

单位代码： 10293 密 级：

南京邮电大学

硕 士 学 位 论 文



论文题目： 全变分模型图像复原的一阶前向后向优化算法研究

学 号 1014051322

姓 名 陈少利

导 师 杨敏 副教授

学 科 专 业 模式识别与智能系统

研 究 方 向 图像处理与模式分类

申请学位类别 工学硕士

论文提交日期 二零一七年三月

First - order Forward backward Algorithm for Image Restoration Based on Total Variational Model

Thesis Submitted to Nanjing University of Posts and
Telecommunications for the Degree of
Master of Engineering



By

Chen Shaoli

Supervisor: Prof. Yang Min

March 2017

南京邮电大学学位论文原创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得南京邮电大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

本人学位论文及涉及相关资料若有不实，愿意承担一切相关的法律责任。

研究生学号：_____ 研究生签名：_____ 日期：_____

南京邮电大学学位论文使用授权声明

本人授权南京邮电大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档；允许论文被查阅和借阅；可以将学位论文的全部或部分内容编入有关数据库进行检索；可以采用影印、缩印或扫描等复制手段保存、汇编本学位论文。本文电子文档的内容和纸质论文的内容相一致。论文的公布（包括刊登）授权南京邮电大学研究生院办理。

涉密学位论文在解密后适用本授权书。

研究生签名：_____ 导师签名：_____ 日期：_____

摘要

图像噪声和图像模糊往往影响了图像的后期处理，例如图像分割、特征提取、目标跟踪等。因此有效的去除图像噪声与去模糊的图像复原成为数字图像处理领域的关键问题。

由于全变分模型能够很好的保留图像边缘信息从而获得更好的图像复原效果。于是本文着重研究全变分模型下的图像复原算法。根据有界变分图像处理，将全变分图像复原问题分解为两个凸函数的最小化问题，解决此类问题，可通过一阶和二阶方法，由于二阶方法迭代计算量较大，因而不适用于求解大规模数据的图像复原问题。而一阶方法的求解仅仅涉及到梯度和目标函数值信息，因此算法的计算量小、迭代效率较高，适用于求解大规模的实际问题。因此本文研究全变分图像复原的一阶方法。基于以上介绍，本文的主要研究内容为以下几点：

首先，本文应用前向后向分裂算法求解全变分图像复原模型，针对该算法的固定步长限制了算法的收敛速率的不足之处，提出一种自适应步长选择的快速前向后向分裂算法。该算法将前向后向分裂算法结合了 Barzilai-Borwein 自适应步长选择算子。在算法进行迭代时，更新选择步长因子，从而获得更快的算法收敛效果。数值实验表明，自适应步长快速前向后向分裂算法不仅提高了算法的收敛速率，同时提高了图像复原的质量。

其次，对改进的快速前向后向分裂算法做进一步研究，发现采用了 Barzilai-Borwein 谱步长进行更新的新算法在数值实验中并不总能保证算法的严格收敛，针对该问题，将非单调线性搜索的思想引入到改进快速前向后向分裂算法中，提出了一个非单调线性搜索快速前向后向分裂算法，避免了算法进入局部最优。从而使得算法在运算精度和时间复杂度上达到最优。数值实验表明算法的有效性。

最后，通过对快速前向后向分裂算法的深入研究，实验表明目标函数呈现一定的“震荡”现象，即目标函数值增加，因此减缓了目标函数的收敛。研究发现该算法对判定因子的选择具有很强的敏感性。据此，本文提出了自适应重启动快速前向后向分裂算法。通过加入判定因子，适时重启算法，从而使目标函数呈现单调下降现象。数值实验表明，自适应重启动前向后向改进算法加快了算法的收敛，减少了目标函数的震荡，提高了算法对于图像去噪和图像去模糊的复原质量。

关键词：全变分，图像复原，前向后向分裂算法，Barzilai-Borwein 算法，非单调线性搜索，重启动算法

Abstract

Image noise and image blur often affect the post-processing image, such as image segmentation, feature extraction, target tracking and so on. So the effective removal of image noise and image blur becomes a key problem in the field of digital image processing. The purpose of image denoising and restoration is to recover the original image with clear boundary and clear texture as far as possible from the degraded image. Therefore, the key to image denoising and image restoration is to preserve the edge information Texture.

Because the full variational model can preserve the image edge information to obtain better image restoration effect. So this paper focuses on the image restoration algorithm under the full variational model. According to the bounded variational image processing, the whole variational image restoration problem is decomposed into two convex function minimization problems. The first and second order methods can be solved. Because the second-order iterative computation is large, It is not suitable for solving image restoration problems of large-scale data. The first-order method only involves the gradient and the objective function value information, so the algorithm is small in computation and iterative efficiency is high, which is suitable for solving large-scale practical problems. Therefore, this paper studies the first-order method of full variational image restoration. Based on the above description, the main content of this paper for the following points:

Firstly, this paper applies the forward backward splitting algorithm to solve the full variational image restoration model. The fixed step size of the algorithm limits the shortcoming of the convergence rate of the algorithm, and proposes a fast forward backward Split algorithm. The algorithm combines the forward backward splitting algorithm with the Barzilai-Borwein adaptive step selection operator. In the algorithm iteration, the update step size factor is selected to obtain a faster algorithm convergence effect. Numerical experiments show that the adaptive step size fast forward backward splitting algorithm not only improves the convergence rate of the algorithm, but also improves the quality of image restoration.

Secondly, the improved fast forward backward splitting algorithm is further studied. It is found that the new algorithm using Barzilai-Borwein spectral step length does not always guarantee the strict convergence of the algorithm in numerical experiments. In this paper, The idea of monotonic linear search is introduced into the improved fast forward backward splitting algorithm, and a fast forward backward splitting algorithm is proposed for nonmonotone linear search, which avoids the

local optimization of the algorithm. So that the algorithm in the computing accuracy and time complexity to achieve the best. Numerical experiments administrate that the effectiveness of the proposed algorithm.

Finally, through the in-depth study of the fast forward backward splitting algorithm, the objective function exhibits a certain "concussion" phenomenon: the objective function value increases and the convergence of the objective function is slowed down. It is found that the algorithm has a strong sensitivity to the selection of decision factors. Based on this, this paper proposes a fast forward backward splitting algorithm for adaptive restart. By adding the judgment factor to judge, timely restart the algorithm, so that the objective function presents a monotonically decreasing phenomenon. Numerical experiments show that the adaptive back-to-back algorithm improves the convergence of the algorithm, reduces the oscillation of the objective function, and improves the recovery quality of the algorithm for image denoising and image deblur.

Key words:Total Variation, Image Restoration, Forward Backward Splitting Algorithm, Barzilai-Borwein Algorithm, Restart Algorithm, Nonmonotone Linear Search Algorithm

目录

专用术语注释表.....	IV
第一章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 图像复原的数学模型及研究现状.....	1
1.2.1 图像复原模型.....	2
1.2.2 图像复原相关算法.....	4
1.3 研究内容与结构安排.....	5
第二章 相关背景知识介绍.....	7
2.1 全变分模型.....	7
2.2 全变分图像复原模型.....	8
2.3 一般优化问题.....	8
2.3.1 凸优化理论.....	8
2.3.2 一般优化模型.....	10
2.3.3 基于梯度的算法.....	10
2.4 图像重建评价标准.....	12
2.5 本章小结.....	14
第三章 全变分模型自适应步长加速前向后向分裂算法.....	15
3.1 前向后向分裂算法.....	15
3.1.1 前向后向分裂算法的一般框架.....	15
3.1.2 前向后向分裂算法的收敛性分析.....	18
3.1.3 混合残差停止准则.....	19
3.2 Borzilai-Borwein 自适应谱步长算子.....	20
3.3 自适应步长前向后向分裂算法.....	21
3.4 实验结果和分析.....	22
3.4.1 基于全变分模型的图像修复.....	23
3.4.2 基于全变分模型的图像去噪.....	24
3.5 本章小结.....	27
第四章 全变分模型非单调线性搜索前向后向分裂算法.....	28
4.1 非单调线性搜索策略.....	28
4.2 非单调线性搜索加速前向后向分裂算法.....	29
4.3 非单调线性搜索前向后向算法的收敛性验证.....	30
4.4 数值实验.....	31
4.4.1 基于全变分模型的图像修复.....	31
4.4.2 基于全变分模型的图像去噪.....	33
4.5 本章小结.....	35
第五章 全变分模型自适应重启加速前向后向分裂算法.....	36
5.1 一阶加速方法.....	36
5.1.1 加速近邻梯度算法.....	36
5.1.2 快速前向后向分裂算法.....	38
5.2 自适应重启加速算子.....	39
5.3 自适应重启加速前向后向分裂算法.....	40
5.4 实验结果与分析.....	41
5.4.1 基于全变分模型的图像修复.....	41
5.4.2 基于全变分模型的图像去噪.....	43

5.5 本章小结.....45

第六章 总结与展望..... 46

6.1 总结.....46

6.2 展望.....47

参考文献.....49

附录 1 攻读硕士学位期间撰写的论文..... 51

致谢.....52

专用术语注释表

符号说明:

R^n	表示一个 n 维实数空间
x^T 或 A^T	表示 x 或者矩阵 A 的转置
$\ \cdot\ $	表示向量的欧几里得范数或矩阵范数
$\ \cdot\ _{TV}$	表示全变分模型
∇	表示离散梯度
I	表示单位矩阵

缩略词说明:

TV	Total Variation	全变分模型
FISTA	Fast Iterative Shrinkage-Thresholding Algorithm	快速迭代收缩阈值算法
FBS	Forward-Backward Splitting Method	前向后向分裂算法
BB	Barzilai-Borwein	BB 自适应步长算法
FBB	Forward-Backward and Barzilai-Borwein Algorithm	快速谱变步长前向后向算法
NLFBB	Nonmonotone line search FBB	非单调线性搜索前向后向分裂算法
RSFISTA	Restart FISTA	自适应重启快速前向后向分裂算法

第一章 绪论

1.1 研究背景及意义

现实生活中，人们获取的信息中 80%来自于图像与视频，而视频本身又是在时间的基础上叠加起来的一帧帧的图像，可以说图像信息占据了人们生活信息来源的绝大部分。因此，一幅清晰的图像对于人们获取高质量信息具有重要意义，尤其在计算机技术发展如此迅速的今天，各种数码产品日益普及，图像已经成为诸多学科研究视觉感知的重要工具。然而图像在获取的过程中由于光学系统的像差问题、光学成像衍射、成像系统的非线性畸变、成像过程的相对运动、大气的湍流效应、环境的随机噪声等种种原因，导致获取到的图像质量出现退化现象。为了改善图像质量，获得高质量、高清晰度的图像，图像复原技术成为国内外科学家不断的努力研究的重要课题。

图像复原(亦可称之为图像恢复与重建)，是图像处理中的一项重要研究内容。图像复原包括图像去噪、图像修复、图像去模糊等等，本文主要考虑图像去噪和图像修复。解决图像复原问题，往往涉及到大量数据集和未知信息。一般地，图像复原的方法是利用图像退化的某种先验信息，建立起图像的退化模型^[1]，然后对该退化模型进行反向演算，以复原原始图像。于是建立图像的退化模型则是重要任务。

全变分模型^[2]是目前解决图像复原问题中最经典也是最成功的模型之一。该模型是在 1992 年由 Rodin,Osher,Fatemi 提出，之后对图像处理所建立的模型都是在此模型基础上提出的。全变分模型与应用图像本身稀疏性求解复原图像的方法相比，将稀疏性应用了梯度上，得到了噪声图像的分段近似解，该模型对于图像复原问题最大的优点是很好地保留图像的边缘信息。近年来，国内外学者在全变分模型基础上提出了许多改进的算法。

全变分模型图像复原问题最大的难点在于该模型的不可微性，以及图像数据的高维性。解决全变分图像复原问题，前向后向分裂算法(Forward-Backward Splitting,FBS)^[3]是解决该类问题的基础方法。

1.2 图像复原的数学模型及研究现状

1.2.1 图像复原模型

图像退化过程模型的描述是保证精确复原图像的关键。图像复原建立一定的退化模型，然后对该模型按照逆问题进行求解，以最大的保真度，恢复到退化前的状态，即图像恢复其实就是沿着图像降质的逆过程进行求解的。因此，建立图像退化的数学模型，一直以来都是图像处理领域中最基本也是最关键的研究内容。

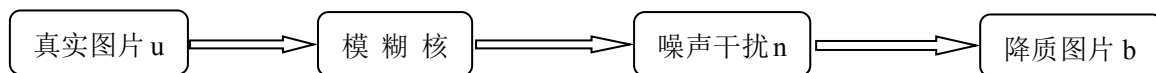


图 1.1 图像退化过程

由于造成图像退化的原因有很多种，且比较复杂，根据图 1.1 的图像退化过程，抽象出的经典图像退化模型如下式所示：

$$b = Au + n \quad (1.1)$$

其中 $A \in \mathbb{R}^{m \times n}$ 是模糊算子， $b \in \mathbb{R}^m$ 是模糊图像， n 是未知的噪声向量， $u \in \mathbb{R}^n$ 是待恢复的未知的图像信号。图像恢复要解决的问题就是从上面的退化模型中获得原始图像的最佳估计值 u 。

根据经典图像复原技术的发展，解决以上的图像复原技术分为 3 个阶段：直接复原法、正则化方法^[4]和自适应法。

直接复原法包括了代数法和频率域法，是数字图像复原技术发展早期研究中常用的方法。常用逆滤波技术^[5]来解决此类问题。其中维纳滤波器是目前基本的滤波方法之一。该法将图像复原问题转化为最小二乘的最佳估计问题。

$$\min_x \{f(u) = \|Au - b\|^2\} (m = n) \quad (1.2)$$

其中当 A 是非奇异值时，最小二乘法的解为： $A^{-1}b$ ，然而由于图像复原问题中矩阵 A 常常是病态的且是不可逆的，因此导致最小二乘法的解具有很大的范数值（有时是无意义的值）。因此此类方法并不是求解图像复原算法的最优方法，仅在需求不高的情况下应用。

为了解决上述问题，人们研究出了很多的方法，其中正则化方法成为得到问题稳定解的经典方法，即利用“近似”良态解取代原始病态问题的解。Tikhonov 和 Arsenin^[6]在 1977 年提出的 Tikhonov 正则化方法是一个在 L_2 空间上求解最小化方差的近似问题，该方法可以兼顾问题的精确性和平顺性，因而得到了专家学者的广泛的研究。

Tikhonov 正则化方法通过迭代以下子问题产生迭代序列 $\{u_k, k=0, 1, \dots\}$ 并高效求解：

$$\min_x \frac{\lambda}{2} \|Au - b\|^2 + c(u) \quad (1.3)$$

其中 $\|\cdot\|$ 表示欧几里得范数，(1.3) 中的第一项则为数据拟合项，保证可以得到近似解。第二项 $c(u)$ 即为引入的正则化项，它控制着解的范数，正则化参数 $\lambda > 0$ ，主要起调节拟合项和正则项的作用。当 A 为单位矩阵时，(1.3) 退化为图像去噪问题。

l_1 正则项是解决图像处理问题的经典模型：

$$\min \{f(u) \equiv \|Au - b\|^2 + \lambda \|u\|_1\} \quad (1.4)$$

其中 $\|u\|_1$ 表示矩阵 u 的元素的绝对值之和，即： $\|u\|_1 = \sum_i |u_i|$ ，向量 u 包含位置图像的系数。

图像可以用 l_1 范数问题解决图像问题的根本原因在于图像在小波域可以稀疏表示，并且可以减少 (1.4) 式的解的稀疏性。另一个重要的原因是 l_1 正则项较 l_2 正则项对图像中锋利的边缘的敏感性较低，该模型为著名的 l_2 - l_1 问题（即：basis pursuit denoising(BPDN)）^[7]。

然而由于 (1.4) 式得到的图像并不是很好，特别是当 x 具有不连续点或尖的边界情况下，图像会有 Gibbs 现象，即光滑掉不连续点或在尖的边界震荡的特别厉害。因此为了保留图像的边缘信息，1992 年，Rudin, Osher 和 Fatemi 提出了著名的 ROF 模型，即全变分模型^[2]，该模型从能量极小化的角度把全变差半范数作为正则化项引入到图像恢复模型中。其定义式如下：

$$\|u\|_{TV} = \begin{cases} |\nabla_x u| + |\nabla_y u| & (\text{各向异性}) \\ \sum_i \sqrt{(\nabla_x u)_i^2 + (\nabla_y u)_i^2} & (\text{各向同性}) \end{cases} \quad (1.5)$$

简言之： $\|u\|_{TV}$ 可表示为 $\sum_{i=1}^{mn} \|D_i u\|_q$ ， $D_i u \in \mathbb{R}^2$ 表示 u 关于像素 i 的离散化梯度，因此，当 $q=1$ 时则为各向异性，当 $q=2$ 时表示各项同性。由于全变分模型不要求解是连续的，因此克服了图像恢复不能很好的保留图像的边缘信息的缺陷，于是解决图像复原问题的全变分模型就是目前最为经典、最成功的图像恢复模型之一。从此以后，很多的图像复原模型也是在该模型之上建立的。于是本文研究的重点即为图像恢复的全变分模型：

$$\min \{f(u) \equiv \|Au - b\|^2 + \lambda \|u\|_{TV}\} \quad (1.6)$$

自适应复原法本质上也是正则化方法，它是在正则化的基础上考虑了局部特性，在外部条件变化的情况下，克服了正则化方法的全局性限制。

目前随着数字图像处理技术与控制理论的发展，一些新的图像复原的技术也在不断涌现，小波分析法^[8]是目前普遍研究与应用的热点，小波变换具有多分辨特性、时频局部特性，很好的解决了经典方法中的噪声抑制、约束最小正则化、自适应等问题。小波方法虽然在处理图像噪声和保留图像细节信息上取得了良好的效果，然而图像边缘常出现 Gibbs 现象，因而

产生震荡现象。

1.2.2 图像复原相关算法

近几年，随着图像复原算法的研究，专家们针对图像复原的不同模型提出了很多新的方法。

解决 l_1 范式问题，可以将其归结为一个二阶锥规划问题然后利用内点法解决。然而图像处理中，由于图像数据量大及数据密度大等问题，因此产生了最简单的基于梯度的算法，该方法主要通过计算矩阵 A 和 A^T 向量相乘问题，例如文献^[9]中将问题（1.3）重塑为一个盒约束二次问题，并利用梯度映射算法^[10]。

ISTA（iterative shrinkage/thresholding algorithm）^[11,12]是基于梯度的算法中最受欢迎的算法之一，该算法建立以下子问题：

$$\min(z - u_k)^T \nabla f(u_k) + \frac{\partial_k}{2} \|z - u_k\|_2^2 + \tau c(z) \quad (1.7)$$

其中 $\partial_k \in \mathbb{R}^+$ ，通过高效迭代上式求解，计算求解 ∇f 及式（1.4）比利用其它算法求解原始目标函数更加简洁。（1.4）式的一个等价算式为：

$$\min \frac{1}{2} \|z - w_k\|_2^2 + \frac{\tau}{\partial_k} c(z)$$

其中 $w_k = u_k - \frac{1}{\partial_k} \nabla f(u_k)$ 。

ISTA 可以被追溯为近似前向后向分裂迭代算法（proximal forward-backward iterative scheme）^[3]。

求解全变分模型复原问题，引入算子分裂的方法，分裂 Bregman 算法^[13]就是将 Bregman 迭代结合了算子分离的思想得到：

$$\begin{aligned} (u_{k+1}, w_{k+1}) &= \arg \min \{F(u) + H(w) + \frac{\lambda}{2} \|Au - w + \lambda^k\|_2^2\} \\ p_{k+1} &= p_k + Au_{k+1} - w_{k+1} \end{aligned}$$

为了降低求解的难度，交替最小化方法（ADMM）^[14]算法将上述算法分裂为如下形式：

$$\begin{aligned} u_{k+1} &= \arg \min \{F(u) + \frac{\lambda}{2} \|Au - w_k + p_k\|_2^2\} \\ w_{k+1} &= \arg \min \{H(w) + \frac{\lambda}{2} \|Au - w + p_k\|_2^2\} \end{aligned}$$

求解凸优化问题的另一种有效算法是快速迭代收缩阈值算法（Fast Iterative Threshold Algorithm）^[15]。该算法在前述 ISTA 算法的基础上改进了算子的全局收敛性，提高了算法的

收敛速度。具体迭代公式如下：

$$\begin{aligned} u_{k+1} &= \arg \min \{prox(y_k - \tau \nabla f(y_k))\} \\ \partial_{k+1} &= (1 + \sqrt{1 + 4\partial_k^2}) / 2 \\ y_k &= u_k + ((\partial_k - 1) / \partial_{k+1})(u_k - u_{k-1}) \end{aligned} \quad (1.8)$$

1.3 研究内容与结构安排

基于全变分模型的图像复原问题是图像处理、机器视觉领域中的经典问题，本文通过对现有图像复原前向后向分裂算法的研究，通过对不同参数的更新优化，提出了更为有效的解决全变分图像复原的一阶算法。具体的主要工作如下：

(1) 原始前向后向分裂算法 (FBS) [3] 对参数的选择具有很强的敏感性，固定步长下，算法收敛速率受限制，Barzilai-Borwein 算法 [16] 利用前两步迭代点信息更新步长因子，从而加快迭代收敛速率。本文将两者结合提出一种自适应谱步长前向后向分裂算法 (FBB)。并将其应用于全变分模型图像复原中，多次试验证明产生的新算法加快了前向后向算法求解图像复原的效率，同时提高了图像复原的质量。

(2) 改进的自适应谱步长的 FBB 算法提高了前向后向分裂算法的收敛速率，然而该算法并不能完全保证算法的收敛。针对该问题加入非单调线性搜索算法 [17]，提出一种非单调线性搜索加速前向后向分裂算法 (NLFBB)。该算法在自适应迭代的基础上，加入非单调线性搜索，通过条件判断条件满足时进行算法迭代，否则减小步长因子，直到满足条件。以此保证算法的绝对收敛。数值实验验证了改进算法提高了算法的收敛速率。

(3) 原始快速前向后向分裂算法应用到图像复原算法中，通过观察目标函数值发现：目标函数的变化并非单调下降，而是出现“震荡”现象。理论证明，算法的收敛性依赖函数的 Lipschitz 常数，然而实际应用中，人们很难获取有关 Lipschitz 常数的准确信息。为了防止目标函数的出现“震荡”现象，同时减少算法的计算量，本文根据自适应重启算子 [18] 改进算法的迭代机制，提出一种自适应重启快速前向后向分裂算法 (RSFISTA)。改进算法通过判断迭代方向朝坏的方向发展时，则重新启动算法的执行，抹除之前迭代信息对求解后续迭代点的影响。数值实验表明，该改进方法具有较好的收敛速率。

现根据以上所列研究内容，则全文结构安排如下：

第一章绪论：介绍了图像去噪和图像恢复的研究背景及其复原模型和相关算法的研究现状。

第二章相关背景知识介绍：简要介绍了有关全变分图像复原的相关背景知识，包括凸函

数问题、全变分模型、图像处理的一阶算法。

第三章自适应谱步长快速前向后向分裂算法：详细介绍了前向后向算法的一般框架，以及对该算法的收敛性分析及停止准则的介绍。针对前向后向算法在处理凸优化问题中采用固定步长，使得算法收敛速率受限的问题，结合 Barzilai-Borwein^[16]谱步长算法，进行算法的改进，将固定步长的前向后向算法改进为自适应步长选择的快速迭代收缩阈值算法，实验验证算法的有效性。

第四章非单调线性搜索快速前向后向分裂算法：针对改进谱步长前向后向分裂算法的收敛性问题，加入非单调线性搜索方法^[17]，提出一种非单调线性搜索快速前向后向分裂算法。自适应谱步长的选择改进了快速前向后向分裂算法的收敛速率，然而该算法并不能完全保证算法的收敛。根据该问题，加入非单调线性搜索，判断条件满足时进行算法迭代，否则减小步长因子，直到满足条件。数值实验结果表明，所提算法应用于全变分图像复原时，图像具有更高的信噪比，同时算法运行时间更短，因而提高了算法的计算效率。

第五章自适应重启动快速前向后向分裂算法：针对快速前向后向分裂算法迭代过程中目标函数出现“震荡”现象。结合梯度式重启动技术^[18]，改进算法为自适应重启动前向后向分裂算法，该算法在不依赖于对参数的估计下，通过梯度式重启动判定因子进行判断，当迭代朝坏的方向发展时，重新启动算法的执行，通过数值实验验证，结果表明基于重启动技术的快速前向后向分裂算法提高了算法的收敛速率，图像复原质量得到了进一步的改善。

第六章总结与展望：首先总结了本文的主要创新及研究成果，然后分析文章研究方法的不足，展望下一步研究计划。

第二章 相关背景知识介绍

2.1 全变分模型

现有的图像退化模型为如下形式：

$$b = Au + n \quad (2.1)$$

其中 $A \in \mathbb{R}^{m \times n}$ 是模糊算子， $b \in \mathbb{R}^m$ 是模糊图像， $u \in \mathbb{R}^n$ 是待恢复的未知的图像信号。图像恢复要解决的问题就是从上面的退化模型中获得原始图像的最佳估计值 u 。 n 是加性噪声，一般假设为高斯噪声。而当矩阵 A 是单位矩阵时，该问题就成为图像去噪问题，当 A 在空间不变情况下表示一个二维卷积算子，此时模型(2.1)为从已观察模糊图像中的图像去模糊问题。从上式中恢复 u 是基本的线性逆问题，该研究涉及到很多应用领域，例如图像/信号恢复、压缩感知等问题。

经典的图像恢复模型为：

$$\begin{aligned} \min & \|Au - b\|^2 \\ \text{s.t.} & \|u\|_1 \leq T \end{aligned} \quad (2.2)$$

其中 T 是非负正实数。

(2.2) 式所示的图像复原模型，用到的是 L_1 正则项，文献^[19]首次提出利用全变分正则化来去除信号中所含噪声。由于全变分模型具有很好的数学特性，该特性使得全变分模型下的图像复原方法能够很好的保留图像的边缘信息，从而获得比较完整的图像相关信息，该模型已经成功的应用到图像处理的诸多问题中。因此，本文图像复原，采用全变分模型。

全变分模型指的是信号绝对梯度的积分值，则一幅二维图像的 $b \in \mathbb{R}^{m \times n}$ 全变分定义式为：

$$\|u\|_{TV} = \left\{ \begin{array}{l} |\nabla_x u| + |\nabla_y u| \\ \sum_i \sqrt{(\nabla_x u)_i^2 + (\nabla_y u)_i^2} \end{array} \right. \quad (2.3)$$

简言之： $\|u\|_{TV} = \sum_{i=1}^{mn} \|D_i u\|_q$ ，其中 $D_i u \in \mathbb{R}^2$ 表示 u 关于像素 i 的离散化梯度。因此，当 $q=2$ 时 $\|u\|_{TV}$ 表示各项同性， $q=1$ 时则为各向异性。

将以上全变分正则项代替(2.2)式中的 L_1 正则项应用到图像恢复中，则得到基于 TV 模型的离散惩罚变体形式为：

$$\min_x \|Au - b\|^2 + 2\lambda \|u\|_{TV} \quad (2.4)$$

其中, A 表示模糊操作算子, 当 A 是单位矩阵时, 则为去噪模型, $b \in R^{m \times n}$ 是观测的含噪图像, $\|\cdot\|$ 表示向量的欧几里得范数, 参数 $\lambda (\lambda > 0)$ 为正则化参数, 均衡数据保真项和噪声敏感度, $\|\cdot\|_{TV}$ 是离散 TV 范数, u 表示理想的未知的待恢复图像。

解决基于 TV 模型的图像恢复问题的难点在于 TV 范数是非光滑的, 且图像处理涉及大规模数据的计算。因此得到快速且简易的算法较困难。由于快速算法通常建立在复杂的算法上且需要大量的计算, 简易算法通常是基于一阶的算法, 但是却收敛速率较低。为了降低计算难度同时提高算法的简易求解速度, 本文的主要目的为在利用一阶前向后向分裂算法^[2]的简易性求解的基础上得到快速算法。

2.2 全变分图像复原模型

全变分模型 (TV) 的优点是能够准确地保留图像的边缘信息, 得到完整的图像信息对于图像复原技术具有较好的恢复效果, 因此, 在图像复原中一般采用全变分(TV)模型代替 L_1 模型。

对于一幅二维图像 u , 第 i 行、第 j 列中的像素值为 u_{ij} 。则图像 u 的离散梯度表示为 ∇u 。图像中各像素点的梯度可表示为 $(\nabla u)_{ij} = (u_{i+1,j} - u_{i,j}, u_{i,j+1} - u_{i,j})^T$ 。

全变分图像复原问题如下

$$\min_u \lambda |\nabla u| + \frac{1}{2} \|Au - b\|^2 \quad (2.5)$$

其中, u 为待复原图像, b 为含噪声的已知观测图像, λ 为正则化参数。 u 的全变分定义式如下

$$|\nabla u| = \sum_{i,j} \|(\nabla u)_{ij}\| = \sum_{i,j} \sqrt{(u_{i+1,j} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2} \quad (2.6)$$

全变分模型具有稀疏性, 是指 u 的梯度值具有稀疏性。因此, 利用 (2.5) 式, 可以计算得到与已知观测图像 b 的近似值。

2.3 一般优化问题

2.3.1 凸优化理论

由于本文所述内容和需要解决的问题均与凸约束优化问题有关, 因此首先介绍有关凸函数的基本概念和定理。 , 如图所示为凸函数示意图:

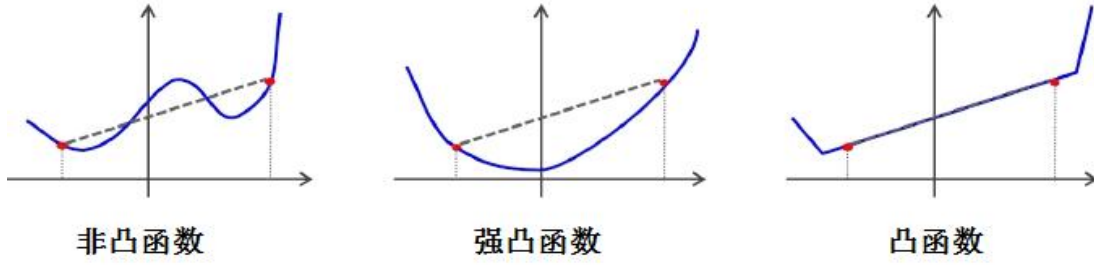


图 2.1 凸函数示意

凸函数在数学上的定义为:

定义 2.1. 若函数 $f: R^n \rightarrow \bar{R} = R \cup \{+\infty\}$, $dom(f) = \{x: f(x) \neq +\infty\}$, 对于任意的 $x, y \in R^n$, $\theta \in [0, 1]$, 满足以下不等式:

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y) \quad (2.7)$$

则函数 f 就是凸函数。用 Q 表示一般凸函数。

若函数 f 满足以下不等式的要求:

$$f(\theta x + (1-\theta)y) < \theta f(x) + (1-\theta)f(y) \quad (2.8)$$

那么函数 f 即为强凸函数。用 \bar{Q} 表示强凸函数。

定义 2.2. 若 ∇F 是 Lipschitz 连续的函数, 则 $L(F) > 0$ 使得:

$$\|\nabla F(x) - \nabla F(y)\| \leq L(F)\|x - y\|, \forall x, y \in R^n \quad (2.9)$$

为了方便起见, 将这种形式定义为 $Q_{L(F)}^{l,1}$ 。

定义 2.3. 令 $\partial F(x)$ 表示凸函数 F 在点 x 的次梯度。那么, 极大单调算子 F 的近似映射^[21]可以表示为:

$$J_{\partial F(z)} := (I + \partial F)^{-1} z = \arg \min F(x) + \frac{1}{2}\|x - z\|^2 \quad (2.10)$$

其中 ∂ 表示次梯度操作, $\partial F(x)$ 是函数 F 在点 x 的次梯度集合。特别地, 当 $F(x) = \mu|x|$, $\mu > 0$ 时有

$$J_{\partial F(z)} := shrink(z, \mu) \quad (2.11)$$

这里的 ' $shrink$ ' 函数为收缩算子, 定义式如下

$$shrink(z, \mu) := \frac{z_i}{|z_i|} \max\{|z_i| - \mu, 0\} = \max\{|z_i| - \mu, 0\} sign\{z_i\} \quad (2.12)$$

F 的共轭凸函数 F^* 定义为:

$$F^*(p) = \sup_{\mu} \langle \mu, p \rangle - F(\mu) \quad (2.13)$$

共轭函数满足: $p \in \partial F(\mu) \Leftrightarrow \mu \in \partial F^*(p)$

2.3.2 一般优化模型

对于非光滑凸优化模型:

$$\min \{F(x) \equiv f(x) + g(x) : x \in E\} \quad (2.14)$$

上述模型满足以下假设:

1. $g: E \rightarrow (-\infty, +\infty]$ 是有界闭合凸函数。
2. $f: E \rightarrow \mathbb{R}$ 是可导且有 Lipschitz 连续梯度 $L(f)$:

$$\|\nabla f(x) - \nabla f(y)\| \leq L(f) \|x - y\|, x, y \in E$$

其中 $\|\cdot\|$ 表示标准欧几里得范数, 表示函数 f 的 Lipschitz 常数。

式 (2.15) 概括了基本的一般优化问题。当 $g(x) \equiv \delta_C(x)$, $C \in E$ 时, δ_C 是有界区域 C 的指示函数, 此时则为标准光滑凸约束优化问题模型:

$$L(x) > 0 \min_x \{F(x) : x \in c\} \quad (2.15)$$

其中, c 代表 \mathbb{R} 中的封闭凸集, 函数 $F(x)$ 表示凸集 c 中的连续凸函数。

有效求解凸优化问题的相关算法有一阶方法^[20]、二阶方法等。其中二阶方法有更好的收敛效果, 且迭代次数较少, 然而, 二阶方法需要计算 $F(x)$ 、 $\nabla F(x)$ 、和 $\nabla^2 F(x)$, 增加了计算的难度, 无法快速得到问题的解, 不适合大规模问题的求解。为了降低求解的难度, 一阶方法成为研究的热点。

2.3.3 基于梯度的算法

为了有效解决凸优化问题, 梯度下降法是最早的一类优化算法。

为了便于分析, 考虑如下无约束的凸优化问题

$$\min \{f(x) : x \in \mathbb{R}^n\} \quad (2.16)$$

$f: \mathbb{R}^n \rightarrow \mathbb{R}$ 是具有 Lipschitz 常数的可微凸函数。即有:

$$\|\nabla f(x) - \nabla f(y)\| \leq L(\nabla f) \|x - y\| (\forall x, y \in \mathbb{R}^n \times \mathbb{R}^n) \quad (2.17)$$

其中 $L(\nabla f) > 0$ 是 ∇f 的 Lipschitz 常量。

解决 (2.16) 式的一般方法是通过简单的固定点迭代, Moreau 等提出的近似映射法^[21, 22] 是关键迭代:

$$\text{prox}_f(x) = \arg \min \left\{ f(u) + \frac{1}{2t} \|u - x\|^2 \right\}$$

其中 $g: E \rightarrow (-\infty, +\infty], t > 0$ 。

采用梯度下降法^[10]最小化函数 f :

算法 1 : 梯度下降算法

1: For $k = 1, 2, 3 \dots$ do

2: $x^{k+1} = x^k - \rho \nabla f(x^k)$

End For

其中 $\rho > 0$ 为步长因子, 当该步长因子满足 $\rho < 2/L(\nabla f)$ 时, 梯度下降算法具有收敛性。 $L(\nabla F)$ 是 ∇F 的 Lipschitz 常量。由于 (2.16) 仅包含凸函数部分, 因此该方法不具有普遍性。因此, 由 (2.16) 式衍生出如下式:

$$\min \{ F(x) = f(x) + g(x) : x \in R^n \} \quad (2.18)$$

其中 $f(x)$ 和 $g(x)$ 均是凸函数, 对算法 1 进一步优化得到了前项后项分裂^[2, 23] 算法 (Forward-Backward Splitting, FBS)

算法 2 : 前项后项分裂算法 (FBS)

1: For $k = 1, 2, 3 \dots$ do

2: $x^{k+1} = J_{\partial g} \left(x^k - \rho \partial f(x^k) \right)$

End For

当 $\rho < 2/L(\nabla f)$ 时, 满足算法收敛性条件, 文献^[23]指出前向后向分裂算法在求解问题 (2.17) 时收敛效率差强人意。在当该问题满足 Lipschitz 连续时, FBS 的全局收敛速度仅为 $O(1/k)$ 。即对于一些常量 C , $F(x^k) - F^* < C/k$ 。之后文献^[24]中, Nesterov 对该算法作出了改进, 在梯度算法的基础之上加入了加速算子, 改进后使得算法收敛效率达到了 $O(1/k^2)$, 具体算法迭代如下:

算法 3 : Nesterov 优化梯度算法

初始化 $t_1 = 1, y_1 = x_0 \in R^n, t < 1/L(\nabla F)$

1: For $k = 1, 2, 3 \dots$ do

```

2:  $x^k = y^k - t\nabla F(y^k)$ 

3:  $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2$ 

4:  $y_{k+1} = x_k + (x_k - x_{k-1})(t_k - 1)/t_{k+1}$ 

End For

```

Nesterov 算法加入一阶分裂算法的思想，它与快速近似梯度算法^[25]极为相似。近年来，随着对压缩感知问题（CS 问题）^[26]的研究的兴起，研究的热点成为了一阶的优化算法问题。Beck 和 Teboulle 提出的快速迭代收缩阈值 (Fast Iterative Shrinkage-thresholding Algorithm, FISTA)^[15] 算法就是一种优化的一阶前向后向分裂算法（FBS）。

算法 4：快速迭代收缩阈值算法（FISTA）

初始化 $t_1 = 1, y_1 = x_0 \in R^n, t < 1/L(\nabla F)$

```

1: For  $k = 1, 2, 3 \dots$  do

2:  $x^k = J_{t\hat{c}g}(y^k - t\nabla f(y^k))$ 

3:  $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2$ 

4:  $y_{k+1} = x_k + (x_k - x_{k-1})(t_k - 1)/t_{k+1}$ 

5: End For

```

FISTA 算法是首个将全局性的加速因子应用于前向后向分裂算法的方法，也可以称之为快速前向后向分裂算法。该算法具有一定的局限性，文献^[27]指出算法对于参数的设置具有很强的敏感性。本文则通过研究各种参数对于算法的影响进行算法的改进，从而提升算法的收敛效率以及对于其凸优化问题的求解。

2.4 图像重建评价标准

图像复原算法中，实验所得结果无法通过肉眼直接判断算法复原性能的优劣。因此需要利用一定的标准来衡量复原后图像的质量的优劣。常见的图像质量评价标准包括：信噪比（Signal-to-Noise, SNR）、峰值信噪比（Peak Signal-to-Noise Ratio, PSNR）、结构化相似（structural similarity, SSIM）。

其中 SNR 方法定义式如下：

$$SNR = \sum_i \frac{u_i(x, y)}{abs(\bar{u}_i(x, y) - u_i(x, y))} \quad (2.19)$$

其中 $u_i(x, y)$ 为原始图像对应像素的灰度值, $\bar{u}_i(x, y)$ 表示复原图像中的对应的像素值。信噪比评价函数结构简单, 运算快速, 对比明显, 因此该参数在图像处理中普遍被应用。

对于大小为 $m \times n$ 的图像, PSNR 定义式如下

$$PSNR = 10 \log_{10} \left\{ \frac{255^2}{\frac{1}{mn} \sum_x \sum_y [u(x, y) - \bar{u}(x, y)]^2} \right\} \quad (2.20)$$

其中, $u(x, y)$ 为原始清晰图像, $\bar{u}(x, y)$ 为复原后的图像。一般地, 该标准数值越大, 图像复原效果越好, 反之则较差。

SSIM (structural similarity) 即结构化相似又叫作基于结构失真的图像质量评价函数, 是一种基于结构信息衡量原始图像和复原图像之间相似程度的方法。该方法相比于普通的 SNR、PSNR 评价函数对于图像失真的检验更加客观。其评价依据是人类视觉系统 (Human Visual System) 是否适合于提取视觉场景中的结构信息, 从测量结构信息的改变与感知图像质量的变化来评价复原质量, 能更好的检测图像在整体结构上是否能更好的贴近原图。其定义式如下:

$$S(x, y) = f(l(x, y), c(x, y), s(x, y))$$

其中 $S(x, y)$ 描述了失真信号与原始信号之间的相似性, 作为失真的度量参数是其三个子函数的整体度量, $l(x, y)$ 是亮度比较函数, $c(x, y)$ 是对比度函数, $s(x, y)$ 是结构比较函数。三个字函数定义式如下:

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, C_1 = (K_1 L)^2$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, C_2 = (K_2 L)^2$$

$$s(x, y) = \frac{2\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

其中 $\sigma_{xy} = 1/(N-1) \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y)$, L 代表了像素值的变化, $K_1, K_2 < 1$, μ_x, μ_y 是亮度值, σ_x, σ_y 是标准方差对比度的估计值。其中 μ_x, μ_y 分别表示各个像素点在坐标轴上的数值。

整合以上子函数, SSIM 的泛函表达式为:

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (2.21)$$

其中 $\alpha, \beta, \gamma > 0$ 是对三个子函数的细微调整。

以上所述SSIM版本是用于视频跟踪的检验方法，运用SSIM检验图像恢复应用则相对简单，不需要第三项结构失真对比。所以本文中采用了SSIM的简化版本，即原结构变化子函数 $s(x, y)$ 没有添加，即 $\gamma=0$ 。化简后SSIM可以表示为：

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_x\sigma_y + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2.22)$$

SSIM 方法是在评判图像的相似度时，其值越大表明两幅图像更加符合人眼对影响品质的判断。其值计算简单、并且其结果与主观质量评价关联性较强。

综上所述，本文衡量图像复原质量的指标有SNR、PSNR和SSIM。

2.5 本章小结

本章首先介绍了图像复原的相关技术，并重点介绍了全变分模型图像复原，解决全变分图像复原问题，由于图像数据量大等原因，考虑选择一阶优化算法进行研究与实现，因此介绍了凸优化的理论及一阶优化算法的发展，其中重点介绍了一阶前向后向分裂算法及其收敛性判定、算法的停止准则。最后介绍了本文评价图像复原算法的相关评价方法。

第三章 全变分模型自适应步长加速前向后向分裂算法

本章根据前向后向分裂算法（FBS）的收敛速率对参数的选择具有很强的敏感性，例如步长因子，于是结合 Barzilai-Borwein 谱步长算子，通过更新迭代步长参数，提出了一种自适应快速前向后向分裂算法（FBB, Forward Backward Splitting and Barzilia Borwein）。该算法利用当前迭代点信息和前一步迭代点信息更新步长因子，大大提升了算法的收敛速率，同时提高了图像复原的质量。通过实验，验证了本章所提改进算法的有效性。

3.1 前向后向分裂算法

全变分最小化是图像处理中的正则化逆问题，通过对该优化问题的计算求解，以期能够复原出具有稀疏梯度的信号或图像。然而，由于函数的不可微性，增加了对该正则化问题的求解难度。因此，研究适用于大规模问题的有效方法是十分必要的。其中，前向后向分裂法（Forward Backward Splitting, FBS）^[3]是求解 TV 该类不可微凸优化问题的基础方法。

FBS 是一种基于算子分裂思想的方法，它通过交替化（次）梯度的逼近值实现用于求解凸泛函总和的最小值问题。FBS 在每次迭代计算时，光滑可导函数由于可以计算到准确的梯度值，因此可以使用梯度值，不可微函数项由于无法得到梯度值，则用近似映射函数替代，同时结合 Nesterov 提出的加速算法理论，迭代过程较为简单，由于该算法简单方便的计算使得他被广泛应用于求解大型数据或者高维度最小值优化问题中。

3.1.1 前向后向分裂算法的一般框架

在早期图像处理方法的本质都是线性，是基于经典的函数分析和线性代数的方法。上世纪 50 年代末 60 年代初，随着非线性分析的发展、快速计算机计算的应用，非线性技术已经慢慢的变得流行，尤其是在凸优化问题中有了更加宽泛的应用。

对于求解全变分模型的凸优化问题的主要难点在于 TV 范数的不可微性，不能利用经典的函数分析等方法解决。为了克服 TV 模型的该缺陷，人们希望通过研究能够得到用于求解图像全变分极小化问题的方法。典型的方法有：Chambolle 投影方法^[28]、分裂 Bregman 迭代算法^[13,29]、Nestrov 算法^[24]、ADM 算法^[14]和前向后向分裂算法（FBS）^[3,30]。其中前向后向分裂算法是一种简单高效的算法。

考虑如下函数：

$$\min h(x) = f(x) + g(x) \quad (3.1)$$

其中 $x \in \mathbb{R}^N$, f 是可导函数, g 是任意的凸函数（无需光滑函数）, 实际解决的问题中, 函数 g 既不是可微的同时函数值又趋于无穷大。该函数模型经常在机器学习、字典学习、云计算、稀疏编码、谱聚类、压缩感知等问题中出现。

许多情况下, 对于既不可微, 也不能取到有限值的函数 g , 不能利用简单梯度下降法求得最小值。可以使用近似算子, 求解函数的近似映射：

$$prox_g(z, \tau) = \arg \min_x \tau g(x) + \frac{1}{2} \|x - z\|^2 \quad (3.2)$$

近似操作算子可以在不偏离起始点 z 的情况下找到函数 g 最小值的近似点。这个算子通常被称之为步长为 τ 的后向梯度（隐式梯度）下降。若近似操作算子较容易计算, 那么 (3.1) 式就可以方便的利用前向后向分裂算法高效的求解。前向后向分裂算法可以解决不可微目标函数和凸约束问题, 同时能保持隐式梯度下降算法的简便性和有效性。

前向后向分裂算法的算法如下：

算法 2: FBS 算法

1: For $k=1,2,\dots$,do

2: $\hat{x}^{k+1} = x^k - \tau^k \nabla f(x^k)$

3: $x^{k+1} = prox_g(\hat{x}^{k+1}, \tau^k) = \arg \min_x \tau^k g(x) + \frac{1}{2} \|x - \hat{x}^{k+1}\|^2$

End For

标量 τ^k 表示第 k 次迭代的步长。步骤 3 为近似映射算子步骤, 也称为后向梯度下降步骤。

(3.2) 式中任一项 x^p 需满足以下优化条件：

$$\tau G + (x^p - z) = 0 \quad (3.3)$$

其中 $G \in \partial g(x^p)$ 是函数 g 的次梯度。当函数 g 可微时, 则 $G = \nabla g(x^p)$ 。上式可进一步表示为：

$$x^p = prox_g(z, \tau) = z - \tau G \quad (3.4)$$

(3.4) 式表明 z 与函数 g 的次梯度相反时就可以得到函数的最优点 x^p 。正因为如此, 近似算子被称作梯度下降算子, 也因次梯度的取值为 x^p 点而非起始点, 因此称之为后向梯度下降算子。

等式 (3.3) 等价于 $0 \in \partial \tau g(x^p) + (x^p - z)$, (3.4) 亦可重新表示如下

$$z \in \tau \partial g(x^p) + x^p = (\tau \partial g + I)x^p$$

基于这个思想，（3.2）式中的近似算子又可表示为

$$x^p = (\tau \nabla g + I)^{-1} z = J_{\tau \nabla g} z$$

其中， $J_{\tau \nabla g} = (\tau \nabla g + I)^{-1}$ 表示 $\tau \nabla g$ 的预解算子^[31]。

特殊情况下的算法^[32]:

(1)GPM (Gradient Projection Method): 当 $g(x) \equiv \delta_C(x)$, 则步骤 3 简化为解决光滑约束最小化问题的梯度映射算法: $x_k = P_C(x_{k-1} - \tau_k \nabla f(x_{k-1}))$

(2)PMA (Proximal Minimization Algorithm): 当 $f(x) \equiv 0$, 此时步骤 3 为近似最小化约束问题, 用来解决非光滑凸优化问题: $x_k = \arg \min \{ \tau_k g(x) + \frac{1}{2} \|x - x_{k-1}\|^2 \}$

(3)ISTA (Iterative Shrinkage/Thresholding Algorithm): 当 $g(x) = \|x\|_1$, 则步骤 3 简化为:

$$x_k = \Gamma_{\lambda \tau_k}(x_{k-1} - \tau_k \nabla f(x_{k-1}))$$

其中 $\Gamma_\alpha: E \rightarrow E$ 是收缩算子。

由于 FBS 的大量应用以及其在图像处理中的利用, 许多 FBS 的变体已经被提出来增强性能, 或更容易使用例如: 快速前向后向分裂算法。同原始的 FBS 形式一样, 需要使用者选择许多的收敛参数, 这些参数较强的影响着算法的稳定性和性能。其中包括步长、停止准则参数、加速方法、稳定情况和初始化等。在正确的修改情况下, FBS 可以在没有使用者监督的情况下执行。

为了应用前向后向分裂算法来解决全变分最小化问题, 则需要将全变分模型 $|\nabla u|$ 重构为一个可微的函数。因此, 针对 (2.5) 式, 引入一个对偶变量 w , 假如 $w_{ij} = (w_{ij}^v, w_{ij}^c)^T$ 是 R^2 的一个向量。应用对偶理论可以得到下式:

$$\max_{\|w_{ij}\| \leq 1} \langle w_{ij}, (\nabla u)_{ij} \rangle = \|(\nabla u)_{ij}\| \quad (3.5)$$

公式 (3.5) 即为 Cauchy-Swartz 不等式, 即 $\langle w_{ij}, (\nabla u)_{ij} \rangle \leq \|w_{ij}\| \|(\nabla u)_{ij}\|$ 。当向量 $(\nabla u)_{ij}$ 与 $(\nabla u)_{ij}$ 平行时, 不等式则取到等号。将 (3.5) 式应用于 (2.6) 式, 可以得到关系式 $|\nabla u| = \max_{\|w\|_\infty \leq 1} \langle w, \nabla u \rangle$ 。应用以上的理论知识, 可重新将全变分模型图像复原问题^[31,33]表示如下

$$\begin{aligned} \min_u \lambda |\nabla u| + \frac{1}{2} \|u - b\|^2 &= \min_u \max_{\|w\|_\infty \leq 1} \lambda \langle w, \nabla u \rangle + \frac{1}{2} \|u - b\|^2 \\ &= \max_{\|w\|_\infty \leq 1} \min_u \lambda \langle w, \nabla u \rangle + \frac{1}{2} \|u - b\|^2 \end{aligned} \quad (3.6)$$

式 (3.6) 中的里面的最小化问题的求解是可微的。假如已知变量 w , 原图像 u 满足条件 $u = b + \lambda \nabla \bullet w$ 时能取到内部的最小值, 其中, $\nabla \bullet w$ 具有离散发散性。该操作算子在点 ij 处的像

素的标量值为 $(\nabla \cdot \mathbf{w})_{ij} = \mathbf{w}_{ij}^g - \mathbf{w}_{ij}^f + \mathbf{w}_{ij}^c - \mathbf{w}_{ij}^c$ 。若将 \mathbf{u} 的最优解代入 (3.6) 式中同时化简该式，此时，对偶变量 \mathbf{w} 的最优解为：

$$\mathbf{w}^* = \arg \max_{\|\mathbf{w}\|_\infty \leq 1} -\frac{1}{2} \left\| \nabla \cdot \mathbf{w} + \frac{1}{\lambda} \mathbf{b} \right\|^2 = \arg \min_{\|\mathbf{w}\|_\infty \leq 1} \frac{1}{2} \left\| \nabla \cdot \mathbf{w} + \frac{1}{\lambda} \mathbf{b} \right\|^2 \quad (3.7)$$

式 (3.7) 简化成为一个包含无穷范数约束项的二次最小化问题。此时，可以应用前向后向分裂算法进行求解 (3.7) 式。使用的算法最小化问题 (3.7) 时交替求解梯度下降步，然后，应用式 $\mathbf{w}_{ij} \leftarrow \mathbf{w}_{ij} / \max\{\|\mathbf{w}_{ij}\|, 1\}$ 将所得到的迭代结果重新投影到无穷范数中。(3.7) 式获取最优解时，再应用公式 $\mathbf{u}^* = \mathbf{b} + \lambda \nabla \cdot \mathbf{w}^*$ 可得复原图像的最优值 \mathbf{u}^* 。

3.1.2 前向后向分裂算法的收敛性分析

本节证明应用 FBS 算法的后向下降步骤能够保证算法的收敛性。

首先，在 FBS 迭代式中考虑固定点 \mathbf{x}^* ，这个点满足：

$$\mathbf{x}^* = \text{prox}_g(\mathbf{x}^* - \tau^k \nabla f(\mathbf{x}^*), \tau^k) = \mathbf{x}^* - \tau^k \nabla f(\mathbf{x}^*) - \tau^k \mathbf{G}(\mathbf{x}^*)$$

其中 $\mathbf{G}(\mathbf{x}^*) \in \text{dg}(\mathbf{x}^*)$ ，将该式简化为： $0 = \nabla f(\mathbf{x}^*) + \mathbf{G}(\mathbf{x}^*)$ 。

满足上式时即为最优情况。该论据简单的证明了存在一个向量是 FBS 迭代的一个固定点当且仅当它是最优解。该等式可以简单的解释为：当算法 FBS 应用到一个最优点 \mathbf{x}^* ，这个点将沿着前向梯度下降步骤移动到新的位置，然后后向梯度步骤将该点带回到它开始的点。前向后向步骤相互作用回到原点，因为梯度值都是依据点 \mathbf{x}^* 计算得到。

简单的固定点特性并不能保证该算法具有收敛性。FBS 当且仅当步长 τ^k 能够满足一定的稳定性条件时才能保证收敛。FBS 的这一稳定性条件不依赖函数 g ，而是取决于函数 f 的曲率。当步长 $\tau^k = \tau$ 时，FBS 收敛条件为下式：

$$\tau < \frac{2}{L(\nabla f)} \quad (3.8)$$

$L(\nabla f)$ 是 ∇f 的 Lipschitz 常数（例如： $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|$ 对于所有的 \mathbf{x}, \mathbf{y} ）。在多数应用中，例如： $f = 1/2 \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ ，此时， $L(\nabla f)$ 是 $\mathbf{A}^T \mathbf{A}$ 的谱半径。

当步长可变时，则步长满足 $0 < \tau^k \leq u < 2/L(\nabla f)$ 能够保证算法的收敛，其中 u 为上边界， l 为下边界。

实际上，人们很少能得到有关 $L(\nabla f)$ 的精确的信息。最好的步长选择依靠问题的解决和每次迭代中的误差。由于这个原因，实际中最好选择自适应的序列 τ^k ，和加强收敛回溯（追踪）

规则而不是确定的步长限制。

3.1.3 混合残差停止准则

随着迭代次数的增加，FBS 的计算精度越来越高，因此需要在有限次的迭代之后使得算法停止迭代。合理的停止判定准则能够在严格的保证该算法能够在经历过有限次迭代后获得所需要的计算精度。

通过研究，本文所提出的算法的停止准则是基于目标函数的导数（当函数不可导时则取函数的次梯度）推导出的相对残差。假设函数 f 光滑，那么可以直接使用函数值进行推导。

当函数 g 不可导时，其次梯度为 $\hat{x}_{k+1} - x_{k+1} / \tau_k \in \partial g(x_{k+1})$ ，相对残差 γ_{k+1} 在点 x_{k+1} 的值为

$$r_{k+1} = \nabla f(x_{k+1}) + \frac{\hat{x}_{k+1} - x_{k+1}}{\tau_k}$$

一个简单的停止准则应该当算法满足 $\|\gamma_{k+1}\| < \text{tol}$ 时停止，其中 $\text{tol} > 0$ 且 tol 是个相当小的值。但该准则存在一个问题，因为它是尺度变化的。

为了获得尺度不变的方法则利用剩余残差取代相对残差，相对残差的定义式如下：

$$\nabla f(x_{k+1}) \approx -\frac{\hat{x}_{k+1} - x_{k+1}}{\tau_k} \quad (3.9)$$

简单的讲，残差是函数 f 的梯度和函数 g 的负次梯度之间的差值。而相对残差 γ_{k+1} 是两个值的差，定义式为：

$$r_{(r)k+1} = \frac{\left\| \nabla f(x_{k+1}) + \frac{\hat{x}_{k+1} - x_{k+1}}{\tau_k} \right\|}{\max \left\{ \left\| \nabla f(x_{k+1}) \right\|, \frac{\left\| \hat{x}_{k+1} - x_{k+1} \right\|}{\tau_k} \right\} + \xi_r} = \frac{\|r_{k+1}\|}{\max \left\{ \left\| \nabla f(x_{k+1}) \right\|, \frac{\left\| \hat{x}_{k+1} - x_{k+1} \right\|}{\tau_k} \right\} + \xi_r} \quad (3.10)$$

其中 ξ_r 表示正常数 ($\xi_r > 0$)

另一个尺度不变停止准则利用了归一化残差，定义式为：

$$r_{(n)k+1} = \frac{\|r_{k+1}\|}{\|r_1\| + \xi_n} \quad (3.11)$$

其中 ξ_n 表示正常数 ($\xi_n > 0$)。相对于精度的绝对测量值，归一化残差测量值给出了近似解相对于 x_1 提高了多少。

以上两个尺度不变的条件各有优缺点。首先相对残差引用面较广，且该准则对初始值 x_0 的选取不敏感。但是相对残差当 $\nabla f(x^*) = 0$ 时则不具有尺度不变性。相对的，归一化残差当

$\nabla f(x^*)=0$ 时仍然有效, 然而, 该收敛测量对初始迭代值的选择具有潜在的敏感性。

一般应用中, 在 $\gamma_{(y)k+1}$ 或 $\gamma_{(n)k+1}$ 变小时调整算法的组合停止条件。在这种情况下, 如果获得高精度利用 $\gamma_{(y)k+1}$ 作为停止准则, 而当存在 0 的情况下使用 $\gamma_{(n)k+1}$ 作为停止准则。因此本文选择组合相对残差和归一化残差的方法作为算法停止准则的判断标准。

3.2 Borzilai-Borwein 自适应谱步长算子

前向后向分裂算法的效率 (一般的梯度法) 对步长 τ_k 的选择非常的敏感。因为这个原因, 大量的工作在研究自适应步长参数。文献^[34]自适应方法自动的调优步长参数可以获得较快速的收敛性。本章讨论谱 (称之为 BB 步长)^[16]步长方法, 以及如何应用到 FBS 算法中。

针对图像复原一般模型:

$$\min h(x) = f(x) + g(x) \quad (3.12)$$

在考虑 (3.12) 中的完整的 FBS 算法之前, 首先考虑 $g=0$ 的情况。此时 $h=f$, 并且 FISTA 简化为简单的梯度下降法的形式:

$$x^{k+1} = x^k - \tau^k \nabla f(x) \quad (3.13)$$

通过 Barzilai-Borwein^[16]算子提出的梯度下降法的谱方法, 则该问题的模型简化为二次方程:

$$f(x) \approx \hat{f}(x) = \frac{a}{2} \|x\|^2 + \langle x, b \rangle \quad (3.14)$$

可以看出最优步长的选择应为 $\tau=1/\alpha$ 。基于此项选择, 梯度下降法在一次迭代下就可以达到简化二次方程 (3.12) 式的最小值。这也激发了以下梯度下降的步长方案: 在应用下降步骤 (3.13) 时, 以 (3.12) 式的二次项近似逼近函数 f 。然后采用步长 $\tau^k=1/\alpha$ 。

谱步长选择已经产生以解决对于特殊选择的 g 的 FBS 方法。当 g 是一个凸函数。则它和自适应步长的结合产生了谱映射梯度法 (SPG)^[35]。SPG 是由 Berg 和 Friedlander^[36,37]中提出的深入解决谱步长的方法。文献^[38]的作者提出一个谱步长规则来解决当 $g=|\cdot|$ 的情况。产生了利用结合了谱梯度下降和子空间的优化方法用来解决 l_1 惩罚项的最小二乘问题。该想法是在文章^[39]中提出的。其中作者对 FBS 算法解决 l_1 惩罚问题提出了修改, 在低秩空间中执行共轭梯度法。

谱步长最初是在文献^[40]中对于应用 SpaRSA 算子解决任意的 f, g 问题时产生的。这个方法是混合了谱方法^[40]和文献^[34]中自适应步长规则。

通过观察, FBS 算法的步长约束依赖函数 f , 而不是函数 g 。FBS 的谱方法利用这个特性。

在每次迭代中构建一个和(3.13)式的 f 方程的二次逼近式, 然后选择最优梯度下降步长: $\tau^k=1/\alpha$ 。

$$\text{令:} \quad \Delta x^k = x^k - x^{k-1} \quad (3.15)$$

$$\Delta F^k = \nabla f(x^k) - \nabla f(x^{k-1}) \quad (3.16)$$

假设 (3.13) 式中 f 的二次模型, 得到下式:

$$\Delta F^k = \nabla f(x^k) - \nabla f(x^{k-1}) = a(x^k - x^{k-1})$$

(3.13) 式适合利用最小二乘法选择 a 最小化 $\|\nabla F^k - a \nabla x^k\|^2$ 或者 $\|a^{-1} \nabla F^k - \nabla x^k\|^2$ 。就像 $g=0$ 的情况。然后我们选择步长 $\tau^k=1/\alpha$ 。结果步长由下式给出:

$$\tau_s^k = \frac{\langle \Delta x^k, \Delta x^k \rangle}{\langle \Delta x^k, \Delta F^k \rangle}, \text{ 且 } \tau_m^k = \frac{\langle \Delta x^k, \Delta F^k \rangle}{\langle \Delta F^k, \Delta F^k \rangle} \quad (3.17)$$

单独的解决二次最小化问题。 τ_s^k 的值就是所谓的“最速下降法”步长。 τ_m^k 是最小冗余步长^[41, 42]。

谱下降法的很多变体将在由 Flecher 所著^[43]中讲到。实际上有一些性能比经典的步长规则 (3.17) 更好。我们指出的 BB 自适应步长^[16]。采用的是如下的规则:

$$\tau^k = \begin{cases} \tau_m^k & \text{if } \frac{\tau_m^k}{\tau_s^k} > \frac{1}{2} \\ \tau_s^k - \frac{1}{2} \tau_m^k & \text{otherwise} \end{cases} \quad (3.18)$$

注意 (尤其是非凸问题), 步长 τ_m^k , τ_s^k 可能是负值, 当变成负值时, 步长需要重塑和替代为前一步的步长。当考虑到复数时, 只用内积的实数部分 (3.18)。

3.3 自适应步长前向后向分裂算法

FBS 算法采用了一阶优化的方法, 参数的设置对于算法性能和稳定性有很强作用, 其中包括步长参数的选择。由于该算法中采用的是固定步长的搜索迭代方法, 这使得算法的搜索速度受到限制。因此, 在原有前向后向分裂算法的基础上加入 BB(Barzilai-Borwein)^[16]算子, 得到自适应谱步长前向后向分裂算法, 以提高算法的收敛速率。

自适应谱步长前向后向分裂算法的具体迭代如下:

算法 5 : Barzilai-Borwein 谱步长的前向后向分裂算法 (FBB 算法)

- 1: 初始化 $x_0, t_0, \rho_0 = 1, u_0, \varphi, k = 1$
- 2: $x_{k+1} = \text{prox}(u_k - \rho_k \nabla f(y_k))$
- 3: 令 $s_{k-1} = x_k - x_{k-1}, y_{k-1} = g_k - g_{k-1}, g_k = \nabla f(x_k)$

$$4: \rho_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}$$

5: 如果 $\rho_k > \frac{1}{L}$, 则用 $\varphi\rho_k$ 替代 ρ_k

$$6: \partial_{k+1} = \left(1 + \sqrt{1 + 4\partial_k^2}\right) / 2$$

$$7: y_{k+1} = x_k + (x_k - x_{k-1})(\partial_k - 1) / \partial_{k+1}$$

8: 更新 $k = k + 1$

9: 如果满足终止标准, 算法停止。

改进的快速前向后向分裂算法, 在原有算法基础上增加了步长选择因子, 而 Barzilai-Borwein 步长因子中梯度的计算刚好可以利用近似算子中梯度的计算, 因而改进算法对原始算法并没有增加计算量的负担。而更新的步长则加快了算法的迭代, 提高了算法的计算效率。

3.4 实验结果和分析

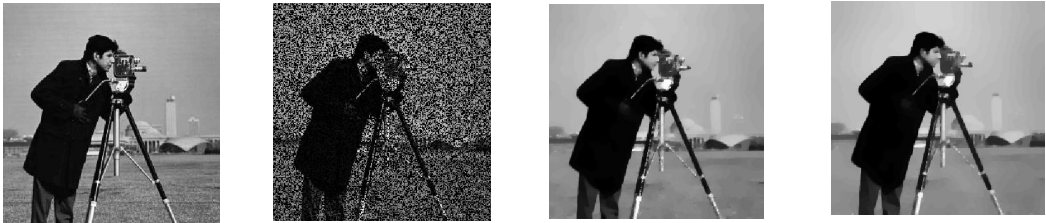
为了验证本文所提算法, 测试图像为灰度图像, 如图(3.1)所示有四幅, 分别为“cameraman” (256×256)、“lena” (512×512)、“barbara” (512×512)、和“peppers” (512×512)。测试平台为 Window7 系统的笔记本, CPU 为 AMD A6-3400M APU with Radeon(tm) HD Graphics 1.40GHz 四核, 4GB 内存, 仿真平台为 MATLAB 7.14(R2012a)。



图 3.1 测试图片

3.4.1 基于全变分模型的图像修复

首先测试图像修复，对原图像添加一个与图像相同尺寸的由 0, 1 组成的随机矩阵，随机的去除图像的像素，导致图像出现缺损，然后再分别采用 FBS 和 FBB 两种算法进行图像修复实验并对比。



(a). 原始图像 (b). 降质图像 (c). FBB 算法修复图像 (d). FBS 算法修复图像
图 3.2. Cameraman 图像修复

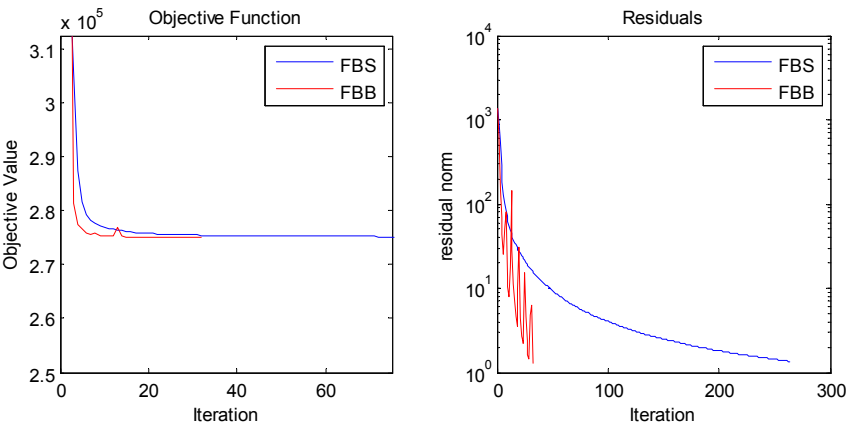


图 3.3. Cameraman 图像修复目标函数值对比图及其残差图

表 3.1 两种算法性能对比

算法	SNR	Cpu time	SNR	PSNR	SSIM	iteratives
FBS	13. 7327	7. 3038s	15. 0392	58. 6287	0. 926	258
FBB		1. 2549s	15. 0466	58. 6290	0. 932	33



(a). 原始图像 (b). 降质图像 (c). FBB 算法修复图像 (d). FBS 算法修复图像
图 3.2. Lena 图像修复

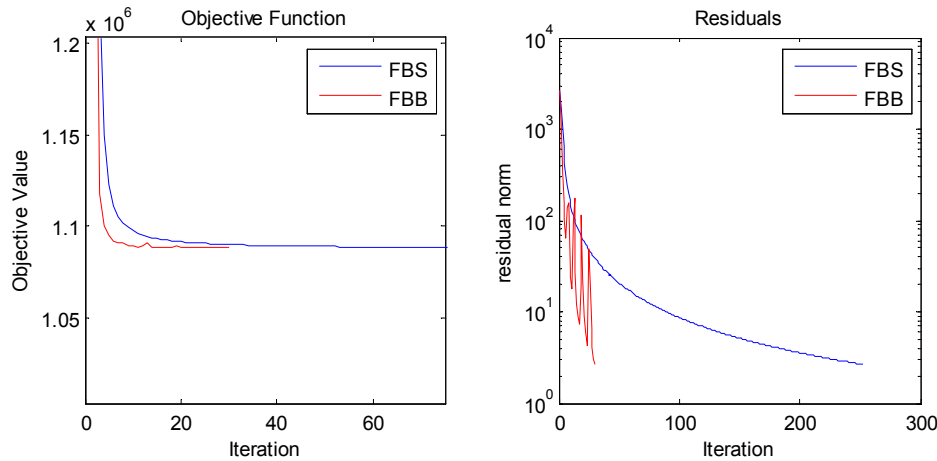


图 3.5. Lena 图像修复目标函数值对比图及其残差图

表 3.2 两种算法性能对比

算法	SNR	Cpu time	SNR	PSNR	SSIM	iteratives
FBS	13. 7327	27. 152s	15. 0392	58. 6287	0. 9877	276
FBB		3. 356s	15. 0466	58. 6290	0. 9897	35

如图所示：采用 FBB 算法的修复图像和采用 FBS 算法修复的图像进行对比，相对误差表示当前迭代的目标函数值与前一步迭代的目标函数值之差与当前目标函数值的比值，该比值越小，表示修复图像效果越好，反之，则较差。两种算法的相对误差对比图则显示了 FBB 算法具有更小的精度值。从修复图像的目标函数值随迭代次数的对比图及其放大图可知，FBB 算法修复下的图像的目标函数值最终小于 FBS 算法的目标函数值，说明图像修复图像更接近原图。表格是对本次实验的图像，分别采用两种算法进行去修复的运行时间、迭代次数、SSIM 和修复图像信噪比的对比。从结果看，FBB 算法在运行时间上比 FBS 算法缩短近一半的时间，大大提高了算法的效率；同时对于信噪比。综上：FBB 算法在图像修复上不仅提高了算法修复效率也提高了算法修复的质量。

3.4.2 基于全变分模型的图像去噪

测试图像去噪，对图像添加方差为 0.01 的 Gaussian 噪声后，然后采用 FBS 和 FBB 算法对图像进行图像去噪实验并作对比分析。如图 5 所示



(a). 原始图像 (b). 降质图像 (c). FBB 算法修复图像 (d). FBS 算法修复图像

图 3.6 Cameraman 图像去噪

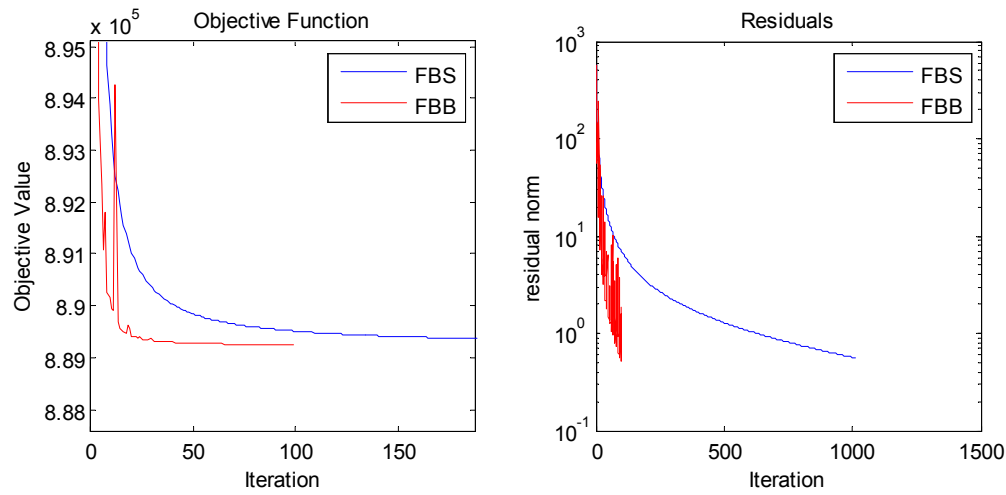


图 3.7. Cameraman 图像去噪目标函数值对比图及其残差

表 3.3 两种算法性能对比

算法	SNR	Cpu time	SNR	PSNR	SSIM	iteratives
FBS	13. 7327	27. 152s	15. 0392	75. 4060	0. 926	996
FBB		3. 356s	15. 0466	75. 4134	0. 932	85



(a) 原始图像 (b). 降质图像 (c). FBB 算法修复图像 (d). FBS 算法修复图像

图 3.8. Lena 图像去噪

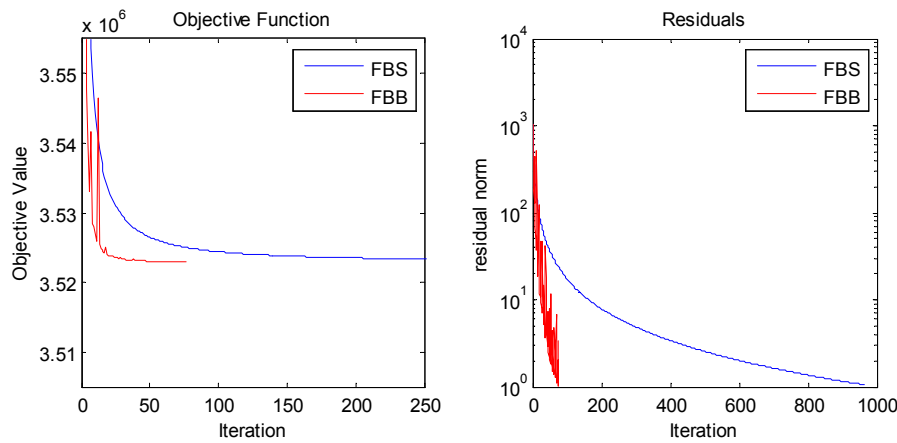


图 3.9. Lena 图像去噪目标函数值对比图及其残差

表 3.4 两种算法性能对比

算法	SNR	Cpu time	SNR	PSNR	SSIM	Iteratives
FBS	5. 5251	120. 386s	15. 2456	78. 3836	0. 912	963
FBB		12. 395s	15. 3654	78. 3889	0. 924	76



(a) 原始图像 (b). 降质图像 (c). FBB 算法修复图像 (d). FBS 算法修复图像

图 3.10. Barbara 图像去噪

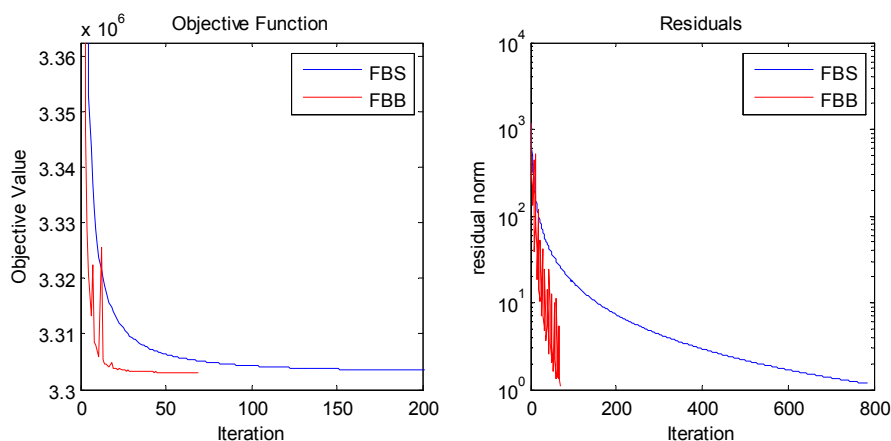


图 3.11. Barbara 图像去噪目标函数值对比图及其残差

表 3.5 两种算法性能对比

算法	SNR	Cpu time	SNR	PSNR	SSIM	Iteratives
FBS	6.7464	100.071s	11.0776	72.7836	0.921	786
FBB		11.127s	11.1172	72.7840	0.927	69



(a) 原始图像 (b). 降质图像 (c). FBB 算法修复图像 (d). FBS 算法修复图像

图 3.12. Peppers 图像去噪

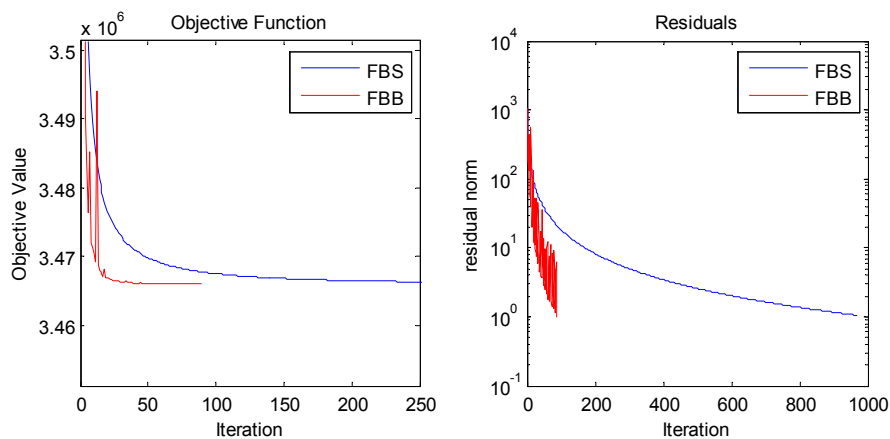


图 3.13. Peppers 图像去噪目标函数值对比图及其残差

表 3.6 两种算法性能对比

算法	SNR	Cpu time	SNR	PSNR	SSIM	Iteratives
FBS	6.6442	121.088s	17.1712	78.8043	0.935	966
FBB		14.306s	17.1755	78.8086	0.939	89

采用 FBB 算法的图像去噪与采用 FBS 算法的去噪图像进行比较。相对误差对比图反映了 FBB 算法最终的相对误差值要小于 FBS 算法下的相对误差值，表示图像去噪效果更好。两算法目标函数随迭代次数的对比图及其放大图中，其中 FBB 算法下的目标函数值小于 FBS 算法下的目标函数值，说明了 FBB 算法去噪图像效果更优。表格是对本次实验图像都添加高斯噪声后利用两种算法对图像去噪的运行时间、迭代次数、SSIM 和信噪比等信息的对比结果。从结果看 FBB 算法提高了图像去噪的效率，缩短了一半的运行时间，同时提高了图像的质量。综上：FBB 算法不仅提高了图像去噪效率，同时提高了图像去噪的质量。

3.5 本章小结

本章在原始前向后向分裂算法的基础上，提出了结合 Bazilai-Borwein 变步长算法的自适应谱步长的快速前向后向分裂算法，该算法利用前两步迭代点的信息来更新当前步长因子，加快了算法的收敛，数值实验表明，利用 FBB 算法所得到的图像，其复原质量更高，迭代时间得到大大提升，因此在一定程度上提高了 FBS 算法的计算效率。

第四章 全变分模型非单调线性搜索前向后向分裂算法

本文第三章主要研究了自适应谱步长选择的快速前向后向分裂算法，该算法利用 Barzilai-Borwein 算子进行步长的更新，加快了 FBS 算法的收敛速率。然而，理论上，谱梯度法通常不能保证算法的收敛性，即使是凸优化问题。因此有必要研究一种保证算法具有稳定收敛特性的算法，文章^[44]曾最早提出一种对于求解光滑问题的线性收敛方法，之后文献^[45,35]将该现象收敛特性应用到梯度映射算法中。文献^[15]提出一种回溯法，该方法是由 Beck 和 Teboulle 提出的。许多专家认为该线性搜索方法过于保守，尤其不适合条件不稳定的问题。于是，经过研究提出了一种非单调回溯线性搜索条件^[46,47]。非单调线性搜索算法与使目标函数单调下降的迭代算法相比，不过分要求目标函数的单调下降性，允许有限次的目标函数的增加。

4.1 非单调线性搜索策略

快速前向后向分裂算法（FISTA）的收敛性是有条件的收敛，即（3.5）式。而实际应用中，人们很难对 ∇f 的特性拥有充足的信息，因此对补充的具体约束值是未知的。经过专家们的研究得出一种非单调线性搜索算法，该算法的目的是在搜索方向上找到目标函数值最小的点，然而，精确的查找到最小值点比较耗时，况且本身搜索方向本来就是近似求解，所以利用较小的计算代价找到最小值点是回非单调线性搜索（Nonmonotone line search）^[46,47]的优势。该算法在机器学习等问题中常用来做为数值优化的算法。

在求解凸优化问题中，步长因子能够保证目标函数的下降性和迭代序列的全局收敛性。早期的算法研究中，线性的搜索的策略是经常被用来确定合适的步长因子来使用的，例如：精确线性搜索和非精确线性搜索。具体如下：

$$(1) \text{ 精确线性搜索: } \alpha_n = \arg \min_{\alpha} f(\hat{x}_{n-1} + \alpha d_n) \quad (4.1)$$

精确线性搜索中只要步长因子能够满足（4.1）式，则目标函数值就有最大可能的下降。但是由于该方法计算量较大，实际应用中效率并不高，因此一般采用的是非精确的线性搜索。

(2) Wolf 非精确线性搜索^[48]:

$$\begin{cases} f(\hat{x}_{n-1} + \alpha_n d_n) \leq f(\hat{x}_{n-1}) + \delta \alpha^n (g^n)^T d^n \\ (d_n)^T g(\hat{x}_{n-1} + \alpha_n d_n) \geq \sigma (g_n)^T d_n \end{cases} \quad (4.2)$$

其中给定 $0 < \delta < \sigma < 1, \delta < 0.5$ ，第一个不等式的要求是使得目标函数下降至少要与切线的下降成比

例，第二个不等式则为了防止步长过小，因而保证了目标函数的足够下降。

以上所述精确线性搜索和非精确线性搜索都要求目标函数单调递减，但当目标函数在可行域中出现了细长而弯曲的峡谷情形时，这种单调性要求则会使得算法的效率大大降低。

针对以上单调线性搜索策略的不足，Grippo 等^[46]针对牛顿法提出了一种非单调线性搜索的策略。非单调线性搜索方法主要拥有两个优势：第一，当目标函数呈现长而窄的沟的差的条件问题时，可能是由于迭代点 x^{k+1} 与点 x^k 距离太近，尽管点 x^{k+1} 有很大的目标函数值，线性搜索将会终止迭代。第二，线性搜索中目标函数需要在每次迭代中求解计算，对于复杂问题，计算目标函数值耗费大量的时间，占据了算法迭代的主要时间。线性搜索条件结束算法很快，因此缓解了计算上的负担。

具体方法是：给定 $M > 0$ 为整数线性搜索参数，定义：

$$\hat{f}^k = \max \{f^{k-1}, f^{k-2}, \dots, f^{k-\min\{M, k\}}\}. \quad (4.3)$$

经过 FBS 迭代后，满足以下线性搜索条件：

$$f(x^{k+1}) < \hat{f}^k + \text{Re} \langle x^{k+1} - x^k, \nabla f(x^k) \rangle + \frac{1}{2\tau^k} \|x^{k+1} - x^k\|^2 \quad (4.4)$$

如果上式的非单调线性搜索条件不满足，则步长减少，直到 (4.4) 式满足为止。

从 (4.4) 可以看出，非单调线性搜索策略的主要思想是：迭代过程中不强制要求每次迭代都满足单调下降，而是综合考虑之前的迭代信息，允许出现一定的波动。实际应用中，非单调线性搜索策略具有比单调线性搜索更好的效果。

4.2 非单调线性搜索加速前向后向分裂算法

根据非单调线性搜索的理论介绍，可以得到具体非单调线性搜索快速前向后向分裂算法（本文称之为 NLFBB）的迭代步骤如下：

算法 9：非单调线性搜索前向后向分裂算法（NLFBB）

While x^k and x^{k+1} violate condition(6.2) do

1: $\tau^k \leftarrow \tau^k / 2$;

2: $x^{k+1} \leftarrow \text{prox}(x^k - \tau \nabla f(x^k), \tau)$

End while

当 (4.4) 式能够保证当 τ^k 小于 ∇f 的 Lipschitz 常数倒数时终止，从而保证算法的收敛性。

4.3 非单调线性搜索前向后向算法的收敛性验证

定义：假设 FBS 应用到凸约束优化问题中，函数 g 为凸函数， f 为可导函数。并且 $h=f+g$ 是合理、半连续有界函数。如果 $\{\tau^k\}$ 为有界正常数并且满足下式：

$$f(x^{k+1}) - \hat{f}^k < \langle x^{k+1} - x^k, \nabla f(x^k) \rangle + \frac{1}{2\tau^k} \|x^{k+1} - x^k\|^2 \quad (4.5)$$

则 $h^* = \lim_{k \rightarrow \infty} h(x^k)$ ，其中 h^* 是函数 h 的最小值。

首先根据近似映射算子： $x^{k+1} = \text{prox}_g(\hat{x}^{k+1}, \tau^k) = \arg \min_x \tau^k g(x) + \frac{1}{2} \|x - \hat{x}^{k+1}\|^2$

当近似操作取得最优值时则该式满足条件： $0 \in \tau^k \partial g(x^{k+1}) + x^{k+1} - \hat{x}^{k+1}$

因此得到下式：

$$0 = \tau^k G^{k+1} + x^{k+1} - \hat{x}^{k+1} = \tau^k G^{k+1} + x^{k+1} - (x^k - \tau F^k)$$

其中 $G^{k+1} \in \partial g(x^{k+1})$ ， $F^k = \nabla f(x^k)$ ，于是：

$$x^k - x^{k+1} = \tau^k (G^{k+1} + F^k) \quad (4.6)$$

$$\text{则有：} \langle x^{k+1} - x^k, F^k + G^{k+1} \rangle = -\frac{1}{\tau^k} \|x^{k+1} - x^k\|^2 \quad (4.7)$$

$$\text{现在，假定函数 } g \text{ 是凸函数，即：} g(x^k) \geq g(x^{k+1}) + \langle x^k - x^{k+1}, G^{k+1} \rangle. \quad (4.8)$$

将（4.5）式减去（4.8）式然后应用到（4.7）式中得到：

$$\begin{aligned} h(x^{k+1}) &= f(x^{k+1}) + g(x^{k+1}) \\ &\leq \hat{f}^k + g(x^k) + \langle x^{k+1} - x^k, F^k + G^{k+1} \rangle + \frac{1}{2\tau^k} \|x^{k+1} - x^k\|^2 \\ &= \hat{f}^k + g(x^k) - \frac{1}{2\tau^k} \|x^{k+1} - x^k\|^2 \dots \\ &= \hat{h}^k - \frac{1}{2\tau^k} \|x^{k+1} - x^k\|^2. \end{aligned} \quad (4.9)$$

其中 $\hat{h}^k = \max\{h^{k-1}, h^{k-2}, \dots, h^{k-\min\{M, k\}}\}$ 。且 $\{\hat{h}^k\}$ 是单调递减有界序列，有界值为 \hat{h}^* 。为了表示 \hat{h}^*

是函数 h 的最小值，观察 $\hat{h}^k = h^{k'}$ ，其中 k' 满足下式：

$$k - M \leq k' \leq k$$

从（4.9）式得知必然存在子序列 $\{x^{k(i)}\}$ 满足 $h(x^{k(i)}) = \hat{h}^k$ ，因此有下式成立：

$$\lim_{i \rightarrow \infty} \frac{1}{2\tau^k} \|x^{k(i)+1} - x^{k(i)}\|^2 = 0 \quad (4.10)$$

其中 h 是有界函数，因此 (4.9) 式也是有界等式。

不失一般性， $\{x^{k(i)}\}$ 是一个收敛子序列，迭代极值点为 x^* 。

根据 (4.6) 式，得到下式：

$$\frac{1}{2\tau^k} \|x^{k(i+1)} - x^{k(i)}\|^2 = \frac{\tau^k}{2} \|G^{k(i)+1} + F^{k(i)}\|^2 \quad (4.11)$$

(4.11) 式结合 (4.10) 式，且 τ^k 是一个有界值为 0 的序列，由此得到：

$$\lim_{i \rightarrow \infty} \|G^{k(i)+1} + F^{k(i)}\|^2 = 0$$

因为 ∇f 是 Lipschitz 连续的，且 $\|x^{k(i+1)} - x^{k(i)}\| \rightarrow 0$ ，即可得到： $x^{k(i)+1} \rightarrow x^*$ ，

$$\lim_{i \rightarrow \infty} \|G^{k(i)+1} + F^{k(i)+1}\|^2 = 0 \quad (4.12)$$

注意 $G^{k(i)+1} + F^{k(i)+1} \in \text{dg}(x^{k(i)+1})$ ，因为凸函数的次梯度是连续的，(4.12) 应用 $0 \in \text{dh}(x^*)$ ，因此函数 h 的最小值为 x^* 。

已知： $\lim_{k \rightarrow \infty} \hat{h}^k = h(x^*) = h^*$ ，由于 $h^* \leq h(x^k) \leq \hat{h}^k$ ，得到如下结论：

$$\lim_{k \rightarrow \infty} h(x^k) = h^*$$

4.4 数值实验

为了验证本文所提算法，测试图像为灰度图像，如图(3.1)所示有四幅，分别为“cameraman” (256×256)、“lena” (512×512)、“barbara” (512×512)和“peppers” (512×512)。测试平台为 Window7 系统的笔记本，CPU 为 AMD A6-3400M APU with Radeon(tm) HD Graphics 1.40GHz 四核，4GB 内存，仿真平台为 MATLAB 7.14(R2012a)。

4.4.1 基于全变分模型的图像修复

首先测试图像修复，对原图像添加一个与图像相同尺寸的由 0, 1 组成的随机矩阵，随机的去除图像的像素，导致图像出现缺损。然后再分别采用 NLFBB 和 FBB 两种算法进行图像修复实验并对比。

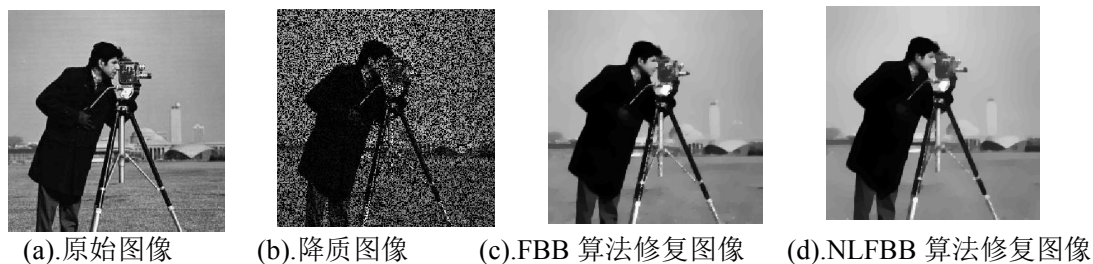


图 4.1 Cameraman 图像修复

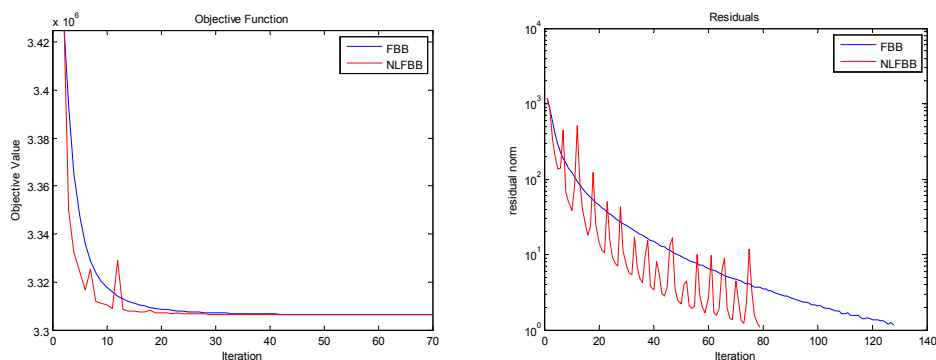


图 4.2 Cameraman 图像修复目标函数值对比图及其残差

表 4.1 Cameraman 图像修复：两种算法性能比较

算法	SNR	Cpu time	SNR	PSNR	SSIM
FBB	7.7840	88.6482s	11.2934	72.8087	0.927
NLFBB		18.9537s	11.2948	72.8101	0.935

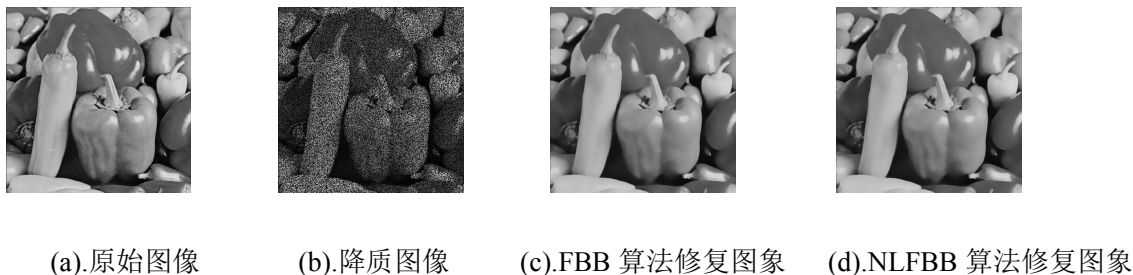


图 4.3 Peppers 图像修复

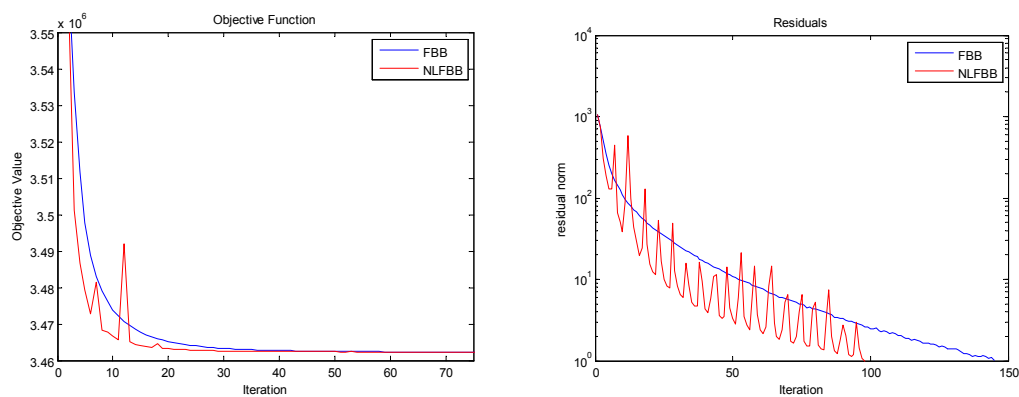


图 4.4 Peppers 图像修复目标函数值对比图及其残差

表 4.2 Peppers 图像修复：两种算法性能比较

算法	SNR	Cpu time	SNR	PSNR	SSIM
FBB	6.7240	109.1880s	16.1903	77.8179	0.932
NLFBB		28.3221s	16.1974	77.8249	0.938

如图所示：采用 NLFBB 算法的修复图像和采用 FBB 算法修复的图像进行对比，相对误差表示当前迭代的目标函数值与前一步迭代的目标函数值之差与当前目标函数值的比值，该比值越小，表示修复图像效果越好，反之，则较差。两种算法的相对误差对比图则显示了 NLFBB 算法具有更小的精度值。从修复图像的目标函数值随迭代次数的对比图及其放大图可知，NLFBB 算法修复下的图像的目标函数值最终小于 FBB 算法的目标函数值，说明图像修复图像更接近原图。表格是对本次实验的四幅图像，分别采用两种算法进行去修复的运行时间、迭代次数、SSIM 和修复图像信噪比的对比。从结果看，NLFBB 算法在运行时间上比 FBB 算法缩短近运行时间，大大提高了算法的效率；同时对于信噪比。综上：NLFBB 算法在图像修复上不仅提高了算法修复效率也提高了算法修复的质量。

4.4.2 基于全变分模型的图像去噪

对图像添加方差为 0.01 的 Gaussian 噪声后，然后分别采用 FBB 和 NLFBB 图像进行图像去噪实验并作对比分析。如图所示。



(a). 原始图像

(b). 降质图像

(c). NLFBB 算法去噪图像

(d). FBB 算法去噪图像

图 4.5 Cameraman 图像去噪

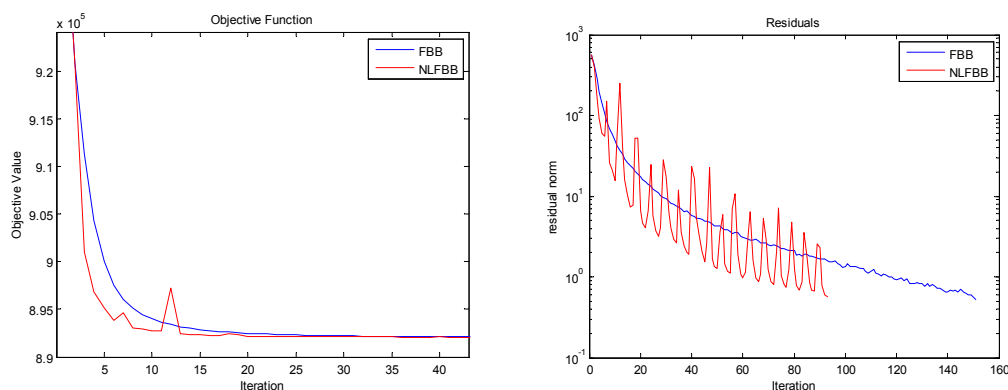


图 4.6 Cameraman 图像去噪目标函数值对比图及其残差

表 4.3 Cameraman 图像去噪：两种算法性能比较

算法	SNR	Cpu time	SNR	PSNR	SSIM
FBB	9.5840	29.9141s	14.3859	74.7320	0.926
NLFBB		5.2295s	14.3878	75.7339	0.931



(a). 原始图像 (b).降质图像 (c).NLFBB 算法去噪图像 (d).FBB 算法去噪图像

图 4.7 Lena 图像去噪

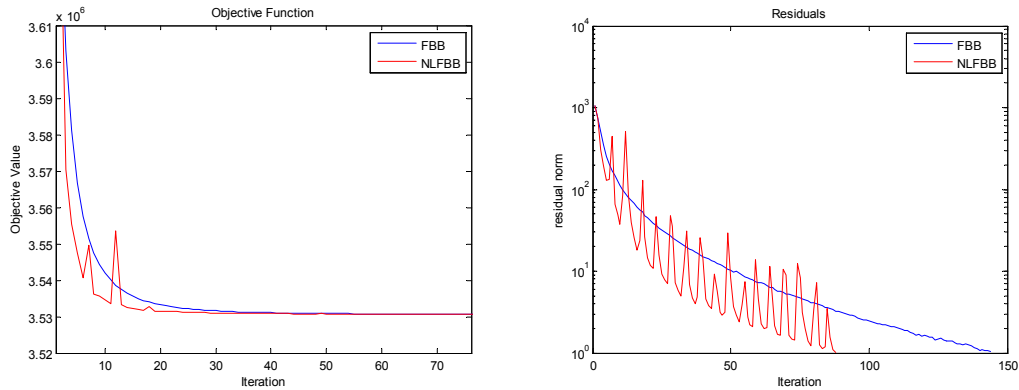


图 4.8 Lena 图像去噪目标函数值对比图及其残差

表 4.4 Lena 图像去噪：两种算法性能比较

算法	SNR	Cpu time	SNR	PSNR	SSIM
FBB	8.6140	106.7026s	15.2105	77.8728	0.935
NLFBB		21.5063s	15.2160	77.8783	0.937

采用 NLFBB 算法的图像去噪与采用 FBB 算法的去噪图像进行比较。相对误差对比图反映了 NLFBB 算法最终的相对误差值要小于 FBB 算法下的相对误差值，表示图像去噪效果更好。两算法目标函数随迭代次数的对比图中，其中 NLFBB 算法下的目标函数值小于 FBB 算法下的目标函数值，说明了 NLFBB 算法去噪图像效果更优。表格是对本次实验图像都添加高斯噪声后利用两种算法对图像去噪的运行时间、迭代次数、SSIM 和信噪比等信息的对比结果。从结果看 FBB 算法提高了图像去噪的效率，缩短了运行时间，同时提高了图像的质量。综上：NLFBB 算法不仅提高了图像去噪效率，同时提高了图像去噪的质量。

4.5 本章小结

本章根据快速前向后向分裂算法的介绍, 引入了非单调线性搜索算法, 在不保证目标函数单调下降的情况下, 保证算法的步长 τ^k 小于函数 ∇f 的 Lipschitz 常数的倒数。从而能够保证算法的收敛性。然后利用非单调线性搜索定理证明该算法的正确性。最后利用数值实验验证该算法的有效性。

第五章 全变分模型自适应重启加速前向后向分裂算法

本章根据快速迭代收缩阈值算法 (FISTA) 迭代过程中目标函数出现“震荡”的现象, 即目标函数值出现的反复情况。结合梯度式重启判定因子, 重新启动算法迭代, 使得在目标函数在线性收敛情况下获得函数局部最优, 从而加快算法的收敛速率。

5.1 一阶加速方法

近几年, 随着对大规模问题求解的需求, 一阶算法成为研究的热点。而该热点研究方向又主要集中在让一阶算法保持原有的优点的基础上利用一定的计算技术提高已有算法的效率。Yuii Nesterov 首先提出了加速优化梯度方法^[24], 他证明了在凸光滑优化问题的应用中, 简单修改梯度下降法就可以减少一阶算法的复杂性从而获得优化的性能。此类方法被称之为“加速”方法。

不同于梯度算法, 加速算法一般不能保证算法中目标函数的单调性。因此执行一般的加速算法, 目标函数的轨迹会出现“震荡”或波动现象, 这使得目标函数值出现反复。如图 (5.1) 所示:

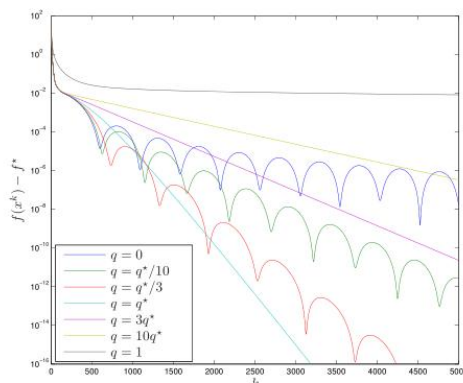


图 5.1 不同加速算法的收敛图

5.1.1 加速近邻梯度算法

考虑光滑无约束优化:

$$\minimize \quad f(x), x \in R^n \quad (5.1)$$

其中 $f: R^n \rightarrow R$ 且是 Lipschitz 连续函数, 其 Lipschitz 梯度为:

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \forall x, y \in R^n \quad (5.2)$$

为了获得光滑凸函数的最小值。利用 f^* 表示 (5.1) 式凸优化问题的最优值，若存在最小值，其值为： x^* 。并且，连续可导函数 f 是强凸函数，强凸参数 $\mu > 0$ ，强凸函数如下式：

$$f(x) \geq f(y) + \nabla f(y)^T (x - y) + (\mu/2)\|x - y\|_2^2, \forall x, y \in R^n \quad (5.3)$$

解决问题 (5.1) 式的加速一阶算法最初是由 Nesterov 于 1983^[49] 年提出的。该方法是将梯度的负方向作为下降方向，在原来的梯度算法的基础上，对迭代点作向外演推计算得到新的迭代点。而加速近邻梯度算法就是通过对梯度法进行简单的修正，改进后的算法保持了原梯度法原有的计算简洁性、不依赖于计算二阶导数等的特点，同时也提高了算法的效率，加速近邻梯度具体迭代如下：

算法 5：加速近邻梯度算法

初始化： $x^0 \in R^n, y^0 = x^0, \theta_0 = 1$ and $q \in [0, 1]$

1: for $k = 0, 1, \dots$, do

2: $x^{k+1} = y^k - t_k \nabla f(y^k)$

3: θ_{k+1} solve $\theta_{k+1}^2 = (1 - \theta_{k+1})\theta_k^2 + q\theta_{k+1}$

4: $\beta_{k+1} = \theta_k(1 - \theta_k) / (\theta_k^2 + \theta_{k+1})$

5: $y^{k+1} = x^{k+1} + \beta_{k+1}(x^{k+1} - x^k)$

6: end for

该算法有很多变体，例如：当 $q=1$ 时则为简单的梯度下降算法。此外，步长的选择也是一个关键环节。

对于光滑凸函数， $t_k \leq 1/L$ 时算法收敛，当步长 $t_k = 1/L$ 且 $q=0$ 时，当存在最小值时，其算法收敛率为：

$$f(x^k) - f^* \leq \frac{4L\|x^0 - x^*\|^2}{(k+2)^2} \quad (5.4)$$

若函数是强凸函数，且强凸参数为 μ ，令 $q = \mu/L$ （条件数的倒数）函数将线性收敛，其收敛率为：

$$f(x^k) - f^* \leq L(1 - \sqrt{\frac{\mu}{L}})^k \|x^0 - x^*\|^2 \quad (5.5)$$

$$\text{此时收敛率为: } o(\sqrt{\frac{L}{\mu}} \log \frac{1}{\varepsilon}) \quad (5.6)$$

强凸函数情况下，如下加速梯度算法能够获得相同的收敛率：

算法 6：加速梯度算法

初始化： $x^0 \in R^n, y^0 = x^0$

1: for $k = 0, 1, \dots, do$

2: $x^{k+1} = y^k - (1/L)\nabla f(y^k)$

3: $y^{k+1} = x^{k+1} + \beta^*(x^{k+1} - x^k)$

4: end for

其中， $\beta^* = \frac{1 - \sqrt{\mu/L}}{1 + \sqrt{\mu/L}}$

算法 5 中，当选择最优值 $q=\mu/L$ 时，则有 $\beta_k=\beta^*$ ，若 β_k 是动力参数，那么 β^* 为最大的动量值；当存在一个比 β^* 更高的 β 值时，则达到最高动量机制。

算法 5 和 6 是由 Nemirovsky 和 Yudin^[50]得出的，其收敛速度是在较低的复杂性边界意义上最佳得出的。然而，这种收敛只有当函数参数 μ 和 L 提前已知时，才能得以保证。

5.1.2 快速前向后向分裂算法

快速前向后向分裂算法（FISTA）^[15]是基于前向后向分裂算法的加速算法，其性能和稳定性同前向后向分裂算法一样，受到诸多收敛参数的影响。该算法是首先做出将全局加速因子加入到原始前向后向分裂算法从而得到加速了的前向后向分裂算法，它分别计算函数的前向梯度映射和后向近似映射来得到问题的近似解。本质上也是一种加速算法，而加速梯度算法本质上被认为是一种动力算法，其迭代中的每一步都依赖前一步的迭代信息，因而随着迭代进行，动力也随之不断增长，当动力超过一个严格的值时，目标函数值的轨迹将出现“震荡”现象，并且“震荡”周期正比于目标函数的局部条件参数的平方根。因此周期性的震荡使得加速算法不能保证目标函数的单调性。

快速迭代收缩阈值算法的具体迭代如下：

算法 7：快速迭代收缩阈值算法（FISTA）

初始化: $t_1 = 1, y_1 = x_0 \in R^n, t < 1/L(\nabla F)$

1: For $k = 1, 2, 3 \dots$ do

2: $x^k = J_{t\partial g}(y^k - t\nabla f(y^k))$

3: $\partial_{k+1} = (1 + \sqrt{1 + 4\partial_k^2})/2$

4: $y_{k+1} = x_k + (x_k - x_{k-1})(\partial_k - 1)/\partial_{k+1}$

End For

快速前向后向分裂算法也算是一种基于梯度的算法法，迭代轨迹亦会出现“震荡”现象，也就是算法迭代过程中目标函数值的大小出现反复的情况。虽然，可以通过设置不同的参数来获得较好的算法效率，但是这需要取决于对算法参数的优化。因此参数 q 若设置的好则可以在一定程度上消除计算迭代过程中的“震荡”现象，从而获得较好的函数收敛效果。

分析算法 7，步骤 2 简单执行了 FBS 算子，第 4 步执行预处理，其中当前迭代点是在它先前一步迭代点的移动方向上进一步向前移动得到，我们称之为算法的“侵略性”。

该“侵略性”是由预处理步骤 4 中的标量 ∂^k 控制。步骤 3 更新该加速参数，并且该加速参数在每次迭代中增加导致该算法更加具有“侵略性”。FISTA 交替的进行预处理函数值和利用 FBS 纠正函数值的运算步骤，该算法最明显的缺点是使得目标函数在每次迭代增加预处理参数 ∂^k 达到了最坏的性能。文章^[51]提出一种迭代收缩算法，该算法类比于 FISTA 算法。

5.2 自适应重启加速算子

为了克服原始快速前向后向分裂算法的缺点，使算法的效率不依赖于参数 q ，Brendan 和 Emmanuel 共同提出了著名的重启技术^[18]。重启技术就是当某种判定条件满足时，以本次迭代产生的最后一个点作为新的迭代点重新启动算法。重启技术^[18]分为两种固定重启和自适应重启，如下解释：

(1) 固定重启：以固定的迭代次数为重启的标准，允许 ∂^k 在算法 3 性能较好时不断增加，当迭代 k 次时，若目标函数值出现震荡，则重启算子；

(2) 自适应重启^[52]：当迭代点向不好的方向发展时重新启动算法，即当目标函数值增大时设置 $\partial^k=1$ ，然而，这种方法需要在每次迭代中计算目标函数值，该方法不仅计算代价大，同时该计算也非必要。文章^[15]提出不需要计算目标函数值，即在当每两次迭代点中目标函数的方向发生变化则重启算法。即

$$\langle y^k - x^k, x^k - x^{k-1} \rangle \geq 0 \quad (5.7)$$

其中 $y^k - x^k$ 平行于目标函数的梯度方向。基于此，当相邻迭代点与梯度方向（最速下降方向）形成一个严重的角度时，即：当动量向不好的方向变化时则重新启动算法，（5.7）为重新启动判定因子。

在快速前向后向分裂算法的基础上，将重启算法加入到原始快速前向后向分裂算法中，得到改进的自适应重启加速前向后向分裂算法，其算法效率得到大大的提升。其中，自适应重启加速前向后向分裂算法不依赖于对参数 q 的估计值。该方法可用于大规模求解问题。因此本文研究自适应重启快速前向后向分裂算法在图像处理问题中的应用。

自适应重启算法根据不同的条件，可分为以下两种情况：

(1) 函数式重启： $f(x^k) > f(x^{k-1})$ ；

(2) 梯度式重启： $\nabla f(y^{k-1})^T (x^k - x^{k-1}) > 0$ ；

函数式重启技术表现为快速前向后向分裂算法迭代轨迹进入函数低谷而梯度式重启技术将进入不好的方向时的算法重启，形式上类似。但是也有不同之处，尤其特别当迭代点值接近最优值时，梯度式重启方法性能更加的稳定，而且由于在对估计局部 Lipschitz 常数时已经计算负梯度值，所以梯度式重启可以减少计算量，在计算的时间复杂度上存在着明显的优势。

重启算法就是利用判定条件，在迭代中当预处理参数 a^k 变大时，尤其当步长 λ 选择的很大时，使得目标函数值将会出现剧烈的震荡现象。此时重新启动执行算法，把当前迭代作为新的起始点。消除前一步迭代的信息，重置参数 $a^k=1$ ，并且重置动力为 0，减少动力的积累。因此当快速迭代收缩阈值算法在执行效果达到最佳时重启算法。

5.3 自适应重启加速前向后向分裂算法

把当前迭代点作为新的起始点，擦除先前迭代的信息并重置动力为 0。结合快速迭代收缩阈值算法，将自适应重启算法加入到快速前向后向分裂算法中得到自适应重启快速前向后向分裂算法（本文称之为 RSFISTA），具体迭代步骤如下：

算法 8：自适应重启快速前向后向分裂算法（RSFISTA）

1: 初始化: $x_0, t_0, \rho_0 = 1, u_0, \varphi, k = 1$

2: For $k=1, 2, 3 \dots$

$$3: x_{k+1} = \text{prox}(u_k - \rho_k \nabla f(u_k))$$

$$4: \partial_{k+1} = (1 + \sqrt{1 + 4\partial_k}) / 2$$

$$5: u_{k+1} = x_k + (\partial_k - 1) / (\partial_{k+1})(x_k - x_{k-1})$$

$$6: \text{若 } \langle y_k - x_k, x_k - x_{k-1} \rangle \geq 0, \text{ 则 } \partial_{k+1} = 1$$

7: 如果满足终止标准, 算法停止。

改进的快速前向后向分裂算法, 在原有算法基础上增加了梯度式判定因子, 通过判定因子, 当迭代方向出现坏方向时, 则重置算法, 将当前迭代点作为初始迭代点重新启动算法执行。改进算法对原始算法并没有增加计算量的负担。并减少了目标函数的“震荡”现象, 提高了算法的收敛速率。

5.4 实验结果与分析

为了验证本文所提算法, 测试图像为灰度图像, 如图(3.1)所示有四幅, 分别为“cameraman” (256×256)、“lena” (512×512)、“barbara” (512×512)、和“peppers” (512×512)。测试平台为 Window7 系统的笔记本, CPU 为 AMD A6-3400M APU with Radeon(tm) HD Graphics 1.40GHz 四核, 4GB 内存, 仿真平台为 MATLAB 7.14(R2012a)。

5.4.1 基于全变分模型的图像修复

首先测试图像修复, 对原图像添加一个与图像相同尺寸的由 0, 1 组成的随机矩阵, 随机的去除图像的像素, 导致图像出现缺损, 然后再分别采用 FISTA 和 RSFISTA 两种算法进行图像修复实验并对比。



(a).原始图像

(b).降质图像

(c).FISTA 算法修复图像

(d).RSFISTA 算法修复图像

图 5.1 Cameraman 图像修复

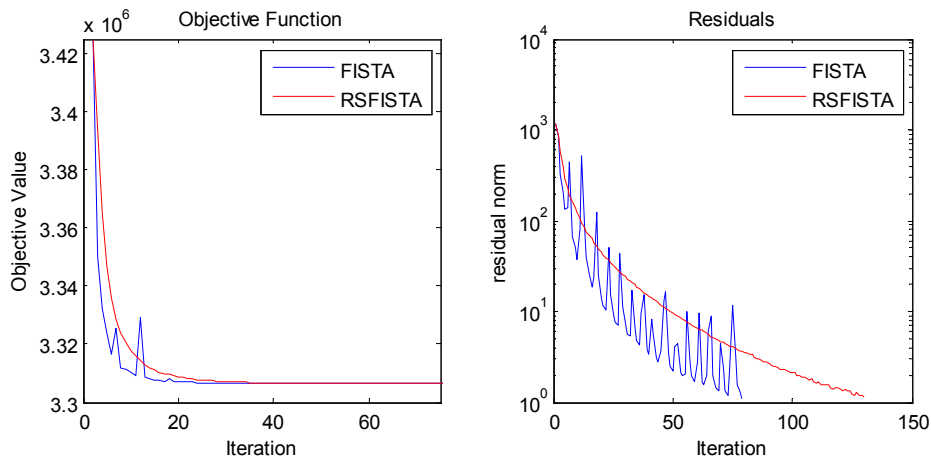


图 5.2 Cameraman 图像修复目标函数值对比图及其残差

表 5.1 Cameraman 图像修复：两种算法性能比较

算法	SNR	Cpu time	SNR	PSNR	SSIM
FISTA	6.6560	123.488s	11.2607	72.7764	0.933
RSFISTA		27.303s	11.2613	72.7770	0.937



(a).原始图像 (b).降质图像 (c).FISTA 算法修复图象 (d).RSFISTA 算法修复图象

图 5.3 Peppers 图像修复

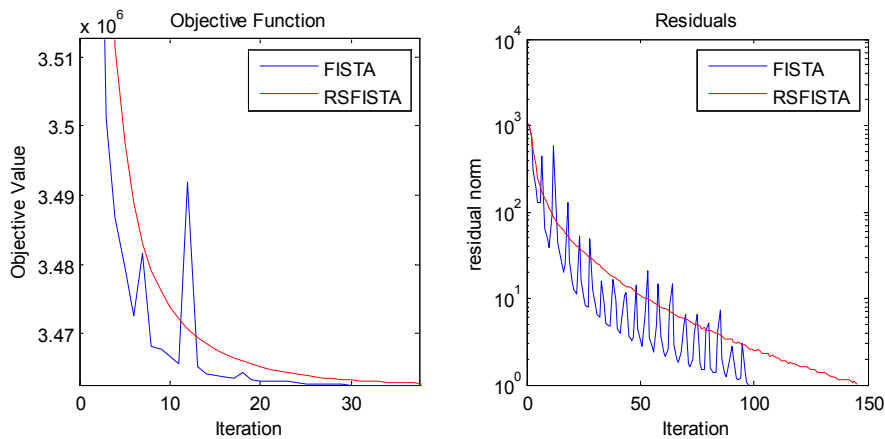


图 5.4 Peppers 图像修复目标函数值对比图及其残差

表 5.2 Peppers 图像修复：两种算法性能比较

算法	SNR	Cpu time	SNR	PSNR	SSIM
FISTA	6.6560	122.8911s	17.1689	78.8021	0.927
RSFISTA		30.6243s	17.1755	78.8086	0.931

如图所示：采用 RSFISTA 算法的修复图像和采用 FISTA 算法修复的图像进行对比，两种算法的相对误差对比图则显示了 RSFISTA 算法具有更小的精度值。从修复图像的目标函数值随迭代次数的对比图及其放大图可知，RSFISTA 算法修复下的图像的目标函数值最终小于 FISTA 算法的目标函数值，说明图像修复图像更接近原图。表格是对本次实验的图像结果的对比，实验分别采用两种算法进行去修复的运行时间、迭代次数、SSIM 和修复图像信噪比的对比。从结果看，RSFISTA 算法在运行时间上比 FISTA 算法大大缩短近运行时间，提高了算法的效率；同时信噪比得到了提升。综上：RSFISTA 算法在图像修复上不仅提高了算法修复效率也提高了算法修复的质量。

5.4.2 基于全变分模型的图像去噪

对图像添加方差为 0.01 的 Gaussian 噪声后，然后分别采用 FISTA 和 RSFISTA 算法对图像进行图像去噪实验并作对比分析。



(a).原始图像

(b).降噪图像

(c).RSFISTA 算法去噪图像

(d).FISTA 算法去噪图

图 5.5 Cameraman 图像去噪

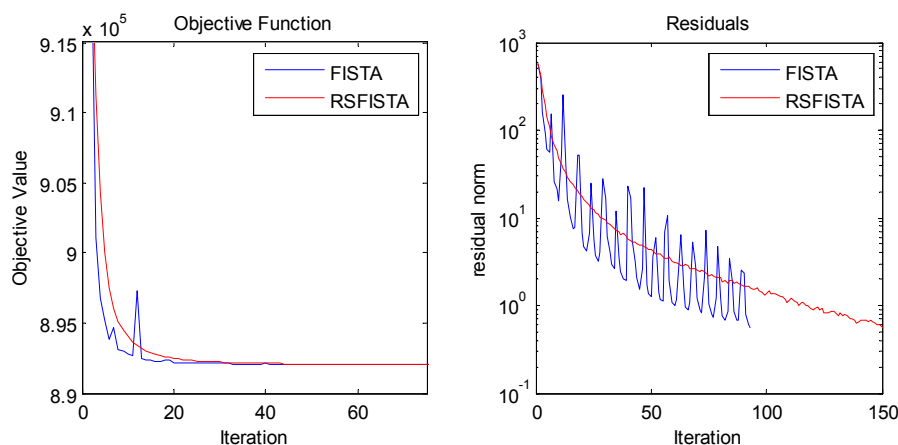


图 5.6 Cameraman 图像去噪目标函数值对比图及其残差

表 5.3 Cameraman 图像去噪：两种算法性能比较

算法	SNR	Cpu time	SNR	PSNR	SSIM
FISTA	6.6560	26.0753s	15.0349	75.4018	0.938
RSFISTA		7.1257s	15.0466	75.4134	0.943



(a).原始图像 (b).降质图像 (c).RSFISTA 算法去噪图像 (d).FISTA 算法去噪图像

图 5.7 Lena 图像去噪

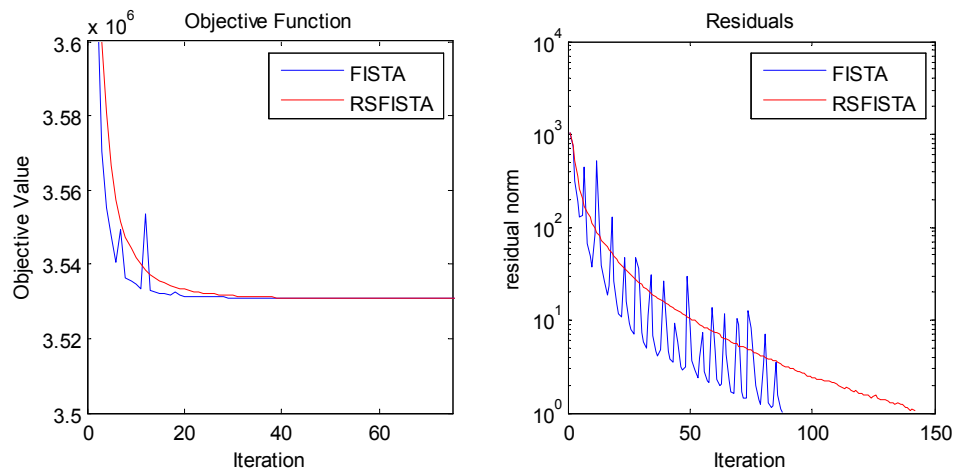


图 5.8 Lena 图像去噪目标函数值对比图及其残差

表 5.4 Lena 图像去噪：两种算法性能比较

算法	SNR	Cpu time	SNR	PSNR	SSIM
FISTA	6.6560	123.445s	15.7177	78.3807	0.935
RSFISTA		33.9094s	15.7260	78.3889	0.942



(a).原始图像 (b).降质图像 (c).RSFISTA 算法去噪图像 (d).FISTA 算法去噪图像

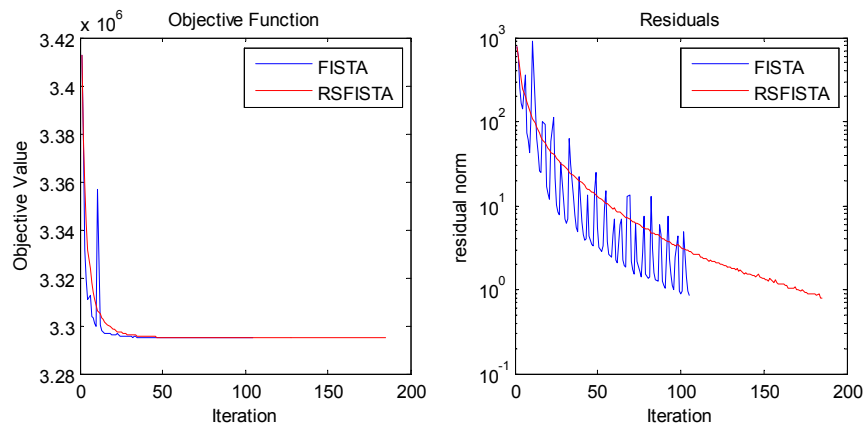


图 5.9 Barbara 图像去噪目标函数值对比图及其残差

表 5.5 Barbara 图像去噪：两种算法性能比较

算法	SNR	Cpu time	SNR	PSNR	SSIM
FISTA	6.6560	100.435s	15.8271	77.2507	0.925
RSFISTA		25.9094s	15.8310	77.2789	0.932

采用 RSFISTA 算法的图像去噪与采用 FISTA 算法的去噪图像进行比较。两算法目标函数随迭代次数的对比图中，其中 RSFISTA 算法下的目标函数值呈现单调下降的趋势，相较于 FISTA 算法，改进算法减少了目标函数的“震荡”现象。并大大的减少了处理图像复原的运行时间，提高了算法的计算效率。表格是对图像都添加高斯噪声后利用两种算法对图像去噪的运行时间、迭代次数、SSIM 和信噪比等信息的对比结果。从结果看 RSFISTA 算法提高了图像去噪的效率，缩短了运行时间，同时提高了图像的质量。综上：RSFISTA 算法不仅提高了图像去噪效率，同时提高了图像去噪的质量。

5.5 本章小结

本章根据快速前向后向分裂算法迭代中存在的“震荡”现象，结合梯度式自适应重启启动技术，得到了自适应重启快速前向后向分裂算法，该算法在不依赖于对参数的估计下，能够通过梯度式判定因子，当迭代朝向坏方向发展时，重启启动算法的执行。通过数值实验，结果表明，基于重启启动技术的快速前向后向分裂算法在计算效率得以提高，算法迭代次数少。

第六章 总结与展望

本文重点研究了全变分模型下前向后向分裂算法的优化算法及其算法性能的相应改进。

6.1 总结

本文的主要工作总结如下：

(1) 全变分模型下的图像复原的自适应谱步长前向后向分裂算法

本章针对全变分模型图像恢复问题，在前向后向分裂算法的基础上提出自适应步长前向后向分裂算法。该算法引入了 Barzilai-Borwein (BB) 法来选取步长，利用算法迭代中之前两步迭代点信息来更新步长因子，达到自适应步长选择的机制。这一迭代过程仅需要计算目标函数值以及梯度信息，降低了算法的计算难度。数值实验表明，自适应前向后向分裂算法获得的复原图像信噪比更高，即图像具有更好的复原质量，同时算法迭代计算时间大大缩短，因此加快了算法收敛的速率。

(2) 全变分图像复原的非单调线性搜索前向后向分裂算法

针对全变分图像复原模型，在快速前向后向分裂方法的基础上提出一种非单调线性搜索前项后项分裂算法，自适应谱步长的选择改进了快速前向后向分裂算法的收敛速率，然而该算法并不能完全保证算法的收敛。根据该项问题，加入回溯线性搜索，判断条件满足时进行算法迭代，否则减小步长因子，直到满足条件。数值结果表明，所提算法应用于全变分图像复原时，图像复原后的信噪比更高，算法运行时间大大缩短，因而提高了算法的计算速率。

(3) 全变分图像复原的自适应重启动前向后向分裂算法

本章针对前向后向分裂算法对参数选择的敏感性的特征，在前向后向分裂算法的基础上加入重启动算子，提出自适应重启动前向后向分裂算法。该算法使用特殊的判定因子，通过计算目标函数差值与相邻迭代点向量之间的关系，判断目标函数出现“震荡”现象。从而启动重启动算子。重启动算法后，之前算法迭代的响应信息将被抹除，将新的迭代点作为初始迭代点进行迭代更新。实验结果表明，改进后的方法大大的提高了算法的收敛速率，同时获得的图像具有更高的峰值信噪比，更清晰的恢复图像。

以上是对一阶前向后向分裂算法的分析与改进，总得来讲，分为自适应改进与加速优化改进，为了呈现不同改进对原始前向后向分裂算法的性能的提高，得到实验图如下：

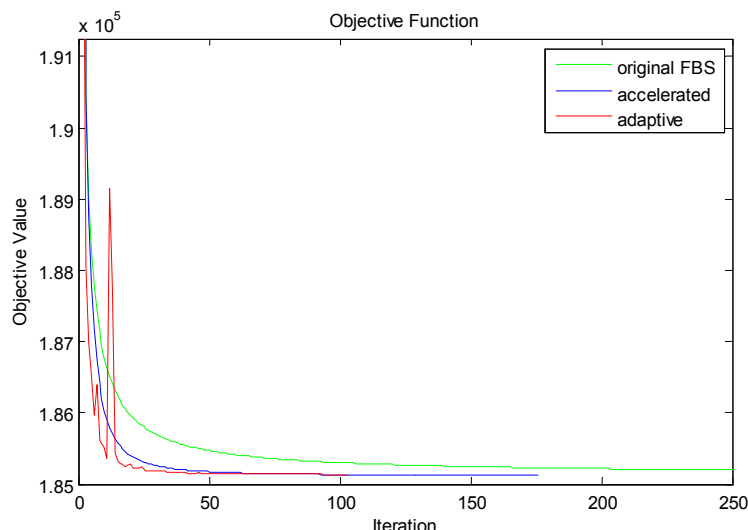


图 6.1 不同优化算法效果对比图

从图中可以看出，自适应和加速算法都提高了原始前向后向分裂算法的性能，不仅使算法收敛的更快，同时也提高了计算的速率。

6.2 展望

本文针对全变分模型图像复原的一阶前向后向分裂算法进行研究，通过对算法参数的改进取得了一定的科研成果, 提高了算法的收敛速率, 在图像复原问题中提高了算法的图像复原质量。但本论文所研究内容还存在不足之处, 还需要孜孜不倦的进行更加深入的科研工作：

(1) 全变分模型图像复原

本文中求解的模型大多数均为全变分图像去噪、修复模型，而在实际中应用的问题中不只是图像去噪，去修复，它们仅仅是图像复原众多子问题中的一个较为简单的应用模型。并且在实验仿真过程中，选用随机的模糊算子来达到图像像素的缺损，而在实际图像去模糊问题中，模糊算子未知并且极为复杂，所以，全变分图像去模糊问题仍就是一个需要克服的难点。

(2) 针对特定应用优化退化模型

现有的图像退化模型大都基于线性或空间不变的假设，而实际应用中并非如此。同时，已知的先验信息的掌握的情况直接影响着图像复原的质量。因此本文研究的图像复原模型由一定的局限性，应该根据不同的应用环境，进行合理的简化，充分的挖掘和利用先验信息，将其添加到退化模型当中。这是值得深入研究和考虑的方法。当然这样做的结果将是增加模型的复杂度，但对图像复原质量将会有更大的提高。在特定的需求下，考虑这种做法。代价

是值得的。

总的来说，由于图像复原是复原时一项广阔发展和应用前景的技术。其处理过程复杂，但随着现代信息处理技术、和数值分析方法的进步，将会有越来越多的新的复原方法不断涌现。希望将来会有更多的学者能够将其领域中适合的研究方法和思路引入到该项领域中，使得图像复原技术能够得到提升。

参考文献

- [1] Chen D Q, Zhang H, Cheng L Z. A fast fixed point algorithm for total variation deblurring and segmentation [J]. *Journal of Mathematical Imaging and Vision*, 2012, 43(3): 167-179.
- [2] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms[J]. *Physica.D.*, 1992,60:259-268.
- [3] Combettes, P., Wajs,V. Signal recovery by proximal forward-backward splitting[J]. *Multiscale Model. Simul.:* *SIAM Interdiscip. J.*, 2005,4:1168-1200.
- [4]S. Osher, M. Burge, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration[J]. *Multiscale Modeling and Simulation*, 2006,4(2):460–489.
- [5] Lim H,Tan K C,Tan B T G. Edge errors in inverse and wiener filter restorations of motion-blurred images and their windowing treatment[J]. *CVGIP*,1991.
- [6] Tikhonov A N, Arsenin V Y. Solutions of ill-posed problems[J]. 1977.
- [7] Chen S S, Donoho D L, Saunders M A. Atomic decomposition by basis pursuit[J]. *SIAM review*, 2001, 43(1): 129-159.
- [8] Chan, T. F., Shen, J. *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*[J]. SIAM, Philadelphia ,2005.
- [9] Figueiredo M A T, Nowak R D, Wright S J. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems[J]. *IEEE Journal of selected topics in signal processing*, 2007, 1(4): 586-597.
- [10]Nesterov Y. Gradient methods for minimizing composite objective function[J]. 2007.
- [11]I. Daubechies, M. De Friesse, and C. DeMol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint[J]. *Communications in Pure and Applied Mathematics*, 2004,57: 1413–1457.
- [12]J. Bioucas-Dias and M. Figueiredo. A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration[J]. *IEEE Transactions on Image Processing*, 2007,16(12):2992-3004.
- [13]T. Goldstein and S. Osher. The split bregman method for l1-regularized problems[J]. *SIAM Journal on Imaging Sciences*,2009,2:323.
- [14]Yang J, Zhang Y. Alternating direction algorithms for l1 problems in compressive sensing [J]. *SIAM journal on scientific computing*, 2011, 33(1): 250-278.
- [15]A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems[J] *SIAM Journal on Imaging Sciences*, 2009, 2:183–202.
- [16]J. Barzilai and J.M. Borwein. Two point step size gradient methods[J]. *IMA J. Numer.Anal.*, 1988, 8:. 141–148..
- [17]Eisenträger A. Non-monotone Line Search and Trust Region Methods for Optimization[J]. University of Oxford, 2007, 15.
- [18]O'Donoghue B, Candes E. Adaptive restart for accelerated gradient schemes[J]. *Foundations of computational mathematics*, 2015, 15(3): 715-732.
- [19]W. K. Allard. Total variation regularization for image denoising, I. Geometric theory[J]. *SIAM J.Math. Anal.*, 2007,39(4):1150–1190.
- [20]Jensen T L. *First-order Convex Optimization Methods for Signal and Image Processing*[M]. *Multimedia Information and Signal Processing*, Department of Electronic Systems, Aalborg University, 2011.
- [21]Micchelli, C.A., Shen, L., Xu, Y. Proximity algorithms for image models: denosing[J]. *Inverse Probl.*2011,27: 30.
- [22]J. Lee, Y. Sun, and M. Saunders. Proximal newton-type methods for convex optimization[J]. *Advances in Neural Information Processing Systems* (P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds.), 2012,25,:836–844.
- [23]Tseng P. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities[J]. *SIAM Journal on Control and Optimization*, 1991, 29(1): 119-138.
- [24]Nesterov Y. A method of solving a convex programming problem with convergence rate $O(1/k^2)$ [C]//*Soviet Mathematics Doklady*. 1983, 27(2): 372-376.
- [25] Lin Z, Ganesh A, Wright J, et al. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix [J]. *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2009,.
- [26] Eldar, Yonina C., Gitta Kutyniok. *Compressed sensing: theory and application*[M]. Cambridge University Press, 2012.
- [27]Goldstein T, Studer C, Baraniuk R. A field guide to forward-backward splitting with a FASTA

- implementation[J]. arXiv preprint arXiv:1411.3406, 2014.
- [28]A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applicationsto imaging[J]. Convergence, 2010,40(1): 1–49.
- [29]Getreuer P. Total Variation Deconvolution using Split Bregman[J]. Image Processing On Line, sub-mitted. <http://www.ipol.im/pub/algo/gtvdeconvolution>, 2012.
- [30]K. Bredies. A forward-backward splitting algorithm for the minimization of non-smooth convex functionals in Banach space[J]. Inverse Problems, 2009,25: 20.
- [31]Micchelli C A, Shen L, Xu Y, et al. Proximity algorithms for the L1/TV image denoising model[J]. Advances in Computational Mathematics, 2013, 38(2): 401-426.
- [32]Beck A, Teboulle M. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems[J]. IEEE Transactions on Image Processing, 2009, 18(11): 2419-2434.
- [33]Chambolle A, Caselles V, Cremers D, et al. An introduction to total variation for image analysis[J]. Theoretical foundations and numerical methods for sparse recovery, 2010, 9: 263-340.
- [34]B. Zhou,L. Gao, and Y.-H.Dai. Gradient methods with adaptive step-sizes[J]. Computer. Optimization. Appl.,2006,35:69–86.
- [35]Birgin E G, Martínez J M, Raydan M. Nonmonotone spectral projected gradient methods on convex sets[J]. SIAM Journal on Optimization, 2000, 10(4): 1196-1211.
- [36]Van Den Berg E, Friedlander M P. SPGL1: A solver for large-scale sparse reconstruction[J]. 2007.
- [37]Van Den Berg E, Friedlander M P. Probing the Pareto frontier for basis pursuit solutions[J]. SIAM Journal on Scientific Computing, 2008, 31(2): 890-912.
- [38]Wen Z, Yin W, Zhang H, et al. On the convergence of an active-set method for ℓ_1 minimization[J]. Optimization Methods and Software, 2012, 27(6): 1127-1146.
- [39]Goldstein T, Setzer S. High-order methods for basis pursuit[J]. UCLA CAM Report, 2010: 10-41.
- [40]Wright S J, Nowak R D, Figueiredo M A T. Sparse reconstruction by separable approximation[J]. IEEE Transactions on Signal Processing, 2009, 57(7): 2479-2493.
- [41]Bonettini S, Ruggiero V. On the convergence of primal-dual hybrid gradient algorithms for total variation image restoration[J]. Journal of Mathematical Imaging and Vision, 2012, 44(3): 236-253.
- [42]He B, Yuan X. Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective[J]. SIAM Journal on Imaging Sciences, 2012, 5(1): 119-149.
- [43]Fletcher R. On the barzilai-borwein method[M]//Optimization and control with applications. Springer US, 2005: 235-256.
- [44]Burachik R, Graña Drummond L M, Iusem A N, et al. Full convergence of the steepest descent method with inexact line searches[J]. Optimization, 1995, 32(2): 137-146.
- [45]Kiwiel K C, Murty K. Convergence of the steepest descent method for minimizing quasiconvex functions[J]. Journal of Optimization Theory and Applications, 1996, 89(1): 221-226.
- [46]Grippo L, Lampariello F, Lucidi S. A nonmonotone line search technique for Newton's method[J]. SIAM Journal on Numerical Analysis, 1986, 23(4): 707-716.
- [47]Zhang H, Hager W W. A nonmonotone line search technique and its application to unconstrained optimization[J]. SIAM journal on Optimization, 2004, 14(4): 1043-1056.
- [48]Yuan G, Lu S, Wei Z. A line search algorithm for unconstrained optimization[J]. Journal of Software Engineering and Applications, 2010, 3(05): 503.
- [49]Nesterov Y. Introductory lectures on convex optimization: A basic course[M]. Springer Science & Business Media, 2013.
- [50]Nemirovskii A, Yudin D B, Dawson E R. Problem complexity and method efficiency in optimization[J]. 1983.
- [51]Michailovich O V. An iterative shrinkage approach to total-variation image restoration.[J]. IEEE Transactions on Image Processing, 2011, 20(5):1281-1299.
- [52]Muckley M J, Noll D C, Fessler J A. Fast Parallel MR Image Reconstruction via B1-based, Adaptive Restart, Iterative Soft Thresholding Algorithms (BARISTA).[J]. IEEE Trans Med Imaging, 2015, 34(2):578.

附录 1 攻读硕士学位期间撰写的论文

(1) 陈少利、杨敏, 改进快速迭代收缩阈值算法, 计算机技术与发展(自然科学版), 已录用。

(2) Chen Shaoli, Yang Min. An Improved Adaptive Fast Iterative Shrinkage-Thresholding Algorithm. Chinese Control and Decision Conference, 2017, 2. Accepted.

致谢

时光荏苒，三年研究生生涯如白驹过隙，回首进入南京邮电大学求学的美好时光，期间得到了许多人的陪伴与关心，在论文即将完成之际，谨向所有关心我、爱护我、支持我的人们送上自己最真挚的感谢。

首先，我要感谢我的导师杨敏老师。感谢杨敏老师对本文的研究工作的悉心指导，他的认真细致和富有启发性的建议确保了论文的顺利完成。杨老师在我的研究生学习期间对我学术提供了极大的帮助，引导我在学术上的研究和创新，研究过程遇到瓶颈，杨老师不厌其烦的为我指点迷津，启发我打开思路。大量的文献阅读，不仅提高了我的学术专业背景知识，丰富了我对图像处理相关领域的认识。

其次，我要感谢我的母校：南京邮电大学。感谢母校给与我研究学习的良好环境，让我能够专心做好科研。在南邮的三年里我不仅遇到了许多优秀的教师，同时也结识了很多志同道合的伙伴。感谢学校、感谢学校的每一位领导、老师和同学，感谢我的室友夏雪婷、王梦欣、庞小双在学习和生活中对我的帮助和照顾，感谢王臻、李策、唐松对我学习上的帮助。

我要感谢我最最亲爱的父母，感谢他们对我养育，在我的学习和生活中给予的无微不至关心和支 持。他们的理解与爱护，让我有勇气面对生活中的各种问题。让我拥有独立奋斗的信心。

最后，再次感谢所有关心、爱护和支持我的朋友们，感谢审阅本论文付出辛勤劳动的各位专家和学者。