Partially coherent ptychography

July 2021

Contents

1 Models

1.1 Coherent model

$$f_j = |\mathcal{F}(\mathcal{S}_j u \circ \omega)|^2 \tag{1.1}$$

In a discrete setting, $u \in \mathbb{C}^n$ is a 2D image with $\sqrt{n} \times \sqrt{n}$ pixels, $\omega \in \mathbb{C}^{\bar{m}}$ is a localized 2D probe with $\sqrt{\bar{m}} \times \sqrt{\bar{m}}$ pixels.

$f_j \in \mathbb{R}_+^{\bar{m}} (\forall 0 \leq j \leq N-1)$ is a stack of phaseless measurements. Here $|\cdot|$ represents the element-wise absolute value of a vector, o denotes the elementwise multiplication, and $\mathcal{F}$ denotes the normalized 2 dimensional discrete Fourier transform. Each $\mathcal{S}_j \in \mathbb{R}^{\bar{m} \times n}$ is a binary matrix that crops a region j of size $\bar{m}$ from the image u.

In practice, as the probe is almost never completely known, one has to solve a blind ptychographic phase retrieval (BP-PR) problem:

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

where bilinear operators $\mathcal{A} : \mathbb{C}^{\bar{m}} \times \mathbb{C}^n \to \mathbb{C}^m$ and $\mathcal{A}_j : \mathbb{C}^{\bar{m}} \times \mathbb{C}^n \to \mathbb{C}^{\bar{m}} \forall 0 \leq j \leq N-1$ are denoted as follows:

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

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

1.2 Specific partially coherent model

1.2.1 Model1[1]

Coherence and vibrations kernels can be combined into one, such that partially coherent ptychography imaging with coherence kernel function κ in a continuous setting:

$$f_{pc,j}(q) = \int |\mathcal{F}_{x \to q}(\mathcal{S}_j u(x) \omega(x-y))|^2 \kappa(y) \mathrm{d}y \tag{1.2}$$

where f_{pc} is the measured partial coherent intensity and $\mathcal{F}_{x \to q}$ is the normalized Fourier transform. κ is a function that spikes at 0 like guassians. Setting κ to the Dirac delta function reduces it to the coherent model (1.1).

The partially coherent intensity in a discrete setting is generated as

$$f_{pc,j} = \sum_i \kappa_i |\mathcal{F}(\mathcal{S}_j u \circ (\mathcal{T}_i \omega))|^2 \tag{1.3}$$

with translation operator $\mathcal{T}_i$, discrete Gaussian weights $\{\kappa_i\}$, and periodical boundary condition for the probe.

Generally speaking, solving (1.3) is a non-linear ill-posed problem with an unknown kernel κ, unknown phobe ω, and unknown target image u.

1.2.2 Model2[9]

Another simpler model mentioned here is:

$$f_{pc} = f * \kappa \tag{1.4}$$

where f_{pc} is the measured partial coherent intensity, f is the coherent intensity in (1.1), $*$ denotes the convolution operator, and κ is the unknown kernel function (Fourier transform of the complex coherence function).

We remark that (1.3) is quite different from (1.4) , since (1.3) illustrates the effects of blurring of images with respect to the probe, while (1.4) can be interpreted as blurring or binning multiple pixels at the detector.

1.3 General partially coherent model

This part explains the partially coherent model proposed by physicists.[3]. It is a blind ptychography model based on quantum state tomography[1]. Phobe w is assumed to be in a mixed state to represent a partially coherent effect.

1.3.1 Decompositon model

$$\begin{aligned} &\text{Find } u, r \text{ othogonal } w_k \ s.t. \\ &f_{pc,j} = \sum_{k=1}^{r} \left|\mathcal{F}\left(\mathcal{S}_j u \circ (\omega_k)\right)\right|^2 (0 \le j \le N-1) \end{aligned} \tag{1.5}$$

Denote $O_j \in C^{\bar{m}\times\bar{m}}$ as a (diagonal) matrix to represent linear transform to w, s.t. $\mathcal{S}_j u \circ \omega = O_j w$. Denote $f_q^* \in C^{1\times\bar{m}}$ as a row vector constructed from Fourier transform $\mathcal{F}$, to represent projection on frepuency element. Construct measurement matrix $\mathcal{I}_{j\mathbf{q}} = O_j^* f_q f_q^* O_j$ and density matrix ρ, we get another form(actually a natural one in quantum state tomography) of the model:

$$\begin{aligned} &\text{Find } u, \rho, s.t. \\ &f_{pc,j}(q) = Tr(\mathcal{I}_{j\mathbf{q}}\rho)(0 \le j \le N-1) \\ &\rho \text{ is positive semi-definite, with rank} \le r \end{aligned} \tag{1.6}$$

Next, we will explain the derivation of this form.

Simple calculation process:

$$\begin{aligned} f_{pc,j}(q) = |f_q^* O_j w|^2 = (f_q^* O_j w)^*(f_q^* O_j w) = w^*(O_j^* f_q f_q^* O_j)w \\ = Tr[w^*(O_j^* f_q f_q^* O_j)w] = Tr[(O_j^* f_q f_q^* O_j)(ww^*)] \\ = Tr(\mathcal{I}_{j\mathbf{q}}\rho) \end{aligned}$$

It is a bit like the process of phase-lift.

When w is in pure state(a vector in Hilbert space), $\rho = w^* w$ is a rank-one matirx. In partially coherent case, **we use mixed state to model** w. Fow example, with probability 0.5 in state ψ_1 and 0.5 in ψ_2 (ψ_1 and ψ_2 are not neccesarily orthogonal here). Now w can no longer be represented by a vector(ps. $w \neq p_1\psi_1 + p_2\psi_2$, the latter is still a determined pure state vector).

[1]`https://homepage.univie.ac.at/reinhold.bertlmann/pdfs/T2_Skript_Ch_9corr.pdf`Theorem 9.1. Many symbols in quantum mechanics are included here.

Instead, mixed state is represented by **generalizing the density matrix to one with higher rank**:

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

Easy to find ρ is a positive semi-definite matrix, we can decompose ρ using spectral theorem, with r(rank of ρ) othogonal state w_k:

$$\rho = \sum_{k=1}^{r} w_k w_k^* \tag{1.7}$$

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

And that is exactly (1.5)$f_{pc,j} = \sum_{k=1}^{r} |\mathcal{F}(\mathcal{S}_j u \circ (\omega_k))|^2$. ($f_{pc,j}(q)$ is a single value at frequency q when $f_{pc,j}$ is the whole diffraction image)

We can write it in another quadratic form:

$$\begin{aligned} f_{pc,j}(q) &= \operatorname{Tr} \mathcal{I}_{j\mathbf{q}} \rho = Tr[(O_j^* f_q f_q^* O_j)\rho] = Tr[(O_j^* f_q)^* \rho (O_j^* f_q)] = (O_j^* f_q)^* \rho (O_j^* f_q) \\ &= g_q^* \rho g_q = \sum_{x_1} \sum_{x_2} \overline{g_q(x_1)} \rho(x_1, x_2) g_q(x_2) \end{aligned} \tag{1.8}$$

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

Then it is a discrete version of the model in [9].

1.3.2 The relation between models

In this section we explain how the general model connects with specific models (1.3) and (1.4).

First we consider (1.3). We put the κ_i inside:

$$f_{pc,j} = \sum_i |\mathcal{F}(\mathcal{S}_j u \circ (\sqrt{\kappa_i} \mathcal{T}_i \omega))|^2 = \sum_i |\mathcal{F}(\mathcal{S}_j u \circ (\hat{\omega}_i))|^2 \tag{1.9}$$

Multiple modes $\hat{w}_i$ are produced by shifted w. Then we can construct density matrix and use truncated SVD to get a low-rank approximation.

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

As for (1.4), we introduce the definition for coherence function:

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

In other word:

$$\gamma = \rho./(ww^*), \rho = \gamma \circ (ww^*) \tag{1.10}$$

where ./ means pairwise division.

In case we assume that coherence function$\gamma(\mathbf{x}_1, \mathbf{x}_2)$ only depends on the difference between the two points $\mathbf{x}_1, \mathbf{x}_2$, i.e. $\gamma(\mathbf{x}_1, \mathbf{x}_2) = \gamma(0, \mathbf{x}_2 - \mathbf{x}_1)$, we can write the far-field intensity as a convolution[9] (QU proof unfinished):

$f_{pc} = \kappa * f$. If we know the point spread function $\kappa(q)$, then we get its inverse Fourier transform $\gamma(0, x) = (\mathcal{F}^{-1}\kappa)(x)$. And we get $\gamma(x_1, x_2)$ based on the assumption above. If we also know the phobe w, then we can get ρ, again SVD helps us find the main modes w_k in model (1.5)

With the connection above, we can get the "standard mode decomposition" for simulation experiment which is a good reference.

2 ADMM-based numerical algorithm

The AP algorithm above can be rewritten into ADMM form, which is more stable and faster. We generalize the ADMM form in [8] to mixed states.

Now $w \in \mathbb{C}^{(px \times py) \times r}$ is a phobe with r mixed states. $u \in \mathbb{C}^{Nx \times Ny}$ is an image. $f \in \mathbb{C}^{(px \times py) \times N}$ is the true(observed) diffraction image stacks. Let $Y = \sqrt{f}$ be the amplitute of stacks.

An auxiliary variable $z = \mathcal{A}(\omega, u) \in \mathbb{C}^{(px \times py) \times N \times r}$ is introduced. $\mathcal{A}$ is an operator generating diffraction image stacks(N frames for each phobe state) from image u and r different states $w_k := w(:,:,k) \in \mathbb{C}^{px \times py}$. For multi-dimensional vectors, symbol : denotes the free dimensions, and we can fix some indexes to extract particular dimensions from original vectors.

Based on the general model1.3, the problem is:

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

Where $\mathcal{A} : \mathbb{C}^{(px \times py) \times r} \times \mathbb{C}^{Nx \times Ny} \to {}^{(px \times py) \times N \times r}$,$\mathcal{A}_j : \mathbb{C}^{px \times py} \times \mathbb{C}^{Nx \times Ny} \to \mathbb{C}^{px \times py}$, $\bar{\mathcal{A}}_j : \mathbb{C}^{(px \times py) \times r} \times \mathbb{C}^{Nx \times Ny} \to \mathbb{R}_+^{(px \times py)}$, and $\bar{\mathcal{A}} : \mathbb{C}^{(px \times py) \times r} \times \mathbb{C}^{Nx \times Ny} \to \mathbb{R}_+^{(px \times py) \times r}(\forall 0 \leq j \leq N-1)$ are denoted as follows:

$z(:,:,j,k) = \mathcal{A}_j(\omega_k, u) := \mathcal{F}(\omega_k \circ \mathcal{S}_j u) \in \mathbb{C}^{px \times py}$,

$z(:,:,:,k) = \left(\mathcal{A}_0^T(\omega_k, u), \mathcal{A}_1^T(\omega_k, u), \ldots, \mathcal{A}_{N-1}^T(\omega_k, u)\right)^T \in \mathbb{C}^{(px \times py) \times N}$,

$z(:,:,j,:) = \left(\mathcal{A}_j^T(\omega_1, u), \mathcal{A}_j^T(\omega_2, u), \ldots, \mathcal{A}_j^T(\omega_r, u)\right)^T \in \mathbb{C}^{(px \times py) \times r}$.

,

$\bar{\mathcal{A}}_j(\omega, u) := \sum_{k=1}^r |\mathcal{A}_j(\omega_k, u)|^2 \in \mathbb{R}_+^{px \times py}$,

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

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

Then $\mathcal{G}(z) = ||\sqrt{\sum_{k=1}^r |z(:,:,:,k)|^2} - Y||^2$ measures the difference between values computed by my our model and the groundtruth.

Let $\mathcal{X}_1$ and $\mathcal{X}_2$ be the prior range for w and u. Let $l = px \times py$, $\mathcal{X}_3$ be the index function for orthogonal $D\alpha \in \mathbb{C}^{l\times r}$ (Orthonormal $D \in \mathbb{C}^{l\times r}$ s.t. $D^*D = I$, and amplitude factors $\alpha \in \mathbb{R}^{r\times r}$). Ω is a reformulation operator, and $\Omega(D\alpha) := reshape(D\alpha, [px, py, r]) \in \mathbb{C}^{(px\times py)\times r}$. And sometimes we simplify $\Omega(D^k\alpha^k)$ to Ω^k and $\Omega(D\alpha)$ to Ω.

Then we get the following:

$$\begin{aligned}&\min_{\omega,u,z} \mathcal{G}(z) + \mathbb{I}_{\mathcal{X}_1}(\omega) + \mathbb{I}_{\mathcal{X}_2}(u) + \mathbb{I}_{\mathcal{X}_3}(D\alpha)\\ &s.t. \quad z - \mathcal{A}(\omega, u) = 0, \quad \Omega(D\alpha) - w = 0.\end{aligned} \tag{2.1}$$

The corresponding augmented Lagrangian reads

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

where $\Lambda, \Lambda_2 \in \mathbb{C}^{(px\times py)\times N\times r}$ is the multiplier. Let $\Lambda = \Lambda/\beta, \Lambda_2 = \Lambda_2/\beta_2$, we can combine the $\Re$ part and the augmented part to get:

$$\begin{aligned}\Upsilon_\beta(\omega, u, z, \Lambda) :=& \mathcal{G}(z) + \mathbb{I}_{\mathcal{X}_1}(\omega) + \mathbb{I}_{\mathcal{X}_2}(u) + \mathbb{I}_{\mathcal{X}_3}(D\alpha)\\ &+\frac{\beta}{2}\|z - \mathcal{A}(\omega, u) + \Lambda\|^2 - \frac{\beta}{2}||\Lambda||^2 + \frac{\beta_2}{2}\|\Omega(D\alpha) - w + \Lambda_2\|^2 - \frac{\beta_2}{2}||\Lambda_2||^2\end{aligned} \tag{2.2}$$

In ADMM, one seeks a saddle point of the following problem:

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

A natural scheme to solve the above saddle point problem is to split them, which consists of four-step iterations for the generalized ADMM (only the subproblems w.r.t. ω or u have proximal terms), as follows:

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

$$\text{Step 6: } \Lambda^{k+1} = \Lambda^k + \left(z^{k+1} - \mathcal{A}\left(\omega^{k+1}, u^{k+1}\right)\right) \tag{2.3}$$

$$\text{Step 7: } \Lambda_2^{k+1} = \Lambda_2^k + (\Omega(D^{k+1}\alpha^{k+1}) - \omega^{k+1}) \tag{2.4}$$

For simplicity, we ignore the stable quadratic terms in Step1 and Step2 in the following analysis.

2.1 Subproblems w and u

w.r.t. the probe ω :

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

with $\hat{z}^k := z^k+\Lambda^k, \hat{\Omega}^k := \Omega^k+\Lambda_2^k$

The close form solution of Step 1 is given as(details are in the Appendix 4.1)

$$\omega^{k+1} = \mathrm{Proj}\left(\frac{\beta\sum_j\left(\mathcal{S}_j u^k\right)^*\circ\left[\left(\mathcal{F}^{-1}\hat{z}^k\right)(:,:,j,:)\right]+\beta_2\hat{\Omega}^k}{\beta\sum_j\left|\mathcal{S}_j u^k\right|^2+\beta_2};\mathcal{X}_1\right) \tag{2.5}$$

with the projection operator onto $\mathcal{X}_1$ defined as $\mathrm{Proj}\,(\omega;\mathcal{X}_1) := \mathcal{F}^{-1}(\mathcal{F}(\omega)\circ C_w)$, where $\mathcal{X}_1 = \{\omega : \mathcal{F}(\omega)$ supports on the index function $C_w\}$. $\mathcal{F}^{-1}$ acts on the first two dimensions of $\hat{z}$ (i.e. $\hat{z}_{j,i} := \hat{z}(:,:,j,i)$) and ω (i.e. $\omega_i := \omega(:,:,i)$).

Similarly we have:

$$u^{k+1} = \mathrm{Proj}\left(\frac{\sum_{j,i}\mathcal{S}_j^T\left(\left(\omega_i^{k+1}\right)^*\circ\mathcal{F}^{-1}\hat{z}_{j,i}^k\right)}{\sum_{j,i}\left(\mathcal{S}_j^T\left|\omega_i^{k+1}\right|^2\right)};\mathcal{X}_2\right)\ . \tag{2.6}$$

Here S_j^T is an operator mapping its augment to target position j in image u.

2.2 Subproblem z

$$z^{k+1} = \arg\min_z \mathcal{G}(z)+\frac{\beta}{2}\left\|z-\mathcal{A}\left(\omega^{k+1},u^{k+1}\right)+\Lambda^k\right\|^2$$
$$= \arg\min_z \frac{1}{2}||\sqrt{\sum_{i=1}^r|z(:,:,:,i)|^2}-Y||^2+\frac{\beta}{2}\left\|z-z^+\right\|^2$$
$$= \arg\min_z \sum_{x,y,j}[\frac{1}{2}(\sqrt{\sum_{i=1}^r|z(x,y,j,i)|^2}-Y(x,y,j))^2+\frac{\beta}{2}||z(x,y,j,:)-z^+(x,y,j,:)||^2]$$

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

The close form solution of Step 3 is given as(details are in the Appendix 4.1):

$$z_i^{k+1} = \frac{z_i^k\dfrac{Y}{M^k}+\beta z_i^+}{1+\beta}, 1\le i\le r \tag{2.7}$$

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

2.3 Subproblem D and α

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

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

The close form solution for D^{k+1} is(details are in the Appendix 4.1):

$$D^{k+1} = UV^* \tag{2.8}$$

The updation of α is easier:

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

Notice that each $\alpha_i \in \mathbb{R}$ can be solved independently, the first optimality condition gives:

$$\alpha_i^{k+1} = \sum_{i_0} \Re[\overline{D^{k+1}(i_0, i)}\hat{w}^{k+1}(i_0, i)](1 \le i \le r) \tag{2.9}$$

Algorithm 1: ADMM for general mixed-state model(1.5)

Initialization: Set the number of states r,
$\omega^0, u^0, z^0 = \mathcal{A}\left(\omega^0, u^0\right), \Lambda^0, \Lambda_2^0 = 0$;
D^0 and α^0 from SVD on ω^0
maximum iteration number Iter $_{\mathrm{Max}}$, and parameter β,β_2
Output: $u^\star := u^{Iter_{Max}}$ and $\omega^\star := \omega^{\mathrm{Iter}_{Max}}$
for $ii = 0$ *to* $Iter_{Max} - 1$ **do**
  Compute ω^{k+1} by (2.5) with $\hat{z}^k := z^k + \Lambda^k$;
  Compute u^{k+1} (2.6). with $\hat{z}^k$ the same as above;
  Compute z_i^{k+1},$1 \le i \le r$ by (4.1). with $z^+ = \mathcal{A}\left(\omega^{k+1}, u^{k+1}\right) - \Lambda^k$;
  Compute D^{k+1} (4.2). ;
  Compute $\alpha_i, 1 \le i \le r$ (2.9). ;
  Update the multiplier as Step 6 and Step 7 of (2.3) and (2.4);
end

3 Numerical experiments

The codes are implemented in MATLAB. First, we introduce the setting in experiments, then we conduct experiments on simulation data generated from specific models in 1.2. In each experiment, we compare the reconstructed images and modes in a different setting. We also generate ideal results as in 1.3.2 and compare ours with them.

3.1 Experiment setting

3.1.1 Parameters

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), 2(hexagonal),3(randomly disturb on 2)
blurFlag	types of partially coherent effect	1(1.4),2(1.3)

In order to deal with nontrivial ambiguities, non-periodical lattice-based scanning can be considered experimentally to remove the periodicity of the scanning geometry, e.g., adding a small number of random offsets to a set of the lattice. ($gridFlag = 3$). Or we can add prior knowledge about masks as an additional constraint, e.g. restricting the masks(frequency domain) in a circle.

$Dist$ is an important parameter for successful reconstruction. Generally speaking, the smaller the $Dist$, the more the overlapping area and redundancy in data, we can get reconstruction images with higher qualities. Here we find $Dist = 8$ is enough while $Dist = 16$ always fails.

3.1.2 Performance metrics

In order to evaluate the performances of algorithms, we introduce 3 metrics.

1. Relative error err and signal-to-noise ratio snr

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

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

$||\cdot||_F$ is the Frobenius norm. err measures the difference between a reconstructed image and the groundtruth image. c is an estimated scale factor, and sum means the sum of all elements in the target matrix.

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 measures the difference between the reconstruction diffraction stacks and groundtruth stacks Y. We don't always know u_{true}, Y is the only input data for our algorithm, and R-factor can be used to verify the convergence.

3. Masks approximation error 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 measures the difference between the reconstructed density matrix ρ^k from masks and the standard density matrix ρ_{true} from the theoretical model.

3.1.3 Operations on modes

1. Initialization

   In the following tests, we set $u^0 = \mathbf{1}_{N_x \times N_y}$ and initial phobe $m = \frac{1}{N}\mathcal{F}^{-1}\left(\sum_j Y(:,:,j)\right)$. We generate other modes by randomly disturb intial mask m. r initial probes were created by multiplying the initial mask by different arrays of random complex values (modulus part varies within the range of $[0,1]$ and phase part $[0,2\pi]$). Then we get w^0.

2. Orthogonalization

   Modes w_k computed in our algorithm are not always orthogonal. However, we can easily orthogonalize them by generating a density matrix ρ and perform spectral decomposition on ρ like in (1.7). The orthogonal representation of modes is always unique(Specifically, when the eigenvalues of ρ are all different, and this always happens in real-world data).

   Notice that the number of modes is always small i.e. $r \ll l := p_x \times p_y$, we perform SVD on the $l \times r$ phobe matrix directly instead of $l \times l$ density matrix ρ. And we can also select the first $r' \le r$ modes for an approximation.

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

1 phobe matrix $ss = reshape(w, [l\ r])$;
2 $[U, S, V] = svd(ss, 'econ')$;
3 $q = U(:, 1:r') * S(1:r', 1:r')$;
4 $w_{ort} = reshape(q, [px\ py\ r'])$;

   Orthogonalization operation is always performed before we display final modes to get a clearer representation. We also want to figure out whether orthogonalization can be added to improve our Algorithm1.

3. Compression

   In noisy case, we consider extracting fewer main modes to avoid the distortion from noise.

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

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

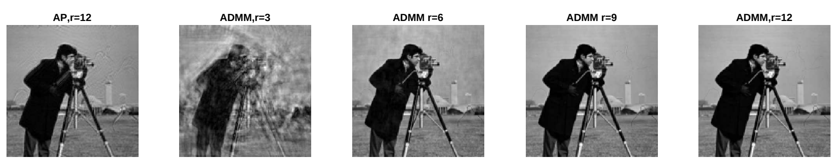
$q = U(:, 1 : r') * S(1 : r', 1 : r') * V(:, 1 : r')$;

$w_{com} = reshape(q, [px\ py\ r])$;

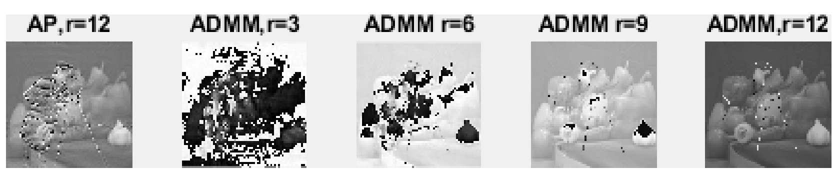
3.2 Approximation by modes

$Dist = 8$, $gridFlag = 1$, $blurFlag = 2$ and κ is a guassian kernel with $\sigma = (15, 15)$. $\beta = 0.05$ is chosen as algorithm parameter for ADMM, and orthogonalization constraint is not considered here so $\beta_2 = 0$.

Figure 1



(a) Amplitude



(b) Phase

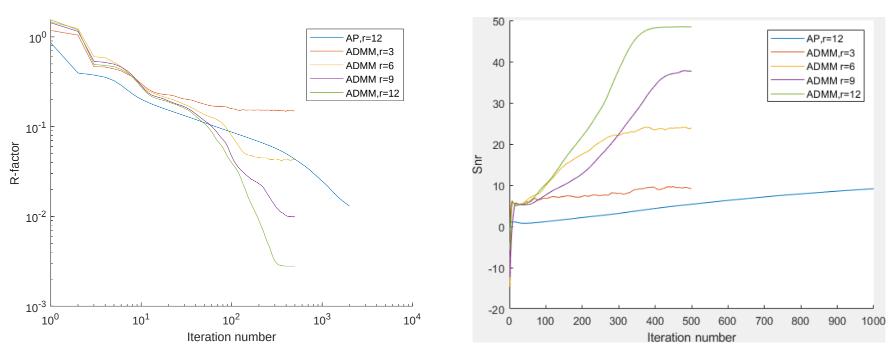


Figure 2: R and snr.

We first compare snr and R for reconstructed images using a different number of modes. When the number of modes increases, the $R - factor$ decreases, and the snr increases. That indicates that the quality of the reconstructed image increases.

The R for 3,6,9,12 modes using ADMM algorithm are stable after 500 iterations at 0.15, 0.042, 0.0099, 0.0028, respectively, when R for AP does not converge after 2000 iterations.

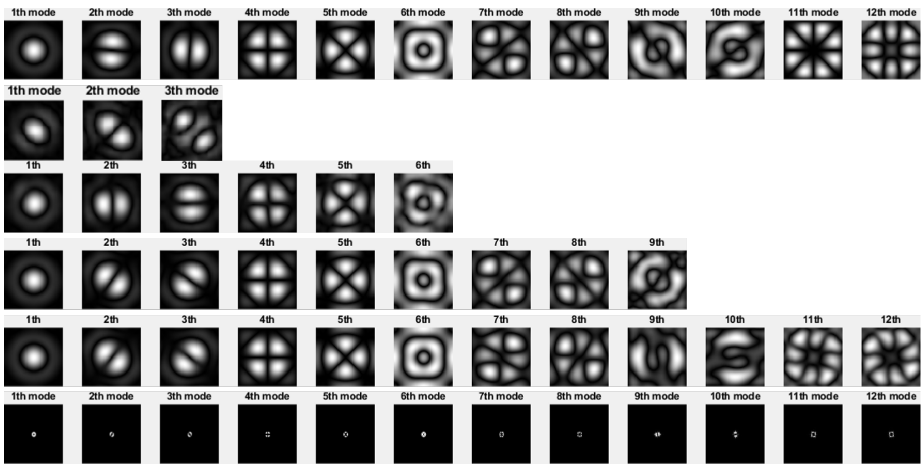


Figure 3: Mode pattern. The first row represents the standard mode pattern. And the last two rows represent the mode pattern for 12 modes. Mode patterns are in the time domain except that the last row is in the frequency domain.

Standard mode pattern is obtained through performing SVD on standard density matrix generated from the model and extracting the first 12 modes. As shown in Figure 3, our algorithm can generally catch the main modes and get an optimal approximation.

Further, denote err_M for r modes as err_M^r. Optimal $err_M^{r,*}$ can be calculated from the theory of low-rank approximation, and we compare our err_M^r with it. Denote the singular values of standard density matrix ρ_{true} as $s_i, i = 1...d = rank(\rho_{true})$

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

,

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

As shown in Figure 5, the err_M^* reaches around 0.01, and err_M is close to err_M^* with 12 modes, which indicates that our algorithm does well in low-rank approximation to standard density matrix.

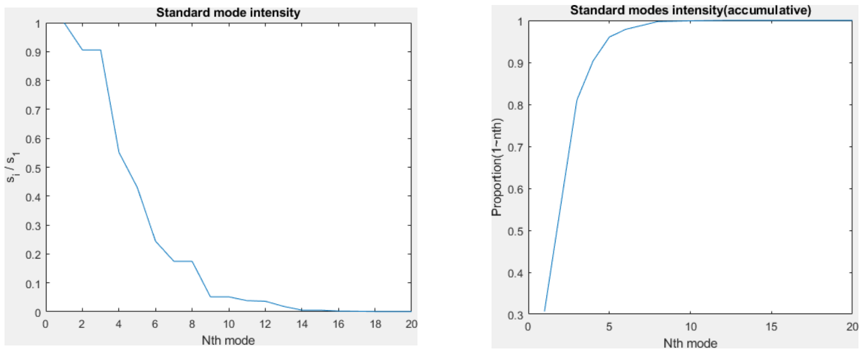


Figure 4: The distribution of singular values of the standard density matrix. The vertical axis in the left subfigure represents the ratio of i^{th} largest singular to the first one s_i/s_1, and that in the right one represents $S_{cum}(i)$. The singular value decreases exponentially and the matrix is approximately low-rank.

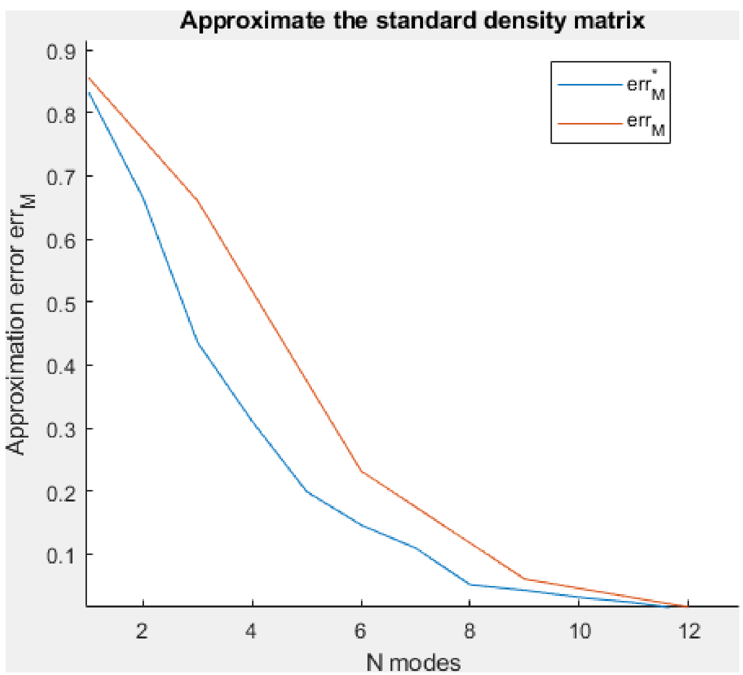


Figure 5

3.3 Add orthogonalization constraint

The experiment setting is the same as above except that we change $ratio = \beta_2\beta$ to introduce orthogonalization constraint in ADMM. The larger the β_2, the stricter the orthogonalization constraint. As a reference, we also tried perform-

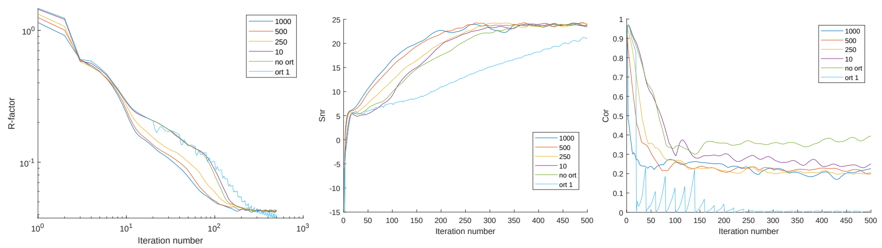


Figure 6: The vertical axis and horizontal axis are set in log-scale in the first subfigure for R-factor. The blue line represents $\beta_2/\beta = 1000$ and works the best in this case. The green line represents the result without orthogonalization. 'ort 1' means performing orthogonalization every 20 iterations.

ing orthogonalization as in Algorithm 2 every 20 iterations.

The reconstructed images are similar after 500 iterations, while *snr* and $Rfactor$ using the ADMM algorithm with orthogonalization constraint improve faster. And the degree of correlation *coherence* between modes also decreases faster with larger β_2.

Remark.

1. We suppose that orthogonalization constraints can improve robustness on different initial values.

2. When the number of modes is large enough(like 12 here), the effect of the orthogonalization constraint seems not noticeable.

3.4 Noisy case

$Dist = 8$, $gridFlag = 3$, $blurFlag = 2$ and κ is a guassian kernel with $\sigma = (15, 15)$, $\beta = 0.05$.

Poisson noise is added on diffraction images Y through MATLAB command:

$$Y_{noise} = poissrnd(Y * (\eta))/\eta;$$

where $\eta = 0.0675$ is used here.

Four different settings are compared: 9 modes with noise, 9 modes with noise, and 6 are kept after compression, 6 modes with noise, and 6 modes without noise(standard). Specifically, we conducted compression as in 3 every 20 iterations, and kept 6 modes each time.

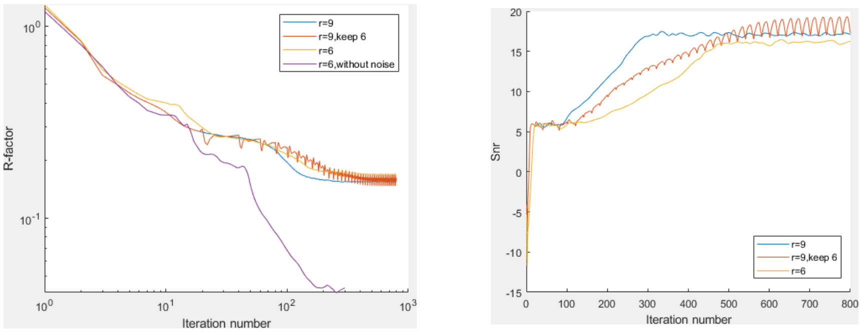


Figure 8: R and snr. The red line represents the result with compression.

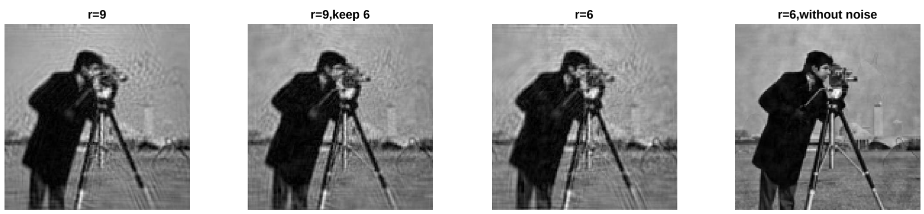


Figure 7: Reconstructed images in the noisy case

The result periodically fluctuates with compression operation, while it can finally behave the best in the noisy case. Although the difference is not noticeable, we can consider adding compression in constraint instead of performing the direct truncated operation.

4 Appendix

4.1 Subproblems in ADMM

4.1.1 ω

Essentially in this subproblem, each state $\omega_i = w(:,:,i)$ is independent. Then we can optimize each w_i seperately.

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

Essentially in this subsubproblem, each element in w_i is independent. such that one just needs to solve the following 1D constraint quadratic problem:

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

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

The derivative of $\rho_t^k(x)$ is calculated [2] as

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

The first order optimality condition is $\nabla \rho_t^k(x) = 0$. Then the close form solution of w_i is given as

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

4.1.2 z

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

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

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

For any fixed x, y, j and free i, the problem can be seen as:

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

Notice that for fixed $||z_{x,y,j}||$, the first term in expression is fixed. To optimize the second term, we should always choose $z_{x,y,j}$ with the same direction as $z_{x,y,j}^+$. So we have $||z_{x,y,j} - z_{x,y,j}^+||^2 = (||z_{x,y,j}|| - ||z_{x,y,j}^+||)^2$

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

To determine $z_{x,y,j}$, we only need to determine $||z_{x,y,j}||$. Denote it as a.

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

[2]Notice that here ρ is a real value function with complex variable, and we use wirtinger derivatives here. More properties and calculation rules are listed in this link: `https://blog.csdn.net/weixin_37872766/article/details/107673096`

The first optimality condition easily gives:

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

The close form solution of Step 3 is given as:

$$z_i^{k+1} = \frac{z_i^k \dfrac{Y}{M^k} + \beta z_i^+}{1+\beta}, 1 \le i \le r \tag{4.1}$$

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

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

This is a special case in Orthogonal Procrustes problem [3]

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

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

Consider the SVD decomposition: $\hat{\omega}^{k+1}\alpha^k = USV^*$

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

We can easily see $\hat{D} = I$ is optimal, and:

$$D^{k+1} = UV^* \tag{4.2}$$

References

[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.

[3] `https://en.wikipedia.org/wiki/Orthogonal_Procrustes_problem`

[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] Multiplexed coded illumination for Fourier Ptychography with an LED array microscope.

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

[6] Introduction to Quantum Mechanics, David J. Griffiths, 12.3

[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.