EE368 Term Project: Incorporating low-resolution image into phase retrieval process

Po-Nan Li
liponan@stanford.edu

Yen-Kai Huang
nykh@stanford.edu

December 6, 2016

1 Introduction

Coherent diffraction imaging (CDI) a lensless microscopic technique that measures an object's diffraction pattern instead of direct image [1]. Usually combined with X-ray light soutce, CDI has been widely used for investigating non-periodic structures of biological cells or nanoparticles. While the resulting diffraction pattern provides the intensity of Fourier transform (FT) of the original image, the image's real-space version is generally inaccessible unless both its spectral amplitude (square root of the intensity) and phase components are available for performing inverse Fourier transform (IFT). This is the motivation behind the phase retrieval problem.

Several phase retrieval algorithms have been developed to extract the phase component from oversampled intensity component [2], the most well-known of which being the Hybrid Input-Output (HIO) algorithm [3]. HIO starts with a set of random phases as initial guess, then applies IFT and FT to the image to move it back and forth between reciprocal and real spaces, where constraints such as finite support and Fourier amplitude are applied to refine the calculated image, as shown in Figure 1(a).

In general HIO can give reasonable reconstruction result, but it usually takes a few thousands of cycles to converge, and sometimes it stagnates at a suboptimal result. In addition, phasing algorithms like HIO are generally vulnerable to data defects such as quantum noise (for example due to limited photon flux) and missing intensities (for example due to low frequency components being blocked). For the reasons above, performing phase retrieval is generally computationally expensive, because people would run with large iteration numbers to reach optimal reconstruction result. While a low-resolution image from other microscope sources such as an optical microscope or scanning electron microscope (SEM) is often used to validate reconstruction results, here we propose that we incorporate it, i.e. low-resolution image, into the phase retrieval algorithm as a priori to guide the phasing process. We expect to see the a priori can (i) enhance reconstruction fidelity, (ii) increase convergence rate, and (iii)

This project aims to improve upon the HIO algorithm by incorporating a low resolution image as a stronger prior than the finite support constraint used by most algorithms, developing a phase retrieval algorithm with a higher reconstruction fidelity and convergence rate. Figure 1(b) shows the pipeline we have developed. Given a reasonably quality of the LR prior, the modified algorithm in general outperforms the conventional HIO, as the LR image contains not only the support boundary but also local intensities information.

The main challenge of the project is to find a mathematically reasonable and computationally practical way to constrain computed real space images based on the LR image. Previously we have experimented with a naïve technique where the on-the-flight real space image is dynamically guided by the LR image with a simple linear combination [4]. For this project, We researched into different methods for optimizing an image to be closer to the given LR image in a dynamic fashion

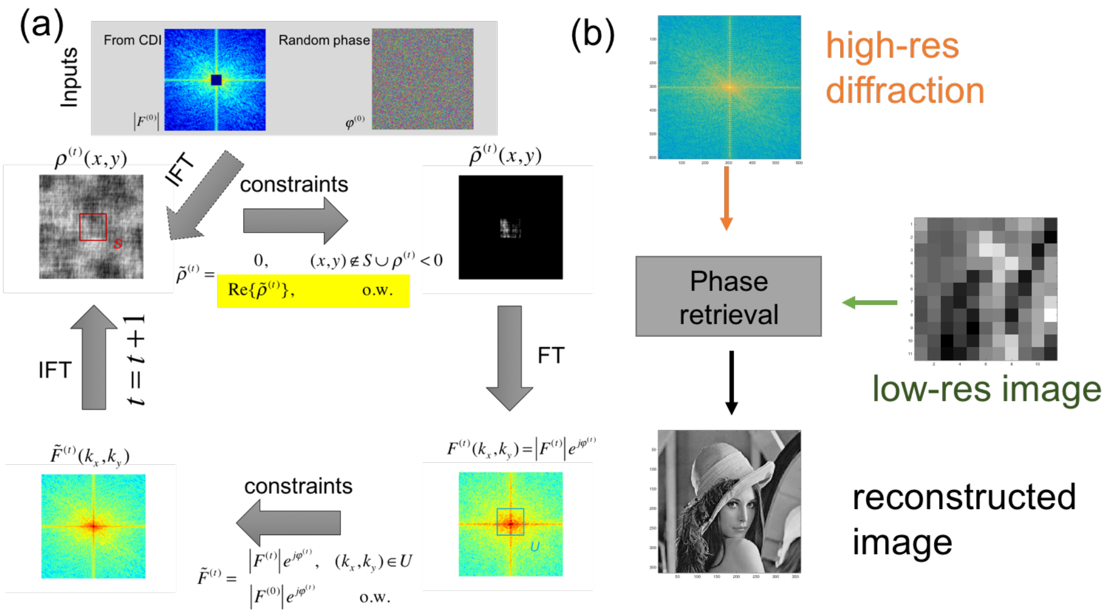

Figure 1: (a) Schematic of the conventional HIO algorithm. The highlighted part indicates the constraint to be studied and improved. (b) Proposed image processing pipeline.

which can be tuned for convergence, and found indeed the alternating direction methods (ADMs) can be employed to implement phase retrieval algorithms, i.e. HIO [5]. In the following sections, we will briefly review how ADM can be used for phase retrieval, and further develop an ADM approach that employs a LR image to enhance the reconstruction fidelity.

2 Theory background

2.1 HIO

HIO algorithm was proposed by Fienup [3] to solve a problem of ER algorithm, where ER converge very fast but often converges to suboptimal solutions. HIO algorithm introduces a step length β into the update of the input image $g(x)$ based on the desired change in the output image $g'(x)$.

$$g_{k+1}(x) = \begin{cases} g'_k(x) & x \notin \gamma \\ g_k(x) - \beta g'_k(x) & x \in \gamma \end{cases}$$

Where γ denotes the set of points where $g'(x)$ violates constraints and thus requires a change. This utilizes the special property of the nonlinear system described above that when the input is set to be the current output ($g_{k+1}(x) = g'_k(x)$) no change will occur. In the paper, the HIO algorithm was shown to converge much faster than ER algorithm. However, the author agrees there was still some trial and error involved to find the best algorithm for different reconstruction purposes.

2.2 ADM

2.2.1 ADM for phase retrieval

Ref. [5] provides an elegant way to formulate the phase retrieval problem in the fashion of optimization. Essentially, the real-space and Fourier-space constraints in the conventional HIO can be formulated as projection operators, i.e.

$$\mathcal{P}_s(x) = \begin{cases} x(r) & r \in S \wedge x(r) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

and

$$\mathcal{P}_M(x) = \mathcal{F}^{-1}\hat{y}, \text{, where } \hat{Y} = \begin{cases} |\hat{x}(k)| \exp j\phi(\hat{x}) & k \in U \\ |b(k)| \exp j\phi(\hat{x}) & \text{otherwise} \end{cases} \tag{2}$$

where $\mathcal{F}$ is the Fourier transform operator, $\hat{x} = \mathcal{F}x$, S denotes the sample support in the real-space, and U denotes the missing center in the diffraction plane.

In order to treat the support and the magnitude constraints equally, now we define our problem as follows:

$$\text{find } x \text{ and } y, \text{, such that } x = y, x \in \mathcal{X} \text{ and } y \in \mathcal{Y}, \tag{3}$$

where $\mathcal{X}$ and $\mathcal{Y}$ are the space of $\mathcal{P}_S$ and $\mathcal{P}_M$, respectively. Now we write down the augmented Lagrangian function for (3)

$$\mathcal{L}(x, y, \lambda) := \lambda^{\mathsf{T}}(x - y) + \frac{1}{2}\|x - y\|^2, \tag{4}$$

where λ is the Lagrangian multiplier. The ADM will now solve the following two subproblems sequentially in each iteration:

$$x^{(t+1)} = \operatorname*{argmin}_{x \in \mathcal{X}} \mathcal{L}(x, y^{(t)}, \lambda^{(t)}) \tag{5}$$

and

$$y^{(t+1)} = \operatorname*{argmin}_{y \in \mathcal{Y}} \mathcal{L}(x^{(t+1)}, y, \lambda^{(t)}), \tag{6}$$

and update update the Lagrangian multiplier λ by

$$\lambda^{(t+1)} = \lambda^{(t)} + \beta(x^{(t+1)} - y^{(t+1)}), \tag{7}$$

where the step length β is usually carefully chosen so the iterative algorithm can find the optima while maximizing convergence rate. Since the L-2 norm in (4) is convex and differentiable, it's not difficult to derive the solutions to (5) and (5), which are

$$x^{(t+1)} = \mathcal{P}_X(y^{(t)} - \lambda^{(t)}) \tag{8}$$

and

$$y^{(t+1)} = \mathcal{P}_Y(x^{(t+1)} + \lambda^{(t)}). \tag{9}$$

With (8), (9) and (7), now we are able the implement HIO by using ADM. The results are presented in Sec. 3.

2.2.2 ADM-HIO with LR prior

In this project we aim to incorporate an LR image into the HIO process, so our objective function is now

$$\mathcal{L}(x, y, \lambda) := \lambda^{\mathsf{T}}(x - y) + \frac{1}{2}\|b - y\|^2, \tag{10}$$

where b is the LR image. So far, we haven't had a way to derive the optimization rigorously, but the following update equations happen to work perfectly:

$$\begin{aligned} x^{(t+1)} &= b - \lambda^{(t)} \\ y^{(t+1)} &= \mathcal{P}_{\mathcal{Y}}(x^{(t+1)} + \lambda^{(t)}) \\ \lambda^{(t+1)} &= \lambda^{(t)} + \beta(x^{(t+1)} - y^{(t+1)}). \end{aligned} \tag{11}$$

Here, what we just did for (11) is replacing the $y^{(t)}$ term in the first update equation with the LR image b, and removing the $\mathcal{P}_X$ projection as we believe the LR image is a stronger constraint than the binary support mask. One of our todos would be to modify (10) to make it make sense.

2.2.3 ADM-HIO with LR and TV-regularization

Furthermore, we would like to add a total variation (TV) regularization time in the objective function to ensure reasonable solution. The TV-regularized function might look like

$$\mathcal{L}(x, y, \lambda) := \lambda^{\mathsf{T}}(x - y) + \frac{1}{2}\|b - y\|^2 + \rho\Gamma(x), \tag{12}$$

where ρ is a constant deciding the strength of the TV-regularization. So far, it's still unclear to us that whether (10) and (12) make sense. Also, we are now having hard time implementing the TV-regularization term, which is indifferentiable.

3 Results

In the setting where the model, i.e. ideal image is known, an error function for real-space

$$E_R = \frac{\sum_{(x,y)\in S} |\rho_{calc}^{(t)} - \rho_{idea}|}{\sum_{(x,y)\in S} |\rho_{idea}|}, \tag{13}$$

where S denotes the sample support and t is the iteration number, is usually useful for comparing the on-the-flight or final reconstructed image with the model.

As a preliminary result, Figure 2 (c) shows the E_R evolution of reconstruction by using four different phase retrieval algorithms: conventional HIO by Fienup [3], HIO implemented by ADM [5], image-assisted HIO by one of the author [4], and the present method. The first two algorithms only employ the intensity of Fourier transform of the image, shown in Figure 2(a), while the other two further employ a low-resolution version of the target image, shown in Figure 2(b). It's unambiguous that our methods outperforms all others, as it quickly converges in first few steps while achieving a super low error.

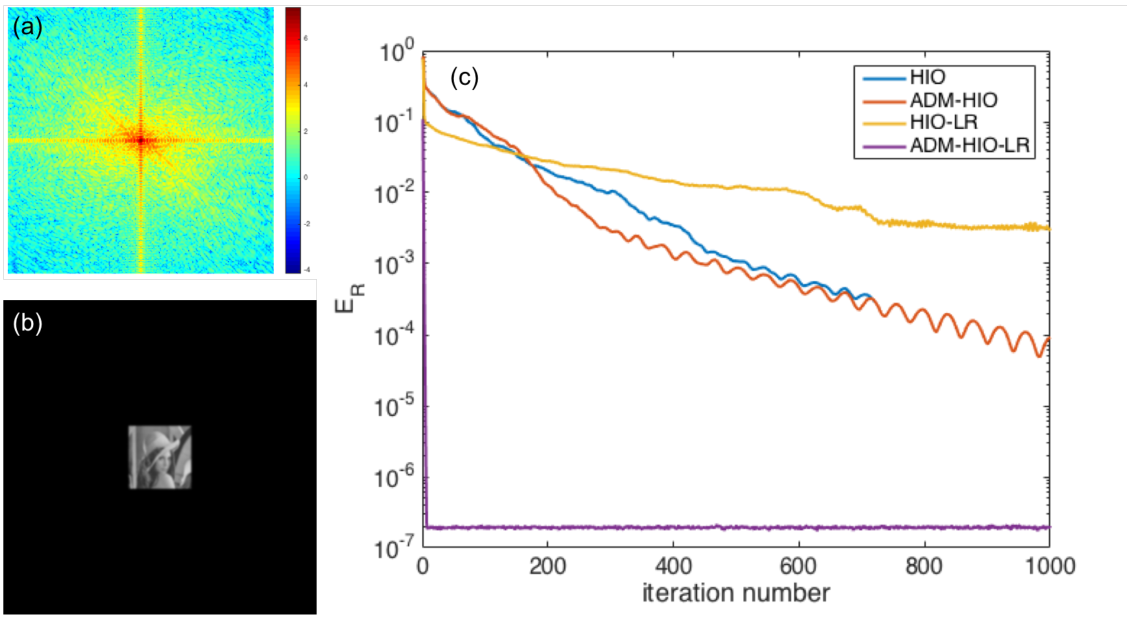

Figure 2: (a) Input to all four algorithms: intensity of Fourier transform of the target image. (b) Additional input for HIO-LR [4] and this ADM-LR (this work): a low-resolution version image. (c) E_R evolution through iterations. Our method (purple line) outperforms other three methods by its quick convergence and achieving super-low error.

4 Discussion

5 Conclusion

References

[1] J. Miao et al., *Nature* **400**, 342 (1999).

[2] J. Miao et al., *Phys. Rev. Lett.* **95**, 085503 (2005).

[3] R. Fienup, *Appl. Opt.* **21**, 2758 (1982).

[4] P.-N. Li et al. (under preparation).

[5] Z. Wen, C. Yang, X. Liu, and Stefano Marchesini, *Inverse Prob.* **28**, 115010 (2012).