Iterative Algorithms for Ptychography

Wenjie Mei

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

The University of Sheffield
Faculty of Engineering
Department of Electronic and Electrical Engineering

February 2024

Abstract

In computational imaging, ptychography is a cutting-edge technique that has garnered significant attention for its ability to address the challenge of the phase problem in the microscope. At its core, ptychography involves breaking the imaging process into a series of overlapping measurements. Instead of capturing the entire image at once, ptychography acquires diffraction patterns from overlapping regions of the specimen. This approach allows for the recovery of both amplitude and phase information with high accuracy. With the development of ptychography, various phase retrieval algorithms have been proposed in the recent decades. These different algorithms can be categorized as direct ptychography and iterative ptychography. Furthermore, iterative ptychography can also be subdivided into two types: sequential projection methods and set projection methods. All of these different categories will be discussed and analyzed in this thesis. The direct ptychography will be introduced in the Chapter 4, while the iterative ptychography will be in the Chapter 5. Further contributions in this thesis are two new blind ptychographic solutions. The first is generalizing the set projection methods to a standard form and introducing the Bayesian Optimization to tune the parameters automatically. The other is a novel approach called Weighted Average of Sequential Projections (WASP), which combines the advantages of both sequential projection methods and set projection methods. This thesis aims to find novel and effective ptychographic approaches, based on evaluating and comparing the existing algorithms. All the different approaches will be tested with different simulations and real-world ptychography experiments to verify their performance in different contexts and provide a deep understanding of different types of ptychographic solutions for future research.

Acknowledgements

Time flies, and my doctoral journey is drawing to a close. These four years have been filled with numerous beautiful moments worth cherishing. I feel fortunate to have encountered so many great people during this meaningful time of my life. I believe these experiences will propel me toward a brighter future.

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr Andrew Maiden. His guidance, encouragement, advice and rigorous attitude have deeply influenced me and are indispensable for both my academic and personal growth. The successful completion of my thesis owes much to his assistance. Also, I would like to thank my second supervisor, Dr Charith Abhayaratne, for his helpful advice and support.

Furthermore, I would like to give my special gratitude to Professor John Rodenburg for leading me into this field of research and giving me the opportunity to join this amazing group. His assistance and guidance are also important for my scientific research.

I am also very grateful to all my colleagues, Dr Shengbo You, Dr Yiqian Zhang, Ziyang Hu, Dr Zhuoqun Zhang, Dr Yangyang Mu and Dr Frederick Allars, who created an excellent research environment in the office. I am so lucky to meet them in Sheffield, and I appreciate the camaraderie and care we shared, making these years full of sunshine.

Of course, I must also express my heartfelt gratitude to my parents, Jufang Li and Jianxin Mei. Throughout these years away from home, they have steadfastly supported me, respected my decisions, and guarded my dreams. I could not come this far without their dedication and unconditional love. Additionally, I'd like to give a special thanks to my aunt, Weifang Li, whose support and encouragement are also important.

Last but not least, I would like to express my sincere gratitude to my lovely wife, Lu Zou. During my doctoral journey, her unwavering love, support, and understanding have been my anchor. Her encouragement and sacrifices have been a constant source of strength, enabling me to persevere and pursue my academic goals. Her belief in me inspired my determination to succeed even in the toughest times. I am profoundly grateful for her presence in my life, and I cherish her companionship on this remarkable journey. This thesis is dedicated to her and our newborn boy, Cadence.

Contents:

1. Introduction

Microscope imaging has been a cornerstone in advancing our understanding of the intricate world at the smallest scales, unravelling the mysteries of the unseen world. There are various imaging techniques for the microscope, one of which is Coherent Diffraction Imaging (CDI) [1], which offers a unique perspective on the interaction of light with matter. Unlike conventional imaging methods, CDI harnesses the coherent nature of light or other waves, enabling the reconstruction of complex structures without the need for lenses. At its essence, CDI involves the measurement and analysis of the diffraction pattern produced when a coherent wave interacts with a sample. By carefully capturing and processing this diffraction information, the high-resolution images of the sample can be computationally reconstructed without conventional lenses. This capability makes CDI particularly valuable in fields such as materials science, biology, and nanotechnology, where traditional imaging methods face limitations [2-5].

In CDI, computational imaging plays a pivotal role in the reconstruction of an image from diffraction patterns. One of the novel computational imaging techniques is called ptychography, which surpasses traditional microscopy by providing detailed reconstructions even in the presence of aberrations and limitations imposed by the optics [6, 7]. Ptychography requires a specific setup which moves either the specimen or the probe to produce partial and overlapping diffraction patterns [8]. By acquiring diffraction patterns from overlapping regions, ptychography not only enhances the resolution but also enables the recovery of complex sample structures [7]. This technique has been used in microscopes with a wide range of wavelengths, including visible light, X-ray and electron. With this great potential for different scientific fields, a wide variety of different algorithms have emerged in ptychography to improve the quality of its reconstruction. The first one is non-iterative ptychography,

requiring a dense scan and the diffraction patterns for every pixel, resulting in heavy computation and a massive dataset [8-10]. Later, a pioneering method called Ptychographical Iterative Engine (PIE) was introduced by Rodenburg and Faulkner, revolutionizing the field by eliminating the need to record diffraction patterns for every reconstructed pixel [6]. Instead, PIE utilized a large illumination probe, scanning the sample through a grid of positions, significantly reducing data requirements. However, PIE encountered a crucial limitation – the necessity for an accurate model of the illuminating probe wavefront. A conjugate gradient algorithm proposed by Guizar-Sicairos and Fienup broke this limitation. As a nonlinear optimization algorithm, it can directly incorporate any form of non-ideality, such as inaccurate knowledge of the probe [11]. Then the idea of set projection started to be used in ptychography; first one is the difference map (DM) [12]. Afterwards, Relaxed Averaged Alternating Reflections (RAAR), the Alternating Directions Method of Multipliers (ADMM), proximal algorithms, and maximum likelihood via least-squares (LSQ-ML) were successively proposed [13-16]. Meanwhile, Maiden and Rodenburg extended the PIE to address the recovery of the probe, giving rise to the ePIE [17]. Moreover, further improvements based on ePIE were achieved by introducing regularization and momentum, resulting in rPIE and mPIE [18]. The continuous advancements in computational algorithms further underscore the growing importance of ptychography in pushing the boundaries of microscopic imaging, opening new avenues for discovery and innovation.

2. Background

In this chapter, I will give an overview of the computational basis of diffractive imaging and theoretical tools needed to understand the physical aspects of diffractive imaging methods. The definition and properties of the Fourier transform, and the wave propagation theory will be introduced at the beginning, leads to the phase problem and the single-shot phase retrieval method as well as the concept of novel phase retrieval method called ptychography.

2.1. Microscope Theory

A microscope is a tool to help us visually see a very small object, over many years in the past, the strategy to see a smaller object is to make one or more lenses as accurately as possible, arrange them and observe through them to see the object [19, 20]. However, this is very demanding on the quality of lenses. Nowadays, new strategies have opened up with the development of computers. Several imaging strategies are illustrated in Figure 2.1.

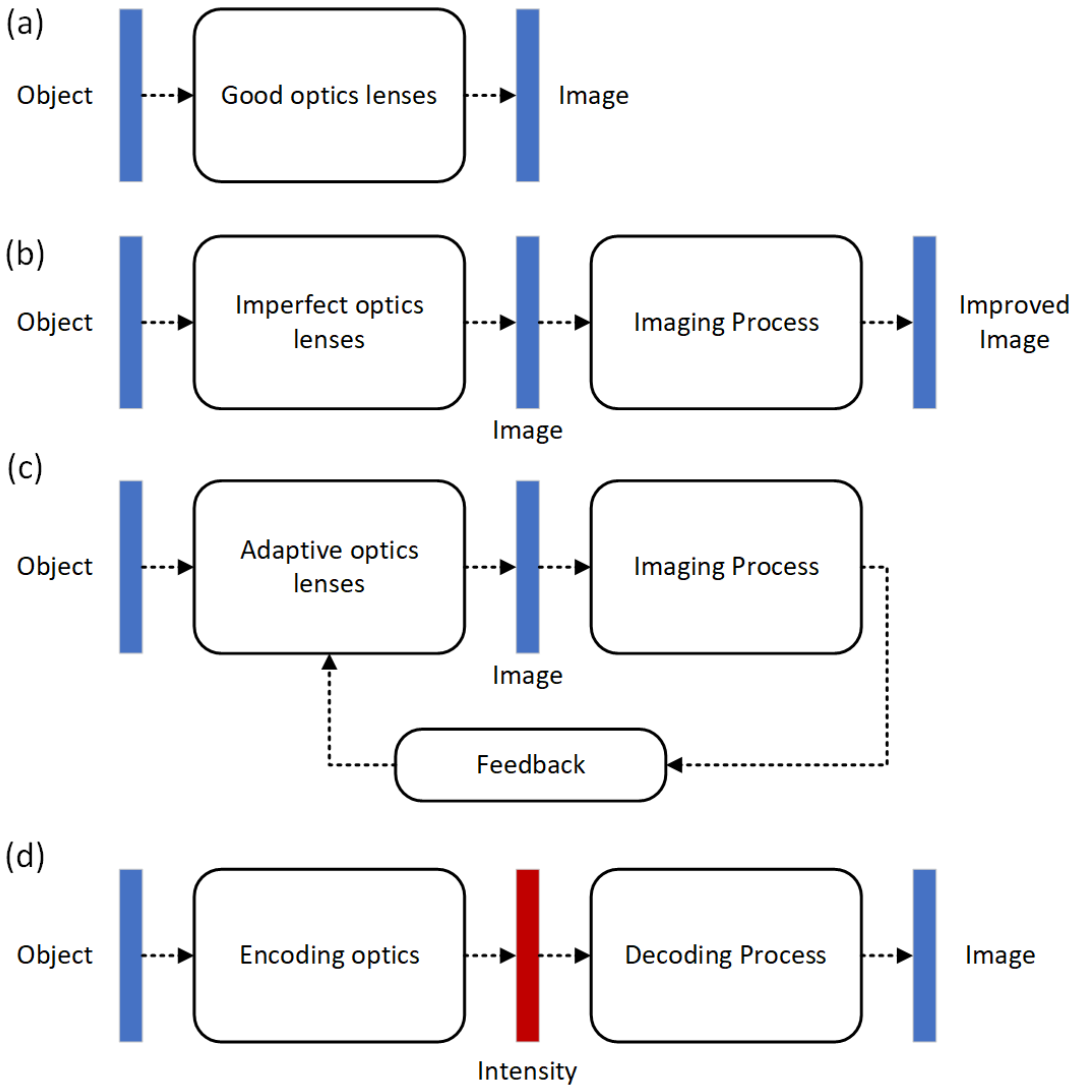

Figure 2.1. Different computational imaging process. (a) The conventional microscope imaging. (b) A post-processing is introduced to correct the imperfect optics. (c) The imaging optics can be adapted by the feedback from the error measurement in the detector plane. (d) The detector plane only records the intensity from the encoding process, then using computer to decode and reconstruct the image.

A conventional microscope in Figure 2.1(a) requires good optics lenses, post-processing techniques are not typically applied to correct optical aberrations or imperfections [21]. They rely on simple optical systems and lenses to magnify and visualize specimens [19, 22]. However, as a matter of fact, there is no absolutely perfect lens in the world. Figure 2.1(b) shows that some post-processing techniques such like filtering, sharpening, noise reduction, contrast

enhancement and deconvolution etc. can be applied to correct or enhance images obtained with a microscope [7, 21]. Furthermore, if the optics lenses are adaptive rather than fixed, the feedback from the post imaging process can be introduced to improve the imaging setup [23], see Figure 2.1(c). All these imaging types mentioned above are direct imaging by the optics. The quality of the lens determines how good the image will be. Because the microscopes are used to see smaller things or structures, the resolution can generally represent its final imaging quality. For a conventional transmission microscope, according to Abbe's theory [24], the resolution of the image is Equation $(2.1)(2.1)$:

$$\gamma = \frac{\lambda}{2nsin\theta} \tag{2.1}$$

where λ is the wavelength of the incident wave, n is the refractive index of the medium between the objective lens and the object, θ is the half-subtended angle by the objective lens. Here, $nsin\theta$ is defined as the numerical aperture (NA). It is obvious that the resolution is limited by the wavelength of the incident wave and the highest scattering angle that can be collected by the objective lens. To overcome the resolution limit imposed by the imperfect objective lens, a new lens-less strategy that gets rid of the lens during imaging was proposed in Figure 2.1 (d) [25]. The measurements from the detector are diffraction patterns which record the intensity information rather than obtaining the image of the specimen directly. This thesis will mainly focus on the decoding process step of this type of lens-less imaging in Figure 2.1 (d).

2.2. Resolution

As mentioned above, resolution is an important metric for quality of images, and it is limited by the wavelength and the numerical aperture (NA) of the objective lens for a conventional microscope [23]. However, in lens-less diffractive imaging, it is limited by the NA of the detector. This is an advantage

of diffractive imaging, as the NA of the detector is usually big enough to capture all the diffraction from the object. Therefore, the limitation is now only the wavelength.

In the far-field propagation, the resolution of the reconstruction will not be changed as long as it satisfies the sampling condition. Assume there is a detector with $N \times N$ pixels, and each pixel size is Δu. The distance between specimen and detector is z, see Figure 2.2. The size of the pixel in the reconstruction can be defined as:

$$\Delta x = \frac{\lambda z}{N \Delta u} \tag{2.2}$$

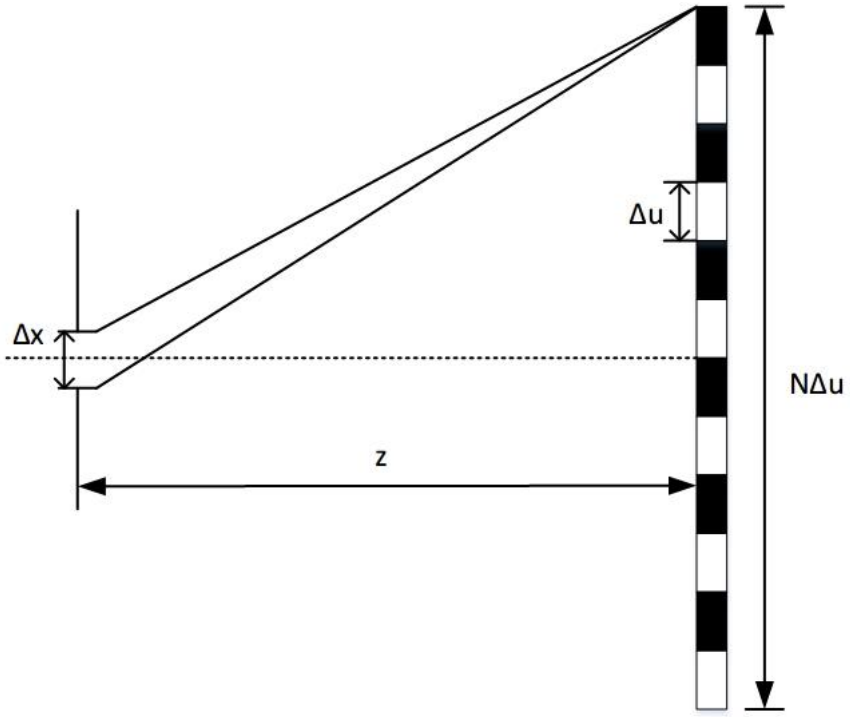

Figure 2.2. The pixel size in a far-field imaging system.

In Equation (2.2), N is the number of pixels and Δu is the pixel size on the detector, therefore, $N\Delta u$ will be the size of the detector which will remain constant, the resolution of Δx will not change no matter how we divide the pixel size in detector. Unlike Abbe's theory which considers the minimum resolvable

distance between two points in the object space, in lens-less imaging, another concept called the Rayleigh Criterion is used to find out the relationship between resolution and imaging system [26]. The Rayleigh Criterion is more concerned with the separation of point sources in an image. Assume there are two point sources, if the central brightest point of the diffraction image of one point source coincides with the first darkest point of the diffraction image of another point source, these two sources can be resolved [27]. For incoherent imaging, according to the Rayleigh Criterion, the resolution which is measured as a distance can be defined as:

$$\gamma = \frac{0.61\lambda}{NA} \tag{2.3}$$

where λ is the wavelength, here, NA is the numerical aperture which can also be written as $NA = nsin\theta$. n is the refractive index of the media and θ is the half angle of the incoming wave.

According to Equation (2.3), the resolution of an imaging system is determined by the wavelength and effective numerical aperture of the system [27]. The numerical aperture can be increased by increasing the incoming wave angle to improve the resolution, but it cannot be greater than 90 degrees. Improving the refractive index of the media around the lens can also make the numerical aperture larger such as by placing immersion oil near the lens [26]. Another way to improve the resolution is using a shorter wavelength source, e.g., UV, X-ray or electrons [26]. This is the physical limitation of the resolution, determined by the experimental configuration in practice. However, this thesis is more focused on computational imaging, which improves the quality of the image as much as possible, with some computational algorithms in the determined configuration.

2.3. 2D Fourier Analysis

Fourier analysis is often utilsed for both linear and nonlinear systems, it is

widely used in signal processing, especially in the field of communication systems and electrical networks. In the context of diffraction imaging in microscopy, Fourier analysis plays a crucial role in the formulation of the propagation theory, it connects the specimen in real space and its spectrum in reciprocal space. In this thesis, The Fourier transform is extensively used in many places. Here we start with its definitions and properties [26].

2.3.1. Fourier Transform Definitions and Properties

In the field of Fourier optics, most problems involve two dimensions, the Fourier transform of a 2D function $\psi(x,y)$ is defined as Equation (2.4):

$$\Psi(u,v) = \boldsymbol{\mathcal{F}}\{\psi(x,y)\} = \iint \psi(x,y)\, e^{-i2\pi(ux+vy)} dxdy \qquad (2.4)$$

where $\boldsymbol{\mathcal{F}}$ is the operation of Fourier transform, i is the imaginary unit, x and y are the coordinates in real space, and u, v are coordinates in reciprocal space, usually referred to as spatial frequencies in Fourier domain.

The Fourier Transform operation is reversible. The inverse Fourier transform of a reciprocal function $\Psi(u,v)$ is defined as Equation (2.5):

$$\psi(x,y) = \boldsymbol{\mathcal{F}^{-1}}\{\Psi(u,v)\} = \iint \Psi(u,v)\, e^{i2\pi(ux+vy)} dudv \qquad (2.5)$$

where $\boldsymbol{\mathcal{F}^{-1}}$ represents the inverse Fourier transform.

For a 2D function, an important property is separability. A separable 2D function can be written as the product of two 1D function:

$$\psi(x,y) = \psi_x(x)\psi_y(y) \qquad (2.6)$$

meaning the *Fourier transform* of a 2D function that is the product of two 1D transforms [28]:

$$\mathcal{F}\{\psi(x,y)\} = \mathcal{F}\{\psi_x(x)\}\mathcal{F}\{\psi_y(y)\} \tag{2.7}$$

The basic definition of Fourier transform leads to many well-known theorems. Here we introduce some of them which are relevant to Fourier optics. Assume there are two real space function $\psi(x,y)$ and $\phi(x,y)$, their Fourier transforms are $\Psi(u,v)$ and $\Phi(u,v)$.

1) **Linearity theorem:** The Fourier transform of a weighted sum of two or more functions is same as the identically weighted sum of their individual transforms, see Equation (2.8):

$$\boldsymbol{\mathcal{F}}\{a\psi(x,y) + b\phi(x,y)\} = a\Psi(u,v) + b\Phi(u,v) \tag{2.8}$$

2) **Similarity theorem:** The scaling of the real space coordinates will result a contraction of the reciprocal space coordinates and an overall contraction of the amplitude, see Equation (2.9):

$$\boldsymbol{\mathcal{F}}\{\psi(ax,by)\} = \frac{1}{|ab|}\Psi\left(\frac{u}{a},\frac{v}{b}\right) \tag{2.9}$$

3) **Shift theorem:** Linear offset in the real space will result a linear phase shift in the reciprocal space, see Equation (2.10):

$$\boldsymbol{\mathcal{F}}\{\psi(x-a,y-b)\} = \Psi(u,v)e^{-i2\pi(au+bv)} \tag{2.10}$$

4) **Parseval's (Rayleigh's) theorem:** The energy contained in the waveform will remains constant after transformed into reciprocal space, see Equation (2.11):

$$\iint_{-\infty}^{\infty} |\psi(x,y)|^2 dxdy = \iint_{-\infty}^{\infty} |\Psi(u,v)|^2 dudv \tag{2.11}$$

where the left-hand side integral can be expressed as the energy contained

in waveform $\psi(x,y)$, the integral on the right-hand side is the energy density in reciprocal space.

5) **Convolution theorem:** The Fourier transform of a convolution of two real space function is equal to the product of their transforms in reciprocal space, see Equation (2.12):

$$\boldsymbol{\mathcal{F}}\{\psi(x,y)\otimes\phi(x,y)\} = \Psi(u,v)\Phi(u,v) \qquad (2.12)$$

where $\otimes$ represents the convolution operation.

6) **Fourier integral theorem:** The result of successive transform and inverse transform of a function is itself, except at points of discontinuity, see Equation (2.13):

$$\boldsymbol{\mathcal{F}\mathcal{F}^{-1}}\{\psi(x,y)\} = \boldsymbol{\mathcal{F}^{-1}\mathcal{F}}\{\psi(x,y)\} = \psi(x,y) \qquad (2.13)$$

7) **Successive transform theorem:** A successive transform of a function leads the change of the sign of its coordinates, see Equation (2.14):

$$\mathcal{F}\mathcal{F}\{\psi(x,y)\} = \psi(-x,-y) \qquad (2.14)$$

8) **Conjugate symmetry theorem:** The Fourier transform of the conjugate of a function will be a conjugate symmetric version of its Fourier transform, see Equation (2.15):

$$\mathcal{F}\{\psi^*(x,y)\} = \psi^*(-u,-v) \qquad (2.15)$$

From Equation (2.4) and (2.5), it is easy to see that, mathematically, the only difference between the Fourier transform and inverse Fourier transform is the sign of the exponent. The integrals in these two equations may not always exist for certain functions, that requires the function being integrated must satisfy to the following conditions [26]:

1) The function must be integrable over the infinite plane.

2) The function must have only a finite number of discontinuities and a finite number of maxima and minima in any finite rectangle.

3) The function must have no infinite discontinuities.

This is the mathematical requirement for Fourier transform, however, in the real life, the signal read by a computer has to be decomposed into discrete points and is not continuous anymore [23]. This is called sampling in signal processing, the smaller interval between two adjacent discrete points means the higher sampling frequency. Therefore, it is essential to represent signal function by discrete arrays of sampled values and find a way to process these discrete signals when we implement Fourier operations on the computer.

2.3.2. Sampling and the Shannon–Nyquist Sampling Theorem

Sampling is the conversion of a continuous signal function in time domain or space domain into a sequence of values (a discrete function). Suppose there is a 2D analytic function $\psi(x, y)$ laying in a 2D space as shown in Figure 2.3 (a). Assume it is sampled in a uniform manner in both x and y directions, see Figure 2.3 (b). We can now write $\psi(x, y)$ in a discrete form:

$$\psi(x, y) \rightarrow \psi(m\Delta x, n\Delta y) \tag{2.16}$$

where the Δx and Δy are the sample interval in different directions, m and n are the integer index of the sample in correlated direction. Alternatively, $1/\Delta x$ and $1/\Delta y$ represent the respective sample rates or frequencies. For instance, we sample $\psi(x, y)$ in a finite space, assuming it has $M \times N$ samples, its integer index m and n are usually defined as:

$$m = -\frac{M}{2}, -\frac{M}{2} + 1, \dots, \frac{M}{2} - 1, n = -\frac{N}{2}, -\frac{N}{2} + 1, \dots, \frac{N}{2} - 1 \tag{2.17}$$

where M and N are normally considered to be even number in a standard index arrangement. The side lengths of the finite physical area now can be expressed as:

$$L_x = M\Delta x, L_y = N\Delta y \tag{2.18}$$

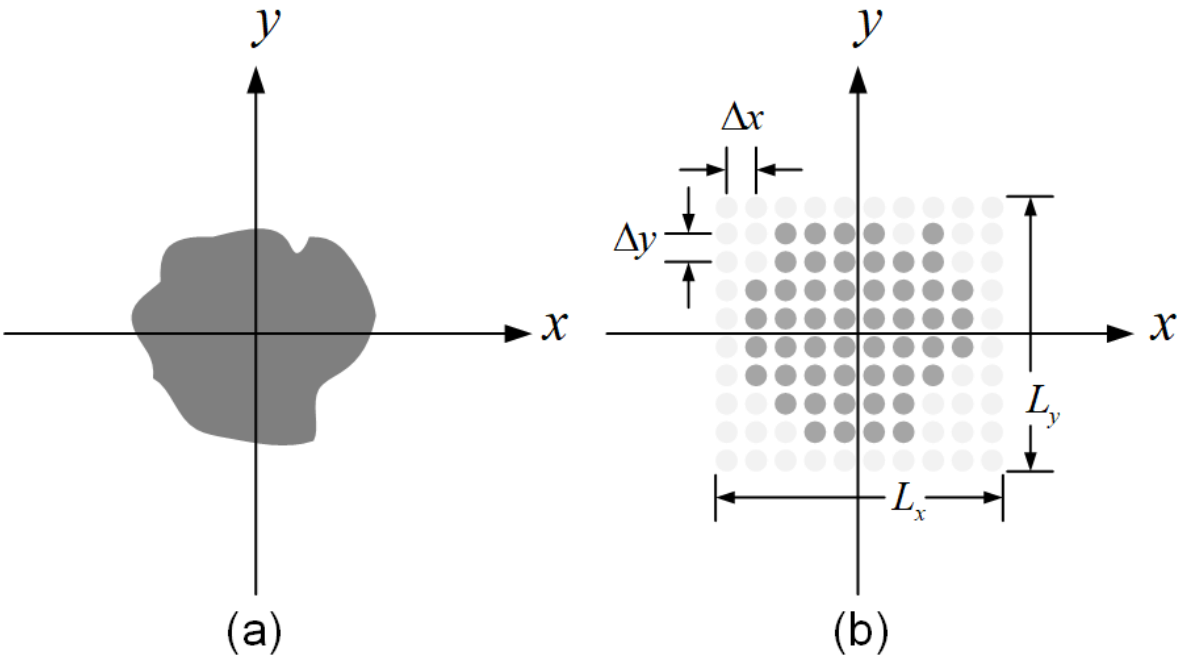

Figure 2.3. (a) An analytic 2D function, (b) the sampled version of (a).

Now, in this case, the first concern of sampling is whether $\psi(x, y)$ it is able to fit within the finite space defined by $L_x \times L_y$. This introduces the first requirement that the support area $D_x \times D_y$ which is the span of $\psi(x, y)$ has to be smaller than the finite space:

$$D_x < L_x, D_y < L_y \tag{2.19}$$

Figure 2.4 indicates a simple function along the x direction, with value one for all the points in the support, (a) shows a support area D_x along the x direction of $\psi(x, y)$, correspondingly, (b) is its spectrogram in the Fourier domain. The null-to-null bandwidth is usually defined as the difference between the

frequencies at which the first zero crossings occur on either side of the central peak, shown in Figure 2.4 (b). It provides a measure of the frequency range occupied by the main lobe of a signal's spectrum while for most practical applications, the contribution of the sidelobes are negligible [28].

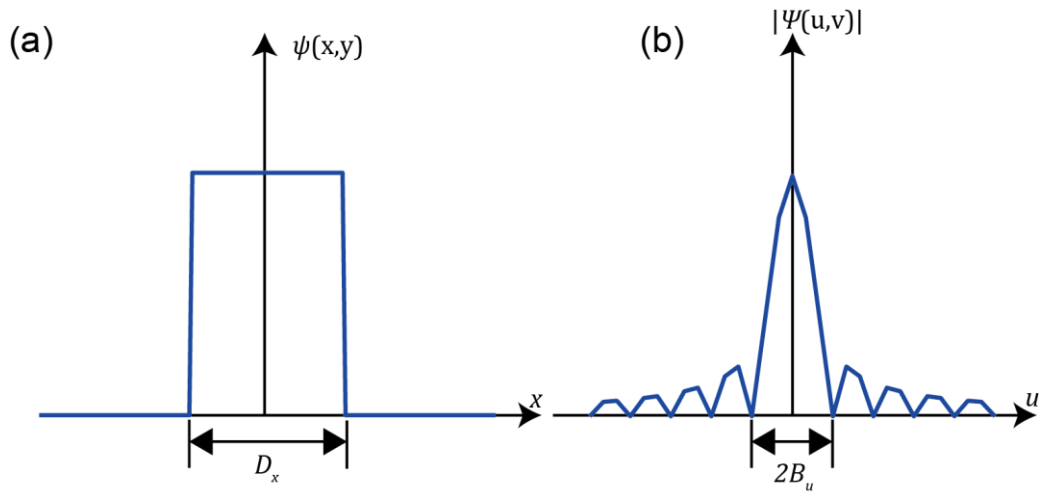

Figure 2.4. (a) A support D_x for a real function $\psi(x,y)$ along the x direction and (b) its bandwidth B_u in Fourier domain.

Another important concern is that whether all the detailed feature of $\psi(x,y)$ can be presented after sampling. For a bandlimited function, a continuous signal can be exactly recovered with a sample interval smaller than a specific value or a sample rate greater than a specific frequency. This is the Shannon-Nyquist sampling theorem [28]. For a 2D function, it requires the function meet the conditions in both directions.

$$\Delta x < \frac{1}{2B_u}, \Delta y < \frac{1}{2B_v} \tag{2.20}$$

where B_x and B_y are the bandwidth of the function in different directions. Correspondingly, the sample rate has to satisfy the follow requirement:

$$f_x > 2B_u, f_y > 2B_v \tag{2.21}$$

In addition, the half of the sample rate Nyquist frequency can be defined by:

$$f_{Nx} = \frac{1}{2\Delta x}, f_{Ny} = \frac{1}{2\Delta y} \tag{2.22}$$

2.3.3. Discrete Fourier Transform (DFT)

As mentioned above, *Fourier Transform* has a requirement for the continuity of the function, but in practice, all the signal or image collected by the computer are discrete. Here, we introduce *discrete Fourier transform* (DFT) and its improved version the *fast Fourier transform* (FFT) [29]. DFT are the fundamental tools on the computer for solving Fourier problems and used extensively in later simulations and experiments in this thesis. Consider a 2D function $\psi(x,y)$ and its sampled function $\psi(m\Delta x, n\Delta y)$ as indicated in Equation (2.16), apply the Fourier transform from Equation (2.4) to this discrete function and use Riemann sum to get an approximation of this integral:

$$\iint_{-\infty}^{\infty} \psi(m\Delta x, n\Delta y) e^{-i2\pi(ux+vy)} dxdy \rightarrow \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{m=-\frac{M}{2}}^{\frac{M}{2}-1} \psi(m\Delta x, n\Delta y) e^{-i2\pi\left(\frac{um\Delta x}{M}+\frac{vn\Delta y}{N}\right)} \tag{2.23}$$

Therefore, the definition of DFT and inverse DFT (DFT^{-1}) can be written as:

$$\Psi(u,v) = \boldsymbol{DFT}\{\psi_{x,y}\} = \sum_{y=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{x=-\frac{M}{2}}^{\frac{M}{2}-1} \psi_{x,y} e^{-i2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \tag{2.24}$$

$$\psi_{x,y} = \boldsymbol{DFT^{-1}}\{\Psi(u,v)\} = \frac{1}{MN} \sum_{v=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{u=-\frac{M}{2}}^{\frac{M}{2}-1} \Psi(u,v) e^{-i2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \tag{2.25}$$

where $\psi(x,y)$ is a discrete real space function and $\Psi(u,v)$ is discrete reciprocal function, $M \times N$ represents its sampling space and M, N are the even integer which are the number of the sample for each direction.

In practice, the DFT and DFT^{-1} are usually implemented by computationally

efficient FFT and FFT^{-1} algorithms [29]. FFT algorithms is not actually using Equation (2.24) and (2.25), but its result is consistent with these two equations [28]. Moreover, all the properties and theorems mentioned in section 2.2.1 can be applied here in the discrete format as well. For example, a convolution of two discrete functions can be computed by multiplying its individual FFT result and then do the FFT^{-1}.

2.4. Diffraction and Wave Propagation

In this section the Fourier theories introduced above are used to explain and model the propagation of coherent light or matter waves.

2.4.1. Huygens-Fresnel Principle and Rayleigh–Sommerfeld Solution

Diffraction is the spreading of waves around obstacles during their propagation. The Huygens-Fresnel principle is an analytical method for studying wave propagation, named for the Dutch physicist Christian Huygens and the French physicist Auguste Fresnel. This principle applies to both the far-field limit and near-field diffraction [27]. The Huygens-Fresnel principle can correctly explain and calculate the propagation of waves. It shows that one breaks the wavefront into a number of short sections and treats each section as a source of a spherical wavelet. Far from the source, say on a distant detector, these wavelets will add together or cancel each other to produce a diffraction pattern [26]. The diagram of Huygens-Fresnel principle is shown in Figure 2.5.

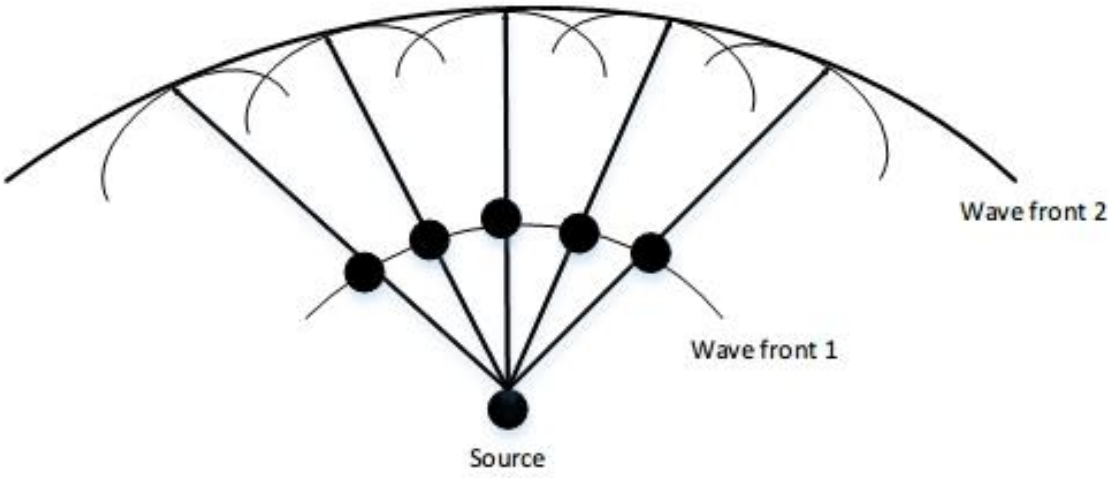

Figure 2.5. Huygens-Fresnel principle.

According to the Huygens-Fresnel principle, we are now able to calculate the amplitude of the wave after its propagation. Assume there is a point source $s(x, y, z)$ in 3-D space, the destination after propagation is point $S(a, b, c)$.

The distance d between these two points is:

$$d = \sqrt{(x-a)^2 + (y-b)^2 + (z-c)^2} \tag{2.26}$$

The amplitude at point S is:

$$A(d) \propto \frac{A_0 e^{ikd}}{d} \tag{2.27}$$

where A_0 is the original amplitude at point s, λ is the wavelength, k is the wavenumber which also can be written as $\frac{2\pi}{\lambda}$. Therefore, the amplitude will decrease as the distance d increases.

Now in a 2-D plane with coordinate (x, y). The propagation model between two planes is shown in Figure 2.6.

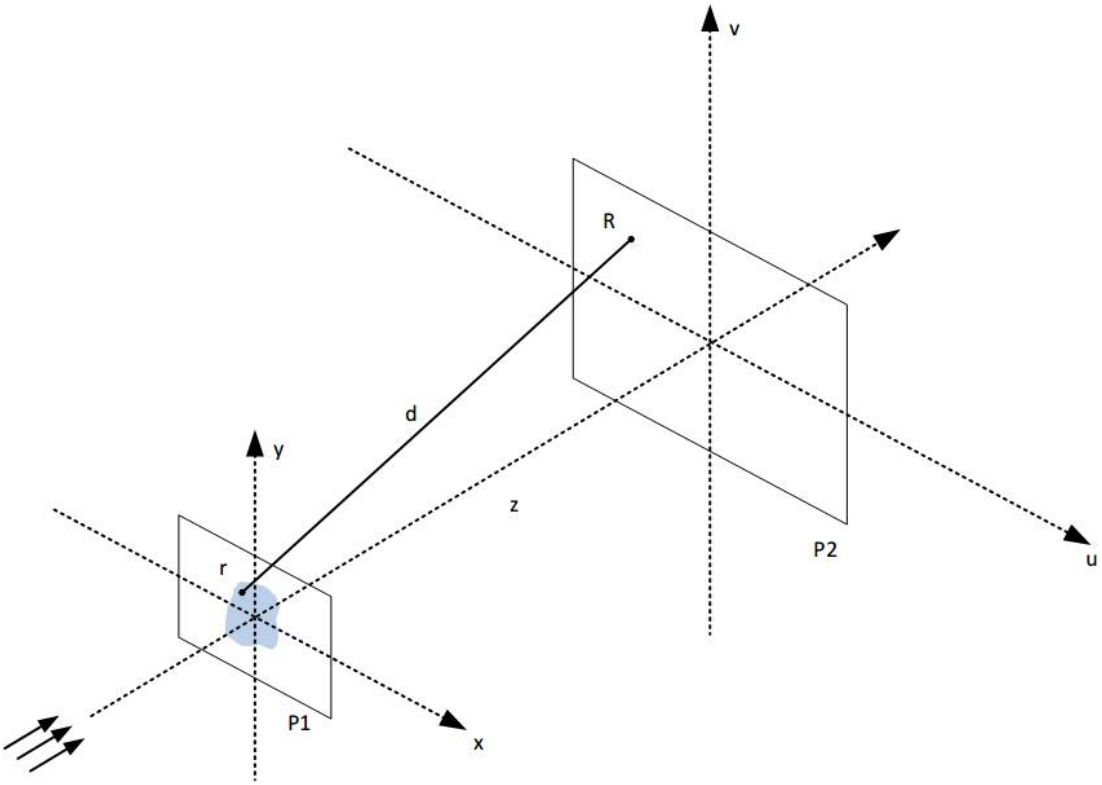

Figure 2.6. Propagation model between planes.

A wave illuminates from the source to plane $P1$, then its exit wave which has interacted with the specimen passes to the detector plane $P2$. The coordinates for the planes are (x, y) and (u, v). Therefore, the wave that arrives at the related point in $P2$ can be written as

$$\Psi(u,v) \propto \frac{\psi(x,y)e^{-\frac{i2\pi}{\lambda}d}}{d} \tag{2.28}$$

where $\psi(x,y)$ is the exit wave from plane $P1$, which is the interaction of the object and the incoming wave from the source. The distance d can be written as:

$$d = \sqrt{z^2 + (x-u)^2 + (y-v)^2} = z\sqrt{1 + \frac{(x-u)^2 + (y-v)^2}{z^2}} \tag{2.29}$$

where z is the distance between two planes.

Based on this propagation model, another important concept called Rayleigh–Sommerfeld solution is often used to predict the wave distribution on the observation plane [28]. The calculation of the wave that arrives at point R on the $P2$ plane, based on the Rayleigh-Sommerfeld expression [26], can be written as:

$$\Psi(u,v) = \frac{1}{i\lambda}\iint \frac{\psi(x,y)e^{-\frac{i2\pi}{\lambda}d}}{d}dxdy \qquad (2.30)$$

where $\psi(x,y)$ is the exit wave from plane $P1$, and d is the distance between two points. This is just the superposition of all the spherical waves from each point in plane $P1$.

2.5. Propagation Solutions

It is obvious that this transmission relationship in Equation (2.30) is related to distance d, the square root in the distance terms will make the Rayleigh-Sommerfeld solution more difficult and increase the execution time of computational simulations [28]. Here, we introduce two more convenient approximations for wave propagation, one is Fresnel approximation and the other is Fraunhofer approximation. These two types of approximation can be distinguished by Fresnel number, which defined as:

$$F = \frac{a^2}{d\lambda} \qquad (2.31)$$

Where a is the radius of the aperture in the source plane and d is the distance between the aperture and observation plane.

2.5.1. Fresnel Approximation

When $F \geq 1$, it can be considered as Fresnel propagation, which is also called near-field propagation. Consider a binomial expansion:

$$\sqrt{1+a} = 1 + \frac{1}{2}a - \frac{1}{8}a^2 + \cdots \tag{2.32}$$

where a is smaller than 1. Similarly, expand Equation (2.29) and only keep the first two terms:

$$d = z\sqrt{1 + \frac{(x-u)^2 + (y-v)^2}{z^2}} \approx z\left[1 + \frac{1}{2}\left(\frac{x-u}{z}\right)^2 + \frac{1}{2}\left(\frac{y-v}{z}\right)^2\right] \tag{2.33}$$

Then, this approximation of distance is applied in the exponential term in Equation (2.30), resulting in an assumption of a parabolic radiation wave instead of a spherical wave for the point source. Furthermore, substitute the approximation to Equation (2.30) and approximate $d \approx z$ in the denominator, then Fresnel approximation is:

$$\Psi(u,v) = \frac{e^{-\frac{i2\pi}{\lambda}z}}{i\lambda z}\iint \psi(x,y)e^{-\frac{i\pi[(x-u)^2+(y-v)^2]}{\lambda z}}dxdy \tag{2.34}$$

As an approximation, the accuracy of this expression will be affected when modelling scalar diffraction at a close range due to discarding the third and further terms in the series of the expansion [28]. Normally, if we allow 1 rad maximum phase change in the approximation, the distance z between two planes has to meet the following condition:

$$z^3 \gg \max\left\{\frac{\pi}{4\lambda}(x-u)^2 + (y-v)^2\right\} \tag{2.35}$$

where max represents the maximum value.

2.5.2. Fraunhofer Approximation

On the other hand, if Fresnel number $F \ll 1$, it will be the Fraunhofer approximation, which is also called far-field propagation. In this case, the detector is far away enough from the specimen, the coordinate (x, y) is tiny

compared to the distance z between two planes. The condition of z here is:

$$z \gg max\left\{\frac{\pi}{\lambda}(u^2+v^2)\right\} \tag{2.36}$$

Hence, the term x^2 and y^2 in the Equation (2.29) can be neglected and replaced with $2ux$ and $2vy$

$$d \approx z + \frac{u^2+v^2}{2z} - \frac{ux+vy}{z} \tag{2.37}$$

Substitute the approximation to Equation (2.30), then Fraunhofer approximation is:

$$\Psi(u,v) = \frac{e^{-\frac{i2\pi}{\lambda}\left(z+\frac{u^2}{2z}+\frac{v^2}{2z}\right)}}{i\lambda z}\iint \psi(x,y)e^{-\frac{2\pi i}{\lambda z}(ux+vy)}\,dxdy \tag{2.38}$$

where the integral term can be recognized as the Fourier transform of the specimen function $\psi(x,y)$. Therefore, we can consider the relationship of Fraunhofer propagation in a far-field system is a Fourier propagation.

2.6. Phase Problem

Consider the simplest version of a far-field diffraction imaging system, see Figure 2.7. A source wave passes through an aperture, illuminates the specimen, then the exit wave propagates to the detector located in a far-field Fraunhofer diffraction plane. As described in the previous section, the wave that arrives at the detector is the Fourier transform of the exit wave, which can be described as a complex scalar function:

$$\Psi(\vec{u}) = |\Psi(\vec{u})|e^{i\Phi(\vec{u})} \tag{2.39}$$

where $|\psi(\vec{u})|$ is the modulus part and $\Phi(\vec{u})$ is the phase part, and $\vec{u}$ is the coordinate vector in reciprocal space (Fourier space). The phase in the

Fraunhofer diffraction plane is the accumulated phase difference relative to free space, caused by the real part of the refractive index of the specimen as the wave passes through its thickness [7]. In Fraunhofer diffraction, when light passes through a specimen, it interacts with the material properties of the specimen, and the real part of the refractive index introduces phase shifts. The accumulated phase difference is then determined by comparing the phase of the diffracted light wave after passing through the specimen with the phase it would have had in free space. This accumulated phase difference in the Fraunhofer diffraction plane carries information about the internal structure or properties of the specimen, particularly when the specimen is transparent or has variations in refractive index across its thickness [26].

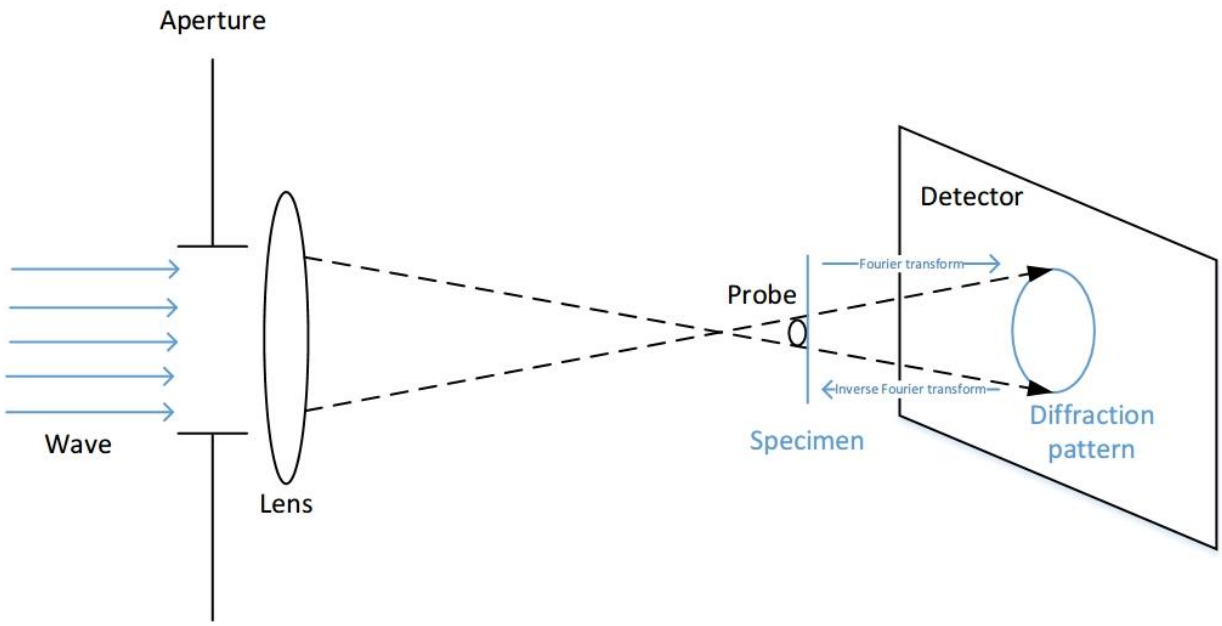

Figure 2.7. Schematic of a concise far-field diffraction imaging system.

According to the propagation theory mentioned above, if the specimen and source are known and certain, the wave $\Psi(\vec{u})$ after propagation on the diffraction pattern can be easily predicted. However, the far-field detector is not able to record the phase information of the exit wave after propagation. The only thing we can get in the diffraction plane is the intensity of the wave after

propagation, denoted as:

$$I(\vec{u}) = |\Psi(\vec{u})|^2 = \Psi(\vec{u})\Psi(\vec{u})^* \quad (2.40)$$

where $*$ is the complex conjugate operator.

Applying the inverse Fourier transform to the intensity will provide only the autocorrelation of the specimen:

$$\boldsymbol{\mathcal{F}^{-1}}\{I(\vec{u})\} = \psi(-\vec{r}) \otimes \psi(\vec{r}) \quad (2.41)$$

where $\otimes$ is the convolution operator, and $\vec{r}$ is the coordinate vector in real space.

Therefore, the phase problem is either solving $\Psi(\vec{u})$ in Equation (2.40) or $\psi(\vec{r})$ in Equation (2.41), using measured intensity $I(\vec{u})$ [30]. This is a classic inverse problem, the forward calculation from a known specimen to its diffraction plane is easy and straightforward. By contrast, the backward calculation from the diffraction plane to an unknown specimen is much more difficult due to the loss of phase information. The relationship between the specimen and the measurements is non-linear and indirect, requiring sophisticated algorithms to solve. These algorithms are usually computationally intensive, requiring significant processing power and memory, especially for high-resolution images or large datasets.

2.7. Single-Shot Phase Retrieval

Like all inverse problems, applying known constraints is a common way to find a solution to the phase problem, more extra prior knowledge is always helpful for the solution. In the phase problem, what we already know is that the experiment setup, the measurement of the intensity from the detector, the support of the specimen which is a delineated finite area.

A key breakthrough that uses this prior information to solve the phase problem is a computational iterative method proposed by Fienup [31], improved from a solution strategy originally from Gercherg and Saxton [6]. In Fienup's method, an iterative loop is set up for the computation, shown in Figure 2.8. Here, the "object" is a sampled representation of the specimen, similar to that shown in Figure 2.3 (b).

In this loop from Figure 2.8, each iteration will start at position A. At the beginning, make an initial guess of object, this guess is not very important, usually made up of an $M \times N$ image matrix filled with 1 for all pixels. In the first step, all the pixels outside of the known support will be 0, to enforce the fact that none of the radiation incident on the sample passed through this region of the specimen. Next, propagation to the detector plane is modelled by taking the Fourier transform of the estimation. Now, it comes to the position B in the loop, where it has got the estimated diffraction pattern. The estimated diffraction pattern has both estimated modulus and estimated phase. The estimated modulus is replaced with the measured data from the detector, keeping the estimated phase, to enforce the fact that the wavefront must have the same intensity as the measured diffraction pattern. Finally, from C to D, do the inverse Fourier transform and go back to real space. Now, the 'constraint' which is the measurement has been applied on the estimation. Therefore, the new object estimation is closer than the previous one due to correct data from measurement.

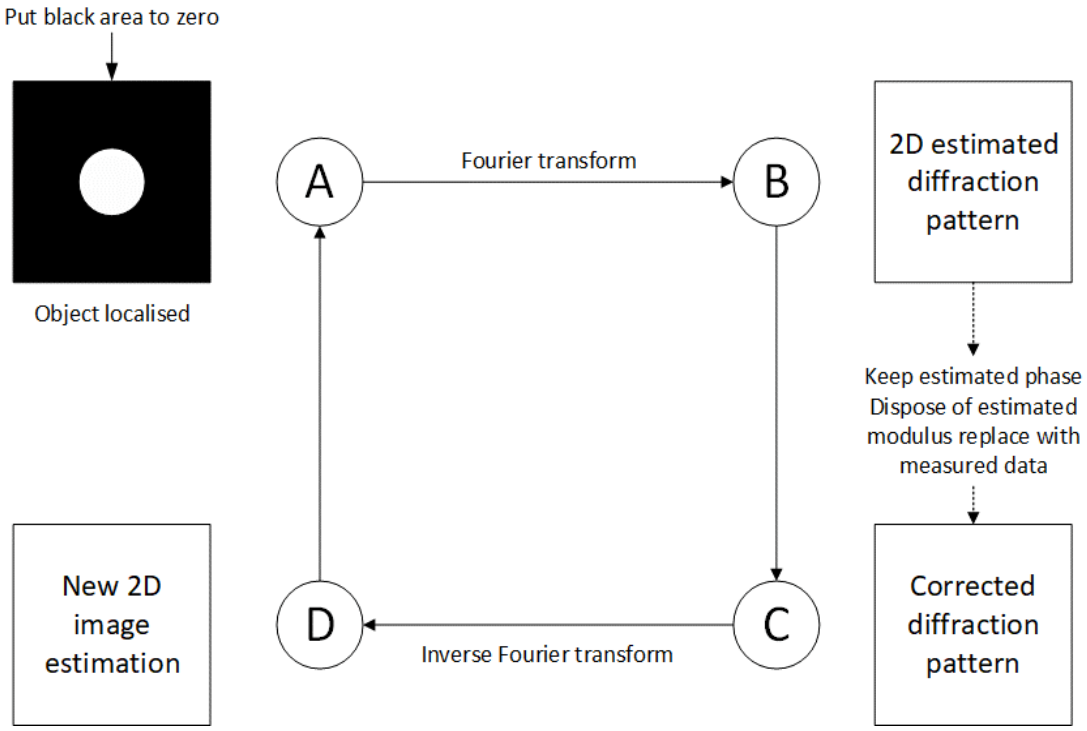

Figure 2.8. Iterative computational loop.

This computational method is called Error Reduction (ER) [31]. By applying the constraints sequentially and iteratively, the error of the reconstruction will certainly reduce. However, the speed of convergence cannot be guaranteed. ER method was successful and pioneering in the early days of phase retrieval, laid the foundation for many later iterative phase retrieval algorithms.

2.8. Ptychography

A computational method called ptychography is now widely used as an alternative to conventional phase retrieval for solving the phase problem which was introduced in the last section. Compared to the conventional method in Figure 2.7, what makes ptychographic imaging system special is that it uses a moveable aperture (or 'probe') to move around the specimen and collect diffraction patterns from different positions on the sample. For each position, the detector will record one diffraction pattern [6]. Each position of the diffraction pattern should be overlapped, as the schematic diagram shows in Figure 2.9.

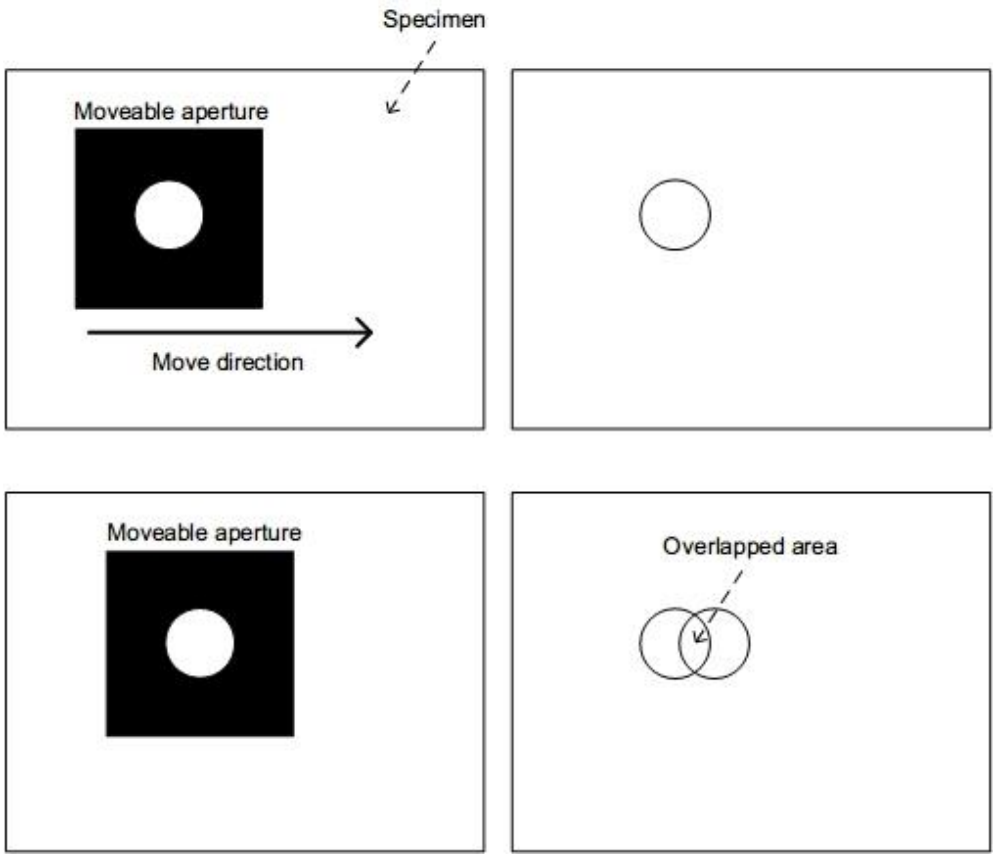

Figure 2.9. Moveable aperture in ptychography.

Unlike the single-shot methods described in the previous section, the overlapped area in ptychography will provide more prior information from a previous scan position when it is involving the next position. Figure 2.10 illustrates the experimental comparison of single-shot method and ptychography. (a) is the conventional imaging using a lens which significantly limits the angular range of scattered waves. The angular range is not limited in (b), but the phase of the scattered wave field is lost. (c) is the ptychography method, using a moving aperture gives a large field of view measurement at wavelength resolution.

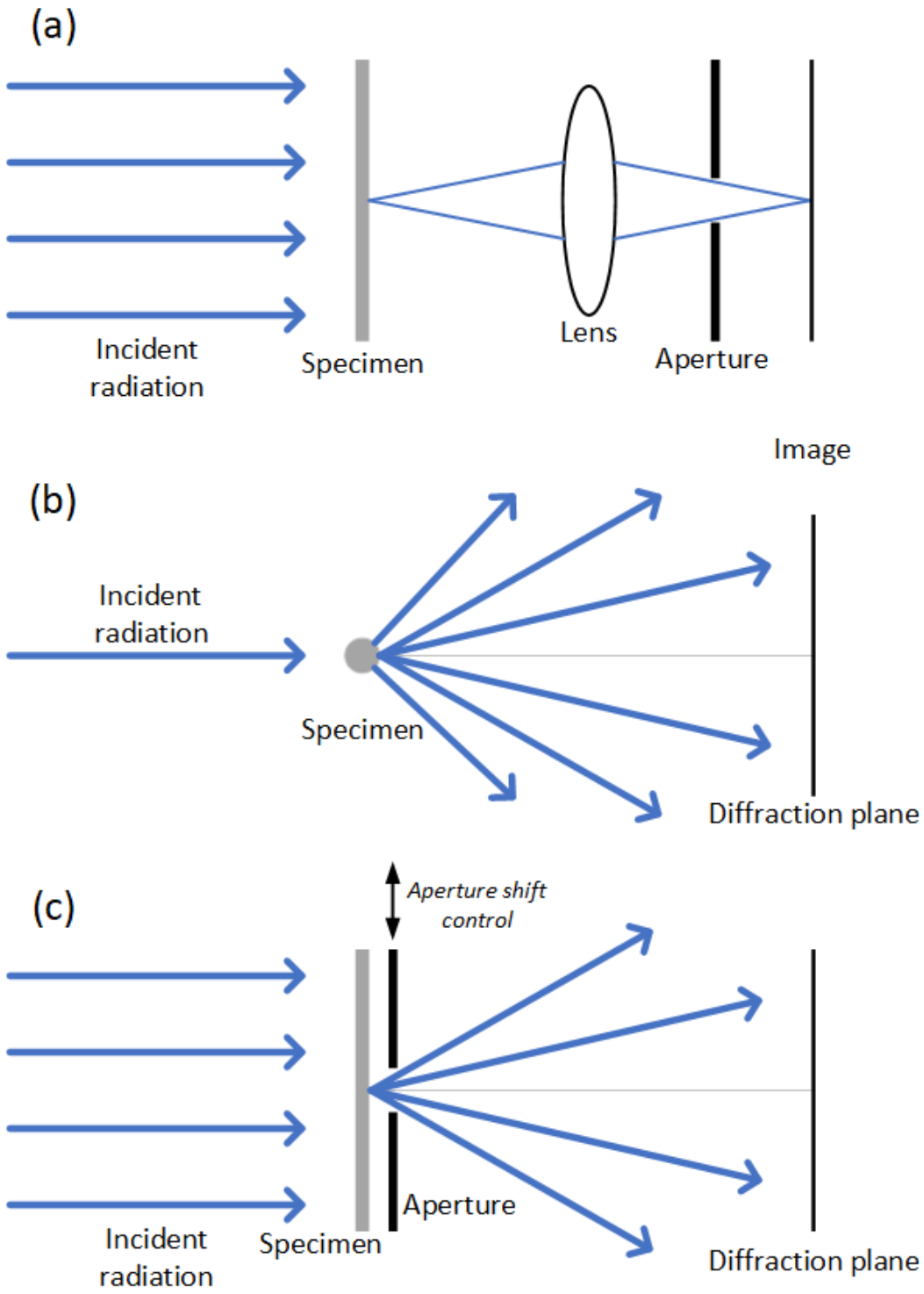

Figure 2.10. (a) Conventional lens imaging experimental setups with a limiting aperture in the lens back focal plane. (b) Unlimited angular range, but the phase is lost. (c) Ptychography experimental setup using a moving aperture.

In ptychography, all the positions of diffraction patterns have a similar iterative loop to Figure 2.8. All the estimation of different positions will be updated one

by one in each iteration. A flow chart of ptychography for an example of two positions is illustrated in Figure 2.11.

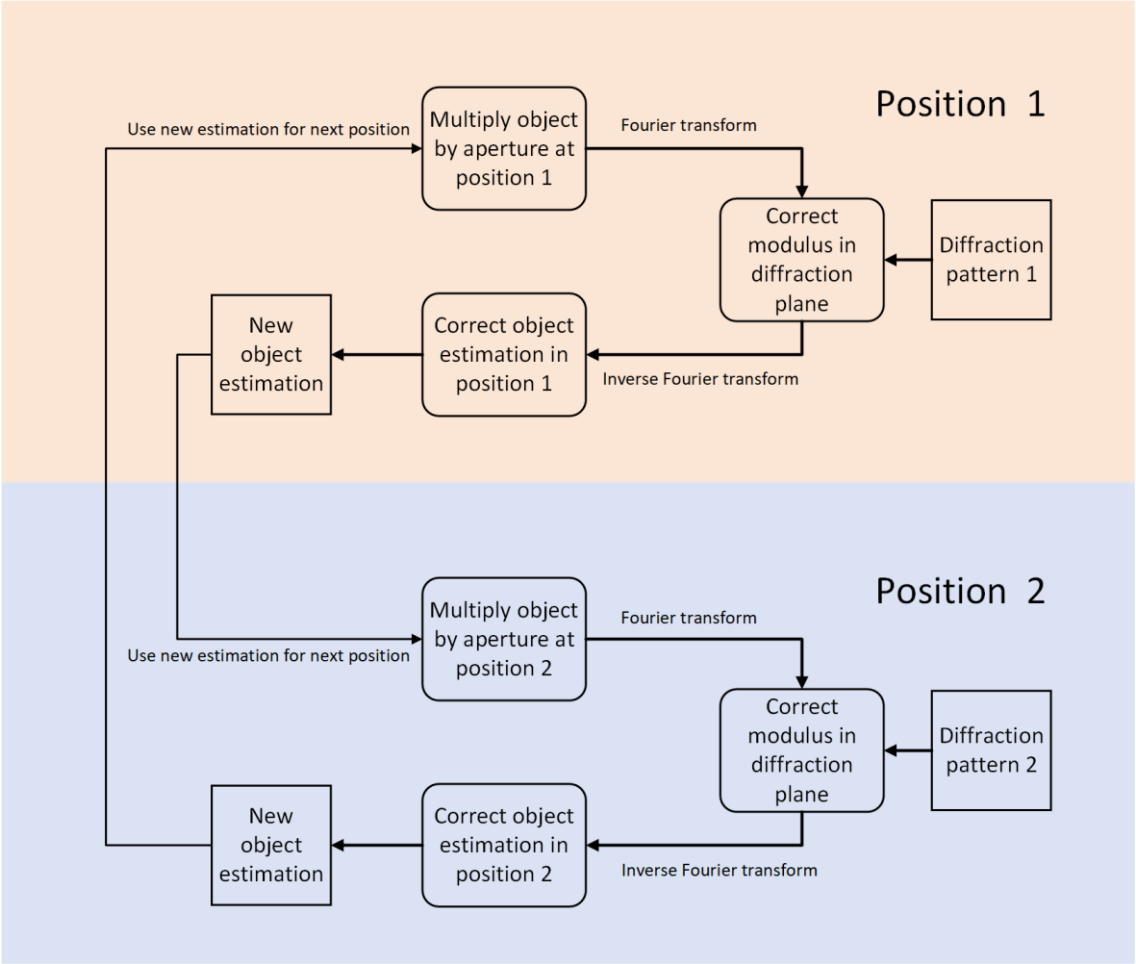

Figure 2.11. Flow chart for two positions in ptychography.

Hence, when the first position is finished in the first iteration, its adjacent positions have already got the new estimation of the overlapped area, which is more accurate than the previous estimation. Then, when doing the iteration of these adjacent positions, it will be closer to the correct result. It is much easier and more accurate to retrieve the phase because of these advantages. Nowadays, there are a lot of ptychography related applications and algorithms for microscope imaging. Ptychography has been successfully implemented in many fields, such as biology, materials science, physics, and chemistry. Also, ptychography is the core concept of this thesis, all the research in later chapters is based on this imaging technique. In the next section, a brief literature review

about existing ptychographic algorithms will be presented.

2.9. Algorithms for Ptychography

2.9.1. Direct Ptychography

Since ptychography was first introduced, numerous algorithms to recover images from the diffraction patterns have been developed. The first method was direct ptychography, also referred as non-iterative ptychography, using a famous method called Wigner Distribution Deconvolution (WDD) proposed in 1989 [10], more than 10 years before the modern iterative solutions. It was abandoned for nearly twenty years because of its complex and data-intensive computation, since in comparison with iterative methods WDD requires a diffraction pattern to be collected for every pixel in the reconstructed phase image. WDD handles the diffraction data as a 4D dataset, involves the use of the Wigner distribution to analyse and correct aberrations in both spatial and frequency domains, ultimately leading to improved image reconstruction. Today the method has seen a revival of interest due to the dramatic increase in computing power of computers. Rodenburg & Bates used a method called ‘’stepping out’’ to break through the cut-off restriction to recover higher resolution [32]. Later in 2014, based on this, a projection strategy proposed by Li [33], improves the robustness and reduces the inconsistencies. Now, there are several successful implementations of WDD in the field of 4D-STEM imaging. Yang et. al. used WDD as a tool for electron ptychographic phase imaging of light elements in crystalline [34]. Clark et al. simulated GaN 4D-STEM datasets of varying thickness and reconstructed via WDD [35]. Apart from these, a further development of WDD for 4D STEM data analysis with the precomputation of the Wiener filter for efficient deconvolution and the implementation of live processing for gradual reconstruction was introduced by Bangun et al. in 2023 [36]. A full derivation of WDD and detailed introduction of

a new WDD approach for blind ptychographic phase retrieval will be given in Chapter 4.

2.9.2. Iterative Ptychography

The inception of iterative algorithms in the field of ptychography marked a significant advancement in computational imaging techniques. Rodenburg introduced the first iterative algorithm ptychographic iterative engine (PIE) in 2004 [6], laying the foundation for subsequent developments in the field. Building upon PIE, Maiden and Rodenburg introduced an extended version known as ePIE in 2009 [17]. Unlike its predecessor, ePIE not only solves for the object but also concurrently addresses the optimization of the probe function. This innovative approach has gained widespread adoption in various ptychographic applications, underscoring its versatility and effectiveness. In 2017, further enhancements to the PIE family methods emerged with the introduction of regularized PIE (rPIE) and momentum PIE (mPIE) [18]. These variants aimed to refine and optimize the reconstruction updating process, enhancing the robustness and efficiency.

Beyond the PIE family, the Difference Map (DM) method, introduced in 2005 [37, 38], stands out as another notable approach. Similarly, DM not only reconstructs the object but also possesses a unique capability to solve for the probe. Addressing improvements in reliability and efficiency, Luke proposed the Relaxed Averaged Alternating Reflection (RAAR) method in 2005 [13]. This method builds upon the principles of DM but introduces refinements that contribute to a more reliable and efficient reconstruction process. RAAR represents a crucial development by introducing relaxation, offering a viable alternative with enhanced performance characteristics.

Apart from these two categories, another iterative maximum likelihood approach that uses least-squares (LSQ-ML) was proposed by Odstrčil, Menzel

and Guizar-Sicairos in 2018 [16]. LSQ-ML introduced maximum-likelihood into iterative ptychography, providing stable convergence, faster initial convergence, and low memory requirements [16]. More up-to-date, a method called Proximal algorithms, which was originally developed in convex optimization theory, also has been used for ptychographic phase retrieval [14]. The ptychographic proximal algorithm performs well in the recovery of high-frequency features and the smoothness of the images [14].

This thesis is primarily concerned with analysis, comparison and improvements to the algorithms that are used for ptychography. In the next chapter I will model the ptychography and describe the performance metrics used to assess the different algorithms, before moving on to discuss Wigner deconvolution method in Chapter 4, set projection algorithms in Chapter 5; finally WASP algorithms in Chapter 6.

3. Ptychographic Simulation and Reconstruction

This chapter will focus on the simulation and reconstruction of a ptychographic experiment on the computer, which is the basis for later chapters when we introduce the different ptychographic algorithms. A ptychographic algorithm called the *extended ptychographic iterative engine* (ePIE), which is now very popular and widely used, will be used as an example to show the simulation and reconstruction process. Also, the associated error metric of the simulation will be introduced and defined to evaluate the performance of different phase retrieval algorithms.

3.1. MATLAB

MATLAB as a mathematics and graphics software application, is a powerful tool for image processing. It allows comprehensive matrix manipulation and figure plotting. It is the primary tool used in this thesis to simulate and process the data for different phase retrieval methods. In MATLAB, data is usually represented as a matrix, images are no exception. A 2D image can be illustrated as a discrete format in Figure 3.1. The 2D matrix in MATLAB has two very intuitive coordinates called row and column, these coordinates start from 1 at the top left corner of the matrix, any element from the matrix M can be located by its index $M(row, column)$.

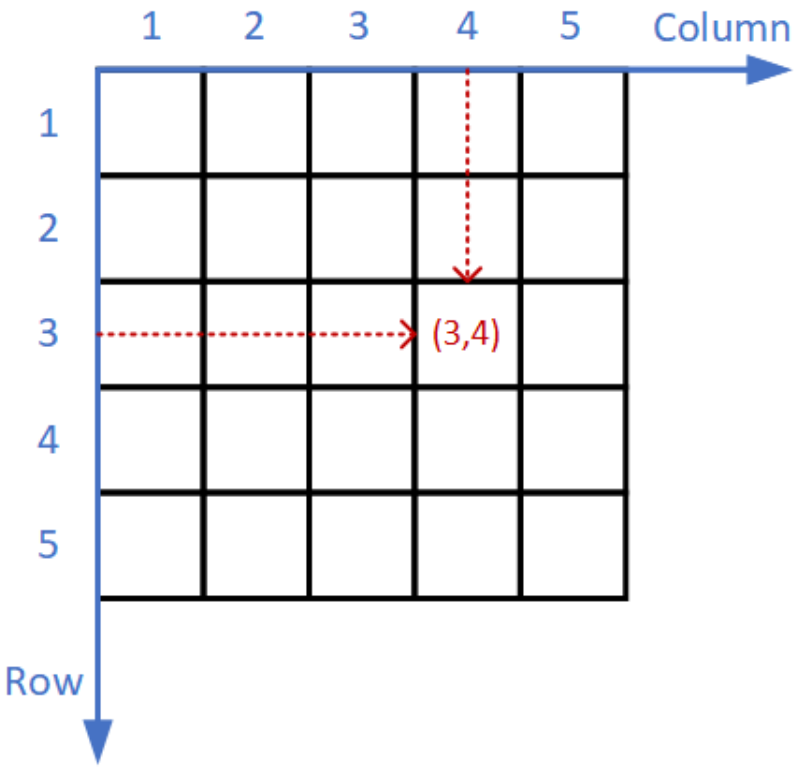

Figure 3.1. A 2D image matrix M, the data of each pixel is stored in each discrete square here. A specific pixel can be accessed by its coordinates, for example, the red dashed arrow shows the pixel $M(3,4)$.

This is the basic form in which the discrete data of an image exists in MATLAB, and all subsequent calculations and simulations in later section are based on this.

3.2. Modelling of Ptychography

To model ptychography on the computer, the highest priority is representing the related experiment data in a discrete format. The core data sets in ptychography, such as specimen, probe and diffraction can be represented as 2D complex images in the MATLAB, stored as matrices with complex values. The size of the diffraction pattern is determined by the detector; if the sensors on the detector are binned into 1024×1024 pixels, the diffraction pattern will be recorded as a matrix with 1024×1024 elements. The size of specimen usually is larger than diffraction pattern, because for each scan position, only a fraction

of the specimen is illuminated. With the known position index, this illuminated fraction can be easily cut from the large specimen matrix, then the aperture is applied to this subset of the specimen. A schematic of a moving aperture over a large discrete specimen is shown in Figure 3.2. On the detector plane, the diffraction pattern collected is the same size as the aperture.

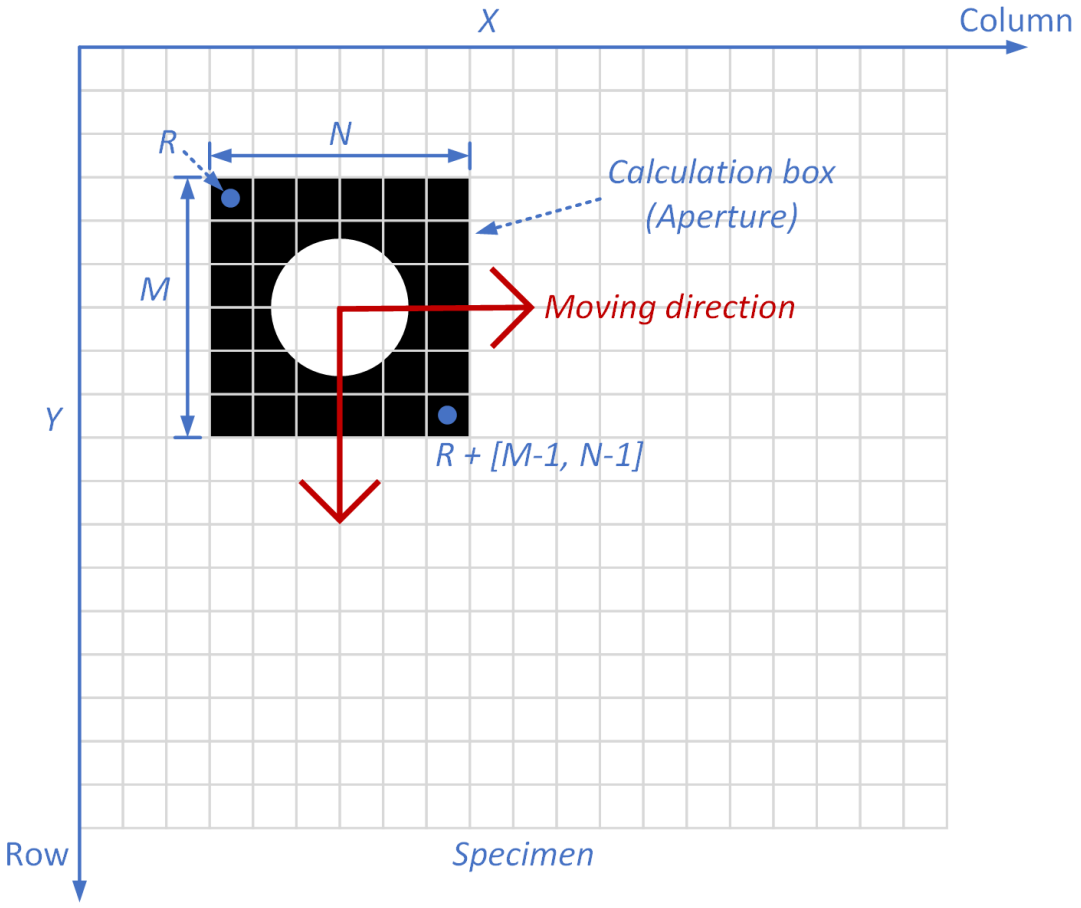

Figure 3.2. A moving aperture over a large discrete specimen, red arrow indicates the moving direction. The calculation box can be indicated as $R + [M-1, N-1]$ by the known top left pixel, where R is the top left pixel in the figure and M, N are the size of the aperture.

3.2.1. Pixel Pitch

Another important problem here is how to convert the scanning interval from physical distance into discrete pixels. In far-field ptychography, the conversion ratio dxy is defined as:

$$dxy = \frac{\lambda}{\theta} \tag{3.1}$$

where λ is illuminating wavelength in meters, θ is the span angle of the detector, usually can be estimated by:

$$\theta \approx \tan\theta = \frac{d}{l} \tag{3.2}$$

where d is the dimension of the detector, l is the distance between source and detector.

In a ptychography experiment, during the scanning process, small random offset will be added to avoid the ambiguity caused by the raster scan, which is referred to as raster scan pathology [39]; then, all the positions will be recorded as a vector or matrix, written as D. With the conversion ratio, the scanning position in pixel can be calculated by:

$$\vec{r} = \frac{D - D_{min}}{dxy} \tag{3.3}$$

where min represents the minimum value, and the position grid D is in metres. Since these positions cannot be exactly integer values of the pixel pitch in the reconstruction, when converted into this form, they will be rounded to the nearest integer value.

Now, with the unit conversion, the physical length distance becomes discrete pixels. The inverse conversion from a reconstructed image with pixels to real world distance also can be calculated by Equation (3.3), to provide a sensible scale bar.

3.2.2. The Formation of Object

With the appropriate pixel size, the specimen in real world can be converted to a 2D discrete matrix, which is usually called ‘Object’ in ptychography model. In

a ptychography experiment, the object with size $[X, Y]$, as shown in Figure 3.2, can be denoted as O. In the practice calculation, the $[X, Y]$ is usually larger than the size the of moveable aperture, $[M, N]$. Therefore, a fraction of the entire object should be cut from O, for a specific scan position k, denoted as:

$$o_k(\vec{r}) = O(\vec{r} + [M, N]) \tag{3.4}$$

where $\vec{r}$ represents pixel index.

3.2.3. The Formation of Probe

To form the interaction between the object and the wave from the source, the next step is to generate the probe function. Consider a coherent light source $\boldsymbol{S}(\vec{r})$, represented by a constant matrix; an aperture $\boldsymbol{A}(\vec{r})$, with a circular hole at the centre; and a lens $\boldsymbol{L}(\vec{r})$, contains phase information, see Equation (3.5), (3.6) and (3.7).

$$\boldsymbol{S}(\vec{r}) = s \tag{3.5}$$

$$\boldsymbol{A}(\vec{r}) = \begin{cases} 1, |\vec{r}| \leq \text{radius} \\ 0, |\vec{r}| > \text{radius} \end{cases} \tag{3.6}$$

$$\boldsymbol{L}(\vec{r}) = e^{-i\omega\vec{r}} \tag{3.7}$$

where $\vec{r}$ represents the position index in the matrix, s is a constant, $radius$ indicates the size of the aperture. The source light passing through the aperture, then converged by the lens, after a far-field propagation, creates a localised illumination which is the probe function, defined by:

$$P(\vec{r}) = \boldsymbol{\mathcal{F}}\{\boldsymbol{S}(\vec{r}) \times \boldsymbol{A}(\vec{r}) \times \boldsymbol{L}(\vec{r})\} \tag{3.8}$$

where $\times$ denotes the operation of the element-wise product or Hadamard product for two matrices.

3.2.4. The Formation of Exit Wave

According to Figure 2.7, the incident wave interacts with the specimen, and then the exit wave propagates to the detector field. When the incident wave passes through the specimen, the propagation speed will change according to the refractive index of the specimen. In ptychography, strong contrast arises from the real part of the refractive index, which is expressed in the phase of the transmission function [7]. The amplitude and phase change after passing through the specimen can be written as Equation (3.9) and $(3.10)(3.10)$:

$$A = A_0 e^{-\alpha d} \tag{3.9}$$

$$\Delta\varphi = \frac{2\pi n d}{\lambda} \tag{3.10}$$

where A is the amplitude after transmitted, A_0 is the amplitude of the incident wave, α is the attenuation coefficient depends on the material of specimen, d indicates the thickness of the specimen, n represents the refractive index of the specimen and λ is the wavelength of the incident wave. Assume the thickness is very tiny and tends to zero ($d \to 0$), both the changes of amplitude and phase in Equation (3.9) and (3.10) are negligible. The specimen or object function is indeed identical to the exit wave under the illumination, only if the specimen is infinitively thin [7]. However, as long as the specimen or object is thin enough in the direction of wave propagation, much thinner than the wavelength of the incident wave, a multiplicative approximation can be used to simplify the mathematical description of the interaction between the object and incident wave, making it more tractable. Therefore, with a simulated probe, the exit wave of one position can be simulated as a multiplication of probe and the cutout object function, written as:

$$\psi(\vec{r}) = P(\vec{r}) \times o(\vec{r}) \tag{3.11}$$

where $\vec{r}$ represents the position. The **Pseudocode 3.1** shows the process of the formation of exit waves for all scan positions. In this thesis, we only consider the 2D specimen that is optically thin enough, assume the multiplicative approximation is valid all the time.

Pseudocode 3.1: The Formation of Exit Waves

Inputs: *position vectors* (R), *specimen* (obj), *probe function* (probe), *probe size* (M, N), *the total number of positions* (K)
Outputs: *exit waves* (`exitWave`)

```
1   For (k = 1 to K) do
      // Form exit waves by the multiplication of object box with the probe
2     exitWave_k = probe · obj(R_k to R_k+[M-1,N-1])
3   End loop
```

3.2.5. The Formation of Diffraction Pattern

With the exit waves, the next step is to propagate them to the detector plane to generate the diffraction patterns, using the propagation theory described in previous section 2.5. In the far-field condition, the diffraction patterns are the intensity part of waves after the propagation, defined as:

$$I_k(\vec{u}) = |\mathcal{F}\{\psi_k(\vec{r})\}|^2 \tag{3.12}$$

where k indicates the scan position.

The **Pseudocode 3.2** describes the formation of the diffraction patterns which are the measured intensity on the detector. Also, **Pseudocode 3.1** and **Pseudocode 3.2** can be combined into one loop to optimise computational costs.

Pseudocode 3.2: The Formation of Diffraction Patterns

Inputs: *exit waves* (exitWave), *the total number of positions* (K)
Outputs: *intensity* (I)

```
For (k = 1 to K) do
  // Propagate exit waves to the detector plane
  detectorWave_k = ℱ(exitWave_k)
  Modulus_k = abs(detectorWave_k)
  I_k = (Modulus_k)^2
End loop
```

Note: $\mathcal{F}$: Fourier transform, **abs**: amplitude.

3.2.6. Revision of Exit Wave

In the previous section, the forward process in Figure 2.8, from real space to reciprocal space has been described; this is the essential step to generate the test data for a ptychography simulation. With a known probe and object, we can generate the diffraction patterns for a ptychographic experiment with specific setups. Therefore, we can simulate the experiment to see if the reconstruction fits the true probe and object.

To reconstruct the object and the probe with only the intensity from the diffraction patterns, the backward process is to replace the modulus with the measurements, as shown in Figure 2.8. This is the general idea of most iterative ptychography methods, although their methods for updating specimen and probe are different. These differences will be discussed in detail in the later chapters; here, a simple pseudocode is provided to show the process, see **Pseudocode 3.3**.

Pseudocode 3.3: The Revision of Exit Waves

Inputs: *exit waves* (`exitWave`), *intensity* (`I`), *the total number of positions* (K)
Outputs: *revised exit waves* (`RevisedexitWave`)

```
For (k = 1 to K) do
  // Propagate exit waves to the detector plane
  detectorWave_k = ℱ(exitWave_k)
  // Replace the modulus with measured intensity
  correctedWave_k = sqrt(I_k)·detectorWave_k / (abs(detectorWave_k) + eps)
  // Back propagate
  RevisedexitWave_k = ℱ^{-1}(correctedWave_k)
End loop
```

Note: $\mathcal{F}$ and $\mathcal{F}^{-1}$: Fourier and inverse Fourier transform. **sqrt**: square root, **abs**: amplitude. ***eps***: a small constant in MATLAB to avoid dividing 0.

3.2.7. Update the Object and Probe (ePIE)

With the revised exit waves from last section, there are many ways to separate the object and probe from the multiplication. More details about these methods for different algorithms will be discussed in Chapter 5. Here, we use the extended ptychographic Iterative engine (ePIE) as an example to demonstrate the updating step [17]. An important improvement of ePIE is that it does not require a known probe function at the beginning. In other words, it can solve the probe function as well as the object by introducing a probe updating step in the algorithm. The updating function of object and probe is written in Equation (3.13) and (3.14).

$$o_{k_{j+1}}(\vec{r}) = o_{k_j}(\vec{r}) + \alpha \frac{P_j^*(\vec{r})}{\left|P_j(\vec{r})\right|^2_{max}} \left(\psi_{k_j}'(\vec{r}) - \psi_{k_j}(\vec{r})\right) \tag{3.13}$$

$$P_{j+1}(\vec{r}) = P_j(\vec{r}) + \beta \frac{o_{k_j}^*(\vec{r})}{\left|o_{k_j}(\vec{r})\right|^2_{max}} \left(\psi_{k_j}'(\vec{r}) - \psi_{k_j}(\vec{r})\right) \tag{3.14}$$

where $P_j(\vec{r})$ and $o_{k_j}(\vec{r})$ are the reconstructed probe and object in j^{th}

iteration and k is the scan position. $\psi_{k_j}'(\vec{r})$ is the revised exit wave and $\psi_{k_j}(\vec{r})$ is the original one. max represents the maximum value and $*$ is the complex conjugate. The constant α and β are the tuning parameters to alter the step-size of the update where ePIE use 1 for both.

Combined with the steps in previous sections, the flow chart of ePIE is expressed in the Figure 3.3 as well as the pseudocode in **Pseudocode 3.4**. The initial guesses of probe and object is required for the first iteration of the algorithm. A free-space guess is usually used for the initial object which simply be a matrix with value 1 everywhere, and a support of the roughly correct size to generate the initial probe guess.

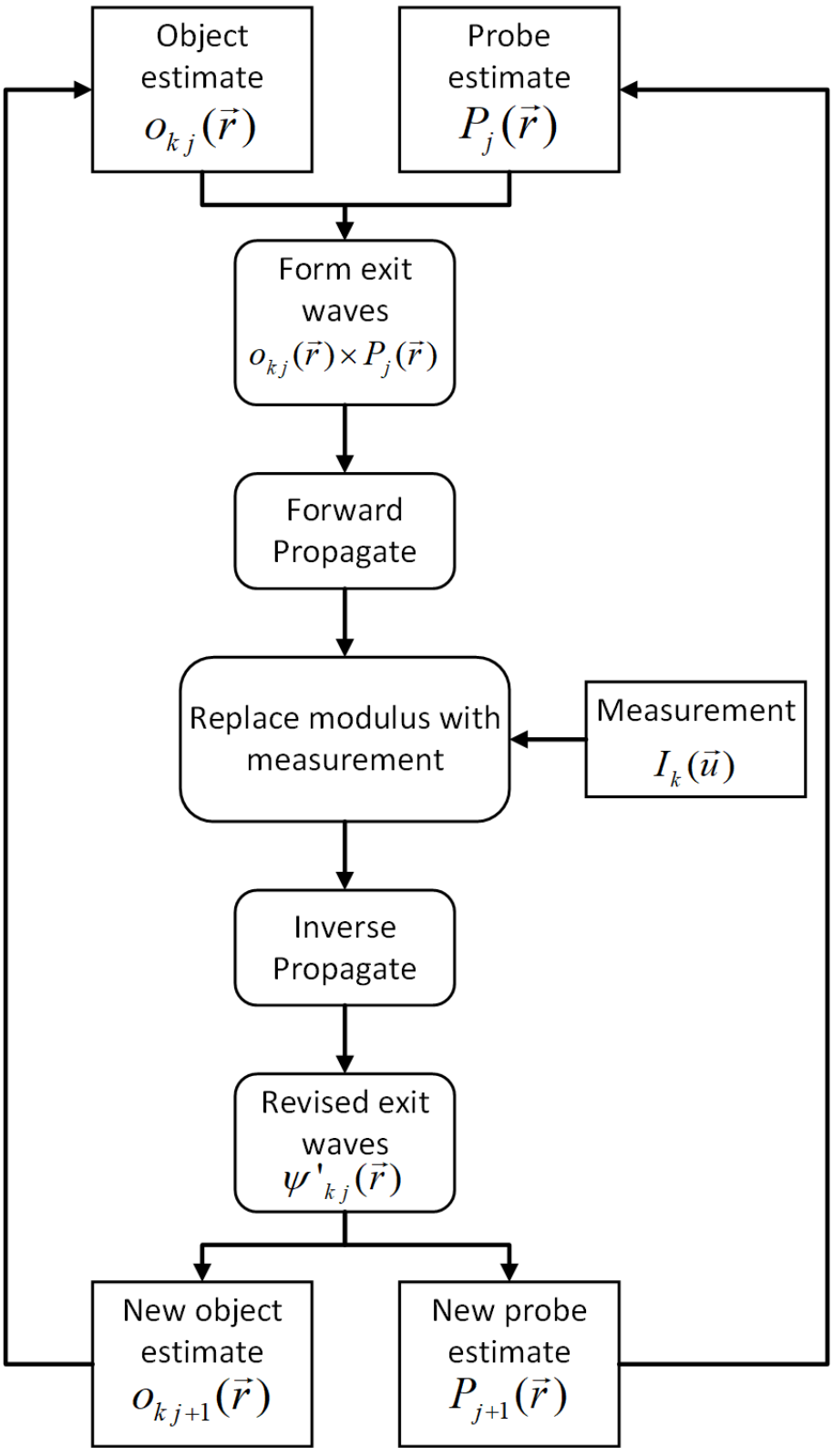

Figure 3.3. The flow chart of ePIE algorithm

Pseudocode 3.4: ePIE algorithm

Inputs: *position vectors* (R), *object* (obj), *probe function* (probe), *probe size* (M, N), *intensity* (I), *the total number of positions* (K), *the total number of iterations* (J), *tuning parameter* (α, β)

Outputs: *reconstructed object* (obj), *reconstructed probe* (probe)

```
For (j = 1 to J) do
  // Random shuffle all the positions
  R = shuffle(R)
  For (k = 1 to K) do
    // Form the exit wave
    objBox = obj(R_k to R_k+[M-1,N-1])
    exitWave_k = probe · objBox
    // Propagate exit wave to the detector plane
    detectorWave_k = ℱ(exitWave_k)
    // Replace the modulus with measured intensity
    correctedWave_k = sqrt(I_k)·detectorWave_k /(abs(detectorWave_k) + eps)
    // Back propagate
    RevisedexitWave_k = ℱ^-1(correctedWave_k)
    // Update the object and probe
    ΔexitWave_k = RevisedexitWave_k - exitWave_k
    obj(R_k to R_k+[M-1,N-1]) += α·conj(probe)·ΔexitWave_k/max(abs(probe)^2)
    probe += β·conj(objBox)·ΔexitWave_k/max(abs(objBox)^2)
  End loop
End loop
```

Note: ***shuffle:*** *a function that randomly change the order of the position sequence.* $\mathcal{F}$ and $\mathcal{F}^{-1}$**:** Fourier and inverse Fourier transform. ***eps:*** a small constant in MATLAB to avoid dividing 0. **sqrt**: square root. **abs**: amplitude. **conj**: complex conjugate. **max**: maximum value.

ePIE is a very classic approach to ptychography, and its process can, to some extent, represent a way of modelling and solving ptychographic problems on the computer, but of course, other algorithms will have a lot of differences with ePIE, which will be mentioned in detail in later chapters.

3.3. Ambiguities in the Reconstruction

The last section has described the basic model of ptychography for simulations with ePIE algorithm. In the simulations, there are several ambiguities between the reconstruction and the true specimen and probe, for instance, a constant

amplitude scaling, a constant phase offset, a global real space translation, a linear phase ramp and the periodic artefacts [18]. The periodic artefact is ignored here, because this is caused by the raster pathology, as a small offset is applied during the scan to erase this effect. Assume a wave after propagation $\Psi_{\mathrm{k}_j}(\vec{u})$ at scan position k in j^{th} iteration, all the ambiguities mentioned above can be explained in Equation (3.15):

$$\Psi_{\mathrm{k}_j}(\vec{u}) = \boldsymbol{\mathcal{F}}\left\{P_j(\vec{r})o_{k_j}(\vec{r})\right\} = \boldsymbol{\mathcal{F}}\left\{\left(ae^{ic}e^{i\vec{b}\cdot\vec{r}}\hat{P}\left(\vec{r}+\vec{d}\right)\right)\left(a^{-1}e^{-ic}e^{-i\vec{b}\cdot\vec{r}}\hat{O}_k\left(\vec{r}+\vec{d}\right)\right)\right\} (3.15)$$

where $P_j(\vec{r})$ and $o_{k_j}(\vec{r})$ are the reconstructed probe and object, $\hat{P}$ and $\hat{O}$ are the true probe and object, a and c are scalar constants, $\vec{b}$ and $\vec{d}$ are constant vectors.

3.3.1. Global Translation

The global translation ambiguity embodied as vector $\vec{d}$ in Equation (3.15) is caused by shifts in the ptychographic measurement. The ptychography reconstruction process attempts to recover object's structure by combining the information from all the diffraction patterns located at different positions. This means that different combinations of shifts can produce the same ptychographic data, leading to ambiguity in determining the actual object at a specific position. The global translation ambiguity will give a shifting effect onto the exit waves in real space, resulting a phase ramp after it is propagated to the reciprocal space. Since applying the measurements only corrects the modulus in reciprocal space, the phase ramp will be retained and affect the final reconstruction.

3.3.2. Phase Ramp

The second ambiguity is the phase ramp in the reconstruction. In blind

ptychography where the probe and object are both unknown, the reconstructed estimates of the object and probe can produce the same diffraction data as the original probe and object, but both with the phase ramp. In other words, a pair of phase shifts that can be applied to both estimates without changing the resulting diffraction patterns. This pair of phase ramps is not unique and can vary depending on the reconstruction algorithm and initial conditions. A phase ramp and its counter ramp are represented by the term $e^{i\vec{b}\cdot\vec{r}}$ and $e^{-i\vec{b}\cdot\vec{r}}$ in Equation (3.15). Their product is equal to the unit, which has no impact on the diffraction estimate.

3.3.3. Complex Scaling

In Equation (3.15), another term ae^{ic}, which usually results in complex scaling ambiguity. Different from the other two, the complex scaling ambiguity has no significant impact on the final reconstruction results. It is only a complex factor that scales the reconstruction. The modulus part a affects the amplitude, and the phase part e^{ic} will introduce a phase offset to the reconstruction. In the real experiment, only the relative phase is important to determine the structure of the specimen. This phase offset caused by complex scaling is global and does not affect the relative phase of the reconstruction. However, complex scaling ambiguity will affect the error metric calculation in the simulation since the reconstruction will be compared with the ground truth in the simulation, which will be discussed in the later section.

3.3.4. Remove Ambiguities

In order to get a more accurate reconstruction in the simulation, we can remove these ambiguities since the ground truth of the object and probe is known in the simulation. Firstly, the global translation is the main reason for the fluctuation at the beginning of the reconstruction. The global translation can be estimated through an indispensable method from Guizar-Sicairos [40]. A cross-correlation

between reconstructed probe amplitudes and the ground truth, to sub-pixel precision, can calculate the shift vector. Once the shifting vector is estimated, a counter shifting can be applied to remove the ambiguity. This is the precursor to the correction of the rest of the ambiguities.

After removed the global translation, the phase ramp ambiguity can be calculated by multiplying the reconstructed object with the conjugate of the true object without the global translation ambiguity, shown in Equation (3.16):

$$\begin{aligned}\angle\left\{(a^{-1}e^{-ic}e^{-i\vec{b}\cdot\vec{r}}\hat{O}(\vec{r}))\times\hat{O}^{*}(\vec{r})\right\} &= \angle\left\{a^{-1}e^{-ic}e^{-i\vec{b}\cdot\vec{r}}\left|\hat{O}(\vec{r})\right|^{2}\right\}\\ &= \angle\left\{a^{-1}\left|\hat{O}(\vec{r})\right|^{2}e^{-i(\vec{b}\cdot\vec{r}+c)}\right\}\\ &= -\vec{b}\cdot\vec{r}-c\end{aligned} \tag{3.16}$$

where $\angle$ represents the phase part of a complex value. Take the derivation of Equation (3.17) with respect to $\vec{r}$ can give out the vector $\vec{b}$ which indicates the phase ramp. With the approximation of $\vec{b}$, a counter phase ramp can be produced and applied to balance and remove the phase ramp ambiguity.

The final step is removing the complex scaling ambiguity. For a more accurate error analysis, the complex scaling factor can be estimated by Equation (3.17):

$$ae^{ic}\approx\frac{\sum_{\vec{r}}\bar{O}_{J}(\vec{r})\,\hat{O}^{*}(\vec{r})}{\sum_{\vec{r}}\left|\hat{O}(\vec{r})\right|^{2}} \tag{3.17}$$

where $\bar{O}_{J}(\vec{r})$ represents the object reconstruction after removing the global translation and phase ramp.

The process of removing ambiguities is illustrated in the Figure 3.4.

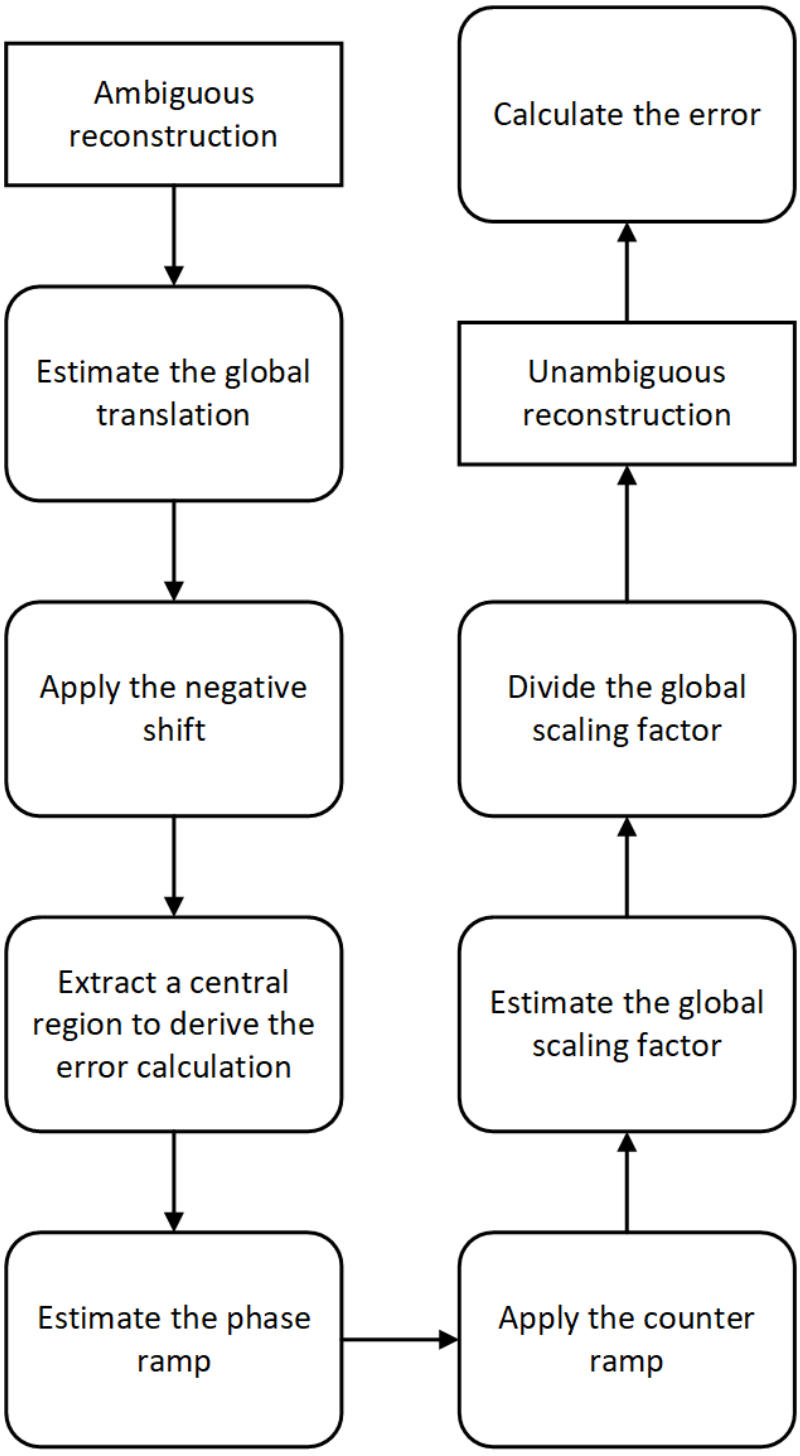

Figure 3.4. The flow chart of removing ambiguities.

3.4. Error Metric

3.4.1. Simulation Error

After removing all the ambiguities mentioned above, error metrics for the

simulation are designed to quantitatively evaluate the quality of reconstructed images by comparing them with the available ground truths, defined as Equation (3.18). Both the modulus and phase difference between two complex values are directly under comparison, which gives no uncertainty for the error. However, this error metric is only capable when the true object and probe is known in a simulation.

$$E_{sim} = \frac{\sum_{\vec{r}} \left| \hat{O}(\vec{r}) - \overline{O}_j(\vec{r}) \right|^2}{\sum_{\vec{r}} \left| \hat{O}(\vec{r}) \right|^2} \tag{3.18}$$

where $\overline{O}_j(\vec{r})$ represents the object reconstruction in j^{th} iteration after removing all the ambiguities.

3.4.2. Diffraction Intensity Error

In a real ptychography experiment, the prior knowledge of object and probe are usually unavailable, the simulation error in last section is invalid as the ambiguities cannot be removed anymore. Therefore, here, we introduce another error metric called diffraction intensity error to calculate the normalized mean square error (MSE) between the diffraction intensities and the measurements, defined as Equation (3.19):

$$E = \frac{\sum_k \sum_{\vec{u}} \left| \left| \Psi_{k_j}(\vec{u}) \right|^2 - I_k(\vec{u}) \right|^2}{\sum_k \sum_{\vec{u}} |I_k(\vec{u})|^2} \tag{3.19}$$

4. Wigner Distribution Deconvolution (WDD)

In section 2.9.1, a direct ptychography solution called Wigner Distribution Deconvolution (WDD) was mentioned and some achievements based on WDD were briefly reviewed. In this chapter, the details of Wigner Distribution Deconvolution (WDD) will be mathematically explained. Then, a new approach of WDD that can solve the probe function as well as the object function will be introduced and tested with some simulation data.

4.1. Definition of Wigner Distribution Deconvolution

Unlike iterative methods, WDD can be considered as a direct closed solution of ptychography. This means that WDD is not a minimisation problem, but a problem similar to solving equations. WDD requires a more intensive scan grid for ptychography compared to iterative methods. Assume an object with the number of pixels $[X, Y]$, iterative methods normally have a probe with size $[M, N]$, where $[M, N]$ are smaller than $[X, Y]$. The size of the scan grid will also be smaller than $[X, Y]$, depending on the percentage of overlapping area. By contrast, the scan grid for WDD will be the same size as $[X, Y]$, which means one pixel size step during the scanning process. Also, the size of the probe will be the same as object. This results in a massive dataset for WDD and the heavy computation for reconstruction.

4.1.1. Notes on Nomenclature

Because WDD treats this massive dataset in a 4D format, a different nomenclature will be used in this chapter for better understanding, compared to other chapters of this thesis. A typical ptychography imaging system was displayed before, in Figure 2.7. As mentioned above, in WDD, there is no need

to extract a fraction of the object for the calculation of a specific diffraction pattern, since the size of the probe and diffraction pattern is $[X, Y]$, same as the object. The nomenclature for the object and probe function is shown in the Table 1.

Table 1. The new nomenclature for WDD

	Object function	Probe function	Detector plane coordinate	Scan position coordinate
Real space	$o(r)$	$p(r)$	r: $[r_x, r_y]$	R: $[R_X, R_Y]$
Reciprocal space	$O(u)$	$P(u)$	u: $[u_v, u_w]$	U: $[U_V, U_W]$

Notice that the r, R, u and U are 2D coordinates, and the functions in reciprocal space are the Fourier transforms of the functions in the real space: $O(u) = \boldsymbol{\mathcal{F}}\{o(r)\}$, $P(u) = \boldsymbol{\mathcal{F}}\{p(r)\}$. All the multiplication operations between matrixes, except those specifically stated, are element-wise product.

4.1.2. Mathematical Definition

With the new nomenclature, the I-set in WDD which is the intensity $I(\vec{u})$ in Equation (3.12), now can be rewritten as Equation (4.1):

$$I(u, R) = |\boldsymbol{\mathcal{F}}_{\mathrm{r}}\{p(r - R)o(r)\}|^2 \qquad (4.1)$$

where the exit wave is formed by the multiplication of object and a shifted probe and $\boldsymbol{\mathcal{F}}_{\mathrm{r}}$ represents the Fourier transform with respect to r. Because both u and R are 2D coordinates, the intensity $I(u, R)$ is actually in the form of $I(u_x, u_y, R_X, R_Y)$. This the presentation of 4D dataset in WDD, but here, we will abridge it as $I(u, R)$ in the later text.

In Equation (4.1), the scanning process is indicated by the shifted probe function. The shift of the probe in real space can be considered a phase ramp added to its Fourier transform in reciprocal space according to Fourier shift

theorem. Also, based on the convolution theorem, the Fourier transform of a multiplication of two real space functions, like Equation (4.1), can be written as a convolution of their Fourier transforms. Therefore, the intensity in Equation (4.1) can be rewritten as:

$$I(u,R) = \left|P(u)e^{-i2\pi Ru} \otimes_u O(u)\right|^2 \tag{4.2}$$

where $O(u) = \boldsymbol{\mathcal{F}}\{o(r)\}$ and $P(u) = \boldsymbol{\mathcal{F}}\{p(r)\}$. $e^{-i2\pi Ru}$ represents the phase ramp caused by the probe shift in real space. $\otimes_u$ is the convolution operator along the u direction. To calculate the amplitude in Equation (4.2), we can write it as a conjugate product:

$$I(u,R) = \left[P(u)e^{-i2\pi Ru} \otimes_u O(u)\right] \times \left[P(u)e^{-i2\pi Ru} \otimes_u O(u)\right]^* \tag{4.3}$$

where $*$ is the complex conjugate operator.

Note that the definition of the convolution for two functions $f(t)$ and $g(t)$ is given by Equation (4.4):

$$f(t) \otimes g(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{4.4}$$

where τ is the variable of integration.

Here, this will be extended to a 2D application, the convolution in Equation (4.3) also can be expanded by its definition:

$$\begin{aligned} I(u,R) &= \int P(u_a)O(u-u_a)e^{-i2\pi Ru_a}du_a \times \int P^*(u_b)O^*(u-u_b)e^{i2\pi Ru_b}du_b \\ &= \iint P(u_a)O(u-u_a)P^*(u_b)O^*(u-u_a)e^{-i2\pi R(u_a-u_b)}du_a du_b \end{aligned} \tag{4.5}$$

where the conjugate of the convolution of two functions is equal to the convolution of their respective conjugates. u_a and u_b are the variables of

integration, range from $-\infty$ to ∞.

A Fourier transform of $I(u,R)$ with respect to the coordinate R forms another important dataset in WDD called G-set, defined as Equation (4.6):

$$G(u,U) = \boldsymbol{\mathcal{F}}_R\{I(u,R)\} = \int I(u,R)e^{-i2\pi RU}\,dR \qquad (4.6)$$

Substituting Equation (4.5) into Equation (4.6), then gives, the definition of the G-set in the form of the triple integral, shown in Equation (4.7):

$$\begin{aligned} G(u,U) &= \iiint P(u_a)O(u-u_a)P^*(u_b)O^*(u-u_a)e^{-i2\pi R(u_a-u_b)}e^{-i2\pi RU}du_a du_b dR \\ &= \iiint P(u_a)O(u-u_a)P^*(u_b)O^*(u-u_a)e^{-i2\pi R(u_a-u_b+U)}du_a du_b dR \end{aligned} \qquad (4.7)$$

Because the illumination from the source and the specimen both stay the same all the time in ptychography, the Fourier transform of the probe $P(u)$ and the Fourier transform of the object $O(u)$ have no dependence on the scan position vector R. The integration over R only relates to the exponential term in Equation (4.6). If $R \neq 0$, a fact is that this integral of the complex exponential term equals to 0 over an infinite range. Otherwise, if $R = 0$, the exponential term equals to 1 and the integral diverges. Hence, the integration over R gives a delta function, the G-set can be written as Equation (4.8):

$$G(u,U) = \iint P(u_a)O(u-u_a)P^*(u_b)O^*(u-u_b)\delta(u_a-u_b+U)du_a du_b \qquad (4.8)$$

where δ represents the delta function, which has a value of zero everywhere, except when $u_a - u_b + U = 0$. Under this condition, we can continue to integrate over either u_a or u_b. For example, integrating over u_b, substituting $u_b = u_a + U$, gives Equation (4.9):

$$G(u,U) = \int P(u_a)O(u-u_a)P^*(u_a+U)O^*(u-u_a-U)du_a \qquad (4.9)$$

For a clearer representation, we can substitute $\tau = u - u_a$ into the Equation (4.9), rearrange it as Equation (4.10):

$$G(u,U) = \int O(\tau)O^*(\tau - U)P(u - \tau)P^*(u + U - \tau)d\tau \qquad (4.10)$$

According to the definition of convolution in Equation (4.4), Equation (4.10) can be considered as a convolution form of two terms, see Equation (4.11):

$$G(u,U) = O(u)O^*(u - U) \otimes_u P(u)P^*(u + U) \qquad (4.11)$$

So far, it is clear to see that $O(u)O^*(u - U)$ only depends on the object and $P(u)P^*(u + U)$ only depends on the probe. That means the information of object and probe have been separated to some extent, so the next key step is to deconvolve these two terms.

According to the convolution theorem, the convolution along the u direction of the G-set can be rewritten as a product of two Fourier transforms in reciprocal space with respect to u, which is usually called the "H-set", defined by Equation (4.12):

$$\begin{aligned} H(r,U) &= \boldsymbol{\mathcal{F}}_u^{-1}\{O(u)O^*(u - U) \otimes_u P(u)P^*(u + U)\} \\ &= \boldsymbol{\mathcal{F}}_u^{-1}\{O(u)O^*(u - U)\} \times \boldsymbol{\mathcal{F}}_u^{-1}\{P(u)P^*(u + U)\} \\ &= \int O(u)O^*(u - U)e^{i2\pi ru}du \times \int P(u)P^*(u + U)e^{i2\pi ru}du \end{aligned} \qquad (4.12)$$

We introduce the definition of the Wigner distribution to simplify Equation (4.12). For a general reciprocal function F, the Wigner distribution, which is also called the ambiguity function in signal processing theory [7], is given by Equation (4.13):

$$\chi_F(r,U) = \int F(u)F^*(u - U)e^{i2\pi ru}\,du \qquad (4.13)$$

Hence, we can rewrite the H-set as:

$$H(r,U) = \chi_O(r,U)\chi_P(r,-U) \tag{4.14}$$

From the G-set to the H-set, the convolution operation has been simplified into a product of χ_O and χ_P. Assuming the probe function is known, it is easy to form the χ_P by the definition of Wigner distribution, so we can deconvolve the probe part from the H-set by Equation (4.15):

$$\chi_O(r,U) = \frac{H(r,U)}{\chi_P(r,-U)} \tag{4.15}$$

This division will be very unstable when χ_P is very small or zero, therefore, a Wiener filter is introduced to regulate the division operation:

$$\chi_O(r,U) = \frac{\chi_P^*(r,-U)H(r,U)}{|\chi_P(r,-U)|^2 + \varepsilon} \tag{4.16}$$

where ε is a small constant.

Now, $\chi_O(r,U)$ is separated from $H(r,U)$. According to the definition of the Wigner distribution, $\chi_O(r,U)$ can be written as Equation (4.17):

$$\chi_O(r,U) = \int O(u)O^*(u-U)e^{i2\pi ru}du \tag{4.17}$$

Apply the Fourier transform with respect to r to Equation (4.17), gives a new set called the “D-set” in Equation (4.18):

$$\begin{aligned} D(u,U) &= \boldsymbol{\mathcal{F}}_r\{\chi_O(r,U)\} \\ &= O(u)O^*(u-U) \end{aligned} \tag{4.18}$$

The D-set is a function that only relates to the Fourier transform of the object function, can be considered as a convolution between the object function in real space and its conjugate shift. A full derivation of deconvolving and reconstructing the object function from D-set will be introduced in the next section. So far, all the important data sets in WDD have been described, the

relationship between them is shown in Figure 4.1.

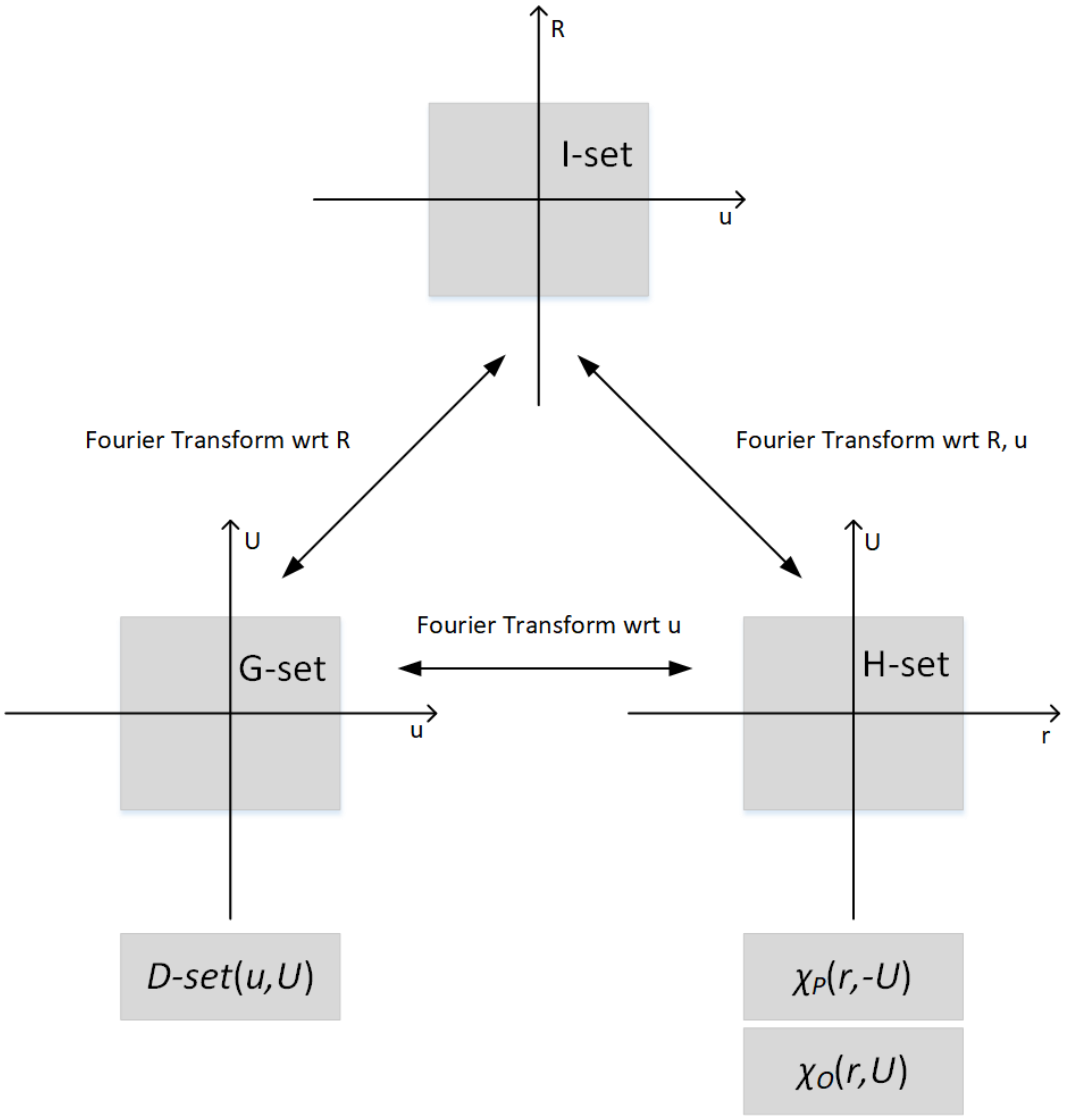

Figure 4.1. The relationship between I-set, G-set and H-set, and the coordinates systems where D-set, χ_P and χ_O belong to.

In Figure 4.1, the I-set is the measurement taken from the detector, and the G-set and H-set are the Fourier transforms of the measured data along different directions. The D-set has the same coordinate system as the G-set, and χ_P, χ_O are the same as the H-set. In Figure 4.1, these data sets appear as 2D axes, but in a real ptychography experiment, any one of these coordinates actually covers the 2D plane, which makes these data sets 4D. In general, the WDD process is shown in the flow chart of Figure 4.2, and **Pseudocode 4.1**.

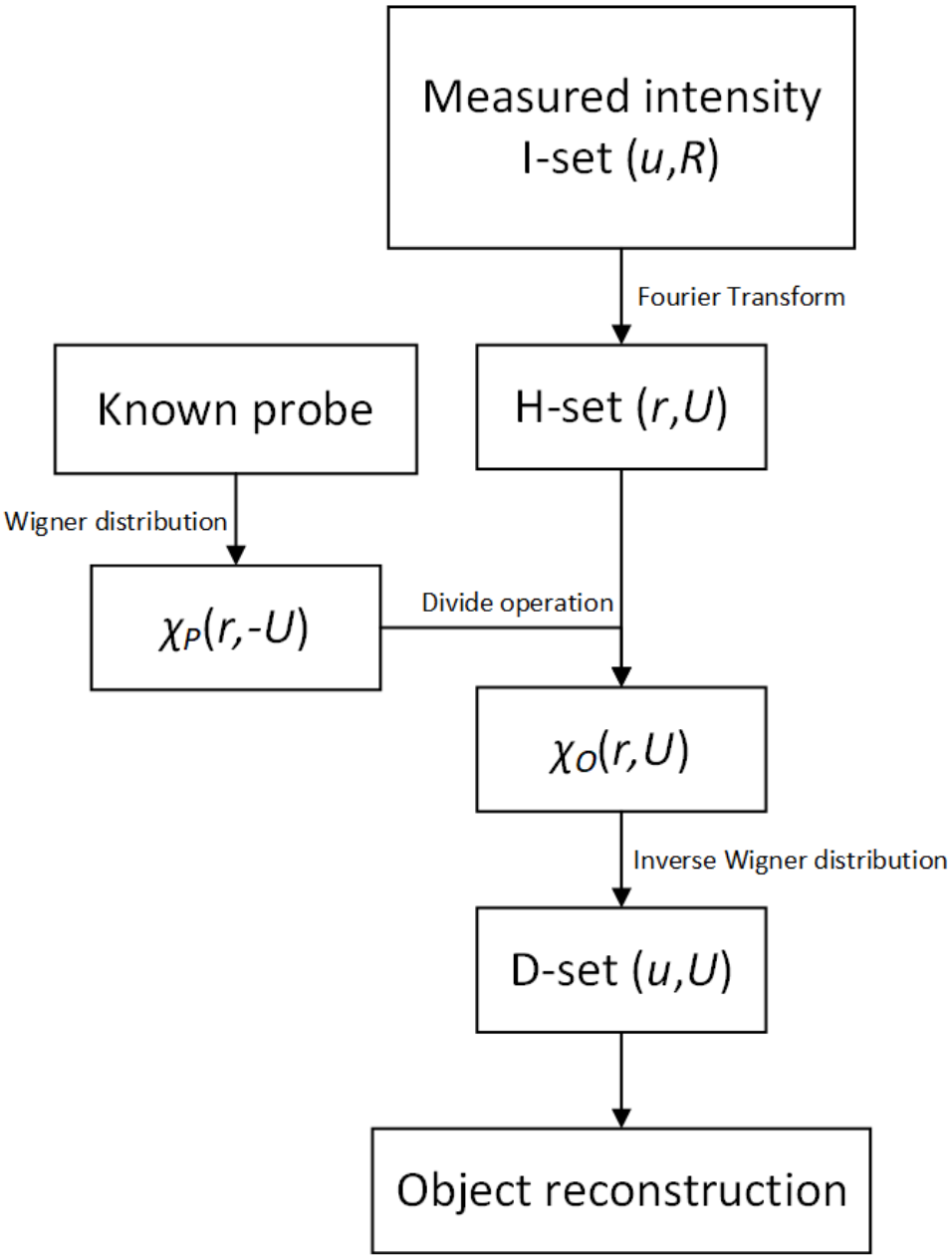

Figure 4.2 The flow chart of WDD process.

Pseudocode 4.1: Basic WDD Process

Inputs: *probe function* (probe), *probe size in reciprocal space* (U), *intensity* (I), *small constant* (epsilon)

Outputs: *D-set* (D)

```
    // Form the G-set from the I-set
1   G = ℱ_R(I)
    // Form the H-set from the G-set
2   H = ℱ_u(G)
    // Calculate the Fourier transform of the probe
3   P = ℱ(probe)
4   For (t = 1 to U) do
        // Form the Wigner distribution of the probe
6       χ_P = P · conj(shift(P, -t))
8   End loop
    // Calculate the Wigner distribution of the object
9   χ_O = conj(χ_P) · H / (abs(χ_P)² + epsilon)
    // Calculate the D-set
10  D = ℱ_r(χ_O)
    // Further calculation to reconstruct the object function from the D-set
```

Note: **shift**: *a function that shifts the input.* $\mathcal{F}$: Fourier transform. **abs**: amplitude. **conj**: complex conjugate.

It is distinct in **Pseudocode 4.1** that WDD is a non-iterative method, which is highly desirable in terms of computation time. However, the final reconstruction is limited by effects of the partial coherence, experimental instabilities and the finite extent of the lens [41]. These factors result in a cut-off frequency in the back focal plane of the lens. They are also the reason that bright field imaging via a conventional microscope has an information limitation. In the WDD, because of this defect, the information from $\chi_P(r,-U)$ is also limited along the U direction when we form it. If we denote the cut-off frequency as U_{max}, the support size will be $2U_{max}$ for $\chi_P(r,U)$ in the U direction. Therefore, after deconvolution, $\chi_O(r,U)$ will be limited within the finite support size, which causes the loss of high frequency information in the reconstruction. Although we cannot directly retrieve the information beyond the cut-off frequency in $\chi_P(r,-U)$ and $\chi_O(r,U)$, it does not mean that they are lost from the beginning. The detector is able to collect all the information along the u direction, in other

words, although the support size of $\chi_P(r, U)$ is restricted vertically along the U direction, after deconvolution, all this high frequency information is still stored in the D-set, horizontally lying along the u direction. Hence, to improve reconstruction resolution, further algorithm will be introduced to retrieve this information in the next section.

4.2. Wigner Distribution Deconvolution Reconstruction

This section will introduce methods to reconstruct the object beyond the cut-off frequency limitation. Start with the oldest one, called “stepping out” [32], and then its improved version, called “projection strategy” [33]. Finally, we will propose a new approach based on “projection strategy” to solve the blind deconvolution, which means the probe function is also unknown in the experiment.

4.2.1. Stepping Out

As mentioned above, in WDD, the reconstruction resolution is restricted by the cut-off frequency. To recover high frequency components of the object, there is a way to step out the horizontal information from the D-set [32]. To explain this process more properly, here we start from a 1D example since it is easy to be represented by images. However, 1D phase retrieval problem is more difficult to solve than 2D. In addition to the trivial transformations such as rotation, translation, and conjugate reflection, the 1D phase retrieval problem usually has many non-trivial solutions, and these solutions may differ significantly from the true signal [42]. Moreover, 1D phase retrieval problem is a non-convex optimization problem which is difficult to solve directly [42]. Assume there is a simple 1D object with only five elements periodically scanned pixel by pixel at five positions, its Fourier transform is expressed as Equation (4.19):

$$O(u) = [A, B, C, D, E] \tag{4.19}$$

According to Equation (4.18), the D-set of this 1D object can be simply considered as the product of the Fourier transform of object and a shifted Fourier transform of the object's conjugate. This can be schematically represented in Figure 4.3, where each block represents a pixel point, and the grey blocks are the unavailable information beyond cut-off frequency.

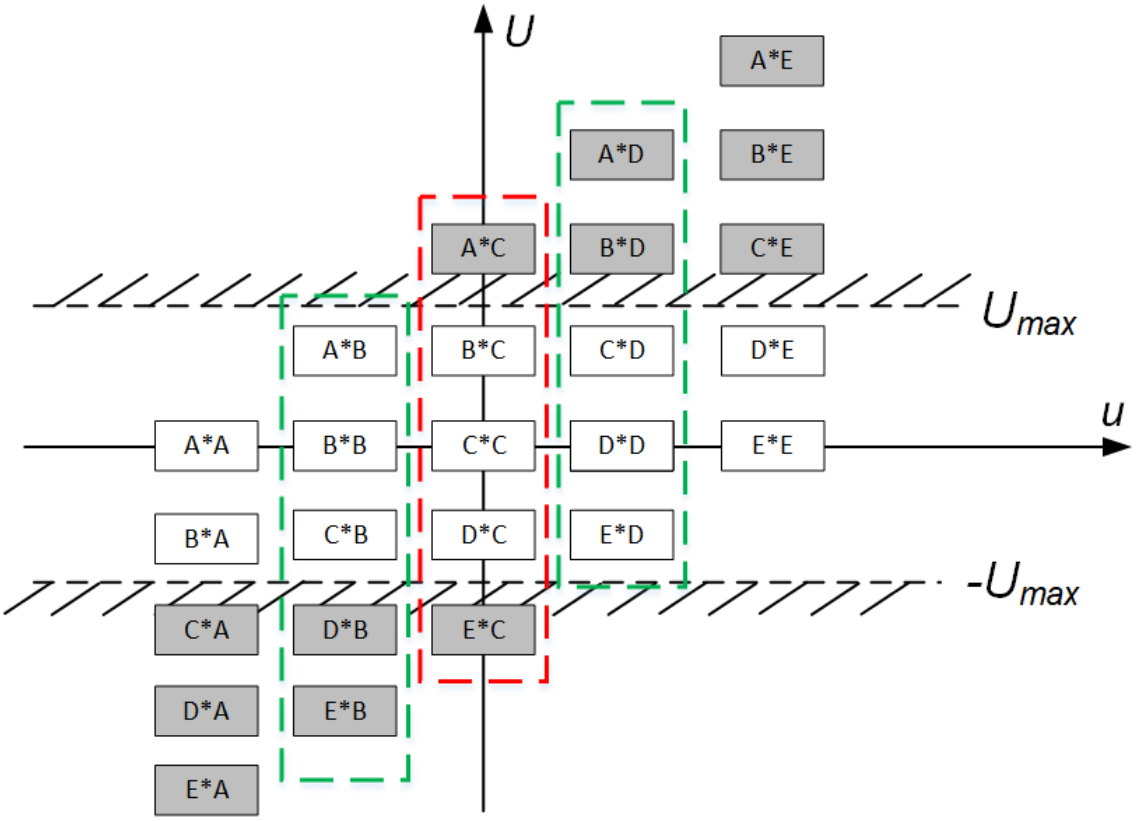

Figure 4.3. Illustration of the D-set with a simple 1D object. Red dashed box represents $D(0,U)$. Green dashed boxes represent $D(-1,U)$ and $D(1,U)$. U_{max} and $-U_{max}$ indicate the cut-off frequency.

In Figure 4.3, the origin point only relates to the values of C, see Equation (4.20):

$$D(0,0) = C^*C = |C|^2 \qquad (4.20)$$

The origin point $D(0,0)$ is known, therefore, we can assign an arbitrary phase to C to calculate the value of C, see Equation (4.21):

$$C = \sqrt{D(0,0)} \tag{4.21}$$

Then, with the value of C, we can calculate the central vertical line in the D-set by setting $u = 0$ in the D-set, as illustrated by the red dashed box in Figure 4.3. This step can be expressed as Equation (4.22):

$$O(u)_0 = \left(\frac{D(0,U)}{C}\right)^* = [0, B, C, D, 0] \tag{4.22}$$

$O(u)_0$ is a temporary reconstruction at $u = 0$. In this case, we can only get the value of B and D, because A and E lie beyond the cut-off frequency limitation. However, if we look at the vertical lines beside the central red box, which is illustrated in the green dashed boxes in Figure 4.3, A^* and E^* are now inside the support size as a product with B and D. Therefore, we can now figure out A and E, using the value of B and D come from Equation (4.22), see Equation (4.23) and Equation (4.24):

$$O(u)_{-1} = \left(\frac{D(-1,U)}{B}\right)^* = [A, B, C, 0, 0] \tag{4.23}$$

$$O(u)_1 = \left(\frac{D(1,U)}{D}\right)^* = [0, 0, C, D, E] \tag{4.24}$$

For a larger object with more elements, use the value of A and E to "step out" the next column to a higher frequency and then repeat. This step size is determined by the cut-off frequency U_{max}. A larger size example is illustrated in Figure 4.4. Firstly, calculate the central column shown in the black box using the value of central point S. Then, take the first element of this central reconstruction and find its corresponding position on the u axis, see the green indication in Figure 4.4. The new point B lies on the u axis is the square of A. Note that $d(A,S) = d(B,S)$, where d represents the distance between two

points. Also, this distance is the same as the half length of the black box, which is the cut-off frequency U_{max}. Point B indicates a vertical line, which is the farthest one we can get from the black box. All the points above B on this vertical line, illustrated by the green box in Figure 4.4, will be the new points beyond the black box, indicated by the blue box in Figure 4.4. Therefore, we can retrieve these new points like Equation (4.23). Then, take the new point C and repeat for the next "stepping out". Similarly, on the right side of the black box, take the bottom point and carry out the same steps to retrieve the points below $-U_{max}$, illustrated by the red indication in Figure 4.4. Combining them together, one "stepping out" can extend $2U_{max}$ length of the reconstruction of the object's Fourier transform, which improves the high frequency part of the object reconstruction.

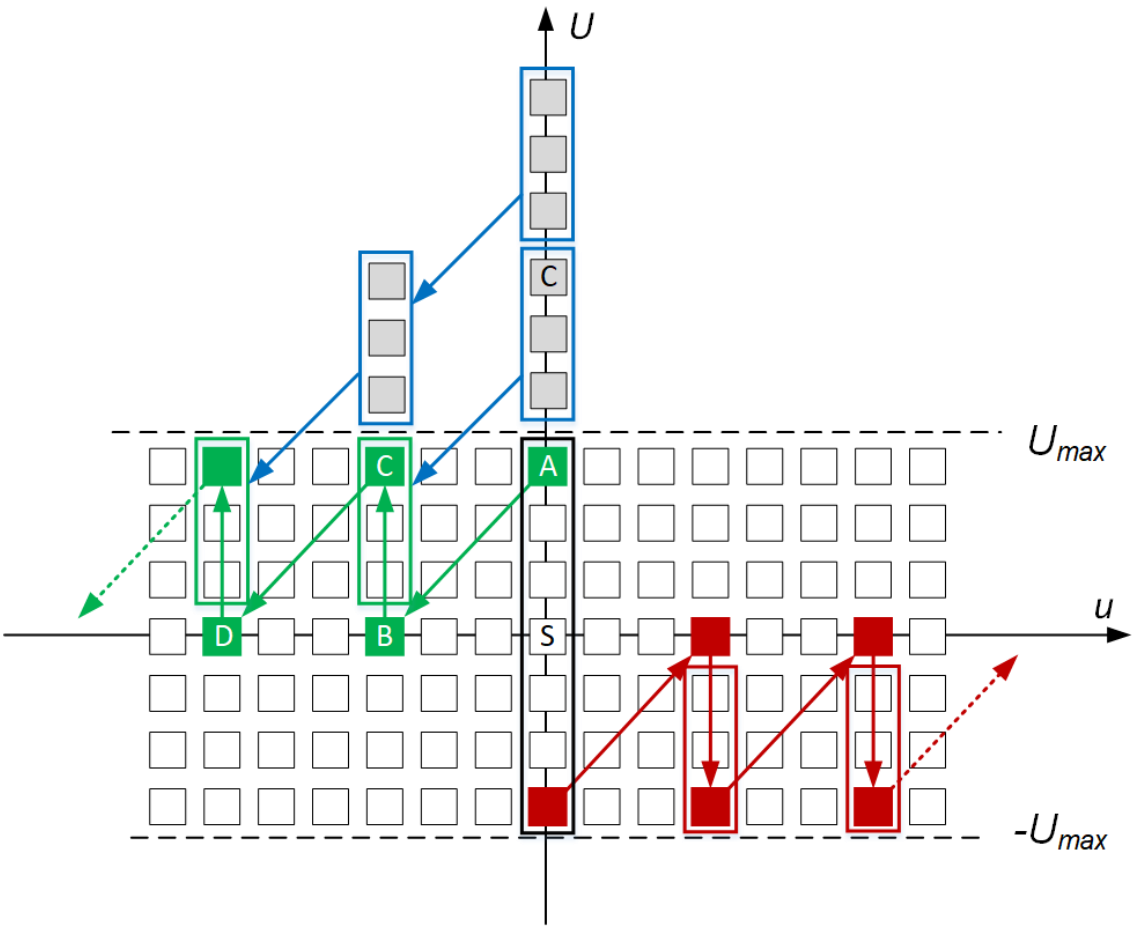

Figure 4.4. The "stepping out" process. Back box indicates the central line. Green

indication shows the left part of the "stepping out" and red one shows the right part. Blue indication shows how points beyond the cut-off frequency are recovered.

Now, the object's Fourier transform can be wholly reconstructed by "stepping out". Finally, apply the inverse Fourier transform to get the object function in real space, the entire process is shown in **Pseudocode 4.2**.

Pseudocode 4.2: Stepping out in WDD

Inputs: *D-set* (D), *the size of the D-set* (X, Y), *cut-off frequency* (f), *small constant* (epsilon)

Outputs: *object* (obj)

```
    // Calculate the central point index
1   M = X/2
    // Calculate the number of "stepping out"
2   n = round(X/(2f))
    // Calculate the central point in the D-set
3   c = sqrt(D(M,M))
    // Calculate the central vertical line
4   O = conj(D(:,M)/c)
5   For (k = 1 to n) do
      // Step out both side
6     O_left  = conj(D(:, M - k·f)/O(M - k·f))
7     O_right = conj(D(:, M + k·f)/O(M + k·f))
      // Update the Object Fourier transform
8     O(M - k·f : M – (k-1)·f) = O_left
9     O(M + (k-1)·f : M + k·f) = O_right
10  End loop
    // Calculate the object in real space
11  obj = ℱ⁻¹(O)
```

Note: **round**: *round to nearest integer*. $\mathcal{F}^{-1}$: inverse Fourier transform. **sqrt**: square root. **conj**: complex conjugate. **D(:,M)**: represents the **M**th column of the matrix **D**.

The key point of the "stepping out" is to utilize the horizontal information from the D-set within vertical calculation step by step, until it can finally recover all the frequency components. However, this process is sequential, so the calculation of the next step is highly dependent on the results of the previous step, which requires a highly reliable fraction of data at the beginning. Otherwise, the errors will accumulate during the "stepping out" process

wherever initial errors occur.

4.2.2. Projection Strategy

To overcome the accumulating error in the “stepping out”, “projection strategy” is introduced to give a more straightforward and reliable way to retrieve the horizontal information [33]. Here, using the same example shown in Equation (4.19) and Figure 4.3 to explain the “projection strategy”.

Firstly, like the “stepping out” method, reconstruct the central vertical line within the cut-off frequency, see Equation (4.22). Now, we get an estimate of the object’s Fourier transform, $O_0(u) = [0, B, C, D, 0]$, even if it only has the part within the cut-off frequency. Then, shift the central estimate $O_0(u)$ to form a new term, $|O(u-U)|^2$, see Figure 4.5.

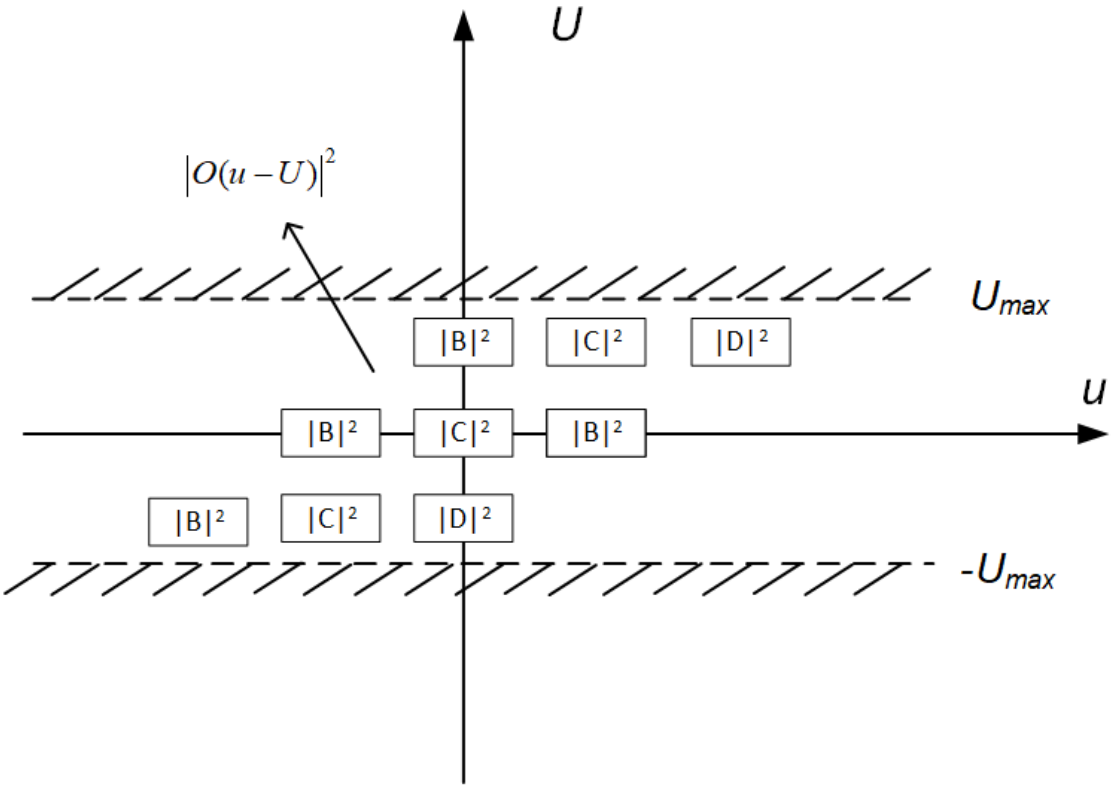

Figure 4.5. Illustration of $|O(u-U)|^2$.

The next step is to multiply the D-set from Equation (4.18) with $O(u-U)$, and gives Equation (4.25):

$$D(u,U)O(u-U) = O(u)O^*(u-U)O(u-U)$$
$$= |O(u-U)|^2 O(u) \qquad (4.25)$$

In this example, Equation (4.25) can be illustrated in Figure 4.6.

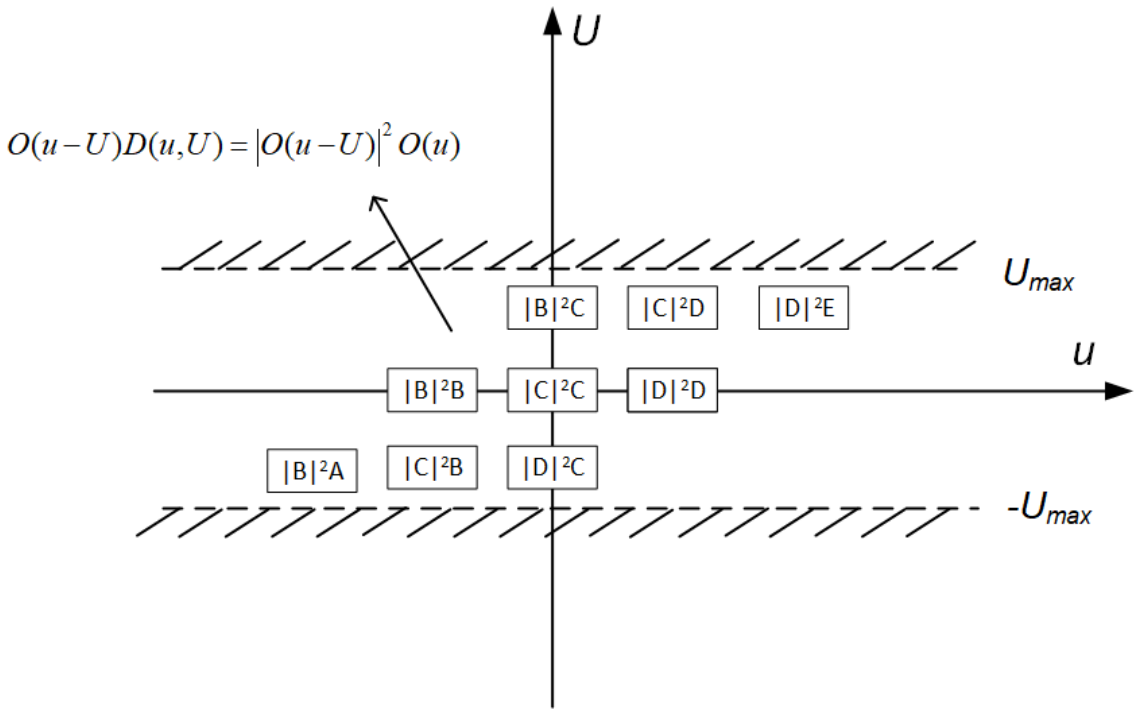

Figure 4.6. Illustration of the Equation (4.25), each block is $|O(u-U)|^2 O(u)$.

As shown in Figure 4.6, a common factor can be extracted from each column, which is exactly the value of $O(u)$. Therefore, we can sum up Figure 4.5 and Figure 4.6 along the U direction, giving Equation (4.26) and (4.27):

$$\sum_U |O(u-U)|^2 \qquad (4.26)$$

$$\sum_U |O(u-U)|^2 O(u)$$
$$= O(u) \sum_U |O(u-U)|^2 \qquad (4.27)$$

This process can be represented by Figure 4.7.

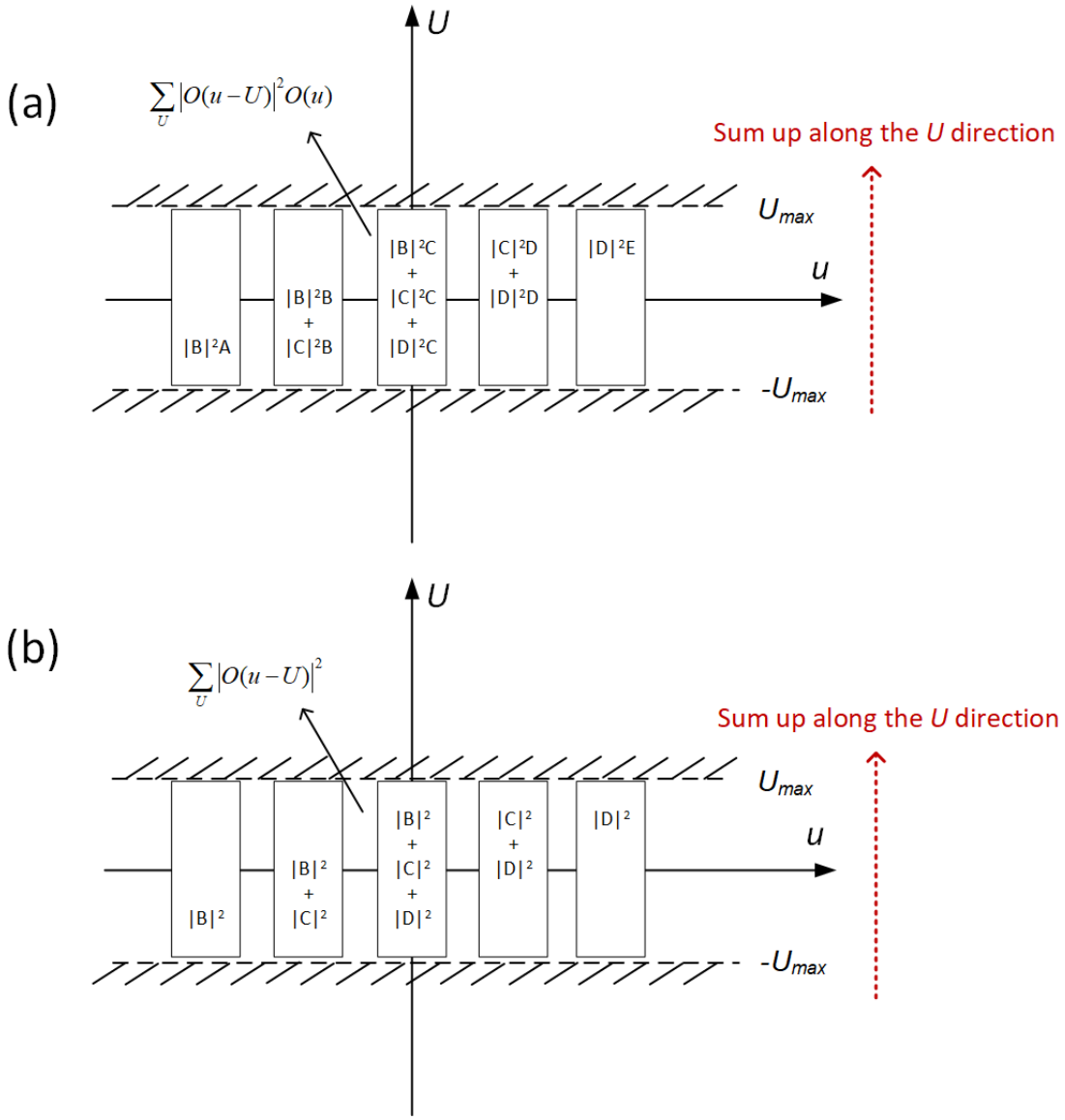

Figure 4.7. Summation procedure. (a) Sum up $|O(u-U)|^2 O(u)$ along the U direction. (b) Sum up $|O(u-U)|^2$ along the U direction.

Now, the reconstruction of the object's Fourier transform can be calculated by

the division between Equation (4.27) and (4.26), defined as Equation (4.28):

$$O(u) = \frac{O(u)\sum_U |O(u-U)|^2}{\sum_U |O(u-U)|^2} \tag{4.28}$$

In terms of the D-set, Equation (4.28) can be rewritten as Equation (4.29):

$$O(u) = \frac{\sum_U O(u-U)D(u,U)}{\sum_U |O(u-U)|^2}, \qquad |U| \le U_{max} \tag{4.29}$$

where U_{max} is the cut-off frequency in vertical.

Compared to the “stepping out” method, projection method makes full use of all the information within the cut-off limitation and improves robustness to noise. Although the forward and backward shift of the low frequency parts of $O(u)$ in the u direction can get up to twice the length of $O(u)$ itself, which means a single projection should double the support size for the next one. However, the cut-off frequency limits the D-set, therefore, one projection still can only extend $2U_{max}$ length, which is the same as the “stepping out”. The details are shown in Figure 4.8.

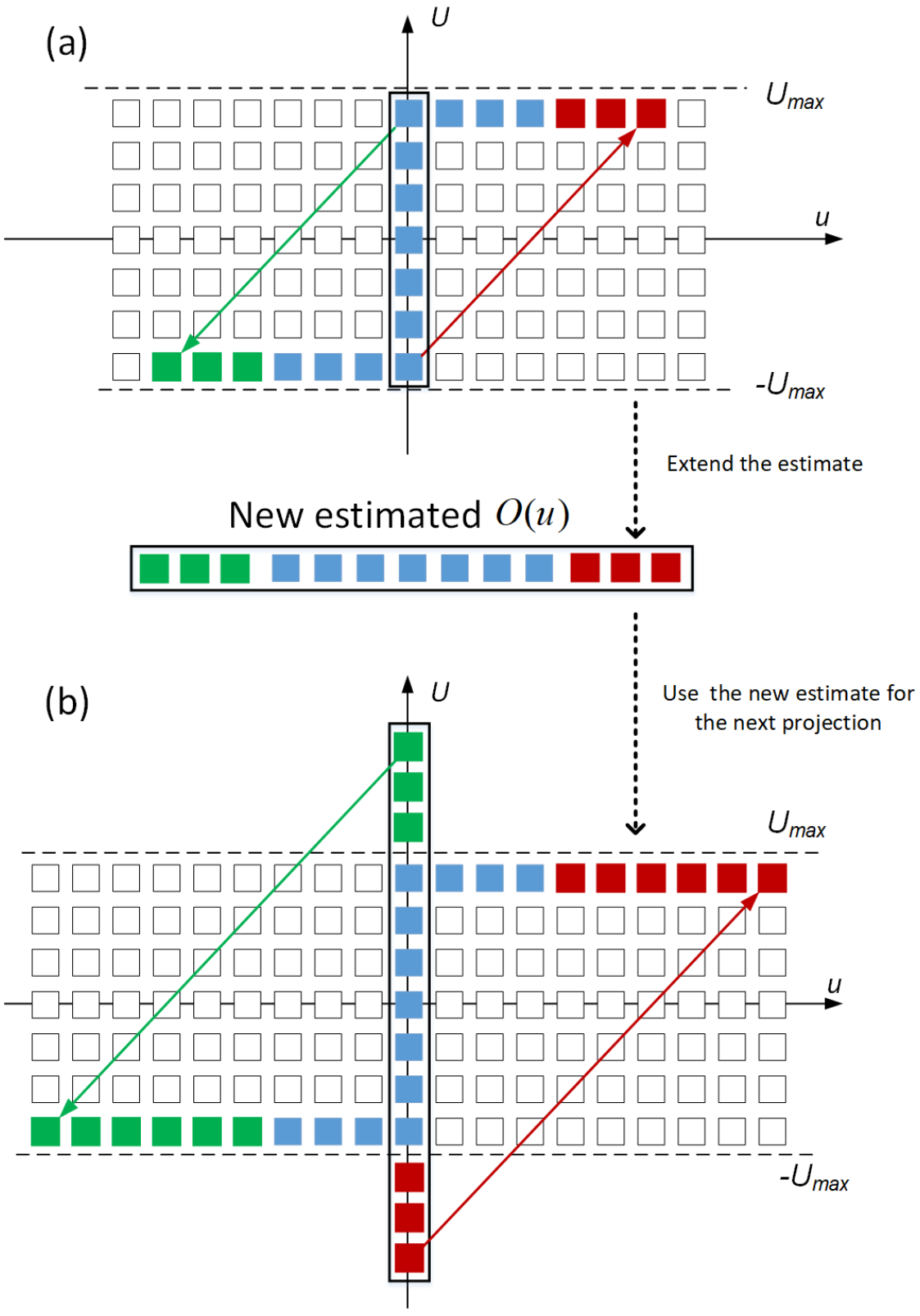

Figure 4.8. Schematic of the projection in WDD. (a) The first projection. (b) The second projection. The blue squares show the points retrieved within the cut-off limitation. The green and red squares indicate the new points we retrieved

beyond the cut-off frequency.

Figure 4.8 (a) is the first projection only using the central information within the cut-off frequency. The projection operation actually projects the top and bottom points at 45 degrees, to the cut-off frequency correspondingly, like the arrows shown in Figure 4.8 (a). The green and red squares in Figure 4.8 (a) indicate the new points we retrieved beyond the cut-off frequency. Hence, we can get a U_{max} extension for both sides, resulting in $2U_{max}$ in total. Then, use the new estimated $O(u)$ for the next projection, like Figure 4.8 (b). Repeat this until the whole object is recovered. A simple version of projection strategy is shown in **Pseudocode 4.3**.

Pseudocode 4.3: Projection Strategy in WDD

Inputs: *D-set* (D), *size of the D-set* (X, Y), *cut-off frequency* (f), *small constant* (epsilon)
Outputs: *object* (obj)

```
    // Calculate the central point index
1   M = X/2
    // Calculate the number of "stepping out"
2   n = round(X/(2f))
    // Calculate the central point in the D-set
3   c = sqrt(D(M,M))
    // Calculate the central vertical line
4   O = conj(D(:,M)/c)
5   For (k = 1 to n) do
      For (t = -f to f) do
        // Calculate the sums
        topSum += shift(O,-t)·D(M + t·k,:)
        botSum += abs(shift(O,-t))²
8     End loop
      // Update the Object Fourier transform
9     O = topSum/(botSum + eps)
10  End loop
    // Calculate the object in real space
11  obj = ℱ⁻¹(O)
```

Note: **round**: *round to nearest integer*. $\mathcal{F}^{-1}$: inverse Fourier transform. **sqrt**: square root. **conj**: complex conjugate. **D(:,M)**: represents the **M**th column of the matrix **D**. **shift**: *a function that shifts the input*. **abs**: amplitude. ***eps***: a small constant in MATLAB to avoid dividing 0.

4.2.3. Probe Solution

According to the methods described above, if the probe is known, it is straightforward to deconvolve it from the H-set to obtain the object. If the probe is unknown, then to solve the object as well as the probe at the same time amounts to blind deconvolution of the H-set [43]. This blind deconvolution problem was first solved in 1993 by B.C. McCallum and J.M. Rodenburg, who proved that it has a unique solution [43]. Here, we proposed a new approach that use the idea of "projection strategy" to solve this blind deconvolution.

According to Equation (4.15), $\chi_P(r,-U)$ can be separated from H-set by

Equation (4.30):

$$\chi_P(r,-U) = \frac{\chi_O(r,U)^* H(r,U)}{|\chi_O(r,U)|^2 + \varepsilon} \tag{4.30}$$

where ε is a small constant to avoid dividing zero.

Similarly, form the D-set for the probe by Equation (4.31):

$$\begin{aligned} D_P(u,-U) &= \boldsymbol{\mathcal{F}}_r\{\chi_P(r,-U)\} \\ &= P(u)P^*(u+U) \end{aligned} \tag{4.31}$$

This projection for the probe solution is exactly the same as we did in the last section but in an opposite direction along the U, see Equation (4.32):

$$P(u) = \frac{\sum_U P(u+U)D_P(u,-U)}{\sum_U |P(u+U)|^2}, \qquad |U| \le U_{max} \tag{4.32}$$

Therefore, with an appropriate estimate of the object, the probe function can be separated and solved from the H-set.

However, when the object and probe are both unknown, this becomes a blind deconvolution problem. Here, an iterative method is introduced to solve the blind deconvolution. The iterative process of blind deconvolution starts with an initial estimate of the probe. The first step is to form its Wigner distribution $\chi_P(r,-U)$. Then, separate $\chi_O(r,-U)$ from H-set by Equation (4.15), and calculate the object D-set from $\chi_O(r,-U)$ by Equation (4.18). The next step is to conduct the projection procedure to reconstruct a new estimated object from D-set. With this new estimate, we can now form a new Wigner distribution of the object, denoted as $\chi'_O(r,U)$. Similarly, a new $\chi'_P(r,-U)$ can now be calculated by Equation (4.30), using $\chi'_O(r,U)$ and the H-set. Then, calculate the D-set for the probe and reconstruct a new estimated probe by the projection. So far, a single iteration has been completed, and the new probe estimate can now be used for the next iteration. The flow chart of the iterative solution is

shown in Figure 4.9.

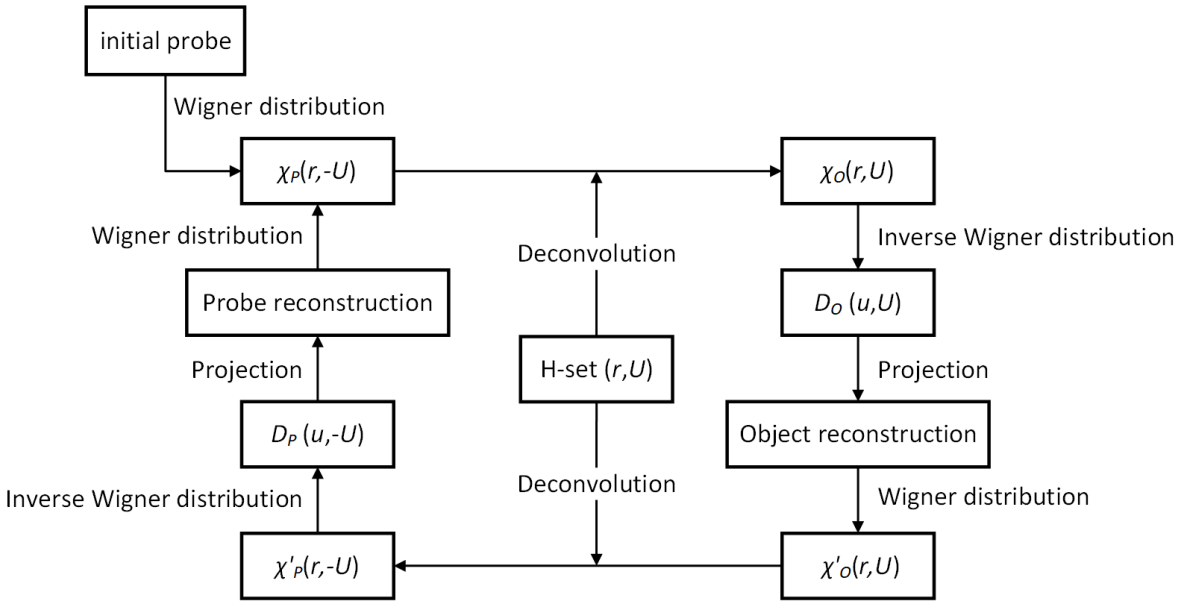

Figure 4.9. The flow chart of the iterative probe solution, H-set is invariable and employed every iteration to both $\chi_P(r,-U)$ and $\chi'_O(r,U)$.

Since the consistency of the H-set, this iterative method applies the constraint of measurement to each side of the deconvolution alternately. The pseudocode of this iterative deconvolution is shown in **Pseudocode 4.4**.

Pseudocode 4.4: WDD Probe Solution

Inputs: *I-set* (I), *initial probe* ($probe_0$), *probe size in reciprocal* (U), *small constant* (epsilon)

Outputs: *object function* (obj), *probe function* (probe)

```
    // Form the G-set from the I-set
1   G = ℱ_R(I)
    // Form the H-set from the G-set
2   H = ℱ_u(G)
    // Calculate the Fourier transform of the initial probe
3   P = ℱ(probe_0)
    For (j = 1 to J) do
      For (t = 1 to U) do
5       // Form the Wigner distribution of the probe
        χ_P = P · conj(shift(P, -t))
      End loop
      // Calculate the Wigner distribution of the object
8     χ_O = conj(χ_P) · H / (abs(χ_P)^2 + epsilon)
      // Calculate the D-set for the object
9     D_O = ℱ_r(χ_O)
10    // Reconstruct the object from the D-set via the projection
      O = projection(D_O)
      For (t = 1 to U) do
        // Form the Wigner distribution of the object
        χ_O = O · conj(shift(O, t))
      End loop
      // Calculate the Wigner distribution of the object
      χ_P = conj(χ_O) · H / (abs(χ_O)^2 + epsilon)
      // Calculate the D-set for the probe
      D_P = ℱ_r(χ_P)
      // Reconstruct the probe from the D-set via the projection
      P = projection(D_P)
    End loop
    // Calculate the object and probe in real space
    probe = ℱ^-1(P)
11  obj = ℱ^-1(O)
```

Note: $\mathcal{F}$: Fourier transform. $\mathcal{F}^{-1}$: inverse Fourier transform. **conj**: complex conjugate. **shift**: *a function that shifts the input*. **abs**: amplitude. ***eps***: a small constant in MATLAB to avoid dividing 0.

4.3. Simulation of Wigner Distribution Deconvolution

In this section, the new approach about the blind deconvolution in WDD will be

simulated and tested. The first simulation is using a 1D object and probe for a better illustration, and the second one will be a ptychographic simulation.

4.3.1. One-Dimensional WDD Simulation

In order to give a more intuitive demonstration of the WDD process, we start from a simple one-dimensional WDD, based on a simulation of a 1D object and a 1D probe in 2D space. In this case, each diffraction pattern is represented as a 1D line, the I-set, G-set, H-set and D-set become 2D and easy to display.

Generally, if we consider a simple 1D top-hat function shown in Figure 4.10.

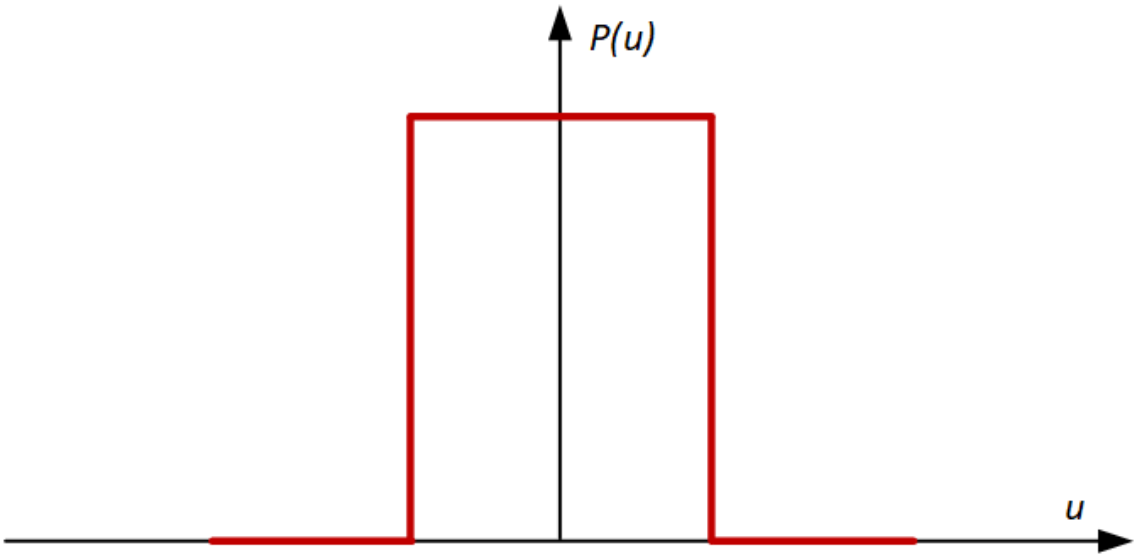

Figure 4.10. Top-hat aperture that only has value one and zero.

This top-hat function defines a 1D support region, all the points outside the support are zero. In ptychography, a 2D circular aperture is usually used to define the support area. Therefore, here, we use a normalized top-hat function as our initial aperture, which is the Fourier transform of the initial probe. To generate the simulation data, the actual probe is generated by putting some defects into the top-hat aperture. The ground truth of the object is transparent with modulus one everywhere but with random phase information, in this case,

the Fourier transform of the object is just real values without any phase in reciprocal space. The 1D simulation setup is shown in Figure 4.11.

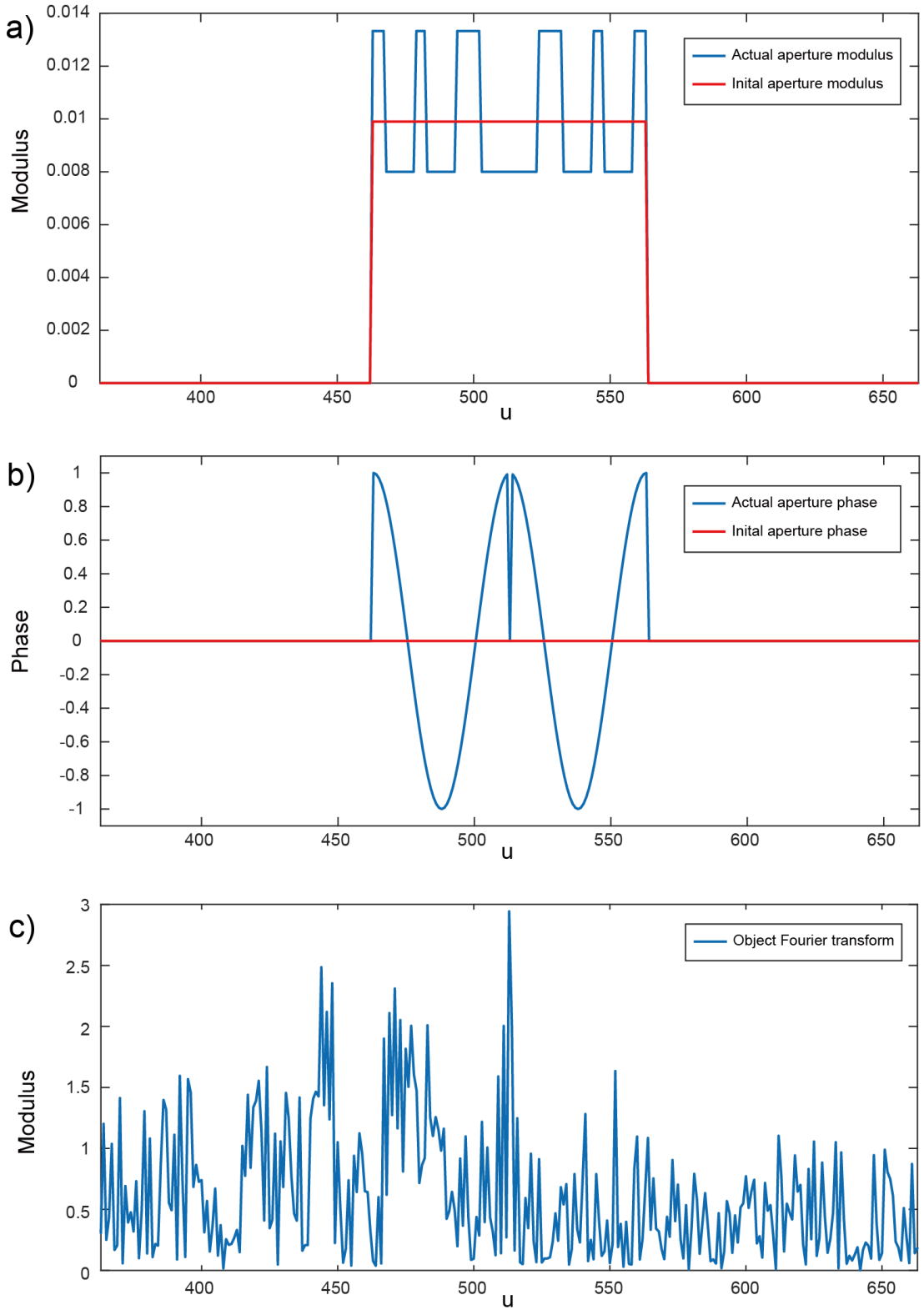

Figure 4.11. The 1D simulation setup. (a) and (b) are modulus and phase of the

correct aperture and the initial estimate, (c) is the modulus of the Fourier transform of a phase object.

Reconstructing this data using the projection method for blind deconvolution results in Figure 4.12. From top to bottom in Figure 4.12, it illustrates the reconstruction of the Fourier transforms of the object and probe at different iterations. After 10 iterations, the reconstructions shown in red lines are nearly perfectly matching the ground truth.

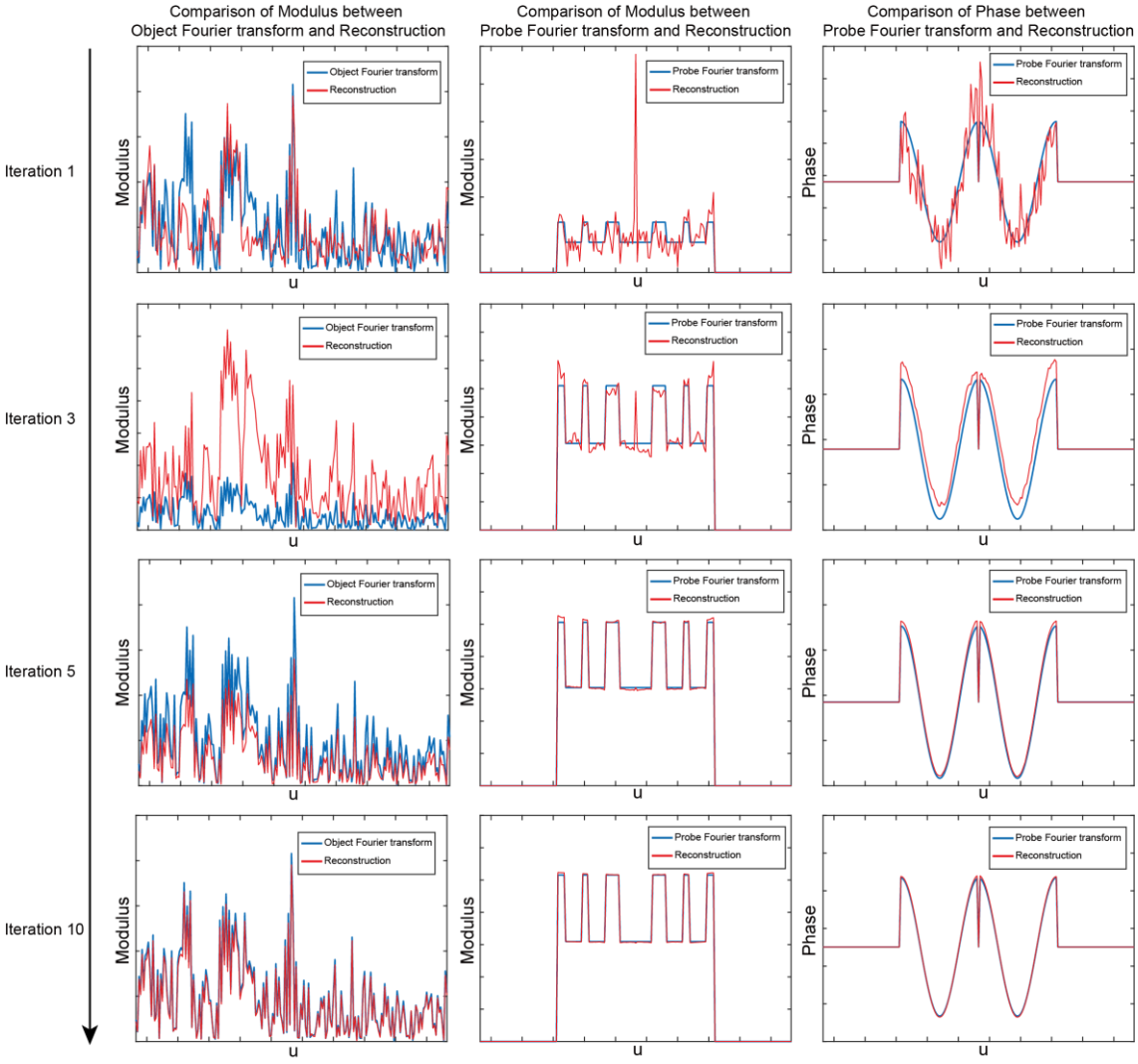

Figure 4.12. The reconstruction process of the 1D simulation. The blue lines are the ground truth, and the red lines are the reconstructions.

4.3.2. Two-Dimensional WDD Simulation

Having validated the project method using 1D data and shown that it is a feasible way to reconstruct the probe simultaneously with the object, we now proceed to a 2D simulation, which requires a 4D dataset deconvolution.

In the 2D simulation, 2 pictures are chosen to generate our object, which is a 2D complex image with 64×64 pixels, see Figure 4.13. The lake picture in Figure 4.13 (a) is the modulus of the object while the cat picture in Figure 4.13 (b) is used to form the phase part of the object.

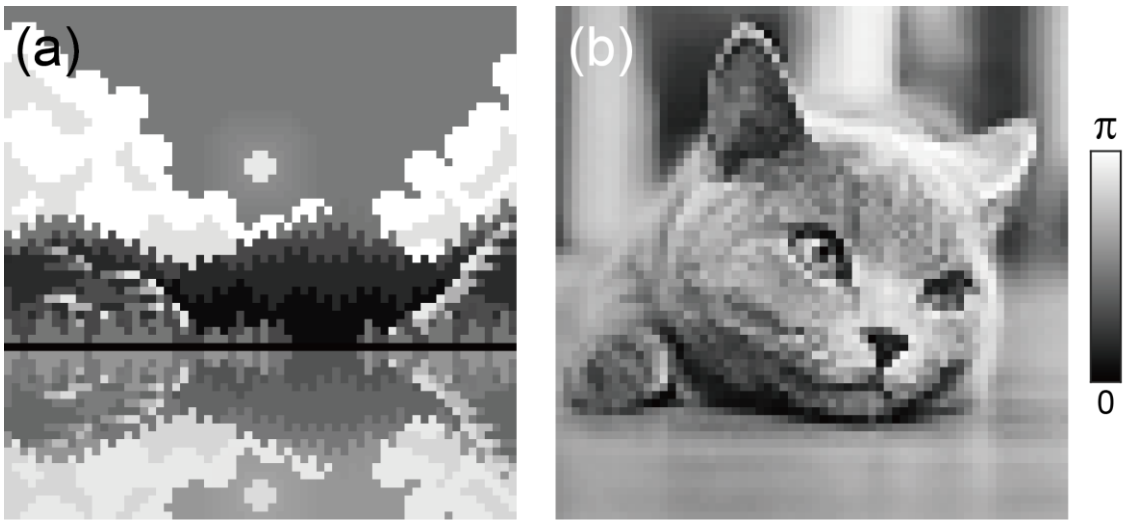

Figure 4.13. The 2D simulation object. (a) The modulus of the object. (b) The phase of the object.

The probe for the simulation is the Airy disc generated by a pinhole aperture with the same size (64×64 pixels). In order to test the probe solution of WDD, some defects were added to the aperture and 25% defocus error was introduced to the probe function, see Figure 4.14.

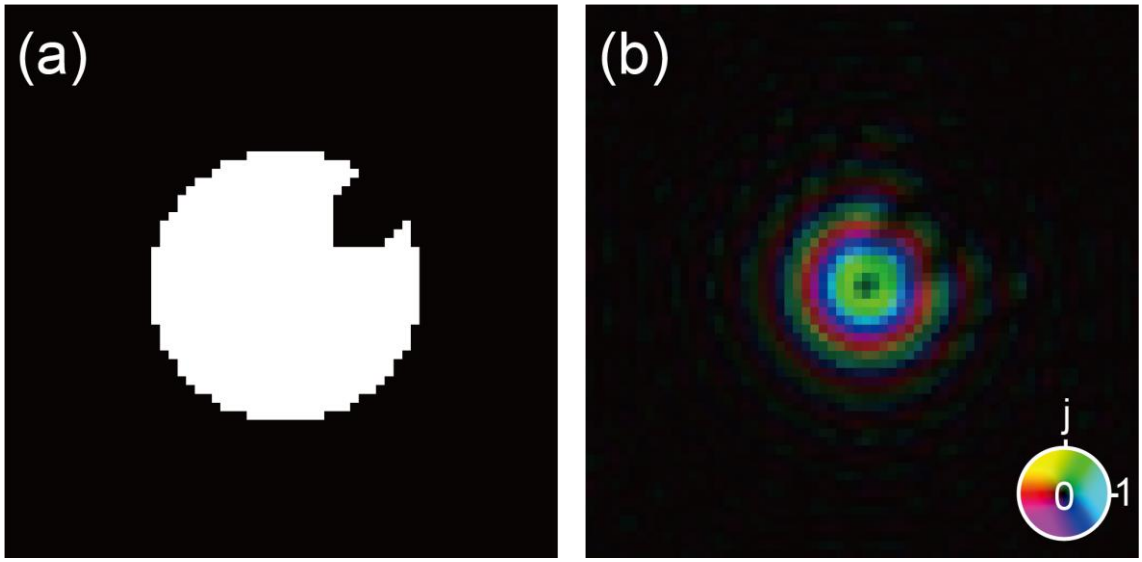

Figure 4.14. The simulation aperture and probe. (a) The aperture with some defects at the top right corner. (b) The probe generated by the aperture in (a), but with 25% defocus error.

To generate the diffraction data from this probe and object, a circular ptychography scan was conducted pixel by pixel. The object was expanded to a larger scale padded with zeros, then duplicated by itself to make a periodic object like Figure 4.15. The red box in Figure 4.15 indicates the calculation window and the arrow shows the probe moving direction.

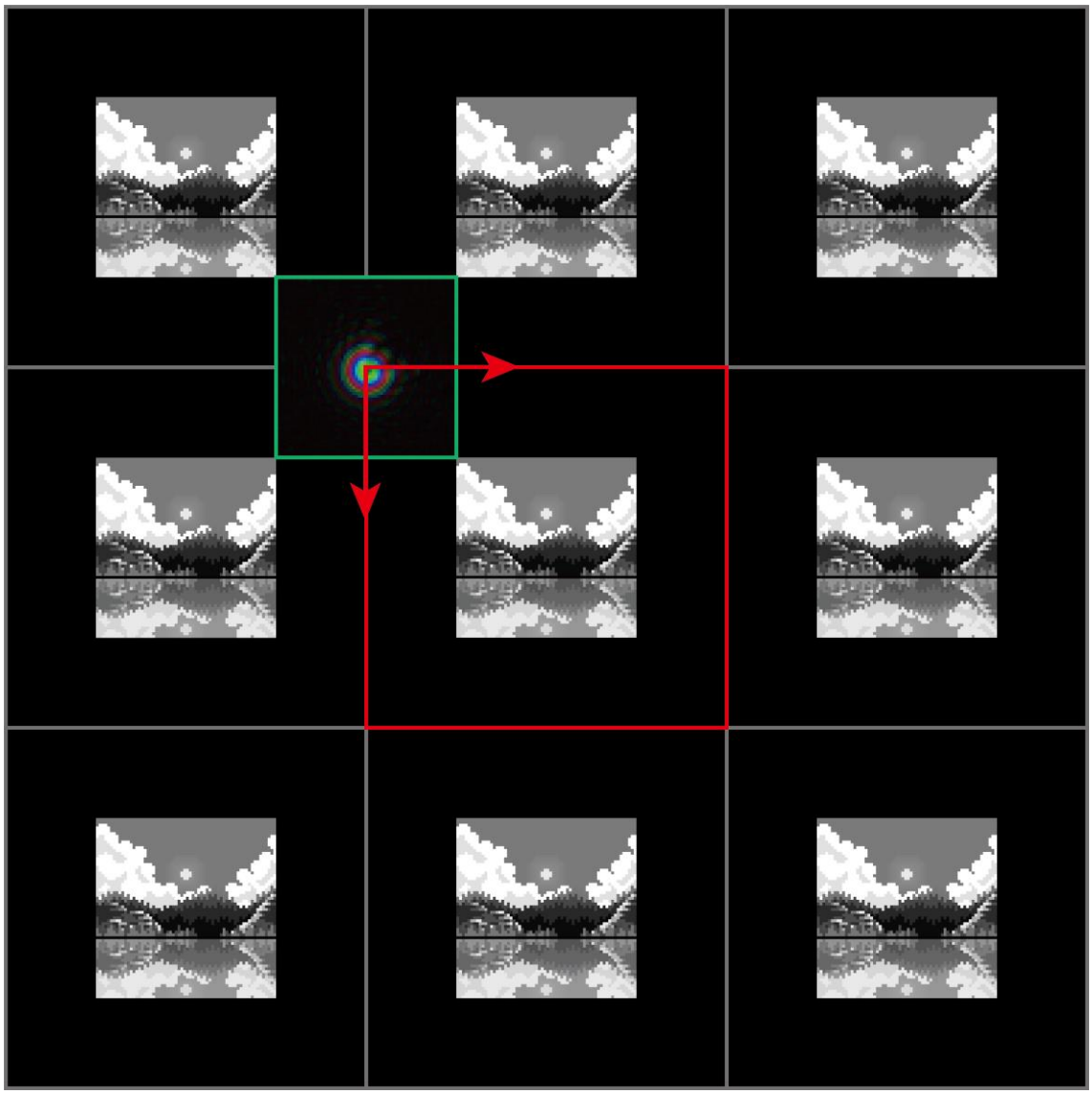

Figure 4.15. The circular scan over the periodic object. The red box is the calculation window of the reconstruction. The red arrows indicate the scanning directions.

To start the reconstruction, we give an initial estimate of the probe, which is an initial Airy disc probe modelled by a perfect pinhole aperture without any defocus error, modelling a perfect focussed beam. A comparison between the initial probe and the ground truth is displayed in Figure 2.1.

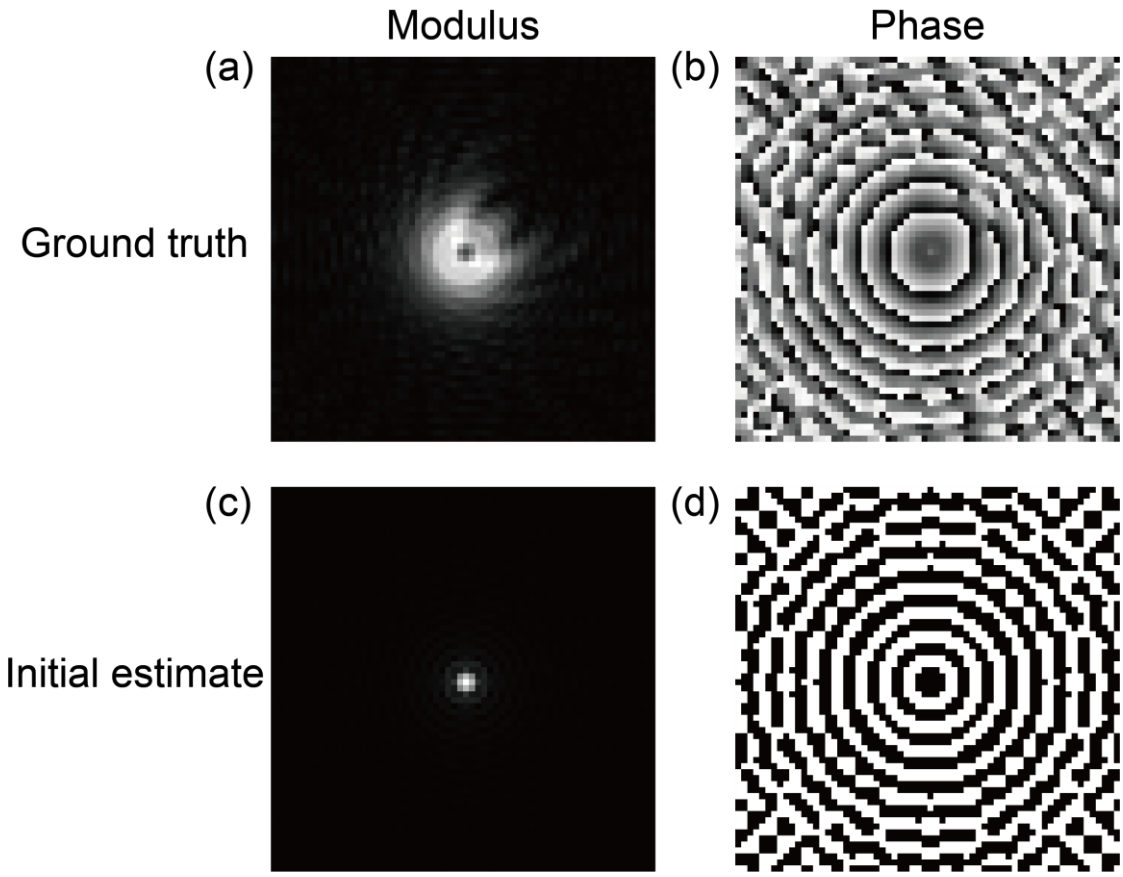

Figure 4.16. The comparison between the initial probe and the ground truth. (a) The modulus of the true probe. (b) The phase of the true probe. (c) The modulus of the initial probe. (d) The phase of the initial probe.

To monitor the performance of the deconvolution, an error between the H-set and the product of χ_P and χ_O is measured here, defined as Equation (4.33).

$$E_H = \int |\chi_O(r,U)\chi_P(r,-U) - H(r,U)|^2 dr dU \qquad (4.33)$$

where the energy of $\chi_O(r,U)\chi_P(r,-U)$ and $H(r,U)$ are normalized to unity before the error calculation.

The reconstruction process and the results are illustrated in Figure 4.17.

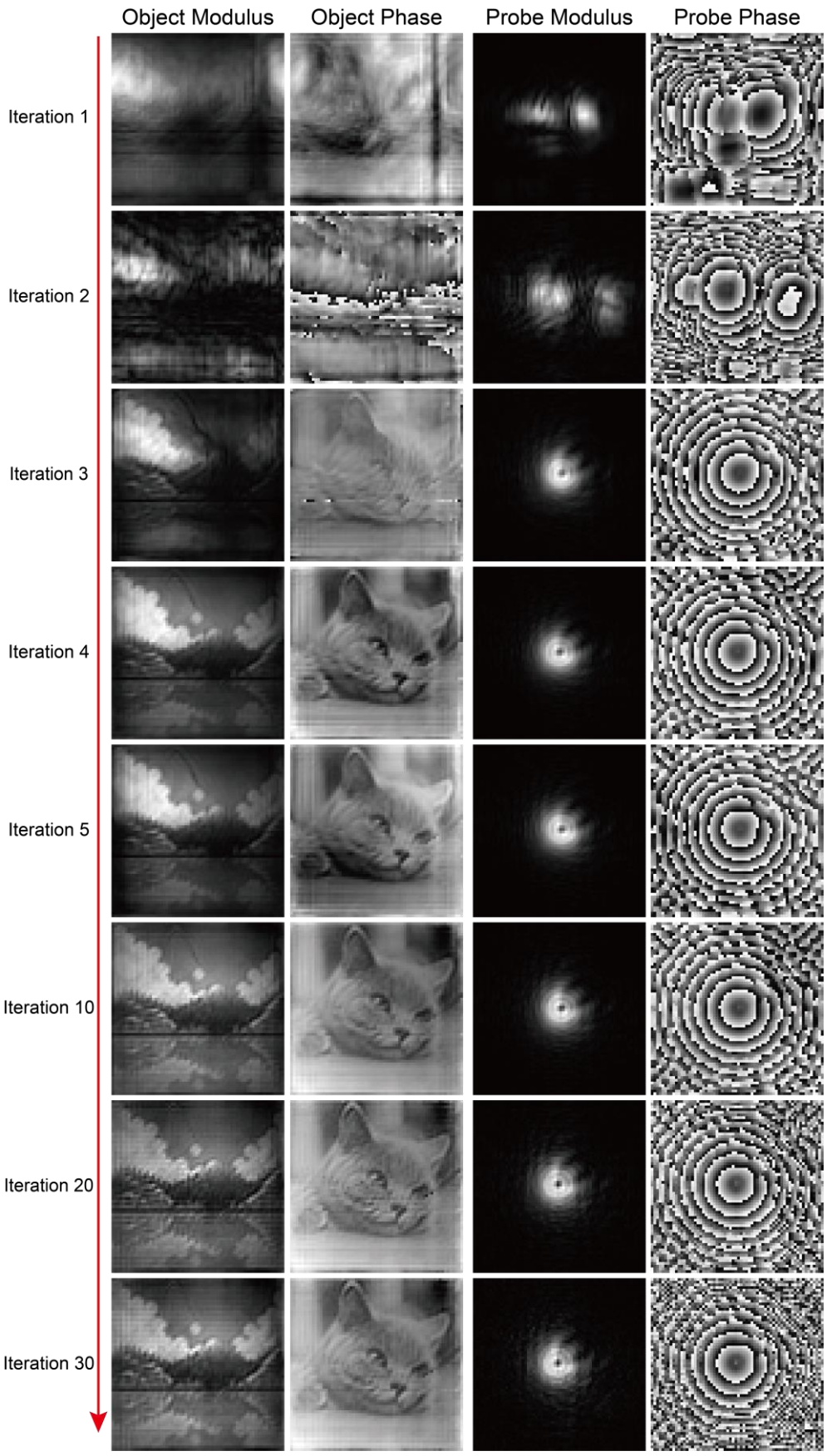

Figure 4.17. The WDD reconstruction process and the results.

The error E_H is displayed in Figure 4.18.

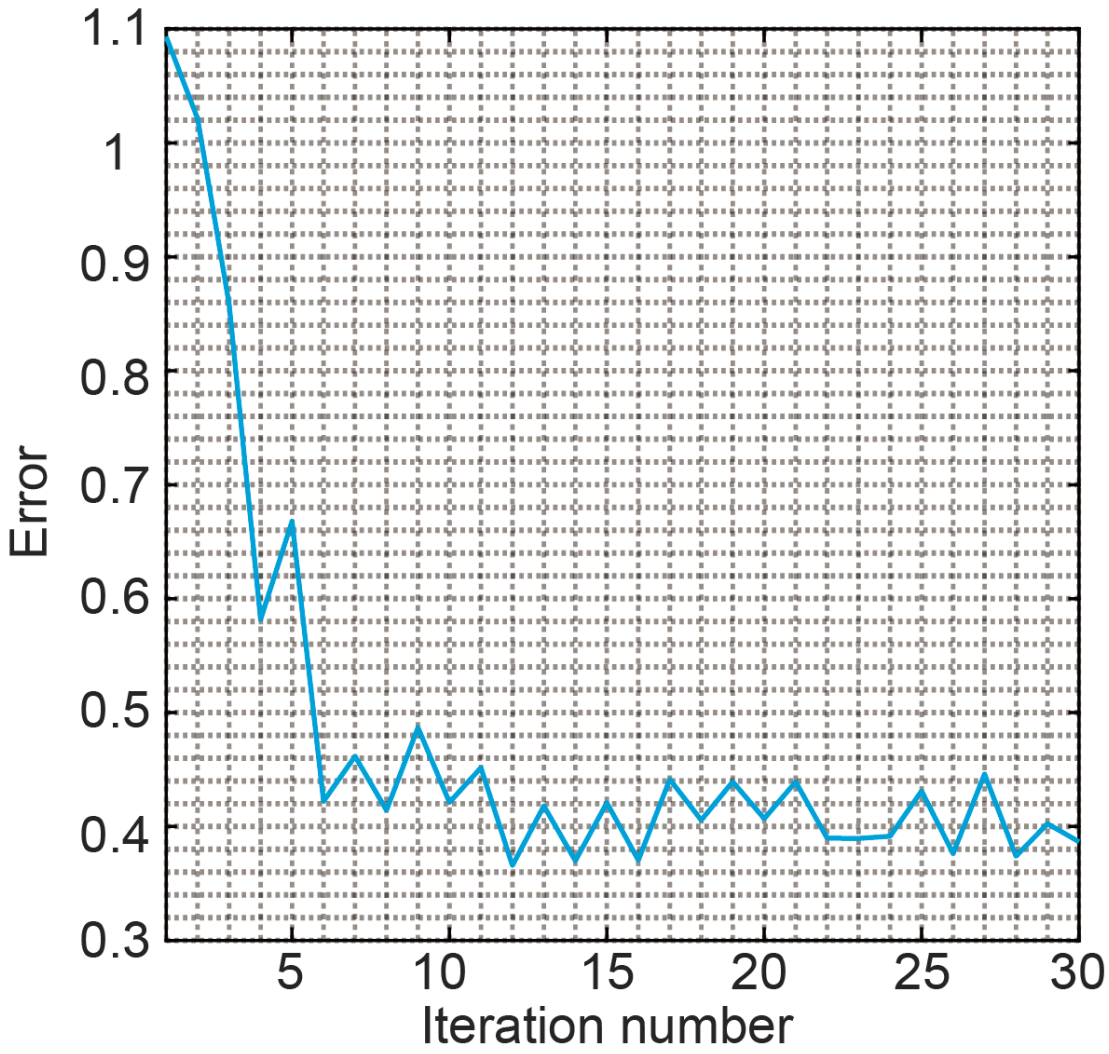

Figure 4.18. The normalized error between the H-set and the reconstructed $\chi_O(r,U)\chi_P(r,-U)$, indicates the performance of the deconvolution.

Moreover, the Fourier transform of the reconstructed probe is compared to the true aperture in Figure 4.19. The defect on the aperture is recovered in Figure 4.19 (b).

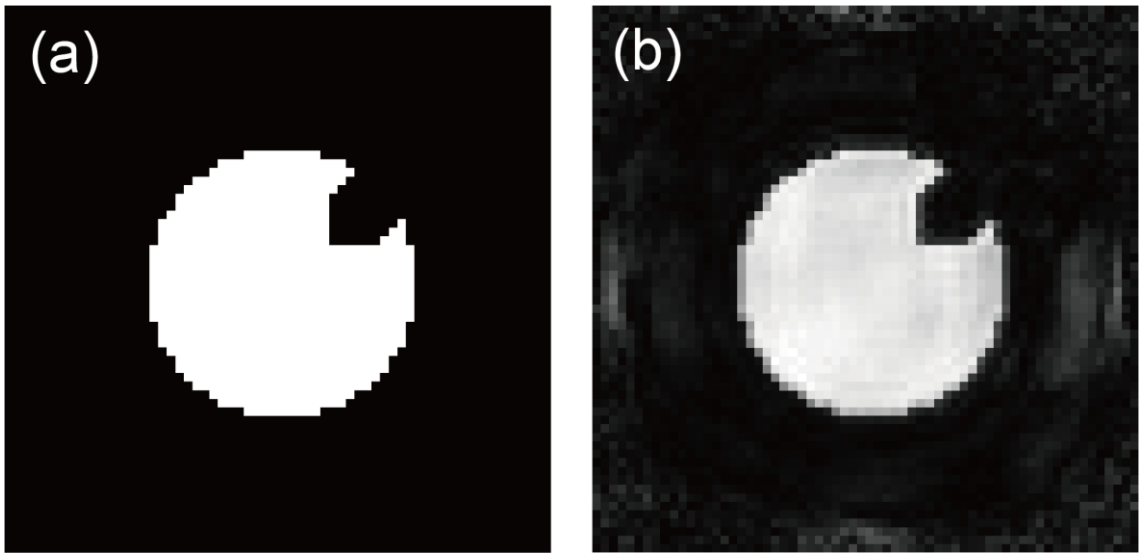

Figure 4.19. The comparison between the reconstructed aperture and the ground truth. (a) The true aperture used to generate the probe. (b) The reconstructed aperture.

The “projection strategy” also plays a crucial role in the reconstruction. A comparison of the reconstruction before and after the projection is shown in Figure 4.20. From Figure 4.20, It is obvious that the projection improves the fine details of the reconstruction, which is the high frequency part of its Fourier transform, especially for the phase reconstruction.

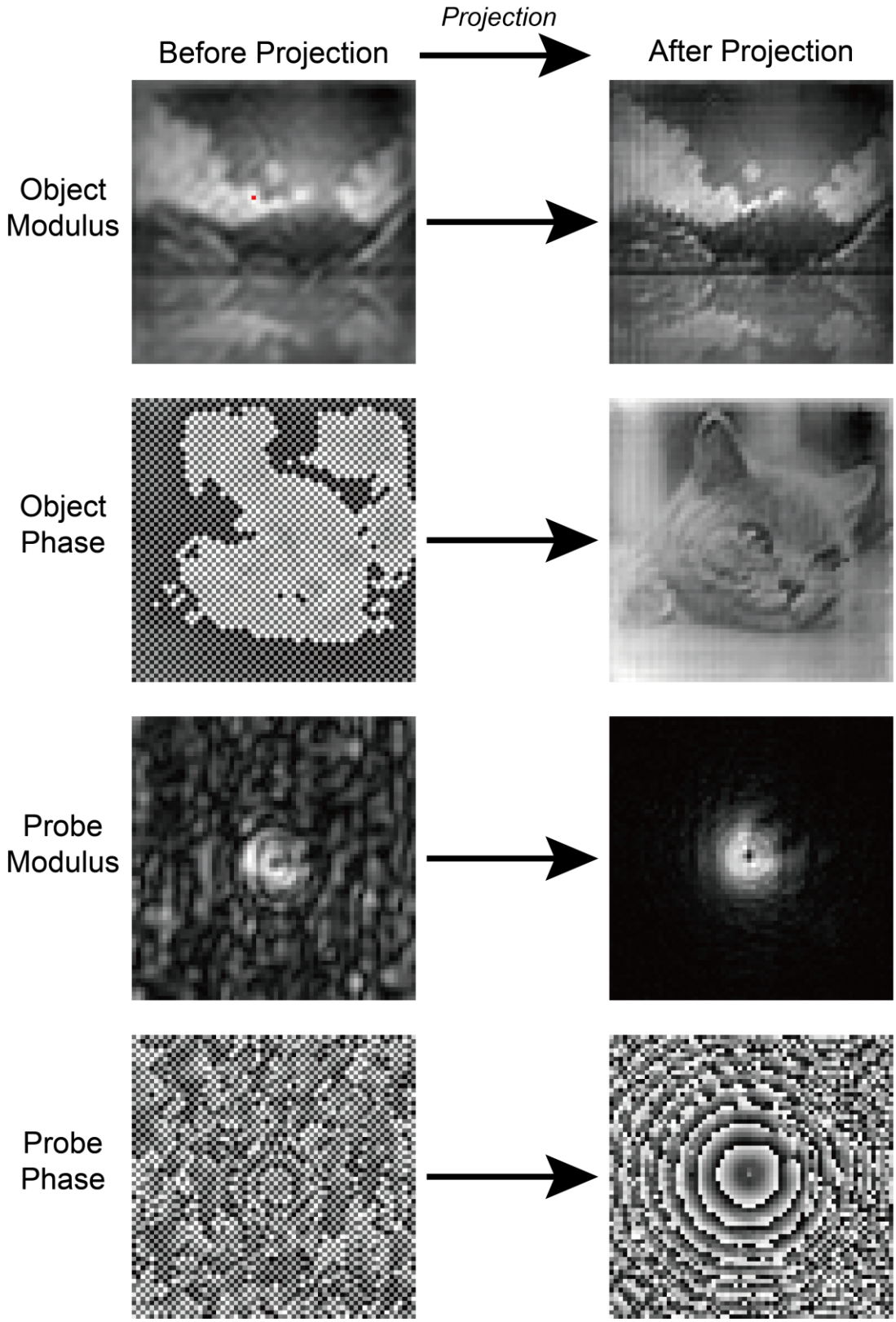

Figure 4.20. The comparison of the reconstruction before and after the projection. Left part is the reconstructions before the projection, and the right part are the reconstructions after the projection.

As the results illustrate, the blind deconvolution under a bad initial condition is solved by the new iterative method we proposed. It only takes around 5 iterations to the convergence, however, each iteration contains massive 4D calculations and the “projection strategy” for both object and probe. This results in heavy computation for each iteration.

4.3.3. Comparison with ePIE

In order to see the difference between WDD and normal iterative ptychography solutions, we tested the same data with ePIE, which was mentioned in Chapter 3.2.7. The result of the reconstructions is shown in Figure 4.21. Compared to Figure 4.17, ePIE generally performs better than WDD in our test. Only 10 iterations provide a considerable good reconstruction that both solved the object and probe.

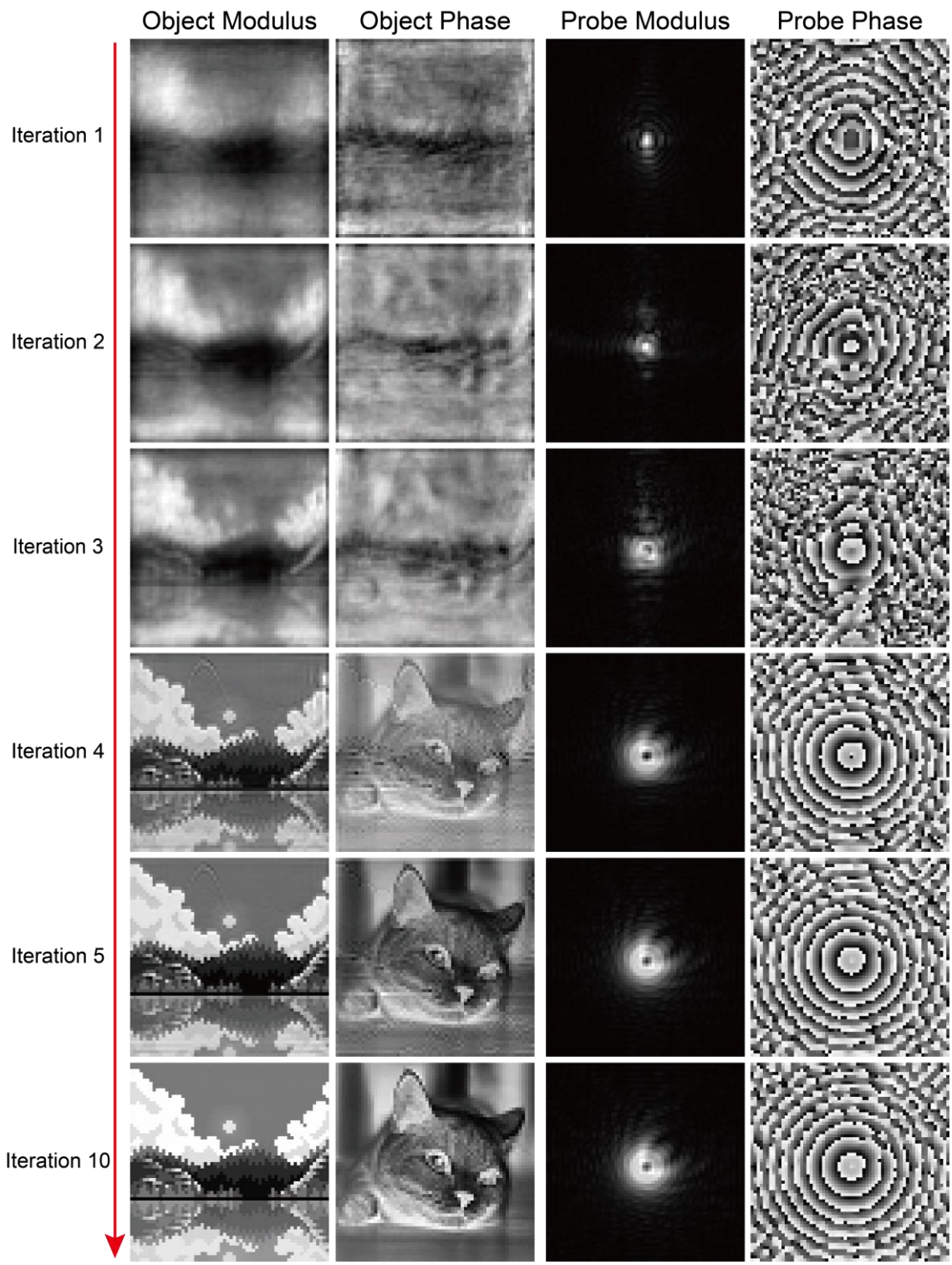

Figure 4.21. The ePIE reconstruction process and the results.

4.4. Conclusion

Although this 4D deconvolution was confirmed to have a unique solution [43],

it is difficult to perfect deconvolve and reconstruct them. The alternating deconvolution between the object and the probe could be stuck at some local minima if there is no additional constraint applied during the iterations. Also, the result of the separation of two Wigner distribution functions is highly dependent on the value of the constant that is used in Wiener filter, see Equation (4.16). The disadvantage of using a Wiener filter in the division is that it forces $\chi_O(r,U)$ to be small wherever $\chi_P(r,-U)$ is small, resulting the decrease of $p(r)$ and $o(r)$. Although the small constant ε in the Wiener filter could be varying during the iteration, a large ε will cause $\chi_O(r,U)$ and $\chi_P(r,-U)$ to be excessively smoothed while a small ε will make them exhibit excessive ringing. Therefore, further improvements could be made by seeking an optimum of ε in the Wiener filter as well as adding more constraints on the object or probe, such as setting the modulus to 1 for a phase specimen or forcing the aperture to be 1 inside the hole. Because of these drawbacks that we mentioned, we are seeking for a better solution for the ptychography. In the next chapter, we will discuss a completely different approach from WDD called “set projection algorithm”.

5. Set Projection Algorithms

In the last chapter, a direct ptychographic phase retrieval solution called WDD was introduced. In addition to this, there is an important class of algorithms in ptychography, which are iterative solutions. In Chapter 3.2, ePIE was introduced as an example of iterative phase retrieval solution in ptychography. Apart from ePIE, there is another kind of phase retrieval method called set projection algorithm. This chapter will describe this category algorithms in detail and explain how the concept of set projection is applied in ptychography, starting from its mathematics background. Existing set projection algorithms for phase retrieval in ptychography will be reviewed and a generalized form of set projection algorithm will be proposed later. Furthermore, we introduce the Bayesian optimization into this generalized form to auto-tuning its parameters and present a new method called Generalized Auto-Tuning (GAT) algorithm. A comparison of these algorithms will be presented at the end of this chapter.

5.1. Mathematics Background

5.1.1. Constraint Satisfaction Problems (CSPs)

In general, set projection algorithms are usually employed to solve Constraint Satisfaction Problems (CSPs). CSPs involve finding values for the problem's variables that simultaneously adhere to a set of inter-dependent constraints. The constraints represent the rules that define the relationships between variables and restrict the values that the variables can simultaneously take. Constraints can be unary, binary, or involve multiple variables. One widely known example of a CSP is the Sudoku puzzle, where the variables are the incomplete squares in the 9×9 Sudoku grid, and the problem is to assign values to these squares that simultaneously satisfy the inter-dependent row, column and block constraints – i.e. that the digits 1-9 appear only once in each

row, column and 3×3 block of the grid [12]. Other problems of this type include the cryptarithmetic puzzle, where the constraints are sets of linear or nonlinear equations [44], and graph colouring or graph connectivity [44, 45], where the constraints are that neighbouring vertices of the graph have different colours. Apart from these, the constraint also can be custom-defined, specific to the problem domain. The phase problem in ptychography is a kind of constraint satisfaction problem where the constraints are the measurements from the detector, the support information from the aperture and all other prior knowledge. The ptychography iterative process shown in Figure 2.8, is actually applying the constraint of the measurements and the constraint of the support alternately, to force the reconstruction approaching the correct solution.

5.1.2. Projection

To show graphically what a constraint set is and how to satisfy it, here, we define a simple geometric constraint S, a circle in the 2D plane, as an example, see Figure 5.1. This is sensible in ptychography because for each pixel point in a diffraction pattern, a circle is a good representation of the unknown phase and the known modulus. Given an arbitrary point x in this 2D plane, satisfying constraint S involves determining the point on the circle that is closest to x. This closest point can be readily ascertained by projecting the given point onto the constraint circle. This operation is defined as a standard projection, written as $P_S^1 x$, where P signifies the projection operator, S represents the constraint set, x is the given point and 1 indicates the standard relaxation degree. Taking constraint set S from Figure 5.1. for example, with centre point $[x_c, y_c]$ and radius r_S, projection of a point $[x_m, y_m]$ onto S is straightforward. The projection is the point a distance r_S from $[x_c, y_c]$ along the line joining $[x_c, y_c]$ to $[x_m, y_m]$, therefore, the standard projection point is:

$$P_S^1[x_m, y_m] = \left[\frac{r_s(x_m - x_c)}{\sqrt{(x_m - x_c)^2 + (y_m - y_c)^2}}, \frac{r_s(y_m - y_c)}{\sqrt{(x_m - x_c)^2 + (y_m - y_c)^2}}\right] \quad (5.1)$$

where the denominator is the distance between two points.

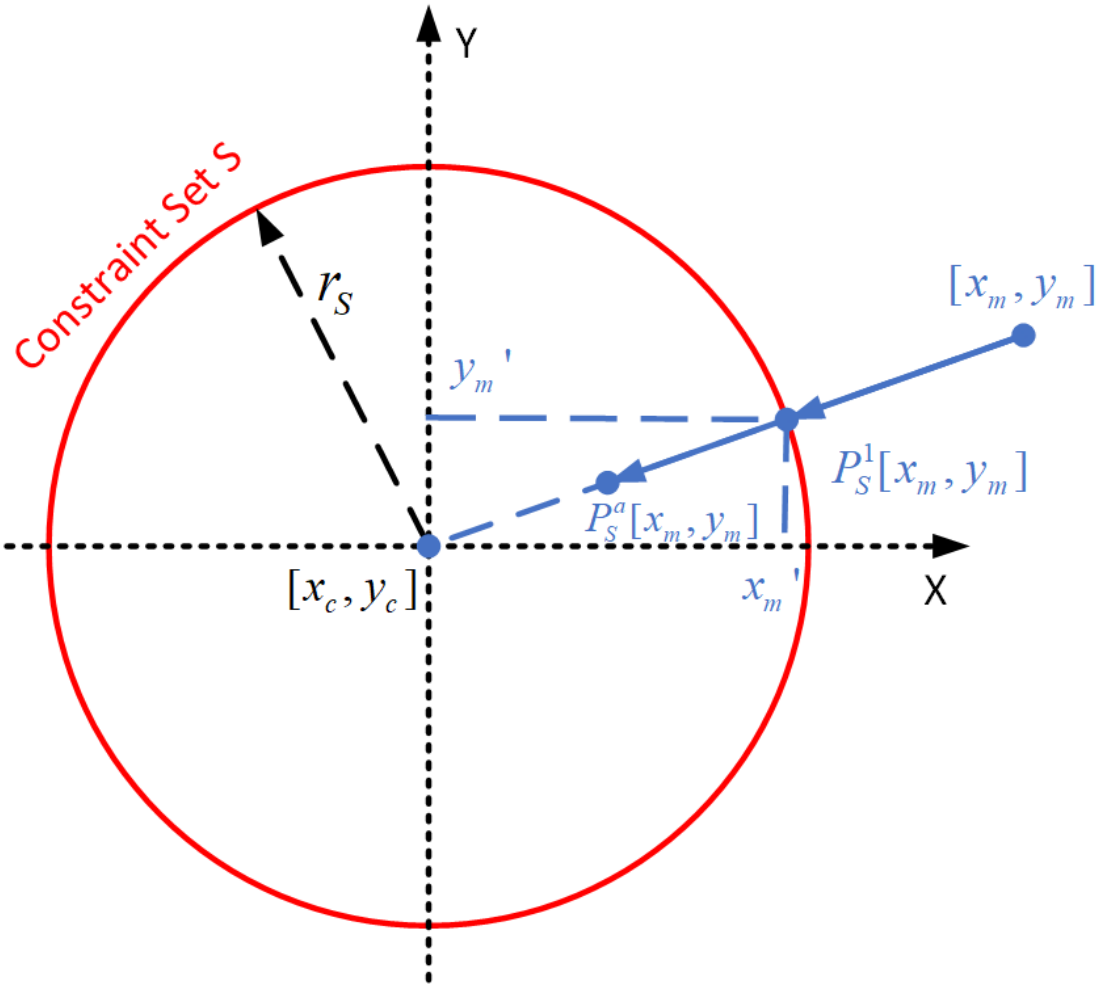

Figure 5.1. Schematic of projection calculation with coordinates.

Additionally, we can introduce the concept of relaxation to the projection operation, denoted by the relaxed projection operation of a point x onto a set S: $P_S^a x$, where S indicates the set and a indicates the degree of relaxation. When $a = 1$, $P_S^1 x$ represents the standard projection of x onto S. In terms of the standard projection, the relaxed projection can be written as Equation (5.2):

$$P_S^a x = (aP_S^1 + (1 - a)I)x \tag{5.2}$$

where I is the identity operator. When $0 < a < 1$, it moves x only a fraction of the distance toward P_S^1, which is called an under-relaxed projection, another

way is over-relaxed projection, where $a > 1$, which moves x beyond P_S^1. Of particular interest is the reflection about a constraint set, where $a = 2$, defined as Equation (5.3):

$$R_S x = P_S^2 x = (2P_S^1 - I)x \tag{5.3}$$

which moves x twice as far as P_S^1 in the same direction. Consequently, the relaxed projection depicted in Figure 1 can be mathematically represented as follows:

$$P_S^a[x_m, y_m] = \left[a\frac{r_s(x_m - x_c)}{\sqrt{(x_m - x_c)^2 + (y_m - y_c)^2}} + (1-a)x_m, a\frac{r_s(y_m - y_c)}{\sqrt{(x_m - x_c)^2 + (y_m - y_c)^2}} + (1-a)y_m\right] \tag{5.4}$$

A CSP involves finding the intersection of multiple constraint sets; a geometric example of three non-convex constraint sets S, T, U is illustrated in Figure 5.2. The arrows from x in Figure 5.2 depict various projections with different degrees of relaxation.

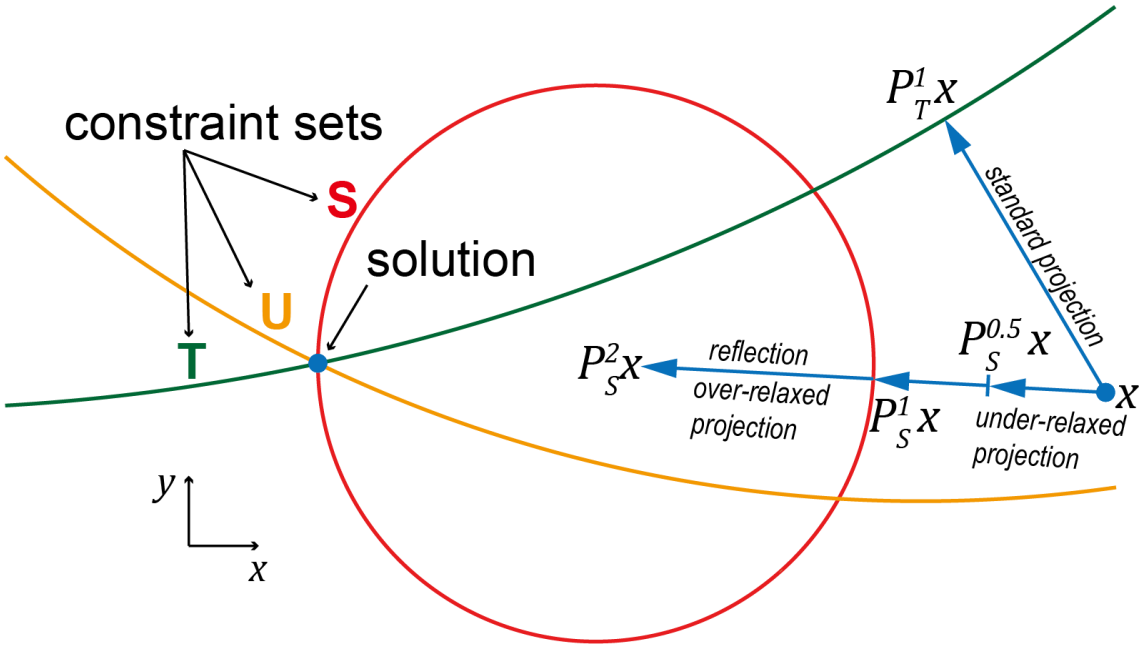

Figure 5.2. Projections with different relaxation degrees onto different sets

5.2. Solutions to the Constraint Satisfaction Problem (CSP)

5.2.1. Sequential Projections (SP)

Continuing with the picture shown above, the problem in Figure 5.2 find the intersection of three circular constraint sets S, T and U, whose centres and radii are known. The Sequential Projections (SP) algorithm is perhaps the most intuitive, and certainly the oldest, way to combine these projection operations into an algorithm, see Figure 5.3.

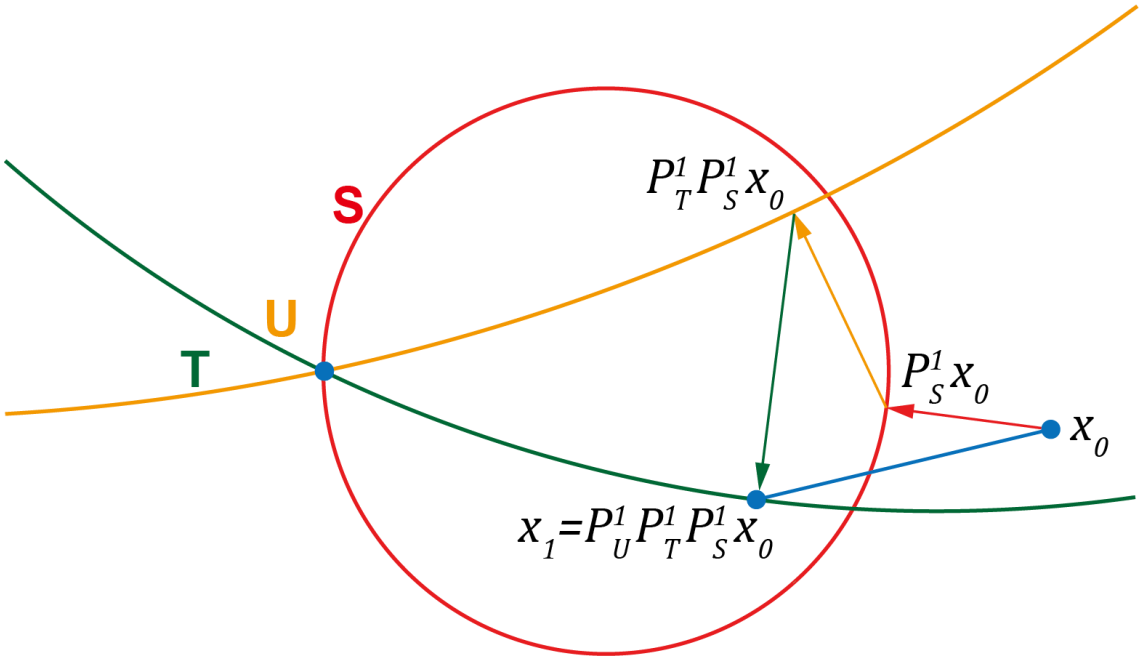

Figure 5.3. The geometric example of sequential projections (SP) algorithm, from $x_0 \rightarrow P_S^1 x_0 \rightarrow P_T^1 P_S^1 x_0 \rightarrow P_U^1 P_T^1 P_S^1 x_0$

Given an initial seed point x_0, the algorithm projects onto one of the constraints, takes the result of this projection and projects it onto a second constraint, then repeats, projecting onto each constraint sequentially until they have all been covered. The result of this sequence of projections, the point x_1 in Figure 5.3, seeds the next iteration of the algorithm and so on, either with the same fixed order of projections or with the order shuffled at random between iterations. For the geometric problem of Figure 5.3 with a fixed projection order of S, T then

U, the k^{th} iteration of SP calculates the Equation (5.5):

$$x_{k+1} = P_U^1 P_T^1 P_S^1 x_k \tag{5.5}$$

Furthermore, the Sequential Projections (SP) scheme can be over or under relaxed in a more general scheme, see Equation (5.6):

$$x_{k+1} = P_U^{a_U} P_T^{a_T} P_S^{a_S} x_k \tag{5.6}$$

Where the relaxation parameters a_S, a_T and a_U are chosen or tuned for best performance.

5.2.2. Product Space

Due to the sequential nature of the SP algorithm, there exists a risk of being entrenched in periodic oscillations rather than reaching a settled point of convergence. This is especially true where the underlying data on which the constraints are built are noisy, so that the sets may not all intersect at a single solution point. The ePIE algorithm in the previous Chapter 3, operates very similarly to a relaxed sequential projection algorithm and this oscillating behaviour is very common for ePIE. Projection algorithms that avoid this problem treat the constraint sets collectively rather than sequentially, by framing the optimisation within a product space. Product space refers to the construction of a new space by combining multiple copies or replicas of an existing space [46]. For instance, if there are N individual constraints for a CSP, they can be expressed as N sets within a Euclidean space K, like the circle problem mentioned above. The product space in this example can be denoted as K^N, which contains N copies of the space K within each of which embedded one primary constraint, see Figure 5.4.

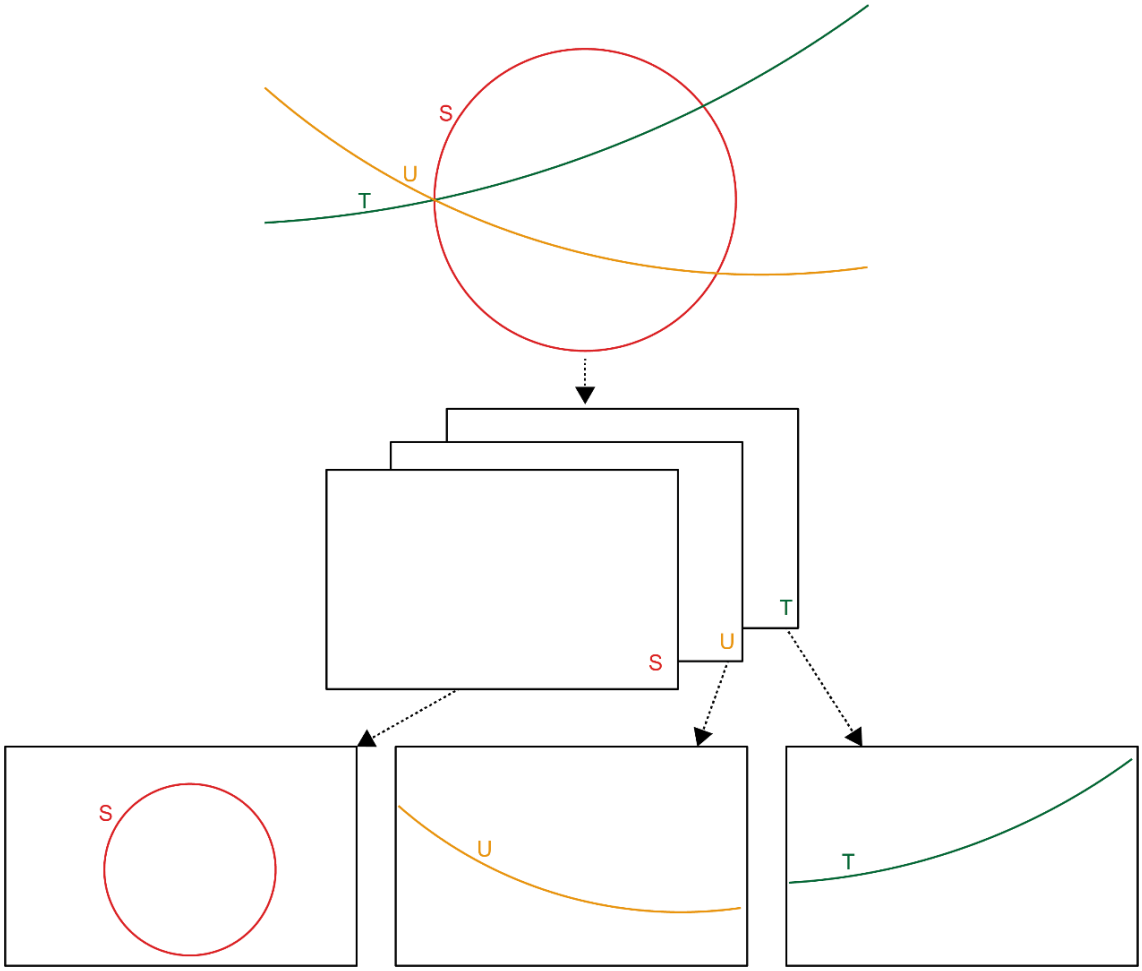

Figure 5.4. Illustration of the product space in three-circle problem.

The construction of the product space simplifies the CSP, it divides a CSP into many subproblems and allows the “parallel projection” operation, discussed next, onto each constraint set at the same time. This is an important concept that will be used in most set projection algorithms, the implementation of product space in ptychography will be demonstrated in later section.

5.2.3. Divide and Concur (DC)

Based on the idea of product space, Gravel and Elser proposed an approach called “divide and concur (DC)” to solve CSPs [46]. The idea is that the projections of a candidate solution, x_k, onto each of the constraints are calculated in parallel (the ‘divide’ step), then these projections are averaged in

some way to reach a single consensus solution (the ‘concur’ step). For a better demonstration, we continue to use the previous three-circles example in Figure 5.3 to help explain the concept more concretely. The product space in this instance requires three copies of the 2D plane, each holding one of the circle constraints. Each 2D plane also has its own estimate of the solution, so that x_k has three components: $x_k = [x_k^S, x_k^T, x_k^U]$, certainly, these three points could have the same value initially, but each of them is still independent with each other.

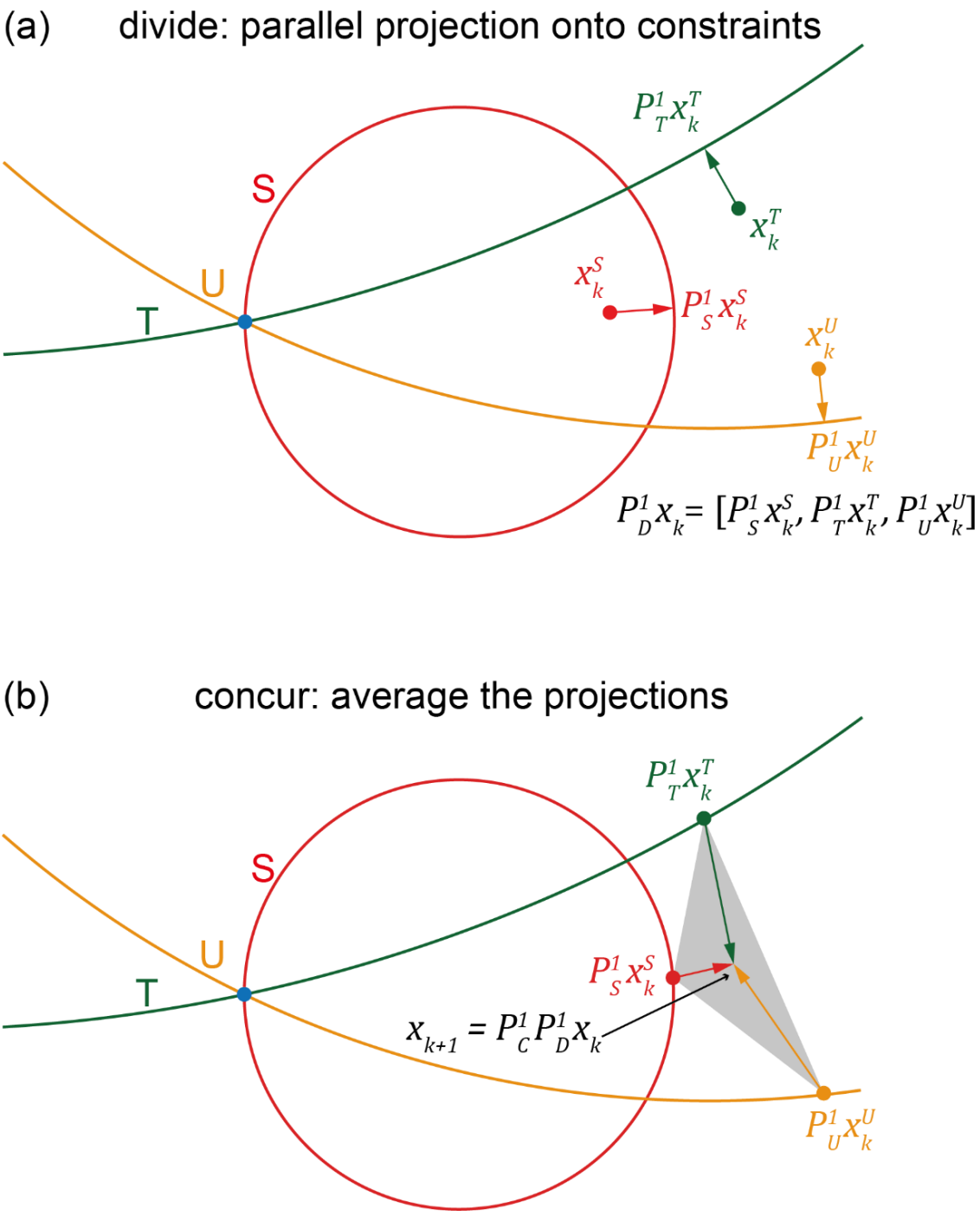

Figure 5.5. The schematic of DC approach, (a) the divide step (each constraint circle is actually lying in an individual copy of the space as each projection is independent, here we draw them together for simplicity), (b) the concur step, the new points from divide step will be averaged to one single solution.

The divide and concur approach begins with the divide step, see Figure 5.5(a), where the projections of x_k^S, x_k^T and x_k^U onto their respective constraints S, T and U are computed, giving three new points $P_S^1 x_k^S$, $P_T^1 x_k^T$ and $P_U^1 x_k^U$. This set

of parallel projections can be considered as a single 'divide' projection P_D^1 in the product space:

$$P_D^1 x_k = [P_S^1 x_k^S, P_T^1 x_k^T, P_U^1 x_k^U] \quad (5.7)$$

The concur step in Figure 5.5(b) averages the individual projections from the divide step so that they agree with each other, enforcing the knowledge that at a solution to the problem, all of the projections must arrive at a single solution point. This averaging operation can also be expressed as a projection in the product space, onto a 'concur constraint' represented by the set C in the product space:

$$P_C^1 x_k = \frac{1}{3}(x_k^S + x_k^T + x_k^U) \quad (5.8)$$

The divide and concur (DC) algorithm, alternates between the divide and concur projections, defined as Equation (5.9):

$$x_{k+1} = P_C^1 P_D^1 x_k \quad (5.9)$$

DC is considerably slower than SP - a much larger number of iterations are required for it to reach a solution. However, it has the significant advantage that the projections are independent and can be calculated in parallel, so that when there are thousands of constraints (as is the case for ptychography) computation time per iteration is greatly reduced. Note that the implementation of DC in ptychography is called Error Reduction (ER), which is an early solution proposed by Gerchberg and Saxton in 1972 [47].

5.2.4. Averaged Reflections (AR)

Like SP, DC can also generalise through over- or under-relaxation of the projections. One such relaxation produces the Averaged Reflections (AR) algorithm, which replaces the projections in DC with reflections, as shown in

Figure 5.6. (In this example the three components of x have been initialised to the same value, x_0.) In terms of the divide and concur projections in product space, AR is expressed as Equation (5.10):

$$x_{k+1} = P_C^1 P_D^2 x_k \tag{5.10}$$

As shown in the Figure 5.6, AR take reflections in the divide step, then average all the reflections in the concur step.

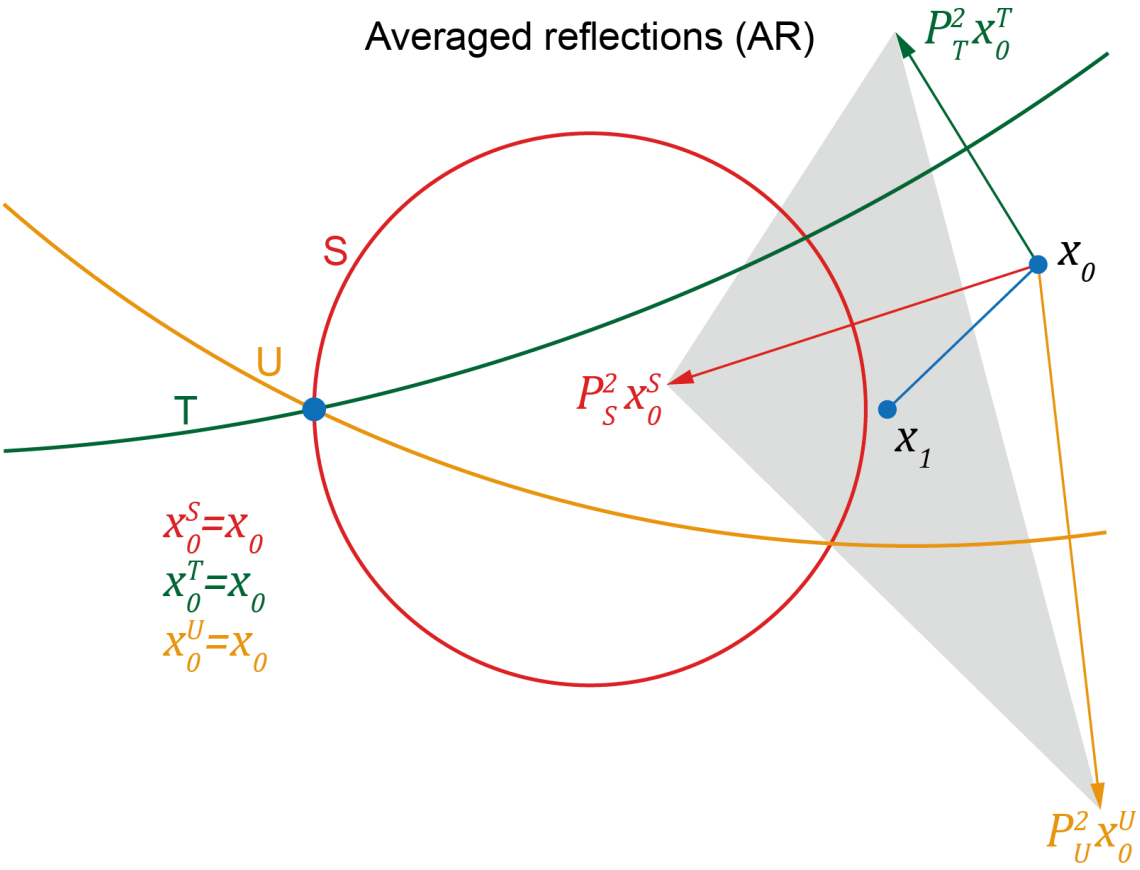

Figure 5.6. The schematic of AR, average the reflections from each constraint into one single point.

5.2.5. Solvent Flipping (SF)

Another relaxed version of the DC method is Solvent Flipping (SF). SF has quite a long history rooted in crystallography, different from AR, SF replaces the projection in the concur step with a reflection rather than in divide step [30]. The

schematic of SF is shown in Figure 5.7.

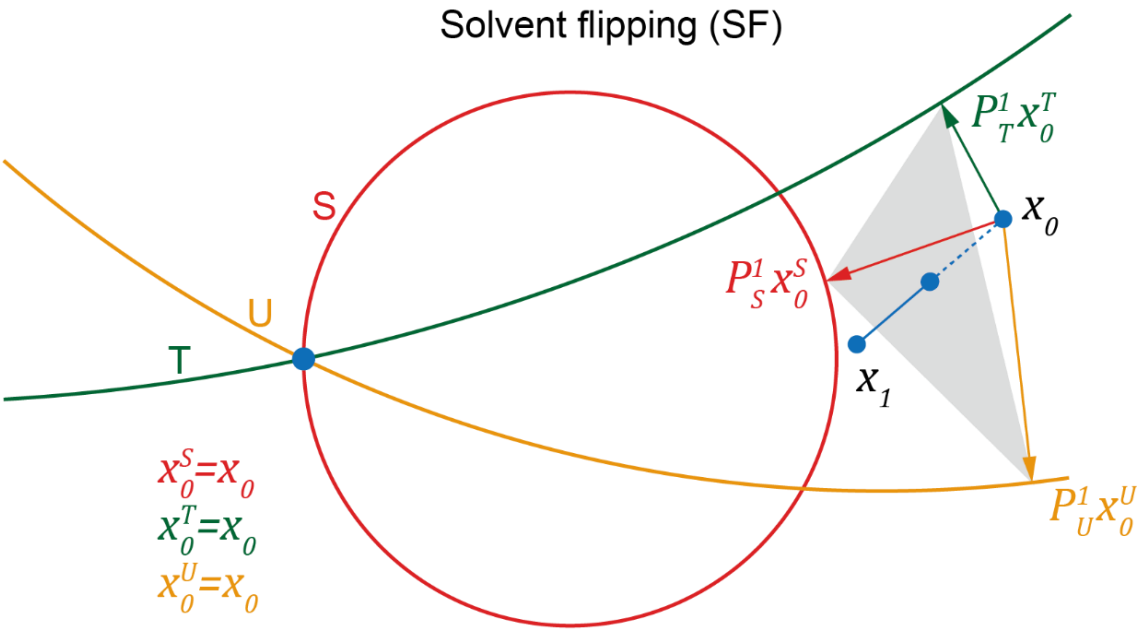

Figure 5.7 The schematic of SF, reflect the standard projections from each constraint into one single point.

In terms of DC projections, SF can be written as Equation (5.11):

$$x_{k+1} = P_C^2 P_D^1 x_k \tag{5.11}$$

Now, Figure 5.8 shows the behaviour of DC, AR and SF as they run through 100 iterations searching for the solution to the three-circle problem. All of the three algorithms become stuck at a local minima between the three constraints, and no number of further iterations can extract them from this hole. To escape local minima like this, more elaborate algorithms are required.

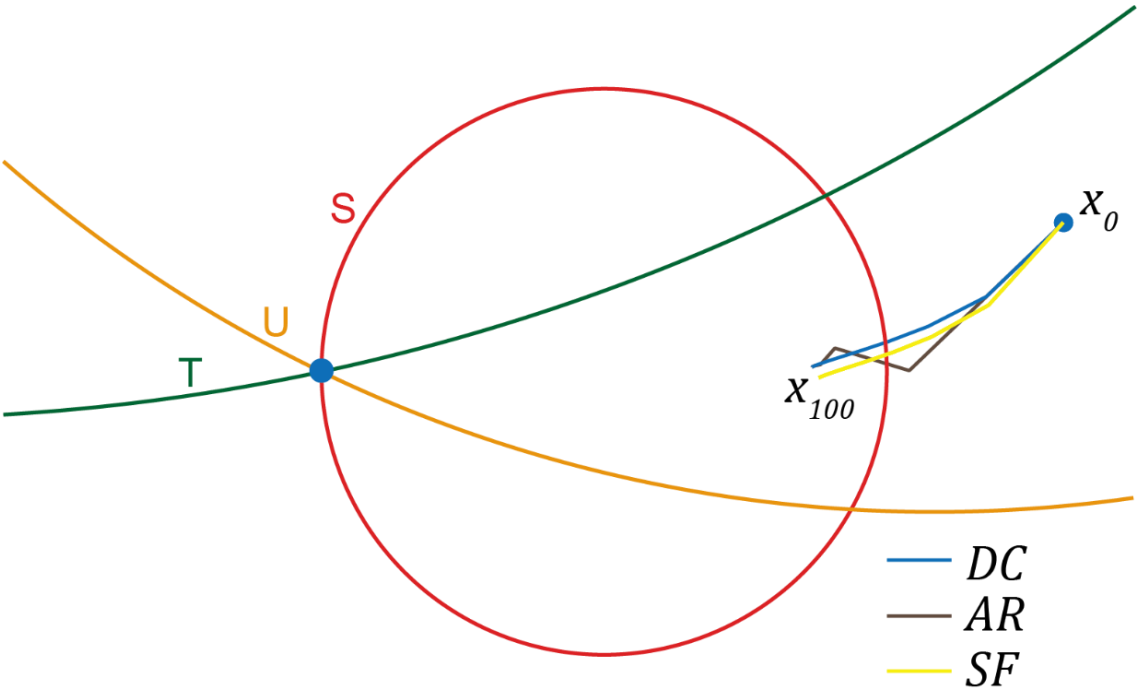

Figure 5.8. The results of DC, AR and SF for the three-circle problem, all of them are stuck at the local minima.

5.2.6. Douglas Rachford (DR)

One algorithm that can be implemented via a product space and which generally does very well at avoiding these local minima, whilst also converging considerably more quickly than DC or AR, was originally proposed by Douglas and Rachford [48].

An iteration of DR begins identically to AR, but after the divide reflection, a reflection through the concur projection point is carried out to give $P_C^2 P_D^2 x_k$ as shown in Figure 5.9 (a). So far, the algorithm could be described as RR - “reflect-reflect”, but in a final step, the twice-reflected points in the product space and the three original product space points are averaged, as shown in Figure 5.9 (c), to give Equation (5.12):

$$x_{k+1} = \frac{1}{2}(P_C^2 P_D^2 + I)x_k \qquad (5.12)$$

This formulation explains an alternative name for the DR algorithm: average successive reflections (ASR).

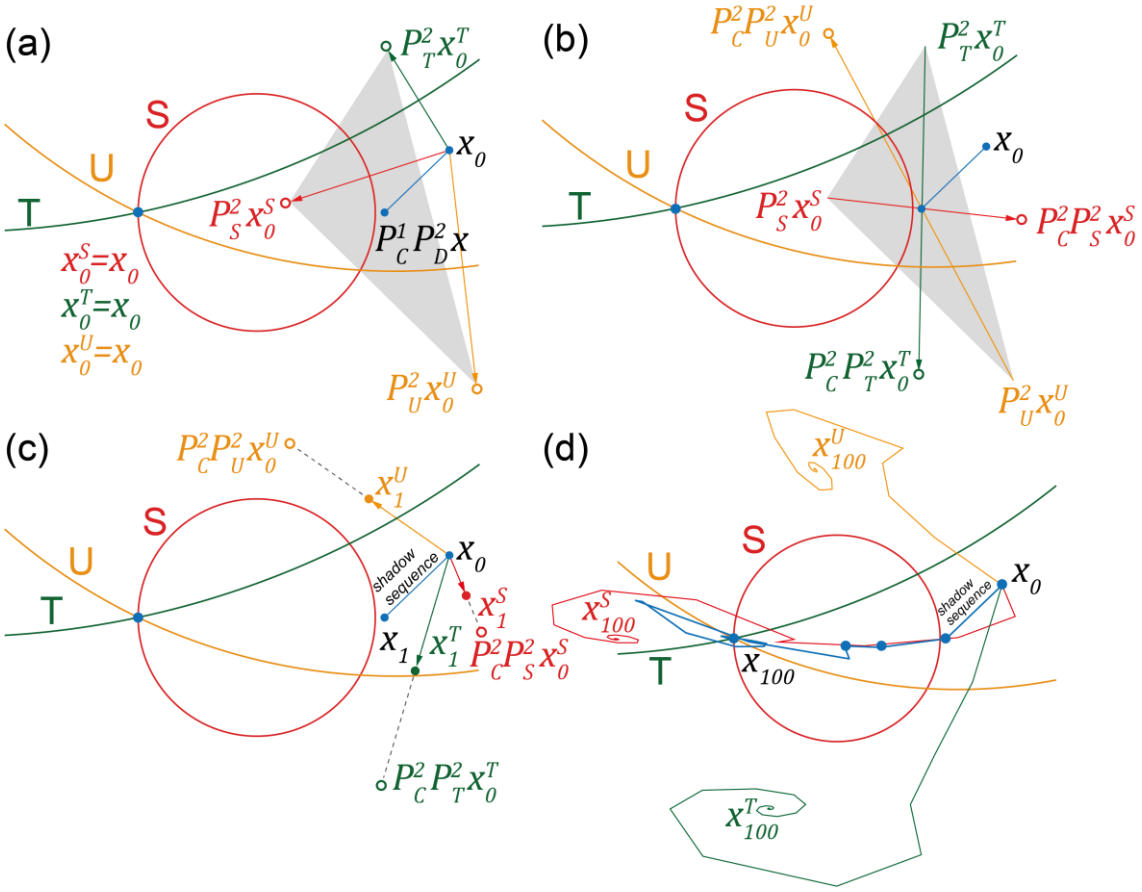

Figure 5.9. (a) The divide step in *DR* are the reflections onto each set, then average them to get a new solution point, (b) reflect each result from divide step about the new solution point, (c) take half of the twice-reflected points for next iteration, (d) are the tracks of each projections in different spaces (different color) as well as the track of concur projection of x (blue track) for 100 iterations.

A distinctive feature of DR is that the output of equation x is not a single point but three distinct points, and these points do not converge toward the solution point, or toward each other. Figure 5.9 (d) tracks the course of these three points over 100 iterations of the DR algorithm. It is the shadow sequence $P_C^1 P_D^2 x_k$ shown in Figure 5.9 (c), and extracted straightforwardly midway through each iteration of the algorithm, which converges to the solution without

stagnating, whilst the three components of x_k spiral toward points whose average is the solution. The need to store the multiple components of x_k becomes an important practical consideration for large scale optimisation problems, where computer memory must be allocated not only to hold the recorded data and the reconstructed image but also the components of x_k. For ptychography, this roughly triples the memory requirements since a complex-valued component of x_k must be stored for every (real-valued) pixel of recorded data.

5.2.7. Difference Map (DM)

Difference Map (DM) is a famous set projection algorithm that was the first algorithm that retrieved the illumination function as well as reconstruct the object [17]. The definition of DM is Equation (5.13):

$$x_{k+1} = \left\{ I + \beta \left\{ P_C^1 \left[P_D^1 + \frac{1}{\beta}(P_D^1 - I) \right] - P_D^1 \left[P_C^1 - \frac{1}{\beta}(P_C^1 - I) \right] \right\} \right\} x_k \qquad (5.13)$$

If substitute $\beta = 1$, Equation (5.13) can be rewritten as Equation (5.14):

$$\begin{aligned} x_{k+1} &= \{ I + P_C^1[P_D^1 + (P_D^1 - I)] - P_D^1[P_C^1 - (P_C^1 - I)] \} x_k \\ &= [I + P_C^1(2P_D^1 - I) - P_D^1 I] x_k \\ &= [I + P_C^1 P_D^2 - P_D^1] x_k \\ &= \frac{1}{2}(2P_C^1 P_D^2 - 2P_D^1 + 2I) x_k \\ &= \frac{1}{2}[2P_C^1 P_D^2 - (2P_D^1 - I) + I] x_k \\ &= \frac{1}{2}(P_C^2 P_D^2 + I) x_k \end{aligned} \qquad (5.14)$$

Therefore, when $\beta = 1$, DM is exactly same as DR, and this is the form of the algorithm used for ptychography.

5.2.8. Hybrid Projection Reflection (HPR)

A relaxation derived from the DR or ASR is the Hybrid Projection Reflection (HPR) algorithm [49]. HPR is an improvement of Fienup's Hybrid Input-output (HIO) algorithm [30]. In terms of DC projections, the HPR can be written as:

$$x_{k+1} = \frac{1}{2}(P_C^2(P_D^2 + (\beta - 1)P_D^1) + I + (1 - \beta)P_D^1)x_k \qquad (5.15)$$

where β is a tuning parameter.

In the practical application, HPR often used as a mixed algorithm, it cycles through a number of iterations of (5.15), followed by a number of iterations of AR, then repeats. In our tests, there are 90 iterations of HPR combined with 10 iterations of AR.

5.2.9. Relaxed Averaged Alternating Reflections (RAAR)

A further improvement of HPR is Relaxed Averaged Alternating Reflections (RAAR) algorithm. The RAAR algorithm is motivated by the hybrid projection reflection (HPR) algorithm [49] and the difference map proposed by Elser [13, 37]. It is a relaxation of the HPR and defined as Equation (5.16):

$$x_{k+1} = [2\beta P_C^1 P_D^1 + (1 - 2\beta)P_C^1 + \beta(I - P_D^1)]x_k \qquad (5.16)$$

Where β is the tuning parameter.

For equation (5.12), (5.15) and (5.16), if $\beta = 1$, DR, HPR and RAAR will coincide. Therefore, the relaxion is controlled by β. In terms of product space concept, to make the relaxion degree more intuitive, Equation (5.16) can be rewritten as Equation (5.17):

$$
\begin{aligned}
x_{k+1} &= [2\beta P_C^1 P_D^1 + (1-2\beta)P_C^1 + \beta(I - P_D^1)]x_k \\
&= [2\beta P_C^1 P_D^1 + (1-2\beta)P_C^1 + \beta I - \beta P_D^1)]x_k \\
&= \frac{1}{2}[4\beta P_C^1 P_D^1 + 2(1-2\beta)P_C^1 + 2\beta I - 2\beta P_D^1)]x_k \\
&= \frac{1}{2}\{2P_C^1[2\beta P_D^1 + (1-2\beta)I] - 2\beta P_D^1 + 2\beta I + I - I\} \\
&= \frac{1}{2}\{2P_C^1[2\beta P_D^1 - (1-2\beta)I] - [2\beta P_D^1 - (1-2\beta)I] + I\}x_k \\
&= \frac{1}{2}(P_C^1 P_D^{2\beta} - P_D^{2\beta} + I)x_k \\
&= \frac{1}{2}(P_C^2 P_D^{2\beta} + I)x_k
\end{aligned} \tag{5.17}
$$

So RAAR introduces a relaxation to the reflection in the “divide” step, compared to the DR method. Figure 5.10 illustrates the results from RAAR for the three-circle problem with $\beta = 0.85$. Compared to the DR method, the tracks of divide projections (red, yellow and green tracks) will converge to the final solution point as well as the concur projection track (blue one).

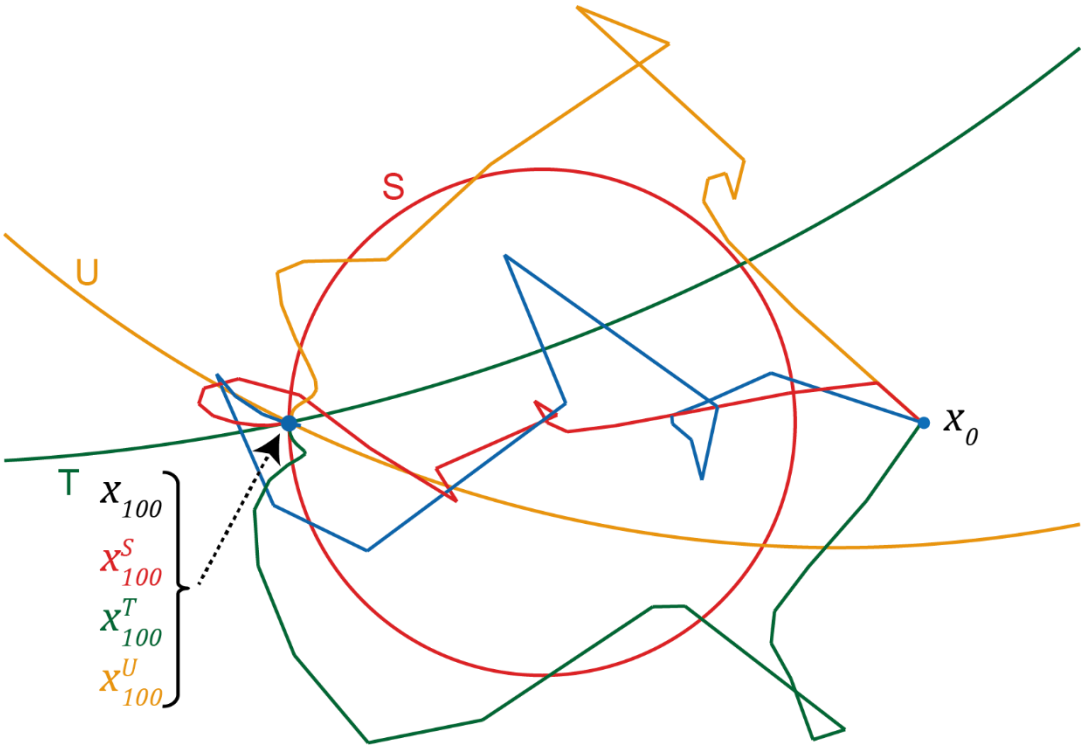

Figure 5.10. The result of RAAR for the three-circle problem, the product space also converges to the final solution (red, yellow and green tracks end at the solution point).

RAAR algorithm allows for a controlled relaxation of the constraints in the optimization problem. This relaxation parameter β can be used to control the step size between successive iterates and steer the iterates towards the solution. A small β makes the reconstruction closer to the constraint set, while a larger β explores more solution space. This flexibility allows for faster convergence and improved stability compared to the DR algorithm.

5.2.10. Reflect, reflect, relax (RRR)

Reflect, reflect, relax (RRR) algorithm is a search algorithm proposed by Elser in 2018 [50], that is used to solve combinatorial problems such as bit retrieval. It is known for its good performance and ease of implementation in two sets problem. Here, we first time apply it in the ptychographic phase retrieval problem. RRR involves two reflections, followed by an average [50], in terms of DC, it can be expressed as Equation (5.18):

$$x_{k+1} = \frac{2}{\beta} P_C^2 P_D^2 x_k + \left(1 - \frac{2}{\beta}\right) x_k \qquad (5.18)$$

The choice of the parameter β in the RRR algorithm is important. It determines the step size in the search map and affects the algorithm's convergence behaviour. Here, we use $\beta = 0.2$ in the test. The Figure 5.11 illustrates the performance of RRR in three-circle problem, RRR has a similar result to DR as it only changes the final updating weighting.

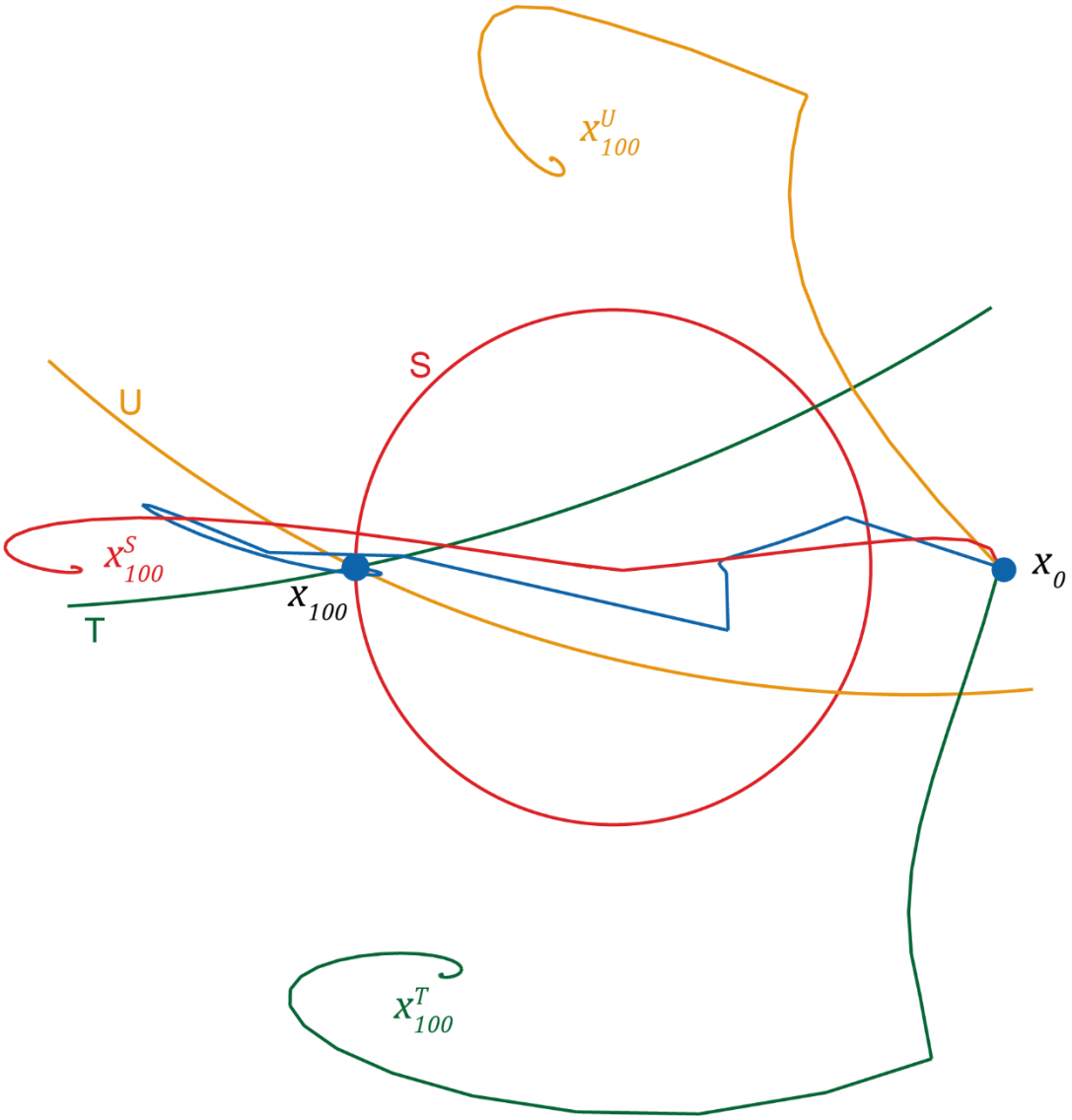

Figure 5.11. The result of RRR for the three-circle problem

5.2.11. T-lambda (T_λ)

Another new solution is T-lambda (T_λ) algorithm, proposed by Thao in 2018 [51], to solve structured optimization problems. T_λ can be viewed as a relaxation of the DR algorithm to overcome the lack of stability of DR when applied to inconsistent problems. The numerical performance of T_λ is better compared to the RAAR algorithm for both consistent and inconsistent sparse feasibility problems [51]. Likewise, we first time implement the T_λ to ptychography, defined as Equation (5.19):

$$x_{k+1} = \frac{1}{\beta+1} P_C^{\beta+1} P_D^{\beta+1} x_k + \left(1 - \frac{1}{\beta+1}\right) x_k \qquad (5.19)$$

where $\beta = 1.75$ for our test. T_λ is an over-relaxed version of DR, it increases the step in both divide projection and concur projection but finally use a smaller updating weighting.

Figure 5.12 shows the performance of T_λ in the three-circle problem, it can converge the product space as well as the global solution.

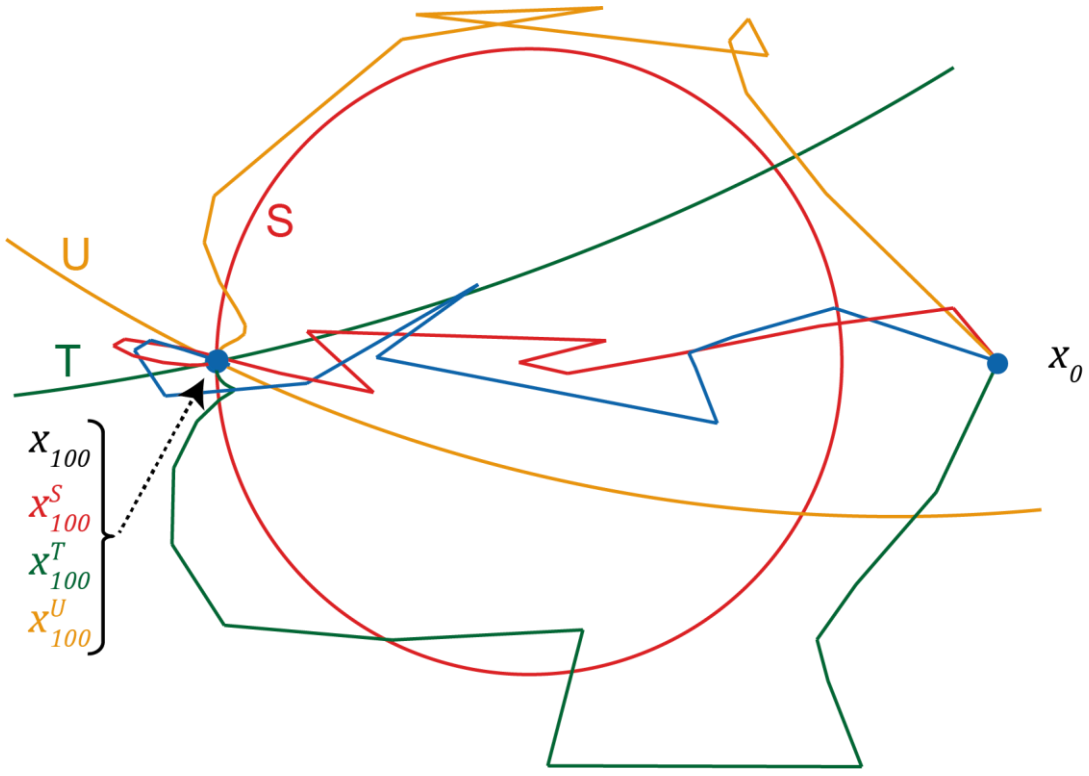

Figure 5.12 The result of T_λ for the three-circle problem

5.2.12. General Projection Algorithm

ER, AR and SF rely on local minimization strategies to address the phase problem, which can often lead to convergence toward a local minimum in various scenarios. In contrast, another type of set projection algorithm is using more global minimizers such as DR and HPR, additional feedback from its product space is introduced in these methods to reach the solution. Further

improvements are tuning the relaxation of DR or HPR, gives RAAR, RRR and T_{λ}. The adjustment of the relaxation gives better convergence and performance in set projection algorithms. To distil their shared attributes, at its most general, the two reflections within the DR algorithm can be relaxed, and the averaging step the factor 0.5 in Equation (5.12) can also be given a tunable weighting. This results in an algorithm with three tuning parameters, a, b and c, that, depending on their values, can realise many set projection algorithms from the literature. This general form we will call the general projection algorithm, the definition is given in Equation (5.20):

$$x_{k+1} = aP_C^b P_D^c x_k + (1-a)x_k \qquad (5.20)$$

where a controls the updating step from previous iteration, b is the degree of relaxation of concur projection and c is the degree of relaxation of divide projection. With appropriate choice of a, b and c this very general update formula can implement any of the product space methods we have discussed. Table 2 gives the values of the three parameters that correspond to various different algorithms, all of which roll the three tuning parameters into a single parameter, β.

Table 2. Different values of a, b and c for different set projection algorithms.

Algorithm	$\boldsymbol{a}$	$\boldsymbol{b}$	$\boldsymbol{c}$
Divide and Concur (DC) or (ER) [46]	1	1	1
Average reflections (AR) [30]	1	1	2
Solvent flip (SF) [30]	1	2	1
Douglas Rachford (DR) or (DM) [12]	0.5	2	2
Relaxed averaged alternating reflections (RAAR) [13]	0.5	2	2β
Reflect, reflect, relax (RRR) [50]	$\beta/2$	2	2
T_λ [51]	$1/(\beta+1)$	$\beta+1$	$\beta+1$

5.3. The Parameter Tuning in General Projection Algorithm

The general projection algorithm proposed in the last section, provides a flexible way to solve the set projection problem. Apart from the fixed methods (DC, AR, SF and DR), RRR, RAAR and T_λ has a tuning parameter β which could seriously affect their performance. This section will further discover different parameters for these three methods, and propose a new approach which can auto-tuning the parameters for the general projection algorithm.

5.3.1. Different Parameters for RRR, RAAR and T_λ

Continuing with the three-circle problem, different values of β have been tested and displayed in Figure 5.13. Figure 5.13 (a) is RRR with $\beta = 0.4$, which is an under-relaxed DR that only changes the final updating weight, (b) is over-relaxed one with $\beta = 1.6$. A more minor updating step makes the process smoother and reduces the spirals. Figure 5.13 (c) and (d) plots the convergence of the three product space points for the RAAR algorithm with $\beta = 0.8$, and $\beta = 0.7$. RAAR adds some under-relaxation in the divide projection compared to DR. As β approaches 1, RAAR will resemble DR. In the case of three-circle problem, β smaller than 0.8 makes it difficult to converge, see Figure 5.13 (d). This threshold varies in different problems, but generally, 0.85 is recommended for β by our own experience. T_{λ} introduces the relaxations to all three parameters, the finally updating weight is inversely proportional to the divide and concur projection steps to give a balance in global. The tracks in Figure 5.13 (f) were struggling for many iterations in a position far from the solution, if β gets smaller, it will be stuck at some local minima. A larger β in T_{λ} gives a quicker converging rate at the beginning and will avoid the local minima problem, 0.8 was chosen for our test. This behaviour is quite consistent for all of the under-relaxed versions of DR. Notice that the relaxation of one of the reflection operators helps the product space points converge to the global solution of the optimisation problem, see Figure 5.13 (c), (e) and (f). This appears to be an important factor in determining algorithm performance, with relaxed variants of DR generally working much better than DR itself for ptychography.

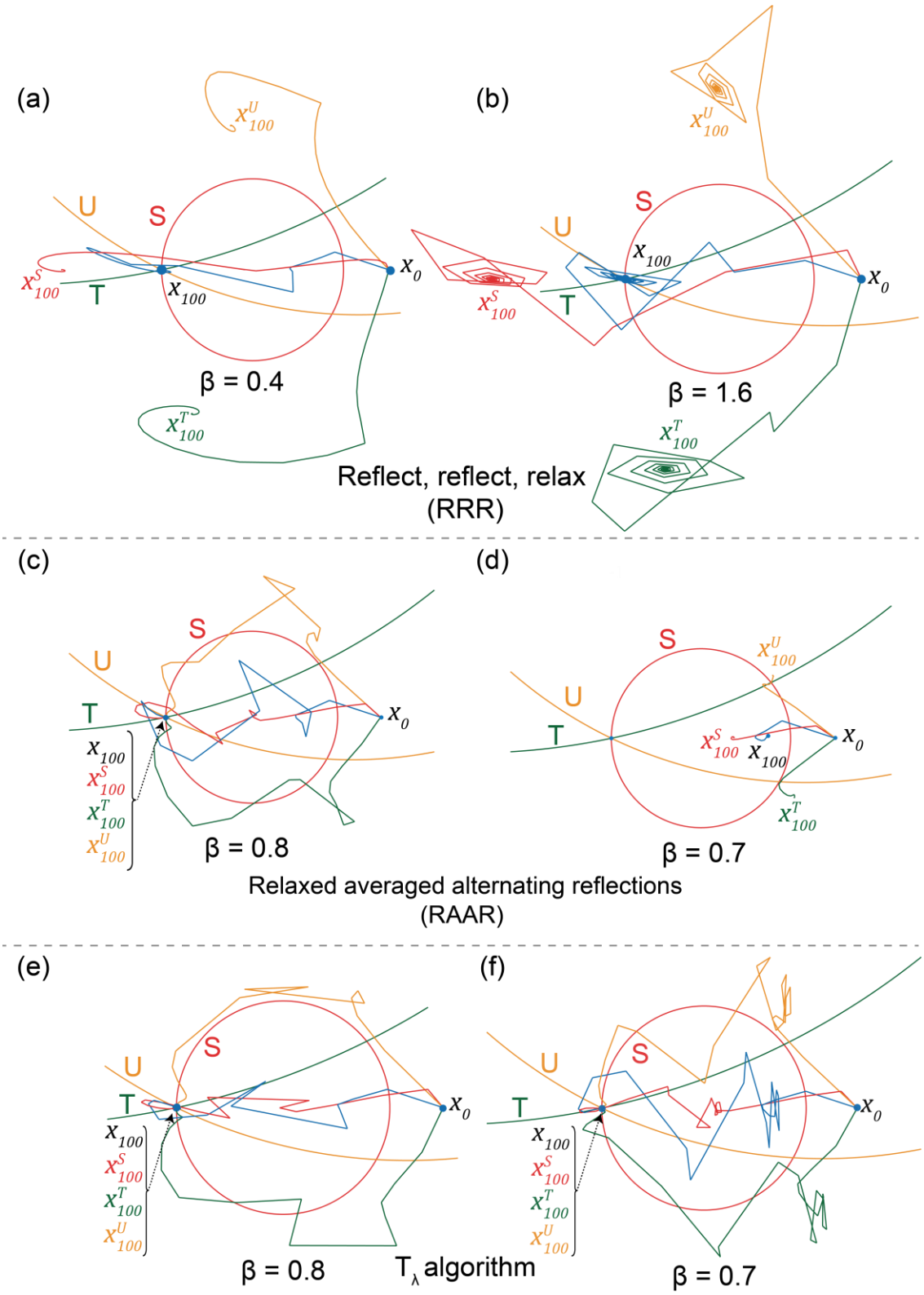

Figure 5.13. The results of different algorithms in the three-circle problem for 100 iterations, the blue track is the solution while the tracks other different color represents the projections in different product space, (a) RRR with $\beta = 0.4$, (b) RRR with $\beta = 1.6$, (c) RAAR with $\beta = 0.8$, (d) RAAR with $\beta = 0.7$, (e) T_λ with $\beta = 0.8$, (f) T_λ with $\beta = 0.7$.

5.3.2. Auto-Tuning via Bayesian Optimization

The performance of the general projection algorithm is very different, even with a tiny change in the relaxation parameters. Therefore, here is a new challenge of finding the best parameters for a particular problem, which is usually called hyperparameter optimization. Hyperparameter is an important concept in machine learning; hyperparameter optimization or tuning is choosing a set of optimal hyperparameters for the algorithm. There are several ways to optimize the hyperparameter, such as grid search, random search and Bayesian optimization. With grid search, a set of hyperparameters and performance metrics are specified, and then the algorithm iterates through all possible combinations to determine the best match. It works well, but it is relatively tedious and computationally intensive, especially when using a large number of hyperparameters. Although random search is based on similar principles as grid search, random search randomly selects a set of hyperparameters at each iteration. The method is efficient when a relatively small number of hyperparameters primarily determine the outcome of the model. Finally, Bayesian optimization is a technique based on Bayes' theorem, which describes the probability of an event occurring that is relevant to current knowledge [52]. When using Bayesian optimization for hyperparameter optimization, the algorithm constructs a probabilistic model and uses regression analysis to select the best set of hyperparameters iteratively. This is a good approach for a noisy black-box function optimization and is selected as the optimizer for our ptychographic problem.

Bayesian optimization aims to minimize a real-valued function. During the optimization, a Gaussian Process (GP) model of the objective function will be maintained and trained internally [53]. Then, an acquisition function will be used to balance sampling at points that have low modelled objective functions and exploring areas that have not yet been modelled well, to determine the next

evaluation point.

In detail, the optimization starts from sampling a random set of initial points in the hyperparameter space and evaluates the objective function values at these points. The results of these initial evaluations are used to construct the initial surrogate model. To refine the surrogate model, the Gaussian Process (GP) is used to build the surrogate model through the covariance between input data points. The Gaussian Process (GP) will assume that the objective function obeys the normal distribution, the probability distribution of the objective function is constructed by means of a covariance function (kernel function) based on the available data points. This can deal with the uncertainty of the model and provide prediction confidence intervals [54]. Once the surrogate model is built up, it can be used to predict the value of the objective function for a given combination of hyperparameters.

In addition to the model, another important concept is the acquisition function during the optimization. The acquisition of function decides the next sample point which is the new combination of hyperparameters. There are three different strategies for the acquisition function, Expected Improvement (EI), Probability of Improvement (PI), Upper Confidence Bound (UCB) [55, 56]. Expected Improvement (EI) is to find the expected improvement in the objective function value over the current best value. It is most commonly used acquisition function for most situations, especially when the objective function has more uncertainty [56]. Probability of Improvement (PI) calculates the probability that the value of the objective function exceeds the current optimal value, which is simple and suitable for the case with limited computational resources, but may be defective in the range of exploration [56]. Finally, Upper Confidence Bound (UCB) requires manual adjustment to achieve the best performance, it is suitable for long-term optimisation tasks, which can be balanced between exploration and exploitation by adjusting the parameters [56]. Based on this,

here, Expected Improvement (EI) was chosen as the acquisition function for the auto-tuning task.

In this thesis, the Bayesian optimization was implemented through MATLAB Statistics and Machine Learning Toolbox. The Gaussian Process (GP) model will be updated with ARD Matern 5/2 covariance kernel function [53, 54], defined as:

$$k(x_i, x_j) = \sigma_f^2 \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right) e^{-\sqrt{5}r} \tag{5.21}$$

where r is the Euclidean distance between x_i and x_j, σ_f is the signal standard deviation. When the auto-tuning is in progress, the quality of reconstruction and corresponding a, b, c parameters will be feed into the Gaussian process model. To sample the next point, the acquisition functions evaluate the “goodness” of a point based on the posterior distribution function from the trained model [52]. The strategy we used is 'expected-improvement', it evaluates the expected amount of improvement in the objective function, ignoring values that cause an increase in the objective [52]. If we define x_{best} as the location of the lowest posterior mean and $\mu_Q(x_{best})$ as the lowest value of the posterior mean. The 'expected-improvement' will be:

$$E(x, Q) = max\{0, \mu_Q(x_{best}) - f(x)\} \tag{5.22}$$

where Q is the posterior distribution, $f(x)$ is the Gaussian process model, μ is the mean value, max is a function that chose the max one between the inputs.

5.3.3. Generalized Auto-Tuning (GAT) Algorithm

In our case of ptychography, we proposed a new method based on Bayesian optimization called Generalized Auto-Tuning (GAT) algorithm. GAT aims to tune

the three parameters a, b, c of the general projection algorithm during the reconstruction. The tuning range are from 0.01 to 1 for a, and 0.01 to 2 for b and c. The Bayesian optimization will be called every 100 iterations, see Figure 5.14.

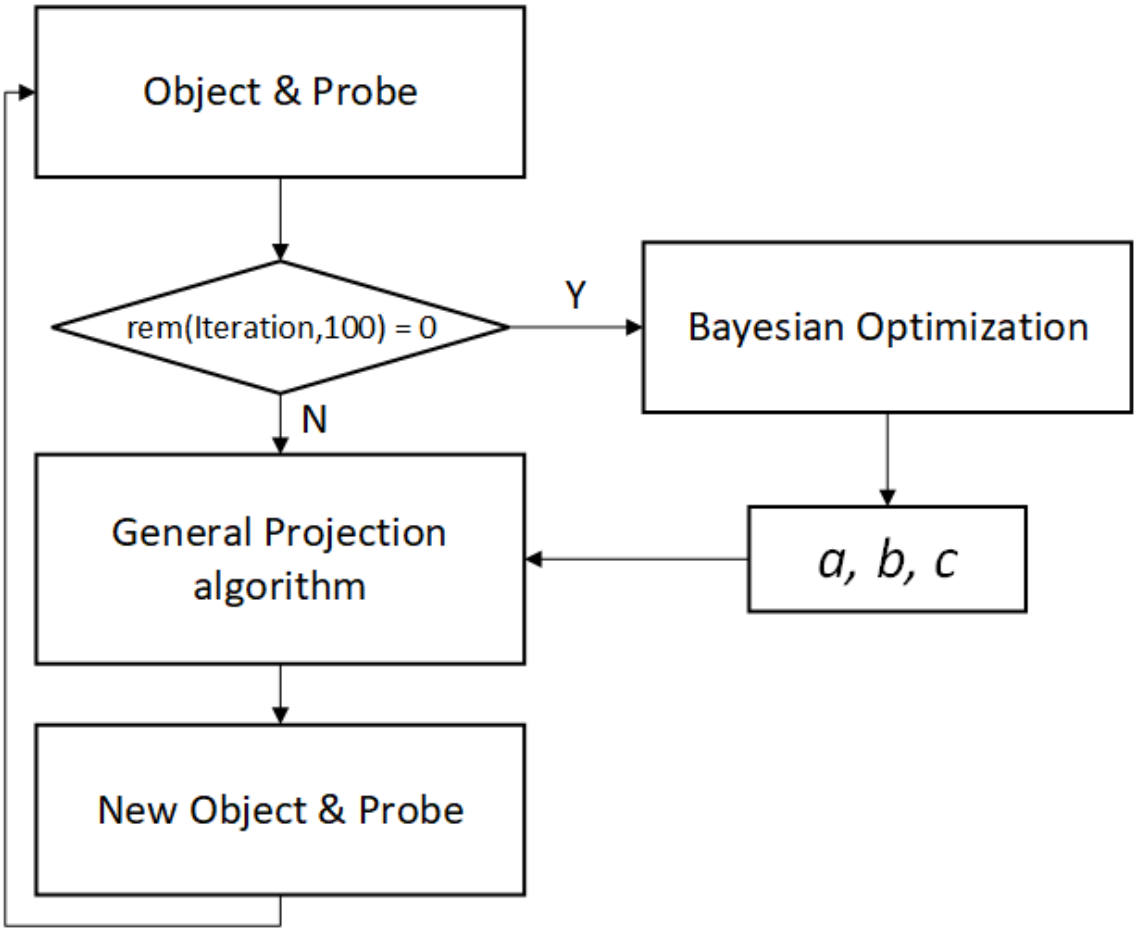

Figure 5.14. The flow chart of generalized auto-tuning algorithm, rem represents the remainder after division.

The gradient of error metric of the next 10 iterations from the current reconstructed object, probe and corresponding a, b, c values will be assessed as the objective function in the Bayesian optimization to ensure the optimized a, b, c will help the reconstruction error decrease, illustrated in Figure 5.15. Moreover, the optimization will be compared to the gradient from the previous 10 iterations, to avoid unnecessary change of a, b, c. Finally, because a large and sudden change of the relaxation in projection sometimes will destroy the

whole reconstruction, an extra insurance is added after tuning. If the gradient goes up successively, the a, b, c values will be reset as the one before tuning. The initial values are 1 for all of a, b, c, which is ER (DC) from Table 2. The worst case is that the tuning is unsuccessful all the time, then it will remain using ER.

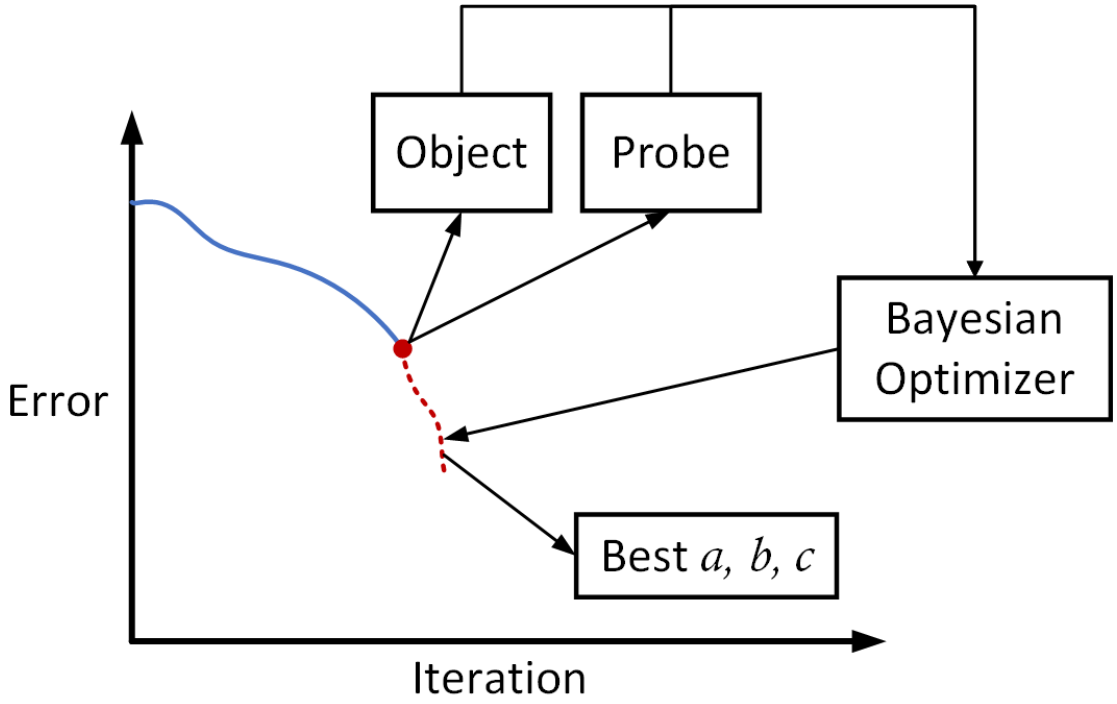

Figure 5.15. The process of the Bayesian Optimization. When the optimization is called at the red point, the reconstruction of the object and probe at this point will be feed to the Bayesian Optimizer. The optimizer aims to reduce the gradient for the next few iterations, the best one will provide the best values for a, b, c.

5.4. Set Projection in Ptychography

In the previous sections, we implemented different set projection methods to the three-circle problem and discussed the effects of the tuning parameters in these approaches. Typically, ptychography is a phase retrieval problem, which is inherently non-convex [14, 57]. This is because the relationship between the measured intensities and the phase information involves quadratic constraints, making the optimization landscape non-convex with many local minima [14, 58].

The set projection ptychographic solutions such like Difference Map (DM) are designed to addresses the non-convexity of phase retrieval. The set projection algorithms in ptychography minimize the error by balancing the projections, gradually improving the consistency with both the measured data and the overlap constraints. In this section, the connection between set projection and ptychography will be described in detail as well as the applications of set projection methods in ptychography.

5.4.1. Iterative Ptychographic Phase Retrieval

As mentioned in pervious Chapter 3.2, in ptychography, a moving probe illuminates a part of the object at a time, collecting the diffraction patterns for different positions. Following the previously mentioned notation, denote the object by $o_{k_j}(\vec{r})$ the probe by $P_j(\vec{r})$, where k is the scan position and j represents current iteration number. The exit wave propagates to the detector is $\psi_k(\vec{r})$. The intensity recorded by the detector, representing the diffraction pattern, is denoted as $I_k(\vec{u})$. Since the diffraction pattern $I_k(\vec{u})$ only encompasses the modulus information of the exit wave, the phase is lost during the propagation, that is the phase problem in microscope imaging. Fienup [31] proposed a remarkable solution to the phase problem in 1980s, building upon the pioneering work of Gerchberg and Saxton [47]. This method was described as single-shot phase retrieval in Figure 2.8. Contrast to it, later, more advanced ptychographic iterative solutions have a similar computational loop as the single-shot method, shown in Figure 5.16. Ptychographic iterative method initially starts from point A with the initial estimation of object and probe, form the exit waves for all the scan positions. Then, from A to B, do the Fourier transform to get estimated diffraction patterns, $\Psi_k(\vec{u}) = \boldsymbol{\mathcal{F}}\{\psi_k(\vec{r})\}$. Subsequently, the estimated phase is retained while the modulus is replaced with measurements from the detector, $\Psi'_k(\vec{u}) = \sqrt{I_k(\vec{u})}\frac{\Psi_k(\vec{u})}{|\Psi_k(\vec{u})|}$. Now, with the corrected modulus, an inverse Fourier transform is executed to return to real

space, $\psi'_k(\vec{r}) = \mathcal{F}^{-1}\{\Psi'_k(\vec{u})\}$. Finally, update the object and probe, the new estimation will be used for the next iteration. The sequential projection methods in ptychography such like ePIE, mentioned in Chapter 3.2.7, solve this problem position by position, which means that the next computation for the next position is dependent on the last one. Unlike this, set projection methods in ptychography treat every position equally. The calculation for each position is independent and can be parallelized.

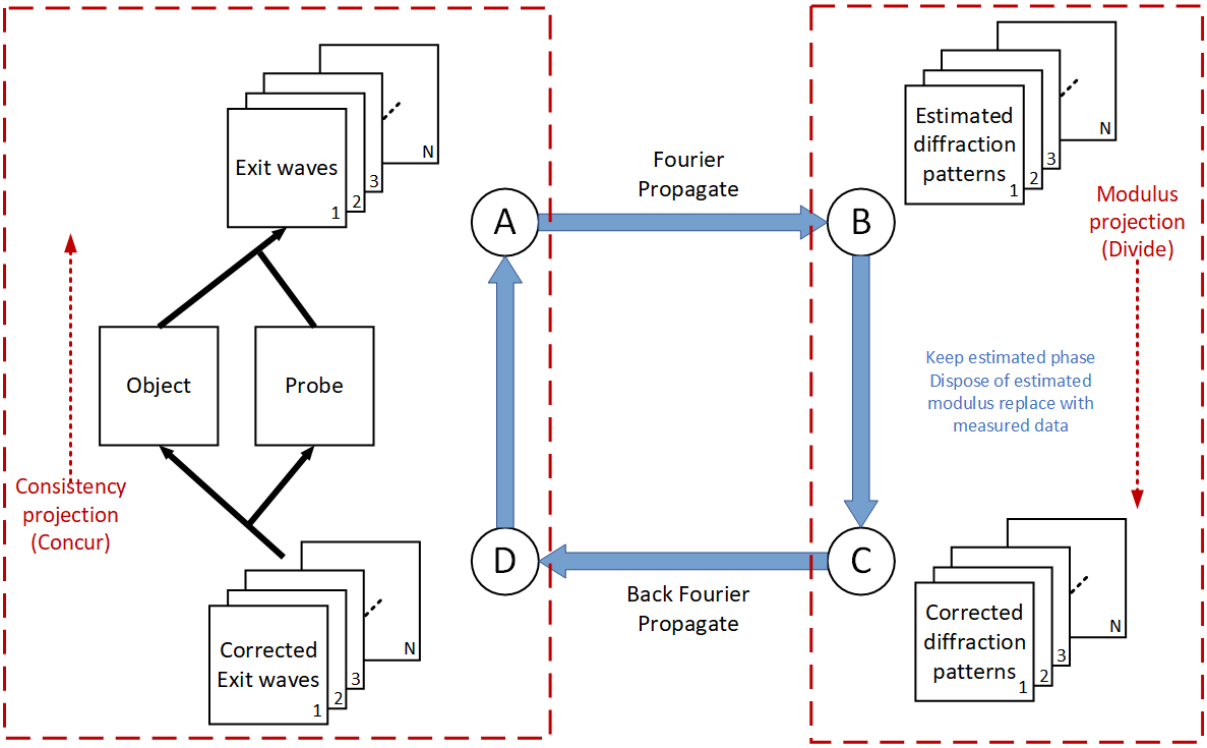

Figure 5.16. The schematic of iterative ptychographic solution. Right part indicates the ‘divide’ step in ptychography and the left part is the ‘concur’ step.

5.4.2. Constraint Sets in Ptychography

In the aspect of set projection, there are two main constraints in ptychography, the first one is the ‘modulus constraint’, which is the prior knowledge from the measurements obtained from the detector. As shown in the red dashed box on the right side in Figure 5.16, the process of replacing the modulus with measured data is the modulus projection in ptychography, the replacement is

pixel to pixel for each diffraction pattern. Each pixel here manifests as a point with given complex value in the Fourier space, the measurement is a circle with the radius defined by measured modulus, same as the example shown in Figure 5.1. The modulus projection for each pixel is to find the closest point on the circle from the given point, that why the three-circle problem was introduced earlier—to serve as an example of ptychography's constraint in the Fourier domain. The product space here will contain N copies of Euclidean spaces, each one has same size as the object and hold one diffraction pattern. The projections onto the modulus constraint are individual with respect to each diffraction pattern. Therefore, the idea of 'divide' can be introduced into modulus projections. All of modulus projections can be done in parallel at the same time.

On the other hand, the aperture in ptychography defines the support, all the pixels outside the support will be zero, this is certain information in the experiment. Also, all these adjacent scan positions are overlapped with each other to some extent in ptychography. This known support and the overlapping areas provide another important constraint called 'consistency constraint' or 'single-probe-and-object constraint'. Essentially, that means the portion of object within the support in the same position across any diffraction pattern remains consistent all the time, likewise, the probe is always the same as well. The consistency projection is the step between D to A in the Figure 5.16, mainly for separating the probe and object from the form of the product and updating the probe and object for the next iteration. It is also can be considered as the 'concur' step. Therefore, iterative ptychography applies these two constraint sets to the reconstruction alternately, which alternates between the divide and concur steps.

5.4.3. Updating the Object and Probe (Sequential Projection)

Also, in order to reconstruct the object and probe, the sequential projection methods are different from the set projection methods. The sequential

projection methods step through all the diffraction patterns, usually in a random order, which is satisfying the modulus constraints one by one. Therefore, the reconstruction can be considered as minimizing the regularized cost functions shown in Equation (5.23) and (5.24):

$$\mathcal{L}_O = \sum_{\vec{u}} \left(\left| \boldsymbol{\mathcal{F}} \left\{ P_j(\vec{r}) {o_k}_j{}'(\vec{r}) \right\} \right| - \sqrt{I_k(\vec{u})} \right)^2 + \sum_{\vec{r}} A \left({o_k}_j{}'(\vec{r}) - {o_k}_j(\vec{r}) \right)^2 \qquad (5.23)$$

$$\mathcal{L}_P = \sum_{\vec{u}} \left(\left| \boldsymbol{\mathcal{F}} \left\{ P_j{}'(\vec{r}) {o_k}_j(\vec{r}) \right\} \right| - \sqrt{I_k(\vec{u})} \right)^2 + \sum_{\vec{r}} B \left(P_j{}'(\vec{r}) - P(\vec{r}) \right)^2 \qquad (5.24)$$

Where ${o_k}_j{}'(\vec{r})$ is the new object estimate, and $P_j{}'(\vec{r})$ is the new probe estimate in j^{th} iteration at k^{th} position. A and B are the regularization functions that dictate how strongly the reconstruction are anchored to their previous estimate. ePIE as an example of sequential projection method, the regularization functions of it are Equation (5.25) and (5.26) [18]:

$$A = \frac{1}{\alpha} \, |P_j(\vec{r})|_{max}{}^2 - |P_j(\vec{r})|^2 \qquad (5.25)$$

$$B = \frac{1}{\beta} \, |{o_k}_j(\vec{r})|_{max}{}^2 - |{o_k}_j(\vec{r})|^2 \qquad (5.26)$$

where the tuning constants α and β are usually set to unity [17], max represent the maximum value. Also, an improved version of this called regularized PIE (rPIE) uses the regularization functions as Equation (5.27) and (5.28) [18]:

$$A = \alpha \left(\left| P_j(\vec{r}) \right|^2{}_{max} - \left| P_j(\vec{r}) \right|^2 \right) \qquad (5.27)$$

$$B = \beta \left(\left| o_j(\vec{r}) \right|^2{}_{max} - \left| o_j(\vec{r}) \right|^2 \right) \qquad (5.28)$$

where α and β are the tuning parameters. When $\alpha = 1$ and $\beta = 1$, it

coincides with ePIE.

Minimizing the cost functions in Equation (5.23) and (5.24) is to satisfy the modulus constraint, which can be considered as a projection. This minimization can be solved by taking Wirtinger derivatives of the cost functions and setting the result to zero. This gives the updating functions for the object and probe in Equation (5.29) and (5.30):

$$o_{k_j}{}'(\vec{r}) = o_{k_j}(\vec{r}) + \frac{P_j^*(\vec{r})\left(\psi_{k_j}{}'(\vec{r}) - \psi_{k_j}(\vec{r})\right)}{|P_j(\vec{r})|^2 + A} \tag{5.29}$$

$$P_j'(\vec{r}) = P_j(\vec{r}) + \frac{o_{k_j}^*(\vec{r})\left(\psi_{k_j}{}'(\vec{r}) - \psi_{k_j}(\vec{r})\right)}{|o_{k_j}(\vec{r})|^2 + B} \tag{5.30}$$

Note that when $\alpha = 1$ and $\beta = 1$, this is the same as Equation (3.13) and (3.14), shown in the previous ePIE example. The object updating function in Equation (5.29) only reconstruct a fraction of the entire object, since it only deals with one scan position. After all the positions solved, the entire object will be reconstructed completely.

5.4.4. Updating the Object and Probe (Set Projection)

Unlike the sequential projection methods, set projection methods consider all the scan positions in a single batch. The cost functions for set projection method are defined as Equation (5.31) and (5.32):

$$\mathcal{L}_O = \sum_k \sum_{\vec{u}} \left(\left| \boldsymbol{\mathcal{F}} \left\{ P_j(\vec{r}) o_{k_j}{}'(\vec{r}) \right\} \right| - \sqrt{I_k(\vec{u})} \right)^2 \tag{5.31}$$

$$\mathcal{L}_P = \sum_k \sum_{\vec{u}} \left(\left| \boldsymbol{\mathcal{F}} \left\{ P_j{}'(\vec{r}) o_{k_j}(\vec{r}) \right\} \right| - \sqrt{I_k(\vec{u})} \right)^2 \tag{5.32}$$

The minimization of these two cost functions is the same way, taking the

derivatives and setting the result to zero, gives the updating functions for the object and probe, written as Equation (5.33) and (5.34):

$$O'_j(\vec{r}) = \frac{\sum_k P_j^*(\vec{r})\psi'_{k\,j}(\vec{r})}{\sum_k |P_j(\vec{r})|^2} = \frac{\sum_k |P_j(\vec{r})|^2 \frac{\psi'_{k\,j}(\vec{r})}{P_j(\vec{r})}}{\sum_k |P_j(\vec{r})|^2} = \sum_k \frac{\psi'_{k\,j}(\vec{r})}{P_j(\vec{r})} \tag{5.33}$$

$$P'_j(\vec{r}) = \frac{\sum_k o_{k\,j}^*(\vec{r})\psi'_{k\,j}(\vec{r})}{\sum_k |o_{k\,j}(\vec{r})|^2} = \frac{\sum_k |o_{k\,j}(\vec{r})|^2 \frac{\psi'_{k\,j}(\vec{r})}{o_{k\,j}(\vec{r})}}{\sum_k |o_{k\,j}(\vec{r})|^2} = \sum_k \frac{\psi'_{k\,j}(\vec{r})}{o_{k\,j}(\vec{r})} \tag{5.34}$$

This is a straightforward way to separate the object and probe from its product which is the corrected exit wave. In the set projection methods, the modulus projection for each position is independent and can be parallelized since each position does not require any information from the others. In Equation (5.33) and (5.34), all the corrected exit waves are lying in its own space. They are the results come from the modulus projections, each one only holds the prior knowledge from one diffraction pattern, which is a part of information about the object. Then, Equation (5.33) and (5.34) aggregate all the planes across the product space, gives a global updating for the object and probe. This step combines all the individual projections from the product space into a single consensus solution, also can be considered as the ‘concur’ projection.

More specific expression of ‘divide and concur’ in ptychography can be written as Equation (5.35), (5.36) and (5.37):

$$\psi_k{}'_j(\vec{r}) = P_D^1\psi_{k\,j}(\vec{r}) = P_m^1\psi_{k\,j}(\vec{r}) = \boldsymbol{\mathcal{F}^{-1}}\left\{\sqrt{I_k(\vec{u})}\frac{\boldsymbol{\mathcal{F}}\left\{\psi_{k\,j}(\vec{r})\right\}}{\left|\boldsymbol{\mathcal{F}}\left\{\psi_{k\,j}(\vec{r})\right\}\right|}\right\} \tag{5.35}$$

$$\psi_{k_{j+1}}(\vec{r}) = o'_{k_j}(\vec{r})P'_j(\vec{r})$$
$$= O'_j(k+\vec{r})P'_j(\vec{r}) \quad (5.36)$$

$$\psi_{k_{j+1}}(\vec{r}) = P_C^1 P_D^1 \psi_{k_j}(\vec{r}) \quad (5.37)$$

where the $O'_j(k+\vec{r})P'_j(\vec{r})$ is calculated by Equation (5.33) and (5.34) can be considered as P_C^1. Set projection algorithms for ptychography oscillates between these two steps as shown in Equation (5.37). The pseudocode of the general projection algorithm in ptychography is shown in **Pseudocode 5.1** and **Pseudocode 5.2** as the example. All the set projection methods mentioned in the Table 2 can be implemented via the general projection algorithm with different a, b, c parameters.

Pseudocode 5.1: General Projection Algorithm in Ptychography

Inputs: *obj* (obj), *probe* (probe), *intensity* (I), *the total number of positions* (K), *the total number of iterations* (J), *tuning parameter* (a, b, c).
Outputs: *exit waves* (exitWave).

```
For (k = 1 to K) do
  // Calculate the exit waves for all the positions
  exitWave_k = obj(R_k to R_k + [M,N]) · probe
End loop
For (j = 1 to J) do
  For (k = 1 to n) do
    // Product space projection
    cProj = b·obj(R_k to R_k + [M,N])·probe – (1-b)·exitWave_k
    // Modulus projections, applying measurements
    mProj = c·(sqrt(I_k)·ℱ(cProj)/abs(ℱ(cProj))) + (1-c)·cProj
    // Update exit waves
    exitWave_k = a·mProj + (1-a)·mProj
  End loop
End loop
// Update the object and probe
// Apply any additional constraints
```

Note: $\mathcal{F}$: Fourier transform. **conj**: complex conjugate. **abs**: amplitude. **sqrt**: square root.

The updating step of the object and probe is shown in **Pseudocode 5.2**.

Pseudocode 5.2: The Updating of the Object and Probe

Inputs: *obj* (obj), *probe* (probe), *exit waves* (exitWave), *the total number of positions* (K).
Outputs: *obj* (obj), *probe* (probe).

```
    // Initialise numerator and denominator sums
1   topO = bottomO = zeros(X,Y)
2   topP = bottomP = zeros(M,N)
3   For (k = 1 to K) do
      // Update numerator and denominator sums for the probe
4     topP += conj(obj(Rk to Rk+[M,N]))·exitWavek
5     bottomP += abs(obj(Rk to Rk+[M,N]))²
6   End loop
7   For (k = 1 to K) do
      // Update numerator and denominator sums for the object
8     topO(Rk to Rk+[M,N])    += conj(probe)·exitWavek
9     bottomO(Rk to Rk+[M,N]) += abs(probe)²
10  End loop
11  obj = topO/(bottomO + eps)
12  probe = topP/(bottomP + eps)
```

Note: zeros: a matrix full of zeros. ***eps***: a small constant in MATLAB to avoid dividing 0. **abs**: amplitude. **conj**: complex conjugate.

5.5. Comparison of Different Set Projection Algorithms

In this section, several ptychographic simulations will be applied to different set projection algorithms and the GAT method. The results will be displayed and analysed in the later part.

5.5.1. Simulation Configuration

Figure 5.17 illustrates the object used for simulations, which is a complex-valued image of frog's red blood cells derived from a real-world optical bench ptychography experiment [59]. Figure 5.17 (a) is the modulus of the object, the red box delineates the reconstruction area, red circles indicate two different sizes of probe, and the blue box shows the area of error calculated.

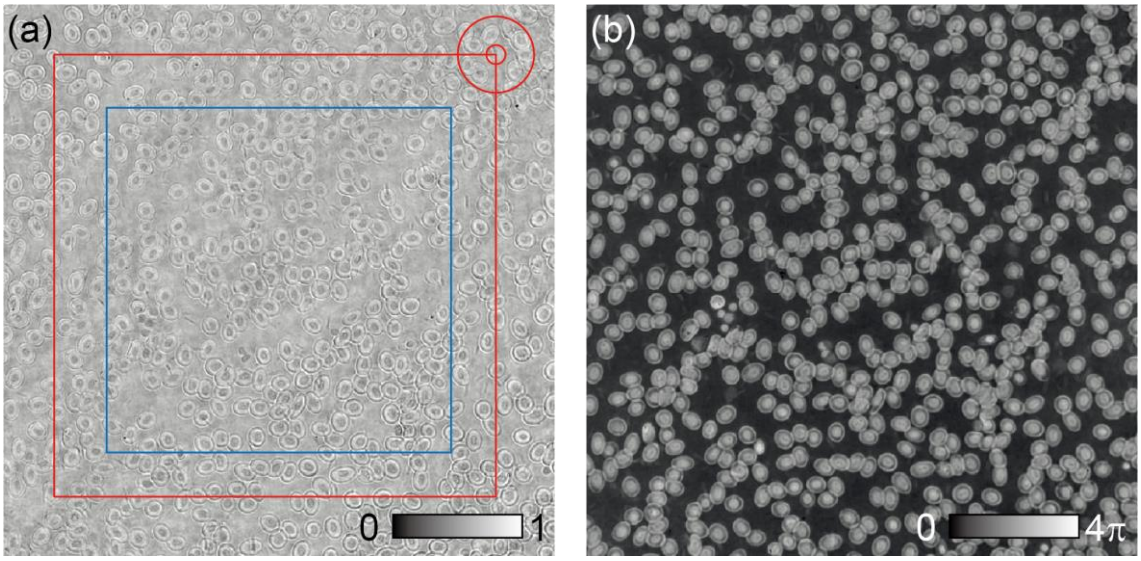

Figure 5.17. The simulation object (a) the modulus of object, (b) the phase of object. The red box is the reconstruction area, red circles are two different sizes of probe, and the blue box is the area where error calculated.

The two different sizes of probe in the red circles in Figure 5.17 (a), are shown in Figure 5.18.

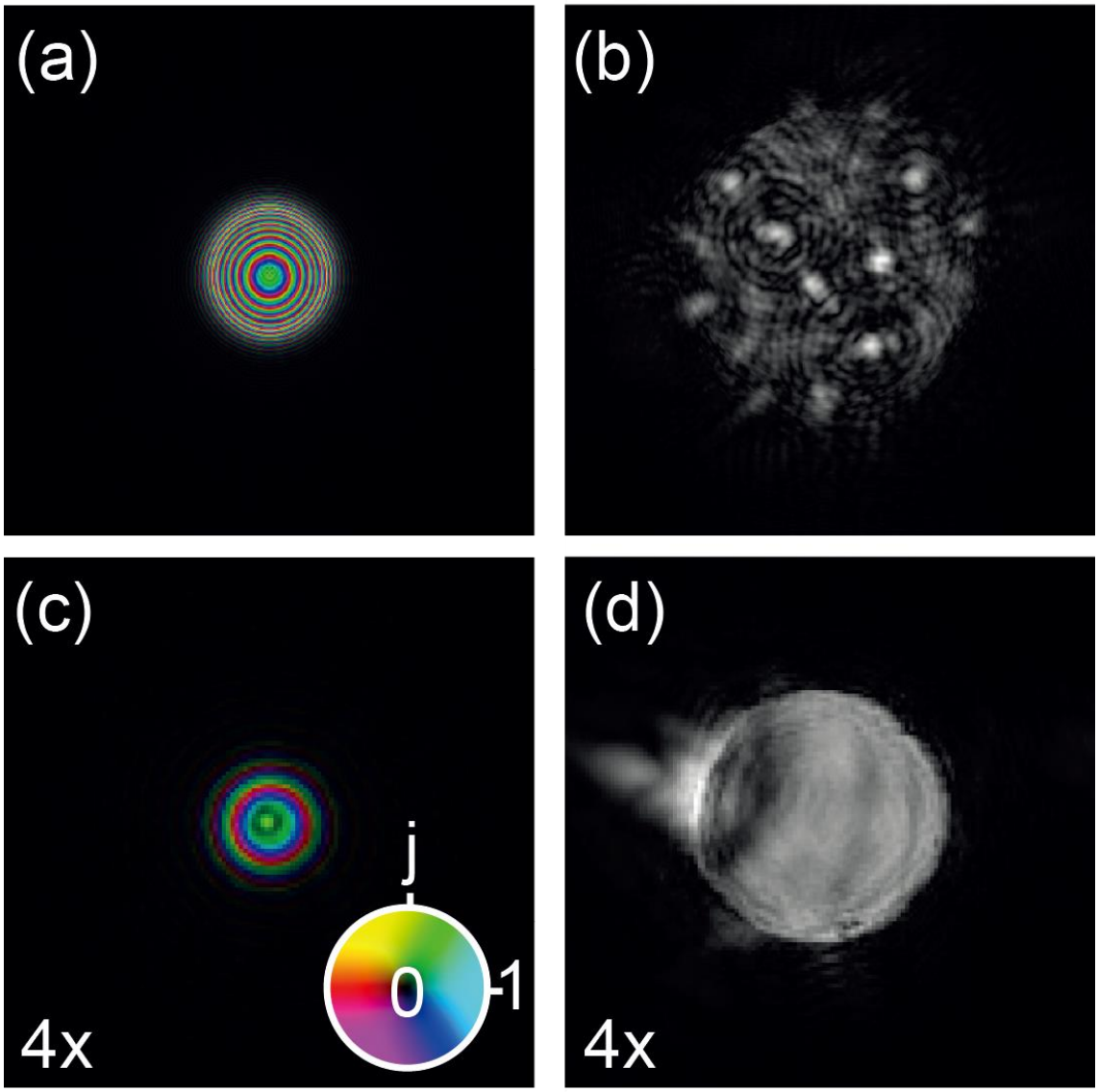

Figure 5.18. Two different sizes of probe. (a) is the large size (512×512 pixels) probe, and (b) is one of the diffraction patterns for its simulation. (c) is the small size (128×128 pixels) probe, and (d) is one of the diffraction patterns for its simulation. (c) and (d) are zoomed in four times.

For the large size simulation, there are totally 400 scan positions arranged in a 20×20 grid with an average step size of 36 pixels and $\pm 20\%$ random offsets. The small size simulation has a 80×80 scan grid with an average step size of 6 pixels and $\pm 20\%$ random offsets, 6400 positions in total.

Apart from the blood cell object simulation, a noisy dataset with a designed

object with phase change is used to test different algorithms, see Figure 5.19(a). Poisson distributed noise is added to this designed object. In practice, the noise level is usually expressed by the number of electrons or other particles that arrive at the detector plane, which is normally called “counts”. The diffraction patterns with different level Poisson noise are shown in Figure 5.19 (b-i). As shown in the figures, in this test, when the counts is below 10^6, there are many noticeable noise spots on the diffraction patterns. The brightness of the diffraction pattern saturates when the counts is greater than 10^8, this is also corroborated by the later SNR calculations.

To demonstrate the effect of Poisson noise to the reconstruction, ER was selected to run the reconstructions at different levels of Poisson noise since it is the most stable algorithm under the noise situation. There are 8 different levels of noise, the results of them are shown in Figure 5.20 (a). Correspondingly, the average Signal-to-Noise Ratio (SNR) between the diffraction patterns and the noiseless one for each noise level was calculated and displayed in Figure 5.20 (b).

From Figure 5.20, the reconstruction quality is decreasing as the level of noise increasing. Combining their SNR value, in this thesis, we will choose 10^4 counts as a very noise test and 5×10^5 counts for a moderate noisy situation for all the ptychographic algorithms in later sections.

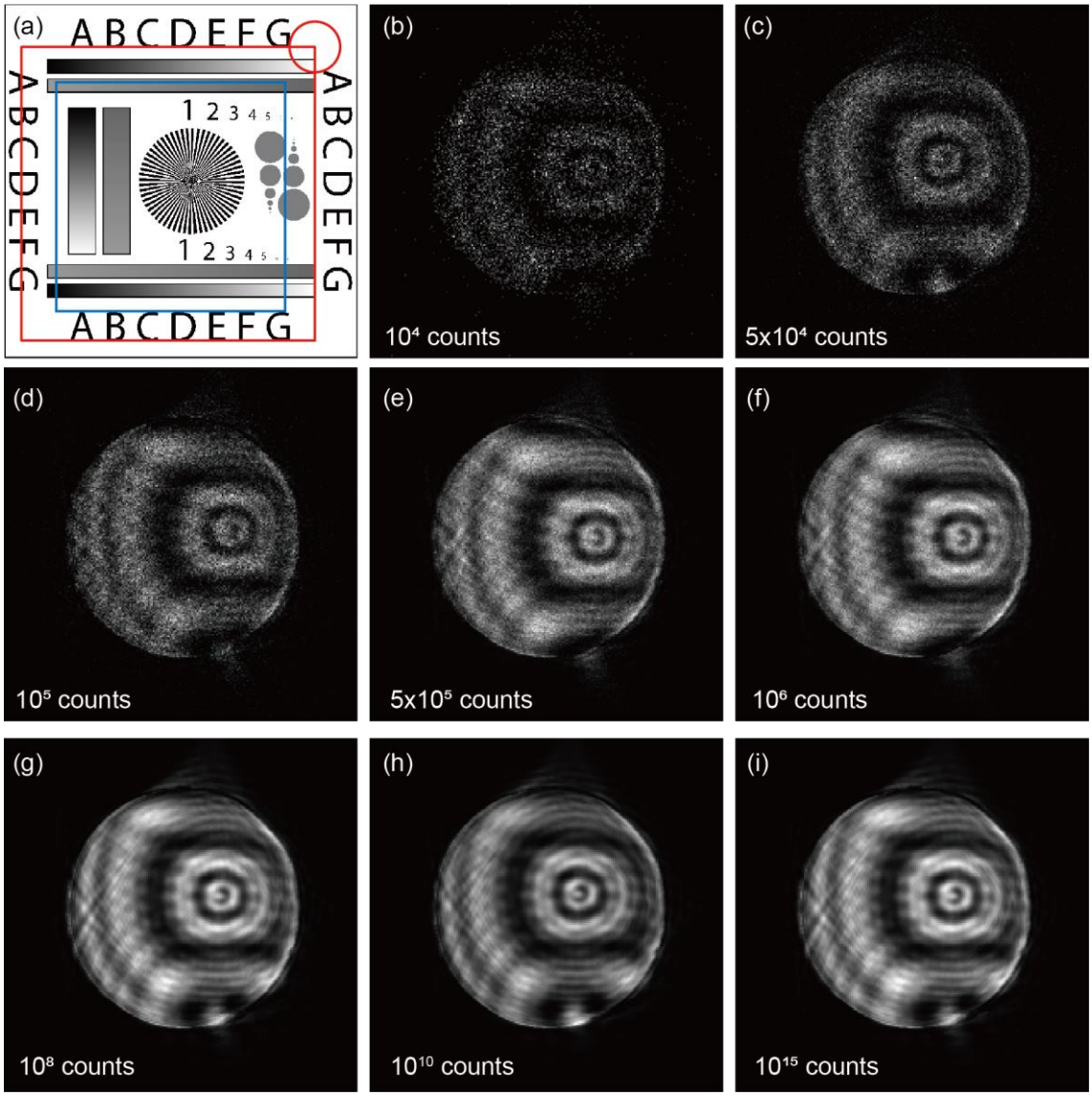

Figure 5.19. (a) The phase object for the noise test, the red box is the scan area, and the circle indicates the probe size. The blue area is error calculation box. (b) The diffraction pattern with 10^4 counts. (c) The diffraction pattern with 5×10^4 counts. (d) The diffraction pattern with 10^5 counts. (e) The diffraction pattern with 5×10^5 counts. (f) The diffraction pattern with 10^6 counts. (g) The diffraction pattern with 10^8 counts. (h) The diffraction pattern with 10^{10} counts. (i) The diffraction pattern with 10^{15} counts.

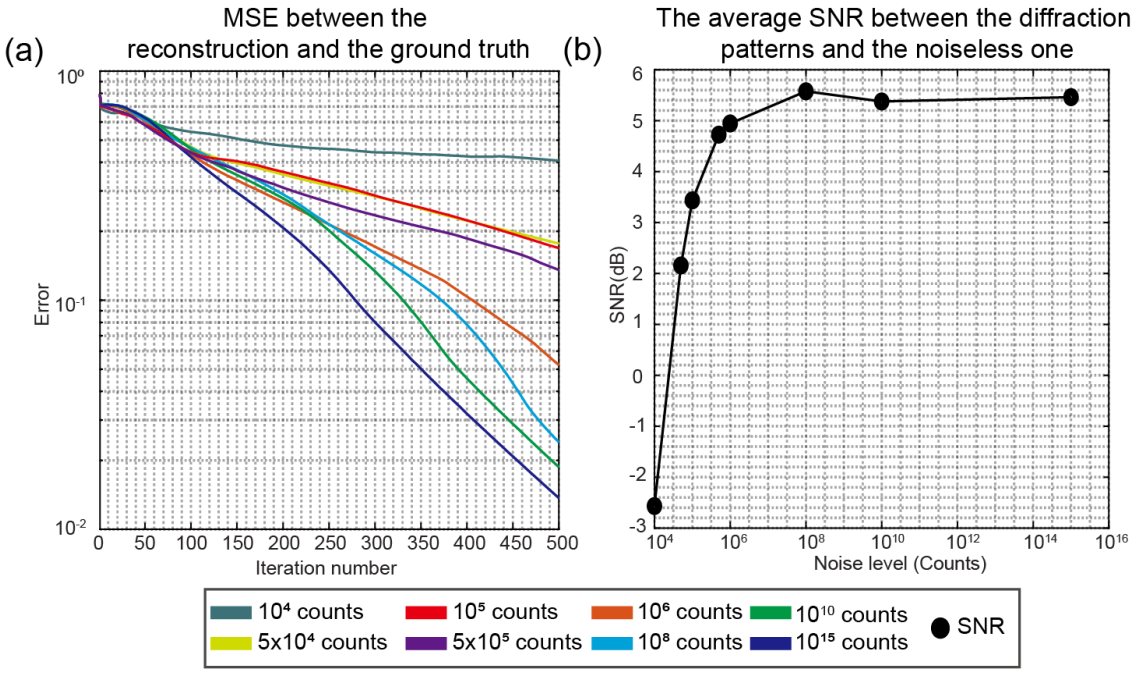

Figure 5.20. (a) The error metric of ER at different levels of Poisson noise. (b) The average SNR at each noise level.

5.5.2. Parameter Tuning for RRR, RAAR and T_λ

In order to get the best performance of RRR, RAAR and T_λ, the simulation with large size probe is used to test different values of β. Here, we tested $\beta = 0.85$, $\beta = 0.7$, $\beta = 0.5$, $\beta = 0.2$. All the conditions result in a under relaxed DR in the final updating step. Note that when $\beta > 1$, which is an over relaxed DR, RRR fails in our test. The test results are shown in Figure 5.21. The error is calculated by Equation (3.18), which was introduced in Chapter 3.4.

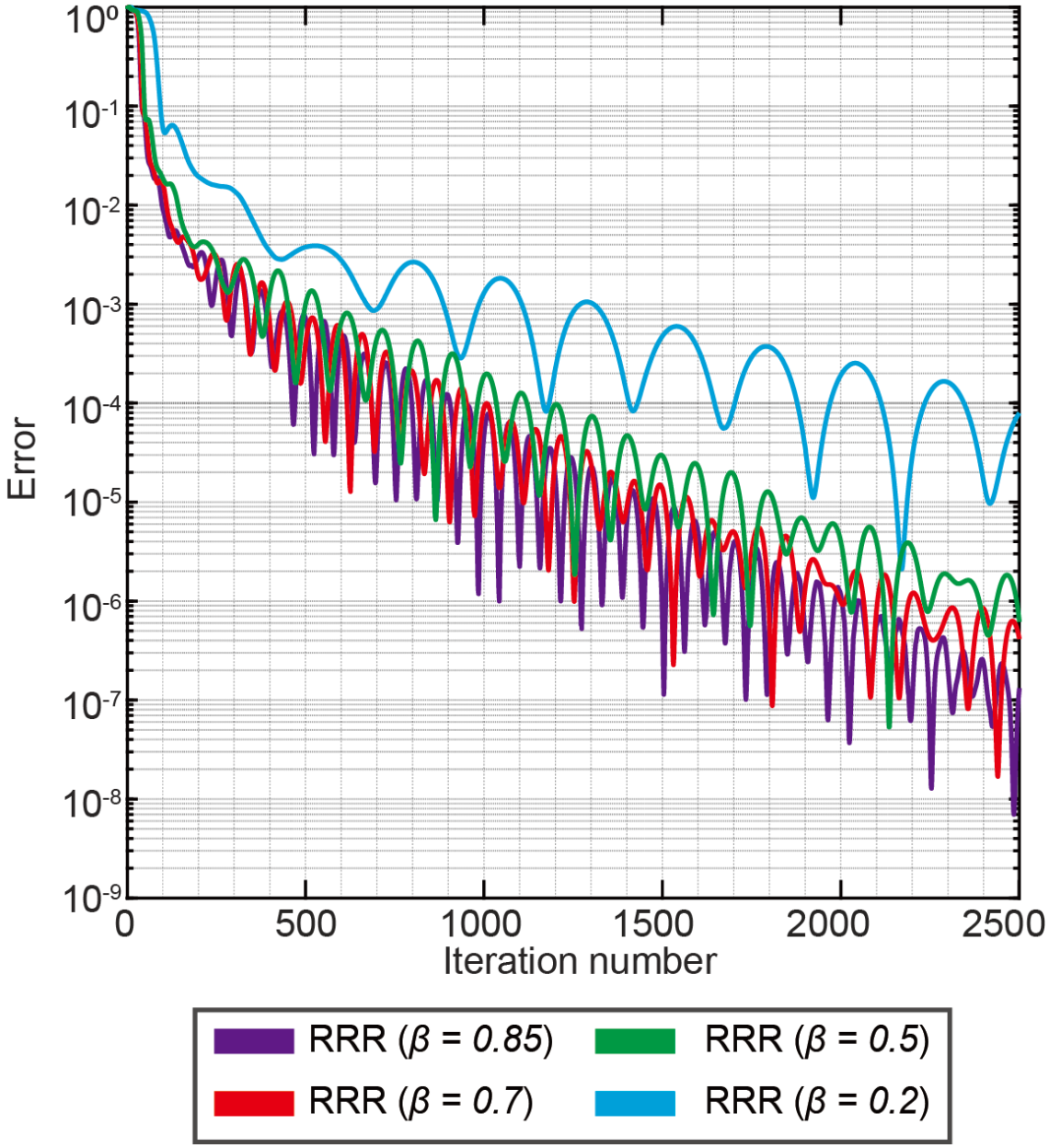

Figure 5.21. The simulation results for RRR with different β.

Since the reconstruction is visually indistinguishable from the ground truth, if the simulation error is lower than 10^{-5}, here, we only display its error curve to indicate its performance. From Figure 5.21, the error of RRR keeps jiggling up and down, a smaller β seems can relax the frequency of this oscillation, but it reduces the convergence rate. Therefore, we will choose $\beta = 0.85$ for our further tests.

The second test is for RAAR, the results are displayed in Figure 5.22.

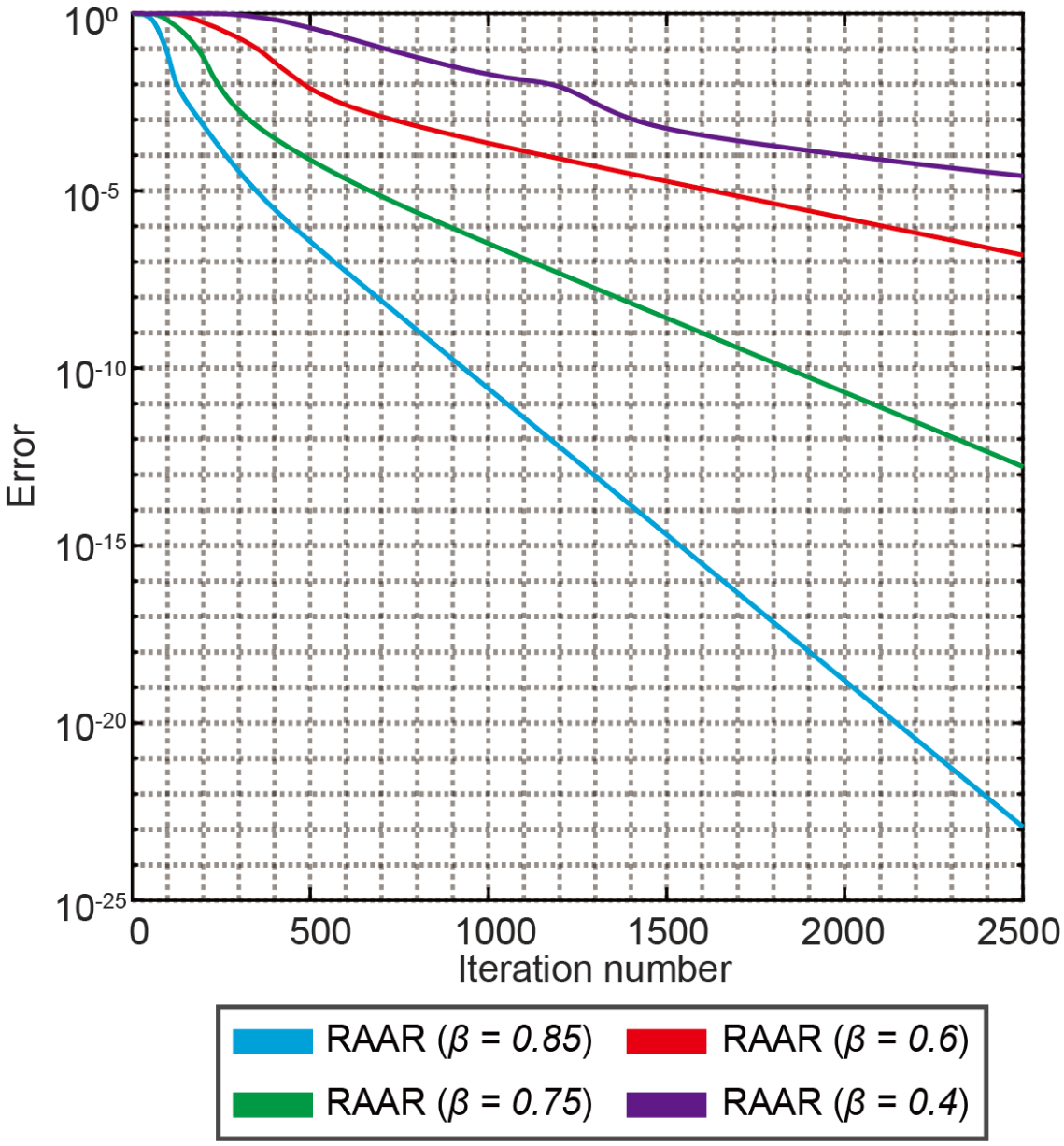

Figure 5.22. The simulation results for RAAR with different β.

Figure 5.22 indicates that smaller β significantly reduces the convergence rate of RAAR. However, similar to RRR, an over relaxed β does not work in our test. Hence, we use $\beta = 0.85$ for RAAR.

Similarly, the test results for T_λ is illustrated in Figure 5.23.

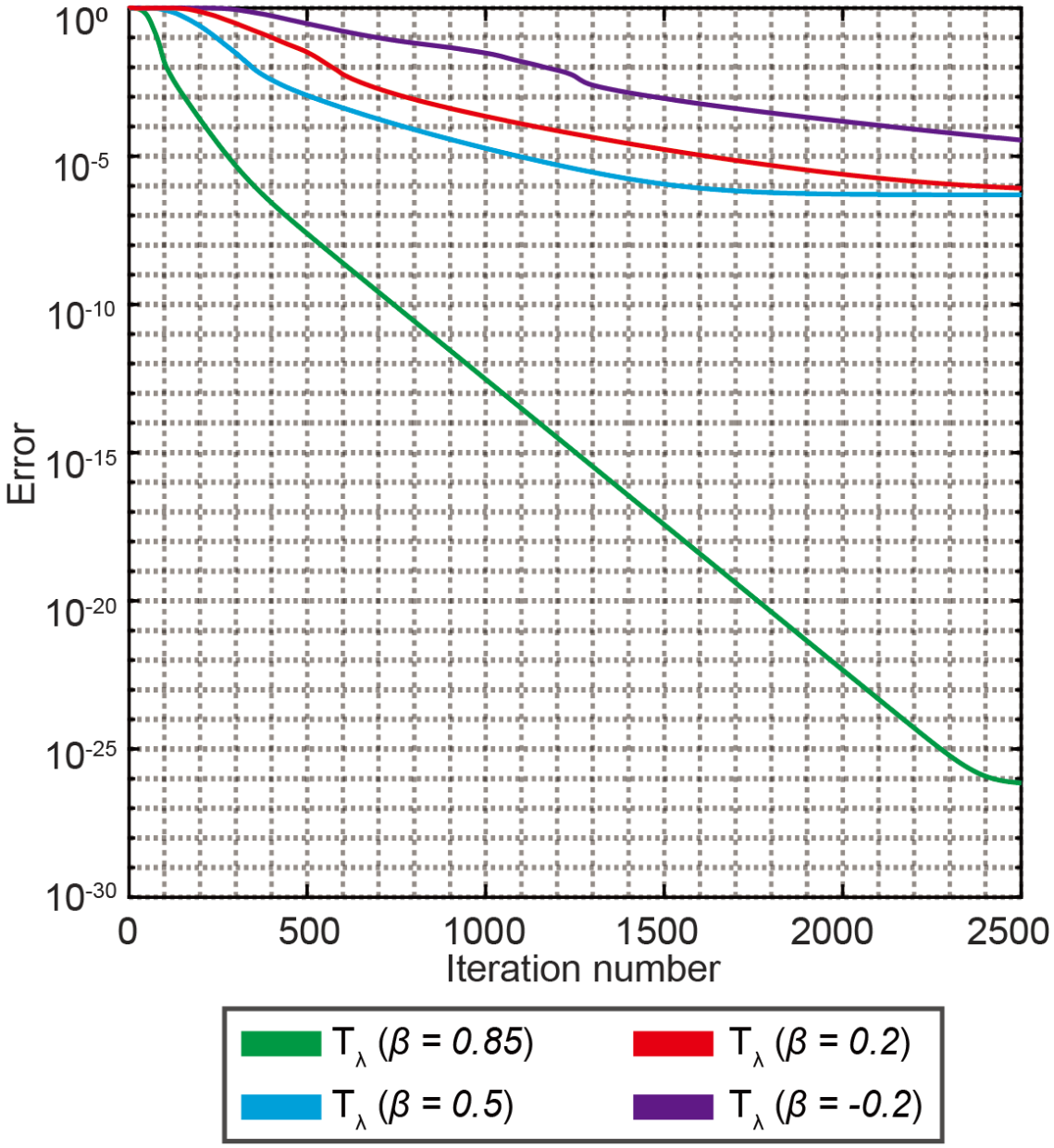

Figure 5.23. The simulation results for T_λ with different β.

T_λ is slightly different from RRR and RAAR since it introduces the relaxations to both b and c, then another relaxation in the opposite way for a. As mentioned before, for the three-circle problem in Figure 5.13 (f), a small β will cause the difficulty to escape from the local minima, which seems to happen here for $\beta = 0.5$, $\beta = 0.2$ and $\beta = -0.2$. Therefore, we will use $\beta = 0.85$ for T_λ in further tests.

5.5.3. Noiseless Simulation Results

In our simulation tests, the first simulation data is perfect data without any noise. All the ambiguities are removed during the reconstruction since the true object and probe are known in the simulation. Hence, the simulation error could be very small, approaching to the minimum accuracy of the computer.

This first result show in Figure 5.24 is using the big size probe.

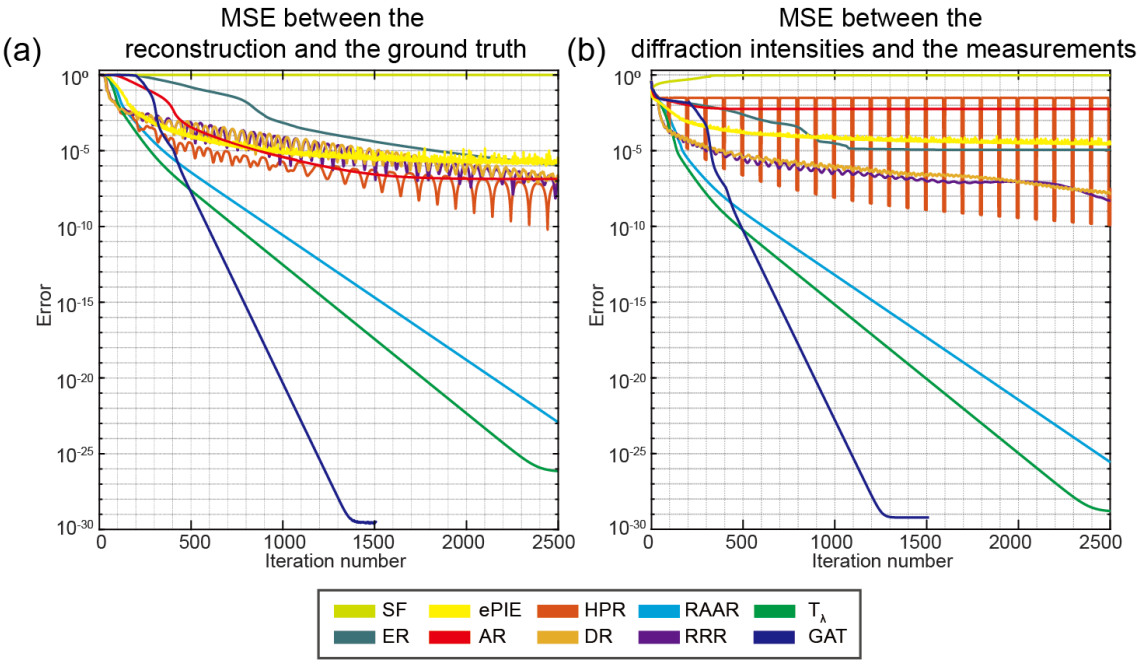

Figure 5.24. The error of the simulation with large size probe (512×512 pixels, 400 diffraction patterns), (a) The MSE between the reconstruction and the ground truth, (b) The MSE between the reconstructed diffraction patterns and the measurements.

For the generalized auto-tuning (GAT) algorithm in this test, the values of a, b, c are displayed in Figure 5.25.

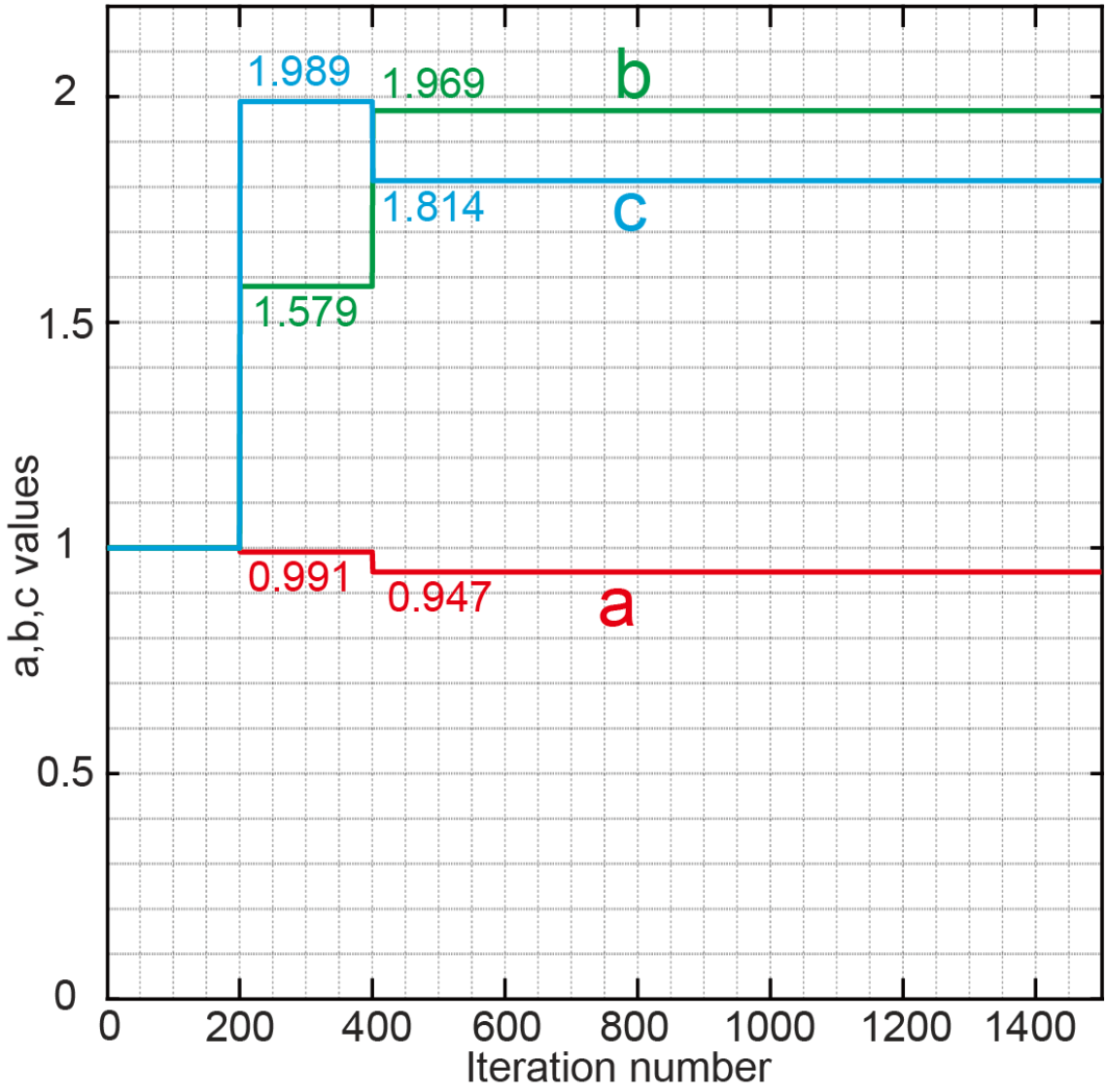

Figure 5.25. The values of a, b, c for Generalized Auto-Tuning (GAT) algorithm in the large size probe simulation.

As shown in Figure 5.24, the error metric demonstrates that T_λ is slightly better than RAAR, and both of them have significantly better performance than other set projection methods except GAT. ePIE a kind of sequential projection algorithm, here, it is used as a reference because of its popularity in ptychography. GAT performs very well in this test; it converges at the smallest error within only around half of the number of iterations that T_λ used. From Figure 5.25, although the auto-tuning is conducted every 100 iterations, GAT only changes the values of a, b, c twice, the final tuning result has a similar b and c, but increased the value of a, compared to RAAR and T_λ. This small

change significantly improves the convergence rate. The phase reconstruction of object and probe is displayed in Figure 5.26, the results from GAT, RRR and T_λ is picked up to show since they can represent the reconstruction at different error levels. Because the error is very low, the difference between the reconstructions and the ground truth is not distinguishable to the eye.

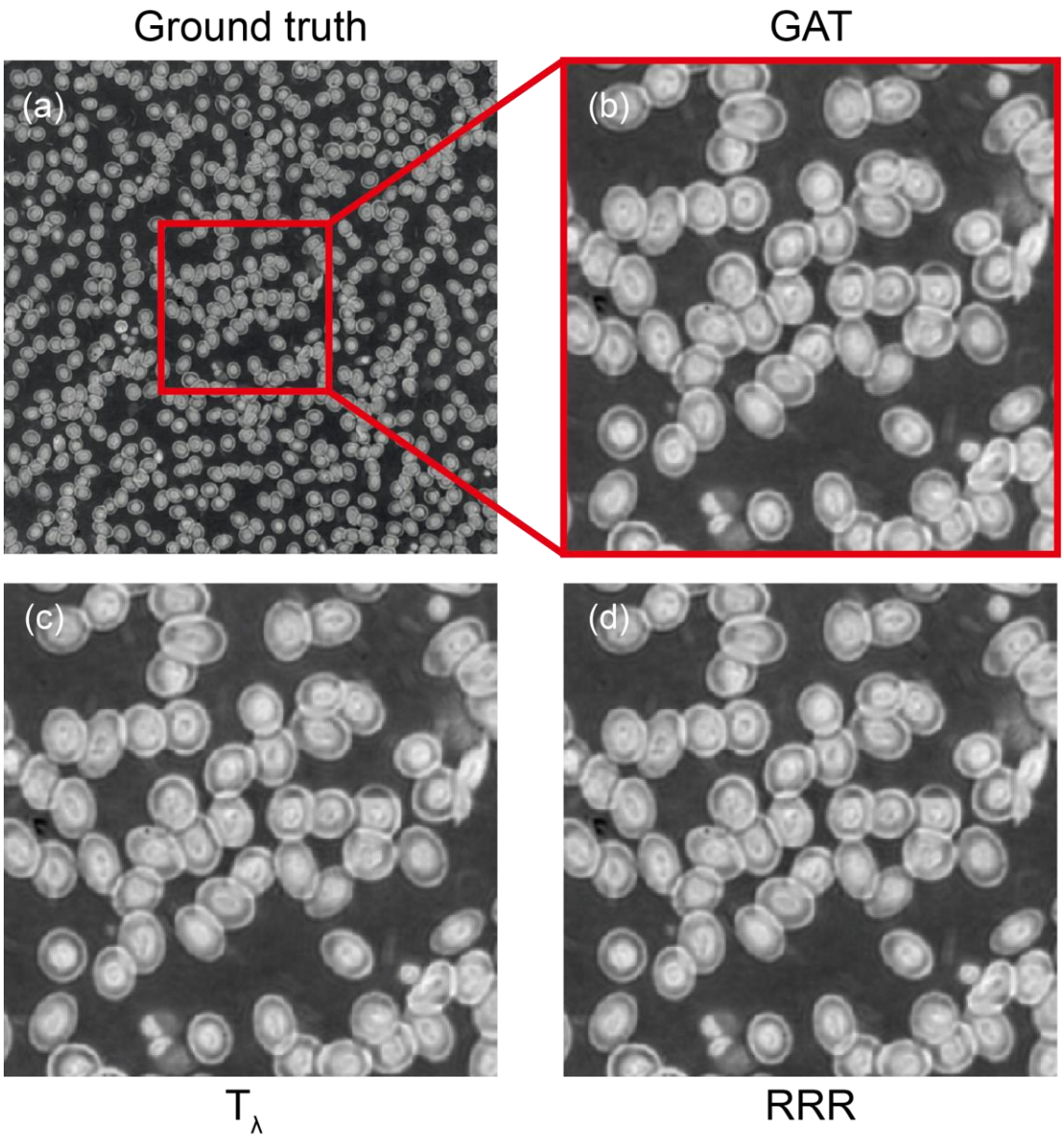

Figure 5.26. The phase reconstructions from GAT, T_λ and RRR. The red box indicates the zoomed area.

Another result for small size probe simulation is shown in Figure 5.27, the

performance is similar to the large size one. More positions and denser data make the reconstruction more difficult. GAT is still the best one in Figure 5.27, and its advantage is magnified. Apart from GAT, all the other algorithms cannot reach the minimum error level this time. The error of T_λ, RAAR and HPR is below 10^{-5}, slightly better than others. SF failed in both large and small size simulation.

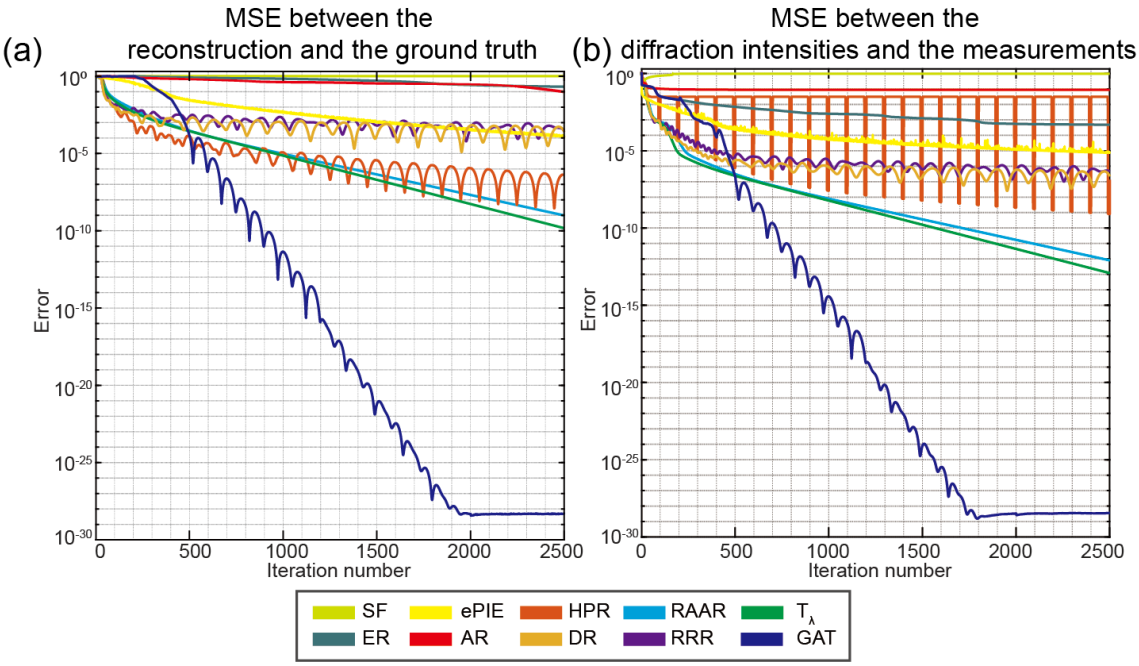

Figure 5.27. The error of the simulation with small size probe (128×128 pixels, 6400 diffraction patterns), (a) The MSE between the reconstruction and the ground truth, (b) The MSE between the reconstructed diffraction patterns and the measurements.

Figure 5.28 indicates the values of a, b, c for GAT during the reconstruction. There are many tuning attempts this time, compared to the simulation with large size probe. The spikes in Figure 5.28 illustrates the unsuccessful tunings, a, b, c are reset to the pervious values after only few iterations. It is also clear to see the rise of the error from the Figure 5.27. In general, the addition insurance prevented some unnecessary changes, increased the stability of the auto-tuning.

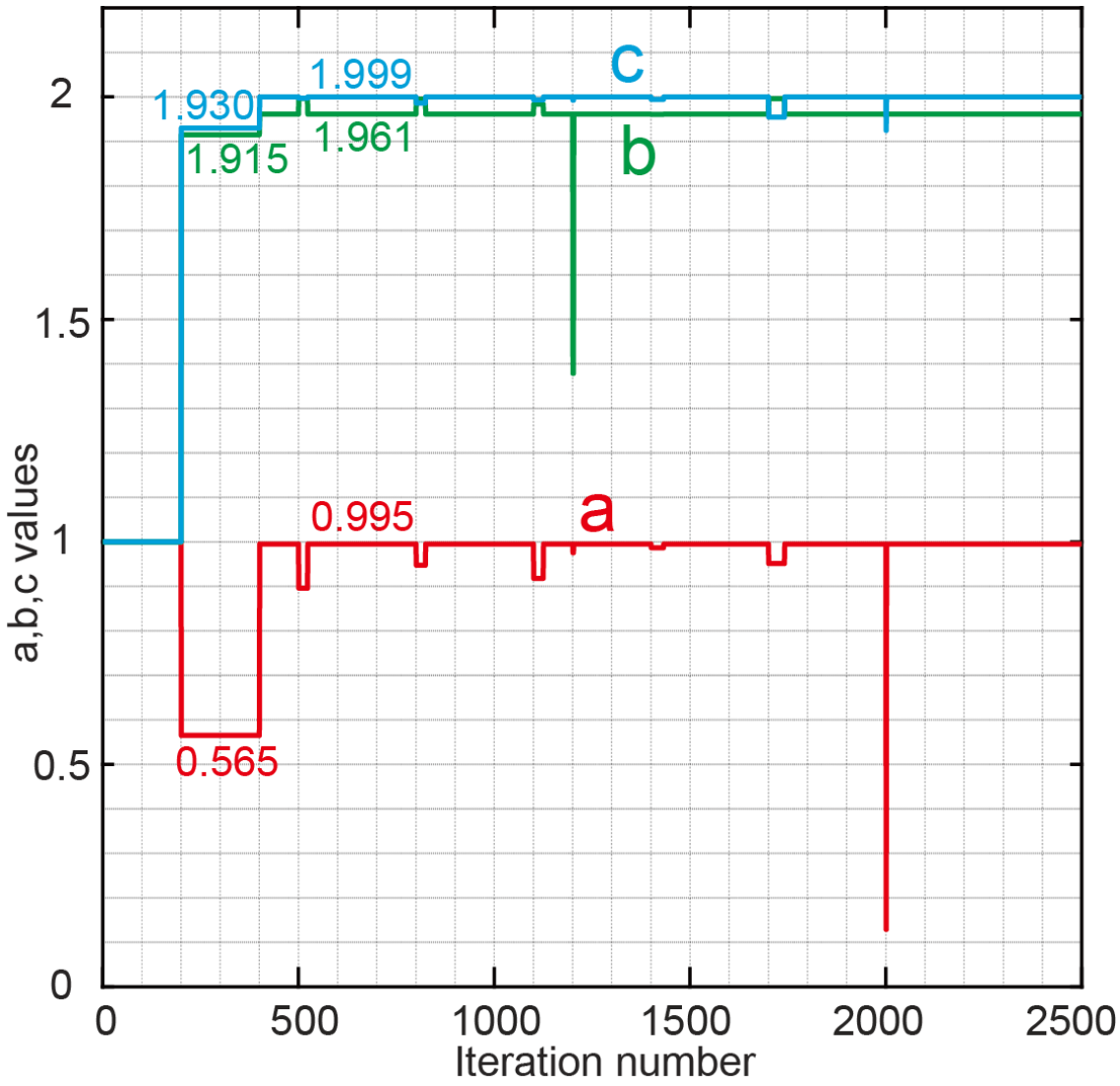

Figure 5.28. The values of a, b, c for Generalized Auto-Tuning (GAT) algorithm in the small size probe simulation.

5.5.4. Noise Simulation Results

The noise test is adding the Poisson distributed noise to the diffraction pattern. The first result in Figure 5.29 has 5×10^5 counts in each diffraction pattern. GAT, AR and ER performs better than other algorithms in the aspect of noise tolerant. T_λ and RAAR are still quite similar and better than the rest as a relaxed version of DR. ePIE works well at the beginning, but it is not stable and suddenly collapsed.

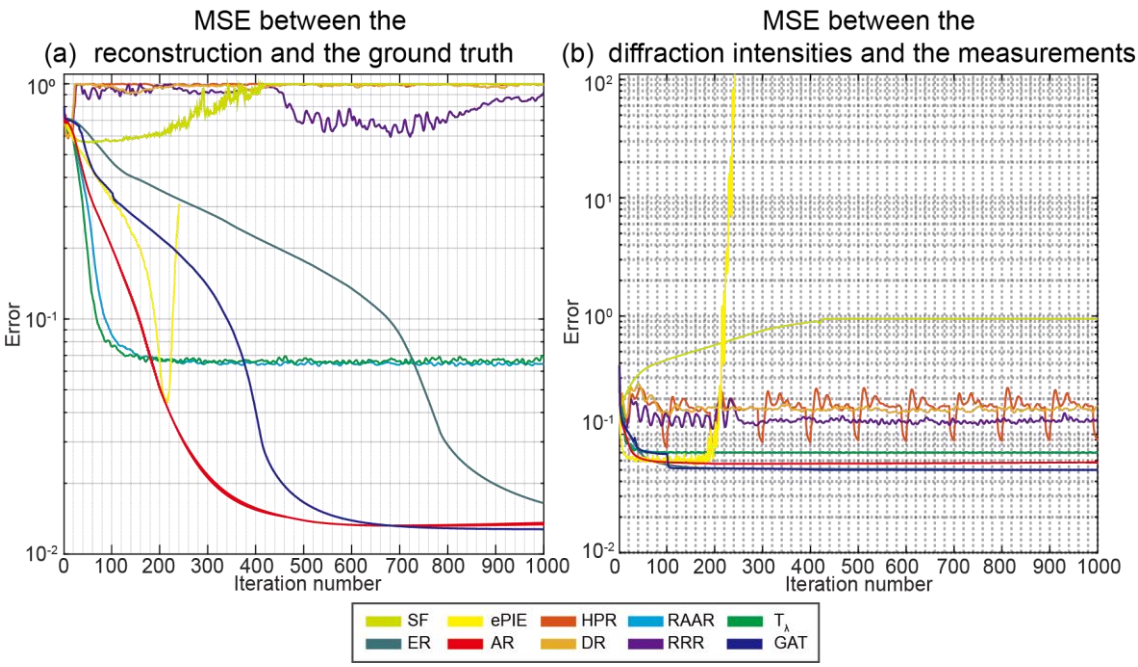

Figure 5.29. Noise simulation results with 5×10^5 counts in each diffraction pattern. (a) The MSE between the reconstruction and the ground truth, (b) The MSE between the reconstructed diffraction patterns and the measurements.

The values of a, b, c are shown in Figure 5.30. There are twice successfully tuning. finally gives similar values as AR.

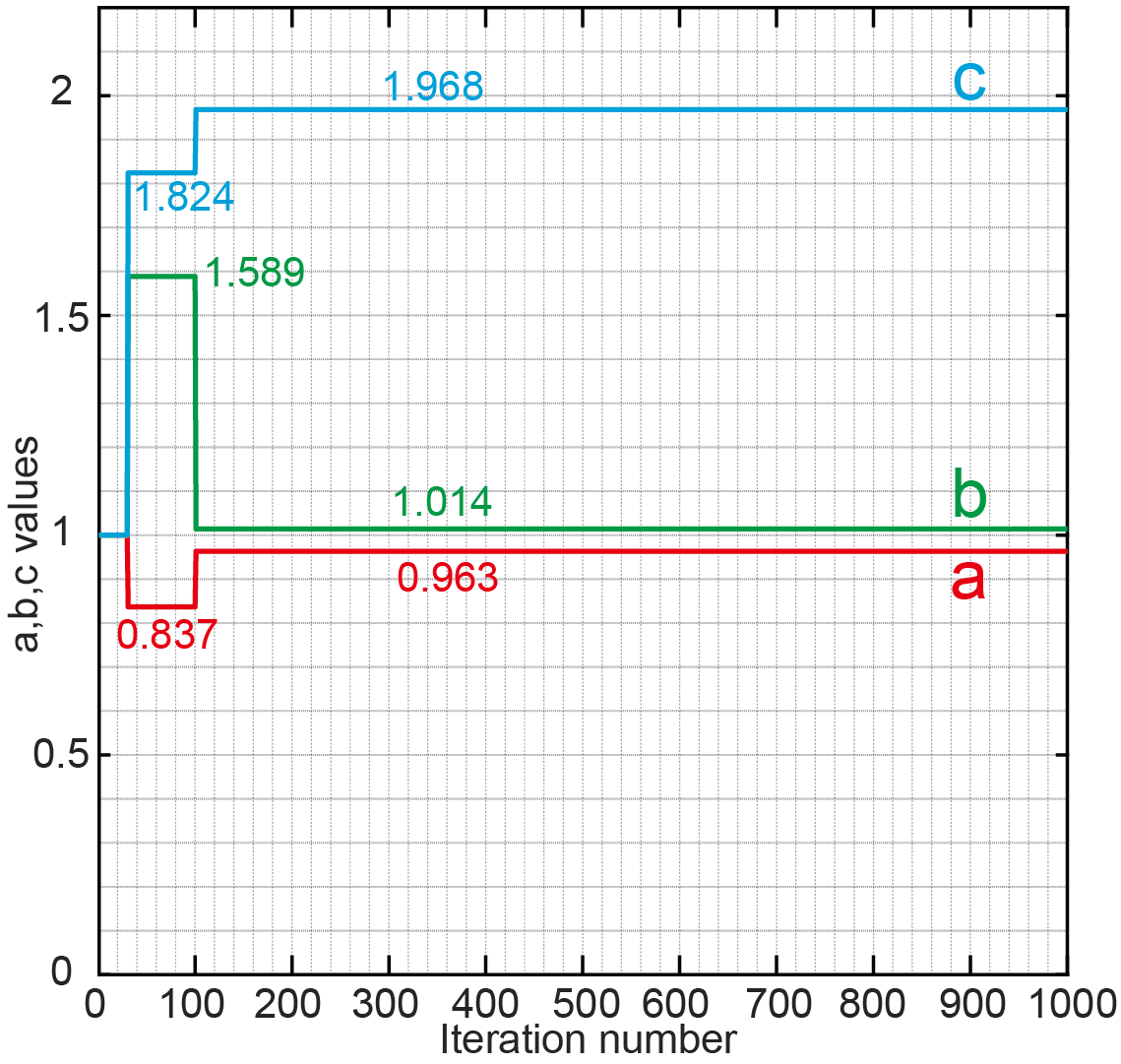

Figure 5.30. The values of a, b, c for Generalized Auto-Tuning (GAT) algorithm in the noise simulation with 5×10^5 counts in each diffraction pattern.

Some reconstructed objects are illustrated in Figure 5.31; a zoomed area in red box is selected to show more details. The reconstructions of GAT, AR and ER are difficult to be distinguished with the naked eye, they have the best resolution and almost to see all the features. The resolution of T_λ and RAAR is slightly worse, fails to reconstruct the fine details in the center. In contrast, ePIE only works well at the central part. The results of the remaining algorithms are unsatisfactory.

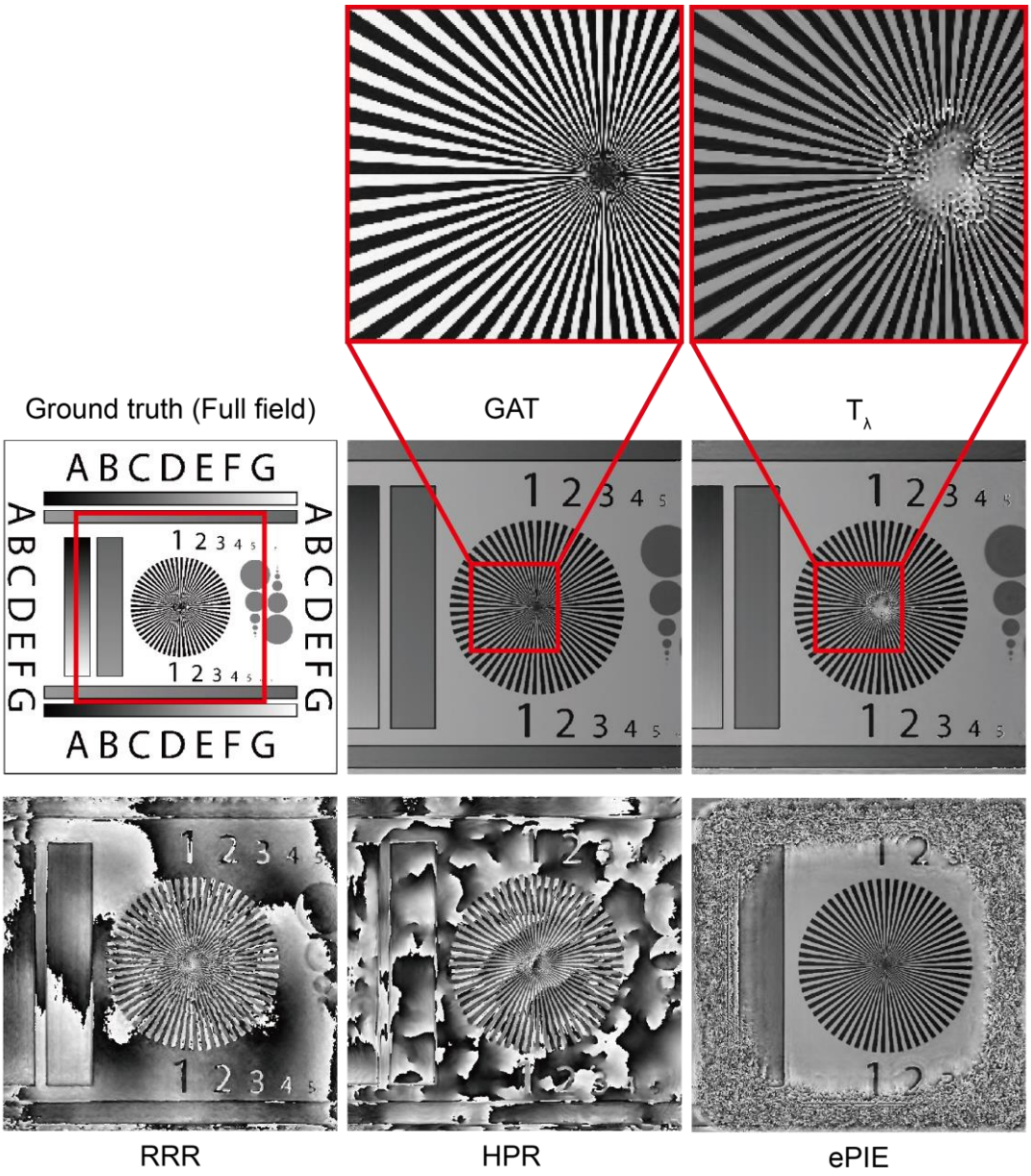

Figure 5.31. The phase reconstruction from GAT, T_λ, RRR, HPR and ePIE for the noise simulation with 5×10^5 counts in each diffraction pattern.

The second test only has 10^4 counts in each diffraction pattern. This can be considered as a very noisy situation in ptychography. The error lines are shown in Figure 5.32.

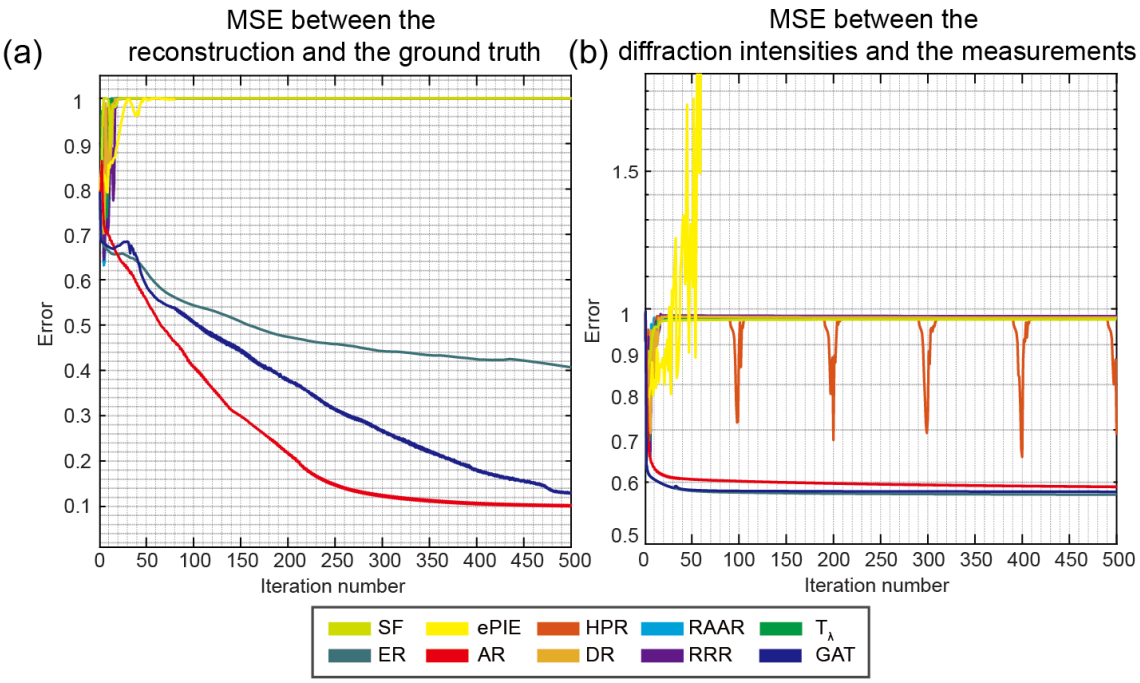

Figure 5.32. Noise simulation results with only 10^4 counts in each diffraction pattern. (a) The MSE between the reconstruction and the ground truth, (b) The MSE between the reconstructed diffraction patterns and the measurements.

In this case, only AR, GAT and ER worked in the test; the rest of the others all failed. The a, b, c values in GAT are shown in Figure 5.33. Here, the auto-tuning was executed every 30 iterations, but only the first time was successful. This single tuning gives the values closer to AR, and improves a lot compared to ER, which is the initial condition of GAT.

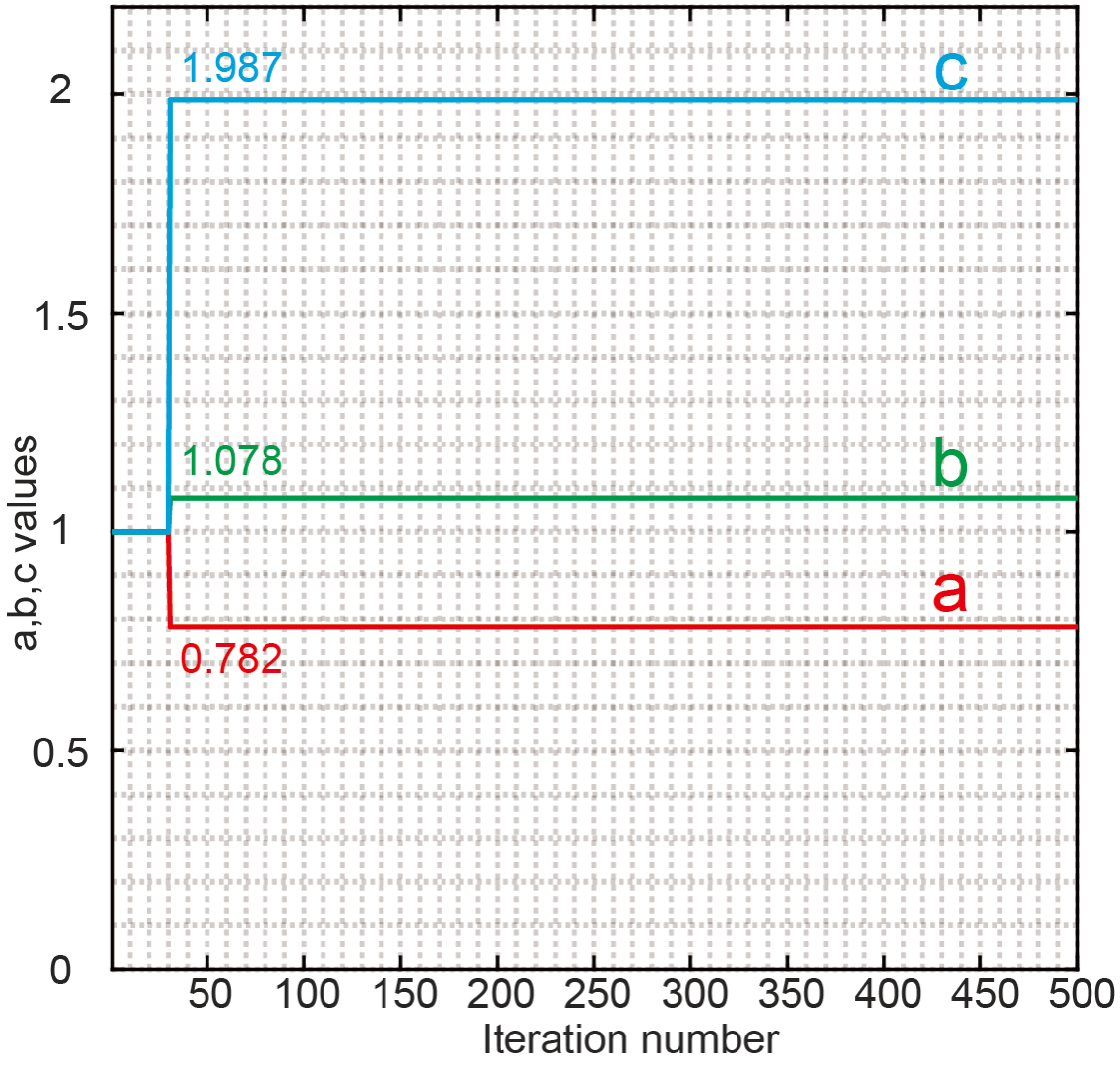

Figure 5.33. The values of a, b, c for Generalized Auto-Tuning (GAT) algorithm in the noise simulation with only 10^4 counts in each diffraction pattern.

The reconstructed objects are illustrated in Figure 5.34. T_λ, as well as other failed set projection methods, their reconstructions are entirely noise. ePIE only reconstructs a small fraction feature at the center, and the feature is not very clear.

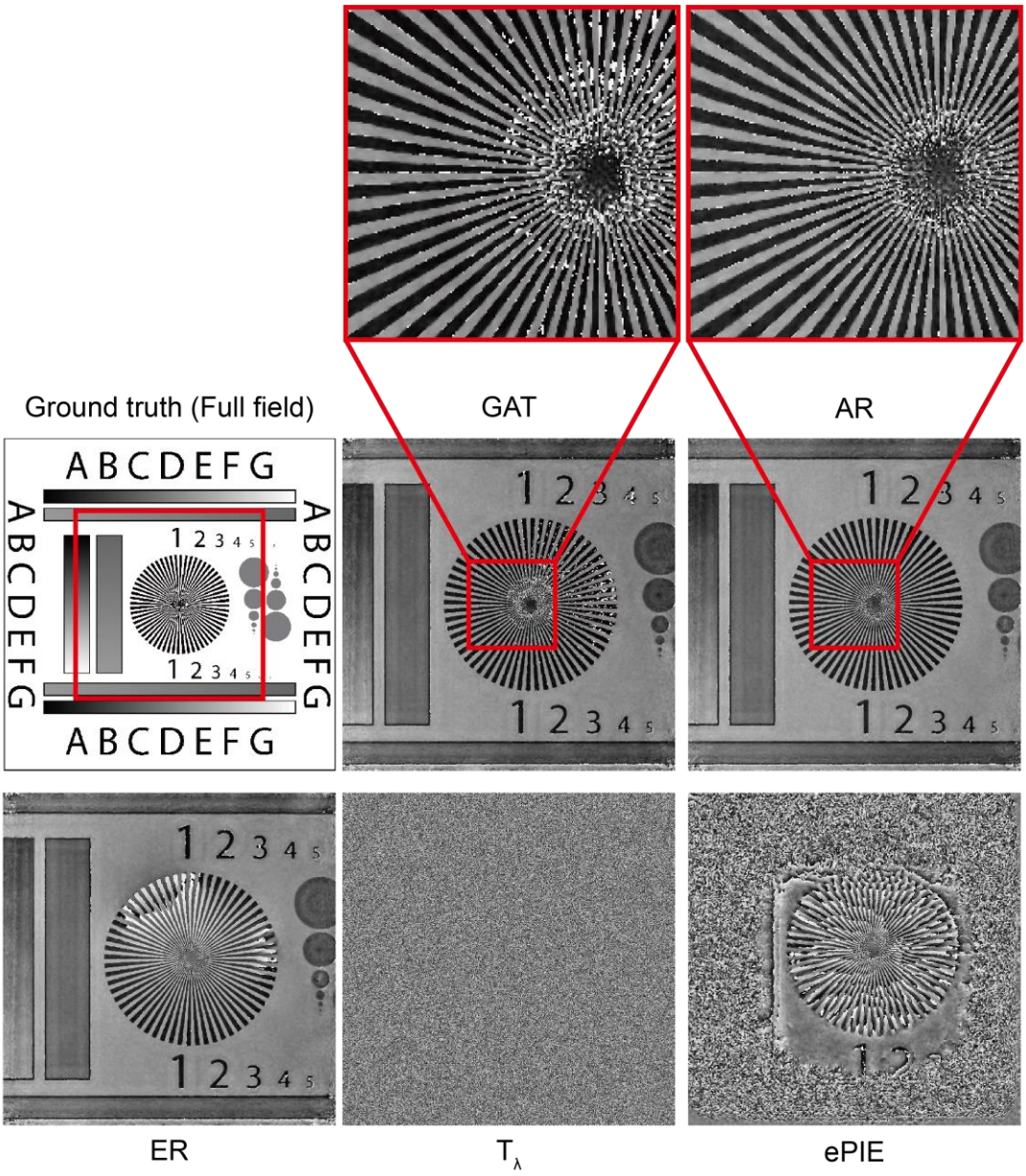

Figure 5.34. The phase reconstruction from GAT, AR, ER, T_λ, and ePIE for the noise simulation with only 10^4 counts in each diffraction pattern.

5.6. Conclusion

In our simulation tests, RAAR and T_λ are similar. As a relaxed version, they have a great improvement compared to the DR or DM. Also, their performance is generally good except for the very noisy data test. By contrast, RRR, as another type of relaxed DR, does not make a significant progress in the ptychography. AR performs much better than other algorithms in the aspect of

noise tolerance for both tests at different noise levels. ER has a stable performance in all the simulations since its error was proved to be reduced continuously [31], however, its convergence rate is not guaranteed. GAT has the best performance in the noiseless simulation, and also does well in the noise tests. As it turns out, only a few times of successful tuning can greatly improve the quality of reconstruction. This method also has good adaptivity to different datasets. However, because the optimization is error curve based, the time and memory cost of objective function during the optimization could be very high, depending on the dataset size. In the future, it is worth to investigate the practical effects of these three parameters on each projection process to find out a more efficient and accurate way to assess the influence of these three parameters. Also, the optimization based on the error curve is not significant in the real experiment since the unambiguous error is not available. An alternative is to use other methods to evaluation the quality of the reconstruction, such like Fourier Ring Correlation (FRC) [60, 61]. The improvement on the objective function for GAT may further enhance its effectiveness.

6. Weighted Average of Sequential Projections (WASP) for Ptychographic Phase Retrieval

In the previous chapter, most existing set projection algorithms have been discussed. ePIE as an example of a sequential projection algorithm was introduced in Chapter 3.2.7. In this chapter, a novel ptychographic approach called Weighted Average of Sequential Projections (WASP) will be presented. Furthermore, a parallel version of WASP will be demonstrated that splits operation across several different computation nodes. Tests and results about WASP and its competitors will be shown at the end of the chapter.

6.1. Definition of Weighted Average of Sequential Projections (WASP)

This section will explain the definition of WASP and illustrate it through the three-circle example as shown in the previous chapter.

6.1.1. Sequential Projections (SP) & Divide and Concur (DC)

The principle behind WASP is to combine sequential projections and the idea of divide and concur (DC). The concepts of both have already been discussed in the last chapter. WASP uses the outputs of sequential projection as the divide part, then averages all the results into one single point by a concur step. An illustration of the implementation of WASP in the three-circles problem is shown in Figure 6.1.

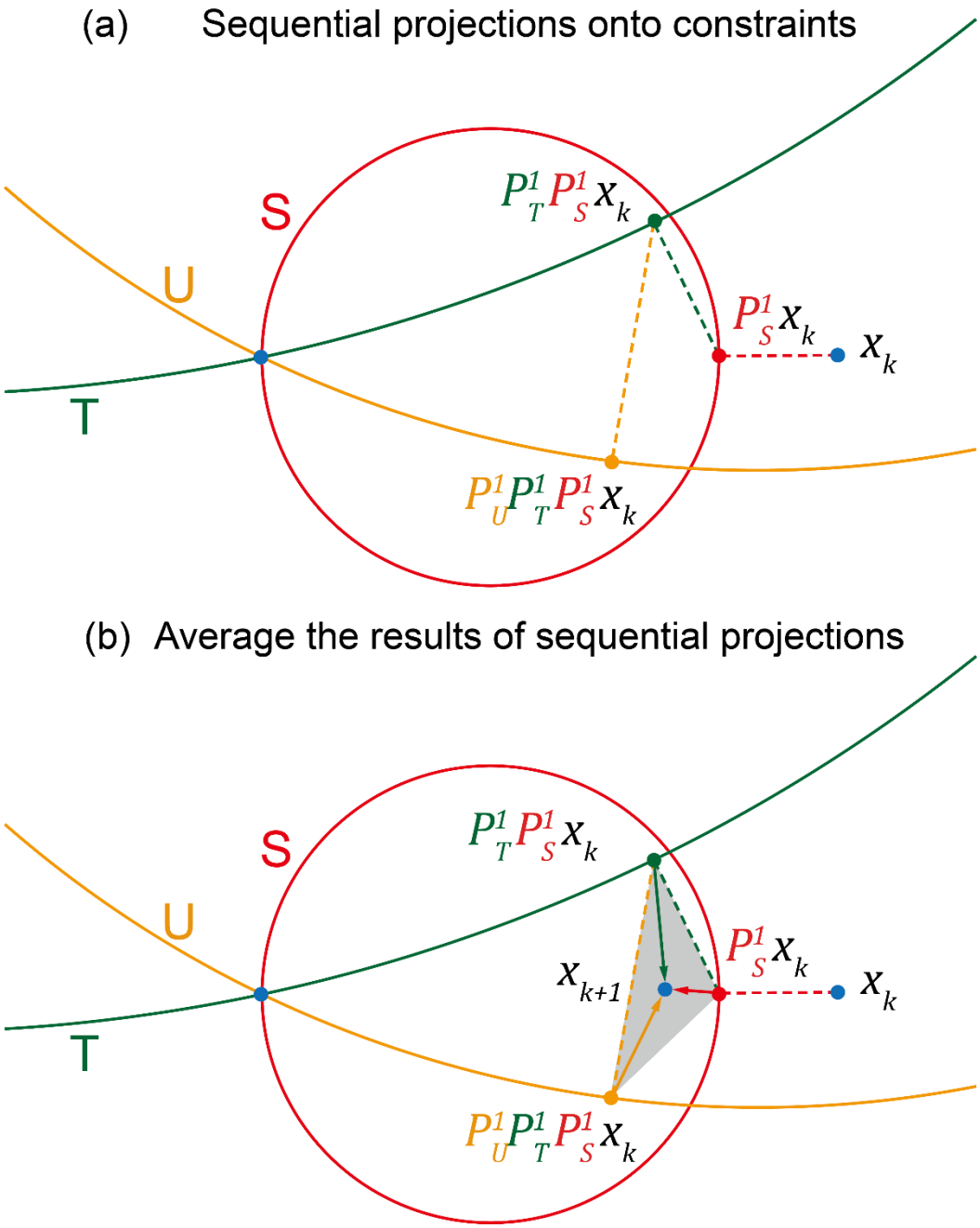

Figure 6.1. The schematic of WASP approach. (a) The divide step in WASP is to make sequential projections onto constraints, record all the results on each constraint (b) The concur step in WASP, the new points from sequential projections will be averaged to one single solution.

The order of sequential projections in Figure 6.1 (a) is not critical, also a relaxation can be introduced into this step. x_{k+1} in Figure 6.1 (b) will be the start point for the next iteration. Therefore, the prior information from each constraint will be better utilized during the sequence. For the example of the three-circle problem, WASP and the set projection methods both take the average of three

result points, but each result point in set projection method is one single projection for one constraint. By contrast, WASP uses sequential projections, therefore, the second point has the prior knowledge from the first one, and the third point has the knowledge from previous two. That results in the global average of sequential projection gives a more weighted and appropriate response to all the constraints. The divide and concur in WASP can be written as Equation (6.1) and (6.2):

$$P_D^{\mathrm{a}} x_k = [P_S^{\mathrm{a}} x_k, P_T^{\mathrm{a}} P_S^{\mathrm{a}} x_k, P_U^{\mathrm{a}} P_T^{\mathrm{a}} P_S^{\mathrm{a}} x_k] \tag{6.1}$$

$$P_C^{\mathrm{b}} x_k = \frac{\mathrm{b}}{3} (P_S^{\mathrm{a}} x_k + P_T^{\mathrm{a}} P_S^{\mathrm{a}} x_k + P_U^{\mathrm{a}} P_T^{\mathrm{a}} P_S^{\mathrm{a}} x_k) \tag{6.2}$$

where a and b are the relaxation degree to adjust the performance.

Unlike set projection algorithms, WASP does not require to store the product spaces which contain the results from the projections. Only the averaged point will be recorded for the calculation in the next iteration. This is the memory efficiency of WASP in comparison to set projection algorithms.

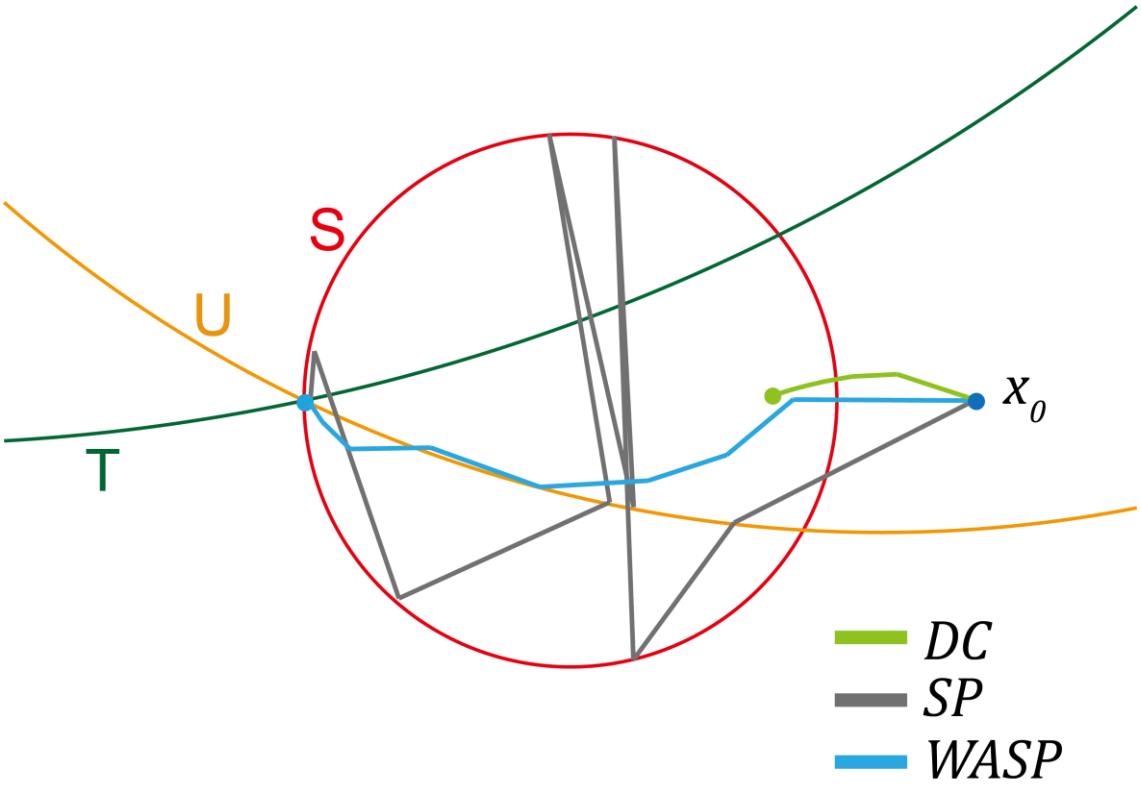

Figure 6.2. The results of DC, SP and WASP for the three-circle problem.

Figure 6.2 shows the results of the three-circle problem with DC, SP and WASP solutions. As it illustrated, DC was trapped at some local minima. In this simple example, SP reached the final solution since there are only three constraints here. However, in the case of ptychography, there are usually hundreds or thousands of modulus constraints, which will cause the instability of SP, especially when the data is noisy. By introducing global optimization in SP, WASP can generally avoid the local minima problem and get to the solution more smoothly.

6.1.2. Application in Ptychography

Ptychography was described in detail in a pervious chapter. It can be generally considered as an alternating problem between ‘consistency projection’ and ‘modulus projection’. To solve the object and probe, the ultimate goal is making the reconstructions satisfy both of these constraints. The minimization of the cost functions and the updating of the object and probe were explained in

section 5.4.3 and 5.4.4. For the sequential projection methods, the updating functions for the object and probe was shown in Equation (5.29) and (5.30), here we refer them again as shown below:

$$o_{k_j}'(\vec{r}) = o_{k_j}(\vec{r}) + \frac{P_j^*(\vec{r})\left(\psi_{k_j}'(\vec{r}) - \psi_{k_j}(\vec{r})\right)}{|P_j(\vec{r})|^2 + A} \tag{5.29}$$

$$P_j'(\vec{r}) = P_j(\vec{r}) + \frac{o_{k_j}^*(\vec{r})\left(\psi_{k_j}'(\vec{r}) - \psi_{k_j}(\vec{r})\right)}{|o_{k_j}(\vec{r})|^2 + B} \tag{5.30}$$

where A and B are the regularization functions. The regularization functions for PIE style methods were given in Equation (5.25), (5.26), (5.27) and (5.28). Generally, both regularizers for ePIE and rPIE are working well in the WASP. Here, we proposed a new regularizer for WASP, defined as Equation (6.3) and (6.4):

$$A = \alpha \langle |P_j(\vec{r})|^2 \rangle \tag{6.3}$$

$$B = \beta \tag{6.4}$$

where $\langle |P_j(\vec{r})|^2 \rangle$ represents the average over all elements of matrix $|P_j(\vec{r})|^2$. Compared to ePIE and rPIE, the average operation is computationally cheaper than the maximum operation, also, the average of probe intensity remains fairly constant, whereas the maximum can be unstable and affect by some extremes. Similarly, the regularizer for probe is just a constant value since the modulus of object is between 0 and 1. More details and comparison between different regularizers for WASP will be discussed in the later section.

In the sequential projection methods, Equation (5.29) only updates the object at position k, which is fraction of the entire object. However, for the set projection methods, the entire object can be updated by Equation (5.33) and

(5.34), referred here:

$$O'_j(\vec{r}) = \sum_k \frac{\psi'_{kj}(\vec{r})}{P_j(\vec{r})} \tag{5.33}$$

$$P'_j(\vec{r}) = \sum_k \frac{\psi'_{kj}(\vec{r})}{o_{kj}(\vec{r})} \tag{5.34}$$

In this case, the updating of the object is the sum of the exit waves over all the positions and dividing by the sum of the probe. The bright areas of the probe will contribute more to the sum than dark areas. Similarly, the transmissive part of the object also contributes more to the probe updating.

The idea of WASP is to combine these two different ways together. Firstly, step through all the scan positions like sequential projection methods, during this, record the sums that needed for Equation (5.33) and (5.34). Finally, when all the positions are solved, use Equation (5.33) and (5.34) for a global updating of the object and probe. This global average step is the “Weighted Average (WA)” part of WASP while the first one is the “Sequential Projections (SP)” part. The flow chart of WASP is illustrated in Figure 6.3.

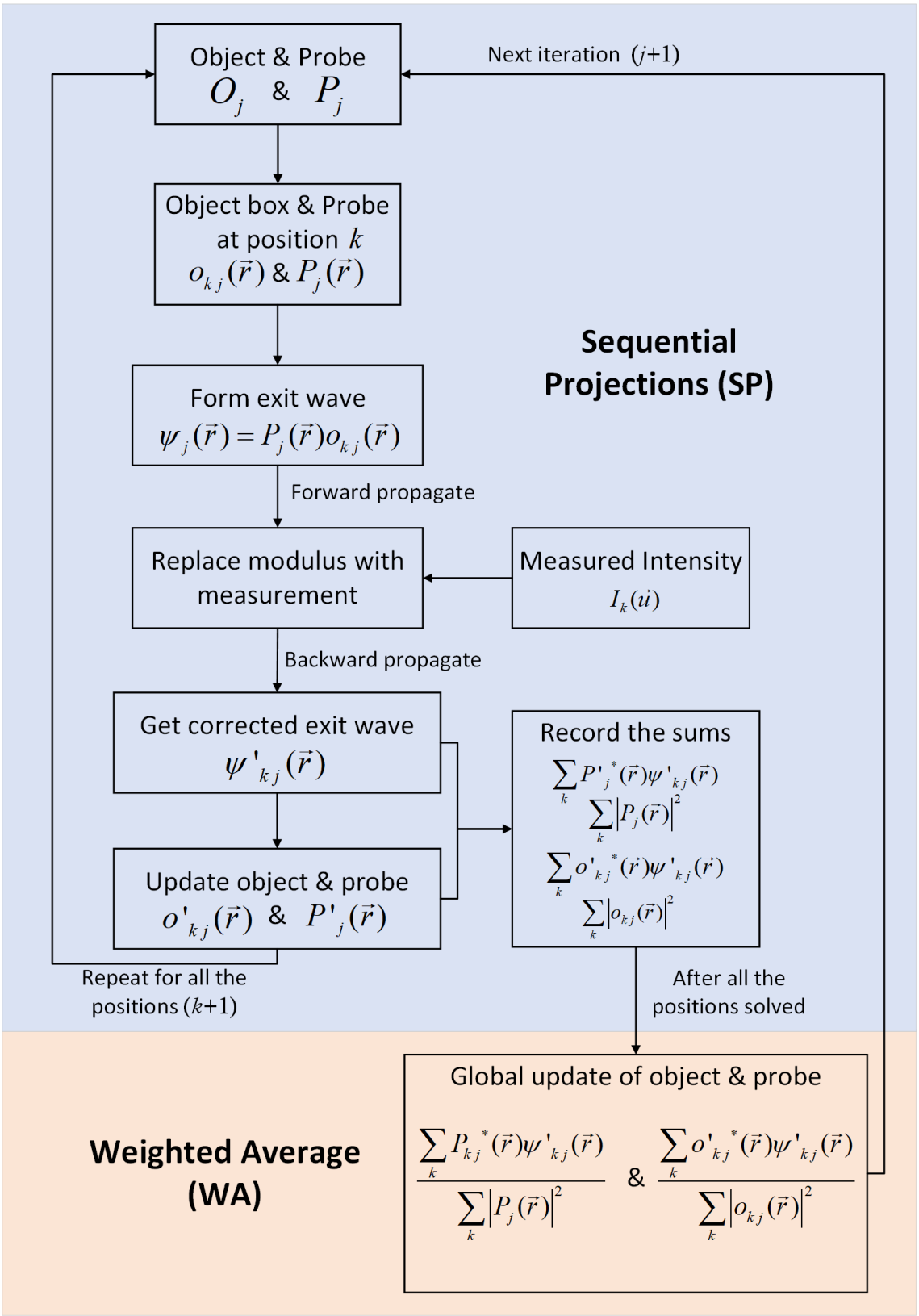

Figure 6.3. The flow chart of WASP, the blue area is the SP part, whereas the orange area represents the WA part.

The iteration starts from SP part, step out all the positions, then do the weighted average (WA), the outcome of WA will be input to the next iteration. The pseudocode of WASP is shown in **Pseudocode 6.1**.

Pseudocode 6.1: Weighted Average of Sequential Projections (WASP)

Inputs*: position vectors* (R), *object* (obj), *object size* (X, Y), *probe function* (probe), *probe size* (M, N), *intensity* (I), *the total number of positions* (K), *the total number of iterations* (J).

Outputs*: reconstructed object* (obj) *and probe* (probe)

```
For (j = 1 to J) do
  // Initialise numerator and denominator sums
  topO = bottomO = zeros(X,Y)
  topP = bottomP = zeros(M,N)
  R = shuffle(R)
  For (k = 1 to K) do
    // Form exit waves and apply the modulus constraint
    objBox = obj(Rk to Rk+[M,N])
    exitWavek = objBox·probe
    detectorWavek = ℱ(exitWavek)
    correctedWavek = sqrt(Ik)·detectorWavek /(abs(detectorWavek) + eps)
    newExitWavek = ℱ⁻¹(correctedWavek)
    ΔexitWavej = newExitWavek-exitWavek
    // Sequential projection update of object and probe
    obj(Rk to Rk+[M,N]) += conj(probe)·ΔexitWavej /abs(probe)² + A
    probe += conj(objBox)·ΔexitWavej /abs(objBox)² + B
    // Update numerator and denominator sums
    topO += conj(probe)·newExitWavek
    bottomO += abs(probe)²
    topP += conj(objBox)·newExitWavek
    topO += abs(objBox)²
  End loop
  // Weighted average update of object and probe
  obj = topO/(bottomO + eps)
  probe = topP/(bottomP + eps)
  Apply any additional constraints
End loop
```

Note: zeros: a matrix full of zeros. **shuffle***: a function that randomly change the order of the position sequence*. $\mathcal{F}$: Fourier transform. $\mathcal{F}^{-1}$: inverse Fourier transform. ***eps:*** a small constant in MATLAB to avoid dividing 0. **sqrt**: square root. **abs**: amplitude, **conj**: complex conjugate.

6.1.3. Memory Footprint

Compared to set projection algorithms, WASP does not require to store all the projection results in the SP part. This allows WASP to run with less memory. Assume there is an object with size $[X, Y]$ and a probe with size $[M, N]$, the total scan positions is K. The memory cost for the sequential projection algorithms, set projection algorithms and WASP is described in Table 3.

Table 3. The basic memory requirements for different algorithms

Algorithm	Number of stored pixels
Sequential Projection Algorithms (ePIE)	$2XY + 2MN + KMN$
Set Projection Algorithms (RAAR, DM, ER)	$2XY + 2MN + 3KMN$
WASP	$5XY + 5MN + KMN$

Normally, the largest part from the table is KMN which represents K diffraction patterns. Set Projection Algorithms need to store the product spaces that results in $3KMN$. This is significantly increasing the memory requirements. By contrast, WASP records the sums during the iteration, this part only needs a small memory space. For the example of the simulation with small size probe in Chapter 5.5.1, $[M, N] = [128, 128]$, $[X, Y] = [1200, 1200]$ and there are 6400 diffraction patterns in total. Therefore, the number of stored pixels can be calculated for different types of algorithms, if we consider all of them are double precision, which is 8 bytes per pixel, see Equation (6.5):

$$
\begin{aligned}
2XY + 2MN + KMN &= (2.88 + 0.03 + 104.86) \times 10^6 \\
&= 107.78 \times 10^6 \text{ pixels} \approx 862.16 \text{ MB} \\
2XY + 2MN + 3KMN &= (2.88 + 0.03 + 314.58) \times 10^6 \\
&= 317.49 \times 10^6 \text{ pixels} \approx 2540 \text{ MB} \\
5XY + 5MN + KMN &= (7.20 + 0.08 + 104.86) \times 10^6 \\
&= 112.14 \times 10^6 \text{ pixels} \approx 897.12 \text{ MB}
\end{aligned} \tag{6.5}
$$

Generally, WASP takes the advantage of the set projection methods but only with a small memory requirement.

6.2. Simulation Results of WASP

In this section, the same simulation tests carried out for the set projection algorithms is now applied for WASP. The simulation configuration was described in detail in Chapter 5.5.1.

6.2.1. Different Regularizers for WASP

As mentioned in the previous section, different regularizers can be applied to the sequential part of WASP. The tests for different regularizers are conducted using the noiseless blood cell data with the small size probe (128×128 pixels, 6400 diffraction patterns). The first one is testing the new WASP regularizer from Equation (6.3) and (6.4) with different parameters. The results are displayed in Figure 6.4.

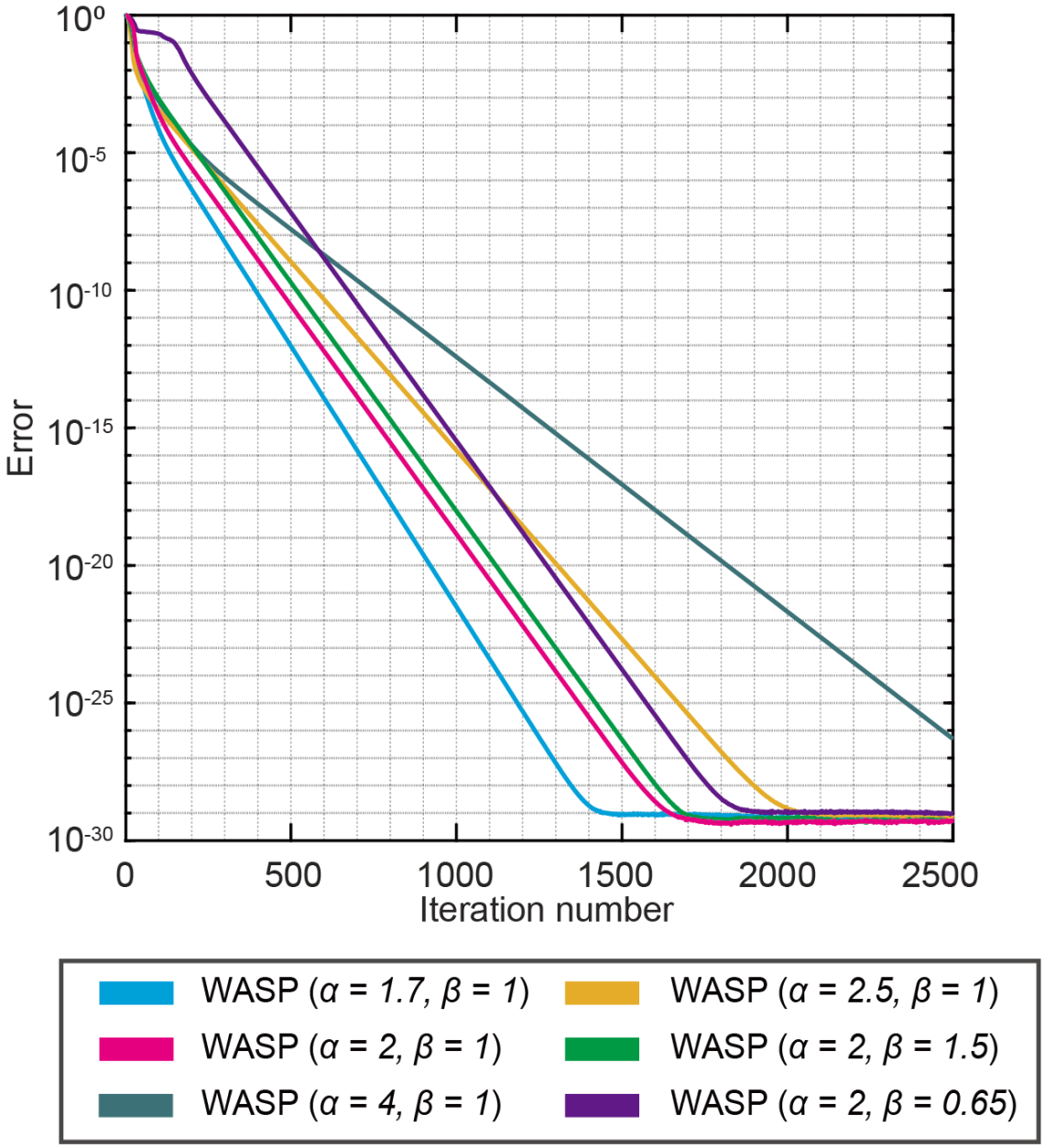

Figure 6.4. The results for the new WASP regularizer with different parameters.

The new regularizer uses the average operation instead of the maximum compared to the PIE style regularizer. It generally works well from Figure 6.4. The value of α has a greater impact on the reconstruction than β, smaller α gives a better performance in the test. Note that when $\alpha < 1.7$ in our tests, the reconstruction of WASP initially collapsed. Therefore, we chose $\alpha = 2$ and $\beta = 1$, for our WASP regularizer.

The second test is for the PIE style regularizer. The results are shown in Figure 6.5, compared to the WASP regularizer with $\alpha = 2$ and $\beta = 1$.

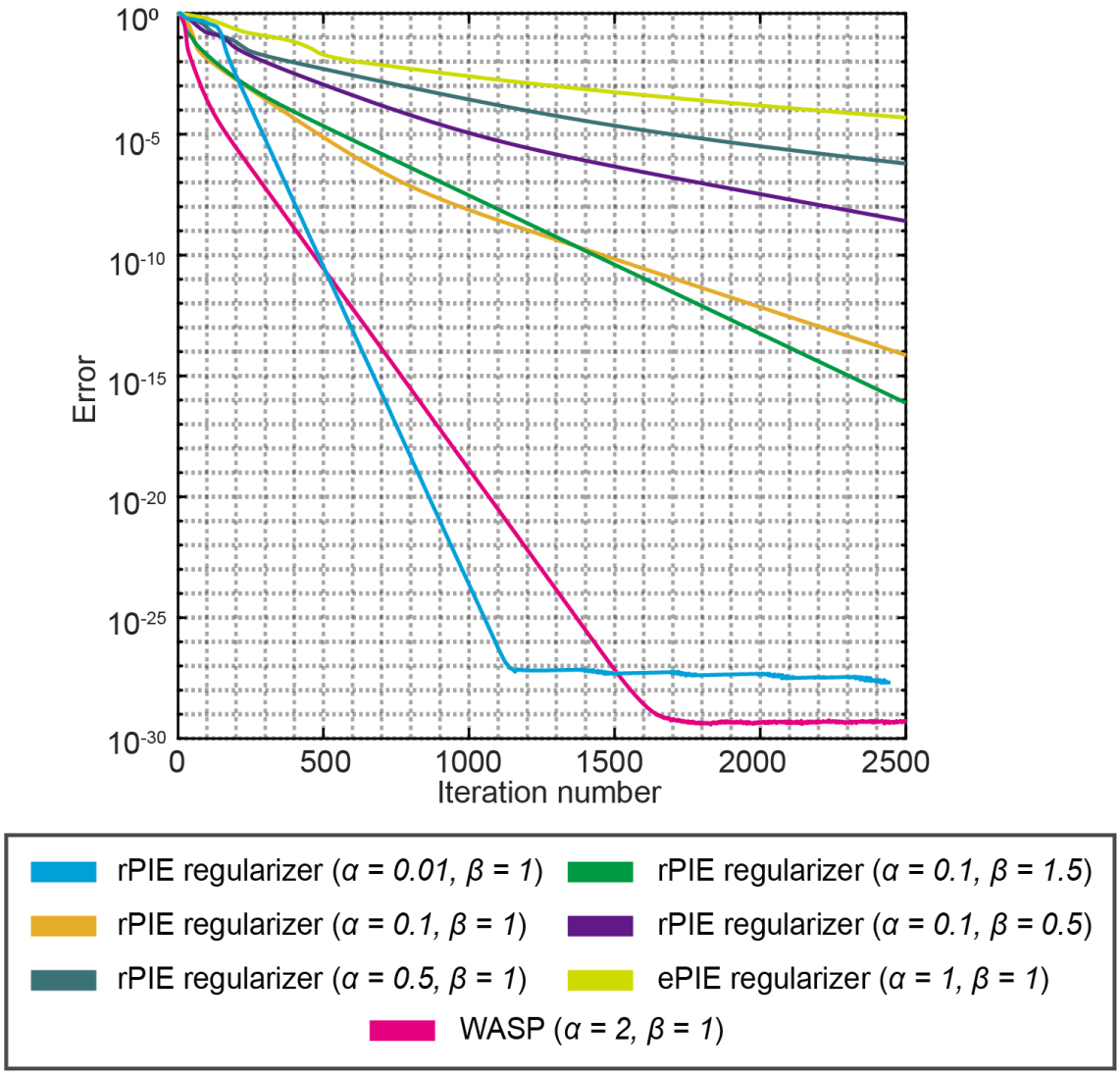

Figure 6.5. The results for the PIE style regularizer with different parameters.

As Figure 6.5 shown, rPIE regularizer with a very small $\alpha = 0.01$ performs well in this test. However, when $\alpha = 0.1$, which is the value normally recommended in rPIE, the convergence rate is much slower, and the different value of α and β give relatively large differences in results. Comparing Figure 6.4 and Figure 6.5, in general, the new WASP regularizer is better than PIE style regularizer in terms of stability and convergence speed. Therefore, we will use the new regularizer with $\alpha = 2$ and $\beta = 1$ for our further tests.

6.2.2. Noiseless Simulation Results

The first noiseless simulation is the blood cell data from an optical bench ptychography experiment. The configuration was described in Chapter 5.5.1. There are two different sizes of probe used in this simulation, the simulation error of the large one (512×512 pixels, 400 diffraction patterns) is shown in Figure 6.6. Note that apart from WASP, the results of other algorithms are the same as shown in Chapter 5.5. The tuning results of GAT can also be found in Chapter 5.5.

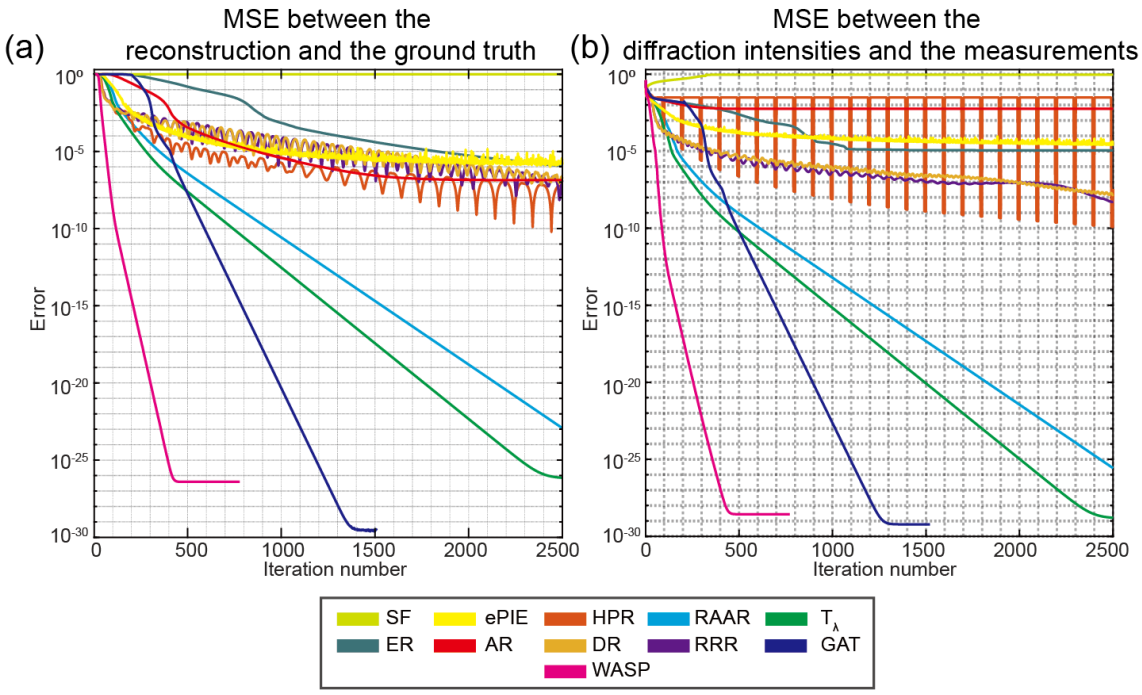

Figure 6.6. The error of simulation with large size probe (512×512 pixels, 400 diffraction patterns), (a) The MSE between the reconstruction and the ground truth, (b) The MSE between the reconstructed diffraction patterns and the measurements.

The reconstruction of WASP in this simulation is displayed in Figure 6.7, the centre part of it is zoomed in to show more details.

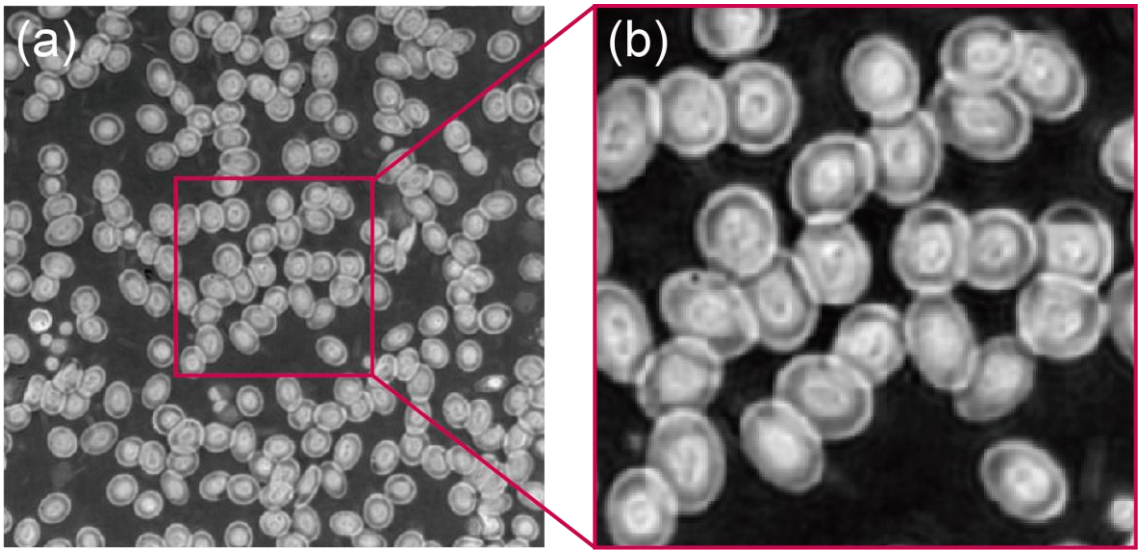

Figure 6.7. (a) The phase reconstruction of the simulation using large size probe (512×512 pixels). (b) The centre region with 200×200 pixels are displayed. The red box indicates the zoomed area.

The second simulation is using a small size probe (128×128 pixels, 6400 diffraction patterns). The simulation error is shown in Figure 6.8.

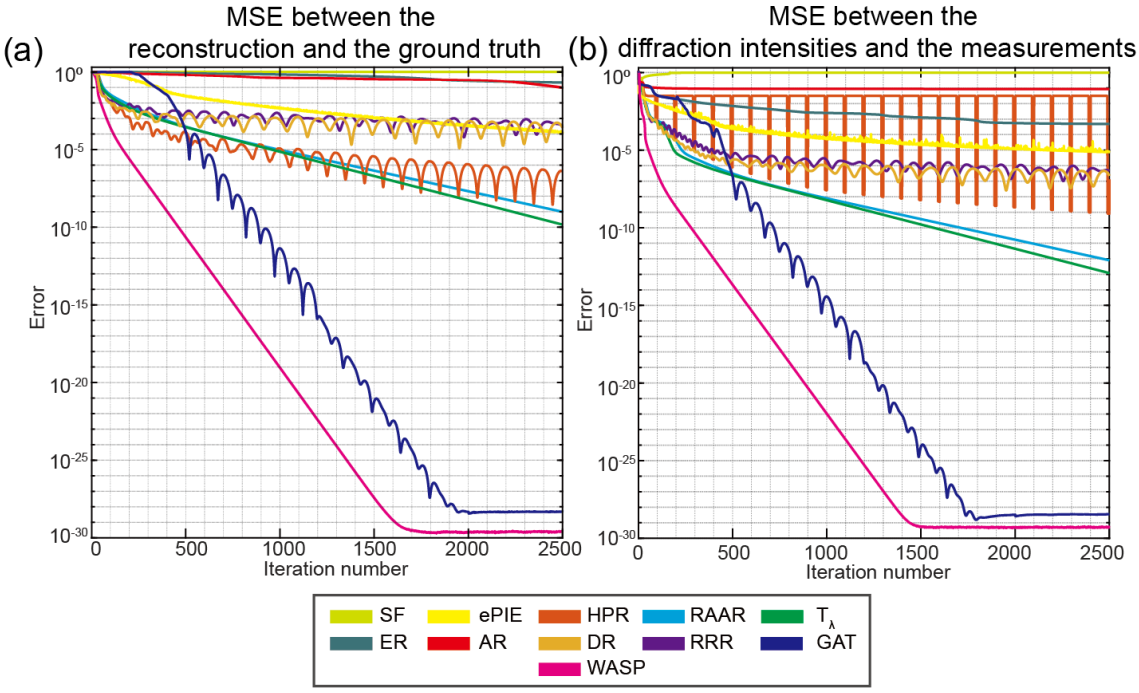

Figure 6.8. The error of simulation with small size probe (128×128 pixels, 6400 diffraction patterns), (a) The MSE between the reconstruction and the ground truth, (b) The MSE between the reconstructed diffraction patterns and the measurements.

As we can see from first simulation with the large probe, most algorithms can reach the threshold of the simulation error that equal to 10^{-5}; the reconstruction below this threshold is visually indistinguishable from the ground truth. In this case, WASP only takes around 400 iterations to reach the minimum error. There are four algorithms (RAAR, T_λ, GAT, and WASP) reached around the global minimum at the working precision of the computer, but RAAR and T_λ take around 2500 iterations to achieve this error level. The error of GAT is the lowest one as it is optimized based on error curve, but it is more time consuming due to the auto-tuning process. This is slightly different in the second simulation with the smaller sized probe. The small probe results in less information in each diffraction pattern and less overlapping between them, only WASP and GAT reached the minimum error within 2500 iterations. RAAR and T_λ seem to have a trend to converge, but it will take thousands of iterations before this eventually happens. WASP in this case is the most successful one from the error curve in Figure 6.8, it converges very fast and has the minimum error. WASP demonstrates a rapid initial convergence rate compared to other methods in both simulations, therefore, in the next section, a test with different initial conditions will be carried out to see its performance.

6.2.3. Initial Convergence

This test is about the robustness to the initial conditions, different initial errors will be added to the probe defocus. The test is using same blood cell object with the large size probes (512×512 pixels) that has different defocus error. The probe modelled the stopped-down beam from a soft X-ray source of 515 eV with an 8 mrad convergence semi-angle and a defocus of 750μm. The initial probes varied from 500 to 1000μm, equivalent to defocus errors from -33 to 33%. The converge requirement for each algorithm is that the reconstruction error reaching 10^{-4}, the number of iterations it used will indicate its initial convergence rate. There are five algorithms selected for this test, basically

picked from three different levels of final error that shown in Figure 6.6 (a). GAT is not chosen here because its convergence rate is highly dependent on the tuning interval. The results are shown in Figure 6.9.

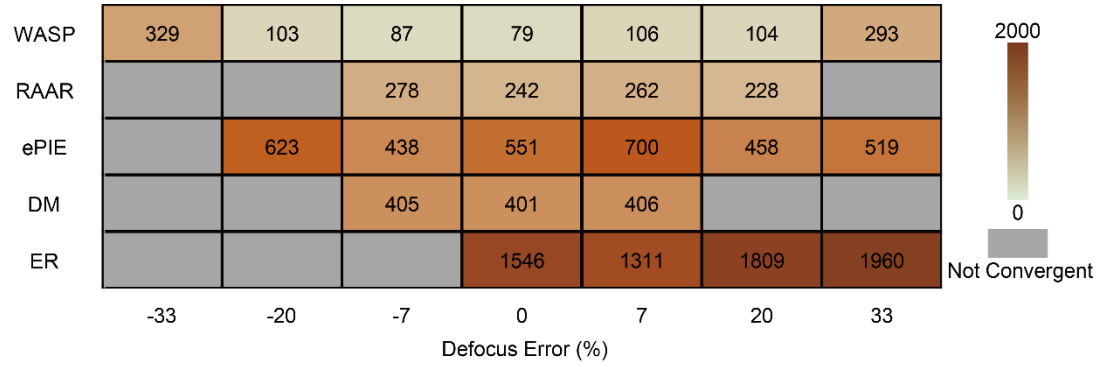

Figure 6.9. The convergence speed of different algorithms with different defocus error in the initial probe.

From the table in Figure 6.9, WASP generally does very well in all the range of defocus error in our test, in most cases, it only takes around 100 iterations to converge. Notable here is that the sequential projection algorithm (ePIE) and WASP can handle a poorer initial condition than set projection algorithms (RAAR, DM, ER) which rapidly diverged in the first few iterations when the initial defocus error was larger than 10%. Because of the characteristic that always reducing the error in ER method, it does converge eventually for a quite wide range of defocus errors, but it costs thousands of iterations.

6.2.4. Noise Simulation Results

Another test is the noise test, same as the previous test for the set projection algorithms in Chapter 5.5.4. The simulation object is a designed phase object; different levels of Poisson-distributed noise were added into the data, the details are shown in Figure 5.19. The first result is from a moderate level error which has 5×10^5 counts in each diffraction pattern, is in Figure 6.10.

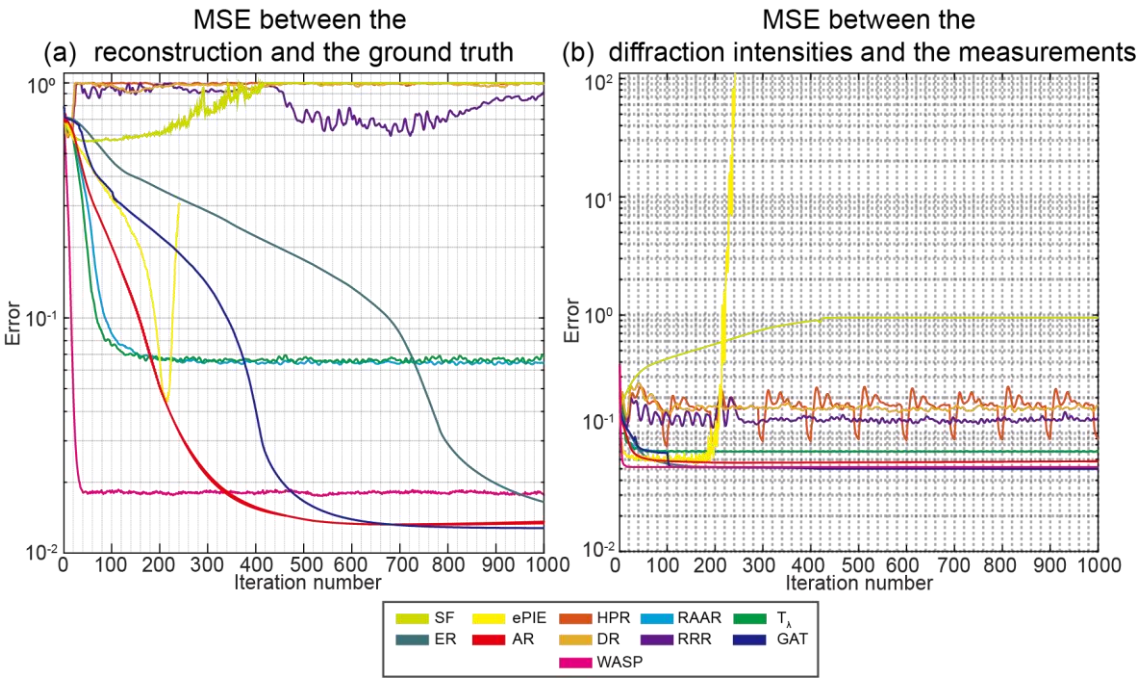

Figure 6.10. The results from different algorithms, for the moderate level noise simulation (5×10^5 counts per diffraction pattern). (a) The MSE between the reconstruction and the ground truth, (b) The MSE between the reconstructed diffraction patterns and the measurements.

WASP takes only 20 iterations to converge in the noise test, this is much quicker than other algorithms. Also, the performance of WASP considerable good, the phase reconstruction of WASP is shown in Figure 6.11, compared to other algorithms.

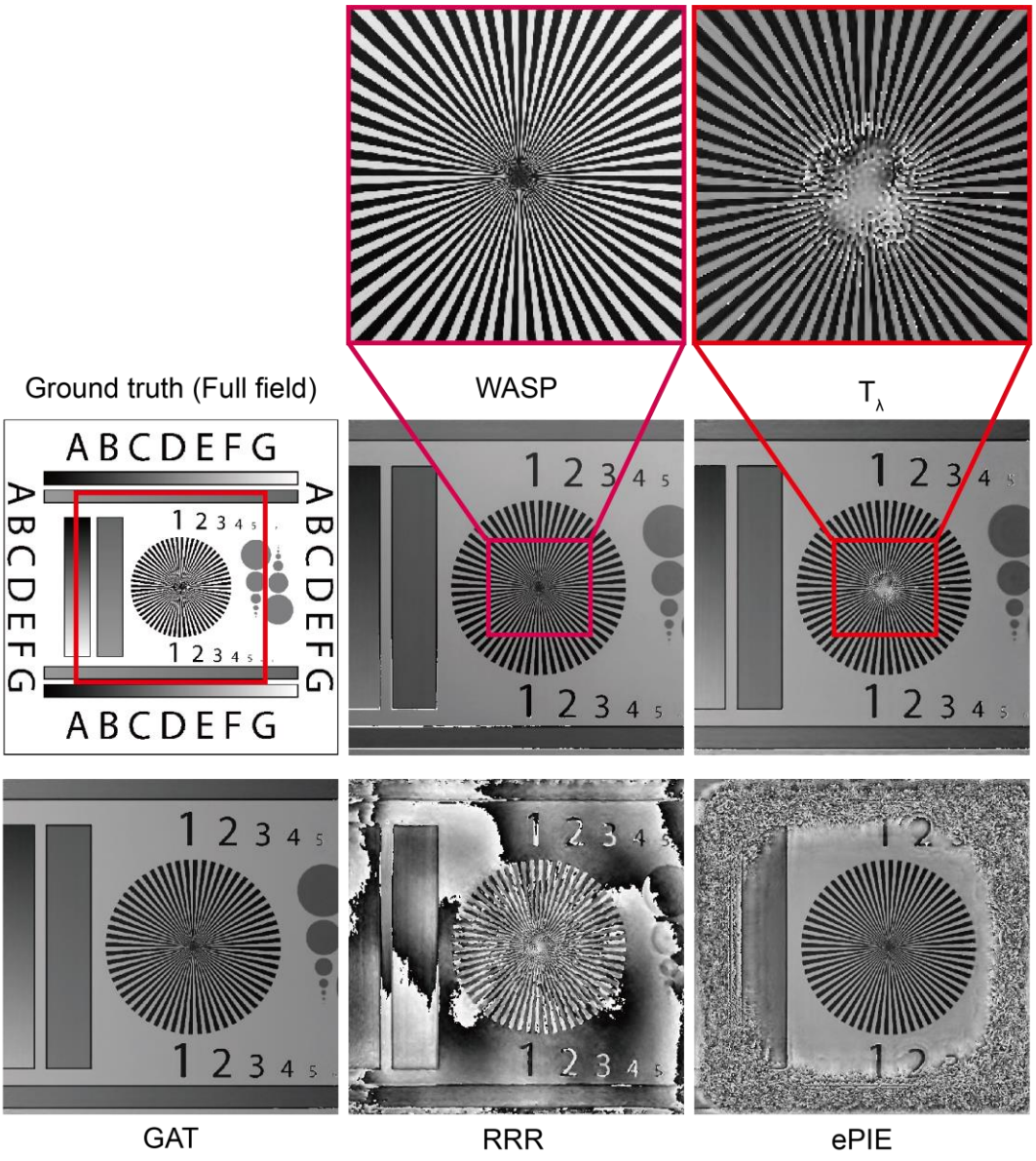

Figure 6.11. The reconstructions from different algorithms for the moderate level noise simulation (5×10^5 counts per diffraction pattern).

As we can see, the reconstruction of WASP has a very high resolution at the centre, it only has some defects at some edges, better than T_λ and RAAR. Compared to GAT, WASP has more advantages in the time cost since the good reconstruction from GAT needs several times tuning.

The second noise test is adding higher level noise to the data, which only has 10^4 counts in each diffraction pattern. The simulation error is displayed in Figure 6.12.

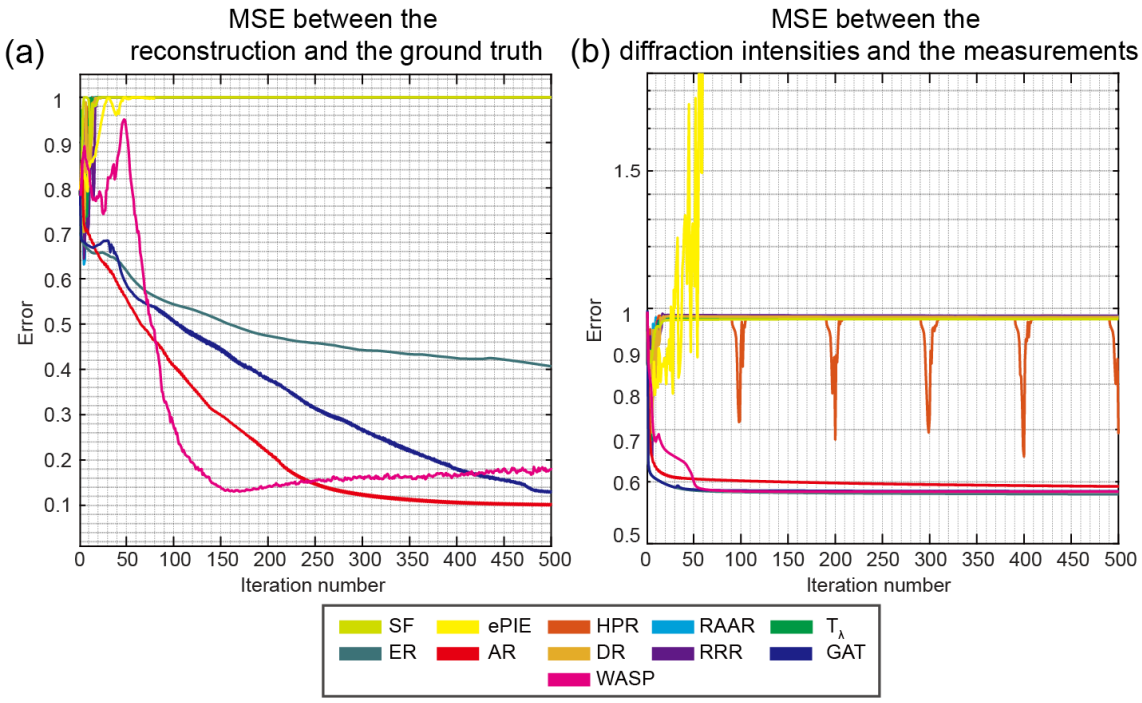

Figure 6.12. The results from different algorithms, for the higher level noise simulation (10^4 counts per diffraction pattern). (a) The MSE between the reconstruction and the ground truth, (b) The MSE between the reconstructed diffraction patterns and the measurements.

In this case, most set projection algorithms failed in the test, including T_λ and RAAR, which generally did very well in noiseless and moderate noise tests. ER has a very good noise tolerance, and it is stable in all the tests. WASP has the fastest convergence rate, however, the error slightly goes up after it reached the minimum. This phenomenon is probably caused by the local minima issue in the sequential part since it also happens in ePIE.

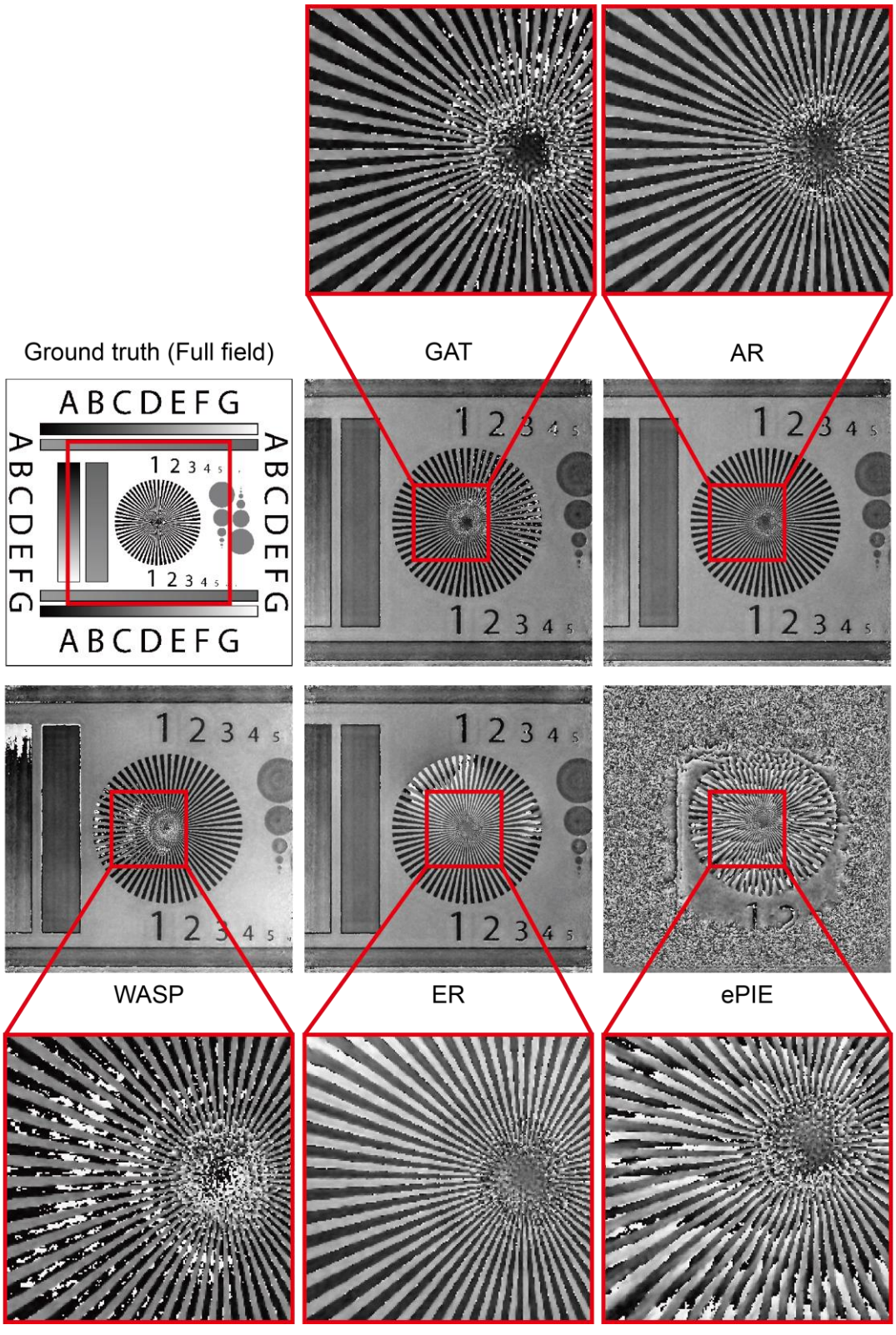

Figure 6.13. The reconstructions from different algorithms for the higher level noise simulation (10^4 counts per diffraction pattern).

Figure 6.13 illustrates the reconstructions from different algorithms, the

performance WASP is not bad in this very noisy simulation. As a combination of ER and ePIE, it improved a lot compared to ePIE, also it is faster than ER.

6.3. Parallel WASP

As mentioned before, apart from the small memory footprint, rapid initial convergence rate and robustness to noise and poor initial conditions, another important advantage of WASP is that it can be parallelized. Unlike set projection algorithms, sequential projection algorithms such like ePIE or rPIE cannot be fully parallelized, since in the sequence, any new estimate from one position has to feed into the next position. One solution of this is to divide all the positions into mini-batches [16], each of them handle a sub-set of the projections. These sub-sets can be processed in parallel by a similar way as set projections algorithm does. However, it is sequentially between mini-batches, the output of one mini-batch need to feed to the next one, finally, the output of batches feed serially into the object and probe updates. Here, we present a different way of parallelizing the sequential projection with the idea of WASP. A parallel example of the circle problem is illustrated in Figure 6.14, another constraint V is introduced for better demonstration. For this four constraints problem, we equally divide them into two mini-batches: $[T, V]$ and $[U, S]$. Then, two sequential projections are conducted on each mini-batches, this step can be fully parallelized and distributed to different workers where each of them will handle one sequential projection. When all the sequential projections are completed, a weighted average (WA) will bring them into a single solution.

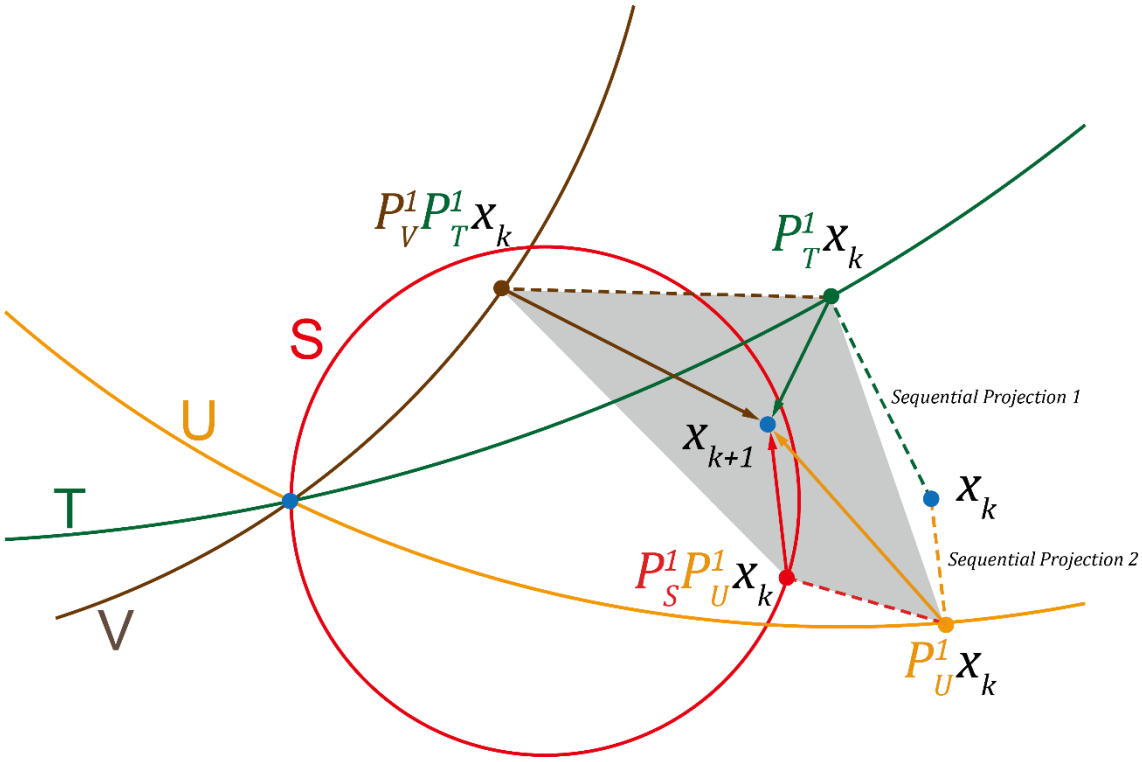

Figure 6.14. Parallel WASP

Therefore, this is a new parallel way of sequential projection method that serial feeds parallel. In the ptychography application, the full data of diffraction patterns will be divided into random mini-batches. Each of them is assigned to a different worker, which conducts the sequential projection (SP) part of WASP. The outputs of each worker are four partially-filled matrices: the sums for the numerator and denominator of updating matrix that was shown in Figure 6.3. Finally, when all the workers finish the assigned job, these outputs will be summed and divided according to the weighted average (WA) part in Figure 6.3. A flow chart of parallel WASP is shown in Figure 6.15.

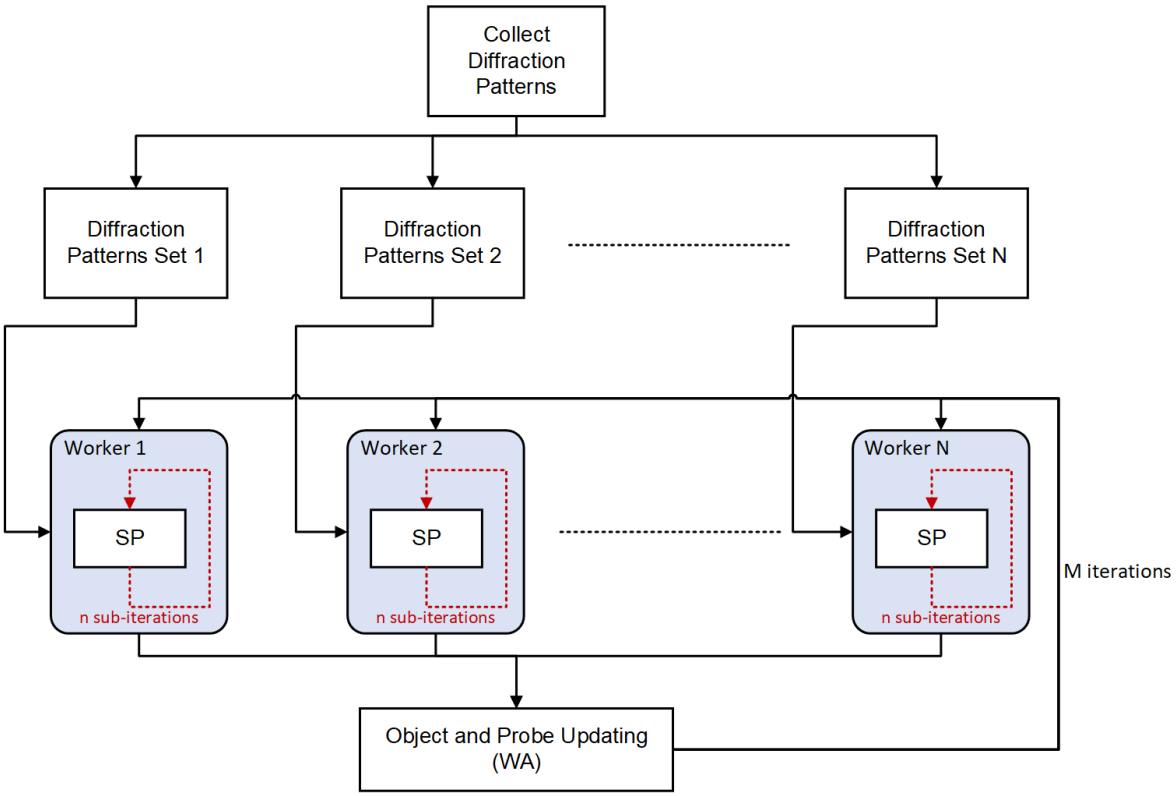

Figure 6.15. Flow chart of Parallel WASP

The algorithm of parallel WASP can be written in two parts, the first one is called WASP Hive, which distribute jobs to different workers, collect and process the outcomes from them. The pseudocode of WASP Hive is shown in **Pseudocode 6.2**.

Pseudocode 6.2: The WASP Hive

Inputs*: object* (obj), *probe function* (probe), *diffraction pattern sets* (DPset), *the number of workers* (N), *the total number of iterations* (J).
Outputs*: reconstructed object* (obj) *and probe* (probe)

```
1   For (j = 1 to J) do
      // Parallel for loop
2     Parfor (n = 1 to N) do
        // Assign diffraction pattern sets to different workers and collect the results from them
3       [topO_n, bottomO_n, top_n, bottomP_n] = WASPwoker(obj, probe, DPset_n)
4     End loop
      // Weighted average update of object and probe
5     obj = ∑_n topO_n /(∑_n bottomO_n + eps)
6     probe = ∑_n topP_n /(∑_n bottomP_n + eps)
7     Apply any additional constraints
8   End loop
```

Note: Parfor: parallel for loop. **WASPwoker***: a function that does SP part of WASP*. ***eps:*** a small constant in MATLAB to avoid dividing 0. $\sum_n$: sum along n direction.

Different diffraction pattern sets are allocated to different workers at the same time in a parallel for loop at line 3 in **Pseudocode 6.2**. Assume each worker has the same computational power, diffraction patterns are suggested to be equally divided into mini-batches. The algorithm for a WASP worker is shown in **Pseudocode 6.3**.

Pseudocode 6.3: The WASP Worker

Inputs*: position vectors* (R), *object* (obj), *object size* (X, Y), *probe function* (probe), *probe size* (M, N), *intensity at relevant positions* (I), *the number of allocated positions* (K), *the number of sub-iterations* (J).

Outputs*: numerator and denominator sums for object* (topO & bottomO) *and probe* (topP & bottomP)

```
1   For (j = 1 to J) do
2     topO = bottomO = zeros(X,Y)
3     topP = bottomP = zeros(M,N)
      // Initialise numerator and denominator sums
4     R = shuffle(R)
5     For (k = 1 to K) do
        // Form exit waves and apply the modulus constraint
6       objBox = obj(Rk to Rk+[M,N])
7       exitWavek = objBox·probe
8       detectorWavek = ℱ(exitWavek)
9       correctedWavek = sqrt(Ik)·detectorWavek /(abs(detectorWavek) + eps)
10      newExitWavek = ℱ⁻¹(correctedWavek)
11      ΔexitWavej = newExitWavek-exitWavek
        // Sequential projection update of object and probe
12      obj(Rk to Rk+[M,N]) += conj(probe)·ΔexitWavej /abs(probe)² + A
13      probe += conj(objBox)·ΔexitWavej /abs(objBox)² + B
        // Update numerator and denominator sums
14      topO += conj(probe)·newExitWavek
15      bottomO += abs(probe)²
16      topP += conj(objBox)·newExitWavek
17      BottomO += abs(objBox)²
18    End loop
22  End loop
```

Note: zeros: a matrix full of zeros. **shuffle***: a function that randomly change the order of the position sequence*. ℱ: Fourier transform. $\mathcal{F}^{-1}$: inverse Fourier transform. ***eps:*** a small constant in MATLAB to avoid dividing 0. **sqrt**: square root, **abs**: amplitude, **conj**: complex conjugate

A single WASP worker will carry out the sequential projection part (SP) of WASP, the **Pseudocode 6.3** also is same as **Pseudocode 6.1** but without line 19-21. Because sequential projection algorithms have a very good initial convergence rate, only a few iterations can return a reasonable initial guess. Running the SP part for a number of sub-iterations generally provides a better quality of the

output sums. Moreover, run time of a single iteration of SP part is relatively small. Hence, a single iteration for each worker will significantly waste more time on the job allocation and data collecting in the Hive to achieve the convergence. On the other aspect, more workers will reduce the number of positions assigned to each one. The allocation of positions is random, however, actually some positions can be assigned more than one times to different workers. This causes the redundancy between WASP workers, higher redundancy will improve the convergence rate but also increase the computation time in each worker. In the next part, the simulation of different number of workers, sub-iterations and the redundancy will be discussed and compared.

6.4. Simulation Results for Parallel WASP

The simulations of parallel WASP use the same blood cell data as shown in Chapter 5.5.1. The test is noiseless and using the small size probe (128×128 pixels, 6400 diffraction patterns).

6.4.1. Simulation for Different Number of Workers

The first simulation shows the influence of different number of workers. In total 6400 diffraction patterns are equally divided into workers with 20% redundancy between each worker. Also, the number of sub-iterations is fixed to 5 in this test. The error metric is displayed in Figure 6.16.

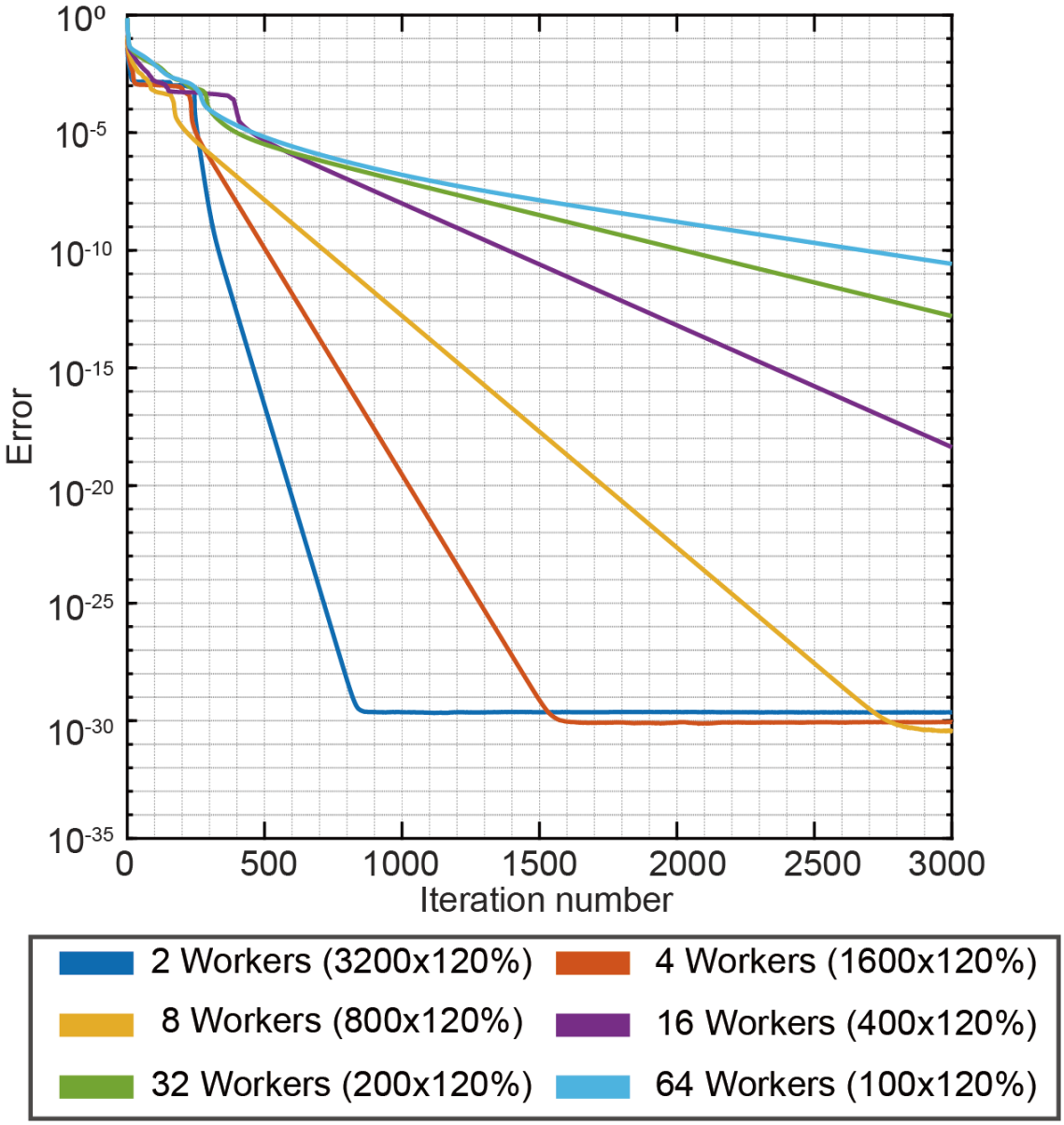

Figure 6.16. The error metric of different number of WASP workers with 5 sub-iterations and 20% redundancy, the brackets indicate the number of diffraction patterns allocated to each worker.

As the number of workers increases, it takes more iterations to converge since each worker handle fewer diffraction patterns. Each time in a single worker, it processes a smaller fraction of the whole data will increase frequency of job distribution at the Hive. This relationship is nearly linear. Notable here is that more iterations of convergence do not mean more time consumption to reach the convergence. More workers will significantly reduce the time of each iteration due to the parallel computing and smaller data fraction. Ideally, if the

time cost by job distribution and collection is not considered, then in the best-case scenario, doubling the number of workers will reduce the time per iteration by half. However, as the parallel WASP has not optimized for the parallel computations sufficiently, it is premature here to make any quantitative time analysis.

6.4.2. Simulation for Different Number of Sub-iterations

The second test is changing the sub-iterations. In this simulation, the number of workers is fixed to 8 with 20% redundancy. The results are shown in Figure 6.17.

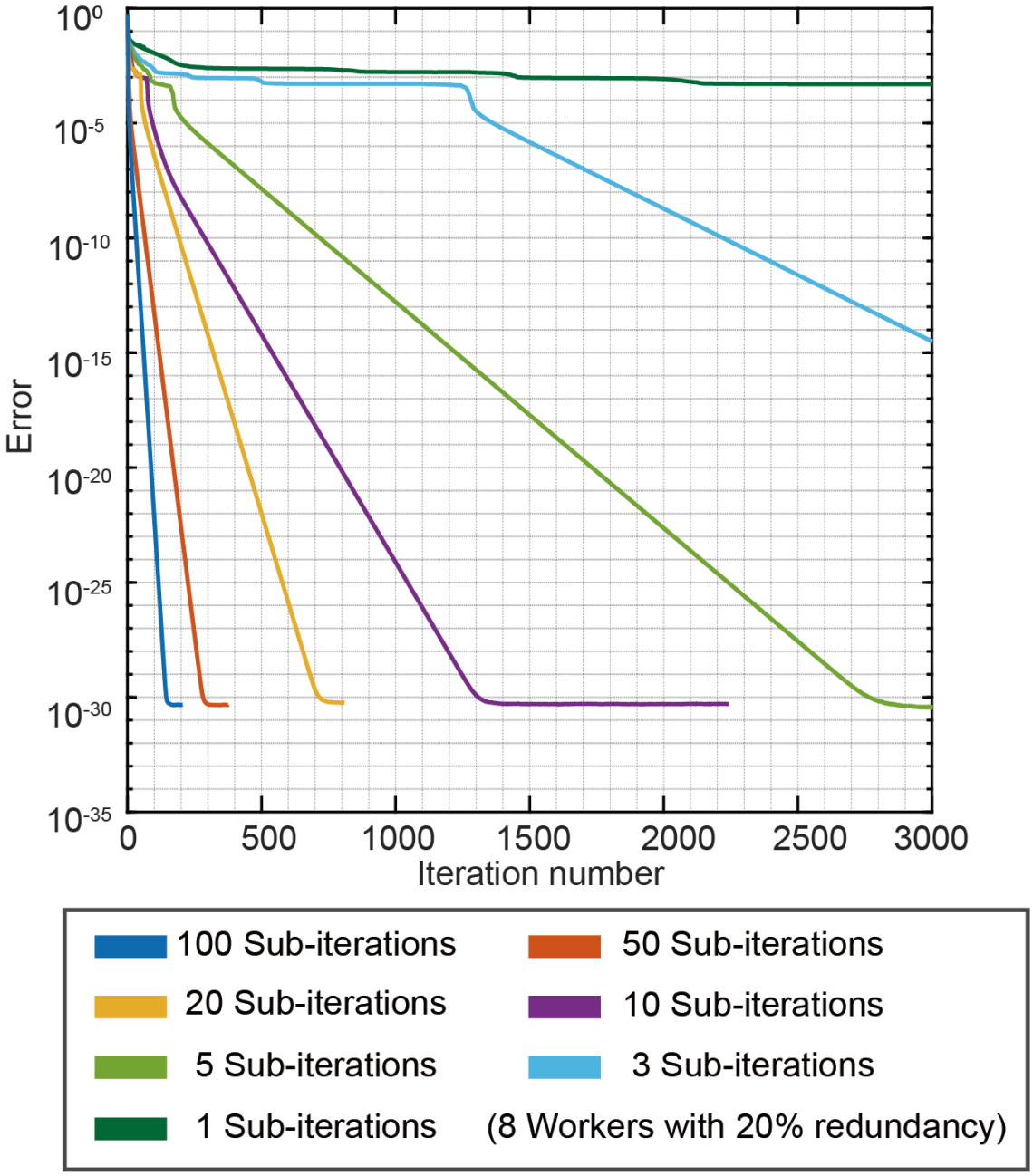

Figure 6.17. The error metric of different number of sub-iterations for 8 workers and 20% redundancy.

Generally, sub-iterations provide more prior knowledge from the overlapping between diffraction patterns through the sequential projections. Figure 6.17 illustrates the convergence rate of different sub-iterations. Similarly, in the best theoretical scenario, doubling the number of sub-iterations will double the time consumption in each worker. In return, a better outcome from the workers will reduce the iterations for the convergence. From the error metric, 100 sub-

iterations cost around 150 iterations while 50 sub-iterations cost about 300 iterations; another pair is that 20 sub-iterations use about 700 iterations to converge while 10 sub-iterations need around 1300 iterations. The ratio in these two pairs is around factor 2, roughly matching the ideal time model. If we multiply the convergence iteration number with the sub-iterations, the total number of iterations is not much different as shown in Table 4.

Table 4. The relationship between sub-iterations and the total iterations for the convergence.

Sub-iterations	Convergence iteration number (approx.)	Total iterations
100	150	15000
50	290	14500
20	700	14000
10	1350	13500
5	2900	14500

Theoretically, it indicates that the number of sub-iterations does not significantly affect the final time consuming of the convergence. However, in practice, sub-iterations affect the times of job allocations, and loading data to each worker will cost time as well.

6.4.3. Simulation for Different Redundancy

The final test is about the redundancy between workers. In this simulation, there are 8 workers with 5 sub-iterations in each. The redundancy represents the overlapping between mini-batches. Each batch will have some percentage of the duplicated diffraction patterns from other batches. This enhances the connection between each worker, improves the final global weighted average updating. The simulation results are shown in Figure 6.18.

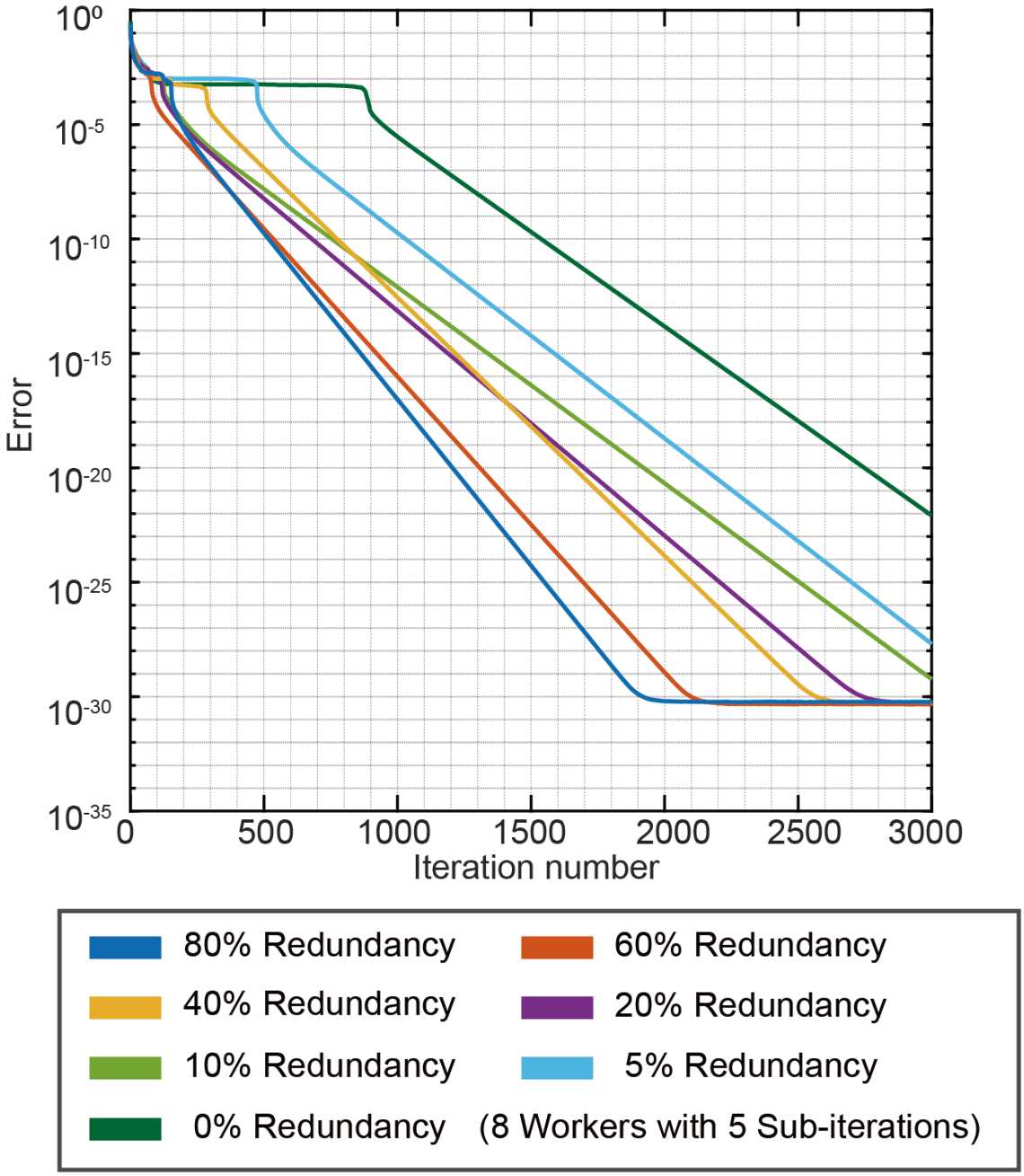

Figure 6.18. The error metric of different redundancy for 8 workers with 5 sub-iterations.

The higher redundancy can reduce the convergence iterations, but the relationship seems not linear from Figure 6.18. Also, the increased time consumption of adding redundancy is difficult to evaluate. From our experience, 20% redundancy generally works well, it can speed up the convergence rate with only a little additional time consumption. In another aspect, if a massive number of workers are used, in this case, there are only a few diffraction

patterns allocated to each worker, the redundancy will affect the max limitation of the parallelism.

6.5. Conclusion

In this chapter, we combined the sequential projection method and the idea of “divide and concur” in set projection methods, proposed a new approach called WASP. Like the sequential projection method, WASP has a rapid initial convergence rate, small memory requirement, and can handle a poor initial condition. On the other hand, WASP also has the ability to reach a global minimum and is parallelizable for large size data. It generally has the advantage of both sequential projection method and set projection methods. Moreover, it also has a good tolerance for the noise. WASP provides a new way to do the parallel computing in ptychography. This chapter analysed three different aspects that could affect the parallelism, such as the number of workers, sub-iterations and redundancy between mini-batches. In the future, optimizing the regularizer for WASP can perhaps further improve its performance. Also, more algorithmic techniques in ptychography such like position correction [62], background noise correction [63-66] can be applied to WASP. Furthermore, WASP provides a basic framework for ptychography. Based on this framework, different ptychographic applications can be implemented as well, such as 3D ptychography [67, 68], multi-slice [69-71] and modal decomposition [72-75], which gives WASP the potential for its future development and extension.

7. Real-World Experiments

In this chapter, the different algorithms mentioned in previous chapters will be applied to the different types of real-world ptychography experiments to compare their performance.

7.1. Optical Experiment

The first data is from an optical experiment with visible light [59, 76]. The specimen is a plant leaf structure held by a microscope slide. It is a far-field experiment using laser illumination with a 675 nm wavelength. The data collection NA is 0.25, and probe NA is 0.16. The reconstruction results of some algorithms are shown in Figure 7.1. The result of RAAR is almost identical as T_λ, and DR is similar to RRR which has a very distinct phase ramp across the reconstruction. The error lines are displayed in Figure 7.2. WASP shows a very fast initial convergence rate, and it has the lowest error at the end. GAT worked as ER for a long time at the beginning, then after the successful tuning, the reconstruction is much improved compared to the ER. The tuning parameters are indicated in Figure 7.3.

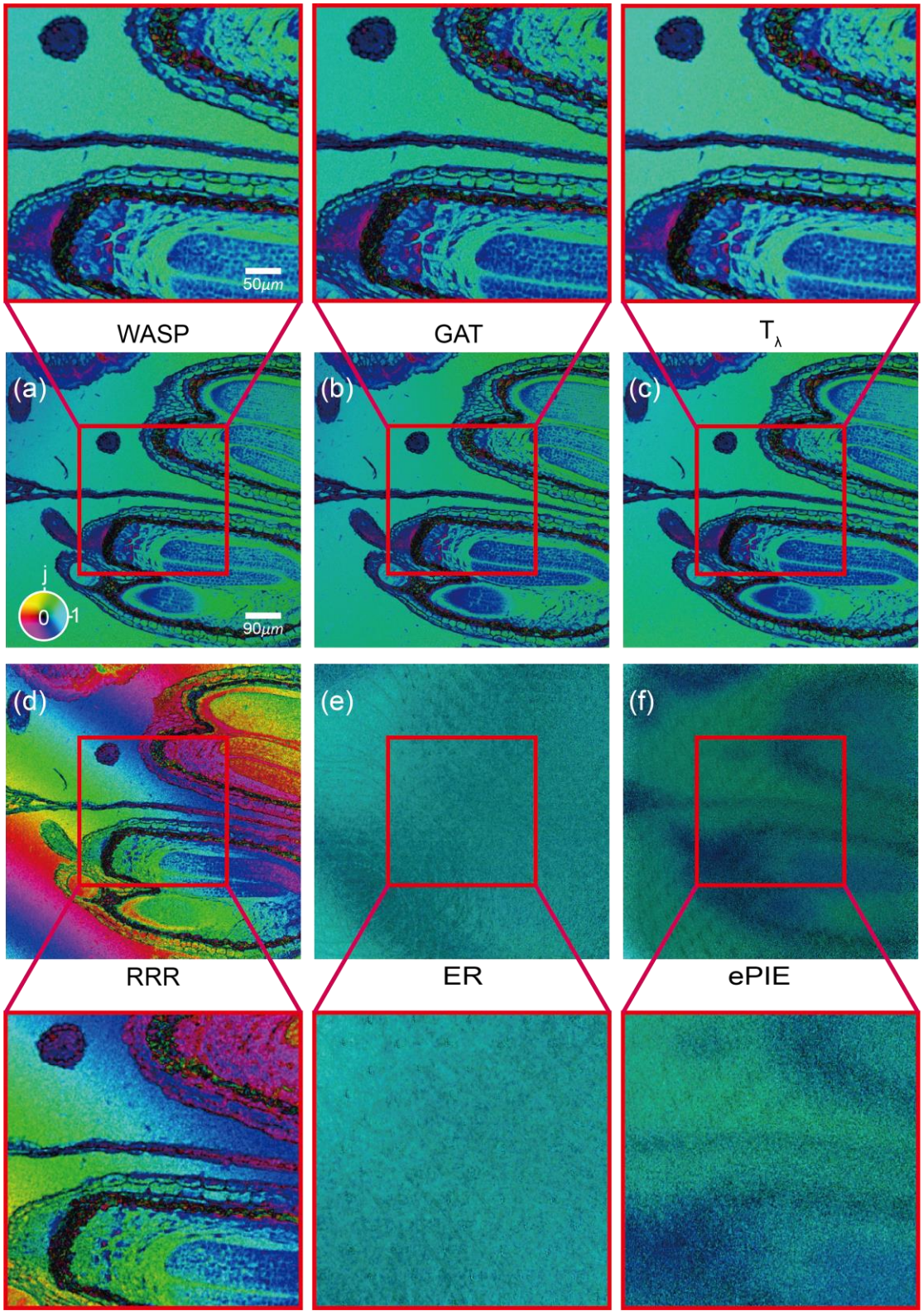

Figure 7.1. The reconstruction results of the plant structure. (a) WASP, (b) GAT, (c) T_λ, (d) RRR, (e) ER, (f) ePIE.

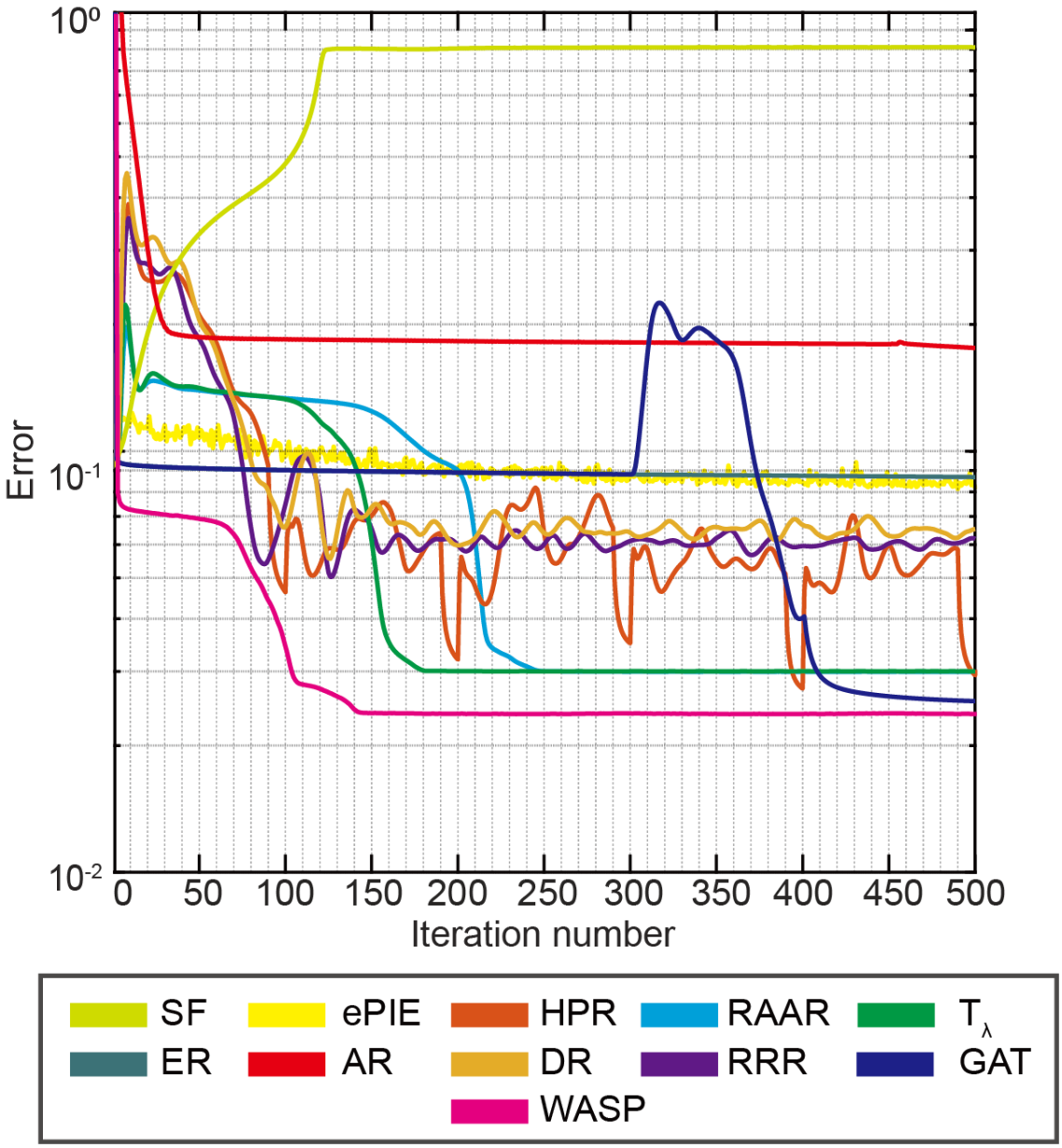

Figure 7.2. The error metric for the plant structure experiment.

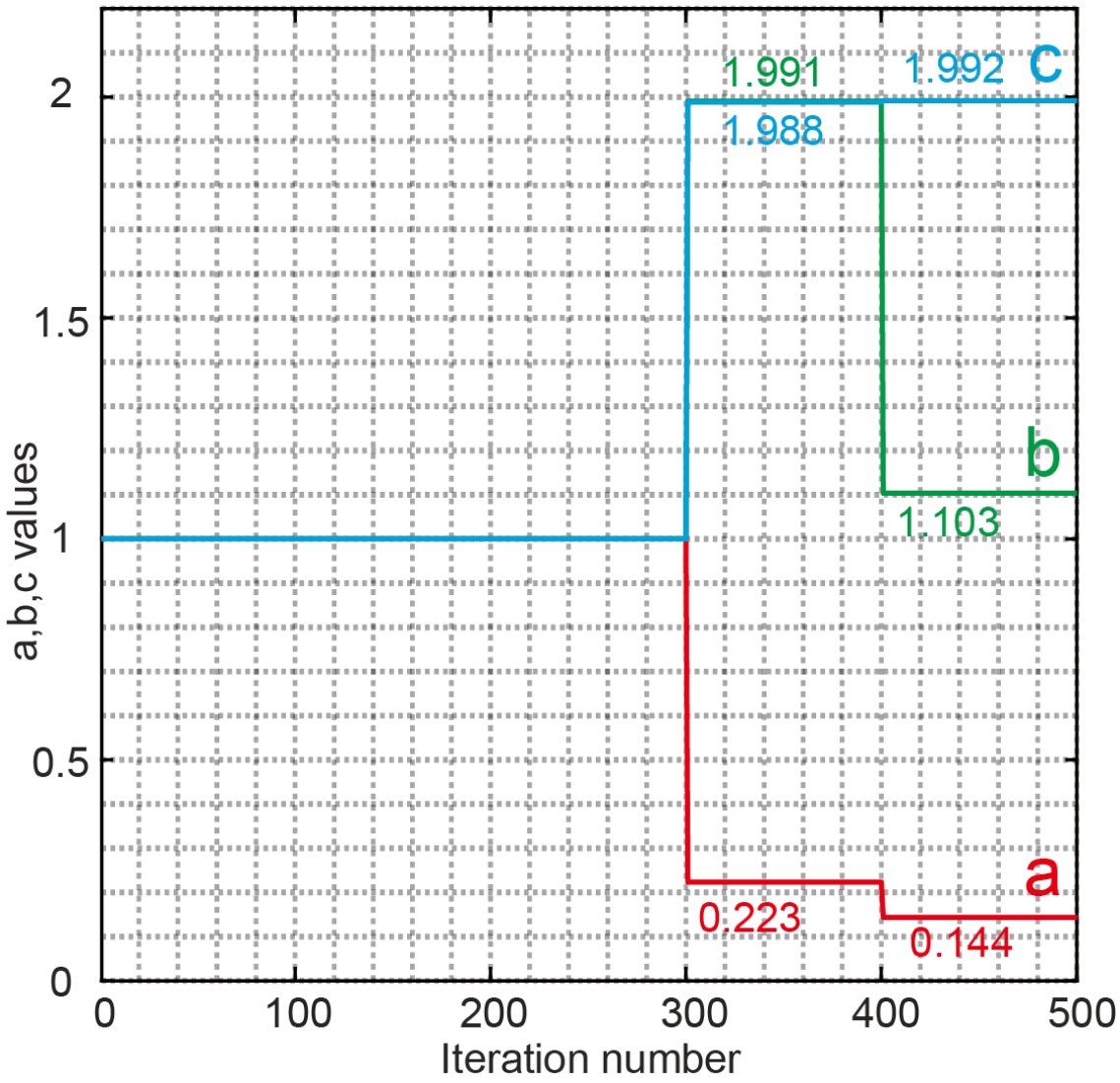

Figure 7.3. The values of a, b, c for Generalized Auto-Tuning (GAT) algorithm for the plant structure reconstruction.

7.2. X-ray Experiment

The second experiment data is from a near-field X-ray experiment [69]. The X-ray experiment used a cone beam geometry with a beam energy of 10 keV. The specimen is a Siemens star, which is designed to have features of varying spatial frequencies, making it useful for evaluating the resolution and contrast capabilities. The reconstruction results are illustrated in Figure 7.4, and the error metric is shown in Figure 7.5.

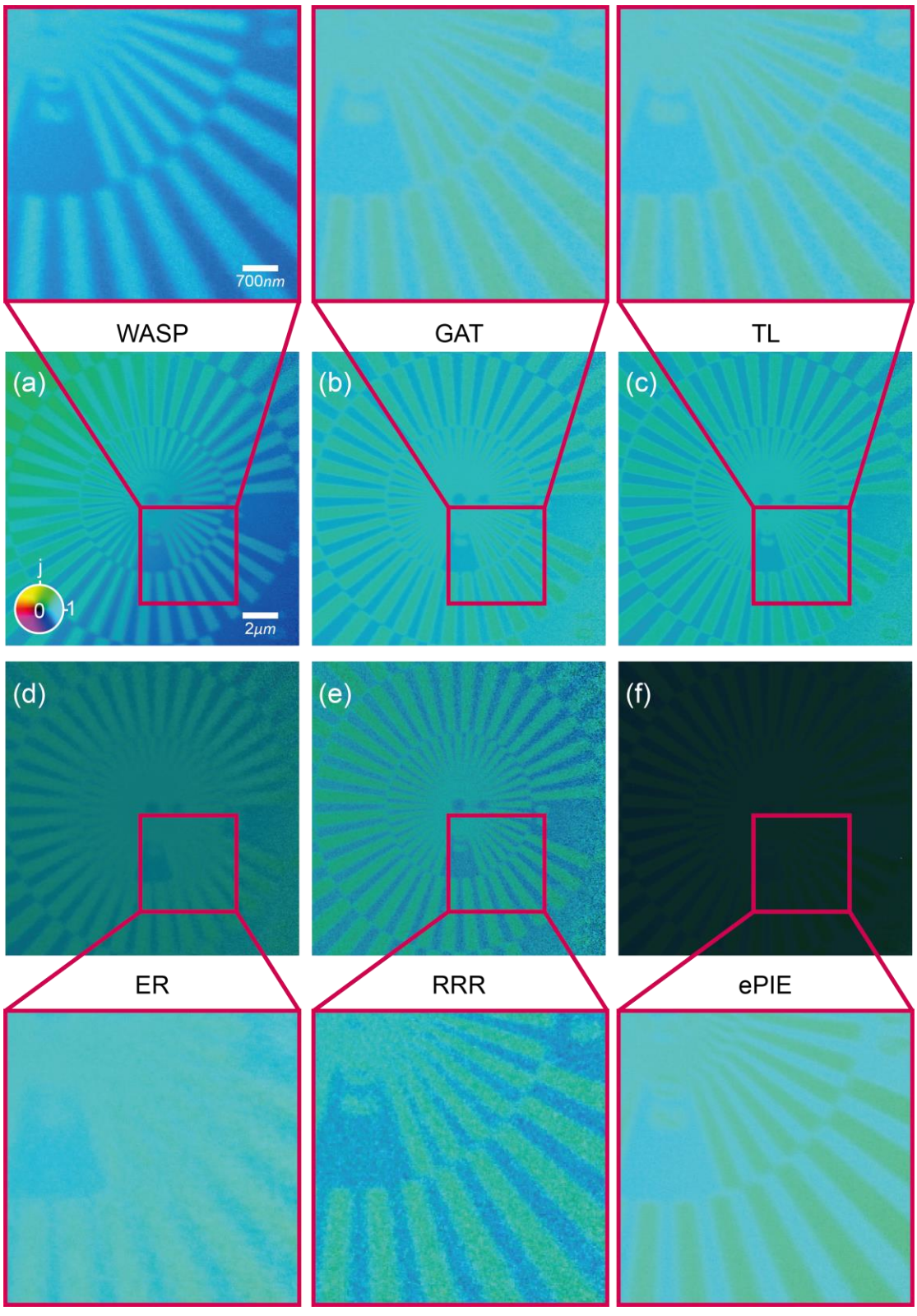

Figure 7.4. The reconstruction results of the Siemens star. (a) WASP, (b) GAT, (c) T_λ, (d) ER, (e) RRR, (f) ePIE.

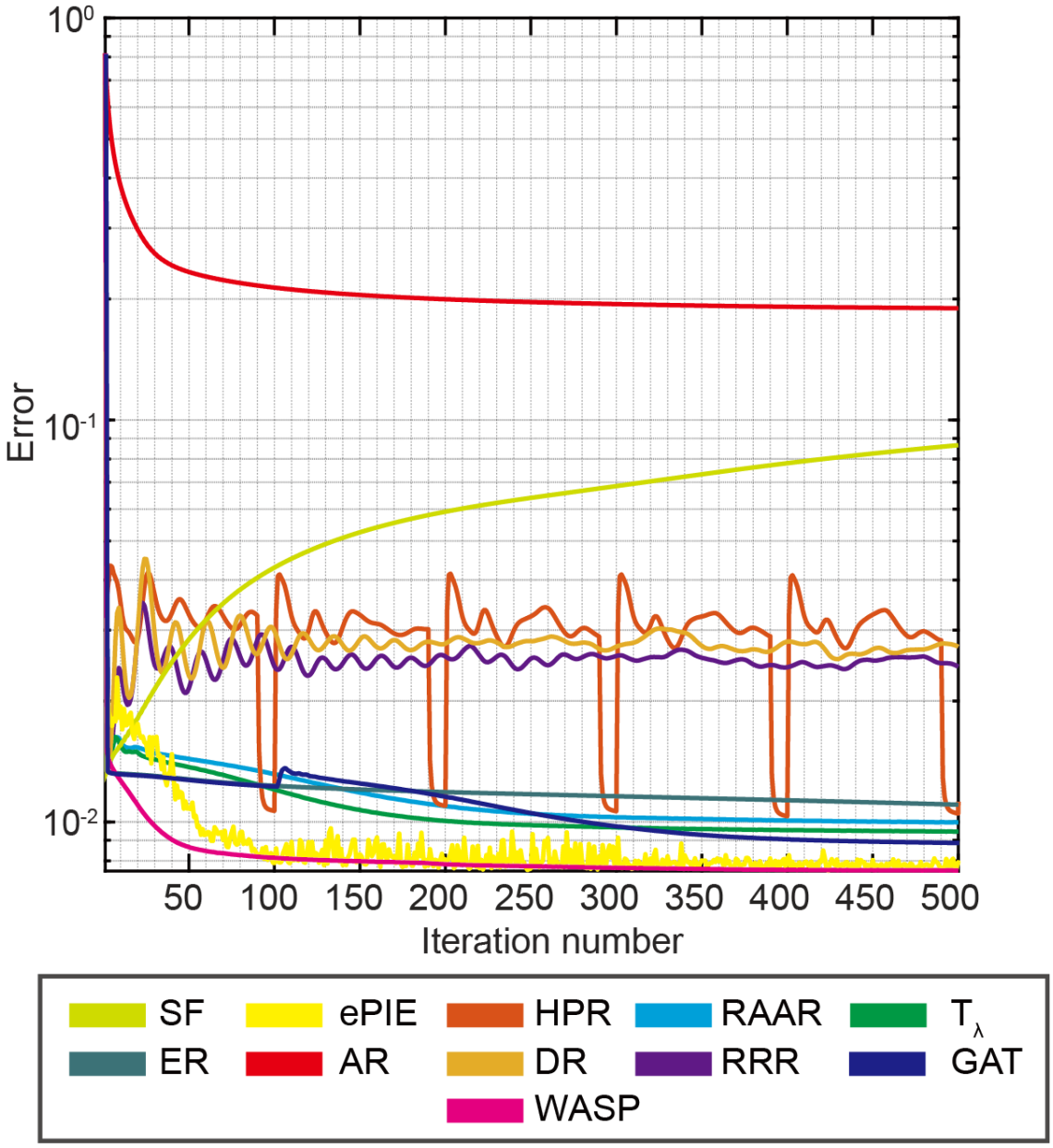

Figure 7.5. The error metric for the Siemens star experiment.

As it shown in Figure 7.4 and Figure 7.5, WASP has the best resolution in the reconstructions, and it converges faster and is more stable than ePIE. GAT had one successful tuning at the first hundred iteration. The tuning results were $a = 0.51, b = 1.70, c = 1.89$, which gives a similar reconstruction as T_λ and RAAR. Although there is a slight phase ramp across Figure 7.4(a) diagonally, WASP was the only one that can roughly see the number of the $0.2\mu m$ resolution target. Also, the reconstruction of WASP has a smoother edge for each spoke

in the Siemens star, while others' edges seem to be brighter than other places in Figure 7.4(b-f).

7.3. Electron Experiment

The final experiment is near-field electron Lorentz ptychography, using a latex sphere sample with a TEM at 300 keV [77]. The phase reconstructions are shown in Figure 7.6, and the error metric is in Figure 7.7. It is the first time that we can see some reconstructed features from SF through all the simulations and real-world experiments tests. The DR type methods have a strong phase ramp at the edges. The tuning of GAT is not acceptable within 500 iterations; therefore, it remains using the parameters for ER, which is good in this case. WASP gives the most successful reconstruction in this test. As a combination of ER and ePIE, it embodies the progress, especially at the edge area. The latex sphere in Figure 7.7(a) are clearly visible, however the edges of the latex sphere in the other methods blend into the background in some places. Moreover, Figure 7.7(b-f) have some dark shadow in the background while the background of WASP is more clear and able to see more detail about the texture.

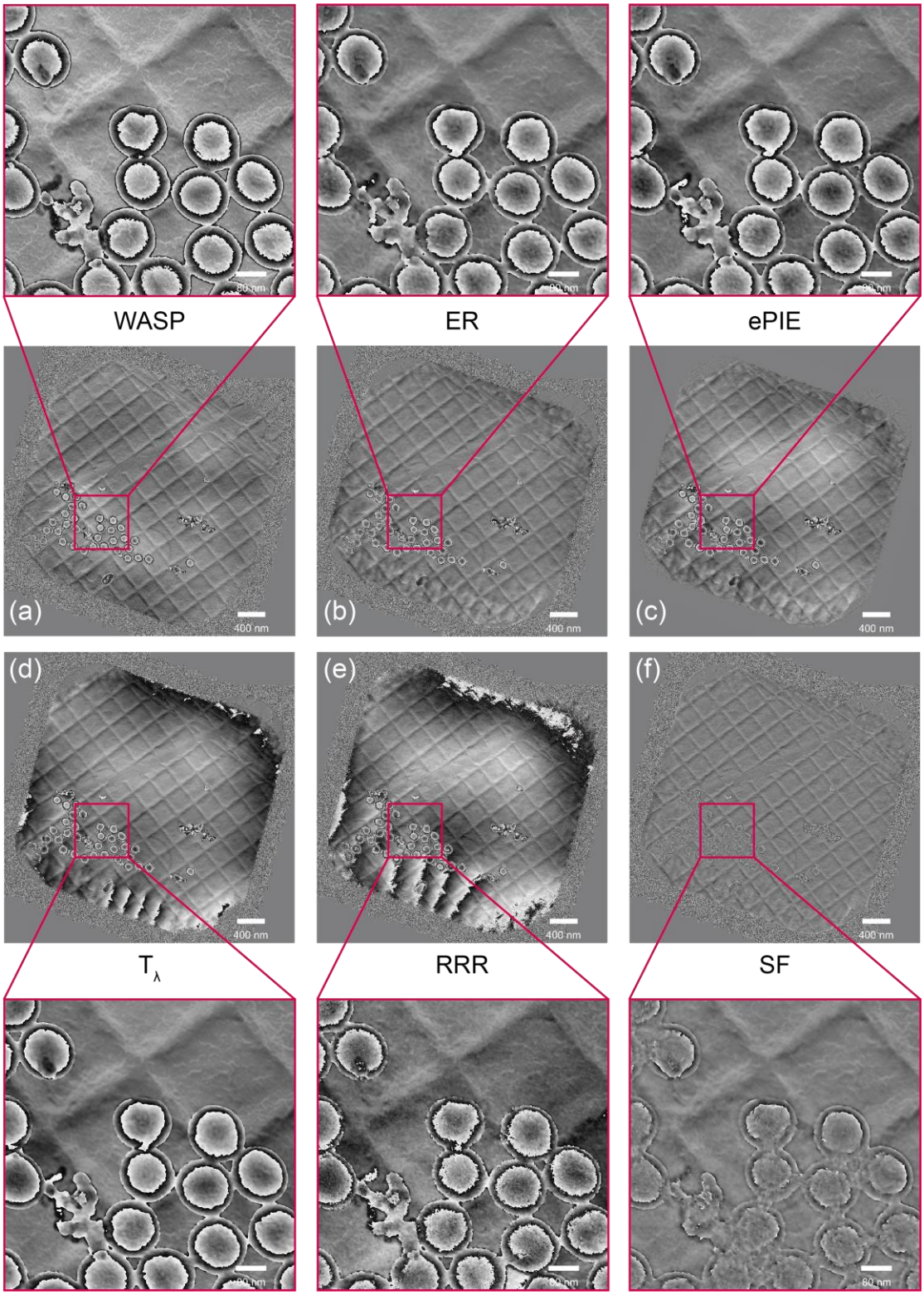

Figure 7.6. The phase reconstruction results of the latex sphere. (a) WASP, (b) ER, (c) ePIE, (d) T_λ, (e) RRR, (e) SF.

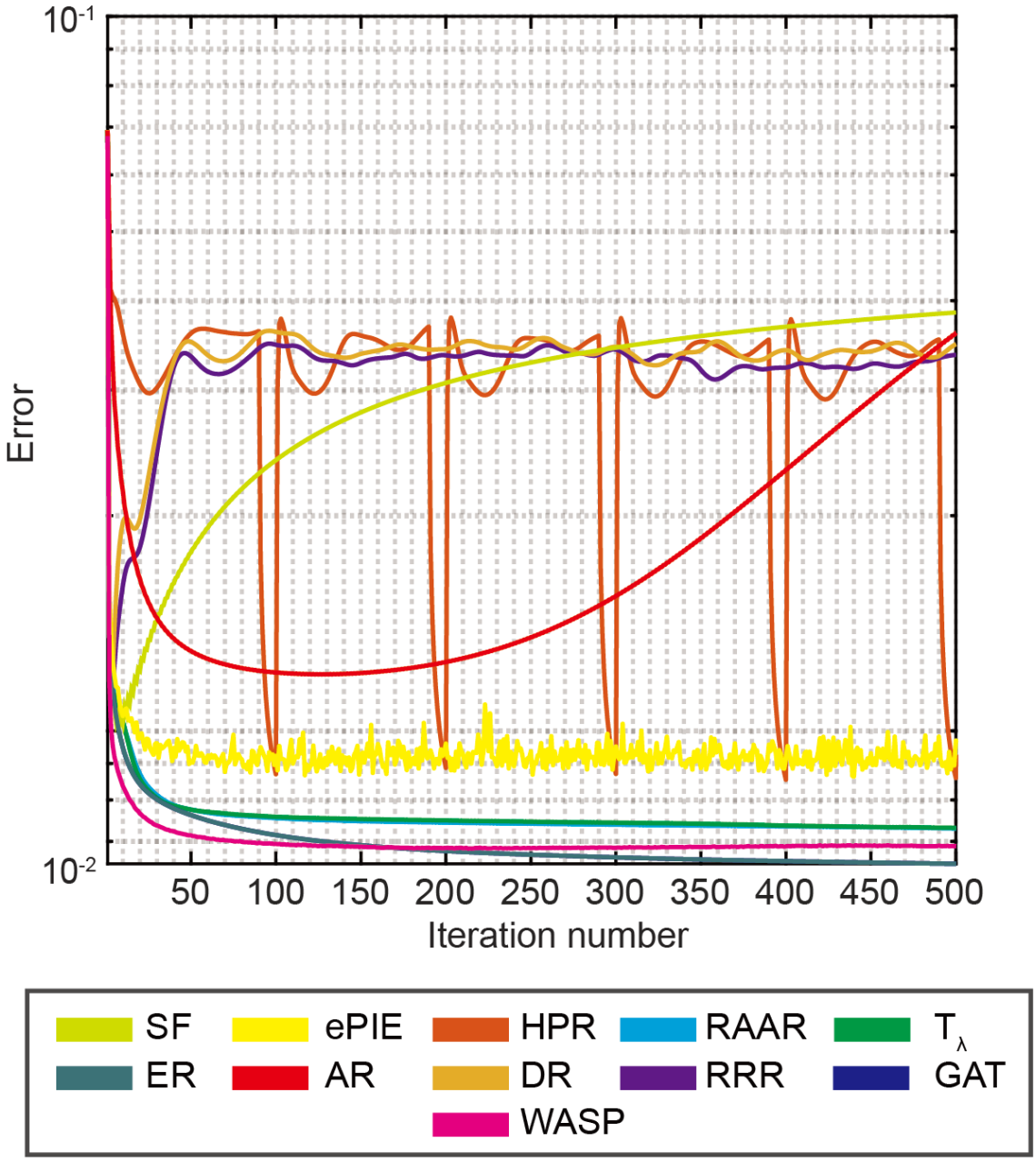

Figure 7.7. The reconstruction results of the latex sphere.

8. Conclusion and Future Work

The development of ptychography brings great potential for solving the phase problem in microscope imaging. As a computational solution to the phase problem, there are many different categories of algorithms for ptychography. This thesis explored and tested these different phase retrieve algorithms with different simulations and real experiments data.

The first one is a direct ptychographic solution called Wigner Distribution Deconvolution (WDD), described in Chapter 4. WDD is a closed and linear solution to the phase problem. However, it requires a dense scan which results in a massive 4D intensity dataset. WDD normally requires the prior knowledge about the probe, and the reconstruction is limited by the cut-off frequency in the 4D dataset. In this thesis, we introduced the “project strategy” [33] which based on the “stepping out” [32] to break the cut-off limitation. Furthermore, we proposed a new iterative way to solve the probe function via the “project strategy” on the opposite direction. The probe solution is feasible in our simulation; however, its performance is not good as expected, compared to iterative method (ePIE). One reason is the value of the small constant in the Wiener filter used in the blind deconvolution. In the future, it is worth to implement a varying Wiener filter during the reconstruction to discover the optimum of the small constant.

The second category is set projection algorithms in Chapter 5. We demonstrated the principles of set projections in detail using a simple three-circle problem and explained the relationship between ptychography and set projections. After analysed most existing set projection algorithms for ptychography, we suggested a general projection algorithm with three tuning parameters a, b, c, which can form most existing set projection algorithms in ptychography. Based on this, T_λ and RRR are first time implemented to the

ptychographic problem. Furthermore, we proposed a new approach called generalized auto-tuning algorithm, which is based on Bayesian Optimization. This method can automatically tune the parameters a, b, c during the reconstruction, and generally performs well in all the simulation tests. However, the time and memory cost of the optimization could be very high, depending on the dataset size. A better model to evaluate the influence of these parameters to the projections may provide a more efficient way for the tuning process.

Finally, with idea of the set projection algorithm and the sequential projection algorithm, we proposed a novel ptychographic solution called Weighted Average of Sequential Projections (WASP) in Chapter 6. As a combination of set projection method and sequential projection method, WASP simultaneously incorporates the advantages of both; it has a rapid initial convergence rate, small memory requirement, robustness to poor initial conditions, noise tolerance, the ability to reach a global minimum and is parallelizable for large size data. Also, a parallel version of WASP was investigated with three different aspects. WASP provided a fundamental framework for ptychography, showcasing the potential for expansion into various types of ptychographic problem. From an algorithmic perspective, the future improvement of its performance may be enhanced by optimizing the regularizer.

Bibliography

[1] J. Miao, R. L. Sandberg, and C. Song, "Coherent X-Ray Diffraction Imaging," *IEEE Journal of Selected Topics in Quantum Electronics,* vol. 18, no. 1, pp. 399-410, 2012, doi: 10.1109/JSTQE.2011.2157306.

[2] I. Robinson, J. Clark, and R. Harder, "Materials science in the time domain using Bragg coherent diffraction imaging," *Journal of Optics,* vol. 18, no. 5, p. 054007, 2016/03/14 2016, doi: 10.1088/2040-8978/18/5/054007.

[3] K. A. Nugent, "Coherent methods in the X-ray sciences," *Advances in Physics,* vol. 59, no. 1, pp. 1-99, 2010/01/01 2010, doi: 10.1080/00018730903270926.

[4] J. Fan, J. Zhang, and Z. Liu, "Coherent diffraction imaging of cells at advanced X-ray light sources," *TrAC Trends in Analytical Chemistry,* vol. 171, p. 117492, 2024/02/01/ 2024, doi: https://doi.org/10.1016/j.trac.2023.117492.

[5] M. Nakasako *et al.*, "Methods and application of coherent X-ray diffraction imaging of noncrystalline particles," *Biophysical Reviews,* vol. 12, no. 2, pp. 541-567, 2020/04/01 2020, doi: 10.1007/s12551-020-00690-9.

[6] H. Faulkner and J. Rodenburg, "Moveable aperture lensless microscopy: a novel phase retrieval algorithm," in *CONFERENCE SERIES-INSTITUTE OF PHYSICS*, 2004, vol. 179: Philadelphia; Institute of Physics; 1999, pp. 337-340.

[7] J. Rodenburg and A. Maiden, "Ptychography," in *Springer Handbook of Microscopy*, (Springer Handbooks, 2019, ch. Chapter 17, pp. 819-904.

[8] J. M. Rodenburg, "Ptychography and Related Diffractive Imaging Methods," (Advances in Imaging and Electron Physics, 2008, pp. 87-184.

[9] P. D. Nellist, B. C. McCallum, and J. M. Rodenburg, "Resolution beyond the 'information limit' in transmission electron microscopy," *Nature,* vol.

374, no. 6523, pp. 630-632, 1995/04/01 1995, doi: 10.1038/374630a0.

[10] R. Bates and J. Rodenburg, "Sub-Ångström transmission microscopy: a Fourier transform algorithm for microdiffraction plane intensity information," *Ultramicroscopy,* vol. 31, no. 3, pp. 303-307, 1989.

[11] M. Guizar-Sicairos and J. R. Fienup, "Phase retrieval with transverse translation diversity: a nonlinear optimization approach," *Optics Express,* vol. 16, no. 10, pp. 7264-7278, 2008/05/12 2008, doi: 10.1364/OE.16.007264.

[12] V. Elser, I. Rankenburg, and P. Thibault, "Searching with iterated maps," *Proceedings of the National Academy of Sciences,* vol. 104, no. 2, pp. 418-423, 2007, doi: 10.1073/pnas.0606359104.

[13] D. R. Luke, "Relaxed averaged alternating reflections for diffraction imaging," *Inverse problems,* vol. 21, no. 1, p. 37, 2004.

[14] H. Yan, "Ptychographic phase retrieval by proximal algorithms," *New Journal of Physics,* vol. 22, no. 2, p. 023035, 2020.

[15] H. Chang, P. Enfedaque, and S. Marchesini, "Blind Ptychographic Phase Retrieval via Convergent Alternating Direction Method of Multipliers," *SIAM Journal on Imaging Sciences,* vol. 12, no. 1, pp. 153-185, 2019, doi: 10.1137/18m1188446.

[16] M. Odstrčil, A. Menzel, and M. Guizar-Sicairos, "Iterative least-squares solver for generalized maximum-likelihood ptychography," *Optics express,* vol. 26, no. 3, pp. 3108-3123, 2018.

[17] A. M. Maiden and J. M. Rodenburg, "An improved ptychographical phase retrieval algorithm for diffractive imaging," *Ultramicroscopy,* vol. 109, no. 10, pp. 1256-62, Sep 2009, doi: 10.1016/j.ultramic.2009.05.012.

[18] A. Maiden, D. Johnson, and P. Li, "Further improvements to the ptychographical iterative engine," *Optica,* vol. 4, no. 7, 2017, doi: 10.1364/optica.4.000736.

[19] M. Sushmasusik and S. Hayath, "History of Microscopes," *Indian Journal of Mednodent and Allied Sciences,* vol. 3, no. 3, pp. 170-179, 2015.

[20] S. Bradbury, *The evolution of the microscope*. Elsevier, 2014.

[21] J. C. Spence, *High-resolution electron microscopy*. OUP Oxford, 2013.

[22] F. Haguenau, P. W. Hawkes, J. L. Hutchison, B. Satiat-Jeunemaitre, G. T. Simon, and D. B. Williams, "Key events in the history of electron microscopy," *Microsc Microanal,* vol. 9, no. 2, pp. 96-138, Apr 2003, doi: 10.1017/S1431927603030113.

[23] F. Merchant and K. Castleman, *Microscope image processing*. Academic press, 2022.

[24] E. Abbe, "Contributions to the theory of the microscope and microscopic perception," *Arch. Microsc. Anat,* vol. 9, pp. 413-468, 1873.

[25] E. McLeod and A. Ozcan, "Microscopy without lenses," *Physics Today,* vol. 70, no. 9, pp. 50-56, 2017, doi: 10.1063/pt.3.3693.

[26] J. W. Goodman, *Introduction to Fourier optics*. Roberts and Company publishers, 2005.

[27] M. Born and E. Wolf, *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier, 2013.

[28] D. G. Voelz, *Computational Fourier optics: a MATLAB tutorial*. Bellingham, Wash. (1000 20th St. Bellingham WA 98225-6705 USA): Bellingham, Wash. 1000 20th St. Bellingham WA 98225-6705 USA : SPIE, c2011, 2011.

[29] E. Brigham, "The Fast Fourier Transform and its Applications, Prentice Hall, Upper Saddle River, NJ," 1988.

[30] S. Marchesini, "A Unified Evaluation of Iterative Transform Technique for Phase Retrieval," Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2004.

[31] J. R. Fienup, "Phase retrieval algorithms: a comparison," *Applied optics,* vol. 21, no. 15, pp. 2758-2769, 1982.

[32] J. Rodenburg and R. Bates, "The theory of super-resolution electron microscopy via Wigner-distribution deconvolution," *Philosophical Transactions of the Royal Society of London. Series A: Physical and*

Engineering Sciences, vol. 339, no. 1655, pp. 521-553, 1992.

[33] P. Li, T. B. Edo, and J. M. Rodenburg, "Ptychographic inversion via Wigner distribution deconvolution: noise suppression and probe design," *Ultramicroscopy,* vol. 147, pp. 106-13, Dec 2014, doi: 10.1016/j.ultramic.2014.07.004.

[34] H. Yang *et al.*, "Electron ptychographic phase imaging of light elements in crystalline materials using Wigner distribution deconvolution," *Ultramicroscopy,* vol. 180, pp. 173-179, 2017.

[35] L. Clark *et al.*, "The Effect of Dynamical Scattering on Single-plane Phase Retrieval in Electron Ptychography," *Microscopy and Microanalysis,* vol. 29, no. 1, pp. 384-394, 2023.

[36] A. Bangun, P. F. Baumeister, A. Clausen, D. Weber, and R. E. Dunin-Borkowski, "Wigner Distribution Deconvolution Adaptation for Live Ptychography Reconstruction," *Microscopy and Microanalysis,* vol. 29, no. 3, pp. 994-1008, 2023.

[37] V. Elser, "Phase retrieval by iterated projections," *JOSA A,* vol. 20, no. 1, pp. 40-55, 2003.

[38] P. Thibault, M. Dierolf, A. Menzel, O. Bunk, C. David, and F. Pfeiffer, "High-resolution scanning x-ray diffraction microscopy," *Science,* vol. 321, no. 5887, pp. 379-382, 2008.

[39] P. Thibault, M. Dierolf, O. Bunk, A. Menzel, and F. Pfeiffer, "Probe retrieval in ptychographic coherent diffractive imaging," *Ultramicroscopy,* vol. 109, no. 4, pp. 338-343, 2009.

[40] M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup, "Efficient subpixel image registration algorithms," *Optics letters,* vol. 33, no. 2, pp. 156-158, 2008.

[41] B. a. R. McCallum, JM, "Two-dimensional demonstration of Wigner phase-retrieval microscopy in the STEM configuration," *Ultramicroscopy,* vol. 45, pp. 371-380, 1992.

[42] T. Bendory, R. Beinert, and Y. C. Eldar, "Fourier Phase Retrieval:

Uniqueness and Algorithms," *arXiv.org,* 2017, doi: 10.48550/arxiv.1705.09590.

[43] B. C. McCallum and J. M. Rodenburg, "Simultaneous reconstruction of object and aperture functions from multiple far-field intensity measurements," *J. Opt. Soc. Am. A,* vol. 10, no. 2, pp. 231-239, 1993/02/01 1993, doi: 10.1364/JOSAA.10.000231.

[44] J. Kalbfleisch and R. Stanton, "On the maximal triangle-free edge-chromatic graphs in three colors," *Journal of Combinatorial Theory,* vol. 5, no. 1, pp. 9-20, 1968.

[45] D. Sherrington and S. Kirkpatrick, "Solvable model of a spin-glass," *Physical review letters,* vol. 35, no. 26, p. 1792, 1975.

[46] S. Gravel and V. Elser, "Divide and concur: a general approach to constraint satisfaction," *Phys Rev E Stat Nonlin Soft Matter Phys,* vol. 78, no. 3 Pt 2, p. 036706, Sep 2008, doi: 10.1103/PhysRevE.78.036706.

[47] R. W. Gerchberg, "A practical algorithm for the determination of plane from image and diffraction pictures," *Optik,* vol. 35, no. 2, pp. 237-246, 1972.

[48] J. Douglas and H. H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," *Transactions of the American mathematical Society,* vol. 82, no. 2, pp. 421-439, 1956.

[49] H. H. Bauschke, P. L. Combettes, and D. R. Luke, "Hybrid projection–reflection method for phase retrieval," *J. Opt. Soc. Am. A,* vol. 20, no. 6, pp. 1025-1034, 2003/06/01 2003, doi: 10.1364/JOSAA.20.001025.

[50] V. Elser, "The Complexity of Bit Retrieval," *IEEE Transactions on Information Theory,* vol. 64, no. 1, pp. 412-428, 2018, doi: 10.1109/tit.2017.2754485.

[51] N. H. Thao, "A convergent relaxation of the Douglas–Rachford algorithm," *Computational Optimization and Applications,* vol. 70, no. 3, pp. 841-863, 2018.

[52] M. A. Gelbart, J. Snoek, and R. P. Adams, "Bayesian optimization with

unknown constraints," presented at the Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, 2014.

[53] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. Springer, 2006.

[54] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in neural information processing systems,* vol. 25, 2012.

[55] R. P. Adams, "A tutorial on Bayesian optimization for machine learning," *Harvard University,* 2014.

[56] P. I. Frazier, "A tutorial on Bayesian optimization," *arXiv preprint arXiv:1807.02811,* 2018.

[57] E. J. Candes, Y. C. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM review,* vol. 57, no. 2, pp. 225-251, 2015.

[58] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and trends® in Optimization,* vol. 1, no. 3, pp. 127-239, 2014.

[59] S. McDermott and A. Maiden, "Near-field ptychographic microscope for quantitative phase imaging," *Optics express,* vol. 26, no. 19, pp. 25471-25480, 2018.

[60] M. C. Cao, Z. Chen, Y. Jiang, and Y. Han, "Automatic parameter selection for electron ptychography via Bayesian optimization," *Scientific Reports,* vol. 12, no. 1, p. 12284, 2022/07/19 2022, doi: 10.1038/s41598-022-16041-5.

[61] G. Harauz and M. van Heel, "Exact filters for general geometry three dimensional reconstruction," *Optik.,* vol. 73, no. 4, pp. 146-156, 1986.

[62] A. Maiden, M. Humphry, M. Sarahan, B. Kraus, and J. Rodenburg, "An annealing algorithm to correct positioning errors in ptychography," *Ultramicroscopy,* vol. 120, pp. 64-72, 2012.

[63] S. Marchesini, A. Schirotzek, C. Yang, H.-t. Wu, and F. Maia, "Augmented projections for ptychographic imaging," *Inverse Problems,* vol. 29, no. 11, 2013, doi: 10.1088/0266-5611/29/11/115009.

[64] C. Wang, Z. Xu, H. Liu, Y. Wang, J. Wang, and R. Tai, "Background noise removal in x-ray ptychography," *Applied Optics,* vol. 56, no. 8, pp. 2099-2111, 2017/03/10 2017, doi: 10.1364/AO.56.002099.

[65] R. Claveau, P. Manescu, D. Fernandez-Reyes, and M. Shaw, "Structure-dependent amplification for denoising and backgroundcorrection in Fourier ptychographic microscopy," *Optics Express,* vol. 28, no. 24, pp. 35438-35453, 2020/11/23 2020, doi: 10.1364/OE.403780.

[66] L. Hou, H. Wang, J. Wang, and M. Xu, "Background-noise Reduction for Fourier Ptychographic Microscopy Based on an Improved Thresholding Method," *Curr. Opt. Photon.,* vol. 2, no. 2, pp. 165-171, 2018/04/25 2018. [Online]. Available: https://opg.optica.org/copp/abstract.cfm?URI=copp-2-2-165.

[67] S. Gao *et al.*, "Electron ptychographic microscopy for three-dimensional imaging," *Nature Communications,* vol. 8, no. 1, p. 163, 2017/07/31 2017, doi: 10.1038/s41467-017-00150-1.

[68] Z. Ding *et al.*, "Three-dimensional electron ptychography of organic–inorganic hybrid nanostructures," *Nature Communications,* vol. 13, no. 1, p. 4787, 2022/08/15 2022, doi: 10.1038/s41467-022-32548-x.

[69] Z. Hu, Y. Zhang, P. Li, D. Batey, and A. Maiden, "Near-field multi-slice ptychography: quantitative phase imaging of optically thick samples with visible light and X-rays," *Optics Express,* vol. 31, no. 10, pp. 15791-15809, 2023/05/08 2023, doi: 10.1364/OE.487002.

[70] A. M. Maiden, M. J. Humphry, and J. M. Rodenburg, "Ptychographic transmission microscopy in three dimensions using a multi-slice approach," *JOSA A,* vol. 29, no. 8, pp. 1606-1614, 2012.

[71] M. Kahnt *et al.*, "Multi-slice ptychography enables high-resolution measurements in extended chemical reactors," *Scientific Reports,* vol. 11, no. 1, p. 1500, 2021/01/15 2021, doi: 10.1038/s41598-020-80926-6.

[72] P. Thibault and A. Menzel, "Reconstructing state mixtures from diffraction measurements," *Nature,* vol. 494, no. 7435, pp. 68-71, Feb 7 2013, doi:

10.1038/nature11806.

[73] X. Shi, N. Burdet, D. Batey, and I. Robinson, "Multi-Modal Ptychography: Recent Developments and Applications," *Applied Sciences,* vol. 8, no. 7, p. 1054, 2018. [Online]. Available: https://www.mdpi.com/2076-3417/8/7/1054.

[74] P. Li, T. Edo, D. Batey, J. Rodenburg, and A. Maiden, "Breaking ambiguities in mixed state ptychography," *Optics express,* vol. 24, no. 8, pp. 9038-9052, 2016.

[75] S. Cao, P. Kok, P. Li, A. Maiden, and J. Rodenburg, "Modal decomposition of a propagating matter wave via electron ptychography," *Physical Review A,* vol. 94, no. 6, p. 063621, 2016.

[76] A. M. Maiden, J. M. Rodenburg, and M. J. Humphry, "Optical ptychography: a practical implementation with useful resolution," *Optics Letters,* vol. 35, no. 15, pp. 2585-2587, 2010/08/01 2010, doi: 10.1364/OL.35.002585.

[77] S. You, P.-H. Lu, T. Schachinger, A. Kovács, R. E. Dunin-Borkowski, and A. M. Maiden, "Lorentz near-field electron ptychography," *Applied Physics Letters,* vol. 123, no. 19, 2023, doi: 10.1063/5.0169788.