

基于多尺度去噪正则化和深度神经网络的相位恢复

李雪明*, 郭兵

吉首大学数学与统计学院, 湖南 吉首

收稿日期: 2024年4月29日; 录用日期: 2024年5月22日; 发布日期: 2024年5月31日

摘要

相位恢复问题是恢复信号丢失或损坏的相位信息, 以实现信号的准确分析、处理和重建。相位恢复在许多领域都有广泛的应用, 例如通信系统、图像处理和计算机视觉、雷达和无线电频谱感知和生物医学工程等。然而, 传统的相位恢复算法在噪声存在的情况下很难实现恢复任务。幸运的是, 各种先进算法的不断提出和改进以及深度学习的兴起为处理相位恢复问题提供了很大的帮助。为了解决单个去噪正则项不能涵盖图像的全部先验信息的问题, 本文考虑多尺度的去噪正则化求解带有噪声的相位恢复问题, 它基于去噪正则化框架, 而且它通过在优化问题中引入正则化项来实现去除图像中的噪声。为了解决将去噪正则化和FFDNet一起考虑求解带有噪声的相位恢复问题时恢复效果不好的问题, 本文充分利用DnCNN和FFDNet这两种去噪器的优点, 在相位恢复的过程中分情况使用这两种去噪器。最后本文通过数值实验表明了改进策略在视觉效果和量化数值上都有更加优异的表现。

关键词

相位恢复, 去噪正则项, 深度学习, Fourier测量, 图像去噪

Phase Retrieval Based on Multi-Scale Regularization by Denoising and Deep Neural Networks

Xueming Li*, Bing Guo

School of Mathematics and Statistics, Jishou University, Jishou Hunan

Received: Apr. 29th, 2024; accepted: May 22nd, 2024; published: May 31st, 2024

*通讯作者。

文章引用: 李雪明, 郭兵. 基于多尺度去噪正则化和深度神经网络的相位恢复[J]. 应用数学进展, 2024, 13(5): 2296-2310. DOI: 10.12677/aam.2024.135217

Abstract

Phase retrieval problem is to recover the lost or damaged phase information in order to realize the accurate analysis, processing and reconstruction of the signal. Phase retrieval is widely used in many fields, such as communication systems, image processing and computer vision, radar and radio spectrum sensing and biomedical engineering. However, the traditional phase retrieval algorithms are difficult to achieve the recovery task in the presence of noise. Fortunately, the continuous proposal and improvement of various advanced algorithms and the rise of deep learning provide great help to deal with the phase retrieval problem. In order to solve the problem that a single denoising regularization cannot cover all the prior information of the image, this paper considers multi-scale denoising regularizations to solve the phase retrieval problem disturbed by noise, which is based on the regularization by denoising framework and it removes the noise in the image by introducing a regularization term into the optimization problem. Additionally, in order to solve the problem of poor recovery effect when regularization by denoising and FFDNet are considered together to solve the phase retrieval contaminated with noise, this paper makes full use of the advantages of DnCNN and FFDNet, so these two kinds of denoising in different cases are combined to deal with the phase retrieval. Finally, numerical experiments show that the improved strategies have better performance in visual effect and quantitative value.

Keywords

Phase Retrieval, Denoising Regularization, Deep Neural Networks, Fourier Measurement, Imaging Denoising

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

在许多实际应用中, 信号的相位往往会受到噪声、失真或其他干扰的影响, 导致相位信息被破坏或模糊化。这可能导致信号在传输、存储或处理过程中的失真, 影响到后续的信号分析和处理。相位恢复[1]旨在通过各种信号处理技术来还原或修复受损的相位信息[2] [3]。其背后的基本原理是利用已知的信号特性或先验知识来推测缺失或损坏的相位[4]。相位恢复的方法通常基于信号的统计特性、周期性特征或其他可用的信息[5]。相位恢复在许多领域都有广泛的应用, 包括通信系统、雷达信号处理、图像处理、音频处理等。通过恢复受损的相位信息[6], 可以提高信号的质量和可靠性, 从而实现更准确的信号分析、恢复和提取[7]。

相位恢复问题是指从线性系统输出的振幅或强度中恢复输入的信号, 所以相位恢复问题是一种逆问题[8] [9]。一般的逆问题表示成下面的优化问题的形式:

$$\min_{\mathbf{x}} l(\mathbf{b}, f(\mathbf{x})) + \lambda R(\mathbf{x}) \quad (1)$$

其中 \mathbf{x} 表示对输入信号的估计, 即优化变量, \mathbf{b} 是观测到的系统输出的值。 $l(\mathbf{b}, f(\mathbf{x}))$ 表示数据保真项, 即保证 $\mathbf{b} \approx f(\mathbf{x})$, $R(\mathbf{x})$ 表示 \mathbf{x} 的先验信息, 作为目标函数的正则项有助于求解优化问题, λ 表示对 $R(\mathbf{x})$ 的惩罚程度。

对于具体的相位恢复问题, 可以把它写成下面的形式:

$$\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda R(\mathbf{x}) \quad (2)$$

相位恢复算法可以追溯到 1970s, 1972 年 Gerchberg 提出了最早的相位恢复算法[10], 然后在 1978 年 Fienup 对其进行了改进, 即 HIO 算法(混合输入输出算法) [11]。相位恢复算法一直不断地发展, 而且与很多领域相交叉[12] [13]。随着优化算法的不断发展, 它们为解决相位恢复问题提供了不同角度的帮助。比如随着压缩感知的发展, 很多学者将压缩感知应用到相位恢复算法, 进而提高相位恢复效果[14] [15]。但是在目标信号是密集信号的时候这种方法不能提高恢复效果。

随着去噪领域算法的成熟, 即插即用框架被提出来了, 很多学者考虑将去噪算法广泛应用到众多逆问题中。基于拉格朗日增广和变量分裂思想的 ADMM 算法(Alternating Direction Method of Multipliers) [16] 在相位恢复问题上展现了良好的表现, 但是基于 ADMM 算法的相位恢复具有多个参数选择的困难。RED(Regularization by Denoising) [17]提出了去噪正则化框架, 并表现出了极大的灵活性和适用性。在 prDeep [18]提出了将去噪正则化框架应用到含有噪声的相位恢复问题, 并且表现了很好的效果。但是 prDeep 在含有高噪声的频谱信息时的恢复效果较差。

随着深度学习[19]的发展, 神经网络广泛应用到图像处理的各个方面, 论文[20]将相位恢复问题看作一般的逆问题, 应用神经网络训练一个观测信号到期望恢复信号的映射。它考虑先打破相位恢复中的对称性, 然后使用深度学习训练一个神经网络, 进而从信号的振幅信息求解出信号的相位信息, 但是这种方法的一个局限是它只适用于与训练数据集相关的图像。prDeep 是一种基于去噪正则和 DnCNN [21]求解带有噪声的相位恢复的方法, 它提升了恢复效果, 但是仍存在对复杂噪声鲁棒性较差的问题。因为在 prDeep 中使用的去噪器是去噪卷积神经网络(DnCNN), 我们分析是单个的去噪器先验不能包含图像的全部先验信息。所以我们考虑使用两个不同的去噪正则化解决相位恢复问题, 即多尺度的去噪正则化。在 FFDNet (Fast and Flexible Denoising Network) [22]中介绍了对于图像去噪, 在高斯噪声的参数即标准差 $\sigma \leq 25$ 时 DnCNN 的去噪效果比 FFDNet 的去噪效果好, 而在 $\sigma > 25$ 时 FFDNet 的去噪效果比 DnCNN 的去噪效果好。所以我们考虑在相位恢复的过程中分情况使用 DnCNN 和 FFDNet 这两种去噪器, 即在 $\sigma \leq 25$ 时使用 DnCNN 作为去噪器, 而在 $\sigma > 25$ 时使用 FFDNet 作为去噪器。然后在测试图像为 Set12 数据集下考虑在不同的 Poisson 噪声水平时图像的恢复情况, 并比较它们的视觉效果和量化数值。

在我们的工作中, 我们做出了三个贡献。第一个贡献是使用多尺度的去噪正则化求解带有噪声的相位恢复问题, 它基于去噪正则化框架, 我们通过数值实验表明了这种改进的优点。第二个贡献是我们将 RED 和 FFDNet 一起考虑求解带有噪声的相位恢复问题。第三个贡献是我们在相位恢复的过程中分情况使用 DnCNN 和 FFDNet 这两种去噪器。最后我们通过数值实验比较了我们的改进策略在视觉效果和量化数值上都有更加优异的表现。

在第一章我们介绍相位恢复的研究背景和动机, 以及我们的论文贡献和论文结构。在第二章我们介绍图像去噪引擎和 FBS (前向后向分裂)算法[23]及其改进版本的相关知识。在第三章介绍我们本文的工作, 包括使用多尺度去噪正则化求解含噪声相位恢复问题和分情况使用 DnCNN 和 FFDNet 这两种去噪器求解含噪声相位恢复问题。在第四章展示了我们的数值实验结果, 包括实验的一些设置、实验的数据和可视化比较以及实验结果分析。第五章是我们的总结和对未来的展望。

2. 预备知识

2.1. 图像去噪引擎

图像去噪[24]是数字图像处理中的一项重要任务, 它旨在从图像中消除或减少噪声的存在, 以提高图像的质量和清晰度。噪声是由于图像获取、传输、存储或处理过程中的各种因素引起的不希望的干扰或失真。常见的噪声类型包括泊松噪声[25]、高斯噪声[26]、椒盐噪声等。图像去噪在很多领域中都有它的应用, 比如计算机视觉、生物医学和半导体制造等。图像去噪的目标是从一个带有噪声的观测图像 \mathbf{y} 中

恢复出干净的图像 \mathbf{x} , 它的模型为: $\mathbf{y} = \mathbf{x} + \mathbf{v}$ 。其中 \mathbf{v} 一般是加性白高斯噪声或者泊松噪声。从这个模型中可以看出图像去噪是一种逆问题。

图像去噪的发展历史可以追溯到早期的数字图像处理领域。在 20 世纪 60 年代和 70 年代, 早期的图像去噪方法主要基于统计滤波理论。这些方法通过应用统计模型和滤波技术来减少图像中的噪声。在 70 年代和 80 年代, 线性滤波方法被广泛研究和应用。这些方法包括均值滤波[27]、高斯滤波和中值滤波[28]等。线性滤波方法在简单噪声模型下表现良好, 但对于复杂噪声和图像细节保护的要求有限。在 80 年代末和 90 年代初, 小波去噪方法[29]的出现引起了广泛关注。小波去噪通过对小波系数进行阈值处理来去除噪声, 并在不同尺度上分别处理图像的细节和噪声。小波去噪方法在保留图像边缘和细节方面表现出色, 成为图像去噪领域的重要方法。2000 年以后, 基于偏微分方程的图像去噪方法引起了广泛关注。这些方法基于偏微分方程的演化过程, 利用图像的局部特征来去除噪声。著名的方法有非局部均值去噪 (Non-local Means Denoising) [30]等方法。近年来, 深度学习方法在图像去噪领域取得了重大突破。通过使用深度神经网络, 特别是自编码器、生成对抗网络(GAN)等模型, 可以学习从含噪图像到无噪图像之间的映射关系, 实现高质量的图像去噪。

去噪正则化框架是一种常用的图像去噪方法, 它通过在优化问题中引入正则化项来约束解的平滑性或稀疏性, 从而实现去除图像中的噪声。该框架的基本思想是在最小化数据保真项和正则化项之间取得平衡, 以获得最佳的去噪结果。与其他即插即用算法一样, 去噪正则化算法可以使用任意去噪器来正则化任意一个图像逆问题[31] [32], 不同之处在于别的方法使用去噪器来最小化隐式损失函数, 但是去噪正则化框架使用去噪器最小化显式损失函数, 并保证收敛到全局最优解。它通过将正则化问题转化为一个去噪问题, 并利用强大的去噪算法来恢复干净的信号。这使得 RED 方法在处理受噪声干扰的数据时表现出很强的鲁棒性。RED 方法可以适用于各种线性逆问题, 包括图像恢复、信号恢复、压缩感知等。它不依赖于特定的问题设置, 因此可以广泛应用于多个领域和任务。相对于传统的正则化方法, RED 方法不需要先验信息或者对噪声进行严格的建模。

去噪正则化框架定义的正则项为

$$R(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T (\mathbf{x} - D(\mathbf{x})) \quad (3)$$

其中 D 是去噪器。

那么我们的优化模型为

$$\arg \min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \frac{1}{2} \mathbf{x}^T (\mathbf{x} - D(\mathbf{x})) \quad (4)$$

2.2. FBS 算法及其改进版本的介绍

FBS 算法考虑这样的优化问题

$$\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x}) \quad (5)$$

其中 \mathbf{x} 属于 R^N , f 是凸的和可微的函数, g 是任意的凸函数。

在很多情况下, 函数 g 既不是可微的也不是有限值, 在这种情况下, 上面的优化问题不能使用一般的梯度下降法求解, 但是我们可以使用 FBS (Forward-Backward Splitting)方法, 又叫作近端梯度下降法。然后我们求函数 g 的近端算子, 其中 τ 是步长。

$$\text{prox}_g(\mathbf{z}, \tau) = \arg \min_{\mathbf{x}} \tau g(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 \quad (6)$$

FBS 算法通过将整体问题分解为前向和后向子问题, 使得问题的求解更加可行。这种分解方式可以帮助降低问题的复杂性, 使得问题更易于理解和处理。**FBS** 算法是一种迭代算法, 每次迭代都通过前向和后向步骤逐渐改进变量的值。这种迭代的方式可以帮助算法逐步逼近最优解, 并在每次迭代中提供一个改进的解。**FBS** 适用于具有分离可加结构的问题, 这种结构在许多应用领域中都很常见。因此, 该算法可以用于解决各种优化问题, 包括信号处理、机器学习和图像处理等领域。通过交替进行前向和后向步骤, **FBS** 算法可以逐渐逼近优化问题的最优解。这个算法的优点是能够有效地处理目标函数中可微和非光滑或不可微部分的组合, 并在每个步骤中针对不同的部分进行优化。而且 **FBS** 算法在处理凸优化问题时可以收敛到全局最小点。具体的 **FBS** 算法过程见算法 1。

由于 **FBS** 算法的优秀思想, 有很多学者对其提出了不同的改进版本, 比如 **FISTA** (**Fast Iterative Shrinkage-Thresholding Algorithm**)对 **FBS** 算法进行加速。**FISTA** 算法相对于 **FBS** 算法的改进主要体现在更新迭代点的步骤上。**FISTA** 引入了额外的加速参数 α 和使用预测校正方法, 使得算法能够更快地收敛。在每次迭代中, **FISTA** 根据前一次迭代的结果和当前的估计值来更新迭代点, 并利用步长参数 α 来加速收敛过程。具体的过程见算法 2。

在算法 2 中, $L(\nabla f)$ 是 ∇f 的 Lipschitz 常数, 即需要满足

$$\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| < L\|\mathbf{x}_1 - \mathbf{x}_2\| \quad \forall \mathbf{x}_1, \mathbf{x}_2 \quad (7)$$

当步长 $\tau < 1/L(\nabla f)$ 时这个算法是收敛的。算法 2 的第三步就是最原始的 **FBS** 算法, 算法第四步是加速参数 α 的更新过程, 这里的 α 随着迭代过程不断增大, 当目标函数增大时采取重启方法, 即令 $\alpha = 1$ 。算法 2 的第五步是一个预测过程, 当前迭代在上一次迭代的前进方向更近一步。这种方法可以加快算法的收敛速度。

算法 1 FBS(Forward-Backward Splitting)

- 1: 输入 $\mathbf{x}^{(0)}$, $k = 0$, $\tau < 1/L(\nabla f)$
 - 2: **for** $k = 1, 2, 3, \dots, MAXITER$ **do**
 - 3: $\hat{\mathbf{x}}^{(k)} = \mathbf{x}^{(k-1)} - \tau \nabla f(\mathbf{x}^{(k-1)})$
 - 4: $\mathbf{x}^{(k)} = \text{prox}_g(\hat{\mathbf{x}}^{(k)}, \tau) + \frac{1}{2} \|\mathbf{x} - \hat{\mathbf{x}}^{(k)}\|_2^2$
 - 5: **end for**
 - 6: 输出 $\mathbf{x}^{(MAXITER)}$
-

算法 2 FISTA

- 1: 输入 $\mathbf{y}^{(1)} = \mathbf{x}^{(0)} \in R^n$, $\alpha^{(1)} = 1$, $\tau < 1/L(\nabla f)$
 - 2: **for** $k = 1, 2, 3, \dots, MAXITER$ **do**
 - 3: $\mathbf{x}^{(k)} = \text{prox}_g(\mathbf{y}^{(k)} - \tau \nabla f(\mathbf{y}^{(k)}))$
 - 4: $\alpha^{(k+1)} = \left(1 + \sqrt{1 + 4(\alpha^{(k)})^2}\right) / 2$
 - 5: $\mathbf{y}^{(k+1)} = \mathbf{x}^{(k)} + \frac{\alpha^{(k)} - 1}{\alpha^{(k+1)}} (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$
 - 6: **end for**
 - 7: 输出 $\mathbf{x}^{(MAXITER)}$
-

当 $\tau < 2/L(\nabla f)$ 时可以保证 FBS 收敛, 但是实际上我们可能不知道 ∇f 的性质, 从而我们不知道步长应该取多少。因此我们可以使用回溯线搜索方法保证收敛性, 即使不知道步长约束条件。这种方法在每次 FBS 的迭代之后检查线搜索条件, 当条件不成立时进行回溯, 即减小步长然后重新执行 FBS 迭代过程直到满足线搜索条件。线搜索方法使目标函数单调下降, 但是很多学者认为它太过于保守, 所以提出了非单调线搜索。单调线搜索坚持在每次迭代之后让目标函数减小, 但是非单调线搜索允许目标函数在一定范围内增加。因为非单调线搜索条件比单调线搜索宽松, 所以非单调线搜索条件更不容易被违背, 这意味着更少的回溯步骤, 那么运行时间也会相应地减少。

设 M 是一个线搜索参数, M 是一个正整数, 然后定义

$$\hat{f}^{(k)} = \max \left\{ f^{(k-1)}, f^{(k-2)}, \dots, f^{(k-\min\{M, k\})} \right\} \quad (8)$$

$$f(\mathbf{x}^{(k+1)}) < \hat{f}^{(k)} + \Re \left\langle \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}, \nabla f(\mathbf{x}^{(k)}) \right\rangle + \frac{1}{2\tau^{(k)}} \left\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\|^2 \quad (9)$$

如果违反了回溯条件(9), 那么减小步长直到它成立, 非单调线搜索的过程在算法 3 中。而且, 使用了非单调线搜索的 FBS 算法是收敛的。

算法 3 非单调线搜索

- 1: **While** $\mathbf{x}^{(k)}$ 和 $\mathbf{x}^{(k+1)}$ 违反条件(9) **do**
 - 2: $\tau^{(k)} \leftarrow \tau^{(k)}/2$
 - 3: $\mathbf{x}^{(k+1)} \leftarrow \text{prox}_g \left(\mathbf{x}^{(k)} - \tau^{(k)} \nabla f(\mathbf{x}^{(k)}), \tau^{(k)} \right)$
 - 4: **end while**
-

在本文中我们使用更加方便和完善的 FASTA (Fast Adaptive Shrinkage-Thresholding Algorithm) 求解器。FASTA 求解器包含了 FBS 的优点, 而且使用了自适应步长加速求解过程, 可以让我们的问题快速收敛。

3. 我们的工作

一个去噪正则项中只是使用了单个去噪器, 那么这时可能不能够涵盖图像的全部先验信息, 所以我们考虑使用多尺度的去噪正则项, 即使用两个去噪正则项:

$$R_1(\mathbf{x}) = \frac{\beta_1}{2} \mathbf{x}^T (\mathbf{x} - D_1(\mathbf{x})) \quad (10)$$

$$R_2(\mathbf{x}) = \frac{\beta_2}{2} \mathbf{x}^T (\mathbf{x} - D_2(\mathbf{x})) \quad (11)$$

其中 β_1 和 β_2 分别是两个不同的去噪器。那么这时优化模型为:

$$\arg \min_{\mathbf{x}} \frac{1}{2\sigma^2} \left\| \mathbf{Ax} - \mathbf{b} \right\|_2^2 + \frac{\beta_1}{2} \lambda \mathbf{x}^T (\mathbf{x} - D_1(\mathbf{x})) + \frac{\beta_2}{2} \lambda \mathbf{x}^T (\mathbf{x} - D_2(\mathbf{x})) \quad (12)$$

其中 σ 是高斯噪声的标准差。

对于这个优化问题, 我们使用 FASTA 算法进行求解, 那么令

$$f(\mathbf{x}) = \frac{1}{2\sigma^2} \left\| \mathbf{Ax} - \mathbf{b} \right\|_2^2 \quad (13)$$

$$g(\mathbf{x}) = \frac{\beta_1}{2} \lambda \mathbf{x}^T (\mathbf{x} - D_1(\mathbf{x})) + \frac{\beta_2}{2} \lambda \mathbf{x}^T (\mathbf{x} - D_2(\mathbf{x})) \quad (14)$$

首先对 $f(\mathbf{x})$ 进行梯度下降, 取步长为 τ , 即

$$\hat{\mathbf{x}}^{(k)} = \mathbf{x}^{(k)} - \tau \nabla f(\mathbf{x}^{(k)}) \quad (15)$$

因为这里的 $f(\mathbf{x})$ 是不可微的, 我们令 $\mathbf{z} = \mathbf{Ax}$, 所以我们取它的一个次梯度

$$\nabla f(\mathbf{z}) = \mathbf{z} - \mathbf{b} \cdot \frac{\mathbf{z}}{\|\mathbf{z}\|} \quad (16)$$

其中 \cdot 表示对应元素相乘。

然后再求 $g(\mathbf{x})$ 的近端算子, 即

$$\mathbf{x}^{(k+1)} = \text{prox}_g(\hat{\mathbf{x}}^{(k)}, \tau) + \frac{1}{2} \|\mathbf{x} - \hat{\mathbf{x}}^{(k)}\|_2^2 = \arg \min_{\mathbf{x}} \tau g(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \hat{\mathbf{x}}^{(k)}\|_2^2 \quad (17)$$

根据 prDeep, 我们发现对它只进行一次迭代就能有很好的效果, 所以我们可以得到

$$\mathbf{x}^{(k+1)} = \frac{\hat{\mathbf{x}}^{(k)} + \lambda \tau (\beta_1 D_1(\hat{\mathbf{x}}^{(k)}) + \beta_2 D_2(\hat{\mathbf{x}}^{(k)}))}{1 + \lambda \tau} \quad (18)$$

另一方面, 我们考虑在相位恢复的过程中分情况使用 DnCNN 和 FFDNet 这两种去噪器。因为我们用四种从大到高的高斯噪声标准差的去噪器先去除高水平的噪声, 再去去除低水平的噪声, 所以在 $\sigma \leq 25$ 时使用 DnCNN 作为去噪器, 而在 $\sigma > 25$ 时使用 FFDNet 作为去噪器。

算法 4 多尺度去噪正则化的相位恢复

- 1: 输入 $\mathbf{y}^{(1)} = \mathbf{x}^{(0)} \in R^N$, $\alpha^{(1)} = 1$, $\tau < 1/L(\nabla f)$
 - 2: **for** $k = 1, 2, 3, \dots, MAXITER$ **do**
 - 3: $\hat{\mathbf{x}}^{(k)} = \mathbf{y}^{(k)} - \tau \nabla f(\mathbf{y}^{(k)})$
 - 4: $\mathbf{x}^{(k)} = \frac{1}{1 + \lambda \tau} (\hat{\mathbf{x}}^{(k)} + \lambda \tau (\beta_1 D_1(\hat{\mathbf{x}}^{(k)}) + \beta_2 D_2(\hat{\mathbf{x}}^{(k)})))$
 - 5: $\alpha^{(k+1)} = \left(1 + \sqrt{1 + 4(\alpha^{(k)})^2}\right) / 2$
 - 6: $\mathbf{y}^{(k+1)} = \mathbf{x}^{(k)} + \frac{\alpha^{(k)} - 1}{\alpha^{(k+1)}} (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$
 - 7: **end for**
 - 8: 输出 $\mathbf{x}^{(MAXITER)}$
-

4. 数值实验

在这一章我们展示数值实验结果, 介绍实验的一些设置并展示不同方法的恢复效果, 包括视觉效果和量化数值。

4.1. 实验设置

4.1.1. 测量和噪声模型

在本文中我们考虑傅里叶测量。傅里叶测量是一个非常著名的测量, 它基于傅里叶变换的原理, 将信号从时间域转换为频率域。傅里叶测量在很多领域都有广泛的应用[33], 包括光谱学、光学显微镜、遥感、化学分析等。它能够提供关于物质的光学性质、化学成分、结构信息等方面的重要数据[34]。

在本文中, 我们考虑泊松噪声。泊松噪声在许多领域中都有应用, 包括通信、成像、物理实验和统

计分析等。我们考虑模型

$$\mathbf{y}^2 = |\mathbf{Ax}|^2 + \omega \text{ with } \omega \sim N\left(0, \alpha^2 \text{Diag}\left(|\mathbf{Ax}|^2\right)\right) \quad (19)$$

其中 $\text{Diag}\left(|\mathbf{Ax}|^2\right)$ 是 $|\mathbf{Ax}|^2$ 的对角元素的对角矩阵。由代数学知识可知 $\mathbf{y}^2/\alpha^2 \sim \text{Poisson}\left(\left(|\mathbf{Ax}|/\alpha\right)^2\right)$, 其中 α 是泊松噪声的参数, \mathbf{y}^2 是加入了泊松噪声之后的值。

4.1.2. 参数选择和初始化

我们使用采样率为 4 的傅里叶测量, 测试图像为 Set12 数据集, 使用的是 128×128 的灰度图像。对于正则项参数我们选取 σ_ω , 其中 σ_ω^2 表示样本噪声的方差。因为傅里叶测量方法对初值的选取是敏感的, 所以我们使用 HIO 算法作为选取初始值的方法。我们的算法迭代 200 次, $g(\mathbf{x})$ 的近端迭代 1 次。其他的实验细节我们参考 prDeep。在多尺度去噪正则项里使用的两个去噪器分别是 DnCNN 和 BM3D [35], 其中 DnCNN 是基于机器学习的去噪器, BM3D 是传统的去噪器, 它们的去噪效果都很好。所有的数值实验均在配备了 Ryzen7 5800H CPU 和 Nvidia Geforce RTX 3070 GPU 的笔记本电脑上使用 Matlab 2022a 进行测试。

4.2. 实验结果与分析

我们进行了五种算法在 Set12 数据集上的数值实验, 我们分别实验了它们在不同泊松噪声的噪声水平和不同测试图像上的相位恢复效果, 下面展示它们的 PSNR(峰值信噪比)、MSSIM(平均结构相似性)、运行时间和视觉比较图。其中运行时间的单位是秒, 当像素值的范围是 $[0, 255]$ 时,

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{mean}\left((\hat{x} - x_0)^2\right)} \quad (20)$$

单位是 dB。

表 1 记录了五种算法在 Set12 数据集的六幅图像上的 PSNR, 表 2 记录了它们的 MSSIM, 表 3 记录了五种算法的运行时间, 我们分别考虑了和的泊松噪声。从表 1 中我们可以看到 HIO 算法的恢复效果最差, 这是因为 HIO 算法不能去除频谱中的噪声。我们发现使用多尺度去噪正则化的相位恢复算法(RED-(DnCNN-

Table 1. PSNR of five algorithms on six images

表 1. 五种算法在六幅图像上的 PSNR

| | | Cameraman | Pepper | Monarch | Parrot | Lena | Barbara |
|--------------|--------------------|-----------|---------|---------|---------|---------|---------|
| $\alpha = 2$ | HIO | 22.4483 | 22.8755 | 18.5480 | 22.3075 | 22.5286 | 22.6210 |
| | prDeep | 37.4303 | 37.6766 | 37.3592 | 35.8606 | 37.1216 | 35.3802 |
| | RED-FFDNet | 32.2827 | 31.0767 | 31.1128 | 29.3208 | 30.2983 | 28.3363 |
| | RED-(DnCNN-BM3D) | 37.5107 | 36.7210 | 37.5263 | 35.9037 | 37.2331 | 35.8834 |
| | RED-(DnCNN-FFDNet) | 38.0472 | 37.8300 | 37.5827 | 36.1371 | 37.7021 | 35.7919 |
| $\alpha = 3$ | HIO | 21.0601 | 21.4198 | 18.3008 | 20.4148 | 20.2428 | 20.2640 |
| | prDeep | 34.4134 | 34.0489 | 34.1102 | 32.5135 | 34.0970 | 33.8306 |
| | RED-FFDNet | 31.4831 | 29.6359 | 30.4202 | 28.3931 | 28.8669 | 28.0485 |
| | RED-(DnCNN-BM3D) | 35.3962 | 34.1891 | 34.4473 | 34.1080 | 34.2989 | 35.4032 |
| | RED-(DnCNN-FFDNet) | 34.5469 | 34.5316 | 34.3465 | 33.0877 | 34.3741 | 34.1381 |

BM3D))和使用 DnCNN 和 FFDNet 分情况去噪的相位恢复算法(RED-(DnCNN-FFDNet))的 PSNR 和 MSSIM 都比使用 FFDNet 去噪的相位恢复算法(RED-FFDNet)高。不论是在还是时, (RED-(DnCNN-BM3D))和(RED-(DnCNN-FFDNet))在六幅图像上大部分情况的 PSNR 和 MSSIM 都比 prDeep 高。从表 3 中我们可以看到(RED-(DnCNN-BM3D))的运行时间最高, 在 100 秒左右, 这是因为 BM3D 是传统的去噪算法, 它需要进行块匹配、聚类 and 三维滤波等多个步骤, 而每个步骤都涉及到大量的计算操作, 所以运行时间较高。

Table 2. MSSIM of five algorithms on six images

表 2. 五种算法在六幅图像上的 MSSIM

| | | Cameraman | Pepper | Monarch | Parrot | Lena | Barbara |
|--------------|--------------------|-----------|---------|---------|---------|---------|---------|
| $\alpha = 2$ | HIO | 0.47647 | 0.62499 | 0.50991 | 0.58321 | 0.56533 | 0.64094 |
| | prDeep | 0.95303 | 0.96719 | 0.98096 | 0.96120 | 0.96561 | 0.97078 |
| | RED-FFDNet | 0.91919 | 0.92341 | 0.94712 | 0.89491 | 0.90700 | 0.90344 |
| | RED-(DnCNN-BM3D) | 0.96800 | 0.97219 | 0.98417 | 0.96525 | 0.97097 | 0.97736 |
| | RED-(DnCNN-FFDNet) | 0.95373 | 0.96904 | 0.97959 | 0.96281 | 0.96916 | 0.97771 |
| $\alpha = 3$ | HIO | 0.41750 | 0.55909 | 0.49224 | 0.49585 | 0.45184 | 0.54067 |
| | prDeep | 0.89789 | 0.93591 | 0.95503 | 0.90835 | 0.93148 | 0.95484 |
| | RED-FFDNet | 0.90840 | 0.90763 | 0.93889 | 0.88147 | 0.88290 | 0.89378 |
| | RED-(DnCNN-BM3D) | 0.94804 | 0.95621 | 0.97272 | 0.94774 | 0.94884 | 0.97067 |
| | RED-(DnCNN-FFDNet) | 0.93882 | 0.93809 | 0.95902 | 0.92212 | 0.93527 | 0.96131 |

Table 3. Runtime of five algorithms on six images

表 3. 五种算法在六幅图像上的运行时间

| | | Cameraman | Pepper | Monarch | Parrot | Lena | Barbara |
|--------------|--------------------|-----------|----------|----------|----------|----------|----------|
| $\alpha = 2$ | HIO | 8.0156 | 9.7258 | 11.1943 | 10.5683 | 10.7991 | 10.2880 |
| | prDeep | 11.4701 | 9.7399 | 12.3588 | 10.9924 | 11.7673 | 10.7682 |
| | RED-FFDNet | 30.0683 | 30.3592 | 27.8742 | 31.4586 | 29.1933 | 29.1252 |
| | RED-(DnCNN-BM3D) | 112.2271 | 101.4158 | 103.5018 | 103.4596 | 104.7078 | 91.4997 |
| | RED-(DnCNN-FFDNet) | 17.1775 | 15.6832 | 16.3206 | 16.8662 | 16.4177 | 16.7262 |
| $\alpha = 3$ | HIO | 7.7552 | 11.0985 | 11.3833 | 11.2330 | 10.5186 | 10.2153 |
| | prDeep | 11.2175 | 10.9837 | 11.4953 | 11.3417 | 12.7648 | 11.3110 |
| | RED-FFDNet | 31.5895 | 30.8179 | 28.5880 | 31.9473 | 29.7577 | 29.0032 |
| | RED-(DnCNN-BM3D) | 127.4946 | 122.2663 | 121.8104 | 119.6482 | 118.3332 | 111.6253 |
| | RED-(DnCNN-FFDNet) | 22.9563 | 16.3123 | 15.6508 | 16.7565 | 16.4121 | 17.5576 |

图 1 展示了 Set12 数据集中的七幅图像在时的不同算法的视觉比较; 图 2 展示了 Set12 数据集中的七幅图像在时的不同算法的视觉比较。图 1 和图 2 中的每列从左到右分别表示原图像、HIO 算法恢复图、(RED-FFDNet)恢复图、(RED-(DnCNN-BM3D))恢复图和(RED-(DnCNN-FFDNet))恢复图。从图像的视觉比较中我们发现 HIO 算法恢复的图像中还存在大量的噪声, 而(RED-FFDNet)算法恢复的图像过于光滑, 导致很多细节部分缺失了。(RED-(DnCNN-BM3D))和(RED-(DnCNN-FFDNet))恢复的图像比较清晰。表 4 记录了图像 House 在五种不同噪声水平下的相位恢复结果。从表 4 中我们发现(RED-(DnCNN-BM3D))

在时的 PSNR 比 prDeep 稍低, 但是在其他四种噪声水平情况下(RED-(DnCNN-BM3D))的 PSNR 都比 prDeep 高。(RED-(DnCNN-FFDNet))也是如此。

表 5 记录了 Set12 数据集中七幅图像的 PSNR、MSSIM 和运行时间的均值比较, 从表 5 中可以看到在时(RED-(DnCNN-FFDNet))的 PSNR 和 MSSIM 都最高, 在时(RED-(DnCNN-FFDNet))的 PSNR 和 MSSIM 也都是最高的。在时(RED-(DnCNN-BM3D))的 PSNR 比 prDeep 稍低, 但是(RED-(DnCNN-BM3D))的 MSSIM 比 prDeep 高。在时(RED-(DnCNN-BM3D))的 PSNR 和 MSSIM 都比 prDeep 高。表 6 记录了 Set12 数据集中七幅图像的 PSNR、MSSIM 和运行时间的方差比较, 从表 6 中可以看到在时(RED-(DnCNN-FFDNet))的 PSNR 的方差最小, 在时(RED-(DnCNN-BM3D))的 PSNR 的方差最小。在和时(RED-(DnCNN-BM3D))的 MSSIM 的方差都是最小的。



Figure 1. Phase retrieval comparison of seven images in $\alpha = 2$

图 1. 七幅图像在 $\alpha = 2$ 时的相位恢复比较

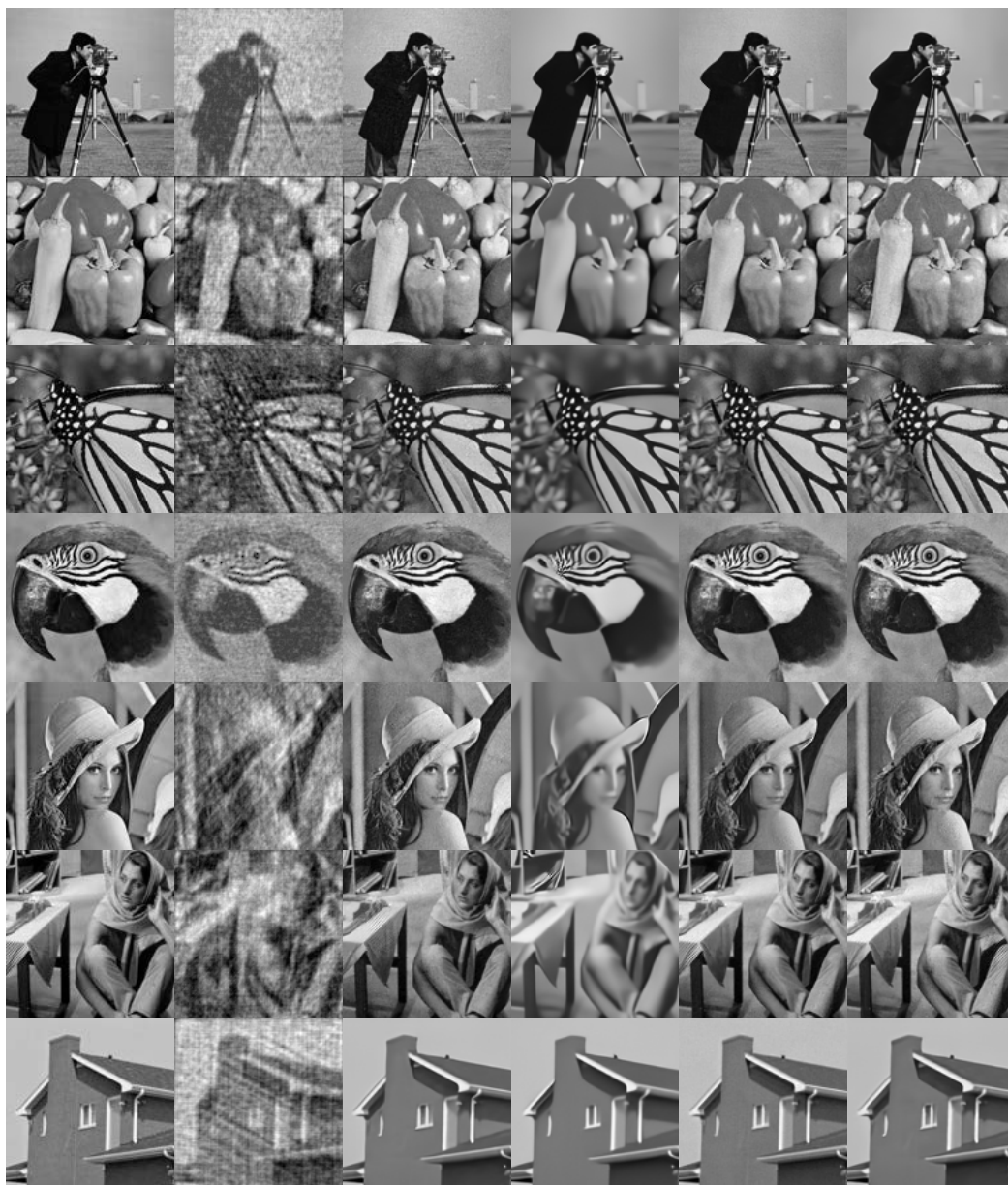


Figure 2. Phase retrieval comparison of seven images in $\alpha = 3$

图 2. 七幅图像在 $\alpha = 3$ 时的相位恢复比较

Table 4. Phase retrieval comparison of image House under different noise levels

表 4. 图像 house 在不同噪声水平下的相位恢复比较

| | | HIO | prDeep | RED-FFDNet | RED-(DnCNN-BM3D) | RED-(DnCNN-FFDNet) |
|--------------|-------|---------|---------|------------|------------------|--------------------|
| $\alpha = 2$ | PSNR | 22.9945 | 39.3964 | 34.0822 | 39.1279 | 38.6843 |
| | MSSIM | 0.47822 | 0.95940 | 0.94089 | 0.96623 | 0.94647 |
| | TIME | 10.5491 | 11.6869 | 31.1347 | 118.4614 | 17.5942 |
| $\alpha = 3$ | PSNR | 21.2353 | 36.4063 | 33.6656 | 36.8918 | 36.5227 |
| | MSSIM | 0.40679 | 0.94945 | 0.93236 | 0.95421 | 0.95158 |

续表

| | | | | | | |
|--------------|-------|---------|---------|---------|----------|---------|
| | TIME | 10.7519 | 11.8520 | 32.4119 | 138.0287 | 21.4820 |
| | PSNR | 19.5553 | 35.9433 | 32.2275 | 36.2154 | 36.0748 |
| $\alpha = 4$ | MSSIM | 0.33439 | 0.94747 | 0.92438 | 0.94726 | 0.94833 |
| | TIME | 10.6222 | 11.7912 | 31.0767 | 153.7877 | 22.1241 |
| | PSNR | 18.3532 | 29.8911 | 31.8003 | 31.6365 | 35.2778 |
| $\alpha = 5$ | MSSIM | 0.26867 | 0.75148 | 0.92042 | 0.92033 | 0.94309 |
| | TIME | 10.2340 | 10.9556 | 30.6033 | 196.4667 | 22.4845 |
| | PSNR | 17.4605 | 28.8539 | 31.3262 | 31.5924 | 33.0065 |
| $\alpha = 6$ | MSSIM | 0.22961 | 0.71100 | 0.91402 | 0.91831 | 0.93026 |
| | TIME | 7.9403 | 11.2893 | 32.3539 | 153.6292 | 22.3241 |

Table 5. Comparison of the average values of PSNR, MSSIM and runtime of seven images
表 5. 七幅图像的 PSNR、MSSIM 和运行时间的均值比较

| | $\alpha = 2$ | | | $\alpha = 3$ | | |
|--------------------|--------------|---------|----------|--------------|---------|----------|
| | PSNR | MSSIM | TIME | PSNR | MSSIM | TIME |
| HIO | 22.0462 | 0.55415 | 10.1629 | 20.4197 | 0.54220 | 10.4223 |
| prDeep | 37.1750 | 0.96545 | 11.2548 | 34.2028 | 0.93328 | 11.5666 |
| RED-FFDNet | 30.9300 | 0.91942 | 29.8876 | 30.0733 | 0.90649 | 30.5879 |
| RED-(DnCNN-BM3D) | 37.1294 | 0.97202 | 105.0390 | 34.9621 | 0.95692 | 122.7438 |
| RED-(DnCNN-FFDNet) | 37.3965 | 0.96550 | 16.6837 | 34.5068 | 0.94374 | 18.1611 |

Table 6. Comparison of the variance of PSNR, MSSIM and runtime of seven images
表 6. 七幅图像的 PSNR、MSSIM 和运行时间的方差比较

| | $\alpha = 2$ | | | $\alpha = 3$ | | |
|--------------------|--------------|---------|---------|--------------|---------|---------|
| | PSNR | MSSIM | TIME | PSNR | MSSIM | TIME |
| HIO | 2.0882 | 0.00390 | 0.9428 | 0.9458 | 0.02759 | 1.3303 |
| prDeep | 1.3624 | 0.00007 | 0.6164 | 1.1336 | 0.00044 | 0.3000 |
| RED-FFDNet | 3.0767 | 0.00032 | 1.3422 | 3.3689 | 0.00044 | 1.9187 |
| RED-(DnCNN-BM3D) | 1.0756 | 0.00004 | 61.8910 | 0.8769 | 0.00010 | 58.5650 |
| RED-(DnCNN-FFDNet) | 0.9370 | 0.00013 | 0.3302 | 0.8942 | 0.00017 | 7.0197 |



Figure 3. Phase retrieval comparison of image House in $\alpha = 4$
图 3. 图像 House 在 $\alpha = 4$ 时的相位恢复比较



Figure 4. Phase retrieval comparison of image House in $\alpha = 5$

图 4. 图像 House 在 $\alpha = 5$ 时的相位恢复比较



Figure 5. Phase retrieval comparison of image House in $\alpha = 6$

图 5. 图像 House 在 $\alpha = 6$ 时的相位恢复比较

图 3、图 4 和图 5 分别展示了图像 House 在其他三种泊松噪声水平下的视觉比较, 即 $\alpha = 4$ 、 $\alpha = 5$ 和 $\alpha = 6$ 。我们发现 HIO 算法的恢复效果最差, 而 (RED-FFDNet) 恢复的图像过于光滑。在高噪声水平如 $\alpha = 5$ 和 $\alpha = 6$ 时, 我们发现 prDeep 恢复的图像还存在大量的噪声, 而在使用了多尺度的去噪正则项即 (RED-(DnCNN-BM3D)) 之后在高噪声水平时恢复出的图像很清晰, 在结合 DnCNN 和 FFDNet 进行分情况选择去噪即 (RED-(DnCNN-FFDNet)) 时恢复出的图像也很清晰。这表明了我们的方法的进步之处。

5. 总结和展望

在本文中, 我们基于去噪正则化框架和不同的去噪器求解带有噪声的相位恢复问题。将成熟多样去噪算法作为正则项解决带有噪声的相位恢复问题展现出了优秀的表现, 基于神经网络的去噪器的强大去噪能力体现了卷积神经网络的强大性能。我们使用多尺度的去噪正则化求解带有噪声的相位恢复问题, 它基于去噪正则化框架, 我们通过数值实验表明了这种改进的优点。我们在相位恢复的过程中分情况使用 DnCNN 和 FFDNet 这两种去噪器。最后我们通过数值实验比较了我们的改进策略在视觉效果和量化数值上都有更加优异的表现。

我们的想法不只局限于相位恢复问题, 我们同样可以把本文的思想应用于其他的逆问题中, 比如图像去模糊, 超分辨率等。而且我们在未来的工作中可以结合其他形式的去噪正则项和其他的去噪器, 并会考虑复值信号的相位恢复问题。

参考文献

- [1] Shechtman, Y., Eldar, Y.C., Cohen, O., *et al.* (2015) Phase Retrieval with Application to Optical Imaging: A Contemporary Overview. *IEEE Signal Processing Magazine*, **32**, 87-109. <https://doi.org/10.1109/MSP.2014.2352673>
- [2] Carlsson, M. and Gerosa, D. (2019) On Phase Retrieval via Matrix Completion and the Estimation of Low Rank PSD Matrices. *Inverse Problems*, **36**, Article ID: 015006. <https://doi.org/10.1088/1361-6420/ab4e6d>
- [3] Zhuang, Z. (2019) On Stability of Generalized Phase Retrieval and Generalized Affine Phase Retrieval. *Journal of Inequalities and Applications*, **2019**, Article No. 14. <https://doi.org/10.1186/s13660-019-1968-z>
- [4] Qin, S., Zhang, Y.B., Wang, H.Q., *et al.* (2020) Simple Accurate Model-Based Phase Diversity Phase Retrieval Algorithm for Wavefront Sensing in High-Resolution Optical Imaging Systems. *IET Image Processing*, **14**, 4513-4519. <https://doi.org/10.1049/iet-ipr.2020.1075>
- [5] Jiang, Z.L., *et al.* (2020) Phase Retrieval of On-Axis Digital Holography with Modified Coherent Diffraction Imaging. *IEEE Photonics Journal*, **12**, Article ID: 6901712. <https://doi.org/10.1109/JPHOT.2020.3035805>
- [6] Palamodov V. (2020) An Analytic Method of Phase Retrieval for X-Ray Phase Contrast Imaging. *Journal of Fourier*

- Analysis and Applications*, **26**, Article No. 79. <https://doi.org/10.1007/s00041-020-09787-x>
- [7] Luo, Q., Lin, S.J. and Wang, H.X. (2021) Robust Phase Retrieval via Median-Truncated Smoothed Amplitude Flow. *Inverse Problems in Science and Engineering*, **29**, 3024-3040. <https://doi.org/10.1080/17415977.2021.1966426>
- [8] Alaifari, R. and Wellershoff, M. (2021) Phase Retrieval from Sampled Gabor Transform Magnitudes: Counterexamples. *Journal of Fourier Analysis and Applications*, **28**, Article No. 9. <https://doi.org/10.1007/s00041-021-09901-7>
- [9] Song, J.W., Liu, K., Sowmya, A., et al. (2022) Super-Resolution Phase Retrieval Network for Single-Pattern Structured Light 3D Imaging. *IEEE Transactions on Image Processing*, **32**, 537-549.
- [10] Gerchberg, R.W. (1972) A Practical Algorithm for the Determination of Plane from Image and Diffraction Pictures. *Optik*, **35**, 237-246.
- [11] Fienup, J.R. (1984) Reconstruction and Synthesis Applications of an Iterative Algorithm. *Transformations in Optical Signal Processing*, **373**, 147-160. <https://doi.org/10.1117/12.934545>
- [12] Cui, J.Z., Sha, H.Z., Yang, W.F. and Yu, R. (2023) Antiferromagnetic Imaging via Ptychographic Phase Retrieval. *Science Bulletin*, **69**, 466-472.
- [13] Shao, Z.H., Li, L.D., et al. (2023) Cancelable Face Recognition Using Phase Retrieval and Complex Principal Component Analysis Network. *Machine Vision and Applications*, **35**, Article No. 12, <https://doi.org/10.1007/s00138-023-01496-x>
- [14] Moravec, M.L., Romberg, J.K. and Baraniuk, R.G. (2007) Compressive Phase Retrieval. *Wavelets XII. SPIE*, Vol. 6701, 712-722. <https://doi.org/10.1117/12.736360>
- [15] Schniter, P. and Rangan, S. (2014) Compressive Phase Retrieval via Generalized Approximate Message Passing. *IEEE Transactions on Signal Processing*, **63**, 1043-1055. <https://doi.org/10.1109/TSP.2014.2386294>
- [16] Venkatakrisnan, S.V., Bouman, C.A. and Wohlberg, B. (2013) Plug-and-Play Priors for Model Based Reconstruction. 2013 *IEEE Global Conference on Signal and Information Processing*, Austin, 3-5 December 2013, 945-948. <https://doi.org/10.1109/GlobalSIP.2013.6737048>
- [17] Romano, Y., Elad, M. and Milanfar, P. (2017) The Little Engine That Could: Regularization by Denoising (RED). *SIAM Journal on Imaging Sciences*, **10**, 1804-1844. <https://doi.org/10.1137/16M1102884>
- [18] Metzler, C., Schniter, P., Veeraraghavan, A., et al. (2018) prDeep: Robust Phase Retrieval with a Flexible Deep Network. *International Conference on Machine Learning*, Stockholm, 10-15 July 2018, 3501-3510.
- [19] Zuo, C., Qian, J., Feng, S., et al. (2022) Deep Learning in Optical Metrology: A Review. *Light: Science & Applications*, **11**, Article No. 39. <https://doi.org/10.1038/s41377-022-00714-x>
- [20] Tayal, K., Lai, C.H., Manekar, R., et al. (2020) Unlocking Inverse Problems Using Deep Learning: Breaking Symmetries in Phase Retrieval. *NeurIPS 2020 Workshop on Deep Learning and Inverse Problems*.
- [21] Zhang, K., Zuo, W., Chen, Y., et al. (2017) Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing*, **26**, 3142-3155. <https://doi.org/10.1109/TIP.2017.2662206>
- [22] Zhang, K., Zuo, W. and Zhang, L. (2018) FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising. *IEEE Transactions on Image Processing*, **27**, 4608-4622. <https://doi.org/10.1109/TIP.2018.2839891>
- [23] Goldstein, T., Studer, C. and Baraniuk, R. (2014) A Field Guide to Forward-Backward Splitting with a FASTA Implementation.
- [24] 方莉, 张萍. 经典图像去噪算法研究综述[J]. 工业控制计算机, 2010, 23(11): 73-74. <https://doi.org/10.3969/j.issn.1001-182X.2010.11.034>
- [25] 孙玉宝, 韦志辉, 吴敏, 等. 稀疏性正则化的图像泊松去噪算法[J]. 电子学报, 2011, 39(2): 285-290.
- [26] 关新平, 赵立兴, 唐英干. 图像去噪混合滤波方法[J]. 中国图象图形学报, 2005(3): 332-337. <https://doi.org/10.3969/j.issn.1006-8961.2005.03.013>
- [27] 王科俊, 熊新炎, 任桢. 高效均值滤波算法[J]. 计算机应用研究, 2010, 27(2): 434-438. <https://doi.org/10.3969/j.issn.1001-3695.2010.02.008>
- [28] 赵高长, 张磊, 武风波. 改进的中值滤波算法在图像去噪中的应用[J]. 应用光学, 2011, 32(4): 678-682. <https://doi.org/10.3969/j.issn.1002-2082.2011.04.017>
- [29] 谢杰成, 张大力, 徐文立. 小波图像去噪综述[J]. 中国图象图形学报, 2002(3): 3-11. <https://doi.org/10.3969/j.issn.1006-8961.2002.03.001>
- [30] Bhujle, H.V. and Vadavadagi, B.H. (2019) NLM Based Magnetic Resonance Image Denoising—A Review. *Biomedical Signal Processing and Control*, **47**, 252-261. <https://doi.org/10.1016/j.bspc.2018.08.031>
- [31] Archana, T. and Phool, S. (2023) A Comparative Analysis of Phase Retrieval Algorithms in Asymmetric Double Image Cryptosystem in Gyration Domain. *Optical and Quantum Electronics*, **56**, Article No. 33. <https://doi.org/10.1007/s11082-023-05524-y>

- [32] Tao, S.W., Tian, Z.H., Bai, L., *et al.* (2023) Phase Retrieval for X-Ray Differential Phase Contrast Radiography with Knowledge Transfer Learning from Virtual Differential Absorption Model. *Computers in Biology and Medicine*, **168**, Article ID: 107711.
- [33] Li, Y.F., Wu, G.D., *et al.* (2023) Phase Retrieval from Intensity Difference of Linear Canonical Transform. *Banach Journal of Mathematical Analysis*, **18**, Article No. 3. <https://doi.org/10.1007/s43037-023-00307-9>
- [34] Ye, Q.L., Wang, L.-W. and Lun, D.P.K. (2023) Towards Practical Single-Shot Phase Retrieval with Physics-Driven Deep Neural Network. *Optics Express*, **31**, 35982-35999. <https://doi.org/10.1364/OE.496418>
- [35] Dabov, K., Foi, A., Katkovnik, V., *et al.* (2007) Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing*, **16**, 2080-2095. <https://doi.org/10.1109/TIP.2007.901238>