

附件 1:

第八届湖南省研究生数学建模竞赛承诺书

我们仔细阅读了湖南省高校研究生数学建模竞赛的竞赛规则。

我们完全明白,在竞赛开始后参赛队员不能以任何方式(包括电话、电子邮件、网上咨询等)与队外的任何人(包括指导教师)研究、讨论与赛题有关的问题。

我们知道,抄袭别人的成果是违反竞赛规则的,如果引用别人的成果或其他公开的资料(包括网上查到的资料),必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们完全清楚,在竞赛中必须合法合规地使用文献资料和软件工具,不能有任何侵犯知识产权的行为。否则我们将失去评奖资格,并可能受到严肃处理。

我们郑重承诺,严格遵守竞赛规则,以保证竞赛的公正、公平性。如有违反竞赛规则的行为,我们将受到严肃处理。

我们授权湖南省研究生数学建模竞赛组委会,可将我们的论文以任何形式进行公开展示(包括进行网上公示,在书籍、期刊和其他媒体进行正式或非正式发表等)。

我们参赛选择的题号是(从组委会提供的赛题中选择一项填写):

我们的参赛编号(请填写完整参赛编号): HN2023457

所属学校(请填写完整的全名): 中国人民解放军国防科技大学

参赛队员(打印后签名): 1.

2.

3.

指导教师或指导教师组负责人(打印后签名):

日期:

(请勿改动此页内容和格式。以上内容请仔细核对,如填写错误,论文可能被取消评奖资格。)

用于图像重构的相位恢复与去噪算法研究

摘要

随着人们对宇宙奥秘探索的深入，对宇宙或天体的成像质量要求越来越高。但它们很难直接成像，需要通过射电强度、频谱信息等相关信息间接成像。本文针对间接成像(图像恢复)中的相位缺失和噪声干扰问题，构建数学模型。基于真实域和傅里叶域相互约束和迭代的思想，通过确定图像一致性等指标，以高质量成像为目标建立了相位恢复数学模型，并使用 HIO 算法和 OSS 算法对模型进行求解。

针对问题一、二：根据傅里叶变换后的幅度信息和相位信息可以容易的恢复出图像，但当相位信息缺失且幅度信息中含有噪声时，图像恢复变得非常困难。此时图像恢复的思路为先进行相位恢复，再进行数据去噪，最后根据得到的相位信息和幅度信息恢复图像。对相位恢复问题构建数学模型，并分析模型的合理性和可解性。对相位恢复模型进行求解可以看作最小二乘问题。使用真实域的幅值信息和傅里叶域的幅值信息相互约束，通过迭代求解出最佳的相位信息，即 HIO 算法。根据本题所给的条件，该算法可以收敛，即找到全局最优且解具有唯一性。对于去噪问题，HIO 算法对噪声很敏感。提出两种去噪方法，直接去噪法和算法优化去噪法。单一噪声情况下，数据获取方法和环境已知，易于分析噪声类型时，可以采用直接去噪法，即针对不同的噪声使用不同的滤波进行去噪，此方法对未知噪声和混合噪声效果不佳。算法优化法即是在 HIO 算法的基础上，通过改变迭代中的约束条件，在迭代中加入高斯滤波以平滑信息，提出的 OSS 算法。OSS 算法可以有效地解决噪声敏感和混合噪声问题并避免求解过程中的振荡。在实验中使用与测量结果的一致性来量化恢复误差，OSS 算法的误差均在 0.1 以下。并且使用 OSS 算法恢复出附件 1、2、3 中的图像。

针对问题三：考虑过采样率和先验信息对图像恢复概率的影响，通过自建数据，使用提出的相位恢复模型和 HIO 算法进行实验。对于过采样率问题，将频域采样率扩大一倍，可以恢复原频域采样点处的相位信息。通过实验结果发现，过采样率对图像的恢复有较大影响，直接影响图像的恢复精度。较高的采样率可以更好的恢复信号的相位信息进而恢复图像的更多细节，实验中恢复图像的 PSNR 在 30 左右。但过大的采样率会增加数据处理和存储的问题。而且在设计的每一过采样率下重复十次实验当中发现，过采样率越大，图像的恢复概率越小。在过采样率为 2 时，图像的恢复概率为 0.8，当过采样率达到 9 时，恢复概率为 0。对于先验知识的影响，常用的先验知识有非负性和信号稀疏性等，这些先验知识可以使模型原本非凸的相位恢复问题，成为凸问题，使模型具有全局且唯一解。

最后，我们对提出的模型进行全面的评价：本文的模型贴合实际，能合理解决提出的问题，具有实用性强，算法效率高等特点，该模型在天文成像、全息成像、电子显微镜、射线晶体学和地震信号处理等领域中均可使用。

关键词：相位恢复 傅里叶变换 去噪成像 图像恢复 约束迭代法

目录

摘要.....	I
1 问题综述.....	1
1.1 问题背景.....	1
1.2 问题提出.....	1
1.3 资料条件.....	2
2 模型假设与符号说明.....	2
2.1 模型基本假设.....	2
2.2 符号说明.....	2
3 问题分析与模型建立.....	2
3.1 问题分析.....	2
3.2 模型建立.....	4
3.2.1 相干衍射成像.....	4
3.2.2 相位恢复问题数学模型.....	4
4 模型求解与算法.....	5
4.1 模型解的唯一性讨论.....	5
4.2 相位恢复模型求解与算法.....	5
4.3 含噪声的相位恢复模型求解与算法.....	7
4.3.1 直接去噪法.....	7
4.3.2 OSS 算法.....	9
4.4 模型指标定义.....	10
5 模型应用及分析.....	10
5.1 HIO 算法应用与分析.....	10
5.2 OSS 算法应用与分析.....	12
5.3 图像恢复的影响因素.....	14
5.3.1 过采样率的影响.....	14
5.3.2 先验知识的影响.....	15
6 模型评价与改进.....	16
6.1 模型的优点.....	16
6.2 模型的不足.....	16
6.3 模型的改进.....	16
参考文献.....	17
附 录.....	18
附录 A: 图像恢复结果.....	18
附录 B: 主要程序/关键代码.....	18

1 问题综述

1.1 问题背景

随着科学技术的不断发展,信息的交流和获取已不再受到时空的限制,图像作为信息的重要载体,在不同科学领域起着不可替代的作用.但在光学成像领域,受到光学探测设备的限制和由于光的波长短和速度快的特性使得现有的光测量装置很难直接测量光波的相位分布,因此无法直接得到需要的复振幅,从而丢失了许多可用的成像信息.为了满足像天眼 FAST 等高精尖设备的高质量成像需求,我们需要寻找有效的解决方案,以实现用仅有的幅值信息进行图像恢复.

在光学中,高频光波的信息可以通过光学设备方便地记录下来,而相位信息很难获取,这也是目前很多光学探测设备都面临的问题.解决这一问题的有效办法是相位恢复,即从受约束信号的傅里叶变换幅值恢复信号,或者等效地根据其自相关恢复信号,该方法在衍射成像、天文成像、全息成像、电子显微镜、射线晶体学和地震信号处理等领域中具有十分广泛的应用.此外,观测到的幅值往往在信号采集过程中受到严重噪声的干扰,这使得图像恢复问题更加复杂.为满足高精尖设备的高质量成像需求,解决从高噪声的图像或信号的傅立叶变换幅值中恢复出原始图像问题.

目前,已经有很多学者对图像恢复面临的相位缺失问题进行了研究,也提出了很多算法来解决此类问题.从最初 Sayre [1] 的猜想,在一些简单的情形下可以根据傅里叶强度测量的某些精细特征(峰值信息)将散射波的相位信息恢复出来;到以 Gerchberg-Saxton 算法(GS 算法)为代表的一系列交替投影迭代算法的提出;再到现在基于深度学习的方法,这些方法对于根据傅里叶变换信号的幅值重建图像是有效的,但大多数忽略了噪声的影响和模型解的唯一性问题.

综上,我们需要针对含噪声的强度或频谱信息建立更合理的数学模型,并考虑相位恢复问题的非凸性和解的唯一性,对模型进行求解,并最终实现对图像的高质量恢复.

1.2 问题提出

在很多实际问题中,很难直接获取图像信息,往往只能测量获得图像的强度、频谱等相关信息.需要基于频谱信息来获取目标的图像信息,最终得到目标的图像.而且我们测得的强度或频谱信息会含有一定噪声的,也就是我们需要基于含噪声的强度或频谱信息,来恢复图像.

题目已经给出含噪声的频谱信息的生成方法以及噪声的种类,并根据数据可以发现频谱信息中确实缺少相位信息.在以下 2 个预设条件下:

- 原始图像大小为 256×256 对其频谱进行过采样,然后进行离散傅里叶变换,可得到相应的 512×512 频谱信息.
- 图像的频谱为含噪声频谱且噪声类型有高斯噪声,泊松噪声,脉冲噪声和瑞利噪声等噪声.附件 1 数据含有一种类型噪声,附件 2 和 3 数据含有多种类型混合噪声.

从数据含噪情况、数学模型构建、图像恢复影响因素角度考虑,解决以下 3 个问题:

- (1) 问题 1: 根据附件 1 提供的数据(含一种噪声),构建基于含噪声频谱信息的相位恢复数学模型,求解模型,给出相应的去噪算法,并恢复原图.
- (2) 问题 2: 根据附件 2 和 3 数据(混合噪声),建立相应的数学模型,求解模型,给出相应的去噪算法,并恢复原图像.

- (3) 问题 3: 在基于频谱信息来恢复图像的过程中, 分析过采样率和相关先验信息对图像的恢复概率的影响. 基于在不同的过采样率以及相关先验信息下, 并结合问题 1 和问题 2 中的模型, 计算图像的恢复概率.

1.3 资料条件

- 附件 1.csv 提供了对于一幅 256×256 的图像, 对其频谱进行过采样, 然后进行离散傅里叶变换, 得到的 512×512 的含一种噪声的频谱信息.
- 附件 2.csv 和附件 3.csv 提供了对两幅 256×256 的图像的频谱进行过采样, 得到各自相应的 512×512 含混合噪声的频谱信息.

2 模型假设与符号说明

2.1 模型基本假设

- (1) 假设人为过采样后生成的含噪频谱信息与实际光学器件测得数相符.
- (2) 假设任意线性算子可以替代离散傅里叶变换, 且误差不计.
- (3) 假设原始图像为非负的, 有界的实值.
- (4) 假设不能被探测器直接记录低的空间频率信息对图像回复的影响可以忽略不计.
- (5) 假设所有数据是在完全相干光条件下获得的, 不考虑部分相干光的情况.

2.2 符号说明

本文定义了如下 10 个使用次数较多的符号, 其余符号在使用时注明.

表 1 符号说明

符号	含义
x	待恢复信号
X	x 的 DFT 变换
a_k	采样向量
y	测量值
$ x[n] $	量的实空间幅值大小
$ X[k] $	傅立叶空间幅值大小
ε	误差阈值
$W[k]$	高斯函数
α	高斯滤波器的大小
R_F	相位恢复质量评价指标

3 问题分析与模型建立

3.1 问题分析

题目以射电望远镜探测射电强度来获取天体图像为背景, 介绍了我国天眼 FAST 和光学衍射间接成像的过程. 在无法直接获得图像信息时, 通过测的图像的频谱或强度信息进行图像恢复.

问题一要求我们根据给定的含噪频谱数据 (含一种噪声), 建立相应的数学模型, 给出相应的去噪算法, 并恢复原图像;

问题二要求我们在问题一的基础上进一步考虑对含有混合噪声的频谱数据进行去噪和图像恢复;

问题三则是分析过采样率、先验信息对图像恢复概率的影响.

根据附件 1、2 和 3 所提供的频谱数据可以发现, 数据中仅含有幅值信息缺少相位信息, 仅使用幅值信很难进行图像的恢复. 再结合本题背景, 在波动光学中, 单色相干光场可以使用二维复振幅函数来描述, 包含幅值和相位两个组成部分. 可见光的高时间频率和现有成像探测器的有限带宽使得光学相位信息无法被直接记录, 通常只能探测到光场的强度 (幅值的平方) 信息, 从而丢失了许多可用的成像信息. 由此我们需要使用一定的方法, 借助测得的振幅信息来确定复振幅的相位, 即由测得的强度数据来恢复相位分布从而实现图像的恢复.

尽管图像传感器无法直接记录光场的相位信息, 但是光场具备的某些特性使得相位分布可以借助于算法恢复出来, 即光波远场的信息对应于近场信息的傅里叶变换. 相位恢复算法提供了在已知远场强度测量值以及某些先验知识的前提下重构相位的可能性. 此外, 在信号采集过程中, 观测到的幅值经常会受到较大噪声的影响, 这使得图像恢复问题变得更加复杂. 本题构建的数学模型主要解决两个问题: 相位恢复和去噪成像. 构建数学模型后, 对模型进行评价, 并分析过采样率和先验信息对图像恢复概率的影响. 图 1 为本题建模求解的整体结构框图.

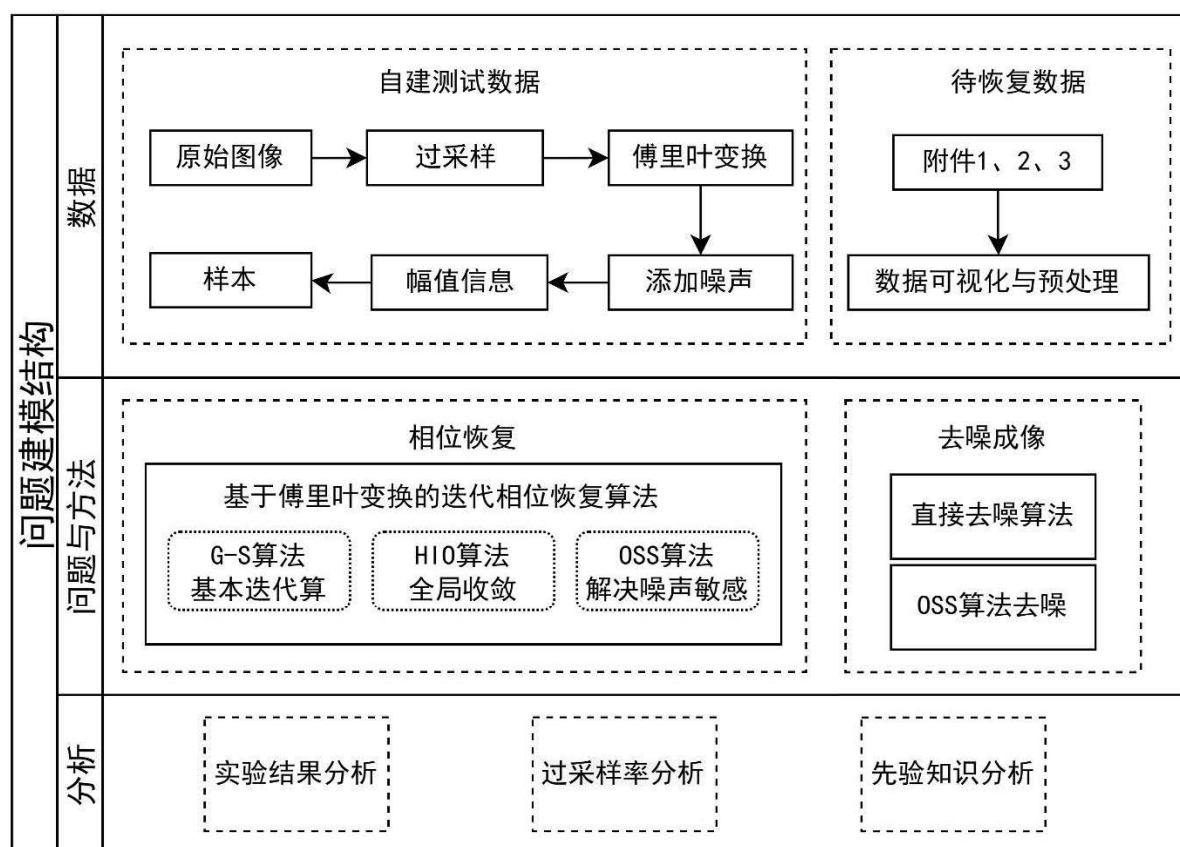


图 1 问题建模与求解结构框图

3.2 模型建立

3.2.1 相干衍射成像

在基本的前向散射相干衍射成像中 Coherent Diffractive Imaging (CDI) [8] , 可以用准单色相干波照射物体, 测量其衍射光强. 当物体较小且测量强度较远时, 测量的强度与物平面上波的傅里叶变换的大小成正比, 并且具有适当的空间尺度.

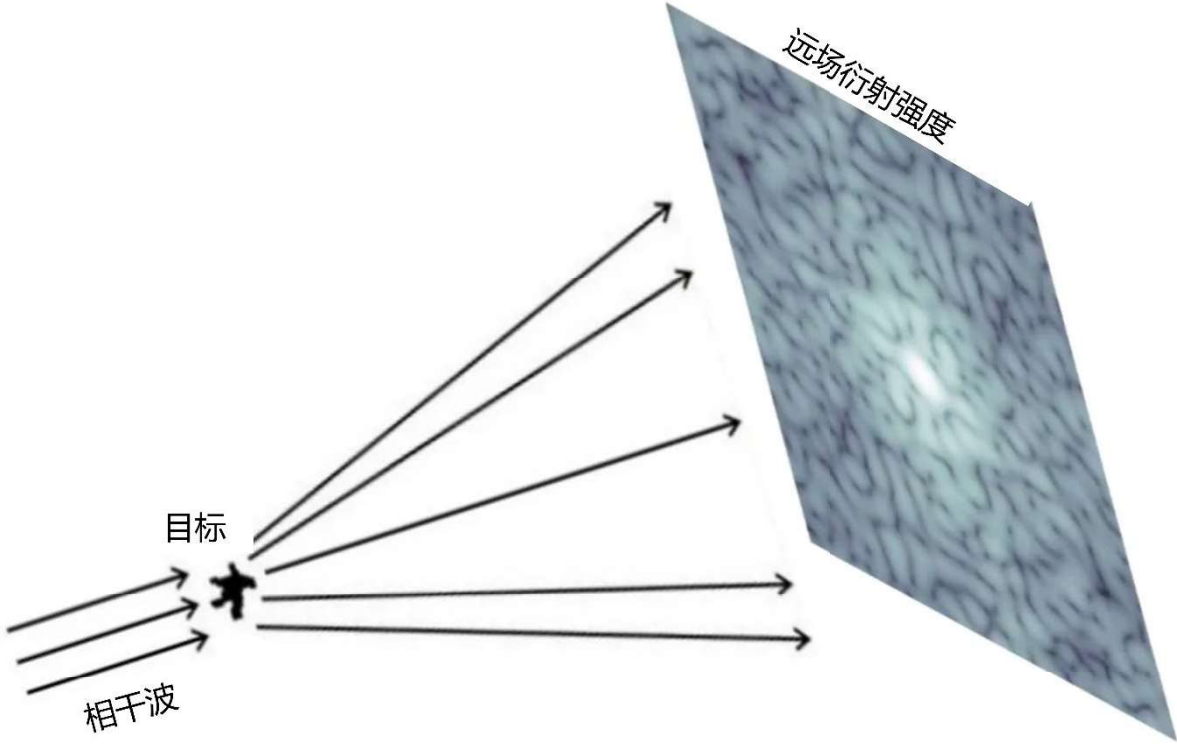


图 2 前向散射 CDI 示意图: 相干波经过目标 (所求信息) 发生衍射, 并产生与目标傅里叶变换幅值相对应的远场强度图样.

在光学中, 当菲涅耳数很小时 ($N_F = \frac{a^2}{\lambda d} \ll 1$, 其中 a 是在物平面内的目标限制半径, d 是物体与测量强度平面之间的距离, λ 是光的波长), 测得的光强 I_{out} 与物平面 E_{in} 上的波之间的关系是:

$$I_{out}(x, y) \propto \left| \hat{E}_{in}\left(\frac{x}{\lambda d}, \frac{y}{\lambda d}\right) \right|^2 \quad (3-1)$$

其中, $\hat{E}_{in} = \mathcal{F}\{E_{in}\}$, $\mathcal{F}$ 表示傅里叶变换.

一旦测量了远场强度, 下一步就是从 I_{out} 中恢复 E_{in} (相当于恢复目标), 在这个过程中需要解决相位恢复问题.

3.2.2 相位恢复问题数学模型

相位恢复问题的目标是从信号强度或幅度测量值中恢复原始信号及其相位. 使用一维离散向量 $\mathbf{x} \in \mathbb{C}^n$ 来表示待恢复信号 (可以直接地向更高维度的信号领域拓展). 例如, 在 CDI 中, 这对应于目标的透射函数. $\mathbf{x}$ 通常是复数, 这对应于从物体上的不同点发出

的电磁场不仅具有大小, 而且具有相位. 测量强度的远射场于入射场之间满足傅里叶函数关系. x 的一维离散傅里叶变换 (DFT) 为:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{kn}{N}}, \quad k = 0, 1, \dots, N-1 \quad (3-2)$$

过采样率的 DFT 指的是在 $x \in \mathbb{C}^n$ 且 $M > N$, 对 M 个点的离散傅里叶变化, 其表示为:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{kn}{M}}, \quad k = 0, 1, \dots, M-1 \quad (3-3)$$

通过简单地逆 DFT 变换就可以从 X 的测量中恢复 x . 将式(3-3)改写为指数形式 $X[k] = |X[k]| \cdot e^{j\phi[k]}$, 则傅里叶相位恢复问题是在只测量到 X 的幅值时恢复 x , 即在给定 $|X[k]|$ 时恢复 $x[n]$. 由于 DFT 算子是双射算子, 该过程相当于恢复 $X[k]$ 的相位, 因此该问题称为相位恢复问题 (phase retrieval).

用 $\hat{x}$ 表示用 $N-1$ 个零填充 x 后的向量, 然后将 $\hat{x}$ 的自相关序列定义为:

$$g[m] = \sum_{i=\max\{1, m+1\}}^N \hat{x}_i \bar{\hat{x}}_{i-m}, \quad m = -(N-1), \dots, N-1 \quad (3-4)$$

将 $g[m]$ 的 DFT 表示为 $G[k]$, 其满足 $G[k] = |X[k]|^2$. 因此, 从其傅立叶幅度恢复信号的问题等效于从其过采样的自相关序列恢复信号的问题. 在实际中也存在很多是连续变量的情况, 只需要改为连续的傅里叶变换即可, 由于本体处理的数据为数字化的, 因此只讨论离散化的情况.

傅里叶相位恢复问题是更一般的相位恢复问题的一个特例. 通过已知的采样向量 $a_k \in \mathbb{C}^n$ 得到信号的测量值 $\langle a_k, x \rangle$. 信号的测量值不包含相位, 因此用模值的平方来表示强度测量值, 则一般相位恢复问题可以描述为:

$$y_k = |\langle a_k, x \rangle|^2, \quad i = 1, 2, \dots, M \quad (3-5)$$

在离散一维傅里叶相位恢复中, 测量向量为 $a_k[n] = e^{-j2\pi \frac{kn}{M}}$.

4 模型求解与算法

4.1 模型解的唯一性讨论

相位恢复模型的唯一性问题是寻找在什么条件下相位问题的解是唯一的. 从信号的傅立叶幅度恢复信号, 可能不会产生唯一的解. [3] 提到在一维问题中解没有唯一性, 存在多个具有相同傅立叶幅值的一维信号. 即使信号的支持度被限定在一个已知的范围内, 唯一性也不存在. 任何一对具有相同自相关函数的一维信号都产生相同的傅里叶幅度, 因为这两个信号通过傅里叶变换连接起来. 对于高维 (2D 及以上), [4] 已经证明, 除全部为 0 信号外, 对于一组 $d \geq 2$ 支持度为 $\mathbf{N} = [N_1, \dots, N_d]$ 的数据 $x(n_1, \dots, n_d)$, 当 $n_k < 0$ 或 $n_k \geq N_k$, $k = 1, \dots, d$ 时, 它的连续傅立叶变换的大小的解是唯一的 (不考虑歧义情况). 过采样率和先验信息的具体分析见 5.3.

4.2 相位恢复模型求解与算法

要求解的一般相位恢复问题可以看作最小二乘问题:

$$\min_x \sum_{k=1}^M (y_k - |\langle a_k, x \rangle|)^2 \quad (4-1)$$

其中 y 是测量值, a_k 与在式(3-5)中定义相同. 可以发现, 这是一个非凸问题, [7] 中详细解释了凸与非凸问题并证明该模型要求解的问题是非凸的. 即使该问题存在唯一解, 也是很难求解出来的.

迭代法是求解该模型的有效方法之一, 本文使用基于傅里叶变换的迭代相位恢复算法进行求解. 最早由 Gerchberg 和 Saxton 提出 GS 算法, 用于处理从两个不同平面, 实(成像)平面和傅里叶(衍射)平面的幅度测量中恢复复杂图像的密切相关问题. GS 算法迭代的在实平面和傅立叶平面地施加约束, 约束为测量的实空间幅值大小 $|x[n]|$ 和傅立叶空间幅值大小 $|X[k]|$. 恢复误差定义为 $E_i = \|Z_i[k] - |X[k]|\|^2$, E_i 是随 i 单调递减的. 但此算法不能保证求解到真正的解, 因为该算法可能收敛到一个局部极小值.

解决局部最小问题, 除了与测量的傅立叶幅值一致之外, 实平面幅值约束可以用其他类型的约束来代替. 例如实空间约束为非负性, 已知一个信号对于所有 $i > N_0$ 有 $x[i] = 0$, 其中 N_0 是已知(或近似值). 也就是在 GS 算法的基础上, 通过对实空间估计应用校正来代替施加实空间幅度约束, 也就是混合输入输出算法 (hybrid-input output, HIO) [5], 模型求解的具体算法见 **Algorithm 1**.

Algorithm 1 hybrid-input output (HIO)

Input: $|x[n]|$, $|X[k]|$, ε

$|x[n]|$ -真是空间幅值

$|X[k]|$ -傅里叶空间幅值

ε -误差阈值

Output: $z[n]$ -同时满足两个幅值约束的向量

即 $|z[n]| = |x[n]|$, 并且 $|Z[k]| = |X[k]|$, 其中 $Z[k]$

由 $z[n]$ 的 DFT 变换得到

Initialization. 选择初始值 $z_0[n] = |x[n]| \exp(\phi[n])$ 比如用随机的 $\phi[n]$.

General Step ($i = 1, 2, \dots$):

1) 对 $z_i[n]$ 傅里叶变换得到 $Z_i[k]$

2) 保持当前傅立叶相位, 但傅立叶幅值满足约束:

$$Z_i'[k] = |X[k]| \cdot \frac{Z_i[k]}{|Z_i[k]|}$$

3) 对 $Z_i'[k]$ 傅里叶逆变换得到 $z_i'[n]$

4) 通过对实空间图像估计应用校正获得 $z_{i+1}[n]$:

$$z_{i+1}[n] = \begin{cases} z_i'[n], & n \notin \gamma \\ z_i[n] - \beta z_i'[n], & n \in \gamma \end{cases}$$

β 为一个较小的参数, γ 是 $z_i'[n]$ 违反实空间约束的指集.

5) 返回第一步

Until $E_i = \sum_k \|Z_i[k] - |X[k]|\|^2 \leq \varepsilon$

没有证据证明 HIO 算法是收敛的, 但它对先验信息的准确性是已知的 (例如, 在复杂信号的情况下, 需要严格知道真实空间的支持度). 在实践中, 简单的基于 HIO 的方法常用于如 CDI 等光学相位恢复应用. HIO 算法的一个重要特点是它的经验能力, 对于无噪声的过采样衍射图, 可以避免局部极小值并收敛到全局极小值的. 然而, 当衍射强度中存在高噪声时, HIO 的性能将会下降, 算法可能会停滞, 无法找到全局最小解和在迭代过程中出现振荡.

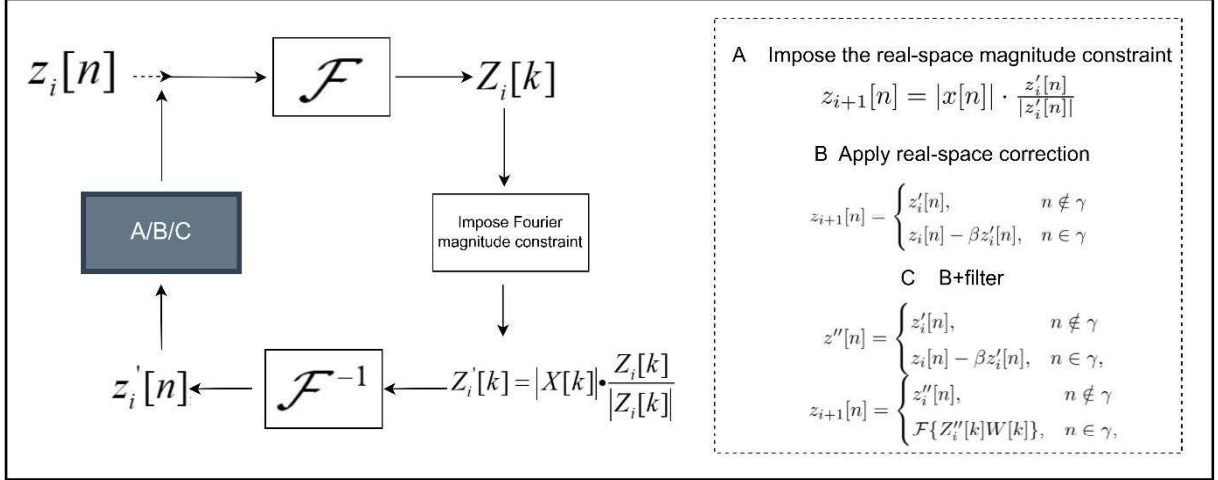


图 3 算法框图, (A) G-S 算法 (B) HIO 算法(C) OSS 算法, 这三种算法不同之处为灰色框处

4.3 含噪声的相位恢复模型求解与算法

4.3.1 直接去噪法

直接去噪法适用于, 已知数据获取的条件和环境, 并通过对数据的观察分析, 可以很容易发现数据分布的情况, 且该方法多用于数据的预处理. 在实验中, 使用了几种滤波器对本题数据进行处理, 再进行相位恢复. 由于无法判断噪声的类型和产生原因, 其效果一般.

以下介绍典型的噪声特点和分布, 并在图四中展示了各类噪声的效果:

1) 高斯噪声: 它是服从高斯分布 (正态分布) 的一类噪声, 其产生的主要原因包括相机在拍摄时视场较暗且亮度不均匀、相机长时间工作使得温度过高, 以及电路元器件自身噪声对其影响也是造成高斯噪声的重要原因之一. 概率密度函数 (PDF) 如下:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \quad (4-2)$$

2) 泊松噪声: 噪声分布符合泊松分布, 又称散粒噪声. 泊松噪声在图像中表现为随机的亮度变化或像素值的波动. 它会导致图像中的细节部分出现随机的亮度增加或减少, 使图像看起来更加嘈杂和不稳定. 泊松噪声通常在低光条件下更加明显, 其 PDF 为:

$$p(z=k) = \frac{e^{-\lambda} \lambda^k}{k!} \quad (4-3)$$

3) 椒盐噪声: 椒盐噪声又称为脉冲噪声, 是在图像上随机出现黑色白色的像素. 在图像处理中, 椒盐噪声是由于图像采集或传输过程中的干扰或错误引起的. 它可能是由于传感器损坏、信号干扰、传输错误等原因造成的. 椒盐噪声会导致图像中出现明显的黑白点, 使图像的质量下降, 影响图像的可视化和分析. 其 PDF 如下, 其中 V 是区间 $0 < V < 2k-1$ 的任意整数.

$$p(z) = \begin{cases} P_s, & z = 2^k - 1 \\ P_p, & z = 0 \\ 1 - (P_s + P_p), & z = V \end{cases} \quad (4-4)$$

4) 瑞利噪声：瑞利噪声的出现来源于光的散射、电子传感器的噪声以及其他信号传输和采集过程中的干扰。在光的散射中，瑞利噪声是由于光波在经过不同介质或障碍物散射时引起幅度的随机变化。在电子传感器中，瑞利噪声是由于信号电荷的随机生成和放大引起的，其 PDF 为：

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b}, & z \geq a \\ 0, & z < a \end{cases} \quad (4-5)$$

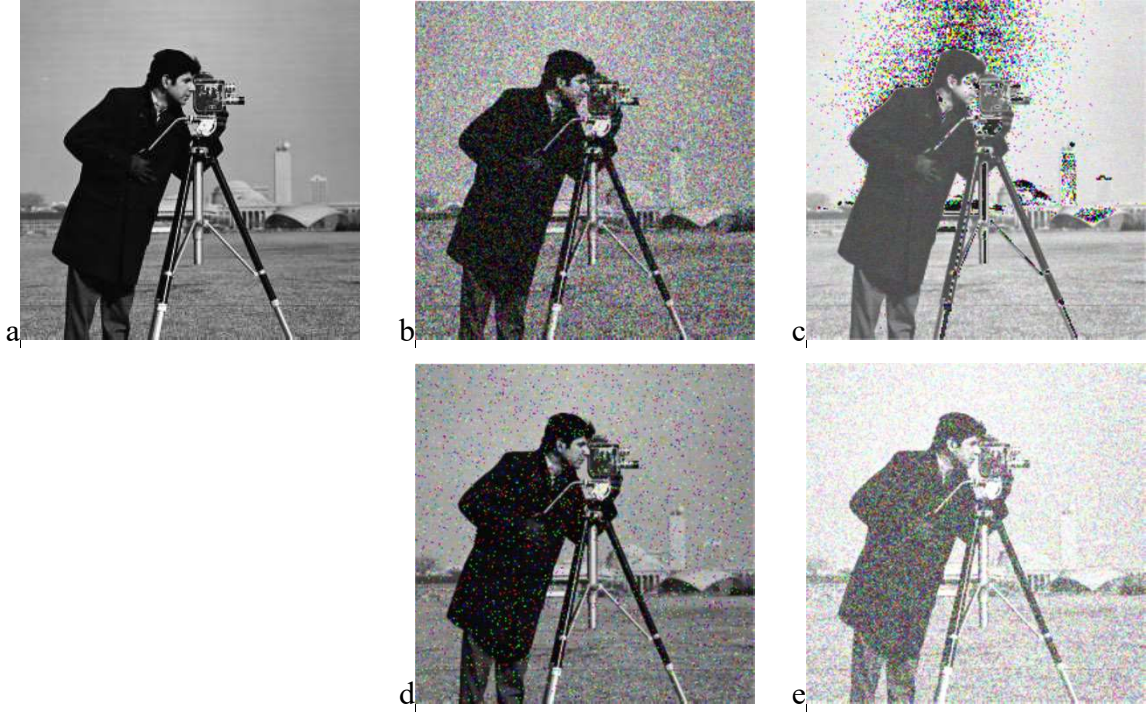


图 4 四种噪声样式, a 原始图像 b 高斯噪声 c 泊松噪声 d 椒盐噪声 e 瑞丽噪声

根据不同噪声的特点和它们的 PDF 分布，可以使用不同的滤波器进行去噪，以下为集中滤波器的介绍：

令 s_{xy} 表示中心为 (x, y) 、大小为 $m \times n$ 的矩形子图像窗口的一组坐标， r 和 c 是 s_{xy} 中包含的像素的行坐标和列坐标， m 和 n 是滤波核的大小。 $g(x, y)$ 表示被污染的噪声图像在 (x, y) 处的像素值， $\hat{f}(x, y)$ 表示滤波后的图像在 (x, y) 处的像素值。

1) 算术平均滤波器

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(r, c) \in s_{xy}} g(r, c) \quad (4-6)$$

2) 几何均值滤波器

$$\hat{f}(x, y) = \left(\prod_{(r, c) \in s_{xy}} g(r, c) \right)^{\frac{1}{mn}} \quad (4-7)$$

3) 谐波平均滤波器

$$\hat{f}(x, y) = \frac{mn}{\sum_{(r,c) \in S_{xy}} \frac{1}{g(r,c)}} \quad (4-8)$$

4) 反谐波平均滤波器 (Q 为滤波器的阶数)

$$\hat{f}(x, y) = \frac{\sum_{(r,c) \in S_{xy}} g(r,c)^{Q+1}}{\sum_{(r,c) \in S_{xy}} g(r,c)^Q} \quad (4-9)$$

5) 维纳滤波器

相位恢复问题通常具有病态性, 因此常常需要进行正则化处理. 经典的维纳滤波是一种典型的正则化处理方法, 其主要思想是设定一个滤波器, 使得滤波后的原始信号估计值与真实值之间的均方误差最小.

对于一幅原始图像 $f(x, y)$, 在退化函数 $h(x, y)$ 和噪声项 $\eta(x, y)$ 的作用下, 得到的退化图像 $g(x, y)$ 的表达式为:

$$g(x, y) = h(x, y) \odot f(x, y) + \eta(x, y) \quad (4-10)$$

根据卷积定理, 在频域中的表达式为:

$$G(u, v) = H(u, v) * F(u, v) + N(u, v) \quad (4-11)$$

频繁使用的一种方法是将退化图像的估计的傅里叶变换由下面式子 (K 为常数) 来近似, 空间域中复原后的图像就由该频率域估计值的傅里叶反变换给出.

$$\hat{F}(x, y) = \left[\frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + K} \right] G(u, v) \quad (4-12)$$

4.3.2 OSS 算法

在实际信号采集过程中, 观测到的幅值经常会受到较大噪声的影响, 噪声会破坏相位恢复的过程, 这使得图像恢复问题变得更加复杂. 为解决噪声带来的影响, 可以利用实际空间目标中的偏离支持区域 (过采样后, 目标以外的区域) 提供的信息. 支持区域以外的区域过采样后理想情况下都被假定为 0, 但噪声存在情况下, 它能够反映噪声的衍射强度特征. 在迭代过程的不同阶段使用适当的滤波器进行卷积, 将平滑的密度分布作用于支持区域之外的区域. 这不仅可以减小噪声的影响, 而且在支撑区域以外施加平滑约束可以减小支撑区域内的振荡. 这也就是 OverSampling Smoothness (OSS) 算法将 HIO 算法中的第四步更换为下式:

$$z''[n] = \begin{cases} z_i'[n], & n \notin \gamma \\ z_i[n] - \beta z_i'[n], & n \in \gamma \end{cases} \quad (4-13)$$

$$z_{i+1}[n] = \begin{cases} z_i''[n], & n \notin \gamma \\ \mathcal{F}\{Z_i''[k]W[k]\}, & n \in \gamma \end{cases}$$

其中 $W[k]$ 是一个高斯函数, 其方差随迭代而减小, 定义为:

$$W[k] = e^{-\frac{1}{2} \left(\frac{k}{\alpha}\right)^2} \quad (4-14)$$

通过修改参数 α , 可以调整高斯滤波器的大小, 以减小对支持区域之外的高频信息的影响, 除此可将高斯滤波器根据数据特点更换为其他滤波器.

4.4 模型指标定义

在迭代过程中, 使用 R_F 衡量恢复的相位的质量, R_F 定义为:

$$R_F = \sum_k \|Z_e[k] - \zeta |Z_r[k]|\| / \sum_k |Z_e[k]| \quad (4-15)$$

其中, $|Z_e[k]|$ 是测量的傅立叶幅值, $|Z_r[k]|$ 是恢复的傅立叶幅值, ζ 是比例因子.

在 OSS 算法中, 按照滤波核的大小划分阶段, 每个阶段中包含数次迭代, 每个阶段结束后选择 R_F 最小的一组相位值作为下一阶段的初始值, 并在完成所有阶段后, 选取 R_F 最小的相位恢复出图像

使用与测量结果的一致性来量化图像的恢复误差:

$$Error = \sum_n |z_r[n] - z_m[n]| / \sum_n |z_m[n]| \quad (4-16)$$

其中, $z_r[n]$ 是最终重建的图像, $z_m[n]$ 是原始图像.

图像的恢复质量使用 PSNR 衡量:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \quad (4-17)$$

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i, j) - K(i, j)\|^2 \quad (4-18)$$

其中, MSE 是原图像与处理图像之间均方误差, MAX_I 表示图像颜色的最大数值, 8 位采样点表示为 255.

5 模型应用及分析

5.1 HIO 算法应用与分析

按照图 1 中自建测试数据的流程生成数据样本用于以下实验, 实验主要目的为分析 HIO 模型的性能.

实验 1: 将一幅 256×256 的图像 (cameraman.png), 对其频谱进行过采样, 即将其放置于 512×512 的网格的中心, 剩余的像素值为 0, 然后进行离散傅里叶变换, 得到相应的 512×512 频谱信息. 用 HIO 算法对图像进行恢复, 恢复结果如图 5 所示.

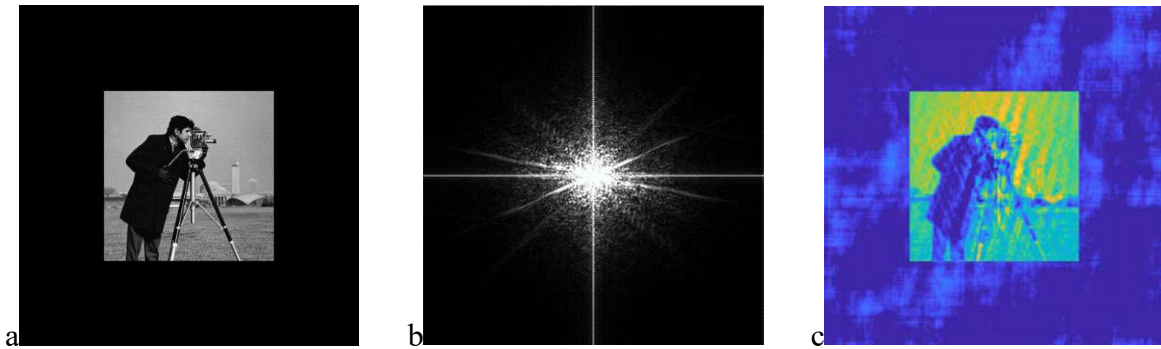


图 5 实验 1 结果 a 过采样的原始图像 b 强度谱 c 恢复的图像

实验 2: 在实验 1 的得到的频谱信息基础上添加高斯噪声, 用 HIO 算法对图像进行恢复恢复结果如图 6 所示.

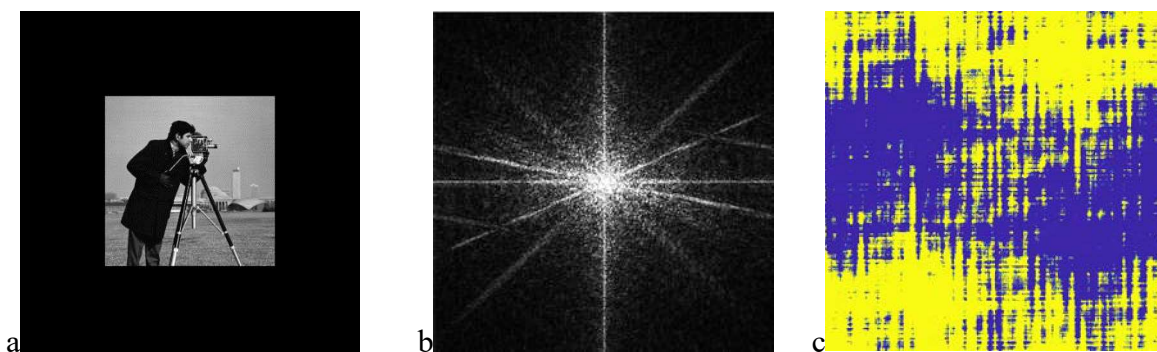


图 6 实验 2 结果 a 过采样的原始图像 b 添加噪声后的强度谱 c 恢复的图像

实验 3: 用 HIO 算法对附件 1 中数据进行恢复, 恢复前先对数据做 fftshift 处理, 恢复结果如图 7 所示.

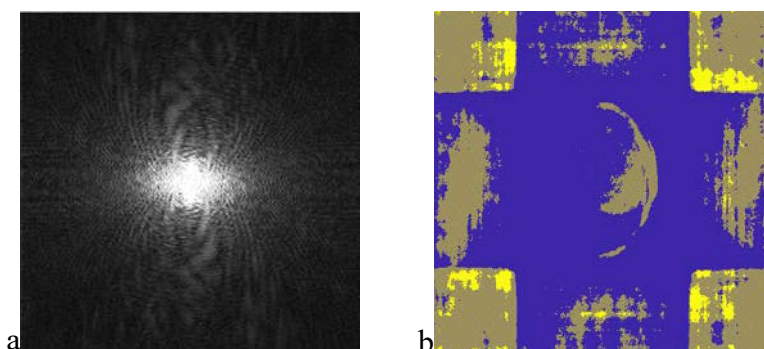


图 7 实验 3 结果 a 强度谱 b 恢复的图像

实验 4: 在实验二的基础上, 对数据先分别进行高斯滤波、拉普拉斯滤波、均值滤波、维纳滤波, 之后通过 HIO 算法恢复, 恢复结果如图 8 所示.

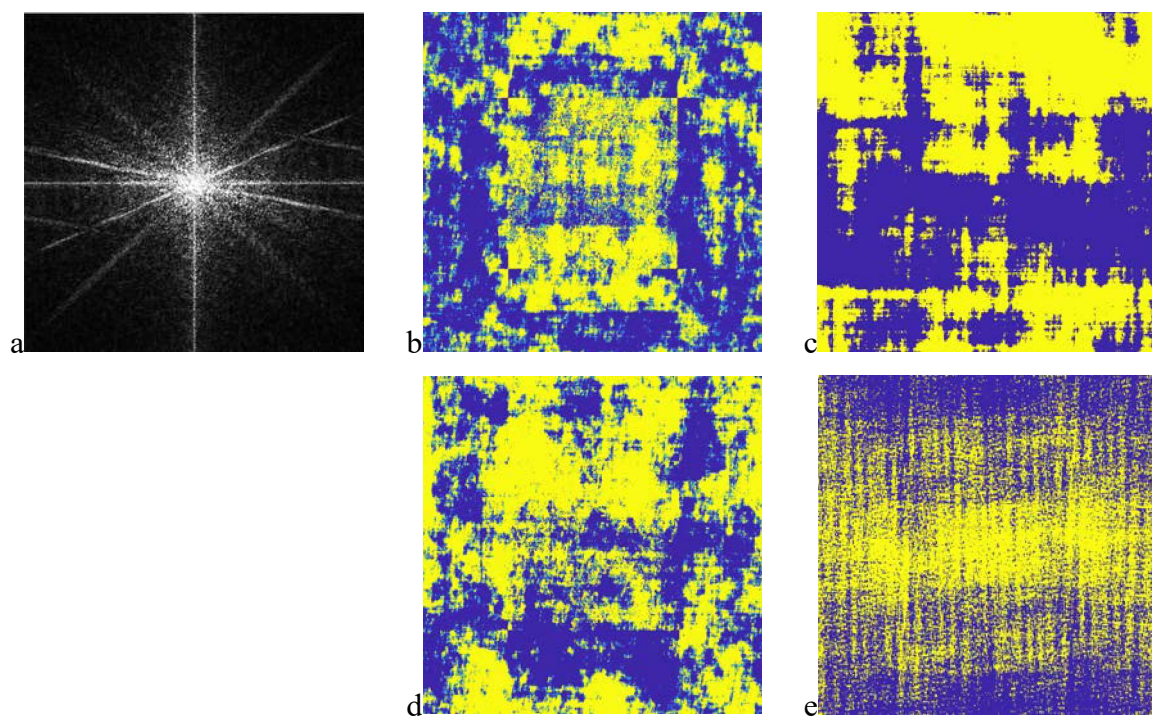


图 8 实验 4 结果 a 含噪声的强度谱 b 高斯滤波后恢复图像 c 拉普拉斯滤波后恢复图像 d 均值滤波后恢复图像 e 维纳滤波后恢复图像

实验 5: 对附件 1 中的数据也分别采用实验 4 中的四种滤波方法后使用 HIO 算法进行恢复. 其中采用维纳滤波后进行恢复的效果最佳, 实验结果如图 9 所示.

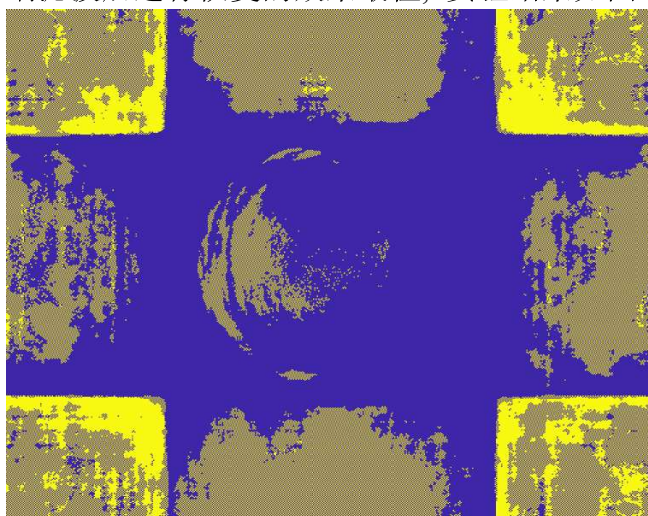


图 9 对附件 1 数据维纳滤波后恢复的图像

根据实验 1 结果可以证明 HIO 算法可以有效地恢复相位并最终恢复出图像, 且求得的相位解是唯一的. 但实验 2 的结果和实验 1 对比, 可以发现 HIO 算法对噪声是非常敏感, 当获取的强度信息中参杂噪声时, 完全无法恢复出原始图像. 从实验 4 的结果发现, 在未知数据获取的环境和条件时, 无法判断数据中含有的噪声类型, 无法针对性去噪时, 即便使用不同的滤波器进行去噪也无法还原出原始图像. 由此可见, 噪声对图像恢复效果影响的关键性, 而且使用上述去噪方法不仅效果不佳, 而且费时费力. 实验 3 和实验 5 的对比结果同样证明了这一结论.

5.2 OSS 算法应用与分析

为解决 HIO 算法对噪声敏感问题, 改进的 OSS 算法在一定程度上对其有效解决. 用 OSS 算法进行如下实验并对结果进行分析.

实验 1: 对一幅 52×28 大小的图像过采样, 形成 512×512 大小的图片, 在经过傅里叶变换获取强度值, 在该强度谱中添加 10% 的泊松噪声, 使用 OSS 算法进行图形恢复, 实验结果如图 10 所示.

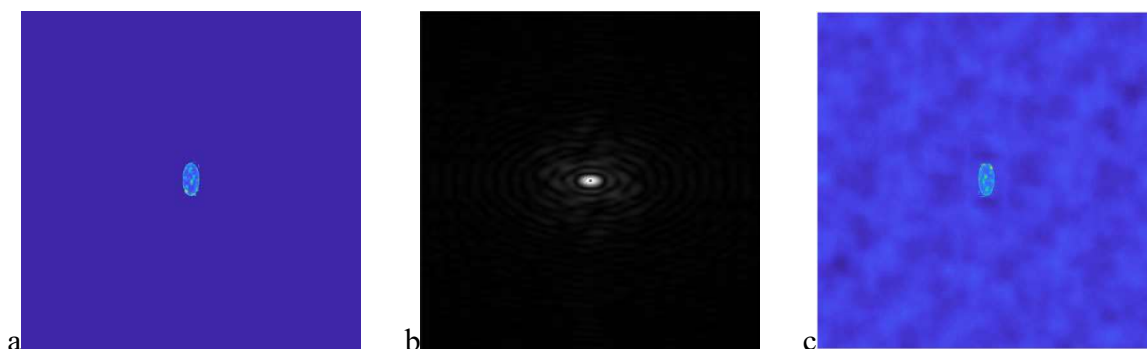


图 10 OSS 算法对含噪强度谱的图像恢复 a 过采样后的原始图像 b 过采样后包含噪声的强度谱 c 恢复的图像

实验 2: 使用 OSS 算法分别对附件 1、2、3 中的数据进行恢复, 恢复结果如图 11 所示.

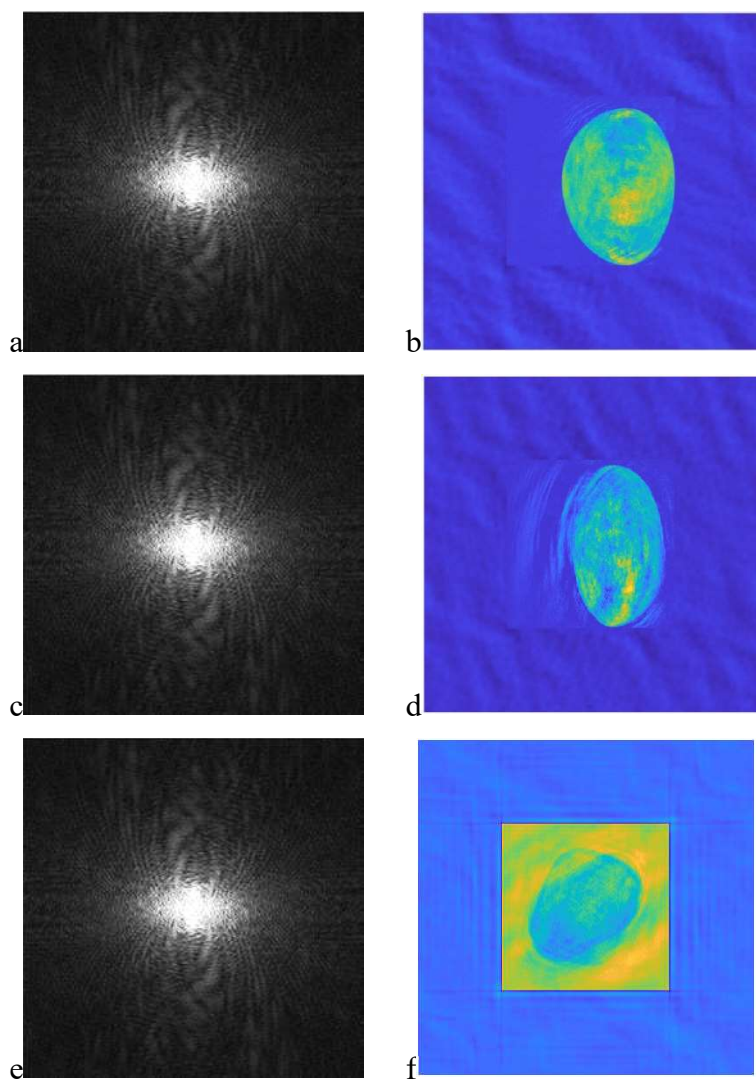
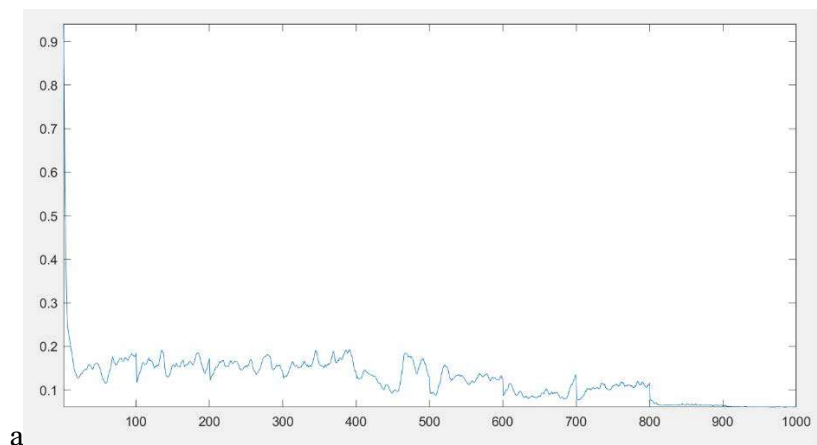


图 11 OSS 算法对附件 1、2、3 的恢复结果 a 附件 1 的强度谱 b 附件 1 恢复的图像 c 附件 2 的强度谱 d 附件 2 恢复的图像 e 附件 3 的强度谱 f 附件 3 恢复的图像

从实验 1 的恢复结果可以知道, OSS 算法有效地解决了噪声敏感问题, 可以实现较高质量的图像恢复. 图 12 为实验 1 过程中 R_F 和 $1-Error$ 的变化曲线, 可以发现 R_F 在迭代过程很快便下降, 并不断趋于 0, 说明 OSS 算法收敛的速度很快且不易振荡. 并根据图 12 (b) 可以发现, 重构的图像与原始图像的而一致性保持在 0.8 附近, 证明恢复的图像的质量较高.



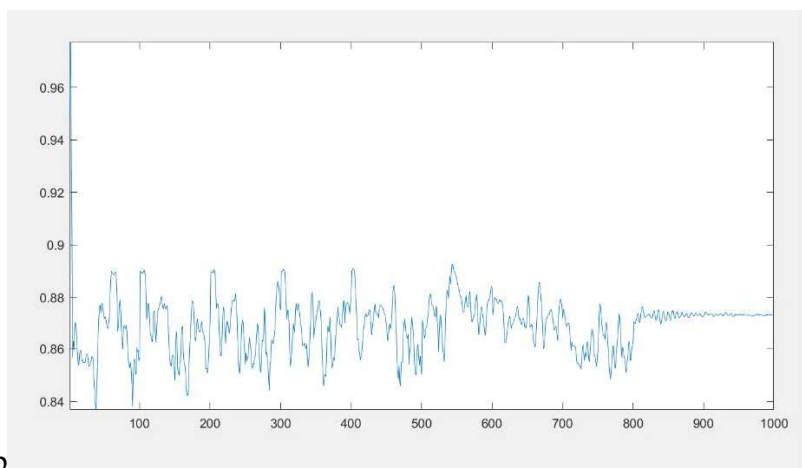


图 12 实验 1 过程中 R_F 和 $1-Error$ 的变化曲线 a R_F 变化曲线 b $1-Error$ 的变化曲线

5.3 图像恢复的影响因素

5.3.1 过采样率的影响

Sayre 指出, 可以通过测量每两个衍射 (频域) 信号之间的强度信息, 即将频域采样率扩大一倍, 来恢复原频域采样点处的相位信息. 在信号 DFT 序列的幅值过采样点 M 满足 $M \geq 2N-1$ (不等式在每个维都成立) 可以保证解的唯一性. 也就是说在多维信号的离散傅里叶相位恢复中, 当过采样倍数为 2 时, 可以从傅里叶测量值中唯一地恢复出原始信号的相位. 这与本题问题 1 和问题 2 所给的条件相对应, 则问题 1 和 2 可以求解出唯一的相位并恢复出图像.

使用 cameraman 图像, 按照过采样、傅里叶变换、保留幅度值的步骤分别生成了过采样率 2-9 的样本, 样本如图 13 所示.

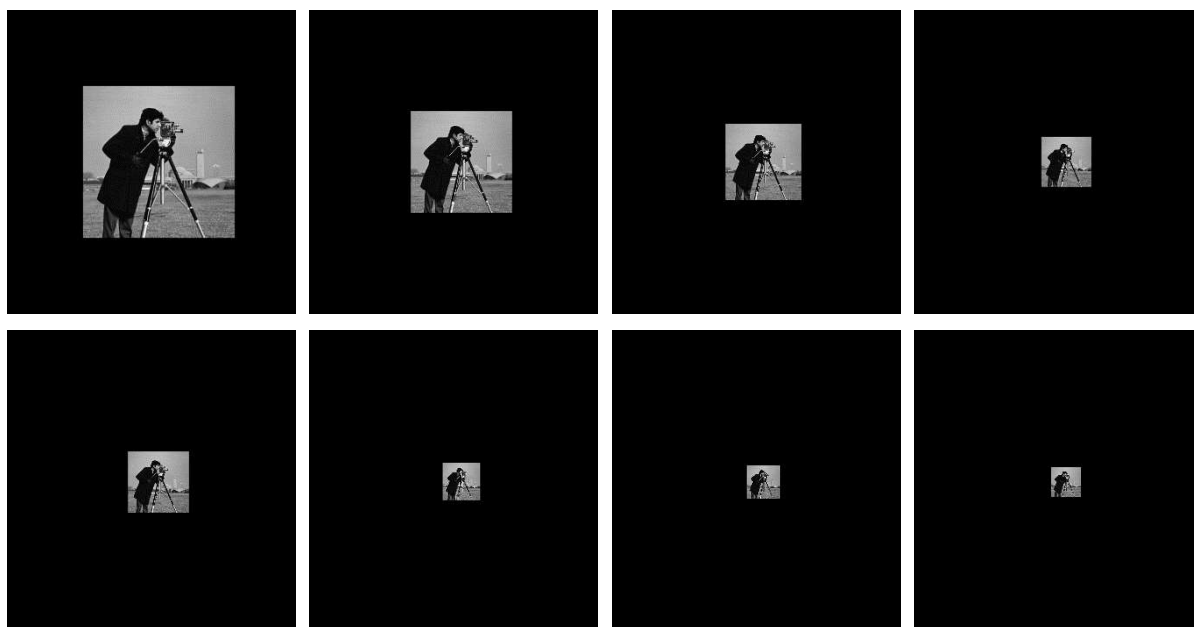


图 13 过采样率为 2-9 的样本

使用 HIO 算法根据上述样本的幅值信息进行图像恢复, 每个样本进行重复 10 次测试, 恢复图像的结果如图 14 所示, 相关数据见表 2. HIO 算法成功恢复出了过采样率为 2 到 8 的样本, 无法恢复出过采样率为 9 和 10 的样本. 而且随着采样率的提高, 图像恢复

成功的概率呈下降趋势，其主要原因是当过采样率较大时，图像使得算法在迭代过程更容易出错，短时间内找不到最优解。PSNR 记录的为对每个过采样率样本进行 10 测试中的最大值，可以发现在 2 到 7 之间 PSNR 变化较小，从图像恢复质量来看，这几个过采样率都是合适的，但是当过采样率继续增加时 PSNR 反而开始下降。综上，过采样率直接影响了模型求解成功的该考虑和图像恢复的质量，因此在选择时应当综合考虑。

表 2 不同过采样率样本恢复数据

过采样率 N	PSNR	恢复概率	过采样率 N	PSNR	恢复概率
2	30.30	0.8	6	25.06	0.3
3	28.65	0.7	7	31.34	0.2
4	29.88	0.7	8	25.26	0.2
5	29.87	0.4	9	-	0

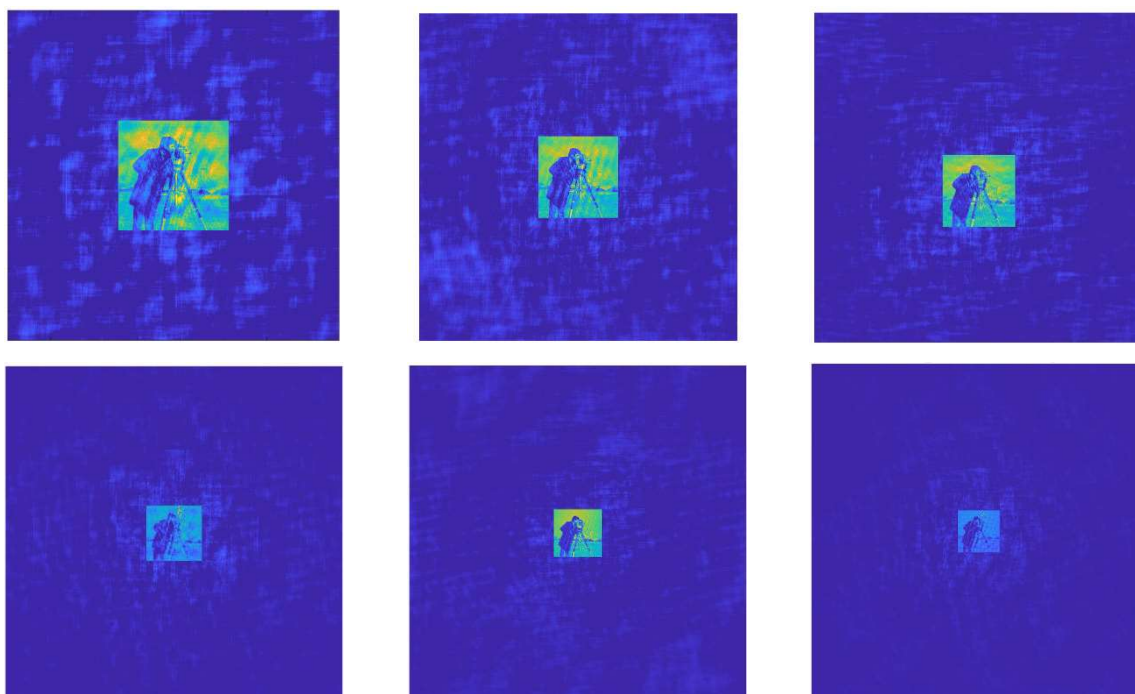


图 14 过采样率分别为 3、4、5、6、7、8 样本的恢复图像

5.3.2 先验知识的影响

除了过采样率可以确保模型的可解，为了尽可能得到傅里叶相位恢复问题的唯一解，还可以添加关于原始信号的先验信息，对信号的重建施加约束。例如，如果真实信号表示某种强度或者概率分布，那么信号本身必须是实值且非负的；另一种先验信息基于信号的稀疏性，即信号向量中仅包含少量的非零元素。在没有先验信息的情况下，模型的唯一解是不存在的。

先验信息可以用于约束相位恢复问题的解空间。基于幅度的相位恢复问题通常存在无数个解，先验信息可以帮助我们在无穷解空间中选择最合适的解。例如，如果我们知道图像的平滑性特征，我们可以使用总变差正则化等方法来推断相位信息，从而获得更加平滑的重建图像。

其次，先验信息还可以在相位恢复问题中引入先验知识。例如，如果我们知道图像中存在特定的结构或纹理信息，我们可以将这些先验知识融入到相位恢复算法中，从而改善重建结果的质量。这种先验信息可以通过基于统计特性的方法、模型驱动的方法或深度学习的方法来获取和利用。

此外，先验信息还可以来自于其他相关任务的学习。例如，在图像去噪任务中学习到的先验信息可以用于相位恢复问题，以帮助恢复出更准确的相位信息。这种迁移学习的方法可以有效地利用已有的先验知识，提升相位恢复算法的性能。在[6] 中有更详细的先验知识的举例，可以参考。

6 模型评价与改进

6.1 模型的优点

- (1) 模型充分结合实际，简化数据获取条件，考虑了诸多重要因素得到合理的模型，这样得到的模型贴合实际，具有较高的应用价值，可以推广到天文成像、全息成像等领域使用；
- (2) 模型合理的描述了相位恢复问题，并且根据题目条件，可以找到最优且唯一解；
- (3) 本文使用的 OSS 算法有效的解决 HIO 算法的噪声敏感和迭代振荡问题，并且可以更准确、快速的恢复相位信息，以实现图像的高质量恢复；

6.2 模型的不足

- (1) 本文建模时完全针对于离散数据进行，离散情况和连续情况是存在差异的，可能无法达到较好的效果；
- (2) 本文提出的模型对于现有条件使用效果较好，由于时间问题没有对其他情况进行检验。对于其他情形(如：更多复杂噪声的混合、实际部分相干光的真实数据)，可能无法达到较好的效果；
- (3) 实际上，真实通过光学等设备过采样得到的数据和傅里叶变换仿真生成的数据是存在差异的，在恢复精度上可能无法达到实验中的指标。

6.3 模型的改进

- (1) 在模型中添加通用正则化项进行约束，使模型更具有泛化性和鲁棒性；
- (2) 将 OSS 算法与深度学习的神经网络进行结合 (例如 Unet 网络)，进一步提高恢复图像的质量；
- (3) 现有模型是针对于理想的远场衍射形况下，具有满足要求的两个平面，而现实中目标平面和像平面的情况是更多变的，因此需要改进模型，使模型在现实中更具有普适性。
- (4) 改进初始化策略，使模型具有更好的收敛性能和噪声鲁棒性能；

参考文献

- [1] 王爱业,潘安,马彩文,等.相位恢复算法:原理,发展与应用(特邀)[J].红外与激光工程, 2022, 51(11):26.
- [2] Mom K , Langer M , Sixou B .Deep Gauss-Newton for phase retrieval[J].Optics Letters, 2023.
- [3] E. Hofstetter, "Construction of time-limited functions with specified autocorrelation functions," Information Theory, IEEE Transactions on, vol. 10, no. 2, pp. 119–126, 1964.
- [4] R. Bates et al., "Fourier phase problems are uniquely solvable in more than one dimension. i: Underlying theory," Optik, vol. 61, no. 3, pp.247–262, 1982.
- [5] J,R,Fienup.Phase retrieval algorithms: a comparison[J].Applied Optics, 1982, 21(15).
- [6] Shechtman Y , Eldar Y C , Cohen O ,et al.Phase Retrieval with Application to Optical Imaging[J]. 2014.
- [7] 戈小娟.基于凸优化的稀疏相位恢复算法研究[D].安徽大学,2015.
- [8] 康旭,刘进.相干衍射成像的相位复原及重建[J].高压物理学报, 2019, 33(3):10.
- [9] Miao,Rodriguez,J.A,等.Oversampling smoothness: An effective algorithm for phase retrieval of noisy diffraction intensities[J].[2023-08-27].
- [10] Hui-Ping L I , Song L I .Phase retrieval with PhaseLift algorithm[J].高校应用数学学报: 英文版 (B 辑) , 2020, 35(4):24.
- [11] Çağatay Işıl, Figen S. Oktem, and Aykut Koç, "Deep iterative reconstruction for phase retrieval," Appl. Opt. 58, 5422-5431, 2019.
- [12] 阮秋琦, 阮宇智译;(美) 拉斐尔·C.冈萨雷斯, 理查德·E.伍兹.国外电子书与通信教材系列 数字图像处理 第4版[M].北京: 电子工业出版社, 2020.
- [13] 朱红云,周哲海,祝连庆,等.相位恢复算法的研究进展[J].激光杂志, 2016, 37(11):5.

附 录

附录 A: 图像恢复结果

结果见压缩包文件夹

附录 B: 主要程序/关键代码

代	操作系统: Windows 10 专业版
码	编程语言: matlab / Python3.7.15
环	编辑器: MATLAB R2020b / Microsoft VS Code
境	代码详见: OSS_samplecode.m / main_hio.m / get_pictures.py

代码清单 1 OSS_samplecode.m 文件

```
function [mask RfacR RfacF RFD]= OSS_samplecode (F2D,supp,iter,beta,showim,modelimg,hiofirst)

%% General assignment of variables
[Rsize,Csize] = size(F2D);
R2D=zeros(Rsize,Csize,10,'single');
toperrs=single(1:10:100);
kfilter=zeros(Rsize,Csize,'single');
realgroup=zeros(Rsize,Csize,2,'single');
realgroup(:,1)=modelimg;

%% Assign variables
stopper = find(F2D== -1);
filtercount=10;
filtnum=0;
store=0;

%% Define support
Rcenter = ceil(Rsize/2);
Ccenter = ceil(Csize/2);
Rsupport = supp(1);
Csupport = supp(2);
half Rsupport = ceil(Rsupport/2);
half Csupport = ceil(Csupport/2);
support = zeros(Rsize,Csize,'single');
support(Rcenter-half Rsupport+1:Rcenter+half Rsupport-1,Ccenter-half Csupport+1:Ccenter+half Csupport-1) = 1;
mask=support;

%% Compute filter parameter alpha
X=1:iter;
FX=(filtercount+1-ceil(X*filtercount/iter))*ceil(iter/(1*filtercount));
FX=((FX-ceil(iter/filtercount))*(2*Rsize)/max(FX))+(2*Rsize/10);
figure(98), plot(X,FX), axis tight; title('OSS Filter Size v. Iterations');

%% Generate initial filter
for kk=1:Rsize
    for jj=1:Csize
        kfilter(kk,jj)=exp( -( ( sqrt((kk-Rcenter)^2+(jj-Ccenter)^2).^2 ) ./ (2* FX(1)^2) ));
    end
end
kfilter=kfilter/max(max(kfilter));

%% Assign random phases
rng('shuffle','twister');
phase angle = rand(Rsize,Csize,'single');

%% Define initial k, r space
initial k=F2D; initial k(initial k== -1)=0;
```

```

k space = initial k.*exp(1i*phase angle);
buffer r space = single(real(ifftn(ifftshift(k space))));
%% Preallocate error arrays
RfacF = zeros(ceil(iter/2),1,'single'); counter1=0; errorF=1;
RfacR = zeros(ceil(iter/2),1,'single'); counter2=0; errorR=1;

%% Image display argument
if showim==1
    figure(1),
end
if nargin<7
    hiofirst=0;
end

%% OSS iterations
for iteration = 1:iter

    %% OSS with Support & Positivity constraint
    r space = real(ifftn(ifftshift(k space)));
    sample = r space.*mask;
    r space = buffer r space-beta*r space;
    sample(sample<0)=r space(sample<0);

    %% Apply frequency filter (OSS)
    if hiofirst==0 || iteration>ceil(iter/filtercount)
        for kk=1:Rsize
            for jj=1:Csize
                kfilter(kk,jj)=exp( -( ( sqrt((kk-Rcenter)^2+(jj-Ccenter)^2).^2) ) ./ (2* FX(ite
ration)^2) ));
            end
        end
        kfilter=kfilter/max(max(kfilter));
        ktemp=fftshift(fftn(r space));
        ktemp=ktemp.*kfilter;
        r space=single(real(ifftn(ifftshift(ktemp))));
    end

    %% Use best result from last filter
    if mod(iteration,ceil(iter/filtercount))==0
        r space=R2D(:,filtnum);
    else
        r space(mask==1)=sample(mask==1);
    end

    %% Update reconstruction
    buffer r space = r space;
    k space = fftshift(fftn(r space));
    phase angle = angle(k space);

    stopper k space = k space(stopper);
    k space = F2D.*exp(1i*phase angle);
    k space(stopper) = stopper k space;

    %% Calculate errors
    if rem(iteration,2)==0

        %% Calculate error in reciprocal space
        Ktemp = sample;
        Ktemp = abs(fftshift(fftn(Ktemp)));
        errorF = sum(sum(abs(Ktemp(F2D~-1)-F2D(F2D~-1)))) / sum(sum(F2D(F2D~-1)));
        counter1=counter1+1; RfacF(counter1) = errorF;

        %% Determine iterations with best error
        filtnum=ceil(iteration*filtercount/iter);
        if errorF<= toperrs(filtnum) && iteration>store+2
            toperrs(filtnum)=errorF;
            R2D(:,filtnum)=r space;
            store=iteration;
        end

        %% Calculate error in real space
        realgroup(:,2)=sample;

```



```

        realgroup2=realgroup(Rcenter-half Rsupport-1:Rcenter+half Rsupport+1,Ccenter-half Cs
support-1:Ccenter+half Csupport+1,:);
        [realgroup2]=align2(realgroup2,0,iteration);
        errorR = sum(sum(abs(realgroup2(:,1)-realgroup2(:,2)))) / sum(sum(realgroup2(:,,
1)));
        counter2=counter2+1; RfacR(counter2) = errorR;

        %% Figure shows progress
        if showim==1

                figure (1),
                subplot(2,2,1), imagesc(squeeze(realgroup2(:,1))), axis image, title(strcat(int2str(FX(iteration)),'--OSS'));
                subplot(2,2,2), imagesc(squeeze(realgroup2(:,2))), axis image, title(int2str(iteration));
                subplot(2,2,3), plot(RfacF), axis([0 ceil(iteration/2) 0 0.8]), title(int2str(errorF*100));
                subplot(2,2,4), plot(RfacR), axis([0 ceil(iteration/2) 0 0.8]), title(int2str(errorR*100));
                drawnow
        end
    end
end

%%% Save results
if rem(iteration,iter)==0
    save ('R2D.mat','R2D','RfacF','RfacR','mask','toperrs','r space');
end

%%% Show image: sum of best 4 steps
s=find(toperrs==min(toperrs));
RFD= squeeze(R2D(:,s));

if showim==1
figure(2),
subplot(2,2,1), imagesc(squeeze(RFD)), axis image;
subplot(2,2,2), imagesc(squeeze(mask)), axis image;
subplot(2,2,3), plot(RfacF), axis tight;
subplot(2,2,4), plot(RfacR), axis tight;
end

%
% F2D = xlsread("F:\数学建模\A题\附件3.csv");
% F2D = ifftshift(F2D);
% F2D(256:278,256:278) = -1;
% modelimg=zeros(512);
% modelimg(129:384,129:384)=1;
% supp = [256,256];
% iter = 5000;
% beta = 0.9;
% showim = 1;
% hiofirst = 1;
% [mask RfacR RfacF RFD]= OSS samplecode (F2D,supp,iter,beta,showim,modelimg,hiofirst) ;
%
```

代码清单 2 main_hio.m 文件

```

sh = 512;                %输入数据图像大小
S=zeros(sh);
S(0.5*sh-127:0.5*sh+128,0.5*sh-127:0.5*sh+128)=1;    %支撑集 ； 将待恢复图片所在区域置为
1
m = xlsread("F:\数学建模\A题\附件1.csv");    %替换对应的强度信息数据
m = ifftshift(m);
```

```

%% 滤波参数-----
% filter_size = 30; % 滤波器大小
% sigma = 3; % 高斯滤波器标准差
% lowpass_filter = fspecial('gaussian', filter_size, sigma);
% % 'gaussian' 可替换为 --> 'sobel', 'prewitt', 'laplacian', 'log', 'average', 'unsharp', 'disk', 'motion'
% m = imfilter(m, lowpass_filter);

% 使用逆滤波恢复图像
% PSF = fspecial('laplacian');
% m = deconvwnr(m, PSF);

% % % 对图像进行高斯滤波
% sigma = 5;
% m = imgaussfilt(m, sigma);

% % 对图像进行维纳滤波
% noise_power = 10;
% m = wiener2(m, [5,5], noise_power);

% 中值滤波
% m = medfilt2(m,[32,32]);

% %均值滤波
% filter_size = [4,4];
% m = imfilter(m, fspecial('average', filter_size));

% % 进行自适应直方图均衡化
% m = adapthisteq(m);

% % 进行小波去噪
% m = wdenoise2(m);

% 拉普拉斯滤波---有些效果
% laplacian_filter = fspecial('laplacian',0.5);
% m = imfilter(m, laplacian_filter, 'symmetric');

% % 带通滤波函数
% D0 = 5; % 截止频率
% n = 2; % 阶数
% m = butterworth_bandpass(m, D0, n);

%%-----

```

```

bata=0.9;           %beta大小
xn=rand(sh);
for n=1:500
    c=Pm(xn,m);
    d=Ps(c,S);
    e=Ps(xn,S);
    xn=(1+bata)*d-e-bata*c+xn;
end

x=uint8(xn);
figure(1);
imshow(x);           %显示灰度图
figure(2);
imagesc(real(x));     %显示颜色缩放图

im = imread("F:\数学建模\cameraman.png"); %原图数据
x = x(0.5*sh-127:0.5*sh+128,0.5*sh-127:0.5*sh+128);
psnr=psnr(im,real(x)); %计算峰值信噪比

```

代码清单 3 get_pictures.py 文件

该程序可以将指定图片按照自定义过采样率采样，并将采样后图像以及最终的强度信息数据保存下来。

```

import cv2
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread(r"cameraman.png",0)
def get_mask(ratio,imgq):
    # 读取图像
    padded_array = np.pad(imgq, pad_width=128*(ratio - 1), mode='constant')
    print(padded_array.shape)
    return padded_array
for i in range(9):
    imgmask = get_mask(i+2,img)
    plt.imsave('caiyang_{:}.png'.format(i+2),imgmask,cmap = 'gray')
    img_fft = np.fft.fft2(imgmask)
    img_abs = np.abs(img_fft)
    ll = pd.DataFrame(img_abs)
    ll.to_csv('caiyang_abs_{:}.csv'.format(i+2),index=False)

```