, \mathbf{p}^*, \mathbf{z}^*; \lambda_{\mathbf{p}}, \lambda_{\mathbf{z}}) \leq \mathcal{L}_{\text{mrs}}(\mathbf{u}^*, \mathbf{p}^*, \mathbf{z}^*; \lambda_{\mathbf{p}}^*, \lambda_{\mathbf{z}}^*) \leq \mathcal{L}_{\text{mrs}}(\mathbf{u}, \mathbf{p}, \mathbf{z}; \lambda_{\mathbf{p}}^*, \lambda_{\mathbf{z}}^*), \\ & \forall (\mathbf{u}, \mathbf{p}, \mathbf{z}; \lambda_{\mathbf{p}}, \lambda_{\mathbf{z}}) \in \mathbf{V}_M \times \mathbf{Q}_M \times \mathbf{V}_M \times \mathbf{Q}_M \times \mathbf{V}_M. \end{aligned} \quad (8.37)$$

It can be shown that the above saddle-point problem (8.37) has at least one solution and each solution $(\mathbf{u}^*, \mathbf{p}^*, \mathbf{z}^*; \lambda_{\mathbf{p}}^*, \lambda_{\mathbf{z}}^*)$ provides a minimizer $\mathbf{u}^*$ of the segmentation problem (8.8).

We now present Algorithm 22 to solve the saddle-point problem (8.37), where the minimization problem (8.38) is left to address in later paragraphs. Similarly to those in TV and vectorial TV models, here we have coupled variables $\mathbf{u}, \mathbf{p}, \mathbf{z}$. We need to separate the variable $\mathbf{u}, \mathbf{p}$, and $\mathbf{z}$ and then apply an alternative minimization procedure to solve (8.38).

In particular, (8.38) is separated into the following three sub-problems:

Algorithm 22 Augmented Lagrangian method for multi-region segmentation

1. *Initialization:* $\lambda_{\mathbf{p}}^0 = 0, \lambda_{\mathbf{z}}^0 = 0, \mathbf{u}^{-1} = 0, \mathbf{p}^{-1} = 0, \mathbf{z}^{-1} = 0$;
2. *For* $k=0,1,2,\dots$: compute $(\mathbf{u}^k, \mathbf{p}^k, \mathbf{z}^k)$ as an (approximate) minimizer of the augmented Lagrangian functional with the Lagrange multipliers $\lambda_{\mathbf{p}}^k, \lambda_{\mathbf{z}}^k$, i.e.,

$$(\mathbf{u}^k, \mathbf{p}^k, \mathbf{z}^k) \approx \arg \min_{(\mathbf{u}, \mathbf{p}, \mathbf{z}) \in \mathbf{V}_M \times \mathbf{Q}_M \times \mathbf{V}_M} \mathcal{L}_{\text{mrs}}(\mathbf{u}, \mathbf{p}, \mathbf{z}; \lambda_{\mathbf{p}}^k, \lambda_{\mathbf{z}}^k), \quad (8.38)$$

where $\mathcal{L}_{\text{mrs}}(\mathbf{u}, \mathbf{p}, \mathbf{z}; \lambda_{\mathbf{p}}^k, \lambda_{\mathbf{z}}^k)$ is as defined in (8.36); and update

$$\begin{aligned} \lambda_{\mathbf{p}}^{k+1} &= \lambda_{\mathbf{p}}^k + r_{\mathbf{p}}(\mathbf{p}^k - \nabla_M \mathbf{u}^k), \\ \lambda_{\mathbf{z}}^{k+1} &= \lambda_{\mathbf{z}}^k + r_{\mathbf{z}}(\mathbf{z}^k - \mathbf{u}^k). \end{aligned} \quad (8.39)$$

- The $\mathbf{u}$ –sub problem: for given $\mathbf{p}, \mathbf{z}$,

$$\min_{\mathbf{u} \in \mathbf{V}_M} \left\{ -((\lambda_{\mathbf{p}}^k, \nabla_M \mathbf{u}))_{\mathbf{Q}_M} - ((\lambda_{\mathbf{z}}^k, \mathbf{u}))_{\mathbf{V}_M} + \frac{r_{\mathbf{p}}}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{\mathbf{Q}_M}^2 + \frac{r_{\mathbf{z}}}{2} \|\mathbf{z} - \mathbf{u}\|_{\mathbf{V}_M}^2 \right\}. \quad (8.40)$$

- The $\mathbf{p}$ –sub problem: for given $\mathbf{u}, \mathbf{z}$,

$$\min_{\mathbf{p} \in \mathbf{Q}_M} \left\{ \beta R_{\text{mrs}}(\mathbf{p}) + ((\lambda_{\mathbf{p}}^k, \mathbf{p}))_{\mathbf{Q}_M} + \frac{r_{\mathbf{p}}}{2} \|\mathbf{p} - \nabla_M \mathbf{u}\|_{\mathbf{Q}_M}^2 \right\}. \quad (8.41)$$

- The $\mathbf{z}$ –sub problem: for given $\mathbf{u}, \mathbf{p}$,

$$\min_{\mathbf{z} \in \mathbf{V}_M} \left\{ ((\mathbf{z}, \mathbf{s}))_{\mathbf{V}_M} + \chi_C(\mathbf{z}) + ((\lambda_{\mathbf{z}}^k, \mathbf{z}))_{\mathbf{V}_M} + \frac{r_{\mathbf{z}}}{2} \|\mathbf{z} - \mathbf{u}\|_{\mathbf{V}_M}^2 \right\}. \quad (8.42)$$

All of these sub-problems can be efficiently solved.

8.6.1 Solving the $\mathbf{u}$ –sub problem

The $\mathbf{u}$ –sub problem (8.40) is a quadratic programming, whose optimality condition is as follows

$$\operatorname{div}_M \lambda_{\mathbf{p}}^k - \lambda_{\mathbf{z}}^k + r_{\mathbf{p}} \operatorname{div}_M \mathbf{p} - r_{\mathbf{p}} \operatorname{div}_M \nabla_M \mathbf{u} - r_{\mathbf{z}}(\mathbf{z} - \mathbf{u}) = 0, \quad (8.43)$$

which is, in channel wise, as follows

$$\operatorname{div}_M(\lambda_{\mathbf{p}})_{\mathbf{m}}^k - (\lambda_{\mathbf{z}})_{\mathbf{m}}^k + r_{\mathbf{p}} \operatorname{div}_M p_{\mathbf{m}} - r_{\mathbf{p}} \operatorname{div}_M \nabla_M u_{\mathbf{m}} - r_{\mathbf{z}}(z_{\mathbf{m}} - u_{\mathbf{m}}) = 0, 1 \leq \mathbf{m} \leq \mathfrak{M}. \quad (8.44)$$

These $\mathfrak{M}$ sparse linear systems with the same coefficient matrix can be very efficiently solved by well-developed numerical packages.

8.6.2 Solving the $\mathbf{p}$ -sub problem

The $\mathbf{p}$ -sub problem (8.41) is decomposable and can be solved triangle-by-triangle. For each triangle τ , we need to solve

$$\min_{\mathbf{p}_{\tau}} \left\{ \beta |\mathbf{p}_{\tau}| + ((\lambda_{\mathbf{p}})_{\tau}^k, \mathbf{p}_{\tau}) + \frac{r_{\mathbf{p}}}{2} |\mathbf{p}_{\tau} - (\nabla_M \mathbf{u})_{\tau}|^2 \right\}, \quad (8.45)$$

which has the following closed form solution

$$\mathbf{p}_{\tau} = \begin{cases} (1 - \frac{\beta}{r_{\mathbf{p}} |\mathbf{w}_{\tau}|}) \mathbf{w}_{\tau}, & |\mathbf{w}_{\tau}| > \frac{\beta}{r_{\mathbf{p}}}, \\ 0, & |\mathbf{w}_{\tau}| \leq \frac{\beta}{r_{\mathbf{p}}}, \end{cases} \quad (8.46)$$

where

$$\mathbf{w} = \nabla_M \mathbf{u} - \frac{\lambda_{\mathbf{p}}^k}{r_{\mathbf{p}}}. \quad (8.47)$$

It can be seen that $\mathbf{p}_{\tau}$ obtained via (8.46) is in the tangent vector space $T\tau$ and thus $\mathbf{p} \in \mathbf{Q}_M$, as long as $(\lambda_{\mathbf{p}})_{\tau}^k \in T\tau, \forall \tau$. This is also indicated by the Hilbert space in the problem formulation.

8.6.3 Solving the $\mathbf{z}$ -sub problem

The $\mathbf{z}$ -sub problem (8.42) is equivalent to

$$\min_{\mathbf{z} \in C} \left\{ ((\mathbf{z}, \mathbf{s}))_{\mathbf{V}_M} + ((\lambda_{\mathbf{z}}^k, \mathbf{z}))_{\mathbf{V}_M} + \frac{r_{\mathbf{z}}}{2} \|\mathbf{z} - \mathbf{u}\|_{\mathbf{V}_M}^2 \right\}. \quad (8.48)$$

We can first minimize the objective functional over $\mathbf{V}_M$ and then project the minimizer to C . The solution to (8.48) reads as follows

$$\mathbf{z} = \text{Proj}_C\left(\mathbf{u} - \frac{\mathbf{s} + \lambda_{\mathbf{z}}^k}{r_{\mathbf{z}}}\right), \quad (8.49)$$

which can be calculated vertex-by-vertex via Michelot's algorithm [166].

We then apply an alternative minimization to solve the problem (8.38). See Algorithm 23, which iteratively and alternatively computes the $\mathbf{u}$, $\mathbf{p}$ and $\mathbf{z}$, according to (8.43), (8.46), and (8.49), respectively. Here L can be chosen using some convergence test techniques. In this chapter we simply set $L = 1$.

Algorithm 23 Augmented Lagrangian method for multi-region segmentation – solve the minimization problem (8.38)

- *Initialization:* $\mathbf{u}^{k,0} = \mathbf{u}^{k-1}, \mathbf{p}^{k,0} = \mathbf{p}^{k-1}, \mathbf{z}^{k,0} = \mathbf{z}^{k-1}$;
 - *For* $l = 0, 1, 2, \dots, L - 1$: Compute $\mathbf{u}^{k,l+1}$ from (8.43) for $\mathbf{p} = \mathbf{p}^{k,l}, \mathbf{z} = \mathbf{z}^{k,l}$; then compute $\mathbf{p}^{k,l+1}$ from (8.46) for $\mathbf{u} = \mathbf{u}^{k,l+1}, \mathbf{z} = \mathbf{z}^{k,l}$; and then compute $\mathbf{z}^{k,l+1}$ from (8.49) for $\mathbf{u} = \mathbf{u}^{k,l+1}, \mathbf{p} = \mathbf{p}^{k,l+1}$;
 - $\mathbf{u}^k = \mathbf{u}^{k,L}, \mathbf{p}^k = \mathbf{p}^{k,L}, \mathbf{z}^k = \mathbf{z}^{k,L}$.
-

By the zero initialization of Algorithm 22 and induction, it is guaranteed that $\mathbf{u}^k \in \mathbf{V}_M, \mathbf{p}^k \in \mathbf{Q}_M, \mathbf{z}^k \in \mathbf{V}_M, \forall k$.

Remark. The problem (8.10) is non-convex and it is usually difficult to find its global minima. One may consider to apply an alternative minimization procedure with respect to $\mathbf{u}$ (with fixed ς) and ς (with fixed $\mathbf{u}$). For fixed ς , Algorithm 22 can be applied to find $\mathbf{u}$. For fixed $\mathbf{u}$, ς can be easily obtained by its first order optimality condition. However, this alternative minimization algorithm is not guaranteed to converge, since the problem (8.10) is non-convex. Currently our tests for (8.10) show correct segmentation results for only the case of two-region segmentation.

8.7 Numerical Examples

8.7.1 Test platform, stopping condition, and the computation of SNR

We implemented the proposed algorithms in C++. All of the numerical experiments were made on a laptop with Intel Core 2 Duo 2.80GHz CPU and 4GB RAM and MS Windows XP Professional x64 Edition.

In our experiments, we applied the following stopping condition

$$\|u^k - u^{k-1}\|_{V_M} < TOL \text{ (or } \|\mathbf{u}^k - \mathbf{u}^{k-1}\|_{\mathbf{V}_M} < TOL), \text{ for a prescribed } TOL > 0.$$

See section 3 for the definitions of $|||_{V_M}$ and $|||_{\mathbf{V}_M}$.

The SNRs given in the examples were computed as follows. For gray images,

$$\text{SNR} = 10 \log \frac{\|u - \bar{u}\|_{V_M}^2}{\|u - g\|_{V_M}^2}, \quad (8.50)$$

where u is the noise-free (clean) image; g is the observed or recovered image; $\bar{u}$ is the average intensity of the image u on M . The formula for color images is similarly defined by using the definition of the norm $|||_{\mathbf{V}_M}$.

8.7.2 Examples

Here we give some examples. In particular, we used the following triangular mesh surfaces in our experiments; see Table 8.1 where the mesh sizes are shown.

A group of examples of ALM applied to TV and vectorial TV based restoration and segmentation of gray and color images are shown in Figures 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, and 8.7. The experimental information of these examples is given in Tables 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, and 8.8, respectively. As one can see, our algorithms are very efficient for both gray and color image restoration and segmentation.

In image restoration, our algorithms generate reconstruction results with greatly improved SNRs in seconds. In all these restoration examples, the parameter r of ALM

Table 8.1: Mesh surfaces used in this chapter and their sizes (numbers of vertices and triangles).

Mesh surface	# vertices	# triangles
Sphere	40962	81920
Horse	48485	96966
Cameraman	66049	131072
Lena	65536	130050
Bunny	34817	69630

and the stopping condition TOL were simply set to be 0.01 and $1.0e^{-7}$, respectively.

In multi-region segmentation, our algorithms also perform efficiently and generate correct segmentation results. Experiments show that they are robust to image noise and inaccurate inputs of intensity means of segments ς present in the segmentation model. The robustness to noise makes our method work for noisy images. The robustness to inaccurate inputs is of great importance in real applications, since in noisy images the intensity means ς are difficult to estimate precisely. In these segmentation examples we used uniform parameters and stopping condition; see Tables 8.7 and 8.8.

At the end of this section, we remark on the choice of the positive parameters $r, r_{\mathbf{p}}, r_{\mathbf{z}}$ in the augmented Lagrangian method. According to our experiments, there seem some ranges for the values of these parameters to make the algorithms stable and efficient. The algorithms may be not numerically stable for too large or too small positive parameters. One potential reason is that, due to the irregularity of the mesh surface, the condition number of the coefficient matrix of the u -sub or $\mathbf{u}$ -sub problem and hence the stability of the algorithms are more sensitive to the parameters $r, r_{\mathbf{p}}, r_{\mathbf{z}}$, compared to the case of 2D planar image processing. Consequently, r (or $r_{\mathbf{p}}, r_{\mathbf{z}}$) depends on the model parameter μ (or β), the gradient and divergence operators. The model parameter μ (or β) depends on the image to be processed. The two differential operators are determined by the structure and distribution of the triangles on the mesh surface as well as the mesh resolution, i.e., the geometry of the mesh surface. To precisely describe the relation between good parameters r (or $r_{\mathbf{p}}, r_{\mathbf{z}}$) and μ (or β) as well as the geometry of the mesh surface is quite difficult. Therefore, it is hard

to obtain optimal values of $r, r_{\mathbf{p}}, r_{\mathbf{z}}$. Fortunately, our tests show that, the effective ranges of $r, r_{\mathbf{p}}, r_{\mathbf{z}}$ are large enough so that common parameters r (or $r_{\mathbf{p}}, r_{\mathbf{z}}$) exist for different mesh surfaces and different model parameters μ (or β); see the examples. Note that the effective ranges of r and $r_{\mathbf{p}}, r_{\mathbf{z}}$ may be quite different even for the same mesh surface and image because they are used in different minimization problems (with different objective functions) in different applications.

Figure 8.1: Denoising two images (with non-symmetric and symmetric structures, respectively) with different levels of noise on the sphere surface based on TV restoration. The first and third rows are noisy images. From left to right, the standard deviations of the noise are 0.04, 0.06, 0.08, 0.1, respectively. The second and fourth rows are recovered images. See Table 8.2.

Table 8.2: Experimental information about the examples in Figure 8.1. The stopping condition is $TOL = 1.e^{-7}$.

Example #	Observation		Parameters		Recovered	
	Noise level	SNR(dB)	μ	r	SNR(dB)	CPU(s)
Column 1 NonSymm	0.04	17.5309	4500	0.01	30.0471	3.235
Column 2 NonSymm	0.06	14.0091	3000	0.01	27.8715	3.282
Column 3 NonSymm	0.08	11.5103	2500	0.01	26.2949	3.641
Column 4 NonSymm	0.10	9.5721	2000	0.01	24.9920	3.781
Column 1 Symm	0.04	17.1230	5000	0.01	27.9418	3.375
Column 2 Symm	0.06	13.6012	3500	0.01	25.5181	3.437
Column 3 Symm	0.08	11.1024	2500	0.01	23.7845	3.656
Column 4 Symm	0.10	9.1642	2000	0.01	22.4252	3.484

Figure 8.2: Denoising a text-image with different levels of noise on the horse surface based on TV restoration. The first row are noisy images. From left to right, the standard deviations of the noise are 0.02, 0.04, 0.06, 0.08, respectively. The second row are recovered images. See Table 8.3.

Table 8.3: Experimental information about the examples in Figure 8.2. The stopping condition is $TOL = 1.e^{-7}$.

Example #	Observation		Parameters		Recovered	
	Noise level	SNR(dB)	μ	r	SNR(dB)	CPU(s)
Column 1	0.02	5.8059	11000	0.01	16.5931	1.672
Column 2	0.04	-0.2147	7000	0.01	13.0695	1.953
Column 3	0.06	-3.7366	5000	0.01	10.9710	2.172
Column 4	0.08	-6.2353	4000	0.01	9.3320	2.359

Figure 8.3: Denoising the cameraman image with different levels of noise on a flat surface based on TV restoration. The first row are noisy images. From left to right, the standard deviations of the noise are 0.06, 0.08, 0.10, 0.12, respectively. The second row are recovered images. See Table 8.4.

Table 8.4: Experimental information about the examples in Figure 8.3. The stopping condition is $TOL = 1.e^{-7}$.

Example #	Observation		Parameters		Recovered	
	Noise level	SNR(dB)	μ	r	SNR(dB)	CPU(s)
Column 1	0.06	12.0609	10000	0.01	19.1417	3.735
Column 2	0.08	9.5621	8000	0.01	17.6617	3.828
Column 3	0.10	7.6239	6000	0.01	16.5206	3.796
Column 4	0.12	6.0384	4500	0.01	15.5818	4.25

Figure 8.4: Denoising a color image with different levels of noise on the Bunny surface based on vectorial TV restoration. The first row are noisy images. From left to right, the standard deviations of the noise are 0.04, 0.06, 0.08, 0.10, respectively. The second row are recovered images. See Table 8.5.

Table 8.5: Experimental information about the examples in Figure 8.4. The stopping condition is $TOL = 1.e^{-7}$.

Example #	Observation		Parameters		Recovered	
	Noise level	SNR(dB)	μ	r	SNR(dB)	CPU(s)
Column 1	0.04	10.9407	1500	0.01	29.5013	5.797
Column 2	0.06	7.4189	1200	0.01	27.5986	6.187
Column 3	0.08	4.9201	1000	0.01	26.2133	7.109
Column 4	0.10	2.9819	900	0.01	25.3085	8.500

Figure 8.5: Denoising the Lena (color) image with different levels of noise on a flat surface based on vectorial TV restoration. The first row are noisy images. From left to right, the standard deviations of the noise are 0.06, 0.08, 0.10, 0.12, respectively. The second row are recovered images. See Table 8.6.

Table 8.6: Experimental information about the examples in Figure 8.5. The stopping condition is $TOL = 1.e^{-7}$.

	Observation		Parameters		Recovered	
Example #	Noise level	SNR(dB)	μ	r	SNR(dB)	CPU(s)
Column 1	0.06	8.7710	7000	0.01	15.7636	8.140
Column 2	0.08	6.2722	5000	0.01	14.3979	9.000
Column 3	0.10	4.3340	4000	0.01	13.3181	9.391
Column 4	0.12	2.7503	4000	0.01	11.8496	9.484

Figure 8.6: Multi-region segmentation of gray and color images based on vectorial TV regularization. From left to right: the images are divided into 3,4,2,4 segments (Note that one segment may have several different disconnected domains whose intensity values are the same or similar), respectively. The 1st row is segmentation results on clean images, while the 2nd row contains segmentation results on noisy images. These examples show the robustness to noise of our algorithms. See Table 8.7.

Table 8.7: Experimental information about the examples in Figure 8.6. In all these examples, the parameters $(\beta, r_p, r_z) = (5.e^{-5}, 5.e^{-8}, 1)$. The stopping condition is $TOL = 1.e^{-5}$.

Example #		SNR(dB)	Intensity means of segments	CPU(s)
Column 1	Row 1	∞	(0.6 0.6 0.6); (0.3 0.3 0.3);	7.485
	Row 2	16.8889	(0.9 0.9 0.9)	8.953
Column 2	Row 1	∞	(0.1 0.1 0.1); (0.4 0.4 0.4);	16.562
	Row 2	16.1223	(0.7 0.7 0.7); (0.9 0.9 0.9)	19.359
Column 3	Row 1	∞	(0.565 0.565 0.565); (0.1 0.1 0.1)	3.39
	Row 2	-4.40196		3.937
Column 4	Row 1	∞	(0.9 0.9 0.9); (0.501961 0.25098 0.25098);	3.922
	Row 2	4.92012	(1 0 0.501961); (0 0 1)	4.937

Figure 8.7: Multi-region segmentation of noisy gray and color images based on vectorial TV regularization with accurate and inaccurate inputs of intensity means of segments. The input intensity means of segments in Column 1 and 3 are accurate, while those in Column 2 and 4 are inaccurate. These examples show that our algorithms still work even for inaccurate inputs of intensity means of segments. This robustness to inaccurate inputs is of importance for segmentation of noisy images, since in noisy images the intensity means of segments are difficult to estimate precisely. See Table 8.8.

Table 8.8: Experimental information about the examples in Figure 8.7. In all these examples, the parameters $(\beta, r_{\mathbf{p}}, r_{\mathbf{z}}) = (5.e^{-5}, 5.e^{-8}, 1)$. The stopping condition is $TOL = 1.e^{-5}$.

Example #	SNR(dB)	Intensity means of segments	CPU(s)
Column 1	16.1223	(0.1 0.1 0.1); (0.4 0.4 0.4); (0.7 0.7 0.7); (0.9 0.9 0.9)	19.359
Column 2		(0.05 0.05 0.05); (0.42 0.42 0.42); (0.72 0.72 0.72); (0.92 0.92 0.92)	20
Column 3	4.92012	(0.9 0.9 0.9); (0.501961 0.25098 0.25098); (1 0 0.501961); (0 0 1)	4.937
Column 4		(1 1 1); (0.55 0.3 0.3); (1 0 0.6); (0.1 0 1)	3.812

9.1 Conclusion

This dissertation has studied efficient numerical methods for image processing based on partial differential equations. Our work has concentrated on variational models in image denoising and segmentation. Fast algorithms were established to solve the energy minimization problems for both continuous and discrete settings in the dissertation. For the discrete approaches, we applied the domain decomposition method to accelerate the graph cuts algorithm for solving discrete energy functionals. For the continuous approaches, we implemented the augmented Lagrangian method to reformulate the energy functionals and applied the operator splitting and decomposition technique to generate simple energy structure according to different variables. We conducted numerous experiments to demonstrate the efficiency of the proposed algorithms in the dissertation.

Firstly, we have proposed novel algorithms to minimize the discrete total variation based models by applying the domain decomposition method in the image domain. We have focused on three basic models incorporating total variation regularization: the ROF model and the TV- L^1 model for noise removal problems in Chapter 4 and the Mumford-Shah model for segmentation problems in Chapter 5. The proposed

algorithms are demonstrated to successfully improve the computational efficiency. Even more, our algorithms greatly reduced the memory cost and enabled us to solve problems with large image sizes. Due to the monotonicity property of the domain decomposition method, the numerical performance of the proposed algorithms was shown to be very robust. It is notable that our algorithms can segment 3D images with around $4 \times 10^8 (512 \times 512 \times 512 \times 3)$ voxels in just a few minutes and the quality is comparable with traditional variational methods.

Secondly, we have used the augmented Lagrangian method to solve the general convex minimization problems arising in image processing in Chapter 6. To get an even faster numerical solver, we applied the fixed-point method to the minimization problem without solving the linear system. The proposed algorithms solved the minimization problems just depending on matrix / vector multiplication and acquired results with comparable quality as other well-known methods (FFT, Gauss-Seidel). We also implemented an acceleration technique on the fixed-point augmented Lagrangian algorithms to achieve the better convergence. The convergence proofs of the proposed algorithms for general convex problems were provided in our work. The algorithms were demonstrated to be efficient by numerous numerical experiments on several variational models including the ROF model, the vectorial TV model, the 4th-order model and the TV- L^1 model.

Thirdly, we have proposed a simple and efficient augmented Lagrangian approach for the Euler's elastica functional in Chapter 7. Minimization of functionals related to the Euler's elastica energy has a wide range of applications in computer vision and image processing. We adopted the operator splitting method to solve the saddle-point problem arising from the augmented Lagrangian formulation. The fixed-point method was artfully applied to one sub-problem to get a closed form solution. All the sub-problems in our method could be solved efficiently by fast iterative methods and closed form solutions. As the numerical results of image denoising problems demonstrated, our method yields good results in terms of computational time.

Finally, we have used the augmented Lagrangian method to solve TV and vectorial TV based image restoration and multi-region segmentation problems on triangular

mesh surfaces, based on defining inner product spaces on mesh surfaces. The provided algorithms worked well for both gray and color images. Due to the use of fast solvers for linear systems and closed form solutions to subproblems, the method was extremely efficient, as demonstrated by our numerical experiments.

9.2 Future works

In this dissertation, we have applied the fixed-point method to the augmented Lagrangian formulations of the general convex optimization problem and the Euler's elastica functional to obtain fast algorithms. We also implemented the augmented Lagrangian method to TV based models for image restoration and segmentation on triangulated surfaces. Beyond the success of these approaches, there still exist some problems we want to address in future work.

- **A fixed-point ALM for TV minimization problems**

Although we provided the numerical results of the TV- L^1 model in Chapter 6, there is no theoretical proof for the convergence. Therefore, one future work is to extend the convergence proof of the fixed-point augmented Lagrangian methods to more general optimization problems, which may not be strongly convex.

Moreover, the idea of the proposed fixed-point augmented Lagrangian method is similar to the linearized Bregman method in [19, 75]. In particular, the author studied the linearized Bregman iteration for the basis pursuit problem in [75], which is expressed as the following constrained minimization problem

$$\min\{\|x\|_1 : Ax = b\}. \quad (9.1)$$

The original Bregman method for (9.1) solves a sequence of convex problems in the iterative scheme

$$x^{k+1} = \arg \min_x \left\{ D_Q^g(x, x^k) + \frac{1}{2} \|Ax - b\|_2^2 \right\}, \quad (9.2)$$

where $Q(x) = \mu\|x\|_1$.

The linearized Bregman method is proposed by linearizing the last term in (9.2) into $\langle A^T(Ax^k - b), x \rangle$ and adding the L^2 -proximity term $\frac{1}{2e}\|x - x^k\|_2^2$, yielding the new iterative scheme

$$x^{k+1} = \arg \min_x \left\{ D_Q^{g^k}(x, x^k) + \langle A^T(Ax^k - b), x \rangle + \frac{1}{2e}\|x - x^k\|_2^2 \right\}. \quad (9.3)$$

The update formula of g can be derived as

$$g^{k+1} = g^k - A^T(Ax^k - b) - \frac{1}{e}(x^{k+1} - x^k). \quad (9.4)$$

Thus, the linearized Bregman iteration can be simplified into the following scheme

$$\begin{cases} x^{k+1} = \arg \min_x \left\{ Q(x) + \frac{1}{2e}\|x - ey^k\|^2 \right\}, \\ y^{k+1} = y^k + A^T(b - Ax^{k+1}). \end{cases} \quad (9.5)$$

In [75], the author introduced a smoothed version of Q as follows

$$f_e(x) := Q(x) + \frac{1}{2e}\|x\|^2.$$

The author in [167] pointed out that the linearized Bregman method can be derived from the multiplier method. Specially, define the Lagrangian $L(x, y) := f_e(x) + \langle y, b - Ax \rangle$. Then the linearized Bregman iteration can be obtained from the multiplier method, i.e.,

$$\begin{cases} x^{k+1} = \arg \min_x L(x, y^k), \\ y^{k+1} = y^k + \nabla_y L(x^{k+1}, y^k). \end{cases} \quad (9.6)$$

Furthermore, we can simplify (9.6) and get

$$\begin{cases} x^{k+1} = \arg \min_x \left\{ Q(x) + \frac{1}{2e}\|x - eA^T v^k\|^2 \right\}, \\ v^{k+1} = v^k + b - Ax^{k+1}. \end{cases} \quad (9.7)$$

By letting $v^k = Ay^k$, the scheme (9.7) becomes exactly the scheme (9.5).

To compare the proposed fixed-point augmented Lagrangian method with the linearized Bregman technique, we apply our method to the augmented Lagrangian formulation of the constrained basis pursuit problem (9.1), which is given as follows

$$\mathcal{L}_{\text{BP}}(x, y) = Q(x) + \langle y, b - Ax \rangle + \frac{r}{2} \|Ax - b\|_2^2. \quad (9.8)$$

We use the operator splitting method and then solve the problem (9.8) by the following scheme

$$\begin{cases} x^{k+1} = \arg \min_x \left\{ Q(x) - \langle y^k, Ax \rangle + \frac{r}{2} \|Ax - b\|_2^2 \right\}, \\ y^{k+1} = y^k + r(b - Ax^{k+1}). \end{cases} \quad (9.9)$$

Let $v^k = \frac{y^k}{r}$. We can reformulate the scheme (9.9) as

$$\begin{cases} x^{k+1} = \arg \min_x \left\{ Q(x) + \frac{r}{2} \|Ax - (b + v^k)\|_2^2 \right\}, \\ v^{k+1} = v^k + b - Ax^{k+1}. \end{cases} \quad (9.10)$$

If we apply our fixed-point method to the variable x in (9.10), we have

$$x^{k+1} = \arg \min_x \left\{ Q(x) + r \langle x, A^T(Ax^k - (b + v^k)) \rangle \right\}. \quad (9.11)$$

Therefore, the fixed-point augmented Lagrangian method for the basis pursuit problem (9.1) is given as

$$\begin{cases} x^{k+1} = \arg \min_x \left\{ Q(x) + r \left\langle x, A^T \left(Ax^k - (b + v^k) \right) \right\rangle \right\}, \\ v^{k+1} = v^k + b - Ax^{k+1}. \end{cases} \quad (9.12)$$

Obviously, the variable v is updated in the exact same way in schemes (9.10) and (9.7), but the computation of the variable x is different between schemes (9.10) and (9.7). Therefore, to further compare these two methods numerically could also be one future work under consideration.

- **Augmented Lagrangian method for Euler's elastica**

The numerical experiments in Chapter 7 suggest the convergence of the proposed augmented Lagrangian method for the Euler's elastica energy, but there is still no theoretical proof for the convergence. Therefore, one future research direction is to derive the theoretical guarantee of the proposed algorithm for Euler's elastica.

It is well-known that the original purpose of Euler's elastica is to have a fundamental role in solving image inpainting problems in [35, 33]. Assume $\tilde{\Omega} \subset \Omega$ is the domain of missing information, the model for inpainting problems is formulated as

$$\begin{aligned} \min_u \int_{\tilde{\Omega}} \left(a + b \left| \nabla \cdot \frac{\nabla u}{|\nabla u|} \right|^\tau \right) |\nabla u| d\Omega, \\ \text{s.t., } u = u_0 \quad \text{on } \partial\tilde{\Omega}, \end{aligned} \quad (9.13)$$

where $\partial\tilde{\Omega}$ denotes the boundary of $\tilde{\Omega}$.

Although we have emphasized the applications of Euler's elastica for image denoising problems in the dissertation, we would like to mention that inpainting problem (9.13) could also be solved efficiently by the augmented Lagrangian formulation in a similar way. Therefore, to extend the augmented Lagrangian formulation of the Euler's elastica model to the image inpainting and the image zooming problems are possible future works.

- **Image restoration and segmentation on triangulated surfaces**

For the TV based image restoration and segmentation on triangulated surfaces, one future work is the stability and convergence analyses of the proposed algorithms. In addition, comparisons among the augmented Lagrangian method and other optimization techniques such as iterative thresholding, Nesterov-based schemes, and primal-dual methods applied to our problems need to be investigated. To completely solve the multi-region segmentation problem in the case where the intensity means of segments are unknown is also worthy of future research. Some special techniques may be introduced to ensure global minimization of the non-convex objective functional.

Besides, the augmented Lagrangian method has been proven efficient in solving the variational models arising from image processing by minimizing sums of convex

functionals subject to linear equality constraints. Therefore, one of the future research directions is to apply the augmented Lagrangian method and operator splitting and decomposition techniques to other well-known models, such as the phase-field model, etc..

- [1] AN Tikhonov. On the stability of inverse problems. *Dokl. Akad. Nauk SSSR*, 39(5):195–198, 1943.
- [2] A.N. Tikhonov, V.I.A. Arsenin, and F. John. *Solutions of ill-posed problems*. Winston Washington, DC:, 1977.
- [3] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.
- [4] P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM Journal on Control and Optimization*, 29(1):119–138, 1991.
- [5] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.
- [6] I. Ekeland and R. Temam. *Convex analysis and variational problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1976.
- [7] R.T. Rockafellar. *Convex analysis*. Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ. Reprint of the 1970 original, Princeton Paperbacks, 1997.

-
- [8] C.R. Vogel. *Computational methods for inverse problems*. Society for Industrial Mathematics, 2002.
 - [9] W.H. Fleming and R. Rishel. An integral formula for total gradient variation. *Archiv der Mathematik*, 11(1):218–222, 1960.
 - [10] A. Chambolle and P.L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2):167–188, 1997.
 - [11] T. Chan, G. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM Journal on Scientific Computing*, 20:1964–1977, 1999.
 - [12] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97, 2004.
 - [13] Y. Huang, M.K. Ng, and Y.W. Wen. A fast total variation minimization method for image restoration. *SIAM Journal on Multiscale Modeling & Simulation*, 7(2):774–795, 2008.
 - [14] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.
 - [15] T. Goldstein and S. Osher. The split Bregman method for L1 regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.
 - [16] X.C. Tai and C. Wu. Augmented Lagrangian method, dual methods and split bregman iteration for rof model. In *Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision*, pages 502–513. Springer-Verlag, 2009.
 - [17] C. Wu and X.C. Tai. Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models. *SIAM Journal on Imaging Sciences*, 3(3):300–339, 2010.

- [18] J. Yang, W. Yin, Y. Zhang, and Y. Wang. A fast algorithm for edge-preserving variational multichannel image restoration. *SIAM Journal on Imaging Sciences*, 2(2):569–592, 2009.
- [19] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for L1-minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences*, 1(1):143–168, 2008.
- [20] Y. Dong, M. Hintermüller, and M. Neri. An efficient primal-dual method for L^1 TV image restoration. *SIAM Journal on Imaging Sciences*, 2(4):1168–1189, 2009.
- [21] M. Zhu, S.J. Wright, and T.F. Chan. Duality-based algorithms for total variation image restoration. *Computational Optimization and Applications*, 47(3):377–400, 2010.
- [22] P. Blomgren and T.F. Chan. Color TV: total variation methods for restoration of vector-valued images. *Image Processing, IEEE Transactions on*, 7(3):304–309, 1998.
- [23] T.F. Chan, S.H. Kang, and J. Shen. Total variation denoising and enhancement of color images based on the CB and HSV color models. *Journal of Visual Communication and Image Representation*, 12(4):422–435, 2001.
- [24] X. Bresson and T.F. Chan. Fast dual minimization of the vectorial total variation norm and applications to color image processing. *Inverse Problems and Imaging*, 2(4):455–484, 2008.
- [25] G. Sapiro and D.L. Ringach. Anisotropic diffusion of multivalued images with applications to color filtering. *Image Processing, IEEE Transactions on*, 5(11):1582–1586, 1996.
- [26] M. Nikolova. Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers. *SIAM Journal on Numerical Analysis*, 40(3):965–994, 2003.

-
- [27] M. Nikolova. A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20(1):99–120, 2004.
- [28] T. Chen and H.R. Wu. Space variant median filters for the restoration of impulse noise corrupted images. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 48(8):784–789, 2001.
- [29] H. Hwang and R.A. Haddad. Adaptive median filters: new algorithms and results. *Image Processing, IEEE Transactions on*, 4(4):499–502, 1995.
- [30] P.E. Ng and K.K. Ma. A switching median filter with boundary discriminative noise detection for extremely corrupted images. *Image Processing, IEEE Transactions on*, 15(6):1506–1516, 2006.
- [31] J.F. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-texture image decomposition modeling, algorithms, and parameter selection. *International Journal of Computer Vision*, 67(1):111–136, 2006.
- [32] T. Chan, A. Marquina, and P. Mulet. High-order total variation-based image restoration. *SIAM Journal on Scientific Computing*, 22(2):503–516, 2000.
- [33] T.F. Chan, S.H. Kang, and J. Shen. Euler’s elastica and curvature-based inpainting. *SIAM Journal on Applied Mathematics*, 63(2):564–592, 2002.
- [34] M. Lysaker, A. Lundervold, and X.C. Tai. Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. *Image Processing, IEEE Transactions on*, 12(12):1579–1590, 2003.
- [35] S. Masnou and J.M. Morel. Level lines based disocclusion. In *Image Processing, International Conference on*, pages 259–263. IEEE, 1998.
- [36] D. Mumford and Center for Intelligent Control Systems (US). *Elastica and computer vision*. Center for Intelligent Control Systems, Massachusetts Institute of Technology, 1991.

-
- [37] L. Ambrosio and S. Masnou. A direct variational approach to a problem arising in image reconstruction. *Interfaces and Free Boundaries*, 5(1):63–82, 2003.
- [38] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *Image Processing, IEEE Transactions on*, 10(8):1200–1211, 2001.
- [39] C. Ballester, V. Caselles, and J. Verdera. Disocclusion by joint interpolation of vector fields and gray levels. *Multiscale Modeling and Simulation*, 2(1):80–123, 2004.
- [40] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.
- [41] T.F. Chan and L.A. Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, 2001.
- [42] L.A. Vese and T.F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.
- [43] X.C. Tai and T.F. Chan. A survey on multiple level set methods with applications for identifying piecewise constant functions. *International Journal of Numerical Analysis and Modeling*, 1(1):25–47, 2004.
- [44] J. Lie, M. Lysaker, and X.C. Tai. A binary level set model and some applications to Mumford-Shah image segmentation. *Image Processing, IEEE Transactions on*, 15(5):1171–1181, 2006.
- [45] E. Bae and XC Tai. Efficient global optimization for the multiphase Chan-Vese model of image segmentation by graph cuts. *UCLA CAM Report 09-53, Department of Mathematics, UCLA, Los Angeles, CA*, 2009.

-
- [46] Y.N. Law, H.K. Lee, and A.M. Yip. A multiresolution stochastic level set method for mumford–shah image segmentation. *Image Processing, IEEE Transactions on*, 17(12):2289–2300, 2008.
- [47] A. Marquina and S. Osher. Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal. *SIAM Journal on Scientific Computing*, 22(4):387–405, 2000.
- [48] CR Vogel. A multigrid method for total variation-based image denoising. *Progress in Systems and Control Theory*, 20:323–323, 1995.
- [49] C.R. Vogel and M.E. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, 1996.
- [50] C.R. Vogel and M.E. Oman. Fast, robust total variation-based reconstruction of noisy, blurred images. *Image Processing, IEEE Transactions on*, 7(6):813–824, 1998.
- [51] Q. Chang, I. Chern, et al. Acceleration methods for total variation-based image denoising. *SIAM Journal on Scientific Computing*, 25(3):982–994, 2003.
- [52] R. Kimmel and I. Yavneh. An algebraic multigrid approach for image analysis. *SIAM Journal on Scientific Computing*, 24(4):1218–1231, 2003.
- [53] S.T. Acton. Multigrid anisotropic diffusion. *Image Processing, IEEE Transactions on*, 7(3):280–291, 1998.
- [54] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling and Simulation*, 4(2):460–489, 2006.
- [55] J. Yang, Y. Zhang, and W. Yin. An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise. *SIAM Journal on Scientific Computing*, 31(4):2842–2865, 2009.

-
- [56] C.L. Wu, J.Y. Zhang, and X.-C. Tai. Augmented Lagrangian method for total variation restoration with non-quadratic fidelity. *Inverse Problems and Imaging*, 5(1):237–261, 2011.
- [57] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001.
- [58] A. Chambolle. Total variation minimization and a class of binary mrf models. In *Energy Minimization Methods in Computer Vision and Pattern Recognition, Lecture Notes in Computer Science*, volume 3757, pages 136–152. Springer, 2005.
- [59] H. Ishikawa. Exact optimization for Markov random fields with convex priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1333–1336, 2003.
- [60] I. Pollak, A.S. Willsky, and Y. Huang. Nonlinear evolution equations as fast and exact solvers of estimation problems. *Signal Processing, IEEE Transactions on*, 53(2):484–498, 2005.
- [61] X.C. Tai and J. Xu. Global and uniform convergence of subspace correction methods for some convex optimization problems. *Mathematics of Computation*, 71(237):105–124, 2002.
- [62] U. Trottenberg, C.W. Oosterlee, and A. Schüller. *Multigrid*. Academic Pr, 2001.
- [63] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34(4):581–613, 1992.
- [64] PS Vassilevski and JG Wade. A comparison of multilevel methods for total variation regularization. *Electronic Transactions on Numerical Analysis*, 6:255–270, 1997.

- [65] K. Chen and X.C. Tai. A nonlinear multigrid method for total variation minimization from image restoration. *Journal of Scientific Computing*, 33(2):115–138, 2007.
- [66] C. Frohn-Schauf, S. Henn, and K. Witsch. Nonlinear multigrid methods for total variation image denoising. *Computing and Visualization in Science*, 7(3):199–206, 2004.
- [67] J. Savage and K. Chen. An improved and accelerated non-linear multigrid method for total-variation denoising. *International Journal of Computer Mathematics*, 82(8):1001–1015, 2005.
- [68] T.F. Chan, K. Chen, and J.L. Carter. Iterative methods for solving the dual formulation arising from image restoration. *Electronic Transactions on Numerical Analysis*, 26(2):299–311, 2007.
- [69] T.F. Chan, K. Chen, and X.C. Tai. Nonlinear multilevel schemes for solving the total variation image minimization problem. *Image Processing Based on Partial Differential Equations, Part IV*, pages 265–288, 2007.
- [70] J. Xu, X.C. Tai, and L.L. Wang. A two-level domain decomposition method for image restoration. *Inverse Problems and Imaging*, 4(3):523–545, 2010.
- [71] X.C. Tai and M. Espedal. Rate of convergence of some space decomposition methods for linear and nonlinear problems. *SIAM Journal of Numerical Analysis*, 35(4):1558–1570, 1998.
- [72] X.C. Tai. Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities. *Numerische Mathematik*, 93(4):755–786, 2003.
- [73] S. Osher, Y. Mao, B. Dong, and W. Yin. Fast linearized Bregman iteration for compressive sensing and sparse denoising. *Communications in Mathematical Sciences*, 8(1):93–111, 2010.

-
- [74] E. Esser. Applications of Lagrangian-based alternating direction methods and connections to split Bregman. *UCLA CAM Report 09-31, Department of Mathematics, UCLA, Los Angeles, CA*, 2009.
- [75] W. Yin. Analysis and generalizations of the linearized Bregman method. *SIAM Journal on Imaging Sciences*, 3(4):856–877, 2010.
- [76] L.M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.
- [77] K.J. Arrow, L. Hurwicz, and H. Uzawa. *Studies in linear and nonlinear programming*. With contributions by H. B. Chenery, S. M. Johnson, S. Karlin, T. Marschak, R. M. Solow. Stanford Mathematical Studies in the Social Sciences, vol. II. Stanford University Press, Stanford, Calif., 1958.
- [78] M.R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.
- [79] M.J.D. Powell and United Kingdom Atomic Energy Authority. “A method for non-linear constraints in minimization problems”. UKAEA, 1967.
- [80] R.T. Rockafellar. A dual approach to solving nonlinear programming problems by unconstrained optimization. *Mathematical Programming*, 5(1):354–373, 1973.
- [81] R. Glowinski and P. Le Tallec. *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*. Society for Industrial Mathematics, 1989.
- [82] DM Greig, BT Porteous, and AH Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B (Methodological)*, 51(2):271–279, 1989.

-
- [83] Y.Y. Boykov and M.P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In *Computer Vision, Eighth IEEE International Conference on*, volume 1, pages 105–112. IEEE, 2001.
 - [84] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.
 - [85] H. Ishikawa and D. Geiger. Segmentation by grouping junctions. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 125–131. IEEE, 1998.
 - [86] J. Darbon and M. Sigelle. Image restoration with discrete constrained total variation part I: Fast and exact optimization. *Journal of Mathematical Imaging and Vision*, 26(3):261–276, 2006.
 - [87] J. Darbon and M. Sigelle. Image restoration with discrete constrained total variation part II: Levelable functions, convex priors and non-convex cases. *Journal of Mathematical Imaging and Vision*, 26(3):277–291, 2006.
 - [88] A. Chambolle and J. Darbon. On total variation minimization and surface evolution using parametric maximum flows. *International Journal of Computer Vision*, 84(3):288–307, 2009.
 - [89] A.V. Goldberg and R.E. Tarjan. A new approach to the maximum-flow problem. *Journal of the Association for Computing Machinery*, 35(4):921–940, 1988.
 - [90] L. Ford and D. Fulkerson. *Flows in networks*. Princeton University Press, 1962.
 - [91] D. Goldfarb and W. Yin. Parametric maximum flow algorithms for fast total variation minimization. *SIAM Journal on Scientific Computing*, 31(5):3712–3743, 2009.
 - [92] G. Gallo, M.D. Grigoriadis, and R.E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–55, 1989.

-
- [93] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *Computer Vision, Ninth IEEE International Conference on*, volume I, pages 26–33, 2003.
- [94] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):147–159, 2004.
- [95] F. Guichard and J. Morel. “Mathematical morphology” almost everywhere. In *Mathematical Morphology: Proceedings of the VIth International Symposium—ISMM 2002: Sydney, 3-5 April, 2002*. Citeseer.
- [96] J. Darbon. Total variation minimization with L1 data fidelity as a contrast invariant filter. In *Image and Signal Processing and Analysis, Proceedings of the 4th International Symposium on*, pages 221–226. IEEE, 2005.
- [97] T.F. Chan and T.P. Mathew. Domain decomposition algorithms. *Acta Numerica*, 3(1):61–143, 1994.
- [98] X.C. Tai. Parallel function decomposition and space decomposition methods: Part II. space decomposition. In *Beijing Mathematics*. Citeseer, 1995.
- [99] M.K. Ng, G. Qiu, and A.M. Yip. Numerical methods for interactive multiple-class image segmentation problems. *International Journal of Imaging Systems and Technology*, 20(3):191–201, 2010.
- [100] S. Alliney. A property of the minimum vectors of a regularizing functional defined by means of the absolute norm. *Signal Processing, IEEE Transactions on*, 45(4):913–917, 1997.
- [101] Y. Dong, M. Hintermüller, and M. Neri. A primal-dual method for L1 TV image denoising. *SIAM Journal on Imaging Sciences*, 2:577–613, 2009.
- [102] A. Dervieux and F. Thomasset. A finite element method for the simulation of a rayleigh-taylor instability. In *Approximation Methods for Navier-Stokes Problems*, volume 1, pages 145–158, 1980.

-
- [103] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.
 - [104] C. Li, C. Xu, C. Gui, and M.D. Fox. Level set evolution without re-initialization: A new variational formulation. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, volume 1, pages 430–436. IEEE, 2005.
 - [105] J.E. Solem, N.C. Overgaard, and A. Heyden. Initialization techniques for segmentation with the chan-veese model. In *Pattern Recognition, 18th International Conference on*, volume 2, pages 171–174. IEEE, 2006.
 - [106] J. Lie, M. Lysaker, and X.C. Tai. A variant of the level set method and applications to image segmentation. *Mathematics of Computation*, 75(255):1155–1174, 2006.
 - [107] X.C. Tai, O. Christiansen, P. Lin, and I. Skjælaaen. Image segmentation using some piecewise constant level set methods with MBO type of projection. *International Journal of Computer Vision*, 73(1):61–76, 2007.
 - [108] A. Delong and Y. Boykov. A scalable graph-cut algorithm for nd grids. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
 - [109] Y. Boykov, V. Kolmogorov, D. Cremers, and A. Delong. An integral solution to surface evolution PDEs via geo-cuts. *Lecture Notes in Computer Science*, 3953:409–422, 2006.
 - [110] J. Darbon. A note on the discrete binary mumford-shah model. *Lecture Notes in Computer Science*, 4418:283–294, 2007.
 - [111] N. El Zehiry, S. Xu, P. Sahoo, and A. Elmaghraby. Graph cut optimization for the mumford-shah model. In *The Seventh IASTED International Conference on Visualization, Imaging and Image Processing*, pages 182–187. ACTA Press, 2007.

- [112] N. Komodakis, G. Tziritas, and N. Paragios. Fast, approximately optimal solutions for single and dynamic mrfs. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 1–8. IEEE, 2007.
- [113] E. Bae and X.C. Tai. Graph cuts for the multiphase mumford-shah model using piecewise constant level set methods. In *SSVM'09: Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, 2009.
- [114] G. Chung and L. Vese. Energy minimization based segmentation and denoising using a multilayer level set approach. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 3757, pages 439–455. Springer, 2005.
- [115] Y.M. Jung, S.H. Kang, and J. Shen. Multiphase image segmentation via modica-mortola phase transition. *SIAM Journal on Applied Mathematics*, 67(5):1213–1232, 2007.
- [116] RD Velasco. Thresholding using the ISODATA clustering algorithm. *Systems Man and Cybernetics, IEEE Transactions on*, 10(11):771–774, 1980.
- [117] J.F. Aujol. Some first-order algorithms for total variation based image restoration. *Journal of Mathematical Imaging and Vision*, 34(3):307–327, 2009.
- [118] E. Esser, X. Zhang, and T. Chan. A general framework for a class of first order primal-dual algorithms for TV minimization. *SIAM Journal on Imaging Sciences*, 3(4):1015–1046, 2010.
- [119] R.Q. Jia and H. Zhao. A fast algorithm for the total variation model of image denoising. *Advances in Computational Mathematics*, 33(2):231–241, 2010.
- [120] M. Nikolova and M.K. Ng. Analysis of half-quadratic minimization methods for signal and image recovery. *SIAM Journal on Scientific computing*, 27(3):937–966, 2006.

- [121] X. Zhang, M. Burger, X. Bresson, and S. Osher. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM Journal on Imaging Sciences*, 3(3):253–276, 2010.
- [122] X. Zhang, M. Burger, and S. Osher. A unified primal-dual algorithm framework based on Bregman iteration. *Journal of Scientific Computing*, 46(1):20–46, 2011.
- [123] P.L. Combettes and V.R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200, 2006.
- [124] J.J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin/Société mathématique de France, Paris*, 93(2):273–299, 1965.
- [125] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.
- [126] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- [127] J.F. Aujol and S.H. Kang. Color image decomposition and restoration. *Journal of Visual Communication and Image Representation*, 17(4):916–928, 2006.
- [128] J. Weickert. *Anisotropic diffusion in image processing*. Citeseer, 1998.
- [129] M. Lysaker and X.C. Tai. Iterative image restoration combining total variation minimization and a second-order functional. *International Journal of Computer Vision*, 66(1):5–18, 2006.
- [130] T.F. Chan and S. Esedoglu. Aspects of total variation regularized L1 function approximation. *SIAM Journal on Applied Mathematics*, 65(5):1817–1837, 2005.
- [131] E. Bae, J. Shi, and X.C. Tai. Graph cuts for curvature based image denoising. *Image Processing, IEEE Transactions on*, (20):1199–1210, 2010.

-
- [132] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order mrfs. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2985–2992. IEEE, 2009.
- [133] RT Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1(2):97–116, 1976.
- [134] X.C. Tai, J. Hahn, and G.J. Chung. A fast algorithm for Euler’s elastica model using augmented Lagrangian method. *SIAM Journal on Imaging Sciences*, 4(1):313–344, 2011.
- [135] R. Lai and T.F. Chan. A framework for intrinsic image processing on surfaces. *Computer Vision and Image Understanding*, 115(12):1647–1661, 2010.
- [136] J. Hahn, G.J. Chung, Y. Wang, and X.C. Tai. Fast algorithms for p-elastica energy with the application to image inpainting and curve reconstruction. In *In: SSVM’11: Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, 2011.
- [137] J.L. Carter. Dual methods for total variation based image restoration. *UCLA CAM Report 02-13, Department of Mathematics, UCLA, Los Angeles, CA*, 2002.
- [138] O.V. Michailovich. An iterative shrinkage approach to total-variation image restoration. *Image Processing, IEEE Transactions on*, (20):1281–1299, 2011.
- [139] R.H. Chan, T.F. Chan, L. Shen, and Z. Shen. Wavelet algorithms for high-resolution image reconstruction. *SIAM Journal on Scientific Computing*, 24(4):1408–1432, 2002.
- [140] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

-
- [141] M.A.T. Figueiredo and R.D. Nowak. An em algorithm for wavelet-based image restoration. *Image Processing, IEEE Transactions on*, 12(8):906–916, 2003.
 - [142] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
 - [143] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.
 - [144] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Fast geodesic active contours. *Image Processing, IEEE Transactions on*, 10(10):1467–1475, 2001.
 - [145] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*. Springer, 2002.
 - [146] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr. Diffusion snakes: Introducing statistical shape knowledge into the mumford-shah functional. *International Journal of Computer Vision*, 50(3):295–313, 2002.
 - [147] B. Appleton and H. Talbot. Globally minimal surfaces by continuous maximal flows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(1):106–118, 2006.
 - [148] M. Nikolova, S. Esedoglu, and T.F. Chan. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, 2006.
 - [149] X. Bresson, S. Esedoglu, P. Vandergheynst, J.P. Thiran, and S. Osher. Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision*, 28(2):151–167, 2007.
 - [150] C. Zach, D. Gallup, J.M. Frahm, and M. Niethammer. Fast global labeling for real-time stereo using multiple plane sweeps. In *Vision, Modeling, and Visualization*, 2008.

- [151] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A convex formulation of continuous multi-label problems. In *Proceedings of the 10th European Conference on Computer Vision: Part III*, pages 792–805. Springer-Verlag, 2008.
- [152] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An algorithm for minimizing the mumford-shah functional. In *Computer Vision, IEEE 12th International Conference on*, pages 1133–1140. IEEE, 2009.
- [153] T. Pock, A. Chambolle, D. Cremers, and H. Bischof. A convex relaxation approach for computing minimal partitions. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 810–817. IEEE, 2009.
- [154] J. Lellmann, J. Kappes, J. Yuan, F. Becker, and C. Schnörr. Convex multi-class image labeling by simplex-constrained total variation. In *Proceedings Second International Conference on Scale Space and Variational Methods in Computer Vision*, pages 150–162. Springer-Verlag, 2009.
- [155] E. Bae, J. Yuan, and X.C. Tai. Global minimization for continuous multiphase partitioning problems using a dual approach. *International Journal of Computer Vision*, 92(1):112–129, 2010.
- [156] E.S. Brown, T.F. Chan, and X. Bresson. A convex approach for multi-phase piecewise constant mumford-shah image segmentation. *UCLA CAM Report 09-66, Department of Mathematics, UCLA, Los Angeles, CA*, 2009.
- [157] T. Goldstein, X. Bresson, and S. Osher. Geometric applications of the split bregman method: Segmentation and surface reconstruction. *Journal of Scientific Computing*, 45(1):272–293, 2010.
- [158] E.S. Brown, T.F. Chan, and X. Bresson. A convex relaxation method for a class of vector-valued minimization problems with applications to mumford-shah segmentation. *UCLA CAM Report 10-43, Department of Mathematics, UCLA, Los Angeles, CA*, 2010.

-
- [159] E.S. Brown, T.F. Chan, and X. Bresson. Globally convex chan-veye image segmentation. *UCLA CAM Report 10-44, Department of Mathematics, UCLA, Los Angeles, CA*, 2010.
 - [160] J. Lellmann, F. Becker, and C. Schnorr. Convex optimization for multi-class image labeling with a novel family of total variation based regularizers. In *Computer Vision, IEEE 12th International Conference on*, pages 646–653. IEEE, 2009.
 - [161] J. Lellmann and C. Schnörr. Continuous multiclass labeling approaches and algorithms. *Universitätsbibliothek der Universität Heidelberg*, 2010.
 - [162] C.L. Wu, J.S. Deng, and F.L. Chen. Diffusion equations over arbitrary triangulated surfaces for filtering and texture applications. *Visualization and Computer Graphics, IEEE Transactions on*, 14(3):666–679, 2008.
 - [163] C.L. Wu, J.S. Deng, F.L. Chen, and X.C. Tai. Scale-space analysis of discrete filtering over arbitrary triangulated surfaces. *SIAM Journal on Imaging Sciences*, 2(2):670–709, 2009.
 - [164] A. Delaunoy, K. Fundana, E. Prados, and A. Heyden. Convex multi-region segmentation on manifolds. In *Computer Vision, IEEE 12th International Conference on*, pages 662–669. IEEE, 2009.
 - [165] A.N. Hirani. *Discrete exterior calculus*. PhD thesis, California Institute of Technology, 2003.
 - [166] C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$. *Journal of Optimization Theory and Applications*, 50(1):195–200, 1986.
 - [167] J.F. Cai, E.J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal of Optimization*, 20(4):1956–1982, 2010.

- ① Duan, Y. and Tai, X.C., Domain decomposition methods with Graph cuts algorithms for total variation minimization, *Advances in Computational Mathematics*, Accepted.
- ② Tai, X.C. and Duan, Y., Domain decomposition methods with Graph cuts algorithms for image segmentation, *International Journal of Numerical Analysis and Modeling*, 8(1):137–156, 2011.
- ③ Duan, Y., and Tai, X.C., A Fixed-point Augmented Lagrangian Method for Total Variation Minimization Problems, Submitted, 2011.
- ④ Duan, Y., Wang, Y., Tai, X.C. and Hahn J., A Fast Augmented Lagrangian Method for Eulers Elastica Model, In: 3rd International Conference on Scale Space and Variational Methods in Computer Vision.
- ⑤ Wu, C., Zhang, J., Duan, Y., and Tai, X.C., Augmented Lagrangian Method for Total Variation Based Image Restoration and Segmentation over Triangulated Surfaces, *Journal of Scientific Computing*, Accepted.