Wavenumber Error in Interference Pattern Lensless Imaging

Benjamin Whetten

A senior thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Bachelor of Science

Richard Sandberg, Advisor

Department of Physics and Astronomy

Brigham Young University

ABSTRACT

Wavenumber Error in Interference Pattern Lensless Imaging

Benjamin Whetten
Department of Physics and Astronomy, BYU
Bachelor of Science

A lensless imaging method known as Interference Pattern Structured Illumination Imaging (IPSII) is discussed. Because it does not require a lens, IPSII has a higher theoretical resolution limit than traditional optical microscopes. However, the quality of IPSII images is currently limited by mechanical noise in its mirror movements. This thesis characterizes the resulting distortions caused by uncertainty in IPSII mirror movements. To do so, a model is presented that simulates the propagation of mechanical noise in the experimental IPSII setup to the resulting image data. It is shown that this noise causes errors in both the phase and amplitude of the Fourier transform of the image. Finally, a method to partially correct the resulting image distortions using phase retrieval algorithms in presented.

Keywords: lensless imaging, structured illumination, phase retrieval

Contents

Chapter 1

Introduction

1.1 Lensless Imaging

Traditional microscopes have a limited resolution due to aperture effects introduced by optical lenses. The physical aperture created by the lens in these systems means that the resolution is diffraction limited according to the Raleigh criterion. Additionally, lenses place constraints on the depth of field and field of view of the resulting images. In our lab, we are currently studying a lensless imaging method known as Interference Pattern Structured Illumination Imaging (IPSII). Because it does not require a lens, IPSII is not subject to the same constraints on resolution and depth of field that limit traditional lens-based microscopes. Instead, IPSII is limited by the numerical aperture determined by the beam splitters used in the setup, which allows for higher resolutions than what is achievable with a lens. Additionally, IPSII is well suited for imaging in the UV and X-ray regimes where it is difficult to manufacture quality lenses for use with traditional microscopy techniques.

Another benefit of IPSII is that it uses simple components. For example, IPSII uses only flat optics which are easier to manufacture than lenses and curved mirrors. Additionally, IPSII only requires two single pixel detectors instead of a large and expensive detector array. Other existing lensless imaging techniques offer many of the same advantages as IPSII but require more expensive and complex optical components. For example, Feldkhun's [1] Doppler Encoded Excitation Patterning (DEEP) is a more complex method of structured illumination microscopy. DEEP requires an acoustic-optic modulator to generate the structured patterns needed to illuminate the target and construct an image. Coherent Diffractive Imaging (CDI), on the other hand, requires a large sensitive photodetector array to accurately retrieve the high-frequency components of an image [2]. IPSII is simpler than other structured illumination and lensless imaging techniques, making it more suited to use with short wavelength radiation.

1.2 Experimental IPSII Setup

To reconstruct an image of the target object, IPSII projects a series of interference patterns onto an object to collect its spatial frequency data. As shown in Fig. 1.1, these interference patterns

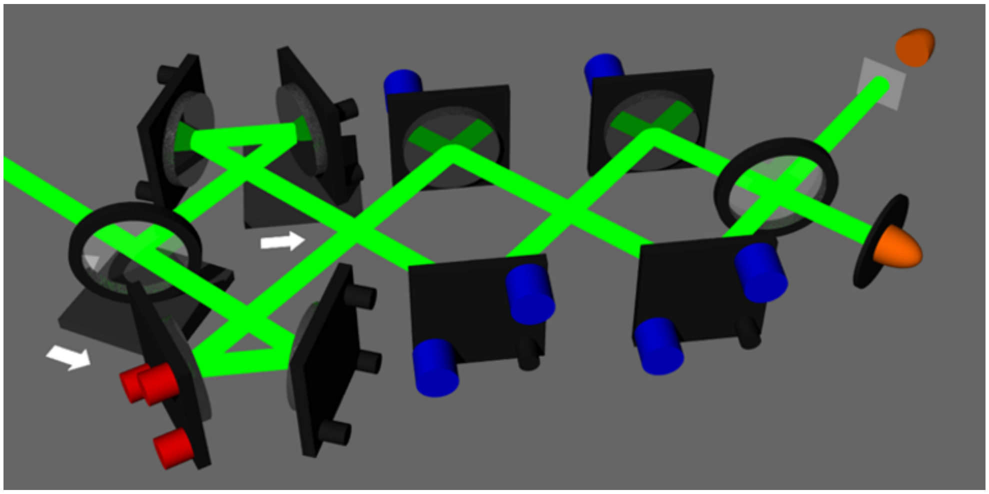


Figure 1.1 Experimental IPSII setup. The target is placed in the gray plane in the upper right, the two orange cones represent single pixel photo-detectors, the blue cylinders represent motorized rotation stages, and the red cylinders represent a piezo-driven translation stage.

are created using a modified Mach-Zehnder Interferometer. A single pixel photodetector is placed behind the target and is used to measure the total intensity of the transmitted light for that specific interference pattern. By rotating the various mirrors in the interferometer, we scan the spatial frequency of the interference pattern and again measure the total intensity of the transmitted light through the target object for each of these frequencies. To obtain the phase of the interference pattern, we place a pinhole and a second photo-detector in the path of the second interferometer arm. By using a pinhole smaller than the width of a single interference fringe, we are able to use the data from this second detector to determine the phase of the pattern.

By scanning the spatial frequency of the interference pattern in both the x and y directions, we are able to construct a two dimensional discrete Fourier transform of the target. The k_x and k_y wavenumbers for the discrete Fourier transform of the object are equivalent to the spatial wavenumbers that characterize the interference patterns used to illuminate it. The amplitude at each point in the transform is simply the square root of the intensity of the light transmitted through the object. Finally, the phase for the transform data is found using the phase data measured by the detector placed after the pinhole in the secondary arm of the interferometer. After constructing the Fourier transform of the target this way, we obtain an image by simply carrying out an inverse Fast Fourier Transform (FFT) on the data. Further details on the IPSII method and experimental setup are given by Jackson and Durfee [3].

1.3 Previous Work at BYU

Previously in our lab, Jarom Jackson demonstrated the feasibility of IPSII by constructing the experimental setup and obtaining images with resolution on the micron scale [4]. An example of one of these experimental images is shown in Fig. 1.2. When analyzing the initial experimental images, we observed that all the images had significant ghosting effects. Ghosting refers to sections

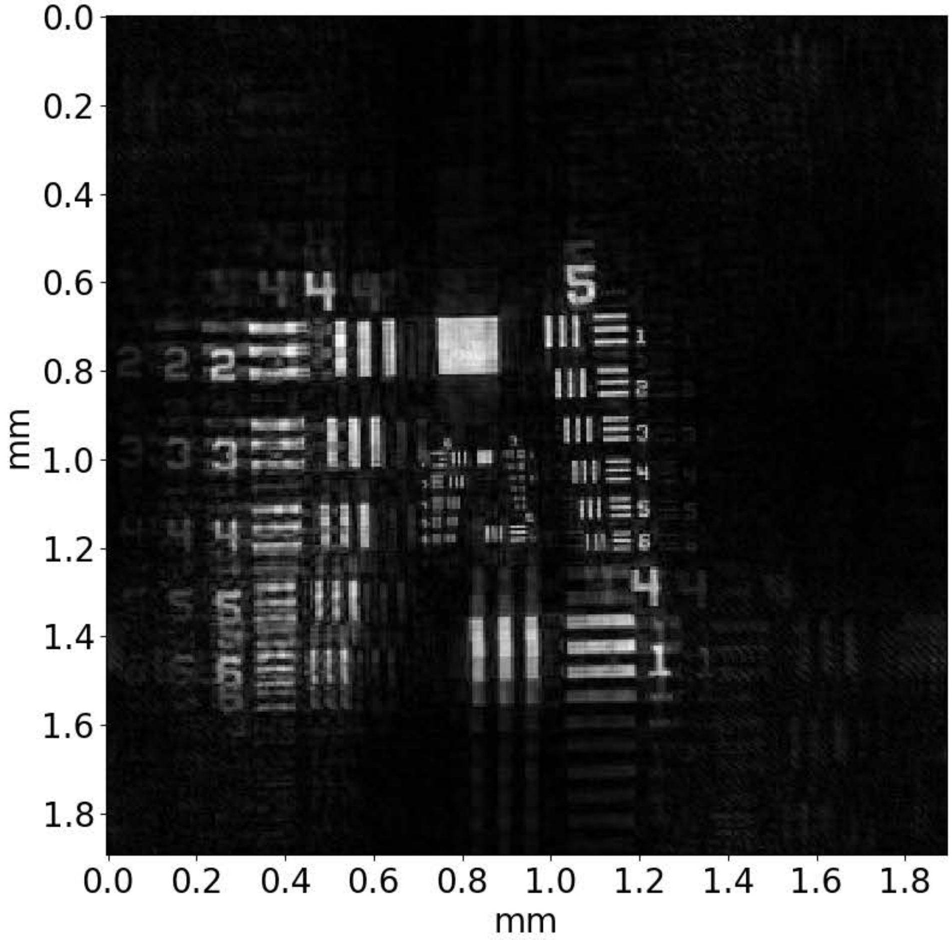


Figure 1.2 IPSII image of the 1951 USAF resolution test chart. Note the ghosting seen around the outside of the image, particularly on the left side.

of the target object being shown in more than one place in the image. This ghosting is strongest near the edges of the IPSII images and is minimal at the center. Jackson theorized that this ghosting and other distortions in the images were due to inaccuracies in the movements of the motorized mounts used to control the interferometer.

Currently, two other undergraduate students are also conducting research on IPSII. Dallen Petersen is characterizing the motorized mirror mounts used in our lab's IPSII interferometer. His results will determine if the custom 3D-printed mirror mounts used in IPSII are precise enough to yield quality images. Additionally, Carter Day is currently studying the capability of IPSII to image objects with significant depth. His work will demonstrate if IPSII is able to accurately image three dimensional objects that cast shadows in the interference patterns.

1.4 Overview

I present a method of simulating the effect of mirror movement errors in IPSII to demonstrate that they could be responsible for the ghosting and blur seen in the experimental images. Based on the results of these simulations, I show that these mirror inaccuracies cause errors in both the amplitude and phase of the image in the Fourier domain. By analyzing the impact of these errors on the data in image space, I argue that the effects of the phase error dominate over the effects of the amplitude error. Finally, I demonstrate that common phase retrieval algorithms can be applied to IPSII to significantly reduce ghosting effects by correcting the errors in the phase of the image.

Chapter 2

Method

In this chapter, I first discuss a method to computationally simulate wavenumber error in IPSII images. I then use this simulation model to show that wavenumber error in IPSII images will cause distortions in the phase and amplitude of the resulting Fourier transform. Finally, I discuss methods to repair the resulting phase errors through phase retrieval algorithms.

2.1 Simulation of Wavenumber Error

To improve the quality of IPSII images, the source of the radially dependent error and ghosting in experimental results must be determined. The current hypothesis is that this error is caused by uncertainties in the interferometer mirror movements. To test our hypothesis, we present a model of the wavenumber error caused by uncertain mirror movements. I treat the uncertainty in the mirror movements as an uncertainty in the sampling of the true Fourier transform of the target object. The experimental process of IPSII is mathematically equivalent to finding the centered discrete Fourier transform of the imaged object on a size N by M array:

$$\mathscr{F}_{k_x,k_y} = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} F_{n,m} \exp\left[-i2\pi\left(\frac{(k_x - \frac{N-1}{2})(n - \frac{N-1}{2})}{N} + \frac{(k_y - \frac{M-1}{2})(m - \frac{M-1}{2})}{M}\right)\right] \tag{2.1}$$

Here, F is the original object in image space, n and m are the pixel indices of this image, and k_x and k_y are the indices in wavenumber space.

The discrete Fourier transform normally requires evenly spaced data points; however the uncertainty in mirror movements in IPSII causes a sampling of the discrete Fourier transform at irregular intervals. Without a method of measuring the unpredictable error in the mirror movements, this effect manifests as errors in the wavenumber of the Fourier transform. I represent this effect by modifying the discrete Fourier transform (Eq. 2.1) to add in random variations in wavenumber:

$$\mathscr{F}_{k_x,k_y} = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} F_{n,m} \exp\left[-i2\pi(\frac{(k_x - \frac{N-1}{2} + \varepsilon_x)(n - \frac{N-1}{2})}{N} + \frac{(y - \frac{M-1}{2} + \varepsilon_y)(m - \frac{M-1}{2})}{M})\right] \tag{2.2}$$

The ε_x and ε_y represent random variations in the wavenumber caused by the random noise in mirror mount movements. A uniform distribution was chosen for these random numbers. Other distributions, such as the Gaussian distribution, were not tested but are expected to give similar results. Throughout the simulations, I define the strength of this wavenumber effect by a unitless real-valued parameter α, used to define the maximum value for the error:

$$\begin{gathered} -\alpha \leq \varepsilon_x \leq \alpha \\ -\alpha \leq \varepsilon_y \leq \alpha \end{gathered} \tag{2.3}$$

The parameter α can be thought of as the maximum error as a percentage of step size in wavenumber. For example, if $\alpha = 0.5$ then Eq. (2.2) generates the distorted Fourier transform at index (k_x, k_y) by sampling the true, undistorted Fourier transform at a random point within the indices $k_x \pm 1/2$ and $k_y \pm 1/2$.

To simulate the effects of mirror movement uncertainty, I first transform an image using Eq. (2.2). The next step is to transform the resulting distorted Fourier transform back to image space. However, transforming back to image space using a centered discrete Fourier transform similar to Eq. (2.1) is computationally inefficient; using an inverse Fast Fourier Transform (FFT) instead saves significant computation time. However, the Fourier transform data generated from Eq. (2.2) is zero-centered, causing it to be shifted a half step from the data that would be obtained from a FFT. Before applying an inverse FFT to our distorted Fourier transform data it first must be multiplied by a phase ramp. The phase ramp shifts the image data in Fourier space to correctly align it with the wavenumbers that would be generated by a FFT. For a square $N \times N$ image, the phase ramp is

$$\Phi(k_x, k_y) = \mathrm{e}^{-i\pi(k_x + k_y - N)} \tag{2.4}$$

where k_x and k_y again represent the indices in wavenumber space. With the data shifted to the appropriate wavenumbers, I then use an inverse FFT to obtain an image containing the simulated wavenumber errors. The Python code used to implement the complete wavenumber error model using Eqs. (2.2) and (2.4) is shown in Appendix A.

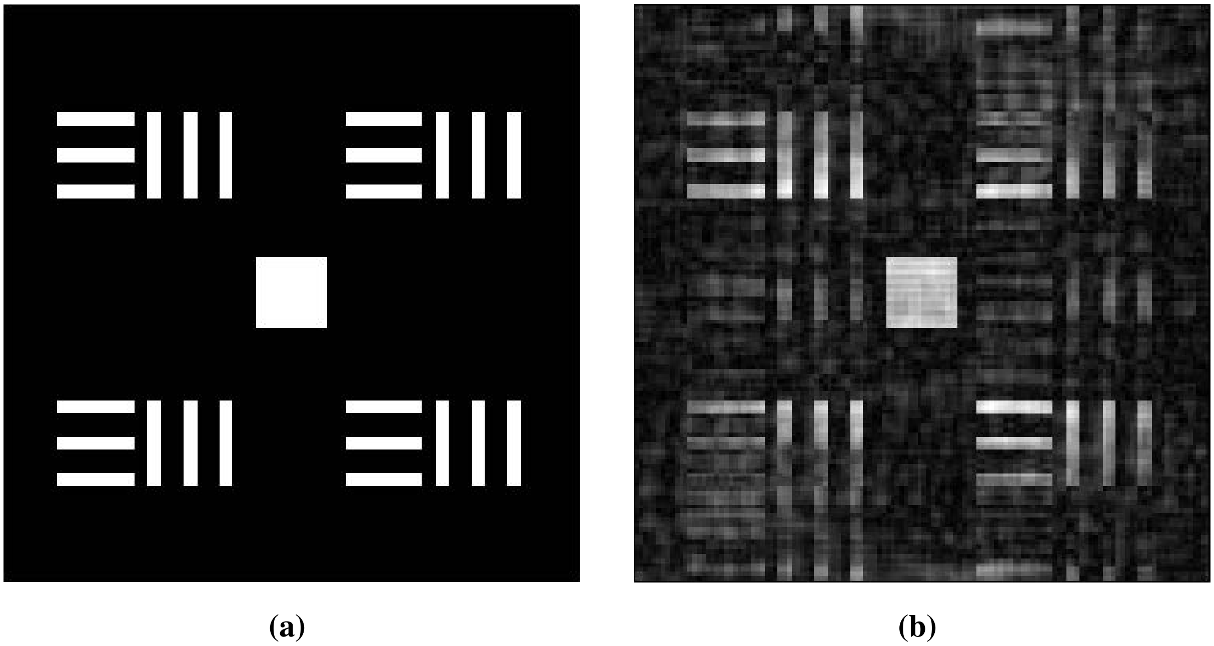


Figure 2.1 Example of a simulated binary image distorted by wavenumber error. Plot (a) shows the original image and (b) shows the image distorted due to wavenumber errors.

To test the simulation model, I introduced wavenumber error using the code in Appendix A into a binary image similar to the Air Force test chart shown in Fig. 1.2. The result is shown in Fig. 2.1. Clear ghosting effects are visible in the distorted image and are especially prominent near the edges. This radially-dependent ghosting closely matches the ghosting seen in the experimental image in Fig. 1.2. It appears that our model is successfully causing errors similar to those in experimental IPSII images.

2.2 Propagation of Error to the Fourier Transform

The wavenumber errors caused by mirror movement uncertainties affect both the phase and amplitude of the data in Fourier space. By comparing the distorted Fourier transform with the error-free version, these amplitude and phase errors become apparent. Figure 2.2 shows a one-dimensional image of three square pulses and the corresponding Fourier transform with and without simulated wavenumber errors. The effect in image space causes what appears to be random noise and some amount of ghosting. In Fourier space, the amplitude of the data is distorted somewhat but still has the same general shape as the undistorted transform due to the slowly varying nature of the amplitude function. On the other hand, the phase of the Fourier transform shows significant errors because the original function oscillates rapidly. In particular the phase with wavenumber error differs at some k values by two radians or more. Although wavenumber error propagates to both the amplitude and phase of the Fourier transform, this error more significantly impacts the phase.

As shown by previous research, the phase of the Fourier transform contains most of the image information [5]. The errors in the phase of IPSII data are responsible for most of the distortions in the final image. Because of this, repairing the phase errors in the data would have a significant impact on the resulting image even with persisting amplitude errors.

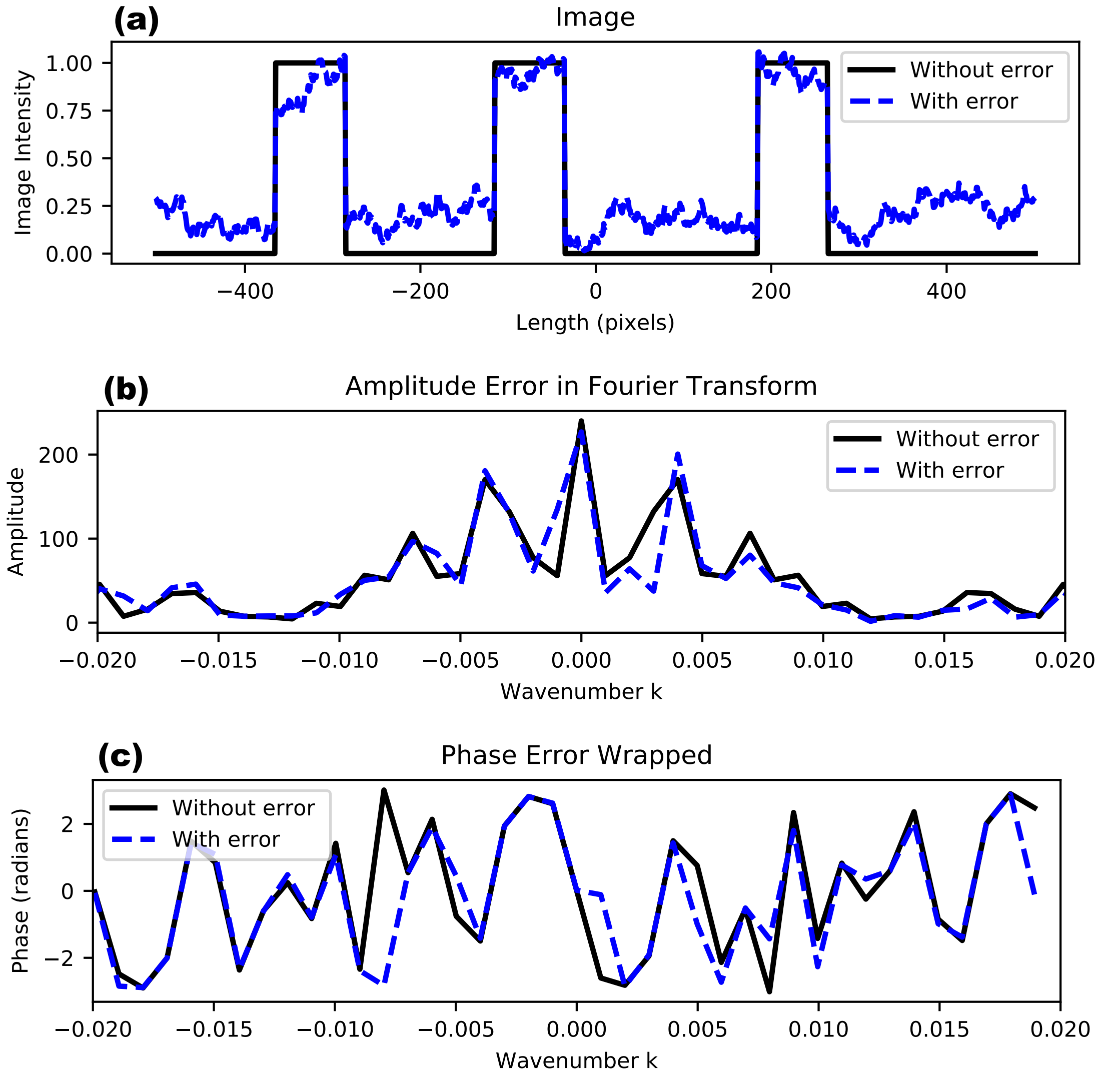


Figure 2.2 Example of a 1-D image distorted by simulated wavenumber errors with $\alpha = 0.5$. (a) shows the original and distorted image, (b) shows a portion of the amplitude of the Fourier transform of the image, and (c) shows the phase over the same interval.

2.3 Phase Retrieval Through Error Reduction

To correct the phase errors caused by wavenumber uncertainty in IPSII images, I apply commonly used phase retrieval algorithms. The most basic of these methods is known as Error Reduction and was developed by Fienup [6]. As seen in Fig. 2.4, Error Reduction consists of iterating between image and Fourier space while applying constraints to the image. In this figure, $F_0(x,y)$ represents the original measured IPSII image, $F_n(x,y)$ is the n^{th} iteration of the image, and $\mathscr{F}_{n+1}$ is the $(n+1)^{\text{th}}$ iteration of the Fourier transform that is used to create F_{n+1}. To begin the algorithm, I manually determine a support region to use as a constraint on $F_n(x,y)$. This region should not contain any portion of the imaged object and instead be empty space. I then force the amplitude of $F_n(x,y)$ to zero within this support region to generate F_{con}. Figure 2.3 shows this constraint applied to the

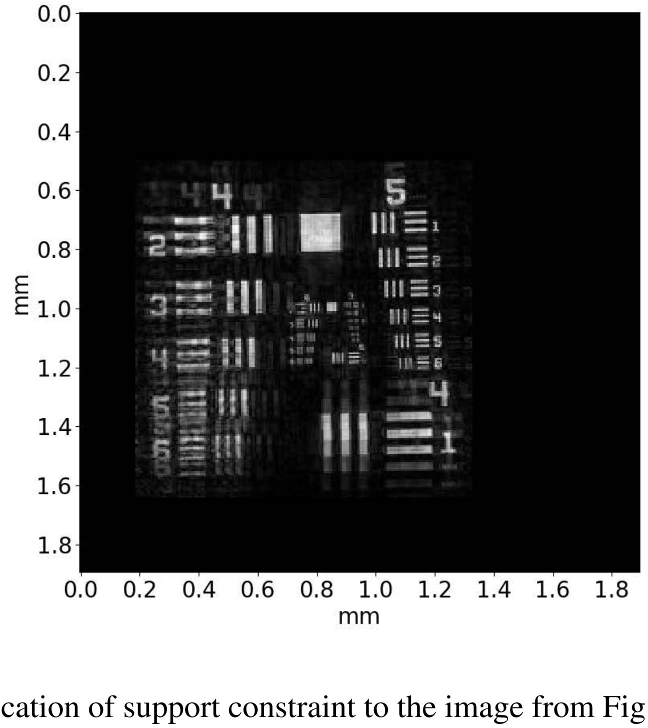


Figure 2.3 Application of support constraint to the image from Fig. 1.2 as part of Error Reduction (ER). Notice that with the constraint applied, the ghosts that were present on the left side of the image are forced to zero amplitude.

image from Fig. 1.2. Taking the FFT of this new constrained image creates $\mathscr{F}_n$. This new Fourier transform, $\mathscr{F}_n$, has phase values that better approximate the true error-free Fourier transform, but also has modified amplitude values as well. Next, I take only the phase data of $\mathscr{F}_n$ and combine it with the experimental amplitude values as measured by IPSII to generate $\mathscr{F}_{n+1}$. By taking the inverse FFT of $\mathscr{F}_{n+1}$, I obtain a new image, labled F_{n+1}, which has improved phase estimates. The original constraint can then be applied to this next generation image and the process repeated until it converges.

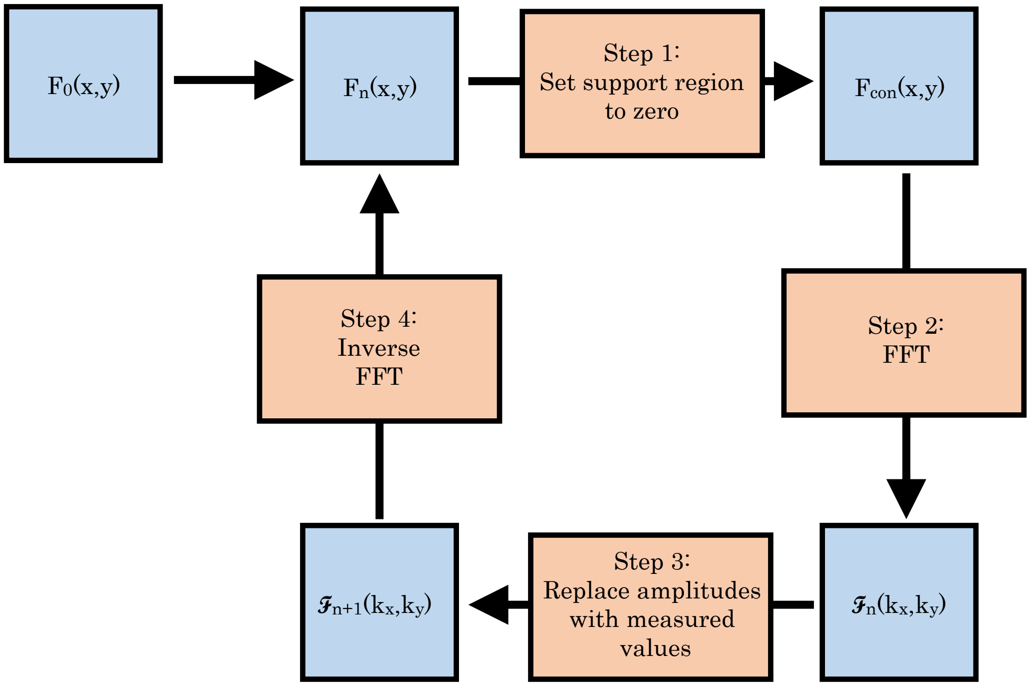

$F_0(x,y)$
$F_n(x,y)$
Step 1: Set support region to zero
$F_{con}(x,y)$
Step 4: Inverse FFT
Step 2: FFT
$\mathscr{F}_{n+1}(k_x,k_y)$
Step 3: Replace amplitudes with measured values
$\mathscr{F}_n(k_x,k_y)$

Figure 2.4 Iterative method used by the Error Reduction algorithm to correct the phase of IPSII image data. A constraint is first applied to the experimental data and then the Fourier transform is taken. The phase of this new transform combines with the experimentally measured amplitudes to create a new Fourier transform. Finally, the inverse transform is taken returning a new version of the image that can be iteratively plugged back into the algorithm.

2.4 Hybrid Input-Output and Shrink Wrapping

The Error Reduction (ER) algorithm quickly converges to a minimum; however, this minimum is likely not the desired absolute minimum [7]. To dislodge the phase retrieval algorithm from a local minimum, other methods aside from ER must be used. One of the most commonly used alternatives to Error Reduction is Hybrid Input Output (HIO) [6]. HIO works similarly to the ER algorithm by iterating between image space and Fourier space while applying a constraint. The difference in HIO is that instead of fixing the support region at zero amplitude in image space, HIO reduces the intensity in this region more gradually. The HIO constraint is

$$I_{n+1}(x,y) = \begin{cases} I_n - \beta I_{n-1} & (x,y)\ \varepsilon\ \gamma \\ I_n & (x,y)\ \not\varepsilon\ \gamma \end{cases}. \tag{2.5}$$

The symbol γ represents the support region that would be set to zero intensity in the ER algorithm and β is a constant representing the strength of the constraint. For these simulations in this chapter, β was set to a value of 0.8.

Hybrid Input Output applies the constraint less aggressively than Error Reduction and, thus, is able to better explore the solution space and avoid local minima. However, HIO sometimes converges to a local minimum that does not trap ER and vice versa. For this reason, I have chosen to implement both algorithms to correct IPSII images, switching between the two periodically to dislodge the algorithm from any local minima.

Beyond HIO and ER, I have also implemented a technique known as shrink wrapping to increase the convergence rate of the algorithm [8]. Shrink wrapping works by periodically increasing the constraint region to include additional regions of the image that have near zero intensity. The process begins by first convolving the intensity of the image with a two-dimensional guassian function. The constraint region is then expanded to include any portion of the convolved image that falls below a set amplitude threshold. The width of the Gaussian used in subsequent shrink wrappings is reduced gradually as the image converges to a solution. I found that a reduction following a $1/8$ power law gives good results. Additionally, the amplitude threshold of the shrink wrapping is gradually increased starting from a value of 10% up to value around 50% using the same $1/8$ power law.

The complete IPSII phase retrieval algorithm consists of a cycle of thirty iterations of HIO, then thirty iterations of ER followed by a shrink wrap. This complete cycle is repeated five times. Because the algorithm still sometimes fails to converge to the correct solution, random phase noise in periodically added to the data to dislodge it from local minima. The complete algorithm including HIO, ER, shrink wrapping, and the addition of random phase noise is in Appendix B. The input parameters used in Appendix B are set to values that give good results when applied to the image in Fig. 1.2. When applied to different images, the algorithm may require different inputs for optimal performance.

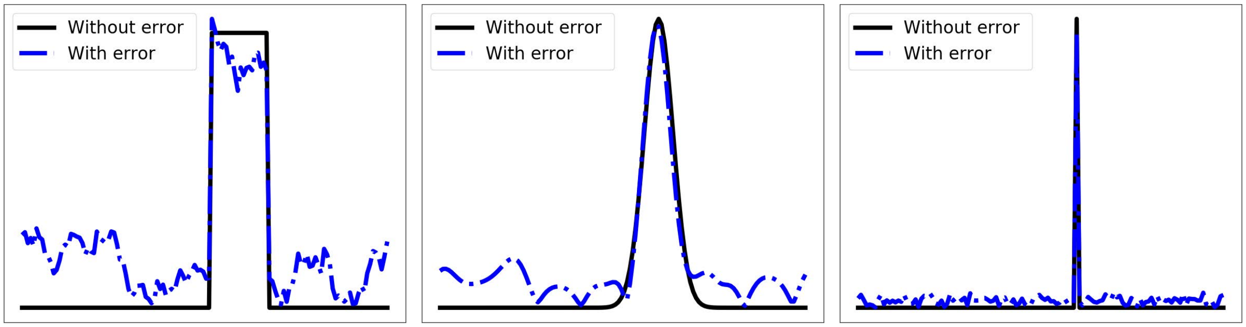
Chapter 3

Results

The primary goal of these simulations was to better understand noise present in experimental IPSII images. In this chapter, I first discuss the characterization of the wavenumber noise in my simulation model. I then analyze the ability of phase retrieval algorithms to repair the effects of this noise. Finally, I discuss future analysis work that is required to better understand IPSII image distortions.

3.1 Characterization of Wavenumber Noise

As described in Sec. 2.1, the modified Fourier transform presented in Eq. (2.2) accurately replicates the increase of noise in IPSII images towards the edge of the images. Additionally, through this model we are able to further characterize the experimental noise in IPSII. The effect of wavenumber error as modeled by Eq. (2.2) is compared in Fig. 3.1 for three different pulse shapes. These one-dimensional plots show that the noise caused by wavenumber error is not white noise or similar random noise. The distorted image of the square pulse shows noise that manifests the same square shape. The image of the gaussian pulse exhibits noise that shows gaussian fluctuations. Meanwhile, the image of the impulse function shows what looks like more traditional random noise. Wavenumber error appears to cause ghosting, or a copying and translation of portions of the imaged



(a) (b) (c)

Figure 3.1 Example of ghosting caused by wavenumber error in the images of one-dimensional pulses. Plot (a) shows a square pulse, (b) a Gaussian pulse, and (c) a delta function pulse. The α parameter was set to one for all three plots.

object to other locations in the image. This ghosting is likely also the source of the blurring seen in experimental images. Ghosts caused by wavenumber error are clearly visible near the lower right corner of Fig. 1.2.

3.2 Effectiveness of Phase Retrieval

Phase retrieval algorithms improve the quality of IPSII images but appear to be limited by amplitude errors that are unaffected by the algorithms. The effects of phase retrieval on an IPSII image is shown in Figure 3.2. Note that the ghosting seen in Figure 1.2 has disappeared almost entirely. The clearer separation between the bars in the lower left of the repaired image shows that the resolution has improved in that portion of the image. However, the phase retrieval process also caused some loss of information in the center of the target where finer details of the target disappeared. Evidently, the phase retrieval algorithms are better able to repair larger structures in IPSII images but struggle to reconstruct and maintain finer details. The disappearance of fine details after phase retrieval may be due to the shrink wrapping process being too aggressive.

To quantify the effectiveness of phase retrieval algorithms in this application, I have compared

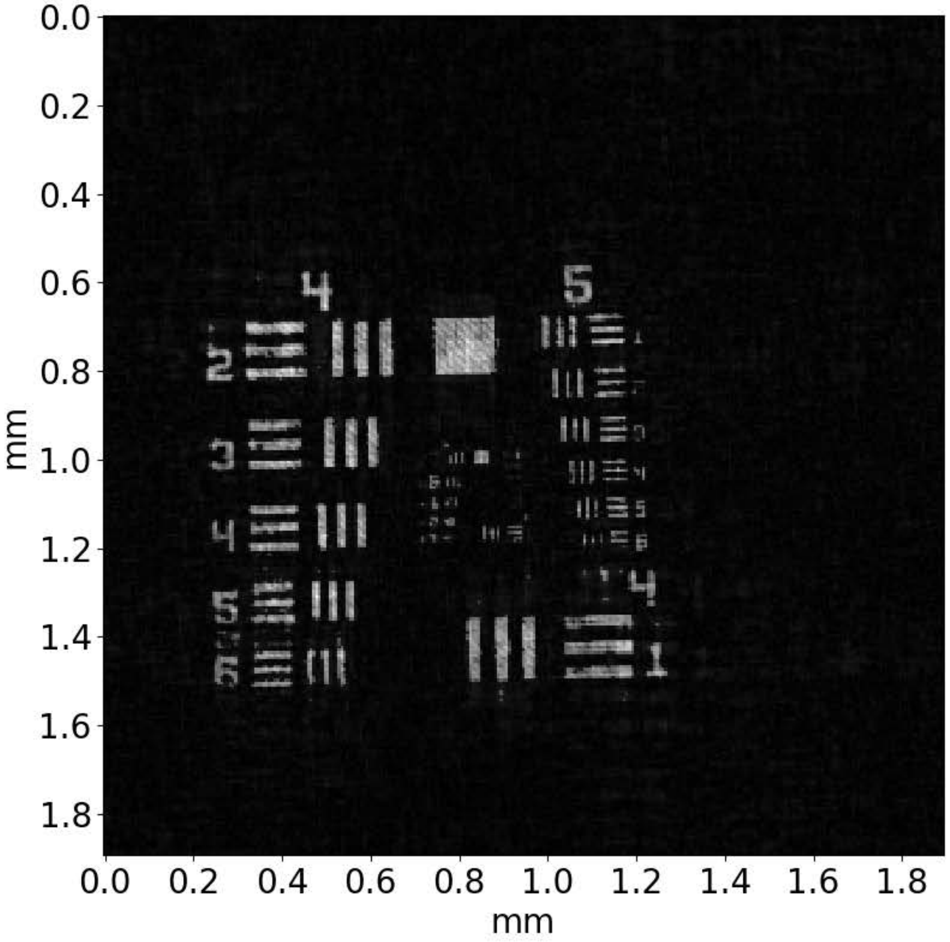


Figure 3.2 The experimental IPSII image shown in Fig. 1.2 after the Error Reduction, Hybrid Input Output, and Shrinkwrapping algorithms have been applied.

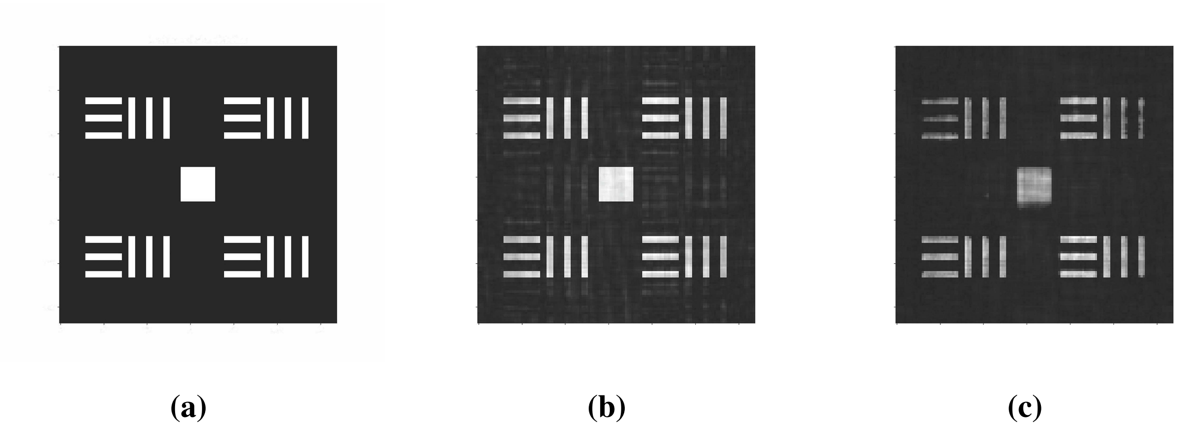


Figure 3.3 Simulated IPSII image (a) without any distortions, (b) with wavenumber error, and (c) repaired with phase retrieval.

the average error present in IPSII images before and after phase retrieval. The error is calculated using the normalized root mean squared error (NRMSE):

$$\mathrm{NRMSE} = \frac{\sqrt{(\mathrm{I}-\mathrm{I}_0)^2}}{\mathrm{I}_0}, \tag{3.1}$$

where I is the pixel intensity of the distorted image and I_0 is the pixel intensity before phase reconstruction. Averaging the NRMSE over the area of the image within the initial support region gives an average error value that can be used to quantify the quality of the image. This error calculation requires knowledge of the undistorted image, and thus, it is not easily applicable to experimental IPSII data where the undistorted image is unknown. This error calculation method is applied to the simulated IPSII images distorted using the code in Appendix A. The results of one trial are seen in Fig. 3.3, which shows a simulated IPSII image, the distorted version with wavenumber error, and the image after applying phase retrieval. For this trial, the image before phase retrieval has an average NRMSE of 1.06 and an average NRMSE of 0.91 after phase retrieval. This shows that phase retrieval does have a small but positive effect on the average quality of IPSII images.

3.3 Conclusion

Based on simulations carried out using Eq. (2.2), uncertainty in mirror movements in the experimental IPSII setup cause wavenumber errors that do affect the resulting images. In particular, the error caused by this mirror movement uncertainty produces ghosting effects that increase in magnitude with distance from the image center.

Analysis of the error in Fourier space shows that wavenumber error in IPSII causes errors in both the amplitudes and phases of the Fourier transform of the image. Phase retrieval algorithms are able to correct the phase errors but not the amplitude errors in Fourier space. The inability to correct the amplitude errors allows only a partial correction of the wavenumber errors present in IPSII images. As shown in Fig. 3.2, phase retrieval by itself is not capable of entirely repairing wavenumber error in IPSII images and leaves some artifacts of ghosting.

3.4 Future Work

Future experimental work is needed to fully confirm the findings of these simulations. The findings presented in this thesis could be confirmed by programming an additional random error into the motors controlling the mirrors in the IPSII experiment and measuring the resulting image distortions. If the ghosting and radially-dependent blurring effects increase in magnitude as the experimental mirror movement error is increased then it would confirm that wavenumber errors in IPSII are responsible for the ghosting present IPSII images. Additionally, higher quality motors with smaller uncertainties could be tested to evaluate if the amount of ghosting decreases.

More work is also needed to improve the capabilities of phase retrieval algorithms when applied to IPSII images. In particular, determining the optical input parameters for the phase retrieval algorithms is difficult and time consuming. Future work could be done to implement a genetic algorithm that would find the optimal input parameters for the phase retrieval process. Alternatively, a modified version of the Guided Hybrid Input-Output algorithm developed for use with CDI could be applied to IPSII [9].

We can also further investigate the effects of the amplitude errors caused by wavenumber uncertainty. I have spent some time attempting to use the ER algorithm to repair the amplitudes of IPSII images. I did this using the ER algorithm as shown in Fig. 2.4 but in step 3 replacing the phases of $\mathscr{F}_n$ with the measured phases instead of replacing the amplitudes to generate $\mathscr{F}_{n+1}$. These initial attempts at amplitude retrieval did not yield significant improvement in the image quality. Further work needs to be done to determine if ER or HIO could be used to correct for amplitude errors in IPSII images. Currently, we are not aware of any other existing methods comparable to phase retrieval algorithms that would be capable of repairing the amplitude errors. If, however, a method exists to repair the amplitude errors similarly to the phase errors, then the quality of IPSII images could be further improved. The elimination of amplitude errors along with phase errors in IPSII images may enable the removal of the ghosting artifacts entirely. More investigation into repairing both the phase and the amplitude of IPSII images in Fourier space could yield higher resolution images with lower amounts of ghosting and noise.

Appendix A

Modified Fourier Transform Code

This is the Python function used to simulate the effect of mirror movement uncertainty on IPSII images. It accepts two input parameters: `fx`, which the original un-distorted image, and `rand_max`, which is the maximum random error α. It returns one output, `newFk` which is the new modified Fourier Transform.

```
import numpy as np
from numpy import fft
import random

def FT_XYshift(fx,rand_max):
    Nrow = len(fx)
    Ncol = len(fx[0])
    xr = np.arange(0,Nrow, dtype=float) - Nrow//2
    xc = np.arange(0,Ncol, dtype=float) - Ncol//2
    XC,XR = np.meshgrid(xc,xr)
    integrand = np.empty(np.shape(fx), dtype=np.complex)
    Fk = np.empty(np.shape(fx), dtype=np.complex)

    for m in range(0,Nrow):
        variationM = random.uniform(-rand_max,rand_max) #This is a list of
            random numbers to add to the ky values
        for n in range(0,Ncol):
            variationN = random.uniform(-rand_max,rand_max) #This is a list
            of random numbers to add to the kx values
        integrand = fx * np.exp(-2j*np.pi*((m - Nrow//2 +
            variationM)*XR/Nrow + (n - Ncol//2 + variationN)*XC/Ncol))
        Fk[m,n] = sum(sum(integrand))
newFk = np.copy(Fk)
N = len(Fk)
for kx in range(0,N):
    for ky in range(0,N):
        newFk[kx,ky] = np.exp(-1j*np.pi*(kx + ky - N))*Fk[kx,ky]
newFk = fft.fftshift(newFk)
return newFk
```

Appendix B

Phase Reconstruction Code

This is the Python code used to repair the phase of experimental IPSII images using Error Reduction (ER), Hybrid Input Output (HIO), and shrink wrapping.

```
from scipy import signal
import numpy as np
from numpy import fft
import matplotlib.pyplot as plt

plt.close('all')

#================================================#
#Definition of initial constraint region:
frameTop = 80
frameBottom = 40
frameLeft = 30
frameRight = 90

#Number of sets of HIO, ER then Shrinkwrapping iterations
numCycles = 5
numER = 30
numHIO = 30

#Strength of HIO
HIOstrength = 0.8
```

```
#Random phase (in radians) added to the image after after shrinkwrapping
phaseAdded = 0.5

#Shrink wrapping paramters
gaussStart = 4
gaussEnd = 0.2
gaussLaw = 1/8
strengthStart = 0.1
strengthEnd = 0.5
strengthLaw = 1/8
#================================================#

#Import experimental data
folder = 'C:/Users/byuwh/Desktop/Research/IPSII Data/'
fileName = 'DistWavefront-data041-bits24-kspace.npy'
rawData = np.load(folder + fileName)
rawImage = fft.ifftshift(fft.ifft2(rawData))
N = np.size(rawImage,0)
rawImage[N//2+1,N//2+1] = 0
rawImage = np.rot90(rawImage)

#Initial FFT and Image values
FFT = fft.fft2(rawImage)
fftAmp = np.abs(FFT)
Image = fft.ifft2(FFT)

#Plot experimental image
plt.figure('Original')
plt.imshow(abs(Image),cmap='gray')
plt.title('Image before phase reconstruction')
plt.show()

#Creating the masks used in the phase reconstruction
mask = np.zeros((N,N))
mask[frameTop:-frameBottom,frameLeft:-frameRight] = 1
invMask = np.array(mask == 0,dtype=int)

#Plot constraint region
plt.figure('Initial Constraint')
plt.imshow(mask + abs(Image),cmap='gray')
plt.show()
plt.pause(0.1)
```

```
#Initialization of data needed to start phase reconstruction:
newImage = np.copy(Image)
newFFT = fft.fft2(Image)
lastImage = newImage + (np.random.rand(N,N) + 1j * np.random.rand(N,N)) * 0.1
gaussSize = np.linspace(gaussStart**gaussLaw,
    gaussEnd**gaussLaw,numCycles)**(1/gaussLaw)
shrinkStrength = np.linspace(strengthStart**strengthLaw,
    strengthEnd**strengthLaw,numCycles)**(1/strengthLaw)


for m in range(numCycles):

    #Hybrid Input Output (HIO):
    print('HIO...')
    for n in range(0,numHIO):
        newData = newImage - invMask * lastImage * HIOstrength
        newFFT = fftAmp * np.exp(1j * np.angle(fft.fft2(newData)))
        if n == 0 and np.mod(m,5) == 0:
            newFFT = newFFT * np.exp(phaseAdded*1j*np.random.rand(N,N))
        lastImage = np.copy(newImage)
        newImage = fft.ifft2(newFFT)

    #Error Reduction (ER):
    print('Error Reduction...')
    for n in range(0,numER):
        newFFT = fftAmp * np.exp(1j * (np.angle(fft.fft2(mask*newImage))))
        if n == 0 and np.mod(m,2) == 0:
            newFFT = newFFT * np.exp(phaseAdded*1j*np.random.rand(N,N))
        lastImage = np.copy(newImage)
        newImage = fft.ifft2(newFFT)


    #Shrinkwrapping:
    print('Shrinkwrapping...')
    gauss1D = signal.gaussian(N,gaussSize[m])
    gauss1D = np.tile(gauss1D,(N,1))
    gauss = gauss1D * np.transpose(gauss1D)
    convolved = signal.fftconvolve(gauss,abs(newImage),
        mode='same')
    mask = np.array(convolved > shrinkStrength[m]*np.amax(np.amax(convolved)),
        dtype=int) * mask
    invMask = np.array(mask == 0,dtype=int)

    if np.mod(m,1) == 0:
        plt.figure("After {} Shrink Wraps".format(m))
        plt.imshow(abs(newImage),cmap='gray')
        plt.title("{:.0f}".format(m))
        plt.show()
        plt.pause(0.1)

finalImage = abs(newImage)

#Plotting the final result:
plt.figure('Final')
plt.imshow(finalImage,cmap='gray')
plt.title('Image after phase reconstruction')
plt.show()
```

List of Figures

Bibliography

[1] D. Feldkhun and K. H. Wagner, “Doppler encoded excitation pattern tomographic optical microscopy,” Applied Optics 49 (2010).

[2] J. Miao, T. Ishikawa, I. K. Robinson, and M. M. Murnane, “Beyond crystallography: Diffractive imaging using coherent X-ray light sources,”, 2015.

[3] J. Jackson and D. Durfee, “Mechanically scanned interference pattern structured illumination imaging,” Optics Express **27,** 14969 (2019).

[4] J. S. Jackson, Ph.D. thesis, Brigham Young University, 2019.

[5] A. V. Oppenheim and J. S. Lim, “The Importance of Phase in Signals,” Proceedings of the IEEE **69,** 529–541 (1981).

[6] J. R. Fienup, “Reconstruction of an object from the modulus of its Fourier transform,” Technical Report No. 1 (1978) .

[7] S. Marchesini, “A unified evaluation of iterative projection algorithms for phase retrieval,” Review of Scientific Instruments 78 (2007).

[8] S. Marchesini, H. He, N. Chapman, P. Hau-Riege, A. Noy, R. Howells, U. Weierstall, and H. Spence, “X-ray image reconstruction from a diffraction pattern alone,” Physical Review B - Condensed Matter and Materials Physics 68 (2003).

[9] C. C. Chen, J. Miao, C. W. Wang, and T. K. Lee, “Application of optimization technique to noncrystalline x-ray diffraction microscopy: Guided hybrid input-output method,” Physical Review B - Condensed Matter and Materials Physics **76,** 064113 (2007).

Index