# 本科生毕业论文（设计）

## Undergraduate Graduation Thesis (Design)

# 题目：层叠成像中的相位恢复问题


院系

School(Department)：数学学院

专业

Major：数学与应用数学

学生姓名

Student Name：吴茼

学号

Student No：17307053

指导教师（职称）

Supervisor(Title)：李嘉


时间：二零二二年四月十一日

Date: April 11th 2022

# 学术诚信声明

本人所呈交的毕业论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。本毕业论文的知识产权归属于培养单位。本人完全意识到本声明的法律结果由本人承担。

本人签名：　　　　　　　　　　日期：

# Statement of Academic Integrity

I hereby acknowledge that the thesis submitted is a product of my own independent research under the supervision of my supervisor, and that all the data, statistics, pictures and materials are reliable and trust- worthy, and that all the previous research and sources are appropriately marked in the thesis, and that the intellectual property of the thesis be- longs to the school. I am fully aware of the legal effect of this statement.

Student Signature：　　　　　　　　　　Date：

# 层叠成像中的相位恢复问题


## 摘要

本文主要讨论偏相干层叠成像中的相位恢复问题。根据被偏向干效应污染的无相位的衍射图序列，恢复出真实图像和用于成像的探针。为了刻画偏向干效应，选用了多种物理模型中的一个，并详细讨论了它与一般化的多模态模型间的关系。将直观的交替投影方法改进为 ADMM 算法，并拓展到多个模态的情形。我们进行了三个仿真数值实验验证了算法的有效性。第一，模态逼近实验。随着模态数目增加，逼近密度矩阵的精度提高，图像的恢复质量提升。算法恢复出来的模态与根据模型产生的标准答案高度相似。第二，尝试在 ADMM 中加入正交化约束，避免搜索过程中不同模态的信息重叠，提升算法的效率。第三，有噪声的情形。



**关键词：相位恢复　偏向干层叠成像　ADMM**

# Partially coherent ptychography

## Abstract

**Key words: Variable Selection Machine Learning Prediction**

# Contents

# List of Figures



# List of Tables

# 1 引言

## 1.1 选题背景与意义

层叠成像是科学领域中一种流行的成像技术，如凝聚态物理、细胞生物学、材料科学和电子学等。在相干光层叠成像实验中，局部的相干 X 射线探头扫描样本，而探测器在远场收集一系列无相位的观测强度值。我们的目标是从强度测量序列中获得样本的高分辨率重建。

相干层叠成像实验通常依靠光圈来定义相干照明。世界各地的研究机构都在投入大量资源来生产更亮的 X 射线源以克服这一限制。同时，大多数产生的 X 射线光子目前都被二级孔径丢弃。即使有足够强度的相干光，曝光期间所需的稳定性通常也是另一个限制因素。总之，相干光源需要严格的实验条件，可能会造成浪费。使用偏相干分析可以减少光强和稳定性限制。

## 1.2 国内外研究现状和相关工作

为了表示偏相干效应，研究者们提出了不同的前向模型。物理学家在[3] 中提出了一个通用模型，它是一种基于 quantum state tomography 描的盲层叠成像模型，它假设探针处于具有 $r$ 个模态的混合状态。诸如交替投影（AP）之类的算法可以从相干情况扩展到偏相干情况以找到 $r$ 个主要模态。在另一种观点中，我们正在重构一个近似的 $rank-r$ 密度矩阵。尽管该模型在各种情况下成功地重建了图像，例如具有平移模糊的“飞行扫描”数据，但多种模式的物理解释尚不清楚，并且与相干函数的关系是间接的。因此，研究者们对不同的实验设置[9]提出了一些特定的模型。

在[1] 中，作者将偏相干效应描述为具有振动核 $\kappa$ 的主要模态 $\omega$ 上的模糊。因为这个前向模型很难计算，他们使用了探针梯度分解 (GDP)，一种新的前向模型，来近似它。然后他们提出了 GDP-ADMM，这是一种迭代求解器，可以联合优化图像、探针和模糊核函数的方差。然而，当模糊核方差大小增加时，逼近精度下降，结果看起来很模糊。他们假设内核是高斯的，只优化方差参数，这可能不适合真实世界的数据。在另一个更简单的模型中，偏相干效应的特征是直接在相干情况下的测量值上添加模糊。它可以被解释为在检测器处模糊或合并多个像素。

在本文中，我们想用数学语言来描述偏相干，并设计一种有效的算法来解决这个问题。我们将主要关注[1]中的振动模型，结合不同模型的优点，得到一个泛化能力和解释能力都很好的模型。然后，我们将利用模型的特殊结构设计一种算

法，并进行数值实验以证明其效率。

## 1.3 本文论文结构与章节安排

本文共分为五节，每章节的内容如下：

第一节：引言。

第二节：模型。详细介绍偏相干效应的不同模型，并阐述他们的联系。

第三节：算法。

第四节：实验。

# 2 模型

## 2.1 纯相干模型

$$f_j = \left|\mathcal{F}\left(\mathcal{S}_j u \circ \omega\right)\right|^2 \tag{1}$$

在离散的情况下, $u \in \mathbb{C}^n$ 是一个 2D 图像，有着 $\sqrt{n} \times \sqrt{n}$ 个像素, $\omega \in \mathbb{C}^{\bar{m}}$ 是一个 2D 的局部探针，有着 $\sqrt{m} \times \sqrt{m}$ 个像素.

$f_j \in \mathbb{R}_+^{\bar{m}} (\forall 0 \leq j \leq N-1)$ 是一叠没有相位的观测值. 这里 $|\cdot|$ 代表一个向量逐个元素取绝对值, o 代表逐个元素的乘法, $\mathcal{F}$ 代表标准化后的离散二维傅里叶变换. 每个 $\mathcal{S}_j \in \mathbb{R}^{\bar{m} \times n}$ 是一个 0-1 矩阵，将大小为 size $\bar{m}$ 的区域 $j$ 从图像 $u$ 中割下来.

在实际中，由于无法完全了解该探针，我们需要解决盲层叠成像的相位恢复 (BP-PR) 问题：

要找到 $\omega \in \mathbb{C}^{\bar{m}}$ and $u \in \mathbb{C}^n$ s.t. $|\mathcal{A}(\omega, u)| = \boldsymbol{a}$,

在这里双线性算子 $\mathcal{A}: \mathbb{C}^{\bar{m}} \times \mathbb{C}^n \rightarrow \mathbb{C}^m$ 和 $\mathcal{A}_j: \mathbb{C}^{\bar{m}} \times \mathbb{C}^n \rightarrow \mathbb{C}^{\bar{m}} \forall 0 \leq j \leq N-1$ 的意义如下:

$\mathcal{A}(\omega, u) := \left(\mathcal{A}_0^T(\omega, u), \mathcal{A}_1^T(\omega, u), \ldots, \mathcal{A}_{N-1}^T(\omega, u)\right)^T$, $\mathcal{A}_j(\omega, u) := \mathcal{F}\left(\omega \circ \mathcal{S}_j u\right)$

和 $\boldsymbol{a} := \left(\boldsymbol{a}_0^T, \boldsymbol{a}_1^T, \ldots, \boldsymbol{a}_{N-1}^T\right)^T \in \mathbb{R}_+^m$.

## 2.2 一些具体的偏相干模型

### 2.2.1 模型一[1]

在连续情形：

$$f_{pc,j}(q) = \int \left|\mathcal{F}_{x \rightarrow q}\left(\mathcal{S}_j u(x) \omega(x-y)\right)\right|^2 \kappa(y) \mathrm{dy} \tag{2}$$

这里 $f_{pc}$ 偏向干情形下的观测强度，$\mathcal{F}_{x \rightarrow q}$ 是标准傅里叶变换。$\kappa$ 是一个在 0 处突起的函数，如高斯函数. 设置 $\kappa$ 为狄拉克函数将它化简为纯相干模型 (1).

在离散情形下：

$$f_{pc,j} = \sum_i \kappa_i \left|\mathcal{F}\left(\mathcal{S}_j u \circ \left(\mathcal{T}_i \omega\right)\right)\right|^2 \tag{3}$$

符号的含义解释如下：平移算子 $\mathcal{T}_i$, 离散高斯函数的权重 $\{\kappa_i\}$, 探针设为周期性边界条件.

广泛地说, 解决 (3) 是一个 non-linear ill-posed problem，我们不知道的有模糊核 $\kappa$, 探针 $\omega$, 和目标图像 $u$.

### 2.2.2 模型二[9]

这里介绍一个更简单的模型:

$$f_{pc} = f * \kappa \tag{4}$$

这里 $f_{pc}$ 是观测到的偏向干强度, $f$ 是纯相干情形下观测到的强度 (1), $*$ 是卷积算子, $\kappa$ 是未知的模糊核函数

我们注意到 (3) 与 (4) 不同，因为 (3) 说明了图像相对于探头的模糊效果，而 (4) 可以解释为探测器模糊或合并多个像素。

## 2.3 一般的偏向干模型

这部分介绍了物理学家提出的偏相干模型。[3]。它是基于 quantum state tomography[1]。假设探针 $w$ 处于混合状态以表示偏向干的效果。

### 2.3.1 一般化的分解模型

$$\begin{aligned}
&\text{Find } u, r \text{ othogonal } w_k \ s.t. \\
&f_{pc,j} = \sum_{k=1}^{r} \left|\mathcal{F}\left(\mathcal{S}_j u \circ (\omega_k)\right)\right|^2 (0 \leq j \leq N-1)
\end{aligned} \tag{5}$$

将 $O_j \in C^{\bar{m}\times\bar{m}}$ 表示为（对角线）矩阵，以表示对 $w$ 的线性变换，s.t. $\mathcal{S}_j u \circ \omega = O_j w$。将 $f_q^* \in C^{1\times\bar{m}}$ 表示为由傅里叶变换 $\mathcal{F}$ 构造的行向量，以表示在频率元素上的投影。构造测量矩阵 $\mathcal{I}_{j\mathbf{q}} = O_j^* f_q f_q^* O_j$ 和密度矩阵 $\rho$，我们得到另一种形式（实际上是 quantum state tomography 中的自然形式）该模型：

$$\begin{aligned}
&\text{Find } u, \rho, s.t. \\
&f_{pc,j}(q) = Tr(\mathcal{I}_{j\mathbf{q}}\rho)(0 \leq j \leq N-1) \\
&\rho \text{ is positive semi-definite, with rank} \leq r
\end{aligned} \tag{6}$$

接下来，我们将解释这种形式的推导。

简单的计算过程：

$$f_{pc,j}(q) = |f_q^* O_j w|^2 = (f_q^* O_j w)^*(f_q^* O_j w) = w^*(O_j^* f_q f_q^* O_j)w$$

[1] `https://homepage.univie.ac.at/reinhold.bertlmann/pdfs/T2_Skript_Ch_9corr.pdf`定理 9.1 的盲 ptychography 模型。这里包含了许多量子力学中的符号。

$$= Tr[w^*(O_j^* f_q f_q^* O_j)w] = Tr[(O_j^* f_q f_q^* O_j)(ww^*)]$$
$$= Tr(\mathcal{I}_{j\mathbf{q}}\rho)$$

这有点像相位提升的过程。

当 $w$ 处于单一状态（希尔伯特空间中的一个向量）时，$\rho = w^*w$ 是一个秩一矩阵。在偏相干的情况下，我们使用混合状态来建模 $w$。例如，在状态 $\psi_1$ 中的概率为 0.5，在 $\psi_2$ 中的概率为 0.5（$\psi_1$ 和 $\psi_2$ 在这里不一定是正交的）。现在 $w$ 不能再用向量来表示了（ps.$w \neq p_1\psi_1 + p_2\psi_2$，后者仍然是一个确定的纯状态向量）。相反，混合状态由将密度矩阵泛化为具有更高秩的矩阵表示：

$$\rho = \sum_k p_k \psi_k \psi_k^*$$

容易发现 $\rho$ 是一个半正定矩阵，我们可以利用谱定理分解 $\rho$，$r(\text{rank of } \rho)$ 正交状态 $wk$：

$$\rho = \sum_{k=1}^{r} w_k w_k^* \tag{7}$$

$$f_{pc,j}(q) = \operatorname{Tr} \mathcal{I}_{j\mathbf{q}}\rho = \operatorname{Tr}[\mathcal{I}_{j\mathbf{q}} \sum_{k=1}^{r} w_k w_k^*]$$
$$= \sum_{k=1}^{r} w_k^* \mathcal{I}_{j\mathbf{q}} w_k = \sum_{k=1}^{r} |f_q^* O_j w_k|^2$$

而这正是 (5)$f_{pc,j} = \sum_{k=1}^{r} |\mathcal{F}(\mathcal{S}_j u \circ (\omega_k))|^2$.（$f_{pc,j}(q)$ 是在频率 $q$ 处的单一值，而 $f_{pc,j}$ 是整个衍射图像)

我们可以把这个写成二次型的形式:

$$\begin{aligned} f_{pc,j}(q) &= \operatorname{Tr} \mathcal{I}_{j\mathbf{q}}\rho = Tr[(O_j^* f_q f_q^* O_j)\rho] = Tr[(O_j^* f_q)^* \rho (O_j^* f_q)] = (O_j^* f_q)^* \rho (O_j^* f_q) \\ &= g_q^* \rho g_q = \sum_{x_1} \sum_{x_2} \overline{g_q(x_1)} \rho(x_1, x_2) g_q(x_2) \end{aligned} \tag{8}$$

where $g_q = O_j^* f_q = \overline{S_j u} \circ f_q$, $\overline{g_q} = S_j u \circ \overline{f_q}$

这是[9]中提到模型的离散版本.

### 2.3.2 模型之间的关系

在本节中，我们将解释通用模型如何与特定模型 (3) 和 (4) 联系。

首先我们考虑 (3)。我们把 $\kappa_i$ 放在里面：

$$f_{pc,j}=\sum_i \left|\mathcal{F}\left(\mathcal{S}_j u\circ\left(\sqrt{\kappa_i}\mathcal{T}_i\omega\right)\right)\right|^2=\sum_i \left|\mathcal{F}\left(\mathcal{S}_j u\circ\left(\hat{\omega}_i\right)\right)\right|^2 \tag{9}$$

多种模态 $\hat{w}_i$ 由移位 $w$ 产生。然后我们可以构造密度矩阵并使用 SVD 得到一个低秩近似。

$$\rho=\sum_i \hat{w}_i\hat{w}_i^*\approx\sum_{k=1}^r w_k w_k^*$$

至于 (4), 我们介绍 coherence function 的定义:

$$\gamma(x_1,x_2)=\frac{\rho(x_1,x_2)}{w(x_1)\overline{\overline{w(x_2)}}}$$

也就是:

$$\gamma=\rho./(ww^*),\rho=\gamma\circ(ww^*) \tag{10}$$

这里 ./ 意味着逐个元素的除法.

当我假设 coherence function$\gamma\left(\mathbf{x}_1,\mathbf{x}_2\right)$ 只与两个点之间的距离有关 $\mathbf{x}_1,\mathbf{x}_2$, i.e. $\gamma\left(\mathbf{x}_1,\mathbf{x}_2\right)=\gamma\left(0,\mathbf{x}_2-\mathbf{x}_1\right)$, 我们可以把远场强度写成一个卷积的形式[9]

$f_{pc}=\kappa * f$. 如果我们知道 point spread function $\kappa(q)$, 那我们可以得到它的傅里叶逆变换 $\gamma(0,x)=(\mathcal{F}^{-1}\kappa)(x)$. 在这个假设上，我们得到 $\gamma(x_1,x_2)$. 如果我们也知道探针 $w$, 那我们可以得到 $\rho$, SVD 帮助我们在模型 (5)中找到主要模态 $w_k$。

通过上面的不同模型间的联系，我们可以得到“标准模式分解”作为参考。

# 3 基于 ADMM 的数值算法

常用的交替投影算法（AP）可以被重写成 ADMM 形式, 它更快而且更稳定. 我们推广 [8] 中的 ADMM 算法到多模态的情形. 为了引导算法找到的多个模态信息不重叠，提高搜索效率，我们还引入了模态间的正交约束。

现在 $w \in \mathbb{C}^{(px \times py) \times r}$ 是 r 个模态 $r$ . $u \in \mathbb{C}^{Nx \times Ny}$ 是图像. $f \in \mathbb{C}^{(px \times py) \times N}$ 是真实观察到的衍射图像. 记 $Y = \sqrt{f}$ 为对应模长.

一个辅助变量 $z = \mathcal{A}(\omega, u) \in \mathbb{C}^{(px \times py) \times N \times r}$ 被引入. $\mathcal{A}$ 是一个从图像 $u$ 和 $r$ 个不同模态 $w_k := w(:, :, k) \in \mathbb{C}^{px \times py}$ 产生衍射图像序列的算子，每个模态产生 $N$ 帧. 对于多维向量，符号: 表示自由维度，我们可以固定一些索引来从原始向量中提取特定维度。

根据一般模型3, 问题叙述为:

$$\text{Find } \omega, u \ s.t.$$
$$\bar{\mathcal{A}}(\omega, u) = Y$$

这里 $\mathcal{A} : \mathbb{C}^{(px \times py) \times r} \times \mathbb{C}^{Nx \times Ny} \to {}^{(px \times py) \times N \times r}$,$\mathcal{A}_j : \mathbb{C}^{px \times py} \times \mathbb{C}^{Nx \times Ny} \to \mathbb{C}^{px \times py}$, $\bar{\mathcal{A}}_j : \mathbb{C}^{(px \times py) \times r} \times \mathbb{C}^{Nx \times Ny} \to \mathbb{R}_+^{(px \times py)}$, 而且 $\bar{\mathcal{A}} : \mathbb{C}^{(px \times py) \times r} \times \mathbb{C}^{Nx \times Ny} \to \mathbb{R}_+^{(px \times py) \times r} (\forall 0 \leq j \leq N-1)$ 被表示为:

$$z(:, :, j, k) = \mathcal{A}_j(\omega_k, u) := \mathcal{F}(\omega_k \circ \mathcal{S}_j u) \in \mathbb{C}^{px \times py},$$
$$z(:, :, :, k) = \left(\mathcal{A}_0^T(\omega_k, u), \mathcal{A}_1^T(\omega_k, u), \ldots, \mathcal{A}_{N-1}^T(\omega_k, u)\right)^T \in \mathbb{C}^{(px \times py) \times N},$$
$$z(:, :, j, :) = \left(\mathcal{A}_j^T(\omega_1, u), \mathcal{A}_j^T(\omega_2, u), \ldots, \mathcal{A}_j^T(\omega_r, u)\right)^T \in \mathbb{C}^{(px \times py) \times r}.$$

,

$$\bar{\mathcal{A}}_j(\omega, u) := \textstyle\sum_{k=1}^r |\mathcal{A}_j(\omega_k, u)|^2 \in \mathbb{R}_+^{px \times py},$$
$$\bar{\mathcal{A}}(\omega, u) := \left(\bar{\mathcal{A}}_0^T(\omega, u), \bar{\mathcal{A}}_1^T(\omega, u), \ldots, \bar{\mathcal{A}}_{N-1}^T(\omega, u)\right)^T \in \mathbb{R}_+^{(px \times py) \times N},$$

和 $Y = \left(\boldsymbol{a}_0^T, \boldsymbol{a}_1^T, \ldots, \boldsymbol{a}_{N-1}^T\right)^T \in \mathbb{R}_+^{(px \times py) \times N}$.

那么 $\mathcal{G}(z) = ||\sqrt{\sum_{k=1}^r |z(:, :, :, k)|^2} - Y||^2$ 测量模型计算的值与真实值之间的差异。

$\mathcal{X}_1$ 和 $\mathcal{X}_2$ 表示 $w$ 和 $u$ 的先验范围. $l = px \times py$, $\mathcal{X}_3$ 是正交指示函数 $D\alpha \in \mathbb{C}^{l \times r}$ ( $D \in \mathbb{C}^{l \times r}$ 标准正交 s.t. $D^* D = I$）, $\alpha \in \mathbb{R}^{r \times r}$ 表示模长. $\Omega$ 是一个形状调整算子, $\Omega(D\alpha) := reshape(D\alpha, [px, py, r]) \in \mathbb{C}^{(px \times py) \times r}$. 有时候我们将 $\Omega(D^k \alpha^k)$ 简写为 $\Omega^k$ ，$\Omega(D\alpha)$ 简写为 $\Omega$.

于是我们得到:

$$
\begin{aligned}
&\min_{\omega,u,z} \mathcal{G}(z)+\mathbb{I}_{\mathcal{X}_1}(\omega)+\mathbb{I}_{\mathcal{X}_2}(u)+\mathbb{I}_{\mathcal{X}_3}(D\alpha) \\
&s.t. \quad z-\mathcal{A}(\omega,u)=0, \quad \Omega(D\alpha)-w=0.
\end{aligned} \tag{11}
$$

相应的增广拉格朗日乘子为：

$$
\begin{aligned}
\Upsilon_\beta(\omega,u,z,\Lambda):=&\mathcal{G}(z)+\mathbb{I}_{\mathcal{X}_1}(\omega)+\mathbb{I}_{\mathcal{X}_2}(u)+\Re(\langle z-\mathcal{A}(\omega,u),\Lambda\rangle)+\frac{\beta}{2}\|z-\mathcal{A}(\omega,u)\|^2 \\
&+\Re(\langle\Omega(D\alpha)-w,\Lambda_2\rangle)+\frac{\beta_2}{2}\|\Omega(D\alpha)-\omega\|^2
\end{aligned}
$$

这里 $\Lambda,\Lambda_2\in\mathbb{C}^{(px\times py)\times N\times r}$ 是乘子. Let $\Lambda=\Lambda/\beta,\Lambda_2=\Lambda_2/\beta_2$, 我们将 $\Re$ 部分和增广部分组合起来得到:

$$
\begin{aligned}
\Upsilon_\beta(\omega,u,z,\Lambda):=&\mathcal{G}(z)+\mathbb{I}_{\mathcal{X}_1}(\omega)+\mathbb{I}_{\mathcal{X}_2}(u)+\mathbb{I}_{\mathcal{X}_3}(D\alpha) \\
&+\frac{\beta}{2}\|z-\mathcal{A}(\omega,u)+\Lambda\|^2-\frac{\beta}{2}||\Lambda||^2+\frac{\beta_2}{2}\|\Omega(D\alpha)-w+\Lambda_2\|^2-\frac{\beta_2}{2}||\Lambda_2||^2
\end{aligned} \tag{12}
$$

在 ADMM 中, 寻求以下问题的鞍点：

$$
\max_{\Lambda,\Lambda_2}\min_{\omega,u,z,D,\alpha}\Upsilon_\beta(\omega,u,z,\Lambda,\Lambda_2,D,\alpha)
$$

解决上述鞍点问题的一个自然方案是将它们分裂，得到包括四步迭代的 ADMM（只有 $\omega$ 或 $u$ 的子问题具有近端项），如下所示：

$$
\begin{aligned}
&\text{Step 1: } \omega^{k+1}=\arg\min_{\omega}\Upsilon_\beta\left(\omega,u^k,z^k,\Lambda^k\right)+\frac{\beta_2}{2}||\omega-(\Omega(D^k\alpha^k)+\Lambda_2)||^2+\frac{\alpha_1}{2}\left\|\omega-\omega^k\right\|_{M_1^k}^2, \\
&\text{Step 2: } u^{k+1}=\arg\min_{u}\Upsilon_\beta\left(\omega^{k+1},u,z^k,\Lambda^k\right)+\frac{\alpha_2}{2}\left\|u-u^k\right\|_{M_2^k}^2, \\
&\text{Step 3: } z^{k+1}=\arg\min_{z}\Upsilon_\beta\left(\omega^{k+1},u^{k+1},z,\Lambda^k\right), \\
&\text{Step 4: } D^{k+1}=\arg\min_{D}\mathbb{I}_{\mathcal{X}_3}(D\alpha)+\frac{\beta_2}{2}\|\Omega(D^k\alpha^k)-\omega^{k+1}\|^2 \\
&\text{Step 5: } \alpha^{k+1}=\arg\min_{\alpha}\frac{\beta_2}{2}\|\Omega(D^{k+1}\alpha^k)-\omega^{k+1}\|^2 \\
&\text{Step 6: } \Lambda^{k+1}=\Lambda^k+\left(z^{k+1}-\mathcal{A}\left(\omega^{k+1},u^{k+1}\right)\right) && (13) \\
&\text{Step 7: } \Lambda_2^{k+1}=\Lambda_2^k+(\Omega(D^{k+1}\alpha^{k+1})-\omega^{k+1}) && (14)
\end{aligned}
$$

为简单起见，我们在下面的分析中忽略了 Step1 和 Step2 中的稳定二次项。

## 3.1 子问题 $w$ 和 $u$

w.r.t. the probe $\omega$ :

$$
\begin{aligned}
\omega^{k+1} &= \arg\min_{\omega\in\mathcal{X}_1} \frac{1}{2}\left\|z^k+\Lambda^k-\mathcal{A}\left(\omega,u^k\right)\right\|^2+\frac{\beta_2}{2}||\omega-(\Omega^k+\Lambda_2^k)||^2 \\
&= \arg\min_{\omega\in\mathcal{X}_1} \frac{1}{2}\left\|\hat{z}^k-\mathcal{A}\left(\omega,u^k\right)\right\|^2+\frac{\beta_2}{2}||\omega-\hat{\Omega}^k||^2 \\
&= \arg\min_{\omega\in\mathcal{X}_1} \frac{1}{2}\sum_{j,i}\left\|\mathcal{F}^{-1}\hat{z}(:,:,j,i)^k-\omega(:,:,i)\circ\mathcal{S}_j u^k\right\|^2+\frac{\beta_2}{2}\sum_i||\omega(:,:,i)-\hat{\Omega}^k(:,:,i)||^2 \\
&\text{with } \hat{z}^k:=z^k+\Lambda^k, \hat{\Omega}^k:=\Omega^k+\Lambda_2^k
\end{aligned}
$$

步骤 1 的近似解为（详见附录 5.1)

$$
\omega^{k+1}=\operatorname{Proj}\left(\frac{\beta\sum_j\left(\mathcal{S}_j u^k\right)^*\circ\left[\left(\mathcal{F}^{-1}\hat{z}^k\right)(:,:,j,:)\right]+\beta_2\hat{\Omega}^k}{\beta\sum_j\left|\mathcal{S}_j u^k\right|^2+\beta_2};\mathcal{X}_1\right) \tag{15}
$$

到 $\mathcal{X}_1$ 的投影算子定义为 $\operatorname{Proj}(\omega;\mathcal{X}_1) := \mathcal{F}^{-1}(\mathcal{F}(\omega)\circ C_w)$, 这里 $\mathcal{X}_1=\{\omega: \mathcal{F}(\omega)$ supports on the index function $C_w\}$. $\mathcal{F}^{-1}$ 作用在 $\hat{z}$ 的前两个维度上 (i.e. $\hat{z}_{j,i}:=\hat{z}(:,:,j,i)$) and $\omega$ (i.e. $\omega_i:=\omega(:,:,i)$).

我们有:

$$
u^{k+1}=\operatorname{Proj}\left(\frac{\sum_{j,i}\mathcal{S}_j^T\left(\left(\omega_i^{k+1}\right)^*\circ\mathcal{F}^{-1}\hat{z}_{j,i}^k\right)}{\sum_{j,i}\left(\mathcal{S}_j^T\left|\omega_i^{k+1}\right|^2\right)};\mathcal{X}_2\right)\ . \tag{16}
$$

这里 $S_j^T$ 是将参数映射到图像 $u$ 中的目标位置 $j$ 的算子。

## 3.2 子问题 $z$

$$
\begin{aligned}
z^{k+1} &= \arg\min_z \mathcal{G}(z)+\frac{\beta}{2}\left\|z-\mathcal{A}\left(\omega^{k+1},u^{k+1}\right)+\Lambda^k\right\|^2 \\
&= \arg\min_z \frac{1}{2}||\sqrt{\sum_{i=1}^r|z(:,:,:,i)|^2}-Y||^2+\frac{\beta}{2}\left\|z-z^+\right\|^2
\end{aligned}
$$

$$
=\arg\min_z\sum_{x,y,j}[\frac{1}{2}(\sqrt{\sum_{i=1}^r|z(x,y,j,i)|^2}-Y(x,y,j))^2+\frac{\beta}{2}||z(x,y,j,:)-z^+(x,y,j,:)||^2]
$$

这里 $z^+=\mathcal{A}\left(\omega^{k+1},u^{k+1}\right)-\Lambda^k$

Step 3 的闭形式解为（详见附录 5.1):

$$z_i^{k+1} = \frac{z_i^+ \dfrac{Y}{M^k} + \beta z_i^+}{1+\beta}, 1 \le i \le r \tag{17}$$

where $z_i := z(:,:,:,i)$ and $M^k = \sqrt{\sum_i |z_i^+|^2} \in \mathbb{C}^{px \times py \times N}$

## 3.3 子问题 $D$ 和 $\alpha$

$$\begin{aligned} D^{k+1} &= \arg\min_D \|\Omega - w^{k+1} + \Lambda_2^k\|^2 \\ &= \arg\min_D \|D\alpha^k - \hat{w}^{k+1}\|^2 \end{aligned}$$

这里 $\hat{w}^{k+1} = reshape(\omega^{k+1} - \Lambda_2^k, [l, r])$,$D^* D = I$

$D^{k+1}$ 的闭形式是 (细节在附录中 5.1):

$$D^{k+1} = UV^* \tag{18}$$

$\alpha$ 的更新更加简单:

$$\alpha^{k+1} = \arg\min_\alpha \|D^{k+1}\alpha - \hat{w}^{k+1}\|^2 = \arg\min_\alpha \sum_i ||\alpha_i D^{k+1}(:,i) - \hat{w}^{k+1}(:,i)||^2$$

注意到每个 $\alpha_i \in \mathbb{R}$ 可以被独立地解决, 一阶优化条件给出:

$$\alpha_i^{k+1} = \sum_{i_0} \Re[\overline{D^{k+1}(i_0,i)}\hat{w}^{k+1}(i_0,i)](1 \le i \le r) \tag{19}$$

**Algorithm 1:** ADMM for general mixed-state model(5)

**Initialization**: Set the number of states $r$, $\omega^0, u^0, z^0 = \mathcal{A}(\omega^0, u^0), \Lambda^0, \Lambda_2^0 = 0$;
$D^0$ and $\alpha^0$ from SVD on $\omega^0$
maximum iteration number Iter $_{\text{Max}}$ , and parameter $\beta$,$\beta_2$

**Output**: $u^\star := u^{Iter_{Max}}$ and $\omega^\star := \omega^{\mathrm{Iter}_{Max}}$

**for** $ii = 0$ *to* $Iter_{Max} - 1$ **do**
  Compute $\omega^{k+1}$ by (15) with $\hat{z}^k := z^k + \Lambda^k$ ;
  Compute $u^{k+1}$ (16). with $\hat{z}^k$ the same as above;
  Compute $z_i^{k+1}$,$1 \leq i \leq r$ by (20). with $z^+ = \mathcal{A}\left(\omega^{k+1}, u^{k+1}\right) - \Lambda^k$;
  Compute $D^{k+1}$ (21). ;
  Compute $\alpha_i, 1 \leq i \leq r$ (19). ;
  Update the multiplier as Step 6 and Step 7 of (13) and (14);
**end**

# 4 数值实验

代码在 MATLAB 中实现。首先，我们介绍实验中的设置，然后我们对 2.2 中的特定模型生成的仿真数据进行实验。在每个实验中，我们比较不同设置中的重建图像和模态。我们还生成了 2.3.2 中的“标准模态分解”，并将我们的结果与它们进行比较。

## 4.1 实验设置

### 4.1.1 参数

| Parameters | Illustration | Values |
|---|---|---|
| $N_x, N_y$ | size of image $u$ | 128,128 |
| $p_x, p_y$ | size of phobe $w$ | 64,64 |
| $Dist$ | scan distance between neighborhood frames | 4,8,16 |
| $N$ | number of frames in diffusion image stacks | |
| $r$ | number of states(modes) | 1(coherent) to 15 |
| gridFlag | types of scan methods | 1(rectangular lattice),<br>2(hexagonal),3(randomly disturb on 2) |
| blurFlag | types of partially coherent effect | 1(4),2(3) |

为了处理不平凡模糊性，非周期性的扫描可以去除扫描几何的周期性，例如，将少量随机偏移添加到扫描网格集中。($gridFlag = 3$)。或者我们可以添加关于模态的先验知识作为附加约束，例如将模态（频域）限制在一个圆圈内。

$Dist$ 是成功重建的重要参数。一般而言，$Dist$ 越小，数据中的重叠区域和冗余越多，我们可以得到质量更高的重建图像。在这里我们发现 $Dist = 8$ 就足够了，而 $Dist = 16$ 总是失败。

#### 4.1.2 度量指标

为了评估算法的性能，我们引入了 3 个指标。

1. 相对误差 $err$ 和信噪比 $snr$

$$err^k = \frac{||cu^k - u_{true}||_F}{||cu^k||_F}, c = \frac{sum(u_{true} \circ \overline{u^k})}{||u^k||_F^2}$$

$$snr^k = -20\log_{10}(err^k)$$

$||\cdot||_F$ 是 Frobenius 范数。$err$ 测量重建图像和真实图像之间的差异。$c$ 是估计的比例因子，$sum$ 表示目标矩阵中所有元素的总和。

2. R-factor $R$

Let $zz = \mathcal{A}_j\left(\omega^k, u^k\right)$

$$R^k := \frac{\left\|\sqrt{\sum_{i=1}^{r}|zz(:,:,:,i)|^2} - Y\right\|_1}{\|Y\|_1}$$

$R$ 测量重建衍射序列和真实虚了 $Y$ 之间的差异。我们并不总是知道真实图像 $u_{true}$，衍射序列 $Y$ 是我们算法的唯一输入数据，因此 R-factor 可以用来验证收敛。

3. 模态逼近误差 $err_M$

$$err_M^k = \frac{||c\rho^k - \rho_{true}||_F}{||\rho_{true}||_F}, c = \frac{sum(\rho_{true} \circ \overline{\rho^k})}{||\rho^k||_F^2}$$

$err_M$ 测量来自模态的重建密度矩阵 $\rho^k$ 与来自理论模型的标准密度矩阵 $\rho_{true}$ 之间的差异。

### 4.1.3 对模态的操作

1. 初始化

   在以下实验中，我们设置 $u^0 = \mathbf{1}_{N_x \times N_y}$ 和初始探针 $m = \frac{1}{N}\mathcal{F}^{-1}\left(\sum_j Y(:,:,j)\right)$. 我们通过随机干扰一个初始模态 $m$ 来生成其他初始模态。$r$ 个初始模态是通过将初始模态乘以不同的随机复数值数组来创建的（模长部分在 $[0,1]$ ，相位部分在 $[0,2\pi]$) 的范围内。）然后我们得到 $w^0$。

2. 正交化

   在我们的算法中计算的模式 $w_k$ 并不总是正交的。然而，我们可以通过生成密度矩阵 $\rho$ 来轻松正交化它们，并像在 (7) 中一样对 $\rho$ 执行谱分解。模式的正交表示一般是唯一的（特别是当 $\rho$ 的特征值都不同时，这总是发生在现实世界的数据中）。

   注意到，模态的数量总是较小，即 $r \ll l := p_x \times p_y$，我们直接对 $l \times r$ 模态矩阵执行 SVD，而不是 $l \times l$ 密度矩阵 $\rho$。我们还可以选择前 $r' \leq r$ 个模态进行近似。

**Algorithm 2:** SVD-based orthogonalization for phobes(Matlab)

**Input**: $w \in \mathbb{C}^{px \times py \times r}$, the number of modes needed $r'$

**Output**: $w_{ort} \in \mathbb{C}^{px \times py \times r'}$

```
phobe matrix ss = reshape(w, [l r]) ;
[U, S, V] = svd(ss, 'econ') ;
q = U(:, 1 : r') * S(1 : r', 1 : r') ;
w_ort = reshape(q, [px py r']) ;
```

   为了获得更清晰的表达，在我们显示最终模态之前总是执行正交化操作。我们还想弄清楚是否可以添加正交化来改进我们的算法1。

3. 压缩

   在嘈杂的情况下，我们考虑提取较少的主要模式以避免噪声失真。

## 4.2 实验：多模态逼近偏向干效应

$Dist = 8$, $gridFlag = 1$, $blurFlag = 2$ 和 $\kappa$ 是一个高斯核，$\sigma = (15, 15)$。ADMM 的算法参数选择 $\beta = 0.05$，这里不考虑正交化约束，所以 $\beta_2 = 0$。

**Algorithm 3:** SVD-based compression for phobes(Matlab)

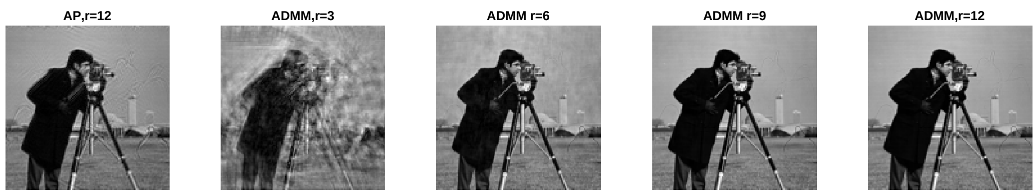
**Input**: $w \in \mathbb{C}^{px \times py \times r}$, the number of main modes kept $r'$

**Output**: $w_{com} \in \mathbb{C}^{px \times py \times r}$

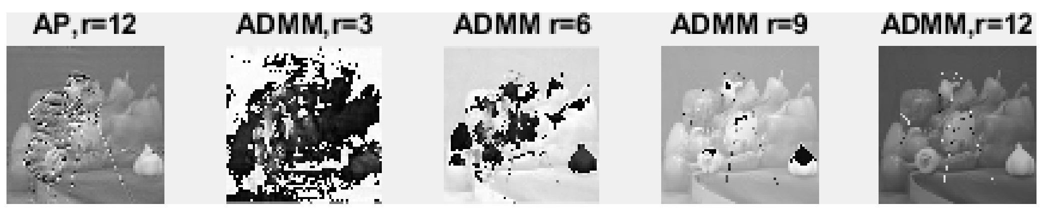
phobe matrix $ss = reshape(w, [l\ r])$ ;

$[U, S, V] = svd(ss, 'econ')$ ;

$q = U(:, 1 : r') * S(1 : r', 1 : r') * V(:, 1 : r')$ ;




(a) Amplitude




(b) Phase

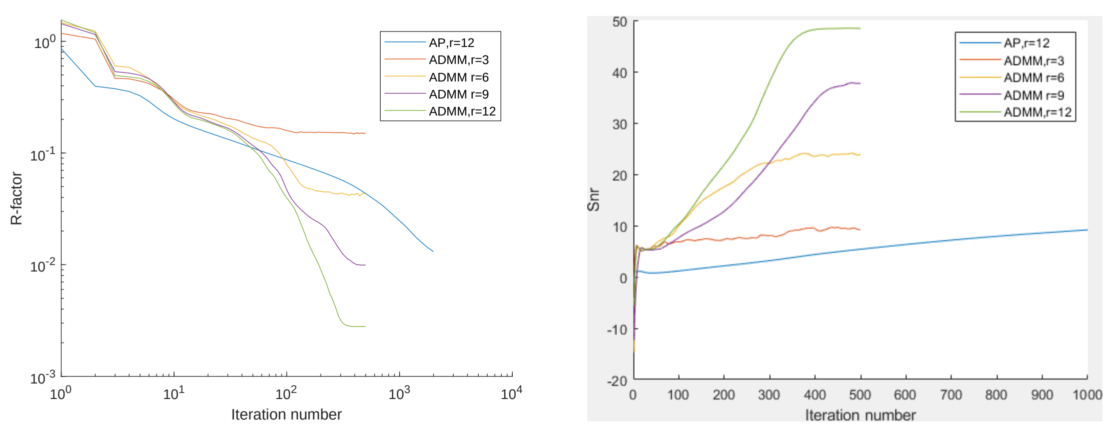



Figure 2: R and snr.

我们首先比较使用不同数量模态的重建图像的 $snr$ 和 $R$。当模态数量增加时，

$R - factor$ 减少，$snr$ 增加。这表明重建图像的质量提高了。

使用 ADMM 算法的 3、6、9、12 个模态逼近算法时，$R$ 在 500 次迭代后稳定在 0.15、0.042、0.0099、0.0028，而 AP 的 $R$ 在 2000 次迭代后不收敛。

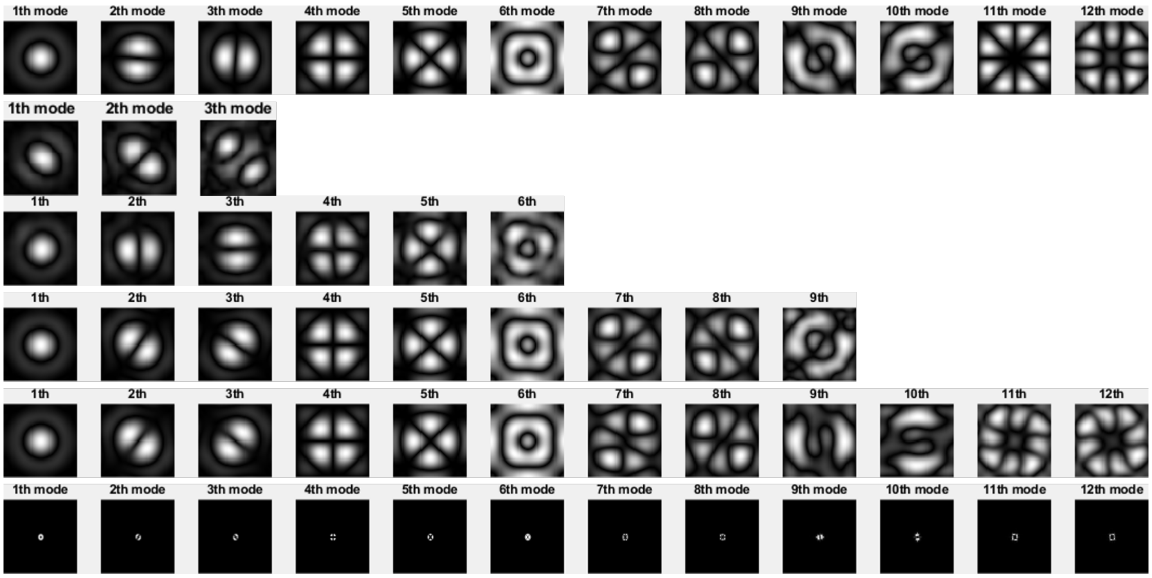



Figure 3: 模态分解。第一行代表“标准模态分解”。最后两行代表 12 种模态的具体样式。这里的模态被表示在在时域中，除了最后一行在频域中。

“模态标准分解”是通过对模型生成的标准密度矩阵执行 SVD 并提取前 12 个模态获得的。如图 3 所示，我们的算法通常可以捕获主要模态并获得密度矩阵的最佳近似。

此外，将 r 个模态的 $err_M$ 表示为 $err_M^r$。最优 $err_M^{r,*}$ 可以从低秩逼近理论中计算出来，我们将我们的 $err_M^r$ 与它进行比较。将标准密度矩阵 $\rho_{true}$ 的奇异值表示为

$$err_M^{r,*} = \min_{\rho, rank(\rho)=r} \frac{||\rho_{true} - \rho||_F}{||\rho_{true}||_F} = \sqrt{\frac{\sum_{i=r+1}^{d} s_i^2}{\sum_i s_i^2}} = \sqrt{1 - s_{cum}(r)}$$

,

$$where\ s_{cum}(r) = \frac{\sum_{i=1}^{r} s_i^2}{\sum_i s_i^2}$$

如图 5 所示，$err_M^*$ 达到 0.01 左右，$err_M$ 接近 $err_M^*$，有 12 种模态，这表明我们的算法在低秩逼近标准密度矩阵中做得不错。

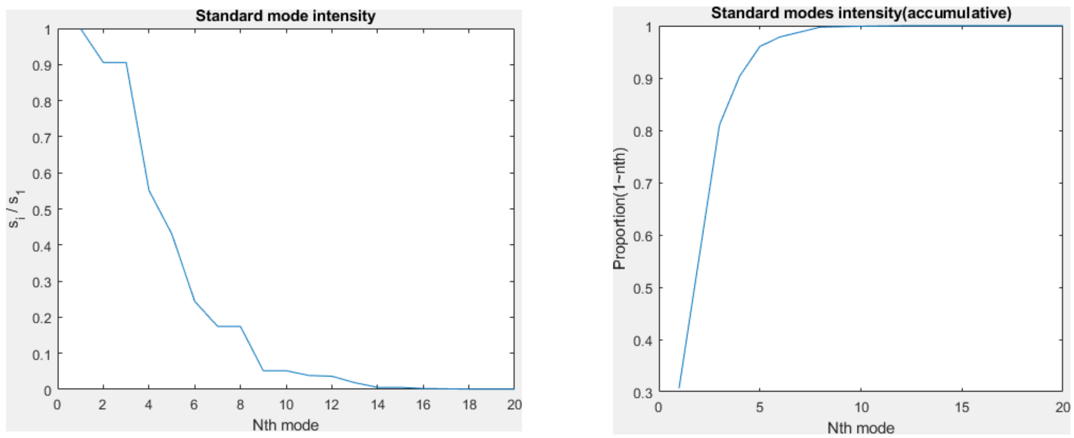



Figure 4: 标准密度矩阵的奇异值分布。左边子图中的纵轴代表 $i^{th}$ 最大奇异值与第一个 $s_i/s_1$ 的比值，右边的纵轴代表 $S_{cum}(i)$。奇异值呈指数下降，矩阵近似低秩。

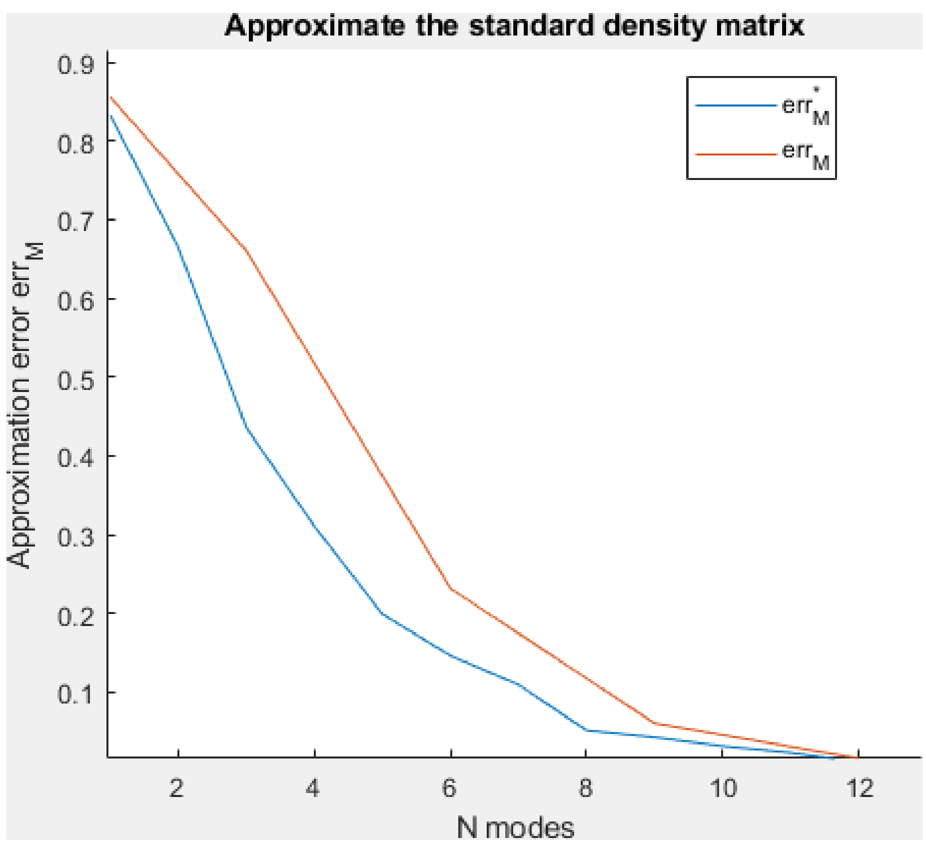



Figure 5

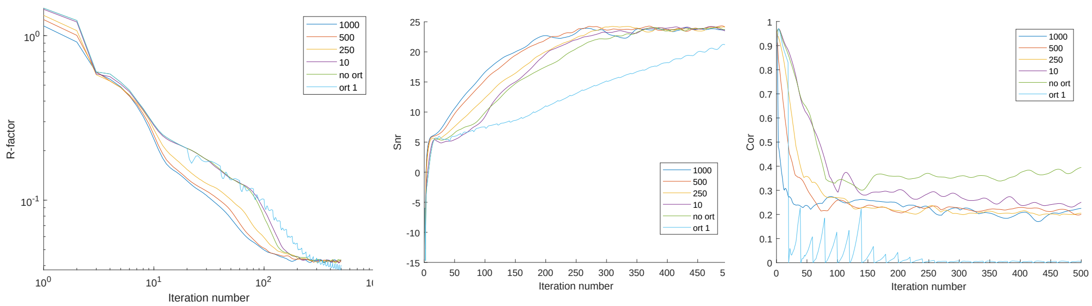



Figure 6: 纵轴和横轴在 R 因子的第一个子图中以对数刻度设置。蓝线代表 $\beta_2/\beta = 1000$，在这种情况下效果最好。绿线表示没有正交化的结果。“ort 1”表示每 20 次迭代执行一次正交化。

## 4.3 加入正交约束条件

实验设置与上面相同，只是我们更改 $ratio = \beta_2/\beta$ 以在 ADMM 中引入正交化约束。$\beta_2$ 越大，正交化约束越严格。作为参考，我们还尝试按照算法 2 每 20 次迭代对模态执行正交化操作。

500 次迭代后重建的图像相似，而使用具有正交化约束的 ADMM 算法的 $snr$ 和 $Rfactor$ 改进更快。并且随着 $\beta_2$ 的增大，模态之间的相关度 $coherence$ 也会下降得更快。

## 4.4 有噪声的情形

$Dist = 8$, $gridFlag = 3$, $blurFlag = 2$ 和 $\kappa$ 是一个高斯核，$\sigma = (15, 15)$, $\beta = 0.05$。

通过 MATLAB 命令在衍射图像 $Y$ 上添加泊松噪声：

$$Y = poissrnd(Y * (\eta))/\eta;$$

这里使用 $\eta = 0.0675$。

比较了四种不同的设置：9 个模态带噪声、9 个模态无噪声、6 个模态在压缩操作之后被保留、6 个模态带噪声和 6 个模态无噪声。具体来说，我们按照 3 每 20 次迭代进行压缩，每次保持 6 个模态。

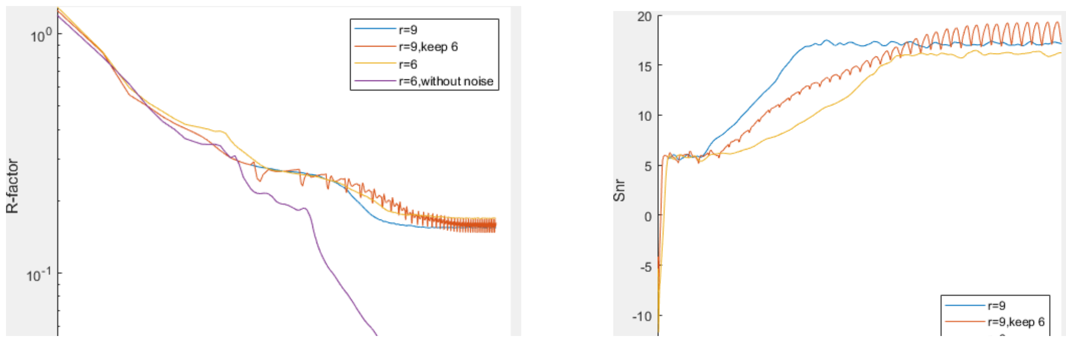




Figure 7: Reconstructed images in the noisy case

结果随着压缩操作周期性地波动，而最终在嘈杂的情况下表现最好。虽然差异并不明显，但我们可以考虑在约束中添加压缩，而不是执行直接截断操作。

# 5 附录

## 5.1 ADMM 中的子问题

### 5.1.1 $\omega$

本质上，在这个子问题中，每个状态 $\omega_i = w(:,:,i)$ 都是独立的。然后我们可以分别优化每个 $w_i$。

$$\omega_i^{k+1} = \arg\min_{\omega \in \mathcal{X}_1} \frac{1}{2} \sum_j \left\| \mathcal{F}^{-1} \hat{z}(:,:,j,i)^k - \omega(:,:,i) \circ \mathcal{S}_j u^k \right\|^2$$

本质上，在这个子子问题中，$w_i$ 中的每个元素都是独立的。这样只需要解决以下一维约束二次问题：

$$\omega_i^{k+1}(t) = \arg\min_{|x| \le C_\omega} \rho_t^k(x).$$

这里 $\rho_t^k(x) := \frac{1}{2} \sum_j \left| \left( \mathcal{F}^{-1} \hat{z}(:,:,j,i)^k \right)(t) - x \times \left( \mathcal{S}_j u^k \right)(t) \right|^2 \forall x \in \mathbb{C}$

计算 $\rho_t^k(x)$ 的导数。注意这里 $\rho$ 是一个复变量实值函数，我们使用 wirtinger 导数。

$$\begin{aligned}
\nabla \rho_t^k(x) &= \frac{d\rho_t^k(x)}{dx^*} \\
&= \sum_j \left( x \times \left| \left( \mathcal{S}_j u^k \right)(t) \right|^2 - \left( \mathcal{S}_j u^k \right)^* (t) \left( \mathcal{F}^{-1} \hat{z}_{j,i}^k \right)(t) \right) \\
&= x \times \left( \sum_j \left| \left( \mathcal{S}_j u^k \right)(t) \right|^2 \right) - \sum_j \left( \left( \mathcal{S}_j u^k \right)^* (t) \left( \mathcal{F}^{-1} \hat{z}_{j,i}^k \right)(t) \right)
\end{aligned}$$

一阶最优条件是 $\nabla \rho_t^k(x) = 0$。然后 $w_i$ 的闭形式解给出为

$$\omega_i^{k+1} = \operatorname{Proj}\left( \frac{\sum_j \left( \mathcal{S}_j u^k \right)^* \circ \left( \mathcal{F}^{-1} \hat{z}_{j,i}^k \right)}{\sum_j \left| \mathcal{S}_j u^k \right|^2}; C_\omega \right)$$

### 5.1.2 z

$$\begin{aligned}
z^{k+1} &= \arg\min_z \mathcal{G}(z) + \frac{\beta}{2} \left\| z - \mathcal{A}\left( \omega^{k+1}, u^{k+1} \right) + \Lambda^k \right\|^2 \\
&= \arg\min_z \frac{1}{2} || \sqrt{\sum_{i=1}^r |z(:,:,:,i)|^2} - Y ||^2 + \frac{\beta}{2} \left\| z - z^+ \right\|^2
\end{aligned}$$

$$= \arg\min_z \sum_{x,y,j} [\frac{1}{2}(\sqrt{\sum_{i=1}^{r} |z(x,y,j,i)|^2} - Y(x,y,j))^2 + \frac{\beta}{2}||z(x,y,j,:) - z^+(x,y,j,:)||^2]$$

这里 $z^+ = \mathcal{A}\left(\omega^{k+1}, u^{k+1}\right) - \Lambda^k$

对于任意固定的 $x, y, j$ 和自由的 $i$，问题可以看成：

$$z^*(x,y,j,:) = \arg\min_{z_{x,y,j} \in \mathbb{C}^r} \frac{1}{2}(||z_{x,y,j}|| - Y_{x,y,j})^2 + \frac{\beta}{2}||z_{x,y,j} - z^+_{x,y,j}||^2$$

请注意，对于固定的 $||z_{x,y,j}||$，表达式中的第一项是固定的。为了优化第二项，我们应该始终选择与 $z^+_{x,y,j}$ 方向相同的 $z_{x,y,j}$。所以我们有 $||z_{x,y,j} - z^+_{x,y,j}||^2 = (||z_{x,y,j}|| - ||z^+_{x,y,j}||)^2$

$$\frac{z(x,y,j,i)}{||z_{x,y,j}||} = \frac{z^+(x,y,j,i)}{||z^+_{x,y,j}||}, z(x,y,j,i) = ||z_{x,y,j}||\frac{z^+(x,y,j,i)}{||z^+_{x,y,j}||}$$

要确定 $z_{x,y,j}$，我们只需要确定 $||z_{x,y,j}||$。将其表示为 $a$。

$$||z_{x,y,j}||^* = \arg\min_{a \in \mathbb{R}} \frac{1}{2}(a - Y_{x,y,j})^2 + \frac{\beta}{2}(a - ||z^+_{x,y,j}||)^2$$

一阶最优性条件很容易给出：

$$a = \frac{Y_{x,y,j} + \beta||z^+_{x,y,j}||}{1+\beta}$$

步骤 3 的闭式形式解为：

$$z_i^{k+1} = \frac{z^+\dfrac{Y}{M^k} + \beta z_i^+}{1+\beta}, 1 \le i \le r \tag{20}$$

where $M^k = \sqrt{\sum_i |z^+(:,:,:,i)|^2} \in \mathbb{C}^{px \times py \times N}$

### 5.1.3 D

$$\begin{aligned} D^{k+1} &= \arg\min_D \|\Omega - w^{k+1} + \Lambda_2^k\|^2 \\ &= \arg\min_D \|D\alpha^k - \hat{w}^{k+1}\|^2 \end{aligned}$$

where $\hat{w}^{k+1} = reshape(\omega^{k+1} - \Lambda_2^k, [px \times py, r])$,$D^*D = I$

这是正交 Procrustes 问题中的一个特例. [2]

$$\begin{aligned}
\|D\alpha^k - \hat{w}^{k+1}\|^2 =& Tr[(D\alpha^k - \hat{w}^{k+1})^*(D\alpha^k - \hat{w}^{k+1})] \\
=& ||\alpha^k||_F^2 - Tr[(\alpha^k)^* D^* \hat{\omega}^{k+1}] - Tr[(\hat{\omega}^{k+1})^* D\alpha^k] + ||\hat{\omega}^{k+1}||_F^2
\end{aligned}$$

$$\begin{aligned}
D^{k+1} &= \arg\max_D Tr[(\alpha^k)^* D^* \hat{\omega}^{k+1}] + Tr[(\hat{\omega}^{k+1})^* D\alpha^k] \\
&= \arg\max_D \Re(Tr[(\alpha^k)^* D^* \hat{\omega}^{k+1}]) \\
&\xlongequal{\alpha\in\mathbb{R}^{r\times r}} \arg\max_D \Re(Tr[D^*(\hat{\omega}^{k+1}\alpha^k)])
\end{aligned}$$

考虑 SVD 分解：$\hat{\omega}^{k+1}\alpha^k = USV^*$

$$\begin{aligned}
D^{k+1} &= \arg\max_D \Re(Tr[D^* USV^*]) = \arg\max_D \Re(Tr[(V^* D^* U)S]) \\
&\xlongequal{\hat{D}=V^*D^*U} \arg\max_{\hat{D}} \Re(Tr[\hat{D}S])
\end{aligned}$$

我们得到 $\hat{D} = I_r$ 时是最优的, $I_r$ 为前 $r$ 个对角元为 1，其余为 0 的对角阵。而且:

$$D^{k+1} = UV^* \tag{21}$$

[2] https://en.wikipedia.org/wiki/Orthogonal_Procrustes_problem

*Bibliography


[1] Chang, Huibin, et al. ”Partially coherent ptychography by gradient decomposition of the probe.” Acta Crystallographica Section A: Foundations and Advances 74.3 (2018): 157-169.

[2] Wolf E. New theory of partial coherence in the space-frequency domain. Part I: spectra and cross spectra of steady-state sources[J]. JOSA, 1982, 72(3): 343-351.

[3] Thibault P, Menzel A. Reconstructing state mixtures from diffraction measurements[J]. Nature, 2013, 494(7435): 68-71.

[4] Tian, Lei, et al. ”Multiplexed coded illumination for Fourier Ptychography with an LED array microscope.” Biomedical optics express 5.7 (2014): 2376-2389.

[5] Thibault P, Dierolf M, Bunk O, et al. Probe retrieval in ptychographic coherent diffractive imaging[J]. Ultramicroscopy, 2009, 109(4): 338-343.

[6] Griffiths, David J., and Darrell F. Schroeter. Introduction to quantum mechanics. Cambridge University Press, 2018.

[7] Fannjiang A, Strohmer T. The numerics of phase retrieval[J]. Acta Numerica, 2020, 29: 125-228.

[8] Chang, Huibin, Pablo Enfedaque, and Stefano Marchesini. ”Blind ptychographic phase retrieval via convergent alternating direction method of multipliers.” SIAM Journal on Imaging Sciences 12.1 (2019): 153-185.

[9] Konijnenberg S. An introduction to the theory of ptychographic phase retrieval methods[J]. Advanced Optical Technologies, 2017, 6(6): 423-438.

[10] Attouch H, Bolte J, Redont P, et al. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality[J]. Mathematics of operations research, 2010, 35(2): 438-457.

[11] Candes E J, Strohmer T, Voroninski V. Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming[J]. Communications on Pure and Applied Mathematics, 2013, 66(8): 1241-1274.

[12] Boyd S, Parikh N, Chu E. Distributed optimization and statistical learning via the alternating direction method of multipliers[M]. Now Publishers Inc, 2011.