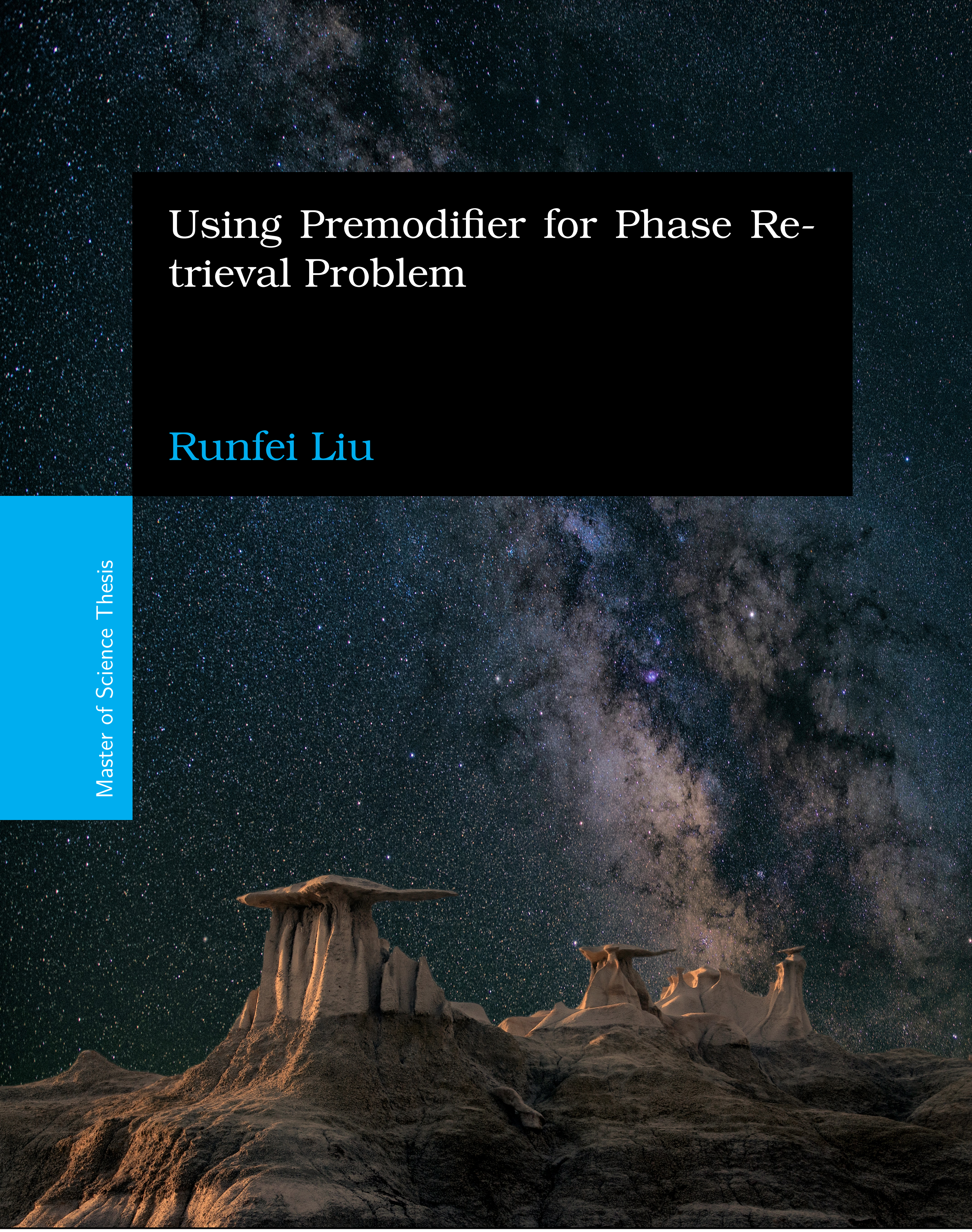


Using Premodifier for Phase Retrieval Problem

Runfei Liu

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Using Premodifier for Phase Retrieval Problem

Master of Science Thesis

Runfei Liu

September 8, 2021

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Abstract

Phase Retrieval Problem is very common and has a lot of applications in fields such as microscope, astronomy, crystallography, optical imaging etc.. In this problem, PSF and the pupil function are given, and the goal is to reconstruct the phase of the object. For camera which has 8 bits color depth, the maximum number of colors that can be displayed at any one time is 2^8, i.e. 256. When the minimal exposure time of the camera is longer than "correct" exposure time or the intensity is changing, e.g. dynamic focusing, too much intensity is measured by the camera and causes saturation, i.e. over-exposure. Over-exposure causes distortion of the image, thereby losing the information. However, on the other hand, the overall brightness of the PSF image will be enhanced due to over-exposure. It highlight structure and increase the number of informative pixels of the image. We can use this to enhance the information in the PSF image to offset the negative effects of noise or background in the image.

Gerchberg-Saxton algorithm is a solution to phase retrieval problem and is a nice example of projection based algorithm. The algorithm has a perfect performance in the case that the PSF is the modules square of the object, i.e. without noise or other preconditioners effect. It is demonstrated in [Nishizaki, 2019] that an over-exposure preconditioner can improve the performance for phase restoration of a machine-learning model based algorithm. In this master thesis, we propose modification of the model of PSF amplitude in Gerchbrg-Saxton algorithm to make the algorithm be able to cope with over-exposed PSF image. This approach is tested both by numerical simulation and with experimental data.

Table of Contents

1 Introduction **1**
1-1 Phase Retrieval Problem . . . 2
1-2 Gerchberg-Saxton Algorithm . . . 3
1-2-1 Description of the Algorithm Process . . . 3
1-2-2 Converge to a Wrong Result . . . 5
1-3 Factors Affecting PSF Image Quality . . . 7
1-3-1 Noise: Theoretical Background . . . 7
1-3-2 Limitation of Camera's Color Depth . . . 8

2 Proposal of Modification on Gerchberg-Saxton Algorithm **11**
2-1 Premodifier: Theoretical Background . . . 11
2-2 Over-exposure as Premodifier . . . 12
2-3 Adjustment of Simple Gerchberg-Saxton Algorithm . . . 12
2-3-1 Modification on the constraints on Fourier Domain . . . 13
2-3-2 Modification to Remove Background . . . 16
2-4 A Modified Gerchberg-Saxton Algorithm . . . 18

3 Numerical Simulation Experiment for Testing Modified GS Algorithm **21**
3-1 Forward Model for PSF Generation . . . 21
3-1-1 Original PSF Images Generation . . . 21
3-1-2 Over-exposure Simulation . . . 23
3-1-3 Noise Simulation . . . 24
3-1-4 Simulation of Color Depth of Camera . . . 26
3-2 Results and Analysis of the Numerical Simulation . . . 27
3-2-1 Robustness against Noise . . . 30
3-2-2 Over-exposure Premodifier . . . 31
3-2-3 Comparison between Modified GS and Simple GS . . . 34

4 Apply Modified GS Algorithm on Real Data 37

4-1 Experimental Data Source 37

4-2 Aperture Modelling 39

4-3 Results and Analysis of the Experiment on Real Data 40

4-3-1 Case 1: Original PSF is Unknown 41

4-3-2 Case 2: Original PSF is Known 42

5 Conclusions and Future Work 49

A Appendix:Fundamental Concepts 51

A-1 An approach for Parameter γ calculation 51

A-2 Gaussian Noise 52

A-3 Zernike Polynomials 53

A-4 Signal-to-noise Ratio (SNR) 54

A-5 Specification of IDS UI-1490 Camera 55

A-6 Centroid Algorithm 57

Bibliography 59

Chapter 1

Introduction

Turbulent atmosphere can cause aberrations, which change the shape of plane wavefront. The optical imaging camera suffers from aberrated wavefront as it distort the point spread function(PSF) images. PSF is the proportional modulus squared of the Fourier transform of a function. Recovering the function from PSF is not only the solution of characterizing aberrations, but also receive the information of the object. Similarly, in many applications of science and engineering, such as microscope, astronomy, crystallography, optical imaging, one is given the modulus squared of the Fourier transform of an unknown signal and then tasked with solving the corresponding inverse problem, known as phase retrieval problem.[1]

Gerchberg-Saxton algorithm (GS algorithm)[2][3] is a nice application of alternating projection method for solving phase retrieval problem. Alternating projection is a method to compute a point in the intersection of two closed convex sets while knowing how to project onto these sets separately.[4] . Gerchberg-Saxton algorithm alternates between Fourier plane and real-space plane by using Fourier transform and the inverse transform, and promotes convergence by imposing constraints. Technically, simple GS algorithm has good performance on PSF images of point source. However, when the PSF image suffers from high level noise or influenced by preconditioners such as defocus and over-exposure, the algorithm is more likely to get in a stagnation or converge to a wrong retrieval.

In practice, it is inevitable that noise will have an influence when taking the image of object. High-level noise can obscure the information of the image. Classical denoise approaches such as noise filtering and apodisation can reduce the intensity of noise. On the other hand, some methods, e.g. enhance exposure, can reinforce the intensity of the image and highlight the structure, thus increases the image's information. When using camera which has 8 bits color depth, the maximum number of colors that can be displayed at any one time is 2^8, i.e. 256. When the minimal exposure time of the camera is longer than "correct" exposure time or the intensity is changing, too much intensity is measured by the camera and causes saturation, i.e. over-exposure. Over-exposure leads to uncompleted information, since the intensity that exceeds the limit of the camera is displayed as uniformly white in image.

Simple GS algorithm cannot retrieve phase perfectly from the over-exposed PSF, but neural networks can. Nishizaki[5] used training data-set that randomly generated by first 32 Zernike modes with coefficients range in $[-0.5, 0.5]$ to train a Convolution Neural Network (CNN) and made the network restore phase from aberrated PSF without preconditioner and over-exposed PSF with aberration, respectively. The experiment results showed that the CNN can retrieve phase from over-exposed PSF and also demonstrated that over-exposure proconditioner can improve the performance of the network, as it increases the number of informative pixels. However, the disadvantage of using neural network method to solve phase retrieval problem is the performance of the network closely depends on the content of the training data-set. It is unlike as GS algorithm, which can deal with high quality point source's PSF images without additional training.

Comparing characteristics of Gerchberg-Saxton algorithm and CNN methods motives the research question: Is it possible to modify the constraints in the iterative process of GS algorithm so that it can handle the overexposed PSF with better results? Is overexposure preconditioner could also improve the performance and the robustness against noise of GS algorithm?

This thesis will conduct a theoretical analysis of this problem, and verify the idea through the experimental results of numerical simulations and actual lab data.

1-1 Phase Retrieval Problem

The study of phase retrieval problem can be tracked back to 60 years ago. At the beginning, researchers tried to solve the problem on one dimension. In 1950, Akutowicz[6][7] showed that the answer to one dimension phase retrieval problem without any prior constraints lacks uniqueness. A simple example is if the modules is vector of all ones, then the phase could be arbitrary if it multiply a standard basic element. A breakthrough was at the end of the 1970s, Bruck and Sodin[8] pointed out that the fundamental barrier to unconstrained 1D phase retrieval does not apply in higher dimensions. It confirmed nicely with the success of simple Gerchberg-Saxton (GS) algorithm[2] and Hybrid-Input-Output(HIO) algorithm[9]. In 1982, Gonsalves[10][11] proposed phase diversity method for phase retrieval problem. Using a second image with an additional known phase variation with respect to the first image, such as defocus, it is possible to estimate the phase even when the object is extended, which is not for GS algorithm. Recent decades, more works are about applying machine-learning theory to this field. Some papers[12][13][5] show that Deep neural network and convolutional neural network succeed on this issue.

In many science and engineering, we only have access to magnitude measurements. For instance, optical detection devices cannot measure the phase of a light wave, but the photon flux, which is proportional to the magnitude squared of the field, like Fig.1-1[14]. The measurements we receive, e.g. PSF image, are the modulus squared of the Fourier transform of the unknown signal $\boldsymbol{x} \in \mathbb{C}^n$ such as

$$p_k = \left|\langle \boldsymbol{a}_k, \boldsymbol{x} \rangle\right|^2, k = 1, ..., K. \tag{1-1}$$

Suppose that $y_k = \sqrt{p_k}$, then a simple statement of the most general form of the phase retrieval problem is:

$$\begin{aligned} \text{given} \quad & y_k = \left|\langle \boldsymbol{a}_k, \boldsymbol{x} \rangle\right|, \quad k = 1, ..., K. \\ \text{find} \quad & \boldsymbol{x} \in C \subset \mathbb{C}^n. \end{aligned} \tag{1-2}$$

The phase retrieval problem is recovering the missing phase of the data $\langle \boldsymbol{a}_k, \boldsymbol{x} \rangle$. Here $\boldsymbol{x} \in \mathbb{C}^n$ is the target phase and the mapping $\boldsymbol{a_k} \in \mathbb{C}^n$ for all k denotes Fourier transform of some kind, which includes the Fresnel transform and some defocused or otherwise imperfect Fraunhofer transform. The set C is some a prior constraint, e.g. support or non-negativity.

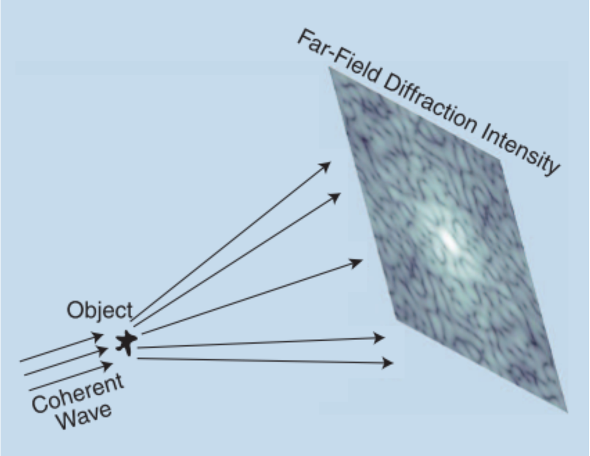


Figure 1-1: Basic coherent diffractive imaging setup. When the object is small and the intensity is measured far away, the far-field intensity pattern corresponding to the modulus of the Fourier transform of the object. When the Fresnel number is small, the measured intensity is proportional to the magnitude squared of the Fourier transform. [14]

1-2 Gerchberg-Saxton Algorithm

Gerchberg-Saxton(GS) algorithm can recover the complex phase image from magnitude measurements at two different planes, i.e. the real plane and the Fourier plane. Suppose the measured PSF data is represented as $p(x)$, and Eq.(1-2) could be written in the form

$$p(x) = |H(x)|^2. \tag{1-3}$$

Then we obtain the Fourier magnitude $|H(x)|$ by getting the square root of $p(x)$. The real space magnitude $|h(k)|$ is the amplitude of the aperture, and is related with the phase ϕ as

$$h(k) = |h(k)|e^{i\phi(k)}, \tag{1-4}$$

$h(k)$ and $H(k)$ are a pair of Fourier transform. GS algorithm is using known aperture amplitude $|h(x)|$ and Fourier magnitude $|H(k)|$ to estimate the unknown phase ϕ via several times iteration. The process of the algorithm is presented as the Algorithm1.

1-2-1 Description of the Algorithm Process

The simple Gerchberg-Saxton algorithm is a method that converges to the result through multiple iterations. The iteration starts from an initial point, which is usually a guessed one with random phase. When the error is under the threshold, the iteration stop. Or we can directly set the iteration time and the algorithm will end when all iterations finish.

```
Algorithm 1 Gerchberg-Saxton algorithm[14]
Input: $|h(x)|$ - real space magnitude,
    $|H(k)|$ - Fourier magnitude,
    $\epsilon$ - the error threshold
Output: $f(k)$ - a matrix that conforms with both magnitude constraints, i.e., $|f(x)| = |h(x)|$
    and $|F(k)| = |H(k)|$, where $f(x)$ and $F(k)$ are a Fourier pair.
Initialization: $f_0(x) = |h(x)| \exp[\Phi(x)]$, with a random $\Phi(x) \in [-\pi, \pi]$.
 General Step (i=1,2,...):
  1: Fourier transform $f_i(x)$ to obtain $F_i(k)$
  2: Use the measured Fourier modulus to replace the calculated Fourier transform modulus
     to form a Fourier transform estimate, i.e. $F_i^{'}(k) = |H(k)| \cdot \frac{F_i(k)}{|F_i(k)|}|$
  3: Inverse Fourier transform the estimate Fourier transform $F_i^{'}(k)$ to gain $f_i^{'}(x)$
  4: Replace the modulus of the resulting computed image with the measured object modulus
     to form a new estimate of the object, i.e. $f_{i+1}(x) = |h(x)| \cdot \frac{f_i^{'}(x)}{|f_i^{'}(x)|}$
  5: Go to 1
Until $E_i = \sum_k ||F_i(k)| - |H(k)||^2 \leq \epsilon$
```

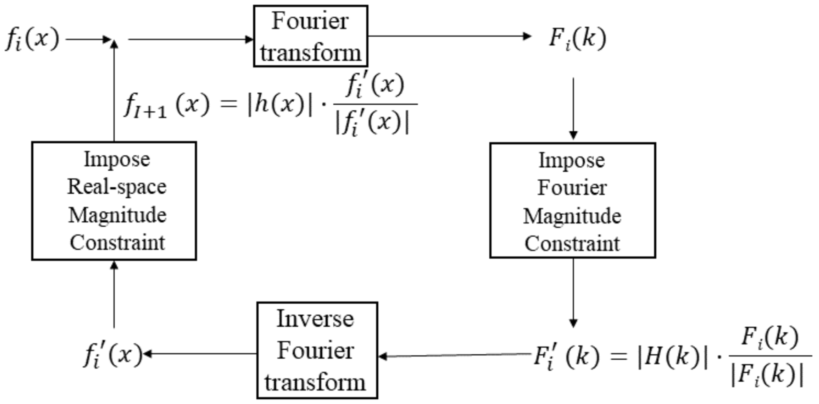

$f_i(x)$
Fourier transform
$F_i(k)$
$f_{I+1}(x) = |h(x)| \cdot \frac{f_i'(x)}{|f_i'(x)|}$
Impose Real-space Magnitude Constraint
Impose Fourier Magnitude Constraint
$f_i'(x)$
Inverse Fourier transform
$F_i'(k) = |H(k)| \cdot \frac{F_i(k)}{|F_i(k)|}$

Figure 1-2: The block diagrams of the GS algorithm.[14]

Magnitude constraints about two fields and the choice of the initial point play important roles and have directly effects on the algorithm performance. Some modified versions based on GS algorithm change constraints to improve the performance or expand the application of the algorithm. An example is hybrid input-output (HIO)[9] method, which applys a correction to the real-space image estimate:

$$f_{i+1}(x) = \begin{cases} f_i'(x), & x \notin \zeta \\ f_i(x) - \beta f_i'(x), & x \in \zeta \end{cases}$$

with β being a small parameter and ζ being the set of indices for which $f_i'(x)$ violates the real-space constraints. This change on constraints makes the algorithm be less likely to converge to a local minima.

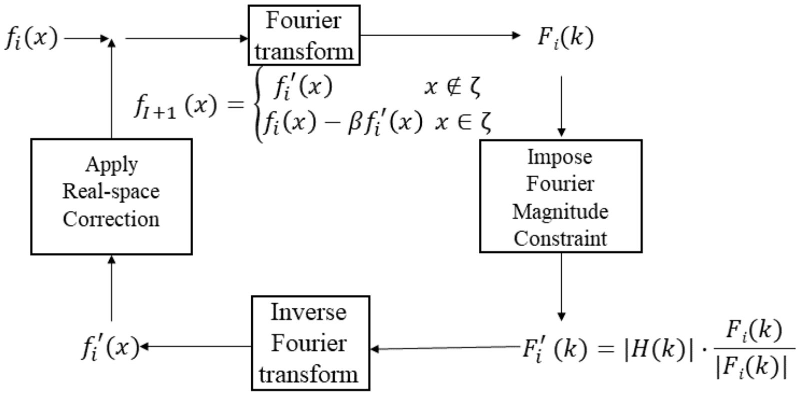

$f_i(x)$
Fourier transform
$F_i(k)$
$f_{I+1}(x) = \begin{cases} f_i'(x) & x \notin \zeta \\ f_i(x) - \beta f_i'(x) & x \in \zeta \end{cases}$
Apply Real-space Correction
Impose Fourier Magnitude Constraint
$f_i'(x)$
Inverse Fourier transform
$F_i'(k) = |H(k)| \cdot \frac{F_i(k)}{|F_i(k)|}$

Figure 1-3: The block diagrams of HIO algorithm.[14]

1-2-2 Converge to a Wrong Result

When the desired solution is complex-valued, GS algorithm is known to suffer from twin-image stagnation. Fig.1-4 is an example. The reason for this stagnation is that the function $h(x, y)$ and its twin $h^*(-x, -y)$(the complex conjugated object rotated by 180 deg) have the same Fourier amplitude $|H(f_x, f_y)|$. When the iterative algorithm starts from an initial guess with random values, then there is an equal probability that the result will be either of these two functions. If feathers of both $h(x, y)$ and $h^*(-x, -y)$ in the result are equally strong, the algorithm may stay at the stagnation. The algorithm tries to reconstruct both together and goes nowhere, as it is unable to suppress one twin image and converge to the other [15].

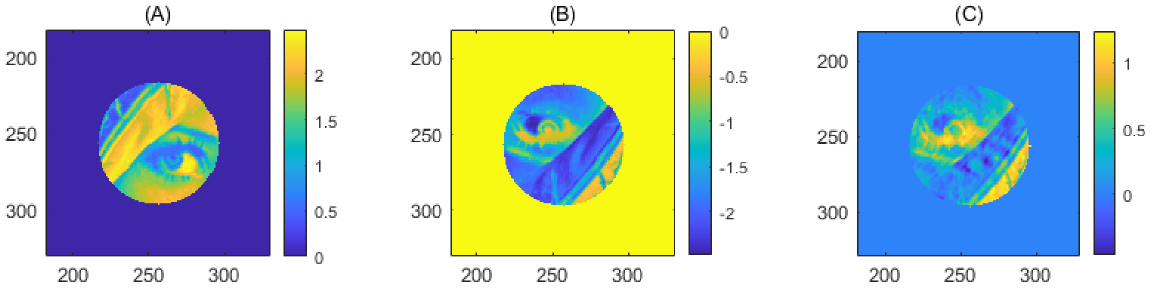


Figure 1-4: Simultaneous twin-images problem. (A) the phase $\phi(x, y)$ of the object $h(x, y)$ of the input field; (B) the phase$-\phi(-x, -y)$ of twin $h^*(-x, -y)$; (C) Output from the iterative transform algorithm that has stagnated with features of both.[15]

When the GS algorithm output a result like (C) in Fig.1-4, the reduced-area support constraint method[15] is a solution, as Algorithm2.

The temporary mask is a mask that covers only a subset of the correct support including at least one of its edges and has no 180° rotational symmetry, as shown in Fig. 1-5. The method does not need a large iteration time, e.g. 10 times is fine, since the purpose of the method is using the temporary mask to increase the difference between the features of the twin-phases. After the iteration with temporary mask, we replace it to the original constraints, i.e. aperture

Algorithm 2 The Reduced-area Support Constraint Method

```
Input: $|t(x)|$ - a temporary mask ,
    $|H(k)|$ - the Fourier magnitude,
    $k$ - the iteration times.
Output: $\Phi(x)$ - half phase without its twin.
Initialization: $f_0(x) = |t(x)|\exp[i\Phi_t(x)]$, with $\Phi_t(x)$ is the phase suffers from twin stagnation .
 General Step (i=1,2,...,k):
  1: Fourier transform $f_i(x)$ to obtain $F_i(k)$
  2: Use the measured Fourier modulus to replace the calculated Fourier transform modulus
     to form a Fourier transform estimate, i.e. $F_i^{'}(k) = |H(k)| \cdot F_i(k)/|F_i(k)|$
  3: Inverse Fourier transform the estimate Fourier transform $F_i^{'}(k)$ to gain $f_i^{'}(x)$
  4: Replace the modulus of the resulting computed image with the temporary mask, i.e.
     $f_{i+1}(x) = |t(x)| \cdot f_i^{'}(x)/|f_i^{'}(x)|$
  5: Go to 1
Until i = k
```

amplitude, and continue the GS algorithm. After further iterations, the GS algorithm finally can output a phase without its twin.

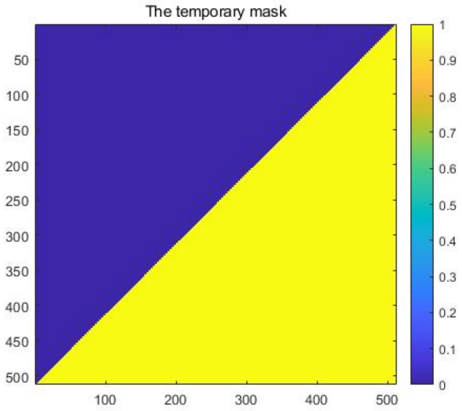


Figure 1-5: The temporary mask.

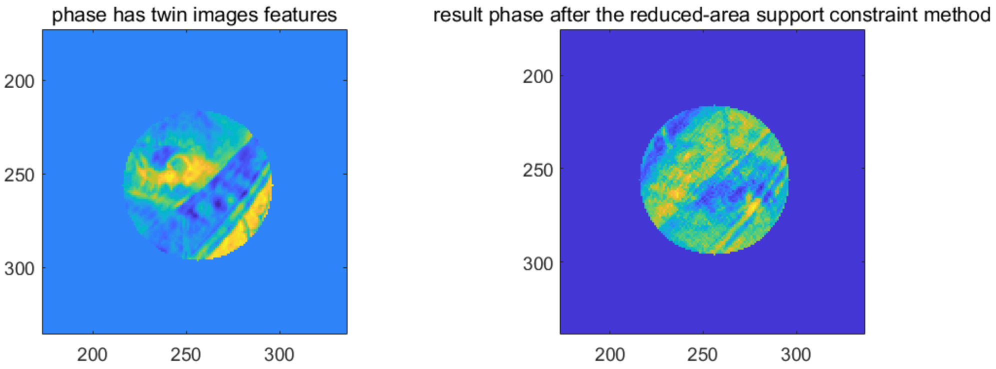


Figure 1-6: The left image is the phase that input to the reduced-area support constraint method which has twin-image features. The right phase is the result after 10 iterations temporary constraints and 3000 times simple GS algorithm iterations.

Fig.(1-6) shows the performance of the reduced-area support constraint method. After using few times temporary mask as the constraints, the simple GS algorithm could converge to the right result.

1-3 Factors Affecting PSF Image Quality

The ideal condition of phase retrieval problem assumes that the PSF image is only the intensity of the object, and it is proportional module's square of the Fourier transform of the wavefront in the pupil plane. But in fact, there is inevitably noise and aberration in the "background" of PSF image. The noise can cause spot in the retrieved phase, or worse, the phase cannot be restored. Besides, the imaging camera cannot accurately record the image of the object. Different colors may appear as the same color in the captured image, as there is limitation of color depth of camera. These effects cause the failure of converging when apply GS algorithm to the measured images. In this section, we are going to discuss the theoretical background of factors that have effects on PSF image quality.

1-3-1 Noise: Theoretical Background

Image noise is the random variation of brightness or color information in the image produced by the sensors and circuits of a scanner or digital camera, and it may also originate from film grain and in the unavoidable shot noise of an ideal photon detector[16]. Noise could be modeled in two kinds, i.e. additive noise and multiplicative noise.

Let $f(\cdot)$ denote an image. We define the desired component of the image as $g(\cdot)$ and the noise component as $n(\cdot)$. For additive noise, it can be formulated as

$$f(\cdot) = g(\cdot) + n(\cdot). \tag{1-5}$$

An example of additive noise is Gaussian noise.

For multiplicative noise, the decomposition of the image is

$$f(\cdot) = g(\cdot)n(\cdot). \tag{1-6}$$

Speckle is a kind of noise that often modeled as multiplicative.

Additive noise does not affect the value of the image, while multiplicative noise may cover and destroy information of the image. A simple example is when $n(\cdot)$ is 0, then $f(\cdot)$ will be 0 and there is no ideal to get value of $g(\cdot)$. Hence, additive noise is easier to deal with, comparing with multiplicative noise. Additive and multiplicative models are used to fit the noise, but not guarantee to fit perfectly. Thus, it is usually impossible to remove all the effects of the noise. But a well-fitted model would be convenient for mitigating the effect of noise.

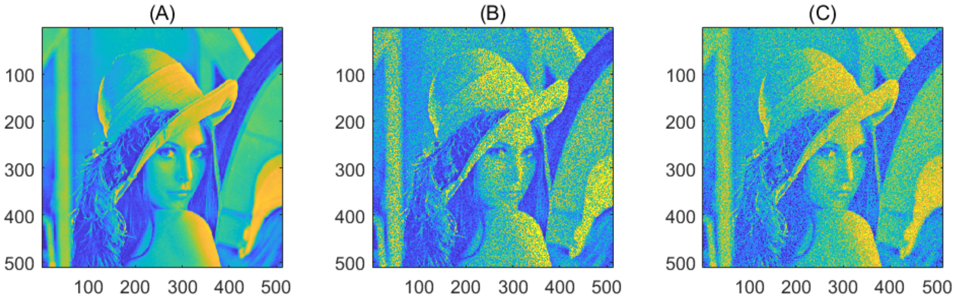


Figure 1-7: The effect of Gaussian noise and speckle noise. (A) is the original figure without noise; (B) is adding speckle noise to (A) and (C) is adding Gaussian noise to (A).

Fig.(1-7) shows the effect of speckle noise and Gaussian noise, respectively. The Gaussian noise is zero-mean Gaussian white noise with a variance of 0.01. For speckle noise, the decomposition of the image is

$$f = g * (1 + \mathbf{n}),$$

where $\mathbf{n}$ is uniformly distributed random noise with a mean of 0 and a variance of 0.05. Both two kinds of noise can impair the quality of the image. But there are some differences between additive noise and multiplicative noise. For example, the light-dark relationship of (C) in Fig.(1-7) has not changed significantly, while the highlight part of (B) is obviously darker than that of (A).

1-3-2 Limitation of Camera's Color Depth

Color depth, also known as bit depth, is either the number of bits used to indicate the color of a single pixel in a bit mapped image, or the number of bits used for each color component of a single pixel. "Bit" is a computer storage unit. Computer uses binary, hence a single bit only have 2 value, 0 and 1, which means that only 2 colors could be stored. Analogously, n bit can express 2^n numbers. Therefore, *8-bit* is 256 possible integer values. If it is a gray-scale image, the color will be expressed as an integer in the range of 0-255 with "0" is black and "255" is white.

The higher bit number the camera has, the more colors can be expressed in the image. Hence the quality of measured image depends on the color depth of camera. When a large range of the similar color appears, lower bit camera is not able to express the smooth color change as the color change is small and the number of color is not enough. Eventually the similar colors in a range will appear in the image as the same color, and cause dividing line between color and color.

Fig.(1-8) shows the simulated PSF results of the same object collected by 2-bit, 4-bit and 8-bit cameras. It is clear that with higher bit number, the changes of colors are more smooth and delicate. Lots of details are lost in low bit number PSF result, and the color layering is also obvious.

If we focus on the brightest part of the PSF, i.e. the central yellow part and compare 2-bit and 8-bit PSFs, it is apparent that there are at least 2 kinds of yellow colors actually but they are presented in the same way in 2-bit PSF. Image distortion occurs in collections of 2-bit camera. Lots of details are lost due to low bit. Compared with this, 8-bit camera can display 256 colors, which is enough for many application.

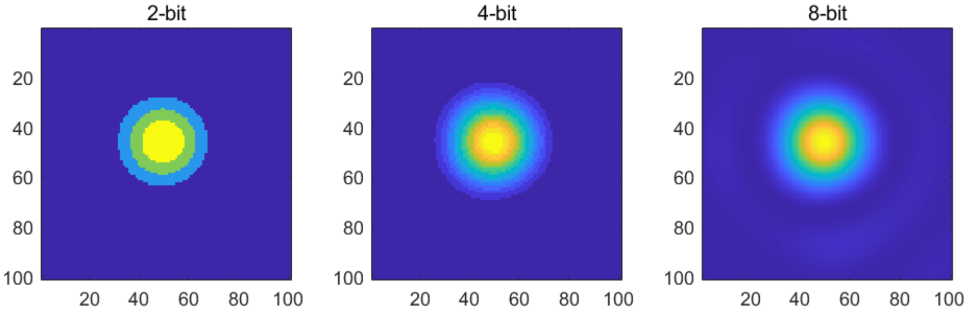


Figure 1-8: One PSF image (zoomed) under 2, 4 and 8 color depth.

In this chapter, "what is phase retrieval problem" is answered. The development of solutions to this problem is reviewed. Among many methods, a classic algorithm, Gerchberg-Saxton algorithm is explained in detail, including twin stagnation, which is a defect of GS algorithm. Meanwhile, two factors that affect the quality of PSF image, noise and camera's color depth, are mentioned as well.

Chapter 2

Proposal of Modification on Gerchberg-Saxton Algorithm

In this chapter, proposal of a modified Gerchberg-Saxton algorithm is introduced. The modification aims to expand the application of Gerchberg-Saxton algorithm on over-exposed PSF, i.e. retrieve phase from PSF with high level saturation, as well as reduce noise interference in the input field to the phase restoration.

2-1 Premodifier: Theoretical Background

Premodifier, also called as preconditioner, conditions a given problem into a form that is more suitable for numerical solving methods. For a linear system

$$Ax = b,$$

with matrix A, instead of solving the system itself, one may solve the right preconditioned system:

$$AP^{-1}Px = b$$

via solving

$$AP^{-1}y = b$$

for y and

$$Px = y$$

for x. Here, matrix P is a preconditioner, and it is common to call $T = P^{-1}$ the preconditioner. Preconditioner is useful in iterative methods to solve the original system.[17]

2-2 Over-exposure as Premodifier

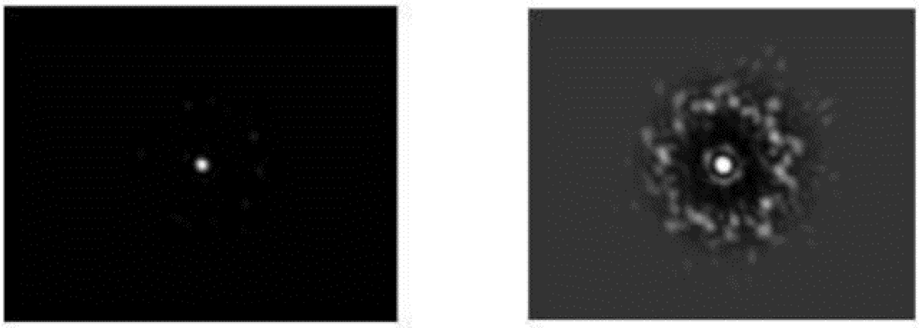
As mentioned in Chapter 1, the measurements collected by imaging camera could be over-exposed in some cases. Over-exposure is when an image appears brighter than it should, or brighter than neutral exposure. When too much light hits the camera's sensor, it results in an extremely bright image, which is overexposed.

Over-exposure can be considered as a premodifier in phase retrieval problem. Suppose that $p(x)$ denotes the real intensity, and $o(x)$ is the over-exposed PSF, i.e. data collected by imaging camera. The mathematical model of over-exposure can be expressed as:

$$o(x) = \begin{cases} p(x), & p(x) < c \\ c, & p(x) \geq c \end{cases},$$

where c is a constant which acts as a cut-line.

Over-exposure can be positive for phase retrieval. Compare with PSF without over-exposure, the over-exposed PSF has clearer structure and more details, since there is more light and the intensity that below the cut-line is actually enhanced. Fig.(2-1)[18] is an excellent example to show the positive effect of over-exposure. The original experimentally measured PSF is simple, and after over-exposure, the structure of the PSF is highlighted and more information could be collected.



(a) Experimentally measured PSF.

(b) Measured PSF overexposed to highlight structure.

Figure 2-1: A good example for positive effect of over-exposure.[18]

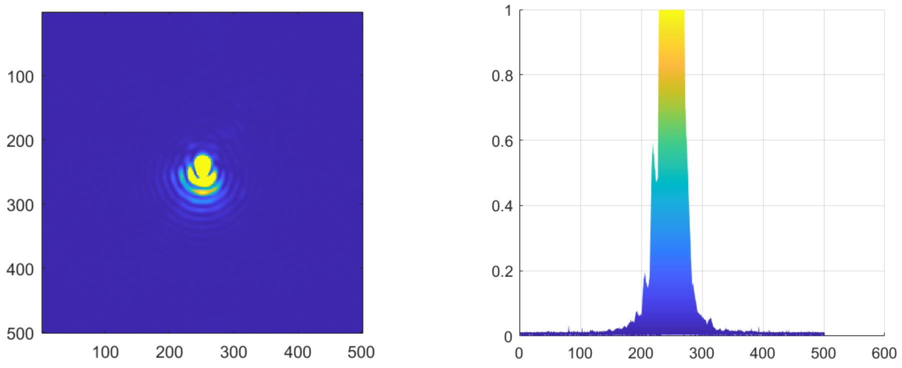
However, if the saturation level of over-exposed PSF is too high, the image would be extreme bright and it limits detail in the image and reduces opportunity for shadowing or distinguishable highlights in the image.

2-3 Adjustment of Simple Gerchberg-Saxton Algorithm

To expand the application of GS algorithm on over-exposed PSF, we propose to modify the algorithm on the constraints, and the algorithm is combined with the phase diversity method to remove bias in background or noise in the image.

2-3-1 Modification on the constraints on Fourier Domain

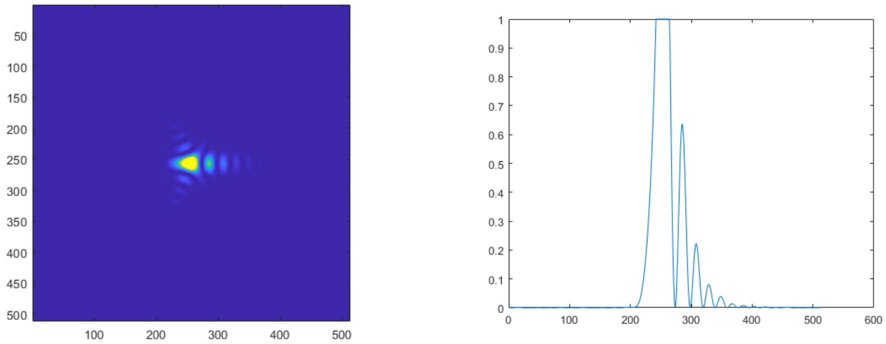
Fig.(2-2) and Fig.(2-3) show over-exposed image and the section view of real and simulation PSF, respectively. It is clear that the central part of the PSF, i.e. the part suffers from over-exposure, has the maximum intensity. Compare with the PSF of the same object but without over-exposure, the real information of these over-exposed pixels are lost.



(a) Real PSF image with high level saturation.

(b) Section view of (a).

Figure 2-2: A real PSF image and the section view of a high level saturated object.



(a) Simulated PSF image with 0.5 over-exposure level.

(b) Section view of (a).

Figure 2-3: A simulate PSF image and the section view of a high level saturated object.

In simple GS algorithm, which is shown in Fig.(1-2), the Fourier domain constraint $|H|$ is the square root of the intensity. It is according to Eq.(1-1). However, for over-exposed PSF, its square root is not the Fourier magnitude, thus the simple GS algorithm does not apply well in this situation. In this proposal of modified GS algorithm, in order to maximize the use of available data, a mask is designed. This "valid data mask" is a binary mask, and only save data that do not reach the maximum. And for the pixels that reach the maximum intensity, there is no constraints on and will keep the amplitude in the iteration.

As presented in Fig.(2-6), the modified constraint on Fourier domain is:

$$F_i^{'}(k) = M \cdot |H(k)| \cdot \frac{F_i(k)}{|F_i(k)|} + \gamma(1 - M) \cdot F_i(k). \tag{2-1}$$

$F_i(k)$ and $F_i^{'}(k)$ denote the Fourier transform and the estimate in the i iteration, respectively. M denotes the valid data mask, and γ is a parameter that adjusts magnification of the un-update part to keep it at the same order of magnitude as other data and make sure the part is the brightest. Fig.(2-5) is an example to show the importance of setting a parameter like γ. Though the un-update part has the highest value, these values are too huge to match other part and the reliable part, where the pixels values are updated, would be ignored in the following iteration process. If we do not make adjustments to this part, the algorithm is likely to regard the input PSF as the PSF of the point light source with piston phase, and finally output the wrong result. Fig.(2-4) is an example of valid data mask when the input is the simulation in Fig.(2-3).

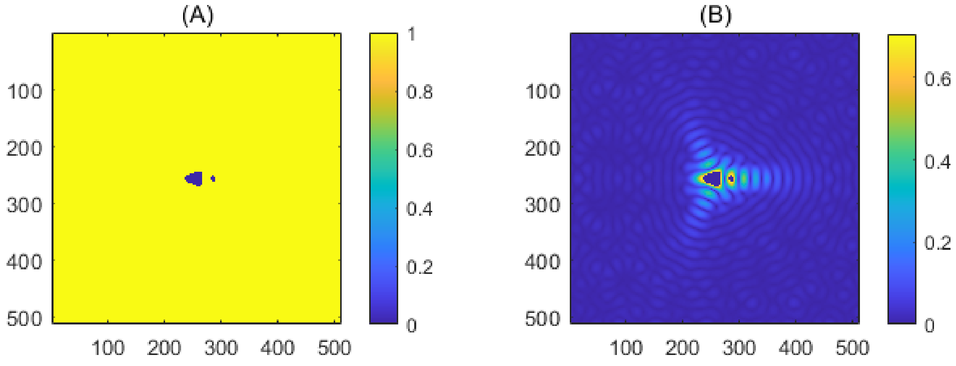
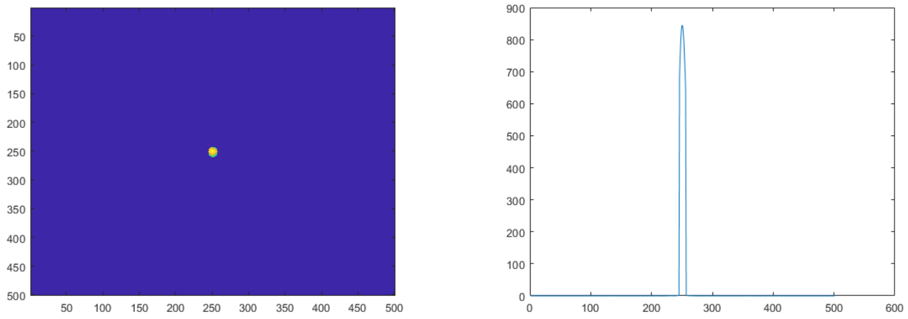


Figure 2-4: (A)Valid data mask of simulation in Fig.(2-3).(B) The Fourier domain constraint of Fig.(2-3).



(a) Fourier domain constrain image without parameter γ.

(b) Section view of the Fourier domain constrain.

Figure 2-5: Figures about Fourier constrain and its section view.

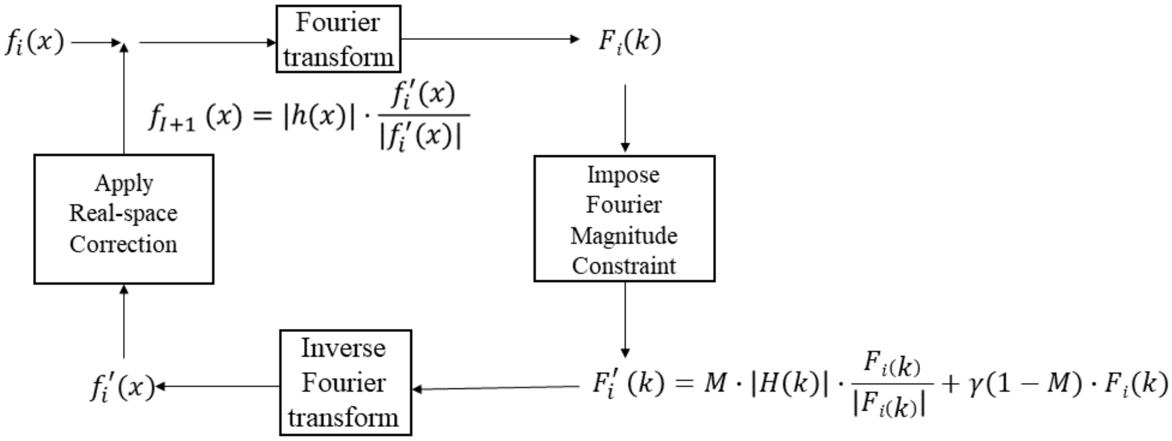

$f_i(x)$
Fourier transform
$F_i(k)$
$f_{I+1}(x) = |h(x)| \cdot \frac{f_i'(x)}{|f_i'(x)|}$
Apply Real-space Correction
Impose Fourier Magnitude Constraint
Inverse Fourier transform
$f_i'(x)$
$F_i'(k) = M \cdot |H(k)| \cdot \frac{F_i(k)}{|F_i(k)|} + \gamma(1-M) \cdot F_i(k)$

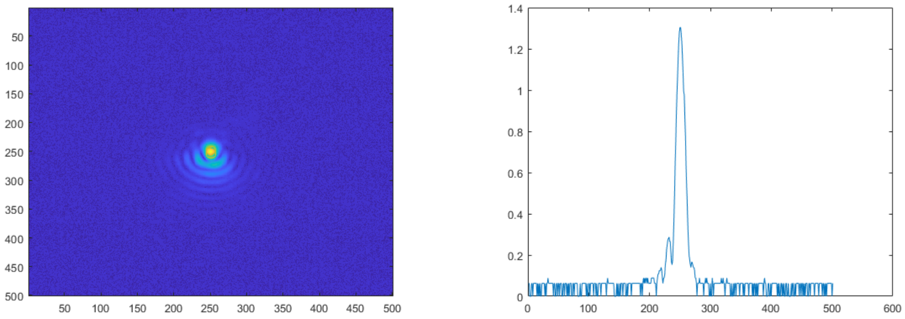
Figure 2-6: The block diagrams of algorithm with a modified constraint on Fourier domain.

In this thesis, we propose 3 ideas to decide the parameter γ. We use Fig.(2-3) to compare the evaluation of these 3 methods as we know the ground truth in this case. In this chapter we explain two of them and introduce the rest in the appendix, since the idea of the rest one is in-comprehensive, though it works to some extent.

The difference between the following two methods is how much we know about prior information. For the case that we only have the over-exposed PSF, γ can be calculated according to

$$\gamma = \frac{\int \sqrt{I} - \int (M \cdot \sqrt{I})}{\int((1-M) \cdot |\mathcal{F}(f_i(x))|)}, \tag{2-2}$$

with I is the input over-exposed PSF data, M is the valid data mask and $f_i(x)$ is a complex-value result after being imposed real-space magnitude constraints in the iteration process. The effect of γ is shown as Fig.(2-7). The details of the diffraction are saved and the un-update part matches well. As the value of γ is related with $f_i(x)$ and $f_i(x)$ is changed after every iteration, γ needs to be updated on each iteration.



(a) Fourier domain constrain image.

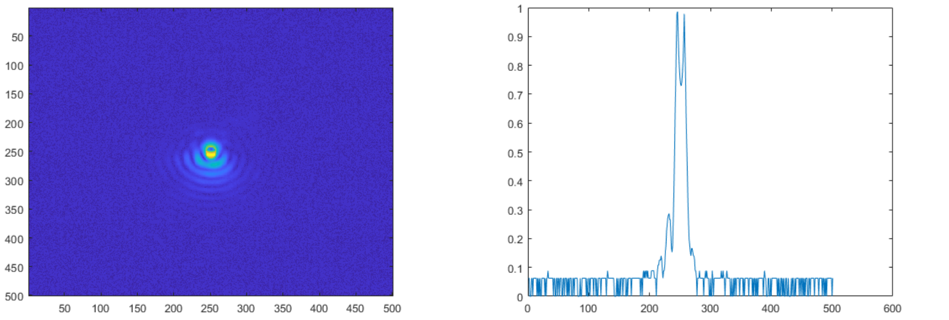
(b) Section view of the Fourier domain constrain.

Figure 2-7: Figures about Fourier constrain and its section view when using γ calculated by Eq.(2-2).

For the case that we know both the original PSF and an over-exposed one, which expressed as I_{exp} and I_{ori} in Eq.(2-2) respectively, the parameter γ is decied by

$$\gamma = \mathbf{mean}(\frac{I_{exp}}{I_{ori}}). \tag{2-3}$$

In this method, we consider that the over-exposed PSF is the cut of magnification of original PSF. Hence we divide the two PSF. Notice that we only use the "valid values", i.e. values under the maximum. Fig.(2-8) presents the effect of γ calculated by Eq.(2-3). In this case γ will not change in the iteration. One thing needs to point out is that in practice the PSF images are influenced by noise, which leads to the quotient of $\frac{I_{exp}}{I_{ori}}$ be different in every pixel, and therefore we calculate the mean of the quotient in every pixel. However, though the method of calculating the average value is more comprehensive, the calculated result may be smaller than the actual magnification, which will lead to double peaks, as Fig(2-8(b)).



(a) Fourier domain constrain image.

(b) Section view of the Fourier domain constrain.

Figure 2-8: Figures about Fourier constrain and its section view when using γ calculated by Eq.(2-3).

2-3-2 Modification to Remove Background

An ideal PSF has a 0 background, which means pixels that are far from the center diffraction area of PSF have zero intensity. However, in practice, the PSF images measured by the optic imaging system contain unavoidable noise and aberration, which cause the background non-zero. The ba ckground will interfere with the convergence direction of the simple GS algorithm and affect the result of the restoration.

To solve this problem, we integrate a method that mentioned by Oleg Soloviev *et al.* in [19]. They suggest to use a phase diversity method to separate the negative background from PSF and to reduce the effect of the background.

Incoherent 3D Phase Retrieval Problem

The background of experimental PSF image, which technically is supposed to be zero, contains noise and stuff thus is non-zero. Removing the background could reduce the iteration times of the GS algorithm and improve the retrieval quality. Suppose that the background is a constant and could be expressed as following equation:

$$I = y^2 = |\mathcal{F}_2(a(x)e^{i\phi(x)})|^2 + b. \tag{2-4}$$

It can be considered as an incoherent 3D phase retrieval problem. 3D phase retrieval problem, also known as phase-diverse phase retrieval, is a method to retrieve phase when we know the information about PSF image and series of real-space amplitude. 2D phase retrieval problem, such as Eq.((1-1), becomes find $\phi \in \mathcal{R}$, if $y, a_1, ..., a_M \in \mathcal{R}$ are known

$$y^2 = |\mathcal{F}_2(a_1e^{i\phi})|^2 + ... + |\mathcal{F}_2(a_Me^{i\phi})|^2. \tag{2-5}$$

Consider a vector $v = (\mathcal{F}_2(a_1e^{i\phi}), ..., \mathcal{F}_2(a_Me^{i\phi}))$, then Eq(2-5) is equivalent to

$$y^2 = \|v\|_2^2 = \|\mathcal{F}_1v\| \,|_2^2, \tag{2-6}$$

and $\mathcal{F}_1v = \mathcal{F}_1((\mathcal{F}_2(a_1e^{i\phi}), ..., \mathcal{F}_2(a_Me^{i\phi})) = \mathcal{F}_3(a_1e^{i\phi}, ..., a_Me^{i\phi})$.

Fig(2-9) and Eq(2-7) are the prior information. x and X are a Fourier pair. $p_{i,j}$ is the amplitude of PSF at position (i, j) and $|x|$ is the amplitude of pupil. Every element in vector $x_{i,j,\cdot}$ has same phase.

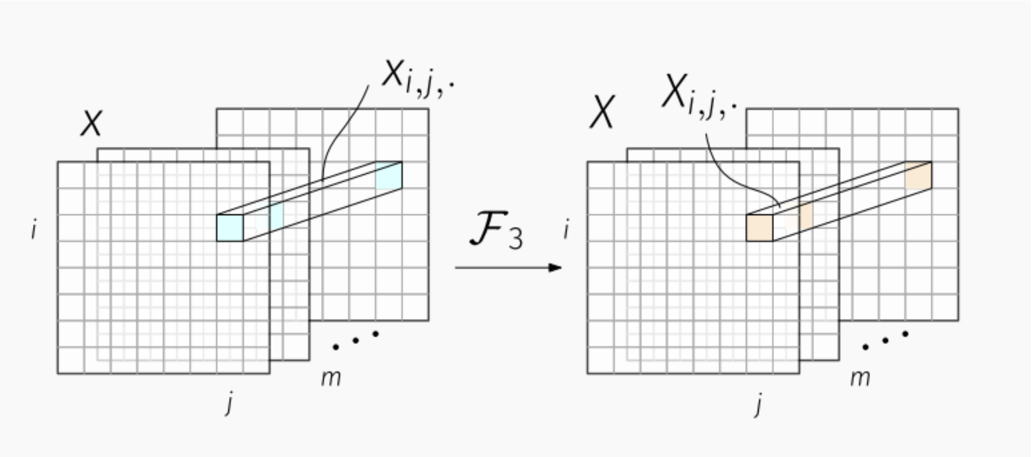


Figure 2-9: Incoherent 3D PR problem.

$$\begin{cases} X = \mathcal{F}_3x \\ p_{i,j} = \|X_{i,j,\cdot}\|_2^2 \\ a = |x| \\ \arg x_{i,j,1} = \ldots = \arg x_{i,j,M} \end{cases} \tag{2-7}$$

We can use above 3D phase retrieval method to solve Eq(2-4) by adding an additional delta amplitude, i.e. only one pixel is non-zero in the amplitude. Hence Eq.(2-4) can be expressed in form:

$$I = y^2 = |\mathcal{F}_2(a(x)e^{i\phi(x)})|^2 + |\mathcal{F}_2(\sqrt{b}\delta(x)e^{i\phi(0)})|^2 \tag{2-8}$$

In equations (2-4) and (2-8), $\mathcal{F}_2$ denotes 2 dimensional Fourier Transform. $\delta(x)$ is the delta amplitude. The specific value of background b is not necessarily known, since it has no influence of phase retrieval when using the following algorithm3.

Algorithm 3 3D Vector Gerchberg-Saxton algorithm[20]

```
Input: $a$ - Real space amplitude constrain,
    $\sqrt{p_{i,j}} = \|X_{i,j,\cdot}\|_2$ - Fourier magnitude constrain,
    $\phi_0$ - Random guessed initial phase
Output: $\Phi$ - Retrieved phase.
Initialization: $x_0 = |a|e^{i\phi_0}$
 General Step (k=1,2,...):
 1: $\hat{X}^k = \mathcal{F}_3 x^k$
 2: $X^k = \sqrt{p_{i,j}} \frac{\hat{X}^k}{\|\hat{X}^k_{i,j,\cdot}\|_2}$
 3: $\hat{x}^{k+1} = \mathcal{F}_3^{-1} X^k$
 4: $\Phi^{k+1}_{i,j,\cdot} = \arg \frac{\sum_m a_{i,j,m} \cdot \hat{x}^{k+1}_{i,j,m}}{\sum_m a^2_{i,j,m}}$
 5: $x^{k+1}_{i,j,m} = a_{i,j,m} e^{i\Phi^{k+1}_{i,j,\cdot}}$
 6: Go to 1.
```

2-4 A Modified Gerchberg-Saxton Algorithm

Based on the previous two modifications in the simple Gerchberg-Saxton algorithm, here is a proposal of a modified Gerchberg-Saxton algorithm.

Algorithm 4 Proposed Method: Modified Gerchberg-Saxton algorithm for over-exposed PSF

```
Input: $a$ - Real space amplitude constrain,
    $\sqrt{p_{i,j}} = \|X_{i,j,\cdot}\|_2$ - Fourier magnitude constrain,
    $\phi_0$ - Random guessed initial phase,
    $M$ - The valid data mask,
    $\gamma$ - A constant parameter.
Output: $\Phi$ - Retrieved phase.
Initialization: $x_0 = |a|e^{i\phi_0}$
 General Step (k=1,2,...):
 1: $\hat{X}^k = \mathcal{F}_3 x^k$
 2: $X^k = M \cdot \sqrt{p_{i,j}} \frac{\hat{X}^k}{\|\hat{X}^k_{i,j,\cdot}\|_2} + \gamma(1 - M) \cdot \hat{X}^k$
 3: $\hat{x}^{k+1} = \mathcal{F}_3^{-1} X^k$
 4: $\Phi^{k+1}_{i,j,\cdot} = \arg \frac{\sum_m a_{i,j,m} \cdot \hat{x}^{k+1}_{i,j,m}}{\sum_m a^2_{i,j,m}}$
 5: $x^{k+1}_{i,j,m} = a_{i,j,m} e^{i\Phi^{k+1}_{i,j,\cdot}}$
 6: Go to 1.
```

Fig.(2-10) shows the block diagram of the process of the proposed method. This proposed is supposed to retrieve phase from PSF images that suffered from over-exposure and noise.

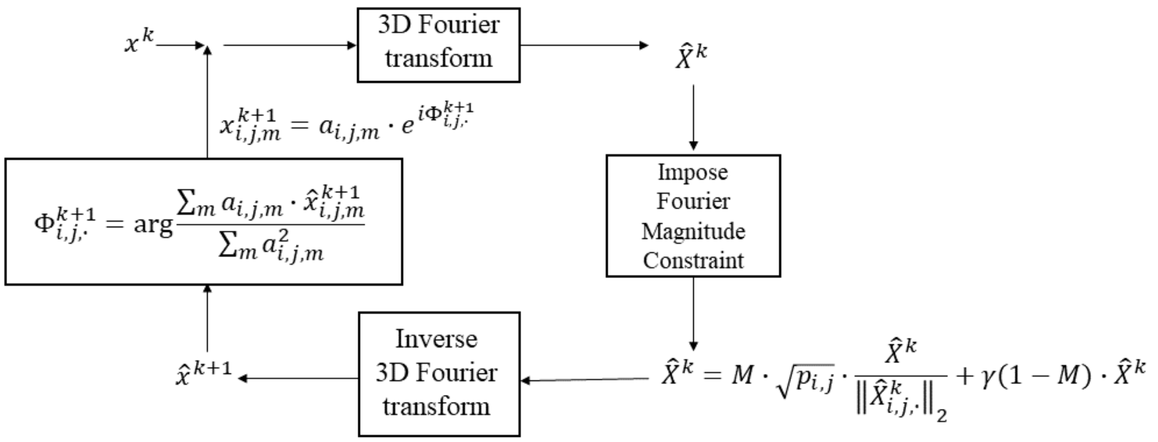

x^k
3D Fourier transform
$\hat{X}^k$
$x^{k+1}_{i,j,m} = a_{i,j,m} \cdot e^{i\Phi^{k+1}_{i,j,\cdot}}$
$\Phi^{k+1}_{i,j,\cdot} = \arg\frac{\sum_m a_{i,j,m} \cdot \hat{x}^{k+1}_{i,j,m}}{\sum_m a^2_{i,j,m}}$
Impose Fourier Magnitude Constraint
$\hat{x}^{k+1}$
Inverse 3D Fourier transform
$\hat{X}^k = M \cdot \sqrt{p_{i,j}} \cdot \frac{\hat{X}^k}{\|\hat{X}^k_{i,j,\cdot}\|_2} + \gamma(1 - M) \cdot \hat{X}^k$

Figure 2-10: The block diagram of the proposed method.

In this chapter, modelling of over-exposure and details of the proposed modified Gerchberg-Saxton algorithm is described. This algorithm has the ability to handle over-exposed PSF images as well as reduce the negative effect of noise and aberrations. In the following chapters, we test our new algorithm on numerical simulation experiments and apply the algorithm on real data, to check its robustness against noise and over-exposure. Meanwhile, whether over-exposure could improve the performance of the algorithm when used as a premodifier will also be answered in the following chapters.

Chapter 3

Numerical Simulation Experiment for Testing Modified GS Algorithm

We firstly implement numerical simulation experiment on MATLAB platform to test and verify the proposed algorithm. Two types of phase are used to generate PSF images in the experiment, one is random multiple low-order Zernike polynomials phase and another is a picture Lena. This chapter illustrates the way to build forward model, the process of the experiment and analysis of results.

3-1 Forward Model for PSF Generation

3-1-1 Original PSF Images Generation

The point light source emits light uniformly in all directions, and its size is negligible. The light wave emitted by the light source can be described as a wavefront. The description of the wavefront at the aperture is called the pupil function. The pupil function describes the wavefront that enters the aperture of the imaging system, and it is expressed as

$$P(x, y) = A(x, y) * e^{i\phi(x,y)}. \tag{3-1}$$

Eq.(3-1) describes a two dimensional pupil function. x and $y$represent the coordinate position. $\phi(x, y)$ is phase, and $A(x, y)$ is the amplitude function of the aperture. Aperture's amplitude function, as shown in Fig.(3-1), is a binary function with the inside aperture value is 1 while the outside value is 0.

One way to describe phase $\phi(x, y)$ is using weighted sum of Zernike polynomials, like Eq. (3-2)

$$\phi = \sum_i a_i Z_i(x, y). \tag{3-2}$$

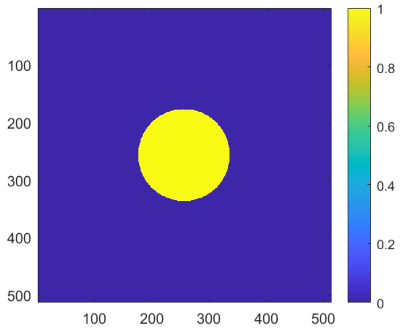


Figure 3-1: An example of aperture function.

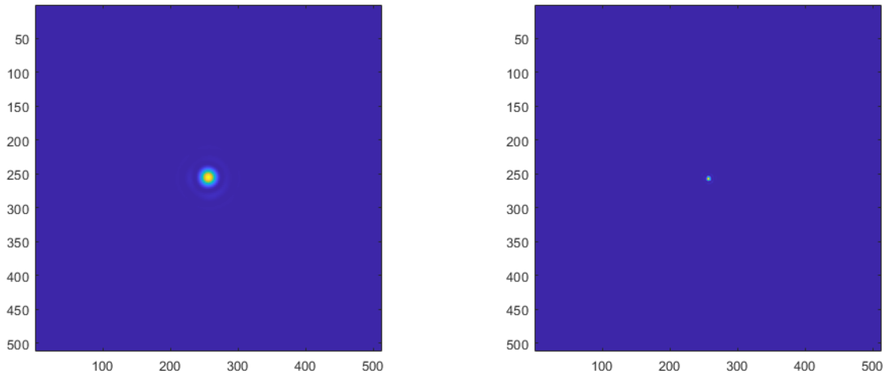
The coefficient a_i determines the amplitude of Zernike polynomial Z_i in the wavefront. In our experiment, we use low-order Zernike modes, i.e. the first 15 Noll's ordering Zernike Polynomials. Description about these Zernike Polynomials is in Appendix A-3.

Point Spread function (PSF) is the image of a point source as seen by the imaging system. The wavefront results in the intensity distribution and the relationship between wavefront and PSF is

$$I = |\mathcal{F}(P(x,y))|^2 = |\mathcal{F}(A(x,y) * e^{i\phi(x,y)})|^2, \tag{3-3}$$

with $\mathcal{F}$ represents Fourier transform. In this case, it is two dimensional Fourier Transform.

In the experiment, we use two kinds of phase to generate PSF image. One is like Eq.(3-2), which is made of some of the first 15 order Zernike modes with random coefficient; another is a certain picture, as shown in Fig.(3-3). The size of phase is 512×512 pixels, and the aperture radius is 15 pixels. The size of aperture determine the diameter of the PSF. Fig.(3-2) shows the difference between radius 15pix aperture and radius 60pix aperture.



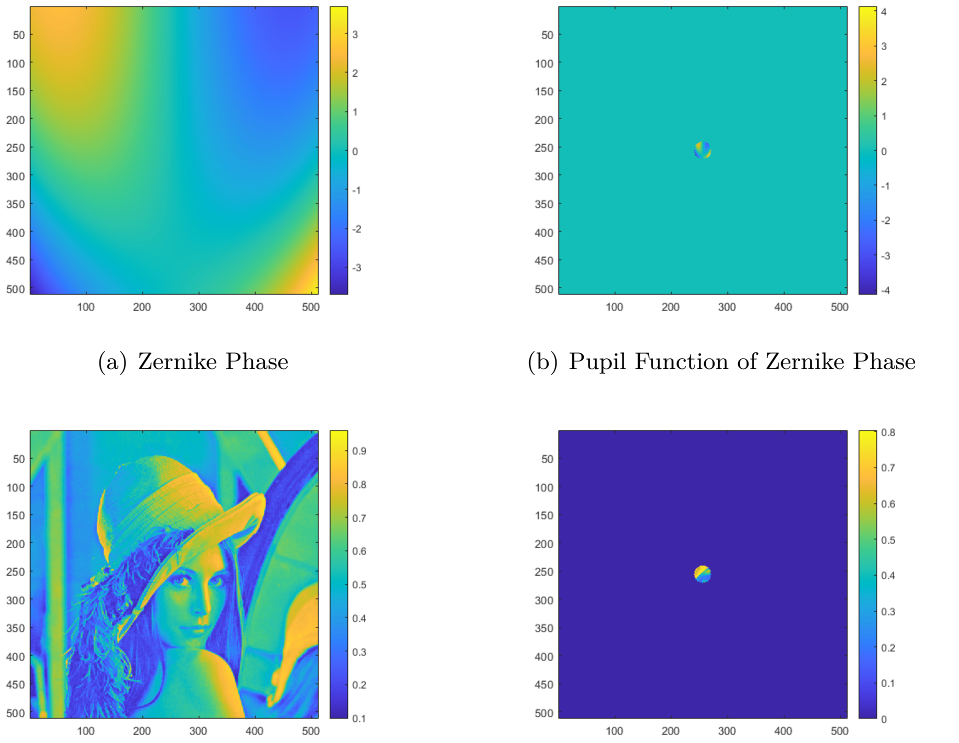
(a) PSF of aperture with radius 15pix.

(b) PSF of aperture with radius 60pix.

Figure 3-2: Difference between PSF generated by radius 15pix aperture and radius 60pix aperture.

It is obvious that the smaller the aperture is, the larger Airy disk will be. Airy disk is description of the best-focused spot of light that a perfect lens with a circular aperture can make, limited by the diffraction of light[21]. When the size of PSFs are the same, bigger size

Airy disk contains more information. In order to make the algorithm work better and obtain good experimental results, we set the aperture radius as 15pix. Fig.(3-3) shows two kinds of phase and their pupil functions that we use in the following numerical simulation experiment.



(a) Zernike Phase

(b) Pupil Function of Zernike Phase

(c) Picture(Lena) Phase

(d) Pupil Function of Picture(Lena) Phase

Figure 3-3: Two kinds of phase and their pupil functions used in the experiment.

3-1-2 Over-exposure Simulation

The final goal of the experiment is to verify feasibility of the modified GS algorithm for over-exposed PSF images. In Section 3-1-1, an ideal PSF image is generated. To simulate over-exposure, firstly the intensity should be enhanced, as more intensity of light is collected by camera, which leads to a brighter image. To simulate the limitation of camera's color depth, we set a constant acts as cut-line. For example, for 8-bit camera, the range of colors is 0-255, then the cut-line is 255 and the over-exposure function could be

$$o(x) = \begin{cases} \alpha \cdot p(x), & \alpha \cdot p(x) < 255 \\ 255, & \alpha \cdot p(x) \geq 255 \end{cases}, \tag{3-4}$$

with α is a constant parameter, which represents the magnification of the original PSF $p(x)$. Fig(3-4) is an example of a PSF and the over-exposed one with $\alpha = 5$.

In our simulation, we normalize PSF images to make them in range $0-1$. This is to facilitate applying noise to the image later. In order to indicate how exposed the image is, we introduce

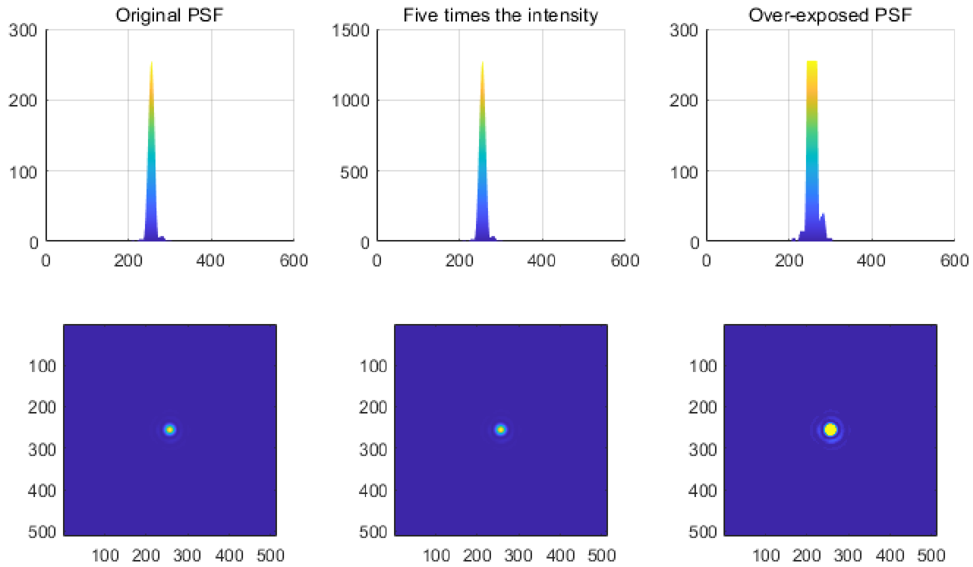


Figure 3-4: An example of generating an over-exposed PSF. Figures in the top row are cross sections of the bottom PSF images.

a parameter, i.e. "over-exposure level", α in Eq.(3-4). Over-exposure level is a constant that denotes how many times the intensity increased by exposure of the part where the intensity is under the limitation of the color depth. Over-exposure level of a PSF that has no effect of exposure is 1, and the example in Fig(3-4) has over-exposure level as 5. The higher over-exposure level, the greater the impact of over-exposure. We set four different over-exposure level in our experiments, i.e. 1, 1.25, 2, 5. Fig(3-5) and Fig(3-6) are PSF images of two kinds of phase with four over-exposure levels.

3-1-3 Noise Simulation

We add white Gaussian noise to the PSF image in this numerical simulation to simulate noise. The basic information about Gaussian noise can be found in A-2.

Gaussian noise is usually modeled as an additive noise. Fig.(3-7) shows the effect of a small amount of Gaussian noise and of an increased noise. Gaussian noise with larger variance value can make more "fuzziness". Various filtering techniques can improve the image's quality, though usually at the expense of some loss of sharpness[22].

In the simulation, we assume the noise is additive Gaussian noise. Using `Matlab` command `imnoise`, we directly add Gaussian noise to PSF images. We use signal-to-noise ratio (SNR) to qualify the influence of noise and the corresponding SNR values are 100dB, 80dB, 50dB and 20dB. We consider that SNR= $100dB$ denotes there is no noise in the image.

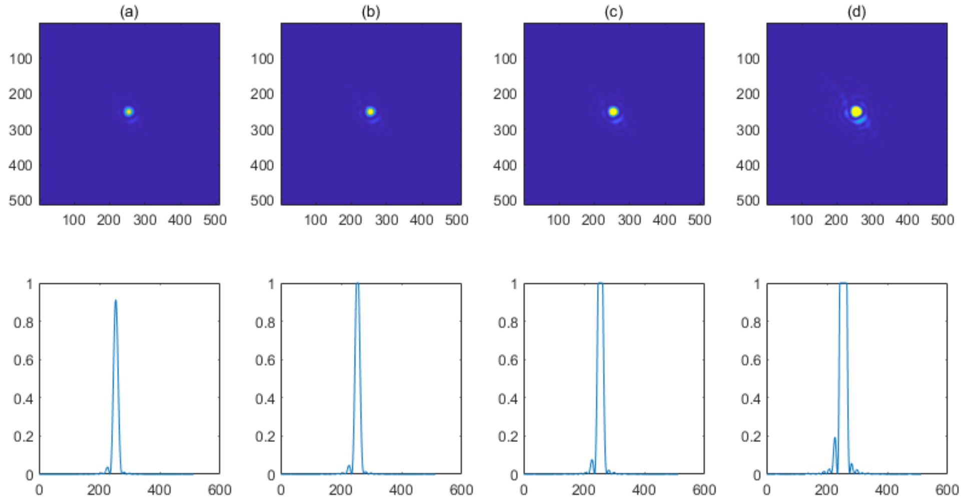


Figure 3-5: Four over-exposure level used in the simulation of object Lena. (a) The original PSF. (b) PSF with over-exposure level 1.25. (c)PSF with over-exposure level 2. (d)PSF with over-exposure level 5. The bottom plots are the section views of the tops.

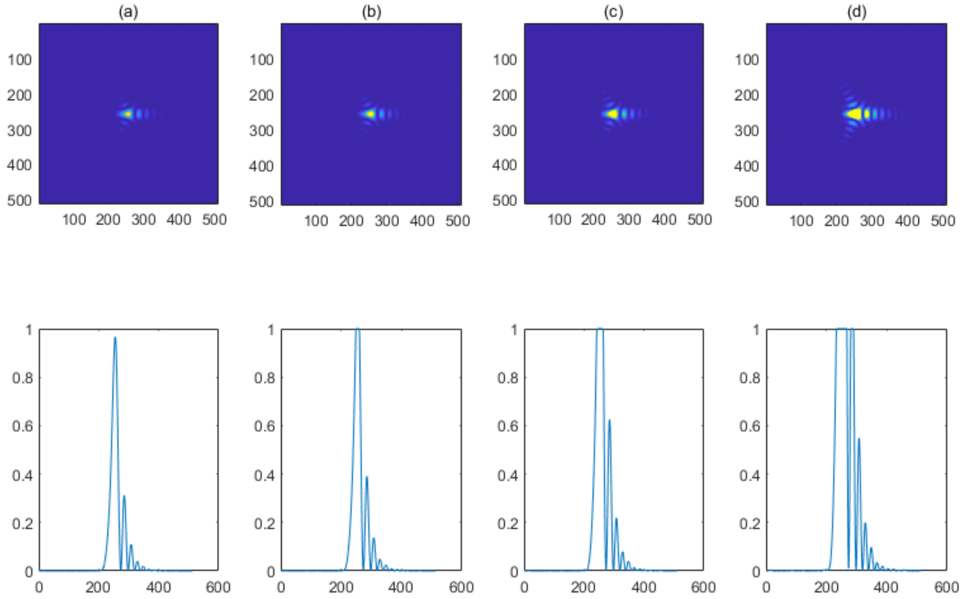


Figure 3-6: Four over-exposure level used in the simulation of object Zernike modes. (a) The original PSF. (b) PSF with over-exposure level 1.25. (c)PSF with over-exposure level 2. (d)PSF with over-exposure level 5. The bottom plots are the section views of the tops.

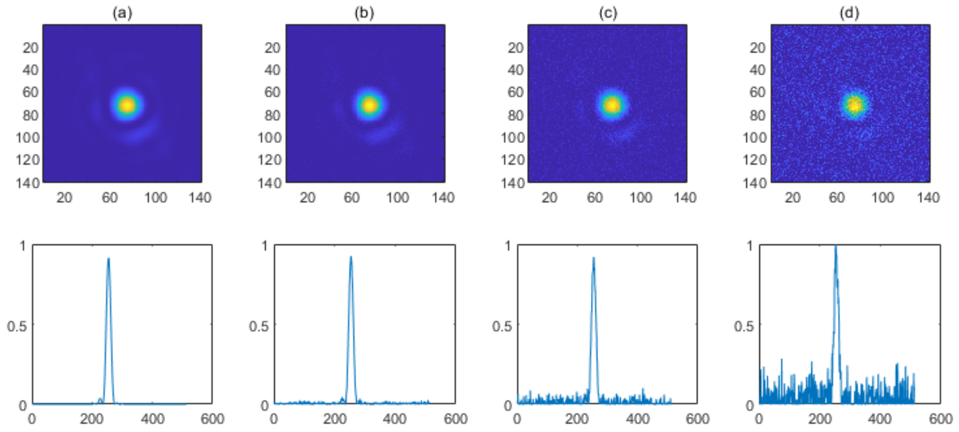


Figure 3-7: PSF images suffering from different level noise. (a) is a PSF image without noise. (b)-(d) are PSFs influenced by Gaussian noise with 0 mean and 10^{-11}, 10^{-8},10^{-5} variance, respectively.

3-1-4 Simulation of Color Depth of Camera

One reason that causes over-exposure is the color depth of imaging camera is not enough to smoothly present all the captured colors. The relationship between color depth, i.e. bit number, and the number of colors that can be presented is as follows:

$$N = 2^n, \tag{3-5}$$

where N denotes how many colors can be shown at a time, and n denotes the color depth. In our experiment, we assume the camera's color depth is 8, i.e. total 256 colors can be expressed in the image. In order to facilitate our operation on the `matlab`, all PSFs have been normalized, and the intensity is in the range of $0 - 1$.

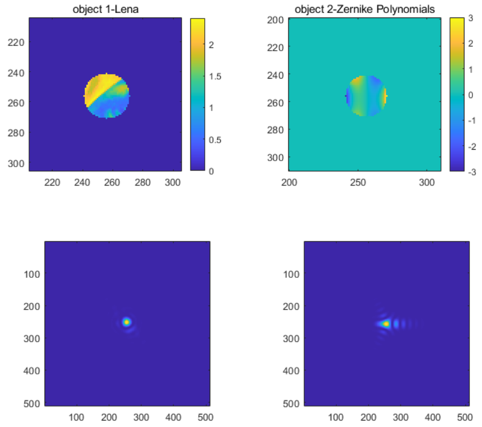


Figure 3-8: The two objects used in simulations in the top row and the corresponding PSF images (no noise and over-exposure) in the bottom row.

3-2 Results and Analysis of the Numerical Simulation

In this section we discuss the results of the simulations with different levels of noise and over-exposure. We compare the proposed modified GS algorithm to the simple GS algorithm. Two kinds of phase, as shown in Fig(3-8), are used to generate PSF images. We set four levels of over-exposure and noise respectively, as introduced in the last section. The details parameters are recorded in Table3-1. Except running the proposed modified GS algorithm on the simulation, we also run the simple GS algorithm with object 1. The aim is to compare the performance of the two algorithms.

parameter	inputs
object	object 1: Lena object 2: Zernike Polynomials
noise variance of Gaussian noise (SNR)	0 (100dB) 10^{-11} (80dB) 10^{-8} (50dB) 10^{-5} (20dB)
over-exposure level	1 1.25 2 5

Table 3-1: Settings of the simulation. The object can be found in Fig(3-8).

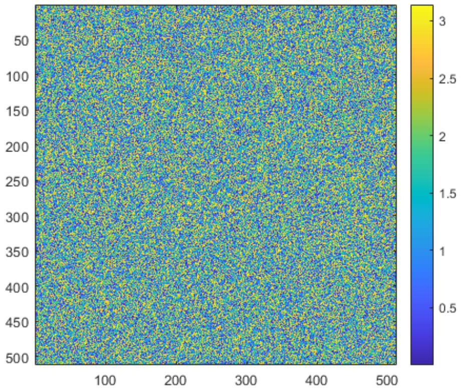


Figure 3-9: The random guess initial phase of the proposed method.

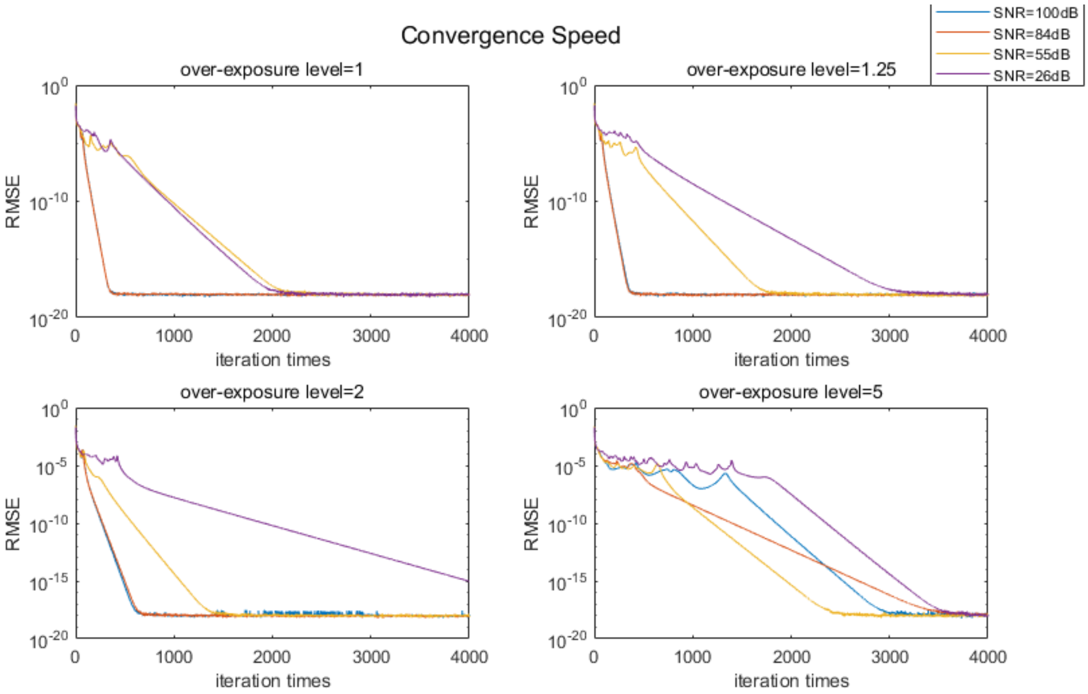


Figure 3-10: Modified GS algorithm convergence speed plots of object1 Lena. The x- axis presents iteration time, and the y- axis is the root mean square of difference of results of every two iterations. In every experiment, the algorithm converges to a result, as plots reach a stable state.

Every experiment the algorithm run 4000 times iteration. The initial phase is a random guess in the range of $(0, \pi)$, as shown in Fig.(3-9). According to Fig(3-10) and Fig(3-15), we can check the converge speed. It plots x- and y-coordinates using a linear scale on the x-axis and a base-10 logarithmic scale on the y-axis. The y- axis presents the root mean square of difference of results of every two iterations. When the plot has a rapid decrease and finally reaches a stable state, we consider the algorithm converges to a result.

The standards we use to compare performance are the average of the error, i.e. **ME**, and Peak signal-to-noise ratio (PSNR), with the ground truth as the reference. The error is using retrieved phase subtract the ground-truth, i.e. the object in Fig.(3-8). Besides, the error between every experiment result and the ground truth are presented in Fig.(3-12) and Fig.(3-17). From this, the quality of retrieved phase can be observed intuitively. PSNR is one of the commonly used standards to measure the effect of restoration. Its unit is decibel. The larger the value of PSNR, the closer the comparison value is to the reference value.

Fig.(3-11) and Fig.(3-16) show results of using modified GS algorithm to retrieve object Lena and Zernike modes from different noise and over-exposure. Fig.(3-13) and Fig.(3-18) plot mean error of results for the 2 objects, while Fig.(3-14) and Fig.(3-19) plot the PSNR of results. Following we analyze these figures and evaluate the performance of the modified GS algorithm from the ability on dealing with noise and over-exposure.

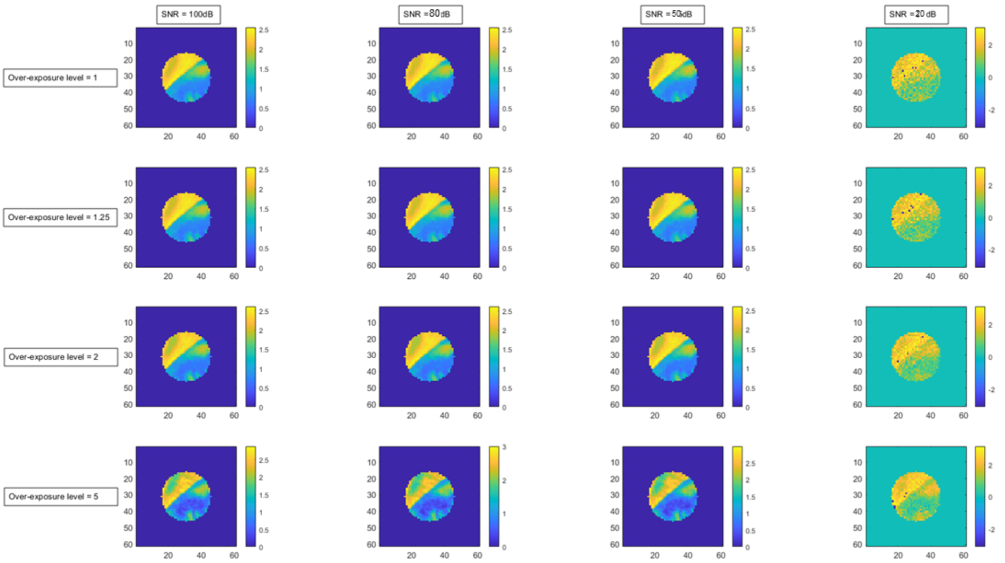


Figure 3-11: The retrieved phases in experiments of object 1 via proposed modified GS algorithm. In the 4×4 images array, the experimental input of each row has the same over-exposure level, and the experimental input of each column has the same SNR.

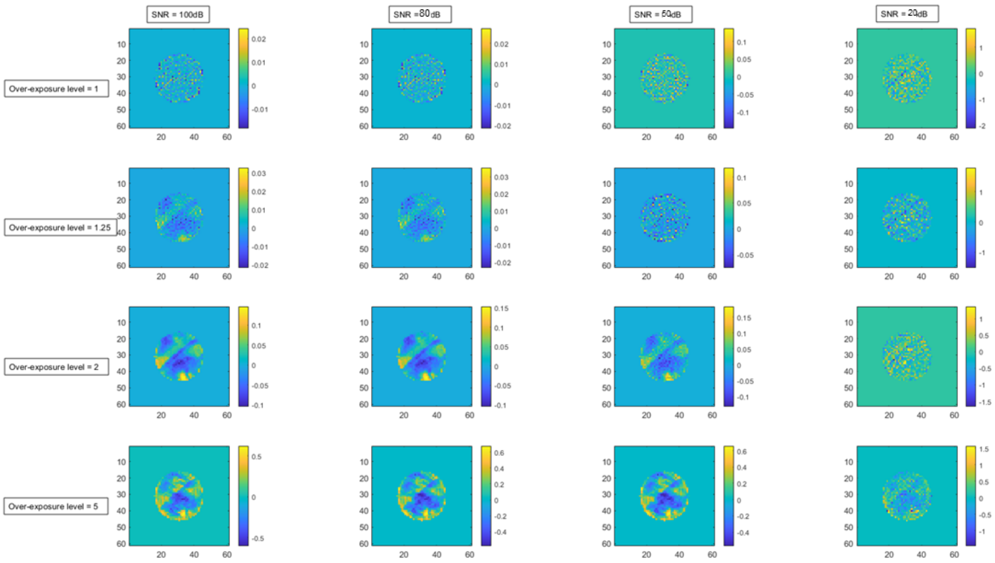


Figure 3-12: The differences between every experiment result and the ground truth Lena. In the 4×4 images array, the experimental input of each row has the same over-exposure level, and the experimental input of each column has the same SNR.

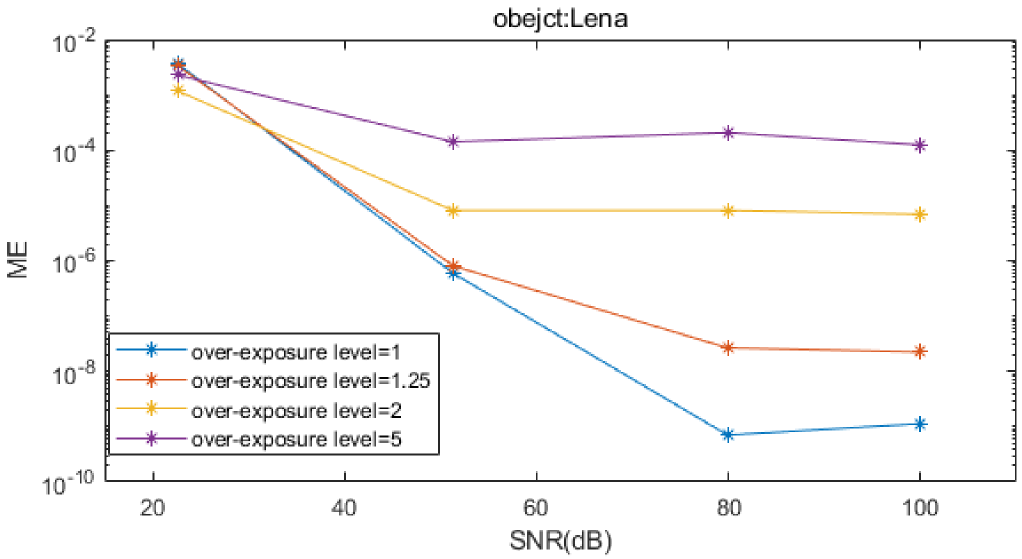


Figure 3-13: Plots of the average error of results about using modified GS algorithm to retrieve object Lena from different noise and over-exposure level. Over-exposure level is a constant that denotes how many times the intensity increased by exposure, e.g. over-exposure level 1 means no over-exposure effect and 5 means the intensity is enlarged 5 times.

3-2-1 Robustness against Noise

As shown in Table3-1, we add different levels of Gaussian noise to the original PSF and make the Signal-to-noise Ratio (SNR) at $80dB$, $50dB$ and $20dB$. The original PSF acts as a control group and its SNR is considered as $100dB$. The larger the value of SNR, the smaller the signal is interfered by noise. For more information about SNR, look up the Appendix(A-4).

From Fig(3-13), we know that when the noise level is low, i.e. SNR is higher than $80dB$, the changes of every plot are slight. With the same over-exposure, low noise interference has little effect on the ability of the algorithm to reconstruct the phase. We can suppose that the algorithm has robustness against low noise. However, when the noise has higher power, the performance of the algorithm decreases evidently, especially for the PSF without over-exposure. It is a support that over-exposure premodifier can offset some of the negative effects of noise. For cases that the PSF images suffer from high level noise, no matter what level of over-exposure premodifier is, the retrieved phase contains clearly noise. Therefore, we cannot assert that over-exposure is effective for high-level noise for this object based on Fig.(3-13).

The information that we can obtain from Fig.(3-14) is that retrievals of the input without over-exposure have the lowest PSNR in a high level noise condition, which intimates in this situation the over-exposure could be helpful for phase restoration.

Results of object2 are slightly different with the object1. Firstly, the range of **ME** enlarges. Fig(3-18) shows that for inputs with low saturation, i.e. the red line and blue line, when the SNR changes from $100dB$ to $80dB$, the error becomes larger but still at a tiny value; for inputs with high saturation, changes in SNR has slight influence on the performance. In Fig.(3-19), as the influence of noise increases, the four plots have not changed much, which also shows the stability of the algorithm.

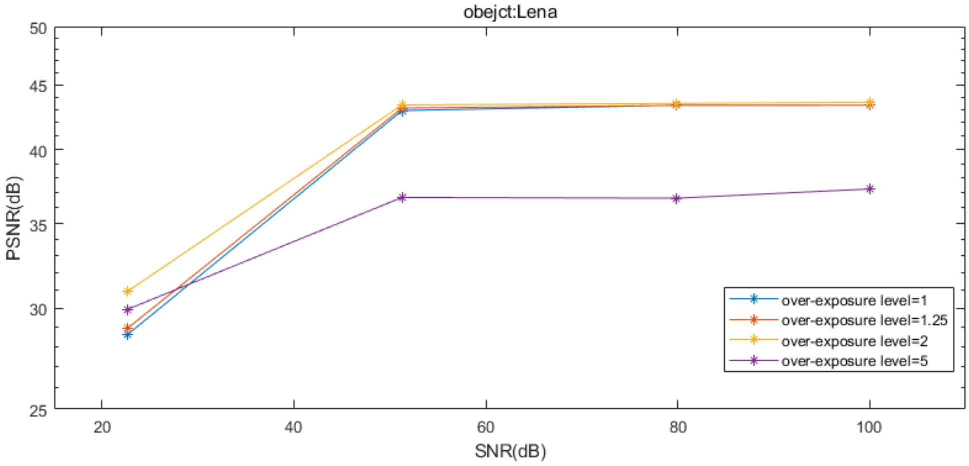


Figure 3-14: Plots of PSNR of the retrieval in every experiment. Every plot has the same over-exposure level. x-axis and y-axis denote SNR in every experiment input and values of PSNR, respectively. The ground truth acts as the reference when calculate PSNR. Larger PSNR is considered as better retrieval.

In every over-exposure condition, the modified GS algorithm presents the robustness against noise. Although the mean error of some results are not good, the algorithm finally converges to a result similar to ground-truth in each set of experiments. Meanwhile, the retrieved phases are as the same or similar with the ground truth. It also means the algorithm as least converges to right solution.

3-2-2 Over-exposure Premodifier

Four level of saturation are simulated in the experiments. Over-exposure level is a constant that denotes how many times the intensity increased by exposure, e.g. over-exposure level 1 means no over-exposure effect and 5 means the intensity is enlarged 5 times.

From Fig.(3-13) we can see that when the input images suffer from higher level noise, the over-exposure premodifier can cancel out a certain degree of negative effect of noise, as the changes in line "over-exposure level = 1" are the most. Fig.(3-14) also supports this. When the inputs only has 20dB SNR, the retrieval of the one where has no over-exposure effect has the lowest PSNR.

Comparing the red line and the blue line in Fig,(3-18) we can consider that over-exposure premodifier has an effect on improving the performance of the algorithm. The low level over-exposure has almost the same result of the control group in the low level noise cases, and have smaller mean error value when the SNR is in the range $30dB - 80dB$. The situation is different when the SNR value is small. No matter which object, the blue points is not the lowest, which means that when the noise interference is strong, over-exposure may help phase recovery.

However, results shown in Fig,(3-19) is unexpected, as results of inputs with the highest over-exposure level have the best performance in PSNR while results of inputs without over-exposure have the lowest PSNR values. Though all PSNR values are at low level, around 20dB.

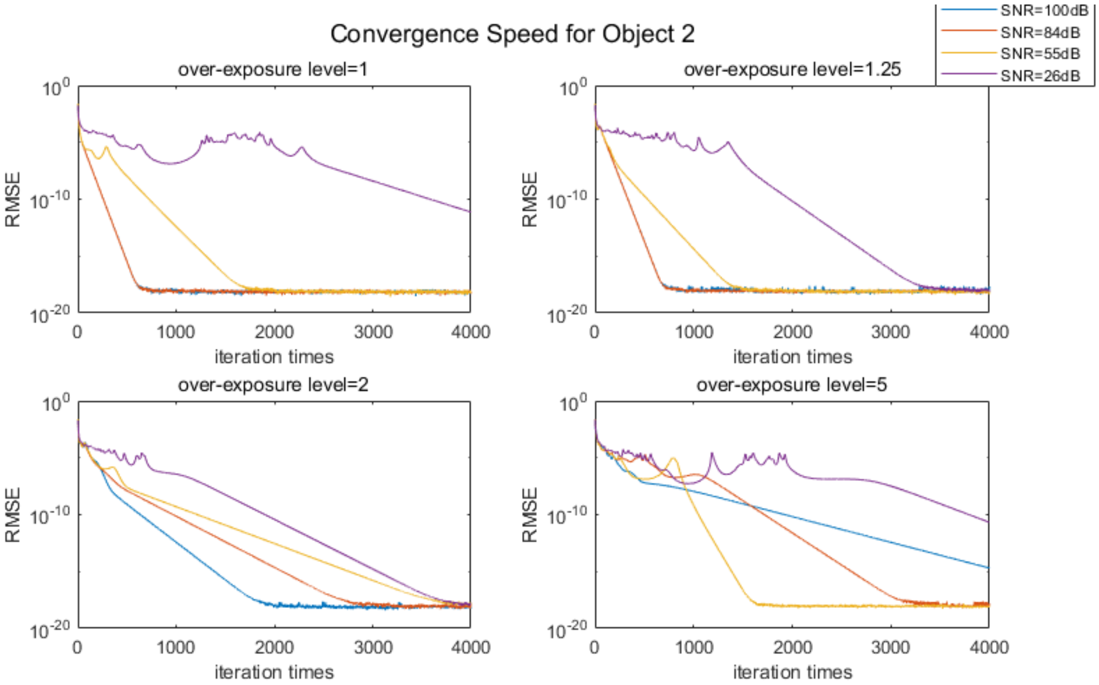


Figure 3-15: Modified GS algorithm convergence speed plots of object2 Zernike modes. The x-axis presents iteration time, and the y- axis is the root mean square of difference of results of every two iterations. We consider that in every experiment the algorithm converges to a result, as plots reach a stable state or have small RMSE at the end of iteration.

Both Fig(3-13) and Fig(3-18) results of inputs with lower saturation are better than those with high saturation. Inputs with high level over-exposure have high mean errors. However, from Fig.(3-12) and Fig.(3-17) we can see that errors of high saturation inputs are similar with the ground truth. We consider it as in these cases the algorithm reconstructs the phase well but the amplitude is multiplied a constant comparing with the object. The over-exposure premodifier highlight the structure of PSF, thus the details that were originally in the shadows emerge, i.e. the size of PSF becomes bigger. The higher the amplitude of phase, the bigger size of PSF. Therefore, when the PSF becomes bigger, the retrieved phase may have larger amplitude. If we only care about the information in the image, then the multiple change of the amplitude does not affect how the phase looks. As shown in Fig.(3-11) and Fig.(3-16), results of high saturation are quite close to the ground truth but with higher contrast.

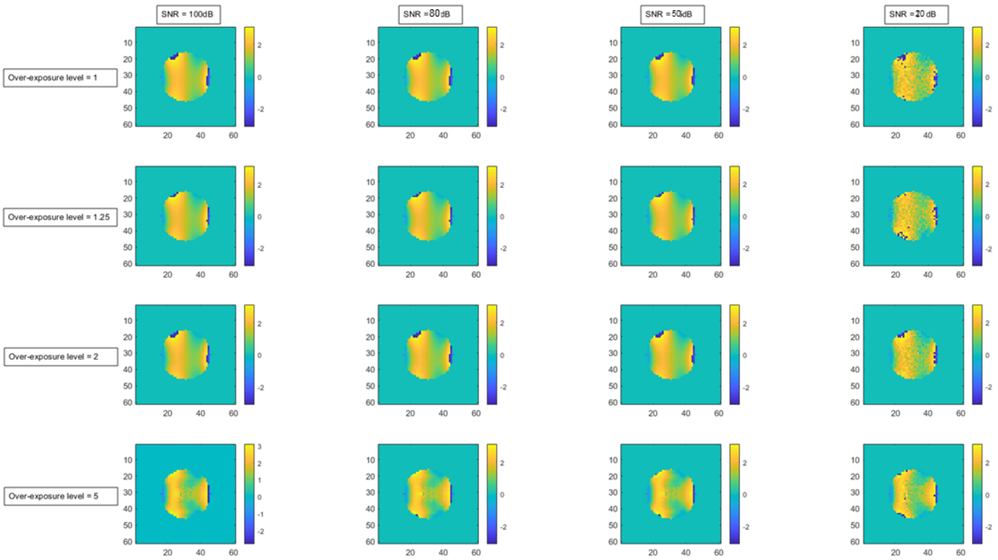


Figure 3-16: The retrieved phases in experiments of object 2 via proposed modified GS algorithm.

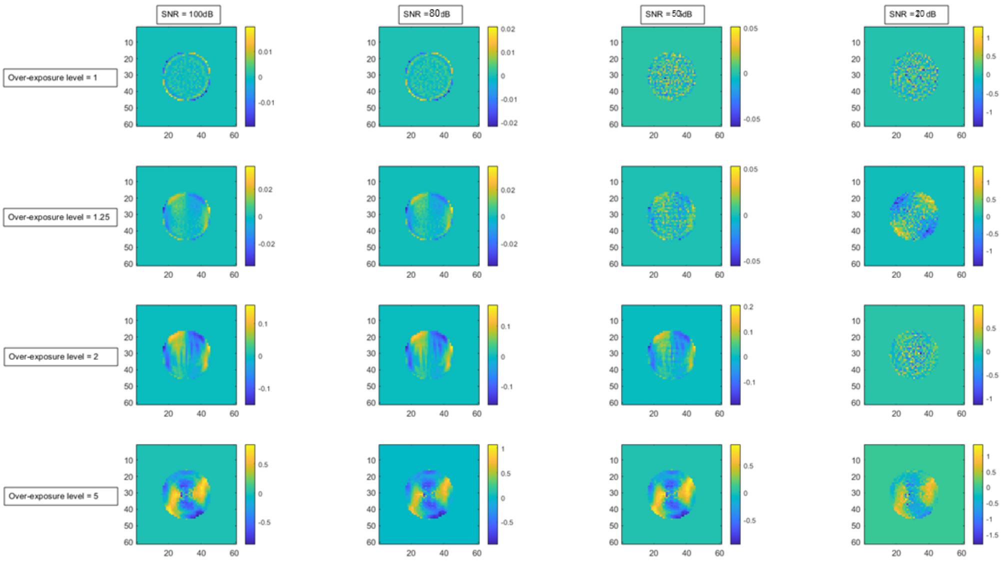


Figure 3-17: The differences between every experiment result and the ground truth Zernike modes. In the 4×4 images array, the experimental input of each row has the same over-exposure level, and the experimental input of each column has the same SNR.

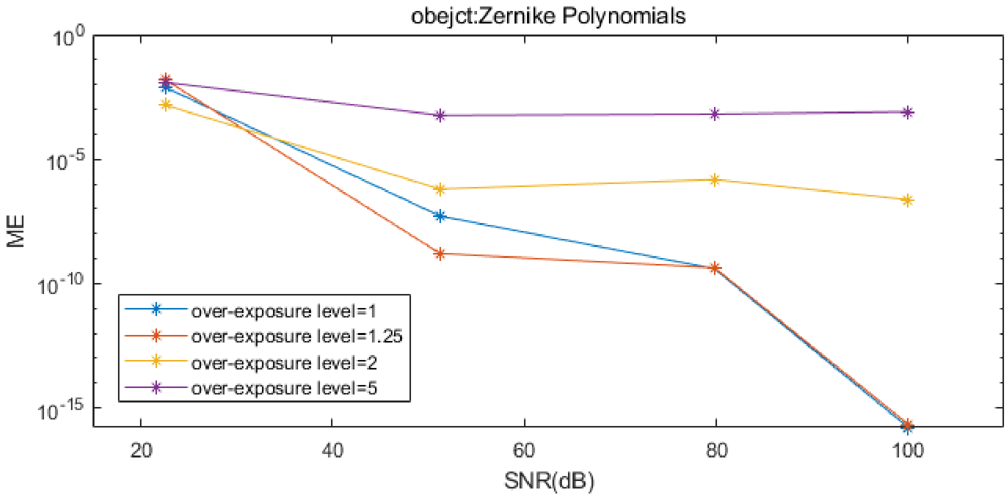


Figure 3-18: Plots of the average error of results about using modified GS algorithm to retrieve object Zernike modes from different noise and over-exposure.

3-2-3 Comparison between Modified GS and Simple GS

To test whether the proposed modified GS algorithm has the ability to deal with the over-exposed and effected by noise PSF, we run the simple GS algorithm on object with phase Lena. Fig(3-20) presents the results of these simulations. Fig.(3-21) and Fig.(3-22) are plots of mean average and PSNR for every experiment.

Generally speaking, performance of the simple GS algorithm for processing input with noise and over-exposure is not as bad as we assumed, as the most results are the same with Fig.(3-11). Only when both noise and over-exposure are in high levels the algorithm cannot converge to a good result.

In Fig.(3-21) and Fig.(3-22), we compare results of two algorithms under the same over-exposure and noise level. Except the case without noise and over-exposure, under the same conditions of noise and overexposure, the results of modified GS algorithm have smaller mean error and larger PSNR than results of simple GS algorithm. This also shows that the modified GS algorithm has indeed achieved our expected goal to a certain extent: it can retrieve the phase from the image disturbed by noise and overexposure.

In this chapter we use PSF images generated with different levels of noise and over-exposure to test the performance of the proposed algorithm. In experiments, the algorithm shows robustness against noise and against low level saturation.

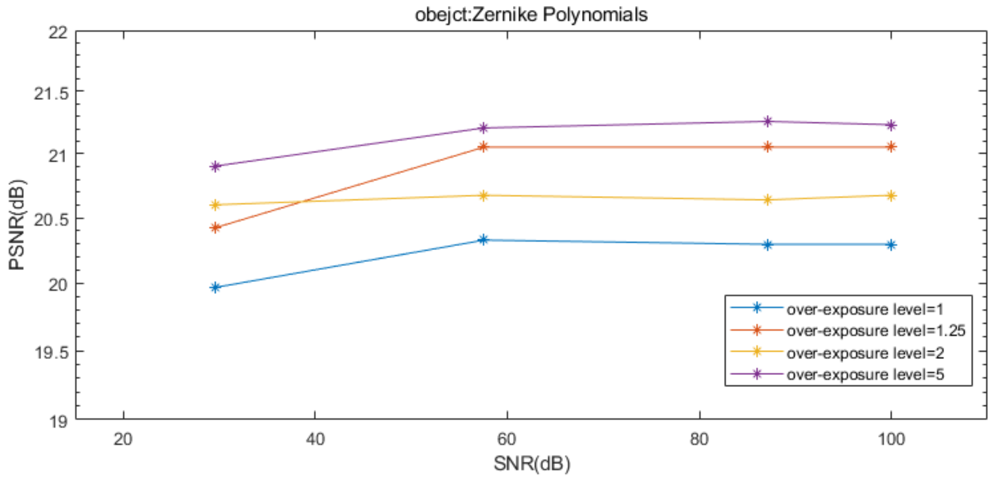


Figure 3-19: Plots of PSNR of the retrieval in every experiment. Every plot has the same over-exposure level. x-axis and y-axis denote SNR in every experiment input and values of PSNR, respectively.

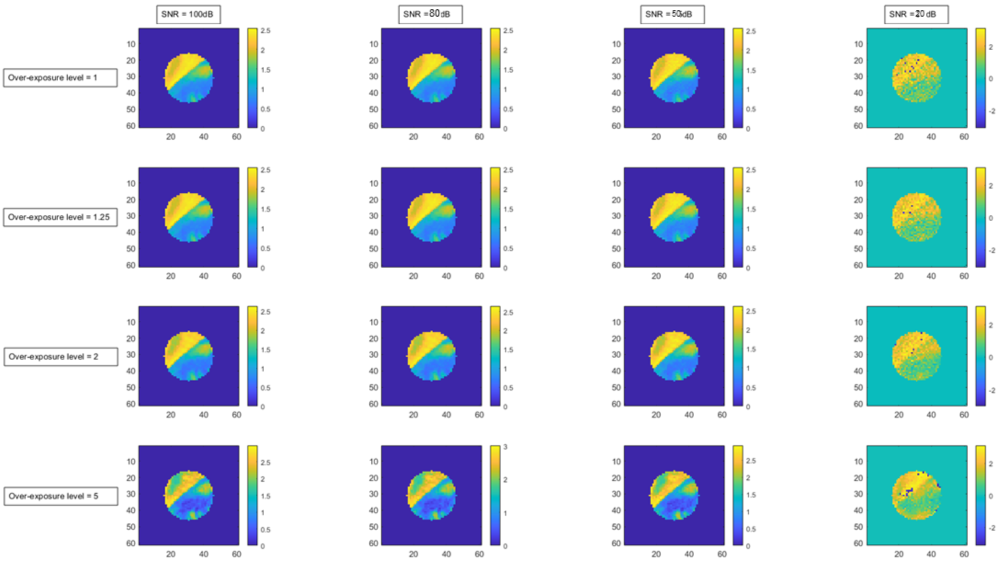


Figure 3-20: The retrieved phases in experiments of object 1 via the original GS algorithm. In the 4×4 images array, the experimental input of each row has the same over-exposure level, and the experimental input of each column has the same SNR.

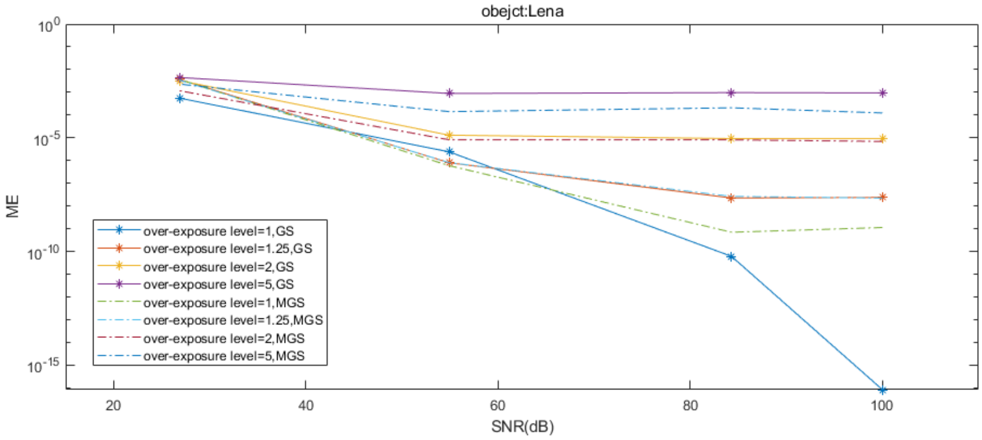


Figure 3-21: Plots of mean error of results about using simple GS algorithm and modified GS algorithm to retrieve phase Lena from different noise and over-exposure. Lines with stars are results of simple GS algorithm while dotted lines are results of modified GS (MGS) algorithm.

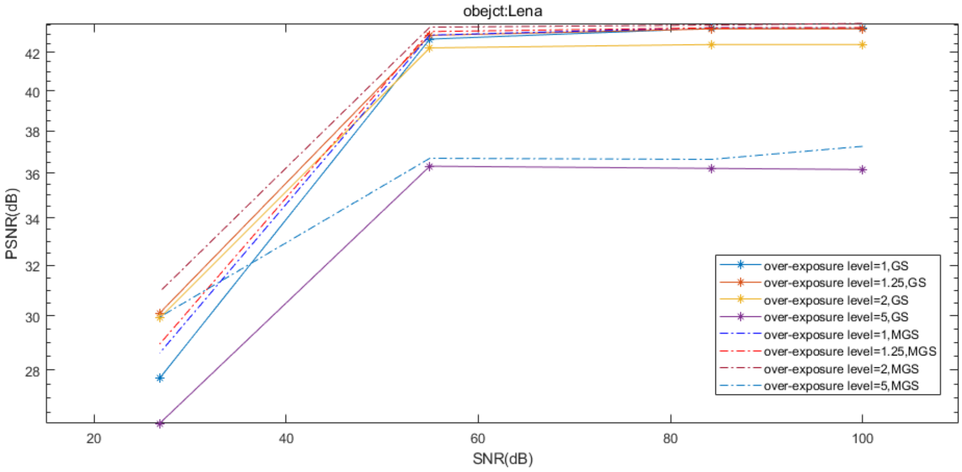


Figure 3-22: PSNR values of retrievals with the ground truth as the reference. Lines with stars are results of simple GS algorithm while dotted lines are results of modified GS (MGS) algorithm.

Chapter 4

Apply Modified GS Algorithm on Real Data

4-1 Experimental Data Source

There are two kinds of PSF data used in the experiments. One are PSFs corresponding to single Zernike modes, which is mainly for deciding and correcting the aperture size of the aperture model in our experiment. Another data are PSFs with different saturation level of an certain object, as shown in Fig.(4-2). All data is obtained with beam diameter approx 10 mm, lens with a focal length of 300mm at wavelength 650nm with IDS UI-1490 camera. The specification of the IDS UI-1490 camera is in Appendix A-5.

In raw data the intensity does not gather at the center of the image, as shown in Fig.(4-1). In order to facilitate the subsequent algorithm operation, first we crop the blank part on the right side of the image, and make the brightest part in the middle of the photo as much as possible. To achieve this, we use a method called *centroid algorithm* to find the center of the intensity. More details about centroid algorithm can be found in the Appendix A-6.

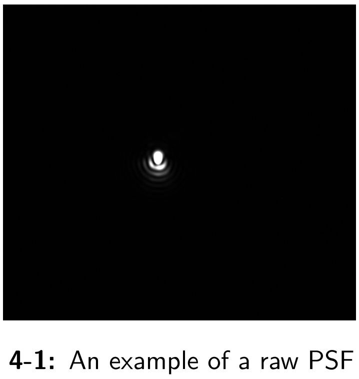


Figure 4-1: An example of a raw PSF image.

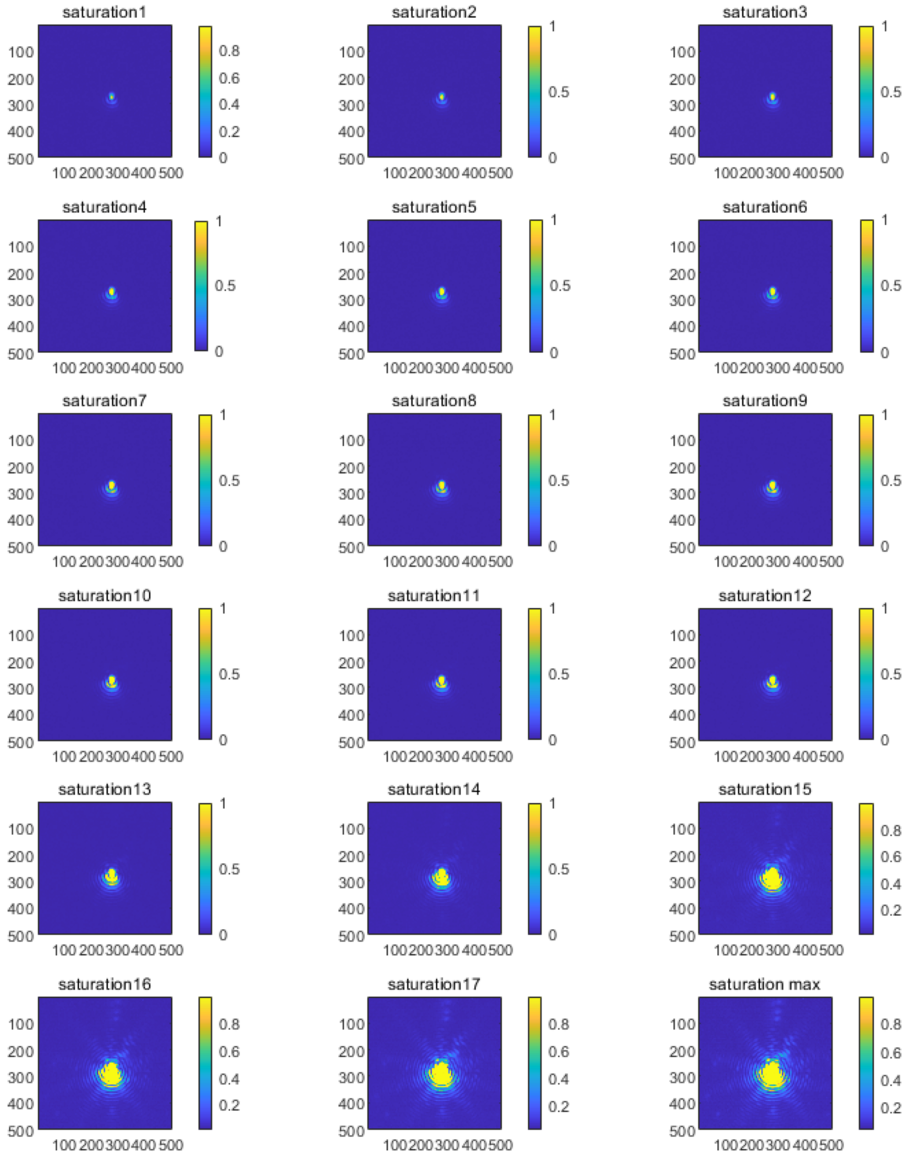


Figure 4-2: Different saturation level PSFs with the same object. The camera is 8-bit and images are normalized. The top-lift image is the original PSF of the object, i.e. no over-exposure. This PSF is named as 'saturation 1', and the saturation of the subsequent images increase in order, so we number them in the order of ascending saturation.

4-2 Aperture Modelling

The aperture decides the real-space amplitude of the PSF, which is a necessary prior information for the algorithm. Hence, the model of aperture function is needed, and it can be achieved by the following steps. Firstly, we calculate the numerical aperture. Numerical aperture is a dimensionless number that characterizes the range of angles over which the system can accept or emit light[23]. Based on Fig.(4-3), the image-space numerical aperture of the lens is

$$\mathbf{NA} = sin\theta \approx \frac{D}{2f} = \frac{10mm}{2 * 300mm} = 0.0167, \tag{4-1}$$

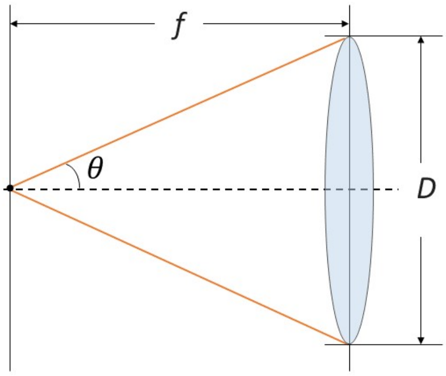


Figure 4-3: Schematic diagram of thin lens numerical aperture(NA). D and f are diameter of beam and focal length of the thin lens, respectively. θ is the angle that relates to NA.

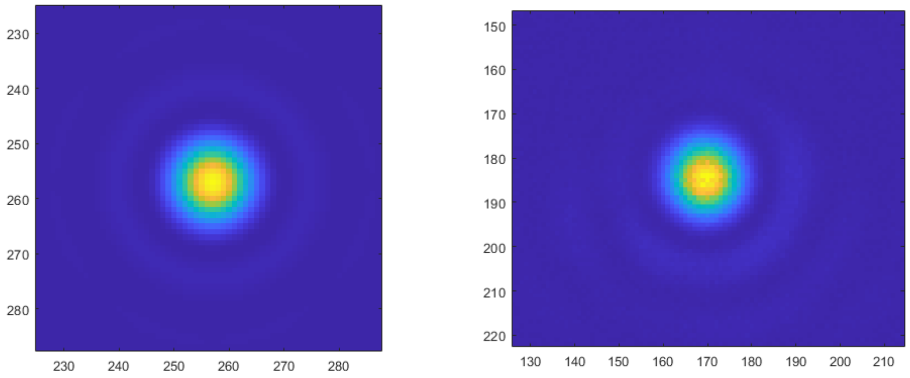
where $\mathbf{NA}$ stands for numerical aperture, D means the diameter of the beam, and f denotes the focal length of lens.

The radius of brightest spot in the PSF with flattened phase and the numerical aperture have a relationship as Eq.(4-2). It is a transform of calculating the angle at which the first minimum occurs in Airy disk. Airy disk, which is mentioned in Section 3-1-1, is the best-focused spot of light that an ideal lens with circle aperture can make limited by the diffraction of light. Fig4-4(b) is an example of Airy disk, as it is the PSF of a spot with zero phase and without any aberration wavefront.

$$r = \frac{1.22\lambda}{2\mathbf{NA}} = 23.7. \tag{4-2}$$

Therefore, the distance between the first minimum and the center of the PSF generated by the modeled aperture with zero phase should be 24 pixels. According to this we model a binary mask with circle aperture. Eq.(4-2) also shows that more orders of diffraction from the object are brought into the lens as a higher numerical aperture is, thus more information it has to form a resultant image.

To verify the correctness of the modeling, we compare the PSF generated with zero phase by the aperture with a flattened Zernike mode PSF. The flattened PSF is considered as an ideal PSF, i.e. no aberration and noise.



(a) PSF generated by modeled aperture with zero phase.

(b) A measured PSF of piston Zernike mode.

Figure 4-4: PSF generated by the modeled aperture and a flattened PSF collected in the lab. The sizes of PSF in two figures are almost the same.

The size of the Airy disk in Fig4-4(a) and Fig4-4(b) are the same, which shows we have a correct modelling of the aperture function. It gives a support to continue the following experiments.

4-3 Results and Analysis of the Experiment on Real Data

We apply our modified GS algorithm on the real PSF data with different level of saturation, as shown in Fig(4-2). Except the first PSF, i.e. the PSF with no over-exposure effect, the others are the inputs to the algorithm. Since in this case we have no information about the true phase of the object, we firstly retrieve a phase from the first PSF and use this result as a basis to measure the results of retrieval from the other PSFs.

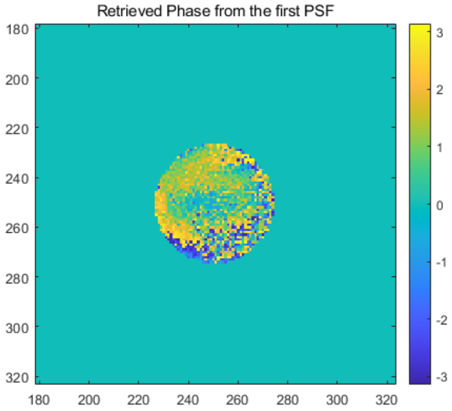


Figure 4-5: The retrieved phase of the first PSF without over-exposure.

Fig(4-5) and Fig(4-6) show the retrieved phase and the corresponding PSF. When we compare the two PSFs in Fig(4-6), we find they have similar diffraction shape. This shows that the

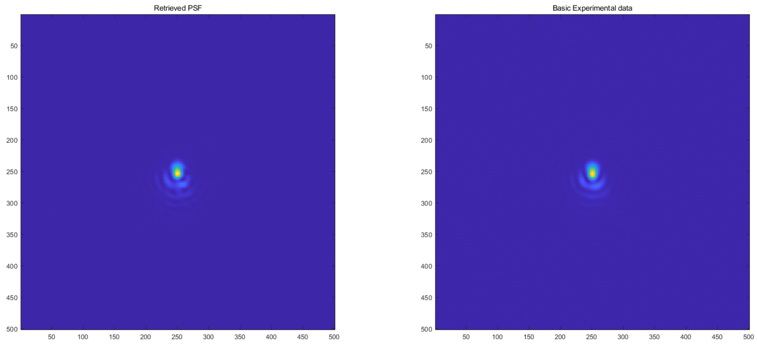


Figure 4-6: The PSF generated by the retrieved phase (left) and the input PSF (right). It can be observed that the shapes of the left and right PSFs are basically the same.

recovered phase we obtained is similar to the actual phase, so we can use the obtained phase as a basis for subsequent reference and comparison.

4-3-1 Case 1: Original PSF is Unknown

In this case we only have one over-exposed PSF information, i.e. we have no idea about the PSF without over-exposure of the same object. As introduced in Chapter 2, in our algorithm the parameter γ is decided as Eq.(2-3).

Fig(4-7) presents the convergence trend of the modified GS algorithm retrieve phases from the other 18 PSFs. At the end of 4000 iterations, though some plots do not reach a stable state, the trend has reached a stage of rapid decline and the values of the final RMSE are small. We consider these experiments finally converge to a result. For experiments whose plot is not even show a rapidly decrease, we suppose that these experiments do not converge. Experiments that fail to converge to the result are those with higher input saturation.

Firstly we compare the retrieved phases with the basis, as presented in Fig.(4-8). Expect the first image, which is regarded as the basis, the other figures show the difference between the phase restored from different saturation level PSFs and the basis. It can be seen that the error is still relatively large, and the difference is mainly in the lower half of the phase.

We reconstruct PSF images using the retrieved phase. Our expectation is that the reconstructed PSF should be similar to the first PSF, so that we could consider that the phase may be close to the actual one. From Fig(4-9) we could see that 'reconstructed saturation 2' and 'reconstructed saturation 3' are close to the basis but with the increase of the saturation, the reconstructed PSFs are more and more different with the reference. The shape of the PSF changes from ellipse to circle, as if the ellipse is cut, and the diffraction pattern becomes "messy" in the end. Starting from saturation 13, the diffraction rings become totally different with the basis, as the algorithm even can not converge to a right result in these cases.

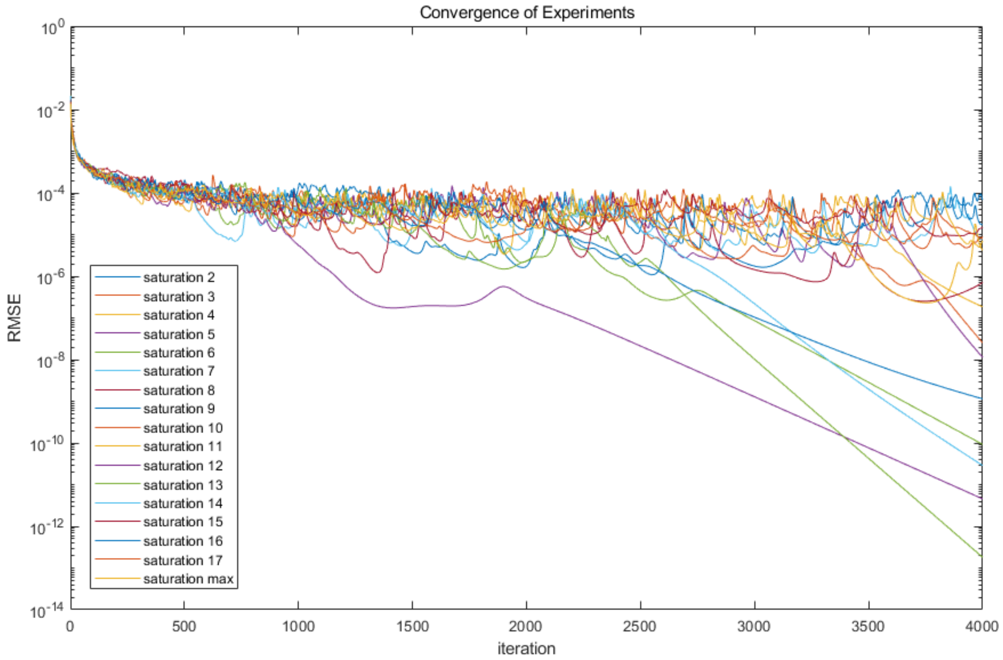


Figure 4-7: The plots of convergence speed of 17 experiments with saturation level from 2 to max. The y- axis is the root mean square of difference of results of every two iterations.

One explain for the change of shape of PSF from ellipse to circle may be the centroid of the PSF is not the centroid of the over-exposed part, which is un-updated and substituted by the product in the iteration. This may cause that the un-updated part not be filled completely then leaves a blank in the part which should be the brightest.

4-3-2 Case 2: Original PSF is Known

Except the over-exposed PSF, in this case we also have the information about the original PSF of the same object. Then we can use the two PSF images to calculate how many times the intensity is enhanced by the over-exposure. This is the second method to decide parameter γ mentioned in Chapter2, as Eq.(2-2).

First we use plots of RMSE to check the convergence state of each experiment, as Fig.(4-10) shown. Similar with Case 1, the algorithm cannot converge successfully when the input PSF with high saturation. This is confirmed in Fig.(4-12). From "reconstructed saturation 15" to "reconstructed saturation max", PSFs are over-exposed and deform.

Using the same way as Fig(4-8), we calculate the error between retrieved phases and the basis phase and obtain Fig.(4-11). Except the last five figures, which do not converge, the rest phase still have large difference with the referenced phase, and the difference become larger with the saturation level increases.

According to Fig.(4-12), we evaluate the function of the algorithm from the similarity between the reconstructed PSFs and the PSF without over-exposure. "reconstructed saturation 2" to

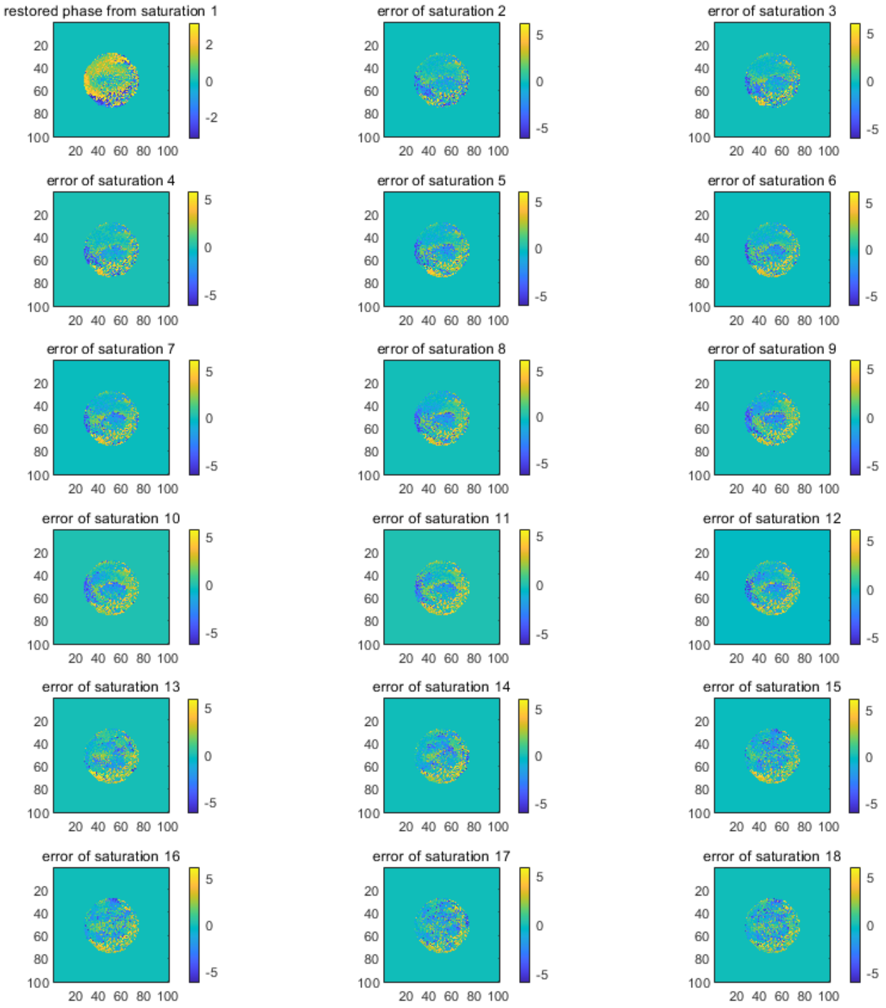


Figure 4-8: The first image is the phase retrieved from saturation 1 in Fig.(4-2). Using the phases retrieved from the rest PSFs in Fig.(4-2) subtract the phase of saturation 1 we get the error between phases.

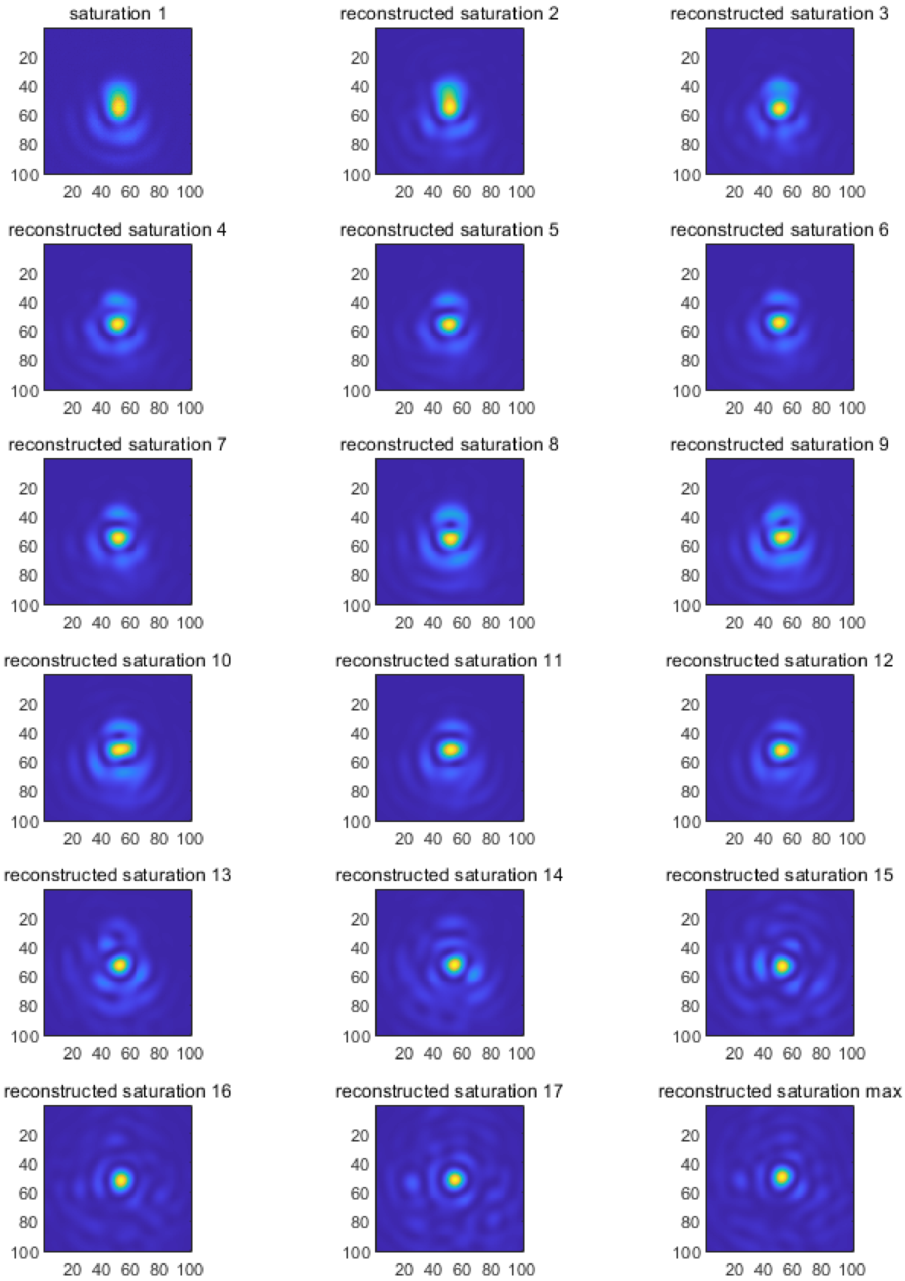


Figure 4-9: PSFs generated by the corresponding retrieved phases. Each phase is restored from the PSF at the same location in Fig.(4-2). By comparing each reconstructed PSF with the first one, we can judge the performance of the algorithm.

"reconstructed saturation 5" have kind of similarity with "saturation 1". However, the size of the central spot shrinks and the diffraction rings are not clear. Starting from "reconstructed saturation 6" the shape of PSFs become distorted. However, the diffraction rings become more and more apparent since "reconstructed saturation 9". Finally, over-exposure occurs in the reconstructed PSF.

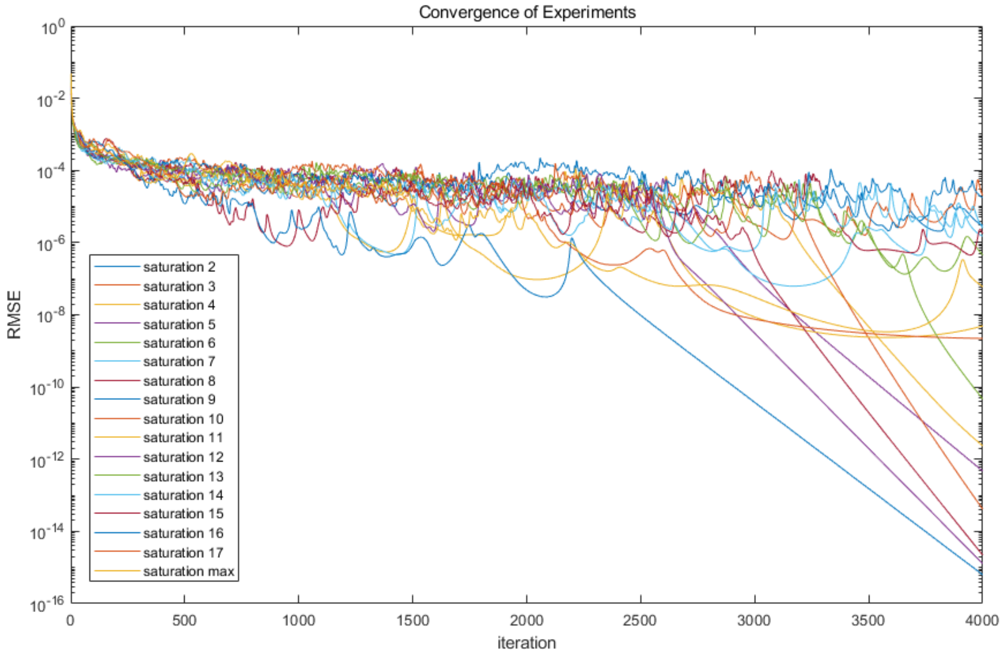


Figure 4-10: The plots of convergence speed of 17 experiments with saturation level from 2 to max. The y- axis is the root mean square of difference of results of every two iterations.

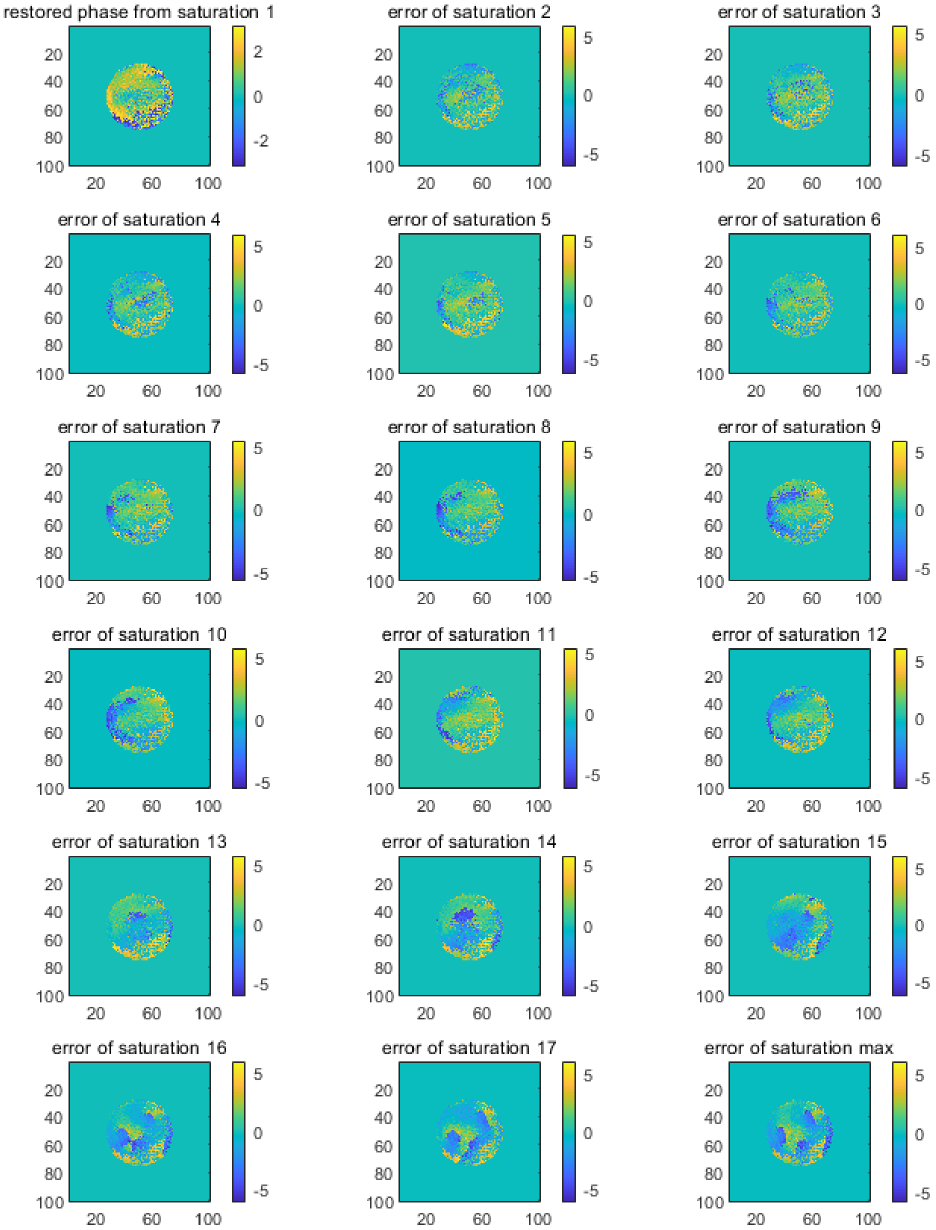


Figure 4-11: The first image is the phase retrieved from saturation 1 in Fig.(4-2). Using the phases retrieved from the rest PSFs in Fig.(4-2) subtract the phase of saturation 1 we get the error between phases.

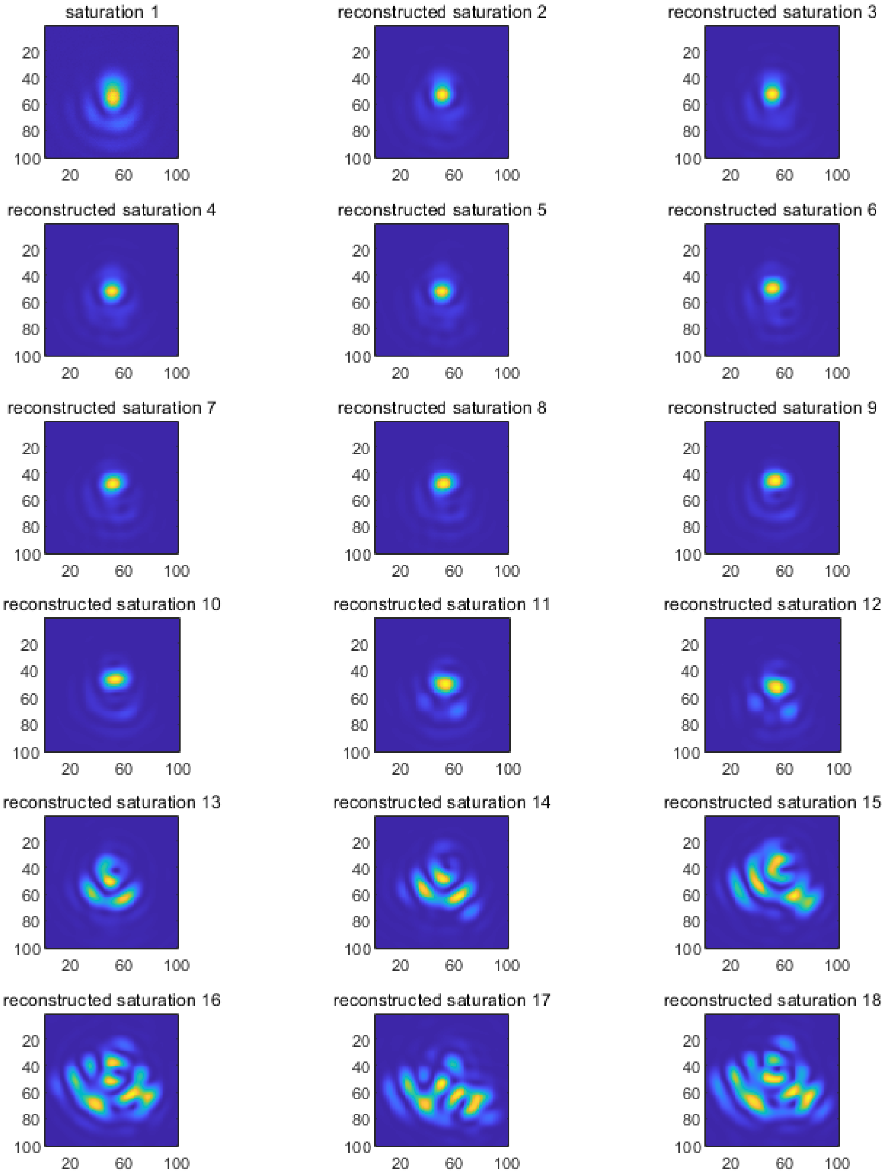


Figure 4-12: PSFs generated by the corresponding retrieved phases. Each phase is restored from the PSF at the same location in Fig.(4-2). By comparing each reconstructed PSF with the first one, we can judge the performance of the algorithm.

Chapter 5

Conclusions and Future Work

The motivation of the research is to implement an over-exposure premodifier on phase retrieval problem and using the classic solution of phase retrieval problem, i.e. Gerchberg-Saxton algorithm to restore phase. The goal of the thesis is to modify the simple GS algorithm so that the algorithm can deal with the input with over-exposure premodifier and test whether the over-exposure can improve the performance of the algorithm when it acts as a premodifier.

In this thesis we proposed a modified GS algorithm that can remove background bias and noise in PSF image and retrieve phase from over-exposed images. In Chapter3, we simulate numerical experiments. Using two kinds of object, Lena and Zernike modes as phase to generate PSF images and set four different levels of noise and saturation. Based on results of 32 experiments (4 noise levels, 4 over-exposure levels and 2 kinds of object), we consider that low level over-exposure premodifier has a positive effect on improving the robustness against low level noise(SNR above than 60dB) of GS algorithm. But high level over-exposure may be negative for phase retrieval, since the information covered by over-exposure are more than the details highlighted by the premodifier.

In Chapter4 we apply our modified GS algorithm on real data. We consider two cases: the original PSF is known and the original PSF is unknown. For each case we design a constraint on Fourier magnitude so that we can use the information that highlight by the over-exposure while ignore the part that covered because of over-exposure. 18 experiments of the different levels saturated PSFs are done and the results show that the modified GS algorithm can only deal with PSFs with low level over-exposure level. When the input PSF suffers from high level saturation, the algorithm cannot converge to a right solution.

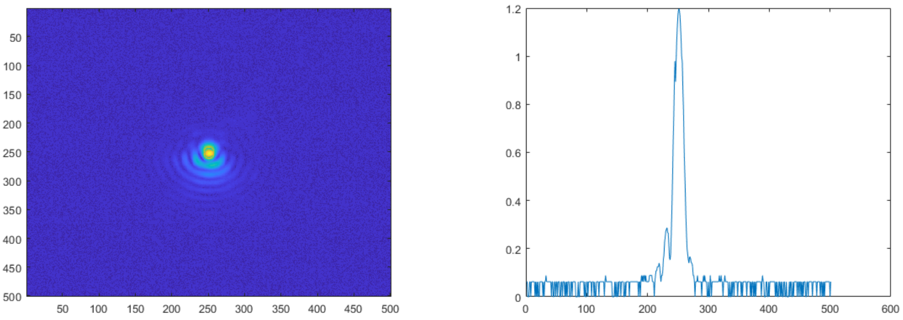
Although compared with the simple GS algorithm, the proposed modified GS algorithm has stronger robustness against noise and over-exposure, there are still some drawbacks of the algorithm. The algorithm has good results in numerical simulation experiments, but it has limitations for real data. The real data may have more complex phase and the PSF shape. Now our choice about the parameter that adjusts magnification of the un-update part to keep it at the same order of magnitude as other data is still rough and has limited scope of application. The future research direction can be refined selection of this parameter. Besides, we combine a 3D phase diversity method to remove the background noise in PSF. We may consider further to use different over-exposure level PSFs as the diversity and modify this method to solve phase retrieval problem.

Appendix A

Appendix:Fundamental Concepts

A-1 An approach for Parameter γ calculation

The idea of this approach to decide parameter γ is to calculate the mean values of the inside and outside pixels around the junction. The main function of parameter γ is to enlarge or shorten values of the part where the Fourier constraints are not be imposed, so that this part can smoothly connect the area where the amplitude is updated, and become the area with the highest intensity in the whole picture.



(a) Fourier domain constrain image.

(b) Section view of the Fourier domain constrain.

Figure A-1: Figures about Fourier constrain and its section view when using γ calculated by Eq.(A-1).

$$\gamma = \frac{\bar{O}_c}{\bar{I}_c}, \tag{A-1}$$

where $\bar{O}_c$ denotes the mean value of the outside pixels and $\bar{I}_c$ denotes the mean value of the inside pixels.

Compare with the situation that no parameter γ is implemented, the approach for γ calculation has a good effect. However, the drawback and limitation of the method is that only a small number of pixels are used in the calculation. This not only leads to a lack of credibility in the calculated results, but also wastes a lot of other information.

A-2 Gaussian Noise

The density function of univariate Gaussian noise n, which is shown as Fig.(A-2), with mean μ and variance σ^2 is

$$p_n(x) = \left(2\pi\sigma^2\right)^{-1/2} e^{-(x-\mu)^2/2\sigma^2} \tag{A-2}$$

for $-\infty < x < \infty$.

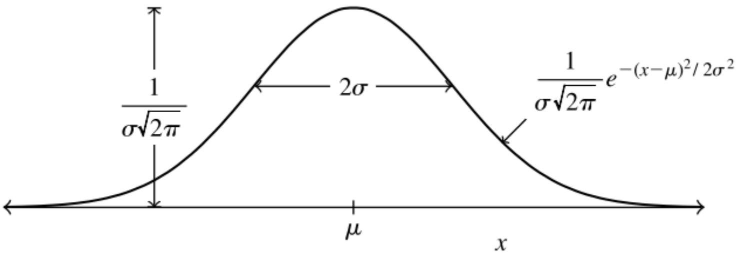


Figure A-2: The Gaussian density.

In situations where v is a random vector, the multivariate Gaussian density becomes

$$p_{\mathrm{v}}(v) = (2\pi)^{-n/2} |\Sigma|^{-1/2} e^{-(v-\mu)^T \Sigma^{-1} (v-\mu)/2}, \tag{A-3}$$

where $\mu = E[\mathrm{v}]$ is a mean vector and $\Sigma = E[(\mathrm{v}-\mu)(\mathrm{v}-\mu)^T]$ is the covariance matrix.

A-3 Zernike Polynomials

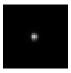
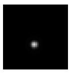
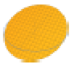
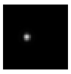
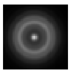
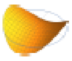
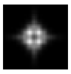
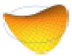
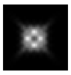
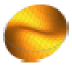
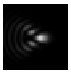
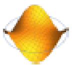
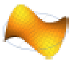
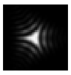
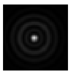
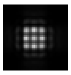
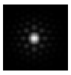
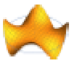
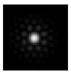
Index	Noll's ordering	Name	Expression	Shape	PSF
$Z_0^0(x,y)$	1	piston	1		
$Z_1^1(x,y)$	2	tip	x		
$Z_1^{-1}(x,y)$	3	tilt	y		
$Z_2^0(x,y)$	4	defocus	$2x^2+2y^2-1$		
$Z_2^2(x,y)$	5	astigmatism	x^2-y^2		
$Z_2^{-2}(x,y)$	6	astigmatism	$2xy$		
$Z_3^1(x,y)$	7	coma	$3x^3+3y^2x-2x$		
$Z_3^{-1}(x,y)$	8	coma	$3y^3+3x^2y-2y$		
$Z_3^3(x,y)$	9	trefoil	x^3-3xy^2		
$Z_3^{-3}(x,y)$	10	trefoil	$3x^2y-y^3$		
$Z_4^0(x,y)$	11	spherical	$6x^4+12y^2x^2-6x^2+6y^4-6y^2+1$		
$Z_4^2(x,y)$	12		$4x^4-3x^2-4y^4+3y^2$		
$Z_4^{-2}(x,y)$	13		$8yx^3+8y^3x-6yx$		
$Z_4^4(x,y)$	14		$x^4-6y^2x^2+y^4$		
$Z_4^{-4}(x,y)$	15		$4x^3y-4xy^3$		

Figure A-3: The first 15 order Zernike modes.

A-4 Signal-to-noise Ratio (SNR)

Signal-to-noise Ratio is a measure used in science and engineering that compares the level of a desired signal to the level of background noise. SNR is defined as the ratio of signal power to the noise power, often expressed in decibels. When the ratio is greater than $1 : 1$, i.e. SNR is higher than $0dB$, signal is more than noise.

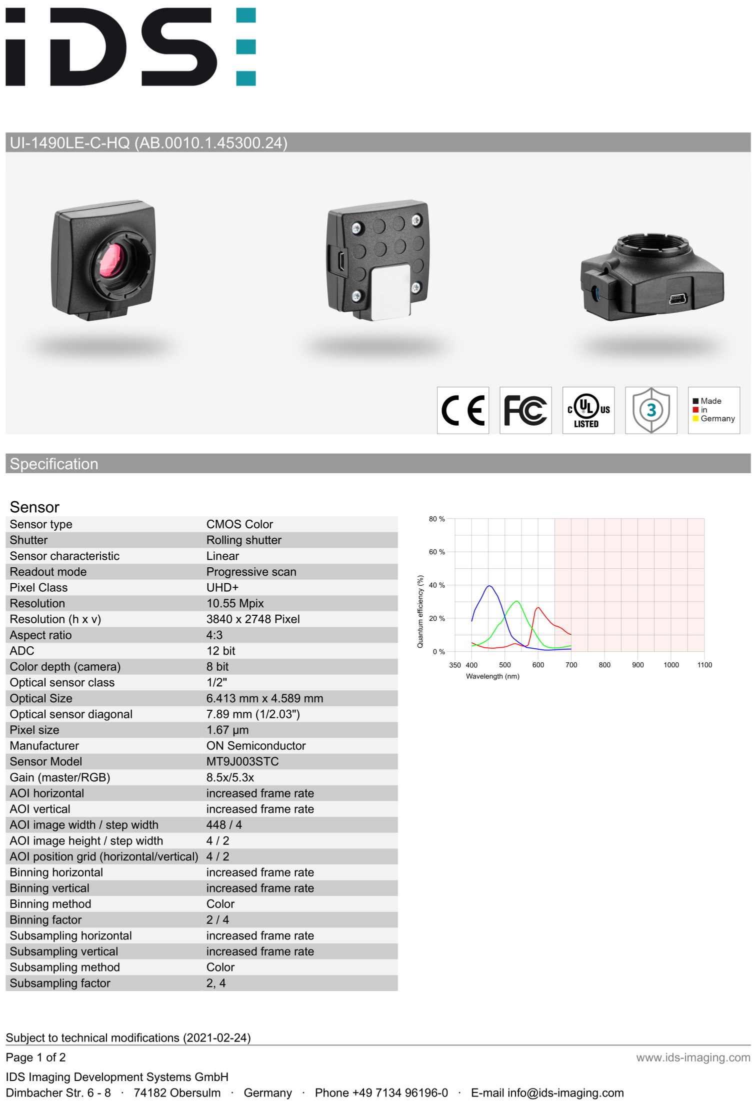
The definition of SNR is the ratio of the power of a signal (meaningful input) to the power of background noise (meaningless or unwanted input), which can be expressed as

$$\mathbf{SNR} = \frac{P_{signal}}{P_{noise}} \tag{A-4}$$

Because many signals have a very wide dynamic range, signals are often expressed using the logarithmic decibel scale. Based upon the definition of decibel, SNR may be expressed in decibels as

$$\mathbf{SNR}_{\mathrm{dB}} = 10\mathbf{log}_{10}(\frac{P_{signal}}{P_{noise}}) \tag{A-5}$$

A-5 Specification of IDS UI-1490 Camera



iDS
CE
FC
c UL us LISTED
3
Made
in
Germany
80 %
60 %
40 %
20 %
0 %
Quantum efficiency (%)
350
400
500
600
700
800
900
1000
1100
Wavelength (nm)

UI-1490LE-C-HQ (AB.0010.1.45300.24)

Specification

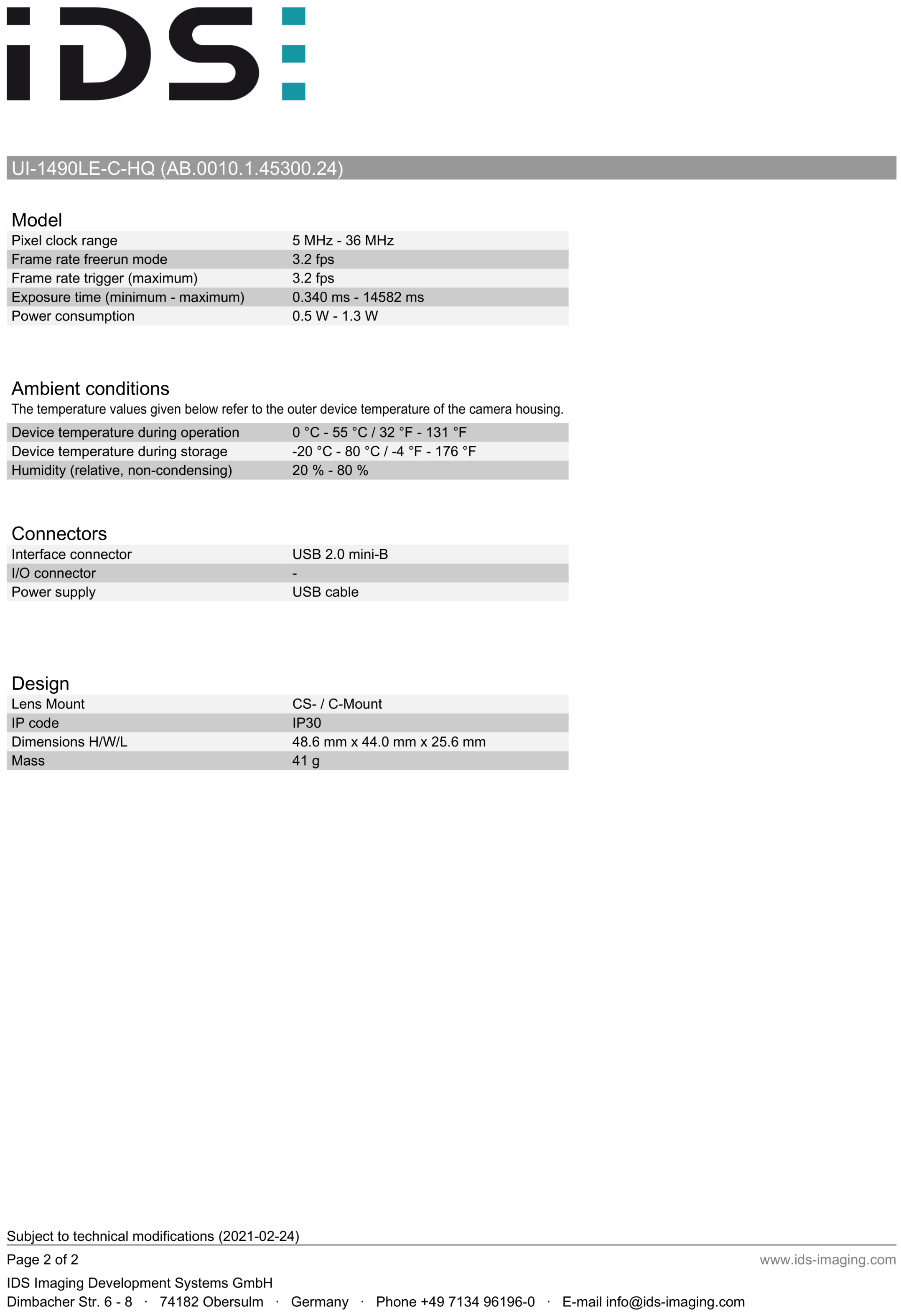
Sensor

Sensor type	CMOS Color
Shutter	Rolling shutter
Sensor characteristic	Linear
Readout mode	Progressive scan
Pixel Class	UHD+
Resolution	10.55 Mpix
Resolution (h x v)	3840 x 2748 Pixel
Aspect ratio	4:3
ADC	12 bit
Color depth (camera)	8 bit
Optical sensor class	1/2"
Optical Size	6.413 mm x 4.589 mm
Optical sensor diagonal	7.89 mm (1/2.03")
Pixel size	1.67 µm
Manufacturer	ON Semiconductor
Sensor Model	MT9J003STC
Gain (master/RGB)	8.5x/5.3x
AOI horizontal	increased frame rate
AOI vertical	increased frame rate
AOI image width / step width	448 / 4
AOI image height / step width	4 / 2
AOI position grid (horizontal/vertical)	4 / 2
Binning horizontal	increased frame rate
Binning vertical	increased frame rate
Binning method	Color
Binning factor	2 / 4
Subsampling horizontal	increased frame rate
Subsampling vertical	increased frame rate
Subsampling method	Color
Subsampling factor	2, 4

Subject to technical modifications (2021-02-24)

Page 1 of 2 www.ids-imaging.com

IDS Imaging Development Systems GmbH
Dimbacher Str. 6 - 8 · 74182 Obersulm · Germany · Phone +49 7134 96196-0 · E-mail info@ids-imaging.com



UI-1490LE-C-HQ (AB.0010.1.45300.24)

Model

Pixel clock range	5 MHz - 36 MHz
Frame rate freerun mode	3.2 fps
Frame rate trigger (maximum)	3.2 fps
Exposure time (minimum - maximum)	0.340 ms - 14582 ms
Power consumption	0.5 W - 1.3 W

Ambient conditions

The temperature values given below refer to the outer device temperature of the camera housing.

Device temperature during operation	0 °C - 55 °C / 32 °F - 131 °F
Device temperature during storage	-20 °C - 80 °C / -4 °F - 176 °F
Humidity (relative, non-condensing)	20 % - 80 %

Connectors

Interface connector	USB 2.0 mini-B
I/O connector	-
Power supply	USB cable

Design

Lens Mount	CS- / C-Mount
IP code	IP30
Dimensions H/W/L	48.6 mm x 44.0 mm x 25.6 mm
Mass	41 g

Subject to technical modifications (2021-02-24)

Page 2 of 2 www.ids-imaging.com

IDS Imaging Development Systems GmbH
Dimbacher Str. 6 - 8 · 74182 Obersulm · Germany · Phone +49 7134 96196-0 · E-mail info@ids-imaging.com

A-6 Centroid Algorithm

Centroid algorithm is a method to find the location of the center. For a 2-dimension PSF image, the position of the center can be calculated by following formula:

$$
\begin{aligned}
c_x &= \frac{\sum(I(x,y) \times x)}{\sum I(x,y)}, \\
c_y &= \frac{\sum(I(x,y) \times y)}{\sum I(x,y)},
\end{aligned} \tag{A-6}
$$

where x and y are the horizontal and vertical coordinates of the point, respectively. $I(x,y)$ denotes the intensity of the point (x,y). Therefore, (c_x, c_y) is the location of the center.

Bibliography

[1] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, “Phase retrieval with application to optical imaging: A contemporary overview,” *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 87–109, 2015.

[2] R. W. Gerchberg, “A practical algorithm for the determination of phase from image and diffraction plane pictures,” *Optik*, vol. 35, pp. 237–246, 1972.

[3] J. R. Fienup, “Phase retrieval algorithms: a comparison,” *Applied optics*, vol. 21, no. 15, pp. 2758–2769, 1982.

[4] I. Ziskind and M. Wax, “Maximum likelihood localization of multiple sources by alternating projection,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 10, pp. 1553–1560, 1988.

[5] Y. Nishizaki, M. Valdivia, R. Horisaki, K. Kitaguchi, M. Saito, J. Tanida, and E. Vera, “Deep learning wavefront sensing,” *Optics express*, vol. 27, no. 1, pp. 240–251, 2019.

[6] E. J. Akutowicz, “On the determination of the phase of a fourier integral, i,” *Transactions of the American Mathematical Society*, vol. 83, no. 1, pp. 179–192, 1956.

[7] E. J. Akutowicz, “On the determination of the phase of a fourier integral, ii,” *Proceedings of the American Mathematical Society*, vol. 8, no. 2, pp. 234–238, 1957.

[8] Y. M. Bruck and L. Sodin, “On the ambiguity of the image reconstruction problem,” *Optics communications*, vol. 30, no. 3, pp. 304–308, 1979.

[9] J. R. Fienup, “Reconstruction of an object from the modulus of its fourier transform,” *Optics letters*, vol. 3, no. 1, pp. 27–29, 1978.

[10] R. A. Gonsalves and R. Chidlaw, “Wavefront sensing by phase retrieval,” in *Applications of Digital Image Processing III*, vol. 207, pp. 32–39, International Society for Optics and Photonics, 1979.

[11] R. A. Gonsalves, “Phase retrieval and diversity in adaptive optics,” *Optical Engineering*, vol. 21, no. 5, p. 215829, 1982.

[12] A. Sinha, J. Lee, S. Li, and G. Barbastathis, “Lensless computational imaging through deep learning,” *Optica*, vol. 4, no. 9, pp. 1117–1125, 2017.

[13] C. A. Metzler, P. Schniter, A. Veeraraghavan, and R. G. Baraniuk, “prdeep: Robust phase retrieval with a flexible deep network,” *arXiv preprint arXiv:1803.00212*, 2018.

[14] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, “Phase retrieval with application to optical imaging: a contemporary overview,” *IEEE signal processing magazine*, vol. 32, no. 3, pp. 87–109, 2015.

[15] J. Fienup and C. Wackerman, “Phase-retrieval stagnation problems and solutions,” *JOSA A*, vol. 3, no. 11, pp. 1897–1907, 1986.

[16] P. Patidar, M. Gupta, S. Srivastava, and A. K. Nagawat, “Image de-noising by various filters for different noise,” *International journal of computer applications*, vol. 9, no. 4, pp. 45–50, 2010.

[17] A. J. Wathen, “Preconditioning,” *Acta Numerica*, vol. 24, 2015.

[18] J. N. Mait, R. D. Martin, C. A. Schuetz, S. Shi, D. W. Prather, P. F. Curt, and J. Bonnett, “Minimum bias design for a distributed aperture millimeter wave imager,” in *2013 IEEE Global Conference on Signal and Information Processing*, pp. 711–714, IEEE, 2013.

[19] O. Soloviev, N. H. Thao, M. Verhaegen, and G. Vdovin, “New dimension for the phase retrieval problem,” in *Imaging and Applied Optics Congress*, p. OF4B.5, Optical Society of America, 2020.

[20] N. H. Thao, O. Soloviev, and M. Verhaegen, “Phase retrieval based on the vectorial model of point spread function,” *JOSA A*, vol. 37, no. 1, pp. 16–26, 2020.

[21] G. B. Airy, “On the diffraction of an object-glass with circular aperture,” *Transactions of the Cambridge Philosophical Society*, vol. 5, p. 283, 1835.

[22] C. Boncelet, “Image noise models,” in *The essential guide to image processing*, pp. 143–167, Elsevier, 2009.

[23] “Numerical aperture and resolution.” https://www.gonda.ucla.edu/bri_core/na.htm, 2007.