PISTON PHASE MEASUREMENTS TO ACCELERATE IMAGE RECONSTRUCTION IN MULTI-APERTURE SYSTEMS

Thesis
Submitted to
The School of Engineering of the UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for
The Degree
Master of Science in Electro-Optics

By
Jeffrey Read Kraczek

UNIVERSITY OF DAYTON
Dayton, Ohio
December 2011
PISTON PHASE MEASUREMENTS TO ACCELERATE IMAGE RECONSTRUCTION IN MULTI-APERTURE SYSTEMS

Name: Kraczek, Jeffrey Read

APPROVED BY:

________________________________________  __________________________________________
Paul McManamon, Ph. D.                        Joseph Marron, Ph.D.
Advisory Committee Chairman                   Committee Member
Technical Director, LOCI                      Lockheed Martin

________________________________________  __________________________________________
Joseph W. Haus, Ph. D.                        __________________________________________
Committee Member                             Associate Dean
Professor & Director,                        School of Engineering
Electro-Optics Program

________________________________________  __________________________________________
John G. Weber, Ph.D.                          Tony E. Saliba, Ph.D.
Associate Dean                               Dean, School of Engineering
School of Engineering                        & Wilke Distinguished Professor
ABSTRACT

PISTON PHASE MEASUREMENTS TO ACCELERATE IMAGE
RECONSTRUCTION IN MULTI-APERTURE SYSTEMS

Name: Kraczek, Jeffrey R.
University of Dayton

Advisor: Dr. Paul McManamon

Multi-aperture imaging has been used to receive high resolution images from arrays of sub-apertures. The use of sub-aperture arrays allows for more compact optical systems and enables conformal aperture imaging. Images collected from these arrays are processed to obtain the high resolution image. A high resolution image is created from an accurate representation of the pupil plane fields in each sub-aperture. Data processing to create an image is time consuming and computationally heavy. Compensating for the unknown piston phase error between the different sub-apertures is one of the more time consuming corrections required to process the image data into a single image.

Sub-aperture phasing simulations are used to explore the processing of multi-aperture arrays. The data is processed for several sub-aperture arrays, including 2 and 3 in-line sub-aperture arrays, and hex 7 and 19 sub-aperture arrays. A scheme is proposed for measuring the piston phases in each sub-aperture. It is shown through
numerical simulations that a system that measures the piston phase could significantly reduce the processing time required to phase the images from a multi aperture system into a single high resolution image.
ACKNOWLEDGMENTS

I would like to thank the people that helped me on this thesis. I would like to thank my advisor Dr. Paul McManamon for spending the time to guide, direct and take my problems to others for solutions. I would like to thank Dr. Joseph Haus for the time he spent helping me understand the answers to my random questions. I would like to thank Dr. Joseph Marron for being on my committee and answering questions within the bounds that he could. He guided a lot of what I learned for this project from far away. I would especially like thank Nick Miller. Nick helped me with many problems, both experimental and theoretical. He also adapted his code to run my simulations, with only minor tweaking needed. I would like to thank Dave Shemano for help with MATLAB related questions. I would like to thank Ed Watson for the questions he answered. I would like to thank the Electro-optics professors, without which I would never have gained the knowledge needed to complete this work.

This project was partially funded by Lockheed Martin.

This effort was supported in part by the U.S. Air Force through contract number FA8650-06-2-1081, and the University of Dayton Ladar and Optical Communications Institute (LOCI). The views expressed in this article are those of the author and do not reflect on the official policy of the Air Force, Department of Defense or the U.S. Government.
# TABLE OF CONTENTS

ABSTRACT .................................................................................................................. III

ACKNOWLEDGMENTS ................................................................................................... V

TABLE OF CONTENTS ................................................................................................. VI

LIST OF FIGURES .......................................................................................................... X

LIST OF TABLES ............................................................................................................ XIII

CHAPTER 1: INTRODUCTION ......................................................................................... 1

1.1 Problem Statement ................................................................................................. 1

1.2 Background ............................................................................................................. 2

1.3 Thesis Overview .................................................................................................... 4

CHAPTER 2: IMAGING THEORY ..................................................................................... 5

2.1 Signal Beam .......................................................................................................... 6

2.1.1 Object Diffraction ............................................................................................. 6

2.1.2 Lens Transformation ......................................................................................... 9

2.1.3 Imaging ............................................................................................................ 13

2.1.4 Imaging with Magnification ............................................................................ 15

2.2 Local Oscillator .................................................................................................. 16

2.2.1 Tilted Plane Wave Local Oscillator ............................................................... 17

2.2.2 Point Source Local Oscillator ....................................................................... 20

2.3 Mode Matching .................................................................................................. 21

2.3.1 Mode Matching with a Plane Wave Local Oscillator .................................. 21
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.2 Pupil Arrays</td>
<td>55</td>
</tr>
<tr>
<td>4.1.3 Hex Arrays</td>
<td>55</td>
</tr>
<tr>
<td>4.2 Processing Simulations: General</td>
<td>58</td>
</tr>
<tr>
<td>4.2.1 Adding Aberrations</td>
<td>58</td>
</tr>
<tr>
<td>4.2.2 Removing Aberrations</td>
<td>59</td>
</tr>
<tr>
<td>4.3 Processing Simulations: Specific</td>
<td>60</td>
</tr>
<tr>
<td>4.3.1 Perfect Image Simulation</td>
<td>60</td>
</tr>
<tr>
<td>4.3.2 General Processing Simulation</td>
<td>60</td>
</tr>
<tr>
<td>4.3.3 Piston Phase Measured Processing Simulation A</td>
<td>61</td>
</tr>
<tr>
<td>4.3.4 Piston Phase Measured Processing Simulation B</td>
<td>62</td>
</tr>
<tr>
<td>4.4 Simulation Results</td>
<td>62</td>
</tr>
<tr>
<td>4.4.1 Two Sub-aperture Pupil Array Results</td>
<td>62</td>
</tr>
<tr>
<td>4.4.2 Three Sub-aperture Pupil Array Results</td>
<td>67</td>
</tr>
<tr>
<td>4.4.3 Hex 7 Sub-aperture Pupil Results</td>
<td>71</td>
</tr>
<tr>
<td>4.4.4 Hex 19 Sub-aperture Pupil Results</td>
<td>74</td>
</tr>
<tr>
<td>4.4.5 Two Sub-aperture Pupil Array Results</td>
<td>62</td>
</tr>
<tr>
<td>5.1 Simulation Conclusions</td>
<td>76</td>
</tr>
<tr>
<td>5.2 Future Work</td>
<td>76</td>
</tr>
<tr>
<td>5.2.1 Measuring Piston</td>
<td>76</td>
</tr>
<tr>
<td>5.2.2 Processing Optimization</td>
<td>77</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>78</td>
</tr>
<tr>
<td>APPENDIX A: MATLAB CODE</td>
<td>80</td>
</tr>
<tr>
<td>A.1 Hex Array Building Code</td>
<td>80</td>
</tr>
<tr>
<td>A.2 Phase Error Addition Code</td>
<td>80</td>
</tr>
<tr>
<td>A.3 Direct Piston Removal Code</td>
<td>81</td>
</tr>
</tbody>
</table>
A.4 Aperture Array MTF Code

APPENDIX B: Pixel Size And Corresponding Frequency Conversion Between Planes

B.1 Pixel Size and Frequency Conversion From the Pupil Plane

B.2 Pixel Size and Frequency Conversion From the Pupil Plane

B.3 Application of Pixel Size and Frequency Conversion for MTF Plots
LIST OF FIGURES

Fig. 2-1: Propagation geometry showing a lens in the (x,y) plane. The object is in the
(ξ,η) plane. The imaging lens is in the (x,y) plane. The image is formed in the (u,v)
plane. The three planes are the same dimensions at a different distance along the
optic axis, though the (u,v) plane has been flip to remove negatives from the final
equations.......................................................... 5

Fig. 2-2: Schematic of digital holography system with variables and planes labeled. The
labeled planes are the same as those in Fig. 2-1. Z₁ is the propagation distance
from the object plane to the imaging lens. Z₂ is the total distance propagated from
the imaging lens to the image plane. The subscripts (a, b, and c) attached to Z₂ are
the components of the total Z₂. It is shown later that Z₂ is equal to fₙ. Z₃ is the
distance the local oscillator propagates with phase curvature.......................... 6

Fig. 2-3: a.) A diagram of a lens with axis and general parts labeled. b.) A diagram of
the first part of the lens. ................................................................................................10

Fig. 2-4: Left: Log scaled magnitude of the field in the pupil plane. Right: Cropped
portion of the pupil plane field..................................................................................24

Fig. 3-1: Top: 2D MTF of 2 in-line sub-apertures. Bottom Left: Horizontal slice of MTF
for of 2 in-line sub-apertures. Bottom Right: MTF comparison plot of horizontal and
vertical slices of the 2 in-line sub-apertures and a monolithic aperture with the same
radius of the max dimension of the 2 in-line sub-aperture array...........................40
Fig. 3-2: Top: 2D MTF of 3 in-line sub-apertures. Bottom Left: Horizontal slice of MTF for 3 in-line sub-apertures. Bottom Right: MTF comparison plot of horizontal and vertical slices of the 3 in-line sub-apertures and a monolithic aperture with the same radius of the max dimension of the 3 in-line sub-aperture array. .............................................41

Fig. 3-3: Top Left: 2D MTF of hex 7 sub-aperture array. Top Right: Horizontal slice of MTF for a hex 7 sub-aperture array. Bottom Left: Vertical slice of MTF for a hex 7 sub-aperture array. Bottom Right: MTF comparison plot of horizontal and vertical slices of the Horizontal slice of MTF for a hex 7 sub-aperture array and a monolithic aperture with the same radius as the max dimension of the hex 7 sub-aperture array..........................................................42

Fig. 3-4: Top Left: 2D MTF of hex 19 sub-aperture array. Top Right: Horizontal slice of MTF for a hex 19 sub-aperture array. Bottom Left: Vertical slice of MTF for a hex 19 sub-aperture array. Bottom Right: MTF comparison plot of horizontal and vertical slices of the Horizontal slice of MTF for a hex 19 sub-aperture array and a monolithic aperture with the same radius as the max dimension of the hex 19 sub-aperture array..........................................................43

Fig. 4-1: Flow chart of propagation simulation code. .......................................................53

Fig. 4-2: USAF-1951 resolution chart gray scale bitmap used in the propagation simulation. ........................................................................................................54

Fig. 4-3: Diagram of hex array aperture centers. ..........................................................56

Fig. 4-4: Flowchart of general processing simulation. .....................................................60

Fig. 4-5: Flowchart of piston phase measured processing simulation A. .................61

Fig. 4-6: Flowchart of piston phase measured simulation B. ........................................62

Fig. 4-7: Left: Perfect image from a single aperture. Right: Perfect image from a 2 aperture array.......................................................................................................................63
Fig. 4-8: Left: 60 speckle realization of aberrated data from one sub-aperture. Right: 60 speckle realization, intra aperture corrected image from one sub-aperture.  

Fig. 4-9: Left: Two sub-aperture composite image. Right: Static tip/tilt corrected image of two sub-apertures.  

Fig. 4-10: Left: Temporal tip/tilt corrected image for 2 sub-apertures. Right: Piston corrected for 2 sub-apertures.  

Fig. 4-11: Left: Temporal tip/tilt corrected image from the piston phase measured processing simulation A. Right: Piston corrected image from the piston phase measured processing simulation A.  

Fig. 4-12: Perfect image of 3 sub-aperture array.  

Fig. 4-13: Images from the processing simulation with 3 sub-aperture array. Top Left: Composite image. Top Right: Static tip/tilt corrected image. Bottom Left: Temporal tip/tilt corrected image. Bottom Right: Piston corrected image.  

Fig. 4-14: Left: Temporal tip/tilt corrected 3 sub-aperture array image from the piston phase measured simulation A. Right: Piston corrected image from same simulation.  

Fig. 4-15: Perfect image of hex 7 array.  

Fig. 4-16: Top Left: Composite image of hex 7 array from the general processing simulation. Top Right: static tip/tilt corrected image. Mid Left: Temporal tip/tilt corrected image. Mid Right: Piston corrected. Bottom Left: Temporal tip/tilt corrected from piston phase measured simulation A. Bottom Right: Piston corrected.  

Fig. 4-17: Left: Perfect image of hex 19 array. Right: Piston corrected image of hex 19 array from general processing simulation.
LIST OF TABLES

Table 4-1: Simulation time data for the three processing simulations with 2 sub-apertures. Times are in seconds.................................................................67
Table 4-2: Data for the 3 sub-aperture array.................................................................................70
Table 4-3: Time data for hex 7 array.............................................................................................74
Table 4-4: Time data for hex 19 array............................................................................................75
Table 5-1: Savings and approximate processing time by sub-aperture array.........................76
CHAPTER 1

INTRODUCTION

1.1 Problem Statement

The limiting factor for resolution in an imaging system is the size of the aperture. Camera technology has the ability to increase sampling, but there is no more possible resolution gain once the camera has sampled the image to the extent that the optical system can resolve based on the aperture size. It is possible to achieve high resolution using digital holography by using two or more sub-apertures and digitally combining the images through post processing algorithms. To digitally combine the field received in multiple sub-apertures into a high resolution image, limited in diffractive resolution by the size of the array, the fields must be known across each sub-aperture, including both amplitude and phase. Current detector technology detects only intensity. To combine the images from the different sub-apertures, the phase of the incident light must also be known. The amplitude and phase variations in the field for each sub-aperture can be retrieved using the spatial heterodyne detection, but these complex fields from each sub-aperture cannot be directly combined after the image is received.

One of the critically important corrections between sub-apertures is determining the piston phases, which is not directly available by spatial heterodyne techniques. On the other hand, temporal heterodyne detection finds the temporal phase by using a
relatively high bandwidth detector to detect the beat frequency between the return signal and a local oscillator beams. Detectors used for temporal heterodyne are single pixel devices. Camera formats for temporal heterodyne detection are under development; this makes them expensive and not readily available for research. Spatial heterodyne detection uses framing cameras to record the interference between two coherent beams arriving at different angles. Intensity fringes are visible on the detector array due to the two beam interference. The fringes are analogous to the beat frequency retrieved from a temporal heterodyne system’s detector, but in this case the change is a spatial variation rather than a temporal variation.

Unknown differences in lens positions and other optical system aberrations spoil the resolution of the image and make it necessary to determine phase corrections needed on each sub-aperture relative to the others. One important relative phase is piston phase; it can be retrieved through computationally intensive search algorithms. In this thesis it is demonstrated that a piston phase measurement for each sub-aperture relative to the other sub-apertures would give a significant savings in processing speed.

1.2 Background

In May of 1948 Gabor’s work demonstrating the first hologram was published\(^1\). The method described in the paper, which is sometimes termed as a Gabor hologram, is an in-line hologram. Images formed by in-line holograms have the desired image on top of its complex conjugate with the autocorrelations of the signal and reference beams centered on the same location. Though there were ways to get the image to show up clearly, his discovery was primarily used to improve image resolution in electron microscopy. At that time light sources did not have the requisite coherence length for optical holography.

It was not until after Theodore Maiman constructed the first working laser in 1960\(^2\), and the invention of the visible CW He-Ne by White and Rigden in 1962\(^3\), that
holography was practical. Leith and Upatnieks expanded the range of holography concepts \(^4^6\). For instance, they developed a method that used an off-axis reference beam instead of Gabor’s in-line approach. The off-axis reference beam separated the images so they were not all centered on the optic axis, as they were with Gabor’s holograms. There are several different names that refer to these holograms: Leith-Upatnieks hologram, offset-reference hologram, off-axis holography, and spatial heterodyne.

In 1967, Goodman and Lawrence created the first digital capture and reconstruction of a hologram \(^7\). In 1971 Huang wrote “Digital Holography.” \(^8\) In his paper, he brought together the ideas of manipulating optical data and creating holograms using computers. Huang’s paper is a summary of the work of the time, for which he gives many references. This early work, and the advances in computing since, has paved the way for modern digital holography.

With the birth of the He-Ne laser at 632 nm, by White and Rigden, another phenomenon was observed. Rigden and Gordon wrote the first paper on laser speckle \(^9\). The coherence of the laser was good for holography but it added noise when creating the image of a diffuse object. Surface roughness, on the object, causes path length differences in the illumination. These path length difference create interference patterns in the images which take the form of patches of light and dark areas. This speckle noise is enough to bring the signal to noise ratio down to 1. Lowenthal and Arsenault determined that the average transfer function for the coherent imaging is the transfer function for incoherent imaging \(^10\). This means that, with enough coherent images, coherent images can be averaged together to retrieve an image without speckle noise.

To average the coherent images, the camera and atmosphere cannot move at all, or the data needs to be processed further to correct for any movement in the images. A method for sharpening images was discussed by Muller, and Buffington, in 1974 \(^11\).
They discuss several sharpening metrics and gave general results for several sharpening metrics used in their computer simulations. Their work was for phase retrieval with incoherent light. Paxman and Marron wrote a paper in 1988 that shows that the image sharpness metrics can be used with coherent speckled data\textsuperscript{12}. These sharpening metrics allow for digital manipulation of the images to eliminate phase aberrations. Without these aberration corrections, speckle averaging would not improve image quality.

Recent work attempts to achieve higher resolution optical systems without large monolithic lenses and to allow conformal apertures. There have been several different ways of approaching higher resolutions without monolithic lenses. Many of the ways involve having arrays of smaller lenses, each known as a sub-aperture. One of the methods of gaining the higher resolutions was proposed by Marron and Kendrick\textsuperscript{13}. In their method, they combined many of the digital holography techniques mentioned earlier to construct higher resolution images using an array of several smaller apertures which are called sub-apertures.

1.3 Thesis Overview

Chapter 2 of this thesis covers imaging theory for a single aperture. We derive the separate pieces of using spatial and temporal heterodyne, and then put those pieces of theory together. We also include sections on speckle averaging and transfer functions. Chapter 3 generalizes the theory in chapter 2 to include optical systems of more than one aperture. We discuss some reasons for using multiple sub-apertures and then include some of the problems that are inherent to using multiple sub-apertures. Different phase errors are described and then the theory of eliminating them is discussed. Chapter 4 gives details about the numerical simulations, and presents the simulation results. Chapter 5 is devoted to a summary of the research and conclusions. Some suggestions are given for future work.
CHAPTER 2

IMAGING THEORY

In chapter two we derive the different pieces of theory required to perform digital holography in a single aperture. We start with the signal beam, local oscillator beam, and mode matching the two beams. We combine these pieces into a discussion of both off-axis digital holography (spatial heterodyne), and temporal heterodyne detection. We will finish the chapter with speckle averaging and transfer functions.

Fig. 2-1: Propagation geometry showing a lens in the \((x,y)\) plane. The object is in the \((\xi,\eta)\) plane. The imaging lens is in the \((x,y)\) plane. The image is formed in the \((u,v)\) plane. The three planes are the same dimensions at a different distance along the optic axis, though the \((u,v)\) plane has been flip to remove negatives from the final equations.
2.1 Signal Beam

2.1.1 Object Diffraction

Fig. 2-1 is a diagram of an object diffracting to and through a lens, and then to the focal plane of the lens. The three different coordinate axis \((\xi, \eta), (x, y),\) and \((u, v),\) are the object, pupil, and image planes. We use this figure as a map of how light propagates through a digital holography system. \(z_1\) and \(z_2\) are the propagation distances from the object to the pupil plane and from the pupil plane to the image plane. Fig. 2-1 is a different view of the top beam path in Fig. 2-2. Fig. 2-2 is a diagram of a sample spatial heterodyne detection system; the top beam path is the signal beam and the bottom beam path is the reference beam. \(z_1\) and \(z_2\) represent the same propagation distances as in Fig. 2-1 and \(z_3\) is the local oscillator propagation distance that contributes to the phase curvature.
Much of the work in section 2.1 is derived following Goodman\textsuperscript{14} chapters 4 and 5, but we strive clarify the results by filling in some left out steps. We assume that the illumination on our target is a uniform, unit amplitude, monochromatic plane wave. The assumption that the illumination is uniform is reasonable if the target is smaller than the beam. This assumption is important because it allows us to ignore how the illumination propagates to the target. We choose unit amplitude because the amplitude doesn’t matter if the beam is uniform. A long temporally coherent makes the monochromatic assumption realistic.

Point $P_1$ in Fig. 2-1 represents a point in the $(\xi, \eta)$ coordinate plane and $P_2$ is a point within the aperture, $P(x, y)$. Equation (2.1) is Huygens-Fresnel principle as stated in Goodman\textsuperscript{14} equation 4-8. The Huygens-Fresnel principle is the general starting point for the mathematical description of diffraction. It describes the field of an optical wave propagating through space. The integral sums the portion of the field from every point in the $(\xi, \eta)$ coordinate plane, to define the field at each point in the $(x, y)$ coordinate plane. The temporal phase portion of the equations is not included in Goodman\textsuperscript{14}, and will be left off for now because diffraction does not change the temporal components. We will add them in at the end of section 2.1.4.

$$U(P_2) = \frac{1}{j\lambda} \int \int \int \frac{U(P_1)}{r} e^{jkr} \cos \theta ds,$$

(2.1)

$U(P_1)$, or more generally $U(\xi, \eta)$ is the object field that is being propagated through space. We rewrite Eq. (2.1) with $\cos \theta = \frac{z_1}{r}$, where $\theta$ is the angle formed between points $P_1$ and $P_2$, in the $(x, y)$ plane, and explicitly introduce the two integration variables $\xi$ and $\eta$. The distance between $P_1$ and $P_2$, $r$, is described by the distance formula,
\[ r = \sqrt{z_1^2 + (x - \xi)^2 + (y - \eta)^2} \].

(2.2)

The distance between \( P_1 \) and \( P_2 \), \( r \), is described by the distance formula.

\[ U_i(x, y) = \frac{z_1}{j \lambda} \int_{-\infty}^{\infty} U(\xi, \eta) e^{j kr} \frac{d\xi d\eta}{r^2}. \]

(2.3)

Note that \( r \) and \( U(\xi, \eta) \) are functions of \( \xi \) and \( \eta \), but they are removed through integration. This is why we can change Eq. (2.1) from \( u(P_2) \) to \( U_i(x, y) \) in Eq. (2.3).

The subscript \( l \) denotes that \( U_i(x, y) \) is the field before the imaging lens. The Fresnel approximation is used on \( r \), which is the first two terms of the Taylor series expansion.

This assumes that \( z_1 \gg (x - \xi) \) or \( (y - \eta) \) or, equivalently, \( \left( \frac{x - \xi}{z_1} \right)^2 + \left( \frac{y - \eta}{z_1} \right)^2 \ll 1 \),

thus

\[ r = z_1 \sqrt{1 + \left( \frac{x - \xi}{z_1} \right)^2 + \left( \frac{y - \eta}{z_1} \right)^2} \approx z_1 \left[ 1 + \frac{1}{2} \left( \frac{x - \xi}{z_1} \right)^2 + \frac{1}{2} \left( \frac{y - \eta}{z_1} \right)^2 \right]. \]

(2.4)

We insert Eq. (2.4) into Eq. (2.3) to get Eq. (2.5)

\[ U_i(x, y) = \frac{z_1}{j \lambda} \int_{-\infty}^{\infty} U(\xi, \eta) e^{j kr} \frac{d\xi d\eta}{r^2} \left[ \frac{1}{z_1} \left( 1 + \frac{1}{2} \left( \frac{x - \xi}{z_1} \right)^2 + \frac{1}{2} \left( \frac{y - \eta}{z_1} \right)^2 \right) \right]. \]

(2.5)

If \( \frac{1}{2} \left( \frac{x - \xi}{z_1} \right)^2 + \frac{1}{2} \left( \frac{y - \eta}{z_1} \right)^2 \approx 1 \), the quadratic terms can be dropped in the denominator. This is an accurate approximation for the denominator of the fraction. The quadratic term needs to be kept in the exponential term because \( r \) is multiplied by the wave number which has \( \lambda \) in its denominator making \( k \) on the order of \( 10^6 \text{ m}^{-1} \) for
wavelengths in the near IR. Also, the complex exponential term describes the phase which wraps around at \(2\pi\). Small variations have large effects in such cases. With these assumptions, Eq. (2.5) becomes

\[
U_i(x, y) = \frac{e^{j k z_1}}{j \lambda z_1} \int \int \left[ U(\xi, \eta) e^{j k \xi (x - \xi)^2 + (y - \eta)^2} \right] e^{j 2\pi \left[ (x - \xi)^2 + (y - \eta)^2 \right]} d\xi d\eta.
\]  

(2.6)

A two-dimensional delta function is introduced to represent a point source for \(U(\xi, \eta)\) to determine the impulse response, also called the point spread function (PSF).

\[
U(\xi, \eta) = \delta(\xi + \xi_0, \eta + \eta_0).
\]  

(2.7)

Inserting Eq. (2.7) into Eq. (2.6) gives

\[
U_i(x, y) = \frac{e^{j k z_1}}{j \lambda z_1} \int \int \delta(\xi + \xi_0, \eta + \eta_0) e^{j k \xi (x - \xi)^2 + (y - \eta)^2} d\xi d\eta.
\]  

(2.8)

Equation (2.8) is evaluated to get

\[
U_i(x, y) = \frac{e^{j k z_1}}{j \lambda z_1} e^{j k \xi (x - \xi_0)^2 + (y - \eta_0)^2}.
\]  

(2.9)

As \(\xi_0\) and \(\eta_0\) could be any point in the \((\xi, \eta)\) plane, we redefine Eq. (2.9) as

\[
U_i(x, y) = \frac{e^{j k z_1}}{j \lambda z_1} e^{j k \xi (x - \xi_0)^2 + (y - \eta_0)^2},
\]  

(2.10)

but remember that it is a single point.

### 2.1.2 Lens Transformation

The lens used for imaging is in the \((x, y)\) plane and the field \(U_i(x, y)\) is incident on it. We assume a perfect, thin, lens that only changes the phase of the field. The phase accumulation over distance is given by Eq. (2.11).

\[
\phi = \frac{2\pi}{\lambda} nL
\]  

(2.11)
where \( n \) is the index of refraction and \( L \) is the physical path length through the material.

![Diagram](image)

**Fig. 2-3:** a.) A diagram of a lens with axis and general parts labeled. b.) A diagram of the first part of the lens.

A diagram of a lens is shown in Fig. 2-3a, with the first face shown in Fig. 2-3b.

In Fig. 2-3a, \( t \) is the total thickness of the lens, and \( t_1, t_2, \) and \( t_3 \) are the thicknesses of the front, middle, and back parts of the lens. In Fig. 2-3b, \( P_{2L}(x, y) \) is the length between the optic axis and the point \( P_2(x, y) \), but this time \( P_2 \) is actually on the front face of the lens. \( R \) is the radius of the lens and \( t_a(x, y) \) is the distance that the light travels through \( t_1 \) before touching the glass.

The front part of the lens gives a phase change of \( \phi_1(x, y) \). Some of \( \phi_1(x, y) \) describes the illumination traveling through air to the lens surface.

\[
\phi_1(x, y) = k \left( n_i (t_1 - t_a(x, y)) + n_a t_a(x, y) \right) = k \left( n_i t_1 + (n_a - n_i) t_a(x, y) \right),
\]

where \( n_i \) and \( n_a \) are the indices of refraction for glass and air respectively. To make sense of the equation, \( t_a(x, y) \) needs to be explicitly in \( (x, y) \)

\[
t_a(x, y) = R - \sqrt{R^2 - x^2 - y^2}.
\]
The square root comes from the Pythagorean theorem, solving for the bottom side of the triangle, in Fig. 2-3b, formed by the two radii and \( P_{2L}(x, y) \). We rewrite \( t_u(x, y) \) to be in a form that can be used with the paraxial approximation. The paraxial approximation, like the Fresnel approximation, uses the Taylor series expansion. This assumes the radius of the lens is much greater than \( P_{2L}(x, y) \)

\[
t_u(x, y) = R \left[ 1 - \sqrt{1 - \left( \frac{x^2 + y^2}{R_i^2} \right)} \right] = R \left[ 1 - \left( 1 - \frac{x^2 + y^2}{2R_i^2} \right) \right] = \frac{x^2 + y^2}{2R_i} \tag{2.14}
\]

We go back to \( \phi_1(x, y) \) now that \( t_u(x, y) \) is in an understandable form. The final form of Eq. (2.14) is inserted into Eq. (2.12)

\[
\phi_1(x, y) = k \left( n_1 t_1 + (n_u - n_l) \frac{x^2 + y^2}{2R_i} \right) \tag{2.15}
\]

Referring to Fig. 2-3a, there are three parts of the lens. The middle part of the lens follows Eqs. (2.11) because it has a uniform thickness with a single index of refraction

\[
\phi_2 = kn_1 t_2. \tag{2.16}
\]

The back side of the lens is similar to front side

\[
\phi_3(x, y) = k \left( n_3 t_3 + (n_u - n_l) \frac{x^2 + y^2}{2R_2} \right). \tag{2.17}
\]

All three contributions are added together to form the phase change across the lens,

\[
\phi = \phi_1 + \phi_2 + \phi_3 = k \left( n_1 t_1 - (n_u - n_l) \frac{x^2 + y^2}{2R_i} + n_1 t_2 + n_3 t_3 - (n_u - n_l) \frac{x^2 + y^2}{2R_2} \right) \tag{2.18}
\]
At this point we assumed that a thin lens is used in this system. We neglect \( \phi_{12} \) and rewrite \( \phi \) as

\[
\phi = k \left( n_1 t_1 + (n_1 - n_a) \left( \frac{x^2 + y^2}{2R_1} \right) + n_2 t_3 + (n_2 - n_a) \left( \frac{x^2 + y^2}{2R_2} \right) \right),
\]

(2.19)

\[
\phi = k n_1 (t_1 + t_3) + k \frac{1}{2} \left( \frac{x^2 + y^2}{n_1} \right) \left( n_1 - n_a \right) \left( \frac{1}{R_1} + \frac{1}{R_2} \right).
\]

(2.20)

The last term in Eq. (2.20) contains a form of the lensmaker’s formula

\[
\frac{1}{f} = (n_1 - 1) \left( \frac{1}{R_1} + \frac{1}{R_2} \right).
\]

(2.21)

The lensmaker’s formula allows us to generalize Eq. (2.20) to allow for more than a single lens, as long as the compound lens system can still be classified as a thin lens.

We assume that \( n_a = 1 \), the last part of Eq. (2.20) can be changed to what is shown in Eq. (2.21). \( (t_1 + t_3) \) is also redefined as \( \Delta_0 \)

\[
\phi = k n_1 \Delta_0 + k \frac{1}{2f} \left( \frac{x^2 + y^2}{n_1} \right).
\]

(2.22)

The change in the field due to a lens is given by

\[
U_{\text{lens}} = e^{j k \Delta_0} P(x, y) e^{-\frac{j k}{2f} (x^2 + y^2)}.
\]

(2.23)

where \( P(x, y) \) describes the lens shape. Because we assume that we used a thin lens, Eq. (2.10) is multiplied by Eq. (2.23) to get

\[
U'_i(x, y) = e^{j k \Delta_0} U_i(x, y) P(x, y) e^{-\frac{j k}{2f} (x^2 + y^2)}.
\]

(2.24)

The thin lens assumption allows us to treat the lens as merely a phase screen. Eq. (2.24) describes the field, after a thin lens. The same result is found in Goodman\(^{14}\)
equation 5-12, except that the constant phase term is neglected. Eq. (2.24) assumes monochromatic light and the paraxial approximation.

2.1.3 Imaging

We use Fresnel diffraction, similar to what was done in Eq. (2.6), to diffract the field defined in Eq. (2.24) to the image plane.

\[ U(u, v) = \frac{e^{jkz_2}}{j\lambda z_2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U_i(x, y) e^{\frac{jk}{z_2}[(x-u)^2+(y-v)^2]} \, dx \, dy \]  

(2.25)

expanding Eq. (2.25) by inserting Eq. (2.24) the result is

\[ U(u, v) = \frac{e^{jkn_0} e^{jkz_2}}{j\lambda z_2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U_i(x, y) P(x, y) e^{\frac{-jk}{z_2}[(x-u)^2+(y-v)^2]} e^{\frac{2\pi}{\lambda}[(x-u)f + (y-v)f]} \, dx \, dy \]  

(2.26)

Recall that in Section 2.2.1 we assumed a point object. The derivation following that Section is general. Once \( U_i(x, y) \) is inserted, \( U(u, v) \) becomes \( h(u, v; \xi, \eta) \) the Impulse response. It defines how a single point is processed through the system

\[ h(u, v; \xi, \eta) = \frac{e^{jkn_0} e^{jkz_2}}{j\lambda z_2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{e^{jkz_1}}{e^{\frac{2\pi}{\lambda}[(x-u)f + (y-v)f]} \, dx \, dy \]  

(2.27)

Expand the exponents in Eq. (2.27) to simplify the expression

\[ h(u, v; \xi, \eta) = -e^{jk(z_1+z_2)} e^{jkn_0} e^{\frac{jk}{z_2}[z_2+\eta^2]} e^{\frac{jk}{z_2}[(x^2+u^2)]} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P(x, y) e^{\frac{jk}{z_2}[(\xi-u)/(z_1+z_2)]^2 e^{\frac{jk}{z_2}[(\eta-v)/(z_1+z_2)]^2} dx \, dy \]  

(2.28)

Two of the quadratic phase terms can be neglected. When \( \frac{1}{z_1} + \frac{1}{z_2} - \frac{1}{f} = 0 \), which is a form of the geometric lens law, then \( e^{\frac{jk}{z_2}[(\xi-u)/(z_1+z_2)]^2 e^{\frac{jk}{z_2}[(\eta-v)/(z_1+z_2)]^2} = 1 \). Assuming \( z_1 \) is much
larger than the physical size of the object, then \( e^{\frac{jk(z_2^2 + v^2)}{2}} = 1 \). The quadratic phase term
\[ e^{\frac{jk(z_2^2 + v^2)}{2}} \]
cannot be neglected as done in Goodman\(^{14}\) because the phase is of interest and the image is not measured on a spherical surface. Using these assumptions we can simply Eq. (2.28)
\[ h(u, v; \xi, \eta) = -e^{jk(z_2 + v_2)} e^{i\lambda h_{MN}} \frac{e^{\frac{jk(z_2^2 + v^2)}{2}}}{\lambda^2 z_2^2 z_1} \int \int P(x, y) e^{-j\left[\left(\frac{\xi}{z_2} + u\right) x + \left(\frac{\eta}{z_2} + v\right) y\right]} dxdy \quad (2.29) \]
Equation (2.29) is rewritten with \( \frac{z_2}{z_1} = M \), M being the magnification of the object due to the system
\[ h(u, v; \xi, \eta) = e^{jk(z_2 + v_2)} e^{i\lambda h_{MN}} \frac{Me^{\frac{jk(z_2^2 + v^2)}{2}}}{(\lambda z_2^2)} \int \int P(x, y) e^{\frac{-j\left[(u-M\xi) + (v-M\eta) + (v_2)\right]}{z_2^2}} dxdy \quad (2.30) \]
The integrand of Eq. (2.30) is a scaled Fourier transform integral
\[ h(u - M\xi, v - M\eta) = e^{jk(z_2 + v_2)} e^{i\lambda h_{MN}} \frac{Me^{\frac{jk(z_2^2 + v^2)}{2}}}{(\lambda z_2^2)} \mathcal{F}\left[P(x, y)\right]_{(u-M\xi), (v-(M\eta)}, \quad (2.31) \]
Goodman\(^{14}\) states that the image field can always be expressed as a superposition integral which is expressed in Goodman\(^{14}\) equation 5-23, which is written as
\[ U_i(u, v) = \int \int U_o(\xi, \eta) h(u, v; \xi, \eta) d\xi d\eta \quad (2.32) \]
\( U_o(\xi, \eta) \) is the object field; h is the PSF, which defines how individual point sources are affected by the system. Imaging extended sources is accomplished by imaging a collection of point sources, the convolution of the object field with the PSF is imaging
each point separately through the system until the entire image is formed. In that case Eq. (2.32) turns into a convolution to get

\[
U_i(u, v) = \int_{-\infty}^{\infty} \int U_o(\xi, \eta)h(u-M\xi, v-M\eta) \, d\xi \, d\eta
\tag{2.33}
\]

If we assume a magnification to be 1, then Eq. (2.33) is explicitly a convolution

\[
U_i(u, v) = \int_{-\infty}^{\infty} \int U_o(\xi, \eta)h(u-\xi, v-\eta) \, d\xi \, d\eta = U_o(u, v) \otimes h(u, v)
\tag{2.34}
\]

h is expanded in Eq. (2.34) to get

\[
U_i(u, v) = e^{ikz_1}e^{ik\beta_0} \frac{M^2e^{i\beta_z(u^2+v^2)}}{\lambda z_1^2} \int_{-\infty}^{\infty} \int U_o(\xi, \eta) \mathcal{F}[P(x, y)]\bigg|_{x=(u-\xi), \ y=(v-\eta)} \, d\xi \, d\eta
\tag{2.35}
\]

Notice that the quadratic term in the \((u, v)\) is not actually part of the convolution due to different variables then those being integrated over. This point needs to be remembered because \(U_o(u, v) \otimes h(u, v)\) would imply that the quadratic term would be part of the convolution. In Section 2.6, we will return to Eq. (2.34), but, to be more general, we will derive imaging with magnification included.

**2.1.4 Imaging with Magnification**

We rearrange terms to define new terms by letting \(\xi' = M\xi\), \(\eta' = M\eta\), \(d\xi = \frac{1}{M} \, d\xi'\) and \(d\eta = \frac{1}{M} \, d\eta'\)

\[
U_i(u, v) = \frac{1}{M^2} \int_{-\infty}^{\infty} \int U_o\left(\frac{\xi'}{M}, \frac{\eta'}{M}\right) h(u-\xi', v-\eta') \, d\xi' \, d\eta'
\tag{2.36}
\]

where we define \(\frac{\tilde{\cdot}}{|M|}\) as
\[
\tilde{h}(u, v) = e^{j(\xi + \zeta) + jk_x \Delta \psi} \left( \frac{Me^{j(\xi + \zeta)}}{|M|} \right) \mathcal{F} \left[ P(x, y) \right] \left( \frac{M_x}{\lambda_{z_1}^2} \right) \left( \frac{M_y}{\lambda_{z_1}^2} \right).
\]

(2.37)

This changes \( U_i(u, v) \) to become the convolution integral

\[
U_i(u, v) = \frac{1}{|M|} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u_o \left( \frac{\xi'}{M}, \frac{\eta'}{M} \right) \tilde{h}(u, v) \mathcal{F} \left[ P(x, y) \right] \left( \frac{M_x}{\lambda_{z_1}^2} \right) \left( \frac{M_y}{\lambda_{z_1}^2} \right) \, d\xi' \, d\eta'.
\]

(2.38)

We defined \( \tilde{h} \) because Goodman\textsuperscript{14} gives the geometrical-optics image as his equation 5-39

\[
U_g(u, v) = \frac{1}{|M|} u_o \left( \frac{u}{M}, \frac{v}{M} \right).
\]

(2.39)

This is a statement that the image is a perfect, scaled, version of the object, otherwise known as the geometrical optics image. We insert Eq. (2.39) into Eq. (2.38) to find

\[
U_i(u, v) = U_g(u, v) \otimes \tilde{h}(u, v).
\]

(2.40)

We redefine \( U_i(u, v) \) as \( U_s(u, v) \), expanding terms, and assuming \( z_2 \) to be the focal length of the imaging lens, \( f_s \). This assumption is reasonable for \( z_1 \) being substantially larger than \( f_s \)

\[
U_s(u, v) = \frac{1}{(\lambda_{z_2}^2)^2} e^{j\xi} e^{j(k_{z_1} + k_{j_f})} e^{j\Delta \phi} U_g(u, v) \otimes \mathcal{F} \left[ P(x, y) \right].
\]

(2.41)

2.2 Local Oscillator

We have two local oscillators (LO), though the forms are identical. There is one LO for spatial heterodyne detection, which has the same wavelength as the signal beam, and one LO for temporal heterodyne detection, which has a different wavelength as the signal beam. We desire two results from our local oscillators. First, the angle on the
spatial heterodyne LO separates the image, twin image, and autocorrelation terms formed by holography. This will be discussed in Section 2.4. Second, the signal and LOs beams should have the same phase curvature on the detectors to maximize received signal strength and to eliminate quadratic phase curvature. This is called mode matching and will be discussed in Section 2.3.

There are two formats adopted for LO beams. The first way is to use a collimated beam, generally referred to as a plane wave. The second way is to use a point source. In the next two sections both cases are examined.

2.2.1 Tilted Plane Wave Local Oscillator

Heterodyning is performed by mixing a LO with a signal beam. A particular kind of heterodyne is off-axis digital holography, sometimes called spatial heterodyne. This is performed using a LO that is of the same wavelength, but the beam is tilted off angle from the optical axis. The interference between the signal beam and LO beam cause spatial fringes to appear across the image. These fringes can be captured and processed to retrieve the spatial complex field instead of only the intensity. For spatial heterodyne to work the best, the quadratic phase term \((u^2 + v^2)\) must be matched on the detector.

Looking back at Fig. 2-2, we are now following the lower beam path. The same field definitions are applied, but now referring to the LO beam. We start by referring to Eq. (2.24), the field directly after the lens. The focal length and pupil are assumed to be different from the imaging arm

\[
U_1'(x, y) = e^{ikn0}U_i'(x, y)P(x, y) e^{-\frac{k}{2} f(x^2 + y^2)}.
\]  

(2.24)

\(U_i'(x, y)\) is diffracted a length \(z_3\).
\[
U(u,v) = \frac{e^{jkz_1}}{j\lambda z_3} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U(x,y) e^{\frac{jk}{2z_3}(u-x)^2+(v-y)^2} \, dx \, dy.
\]

(2.42)

We insert Eq. (2.24) into Eq. (2.42) to find

\[
U(u,v) = e^{jkn_0} e^{\frac{k}{2z_3}(u^2+v^2)} \frac{e^{jkz_1}}{j\lambda z_3} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} U_i(x,y) P(x,y) \frac{e^{\frac{jk}{2z_3}(\frac{1}{\lambda} + 1)x^2+y^2}}{e^{\frac{jk}{2z_3}(\frac{1}{\lambda} + 1)y^2}} \, dx \, dy \quad (2.43)
\]

Equation (2.43) is rearranged to get a scaled Fourier transform and the convolution theorem is applied.

\[
U(u,v) = e^{jkn_0} e^{\frac{k}{2z_3}(u^2+v^2)} \frac{e^{jkz_1}}{j\lambda z_3} \left[ \mathcal{F} \left( e^{\frac{jk}{2z_3}(\frac{1}{\lambda} + 1)x^2+y^2} \right) \right] \otimes \mathcal{F} \left\{ U_i(x,y) \right\} \otimes \mathcal{F} \left\{ P(x,y) \right\}.
\]

(2.44)

When taking the Fourier transforms, \( P(x,y) \) is equal to 1 because the aperture is assumed to be infinite in size

\[
\mathcal{F} \left\{ P(x,y) \right\} = \mathcal{F} \{ 1 \} = \delta(f_u, f_v) = (\lambda z_3)^2 \delta(u,v); \quad f_u = \frac{u}{\lambda z_3}.
\]

(2.45)

The \((\lambda z_3)^2\) comes from the denominator of both delta functions to the numerator on the outside of the delta function by the property of delta function, \( \delta(at) = \frac{1}{|a|} \delta(t) \), where "a" is a constant and "t" the variable. We assume the LO is a tilted plane wave.

\[
U_i(x,y) = e^{-jk(\sin(\theta)x + \sin(\phi)y)}.
\]

(2.46)

Equation (2.44) requires the Fourier transform of \( U_i(x,y) \).

\[
\mathcal{F} \left\{ U_i(x,y) \right\} = \mathcal{F} \left\{ e^{-jk(\sin(\theta)x + \sin(\phi)y)} \right\} = \delta \left\{ \frac{u}{\lambda z_3} + \frac{\sin(\theta)}{\lambda}, \frac{v}{\lambda z_3} + \frac{\sin(\phi)}{\lambda} \right\}.
\]

(2.47)
We pull the \((\lambda z_3)^2\) out using the same scaling property of the delta function.

\[
\mathcal{F}\left\{U_1(x, y)\right\} = (\lambda z_3)^2 \delta(u + z_3 \sin(\theta), v + z_3 \sin(\phi))
\] (2.48)

The Fourier transform of the quadratic phase term is found in Goodman\textsuperscript{14}, Table 2.1, namely,

\[
\mathcal{F}\left\{e^{j\pi(a^2x^2 + b^2y^2)}\right\} = \frac{j}{ab} e^{-j\pi \left(\frac{a^2}{x^2} + \frac{b^2}{y^2}\right)};
\] (2.49)

\[
f_x = \frac{u}{\lambda z_3}, a^2 = b^2 = \frac{1}{\lambda} \left(\frac{1}{z_3} + \frac{1}{f}\right) = \frac{(f - z_3)}{\lambda z_3 f},
\]

\[
\mathcal{F}\left\{e^{j\frac{k}{2z_3} \left(\frac{1}{f - z_3}\right)(x^2 + y^2)}\right\} = \frac{j\pi z_3 f}{|f - z_3|} e^{-j\pi \left(\frac{\lambda z_3 f}{2(z_3 - f)}\right)(u^2 + v^2)}
\] (2.50)

\[
= \frac{j \pi z_3 f}{|f - z_3|} e^{-j \left(\frac{k f}{2(z_3 - f)}\right)(x^2 + y^2)}.
\]

Equation (2.44) with the Fourier transforms explicitly written as

\[
U(u, v) = e^{jk\hat{n}_b} e^{j\frac{k}{2z_3}(u^2 + v^2)} e^{j\frac{\pi z_3 f}{\lambda z_3 |f - z_3|}} e^{-j\pi \left(\frac{\lambda z_3 f}{2(z_3 - f)}\right)(u^2 + v^2)}
\] (2.51)

\[
\otimes (\lambda z_3)^2 \delta(u + z_3 \sin(\theta), v + z_3 \sin(\phi)) \otimes (\lambda z_3)^2 \delta(u, v).
\]

We simplify the two quadratic phase terms,

\[
e^{j\frac{k}{2z_3}(u^2 + v^2)} e^{-j \left(\frac{k f}{2(z_3 - f)}\right)(u^2 + v^2)} = e^{j \frac{k}{2z_3} \left(\left(\frac{f + z_3 - f}{z_3 - f}\right)\right)(u^2 + v^2)}
\] (2.52)

\[
= e^{j \frac{k}{2z_3} \left(\frac{f + z_3 - f}{z_3 - f}\right)(u^2 + v^2)} = e^{j \frac{k}{2(z_3 - f)(u^2 + v^2)}},
\]

The field at the detector is obtained by completing the convolution

\[
U_{LOpW}(u, v) = e^{jk\hat{n}_b} e^{j\frac{\pi z_3 f}{\lambda z_3 |f - z_3|}} e^{-j\pi \left(\frac{\lambda z_3 f}{2(z_3 - f)}\right)(u^2 + v^2 + z_3 \sin(\phi))^2}
\] (2.53)
2.2.2 Point Source Local Oscillator

The math is simpler for a point source LO, though the system to get a point source is more complicated. It is possible to use a spatial filter or a fiber optic cable to approximate a point source. Whatever the process used to create a point source, the LO’s optical path length needs to be matched close enough to the signal’s path length to be within the temporal coherence length of the laser.

We use a shifted delta function to model a point source. The shift on the point source gives the required tilt to separate the holography images. We will start using the \((\alpha, \beta)\) coordinates which will not necessarily coincide with any of the previously described coordinate systems

\[
U(\alpha, \beta) = \delta(\alpha - \alpha_0, \beta - \beta_0). \tag{2.54}
\]

As a point source radiates into a spherical wave, we do not need a lens to create phase curvature. All we need to do is diffract the source to the focal plane of the imaging system

\[
U(u, v) = \frac{e^{j k z_3}}{j \lambda z_3} \int \int U(\alpha, \beta) e^{j k [u(\alpha - \alpha_0)^2 + (v - \beta_0)^2]} d\alpha d\beta \tag{2.55}
\]

Expand Eq. (2.55) to obtain

\[
U(u, v) = e^{j k z_3 (u^2 + v^2)} \frac{e^{j k z_3}}{j \lambda z_3} \int \int \delta(\alpha - \alpha_0, \beta - \beta_0) e^{j k z_3 (u^2 + v^2)} e^{j [u(\alpha - \alpha_0)^2 + v(\beta - \beta_0)^2]} d\alpha d\beta \tag{2.56}
\]

The delta function makes it so that we can directly do the integral instead of the convolution of two Fourier transforms. We add the subscript “LOps” to this result, which after the integrations is

\[
U_{LOps}(u, v) = e^{j k z_3 (u^2 + v^2)} \frac{e^{j k z_3}}{j \lambda z_3} e^{j k z_3 (\alpha_0^2 + \beta_0^2)} e^{j [u(\alpha - \alpha_0)^2 + v(\beta - \beta_0)^2]} \tag{2.57}
\]
2.3 Mode Matching

We want to match the quadratic phase terms in the LO and the signal beams otherwise the phases will interfere to create fast spatial fringes which are difficult to sample. With a poorly mode matched LO there can be multiple fringes over a single pixel leading to a small average signal with near zero value. This problem is eliminated by matching the spatial phases so they completely overlap. We will derive the mode matching condition for both LO cases.

2.3.1 Mode Matching with a Plane Wave Local Oscillator

Equation (2.58) below is the expanded form of Eq.(2.53) with the temporal component added in and \( f = f_{LO} \)

\[
U_{LOpm}(u,v) = e^{j\delta_{nu,LO}} e^{j\delta_{uv}} e^{j(k_{uv}t + \phi_{uv})} \frac{f_{LO}^{4} (\lambda z_{3})}{\left| f_{LO} - z_{3} \right|} e^{j\frac{k}{2(z_{3} - f_{LO})} \left( (u^{2} + v^{2}) + (u z_{3} \sin(\theta) + v z_{3} \sin(\phi)) \right)}
\]

(2.58)

We will restate Eq. (2.41) here as

\[
U_{s}(u,v) = \frac{1}{(\lambda z_{3}^{2})} e^{j\lambda z_{3}} e^{jk(z_{3}f_{s})} e^{j\delta_{nu,LO}} e^{j(k_{uv}t + \phi_{uv})} u_{L}(u,v) \otimes \mathcal{F}[P(x,y)]
\]

(2.59)

When the two quadratic phase terms equal to each other, i.e.

\[
e^{j\frac{k_{LO}}{2(z_{3} - f_{LO})}(u^{2} + v^{2})} = e^{j\frac{k_{s}}{2}(u^{2} + v^{2})}
\]

(2.60)

the mode matching condition is deduced as

\[
\frac{1}{\lambda_{LO}(z_{3} - f_{LO})} = \frac{1}{\lambda_{s} f_{s}} \text{ or } z_{3} = \frac{f_{s} \lambda_{s}}{\lambda_{LO}} + f_{LO}
\]

(2.61)

The \( \lambda \)'s are removed from the k's because they may be different, based on whether the equation is being used for spatial or temporal heterodyne detection.
2.3.2 Mode Matching with a Point Source Local Oscillator

We rewrite Eq. (2.57) with the temporal component. Notice that we use the same temporal components for both LO. This will allow us to pull it out of a general LO equation in Sections 2.4 and 2.5.

\[ U_{LO_{ps}}(u, v) = e^{i\omega optz + \Phi_z} e^{i k z} e^{i k z} e^{i \alpha^2 + j \beta^2} e^{i \alpha^2 + j \beta^2} \]  

We use the same technique to mode match this LO with the signal beam. The quadratic terms are set equal

\[ e^{i k z (u^2 + v^2)} = e^{2 \alpha u^2 + 2 \beta v^2}, \]  

and the curvature matching condition is deduced as

\[ \frac{1}{z_3 \lambda_{LO}} = \frac{1}{f_s \lambda_s} \quad \text{or} \quad z_3 = \frac{f_s \lambda_s}{\lambda_{LO}}. \]  

2.4 Off-axis Digital Holography

Current camera technology makes recording holograms easy. The term holography means whole picture. Present camera technology cannot image the fields, amplitude and phase. The cameras only record intensity, which erases phase information. Holography uses a local oscillator to capture a picture that can be processed, optically or digitally, to recreate the object with all spatial information present. To get the full information we have to mix the fields together on a detector.

The field intensity at the image plane is given by

\[ |U_{lo}(u, v) + U_s(u, v)|^2 = \left| U_{lo}(u, v) \right|^2 + \left| U_s(u, v) \right|^2 + U_{lo}(u, v) U_s^*(u, v) + U_{lo}^*(u, v) U_s(u, v) \]  

The Fourier transform of \[ |U_{lo}(u, v) + U_s(u, v)|^2 \] gives the separated fields in the pupil plane. This is not actually what the field would be in our pupil plane. The field at our
physical pupil plane does not include the LO beams. As shown in the previous section, the signal and LO beams are mixed on the detector. At the physical pupil plane, they are located in different places. We call this the pupil plane because we get the portion of the field that contains the image information is in the shape of the physical pupil. This is more significant for a system with multiple apertures because this is where the pupil fields are arranged in separate locations to receive the high resolution image.

\[
\mathcal{F} \left\{ \left| U_{lo}(u,v) + U_s(u,v) \right|^2 \right\} = \mathcal{F} \left\{ \left| U_{lo}(u,v) \right|^2 \right\} + \mathcal{F} \left\{ \left| U_s(u,v) \right|^2 \right\} + \mathcal{F} \left\{ U_{lo}(u,v) U_s^*(u,v) \right\} + \mathcal{F} \left\{ U_s(u,v) U_{lo}^*(u,v) \right\}.
\]

The separation is caused by the \( e^{j \left[ k \left( z_3 - z_{lo} \right) \sin(\theta) + v z_3 \sin(\phi) \right]} \) or \( e^{j \left[ \frac{\alpha k}{z_3} \left( vz_3 \right) \right]} \) in plane wave or point source LOs respectively, see Eqs. (2.58) and (2.62) respectively. These terms turn into a translation of the field when Fourier transformed. The position shift due to the Fourier transform is in the opposite direction for the \( U_{lo}(u,v) U_s^*(u,v) \) and the \( U_{lo}^*(u,v) U_s(u,v) \) because of the complex conjugate. The \( \left| U_{lo}(u,v) \right|^2 \) removes the tilt due to the complex conjugate and \( \left| U_s(u,v) \right|^2 \) does not have a tilt term. Equation (2.66) leaves the two irradiance terms near the center, and separates the two mixed terms away from them. This is a benefit because the LO irradiance term is much brighter and, with the signal irradiance, will reduce the signal to noise of the desired image.

The shift in the two terms allows us to crop out one of the mixed terms if the tilts are chosen properly,

\[
\mathcal{F} \left\{ U_s(u,v) U_{lo}^*(u,v) \right\}.
\]
Fig. 2-4: Left: Log scaled magnitude of the field in the pupil plane. Right: Cropped portion of the pupil plane field.

Fig. 2-4 shows a sample pupil plane image on the left and the cropped image of the field described by Eq. (2.67). Once the desired terms are cropped, we perform the inverse Fourier transform. This takes us back to the image plane with knowledge of the field’s amplitude and phase

$$U(u, v) = \mathbb{F}^{-1}\left\{\mathbb{F}\left[U_s(u, v)U_{lo}(u, v)^{*}\right]\right\} = U_s(u, v)U_{lo}(u, v)^{*} \quad (2.68)$$

When we crop the pupil out of the pupil plane, we make sure that the pupil is centered in the crop. This removes the tilt term from the LO beam. We rewrite Eqs. (2.53) and (2.57)

$$U_{LO_{psw}}(u, v) = e^{j\kappa_0} e^{jkz_3} f\left(\lambda z_3^4\right) e^{j\left(k \frac{1}{2(z_3-f)}\right)\left(u^2 + (z_3 \sin(\theta))^2 + v^2 + (z_3 \sin(\theta))^2\right)} \quad (2.69)$$

and

$$U_{LO_{psw}}(u, v) = e^{j\frac{k}{2z_3}(u^2+v^2)} e^{jkz_3} e^{j\frac{k}{2z_3}(\alpha_0^2 + \beta_0^2)} \quad (2.70)$$

We can see from Eq. (2.41) for the signal beam and Eqs. (2.58) and (2.62) for the LO that the temporal components are separable from each equation

$$U(u, v) = \left[U_s(u, v) e^{i\omega_s t}\right] \left[U_{lo}(u, v)^* e^{-i\omega_{lo} t}\right] \quad (2.71)$$

For off-axis holography, $\omega_s = \omega_{lo}$
\[ U(u,v) = U_s(u,v)U_{lo}(u,v)^* \left[ e^{j(\alpha u)} \right] \left[ e^{-j(\alpha v)} \right] = U_s(u,v)U_{lo}(u,v)^* \]  \hspace{1cm} (2.72)

2.5 Temporal Heterodyne

We derive theory for using temporal heterodyne with this system because this is a way to measure piston phase in each sub-aperture, which then would allow us to infer relative phase between sub-apertures. There are significant differences between temporal and spatial heterodyne. For temporal heterodyne, \( \omega_s \) and \( \omega_{LO} \) (or \( \lambda_s \) and \( \lambda_{LO} \)) are different from each other. This also means that \( k_s \) and \( k_{LO} \) are also different. Furthermore, the temporal heterodyne LO is on-axis. We will only derive the plane wave LO because the only difference for the point source LO is that the constant phase terms in \( \alpha_0 \) and \( \beta_0 \) are removed.

2.5.1 Plane Wave Local Oscillator

We start from an earlier equation

\[ U(u,v) = e^{jnh_x} e^{j\frac{L}{2z_3}(u^2+v^2)} e^{j\frac{k_{LO}}{j\lambda z_3}} \mathcal{F}\left\{ e^{j\frac{k_s}{j\lambda z_3}(u^2+v^2)} \right\} \]  \hspace{1cm} (2.44)

\[ \otimes \mathcal{F}\{u_i(x,y)\} \otimes \mathcal{F}\{P(x,y)\} \]

We use the same notation, but change \( U_i(u,v) \).

\[ U_i(x,y) = e^{-jk(x+y)} \]  \hspace{1cm} (2.73)

The Fourier transform of \( U_i(x,y) \) is found as

\[ \mathcal{F}\{U_i(x,y)\} = \mathcal{F}\{e^{-jk(x+y)}\} = \delta\left( \frac{u}{\lambda z_3} + \frac{1}{\lambda}, \frac{v}{\lambda z_3} + \frac{1}{\lambda} \right) = (\lambda z_3)^2 \delta(u + z_3, v + z_3) \]  \hspace{1cm} (2.74)

Now Eq. (2.74) is substituted into Eq. (2.44) with the other Fourier transforms found in section 2.2.1.
\begin{equation}
U_{LO}(u,v) = e^{jk_nu} e^{j\frac{f}{2z_3}(u^2+v^2)} e^{j\pi z_j f} e^{-j\left(\frac{hf}{2z_3(f-z_3)}\right)(u^2+v^2)} \delta(z_3) \delta(u+z_3,v+z_3) (\lambda z_5)^2 \delta(u,v).
\end{equation}

The same cleanup as in Eq. (2.52) is applied to Eq. (2.75). Convolting and adding temporal component we have

\begin{equation}
U_{LO}(u,v) = e^{jk_{lo}u} e^{j\pi z_j f} f_{lo} \left(\lambda_{lo} z_3\right)^4 e^{-j\frac{k_{lo}}{2(z_3-f_{lo})}(u^2+v^2)} U_{lo}(u,v) + \lambda_{lo} z_3 U_s(u,v).
\end{equation}

### 2.5.2 Temporal Heterodyne Temporal Components

When we mix the signal and LO in temporal heterodyne, we are not able to separate the mixed signal term up like we could in Section 2.4 because the LO is not angle off axis. This does not matter because we use a single pixel detector for temporal heterodyne detection. Because the detector is a single pixel, detectors are available with bandwidths high enough to detect the beat frequency between the on-axis LO and the signal beam. This means that we do not have to time average the intensity of the mixed signal and LO beams. We will pull the time components out of the mixed field.

\begin{equation}
U_f = U_{lo}(u,v) + U_s(u,v) = U_{lo}(u,v) e^{j\omega_{lo} t} + U_s(u,v) e^{j\omega_s t}.
\end{equation}

We want to get the difference of the \(\omega\)'s into the phase terms

\begin{equation}
U_f = \left( U_{lo}(u,v) e^{j\frac{\omega_s - \omega_{lo}}{2}} + U_s(u,v) e^{j\frac{\omega_s + \omega_{lo}}{2}} \right) e^{j\frac{\omega_s + \omega_{lo}}{2}}.
\end{equation}

Now that we have the difference of the two \(\omega\)'s, we get the intensity.
\[ |U_f|^2 = \left( U_{lo}(u,v) e^{-j\frac{\omega_s - \omega_{LO}}{2}} + U_s(u,v) e^{j\frac{\omega_s - \omega_{LO}}{2}} \right)^2 \]

\[ |U_f|^2 = \left| U_{lo}(u,v) e^{-j\frac{\omega_s - \omega_{LO}}{2}} + U_s(u,v) e^{j\frac{\omega_s - \omega_{LO}}{2}} \right|^2 \]

\[ |U_f|^2 = \left( |U_{lo}(u,v)|^2 + |U_s(u,v)|^2 + U_{lo}^*(u,v) U_s(u,v) e^{j\left((\omega_s - \omega_{LO})t\right)} \right)^2 \]

\[ |U_f|^2 = \left( |U_{lo}(u,v)|^2 + |U_s(u,v)|^2 + 2U_{lo}(u,v) U_s(u,v) e^{2j\left((\omega_s - \omega_{LO})t\right)} \right), \]

If we assume that the mode matching condition is met and collect the constant phase terms into a constant cosine term, C, we get

\[ |U_f|^2 = \left( |U_{lo}(u,v)|^2 + |U_s(u,v)|^2 + 2CU_{lo}(u,v) U_s(u,v) \left( e^{j\left((\omega_s - \omega_{LO})t\right)} + e^{-j\left((\omega_s - \omega_{LO})t\right)} \right) \right)^2 \] (2.79)

If we keep the wavelength of the temporal LO close enough to the signal wavelength then we can obtain detectors that will measure this beat frequency, allowing us to measure temporal phase variations in the returned signal. The cosine term carries the crux of our piston measurement. If \( \omega_s \) and \( \omega_{LO} \) are stable, i.e. no varying non-common path, then the beat frequency phase directly corresponds to the optical phase of the signal beam. This allows us to measure the piston to high accuracy and the offset frequency of the LO does not effect that accuracy.
2.6 Speckle Averaging

Up to this point we have ignored laser speckle. Laser speckle is caused by a laser beam hitting a surface that is rough at least on the order of the laser’s wave length. The roughness causes the beam to interfere with itself, causing a granular interference pattern. This granularity in the perceived Illumination beam, known as speckle, causes granularity in the image. We desire to reduce the speckle in the images that we take. In this section, we will follow closely Lowenthal\textsuperscript{10}.

2.6.1 Object Field Statistics

We make several assumptions about the input of the imaging system. First, the target is a diffuse system. This can either be a diffuse reflective target, or a diffuser with a transparent target place directly against it. We choose the latter so that we can assume the diffuser to be stationary and have independent Gaussian statistics for both the amplitude and phase. For simplicity, we assume that the magnification of our system is 1 as we did in Section 2.1.3 Eq. (2.34). Because the magnification is 1, the \((\xi, \eta)\) and \((u, v)\) coordinates are changed to \((r)\). Look to the subscripts to tell where in the system we are working. We change Eq. (2.34) to the \((r)\) coordinate system

\[
U_i(r) = U_o(r) \otimes h(r). \tag{2.82}
\]

We define \(u_o(r)\) to be the diffuser, \(d(r)\), multiplied by the transparency, \(t(r)\), because the two are pressed together

\[
U_o(r) = t(r)d(r). \tag{2.83}
\]

For reasons that will become clear latter, we will find the first and second order statistics for \(U_o(r)\). The mean value of \(U_o(r)\) is given by

\[
\overline{U_o(r)} = \overline{t(r)d(r)} = \overline{t(r)d(r)} = 0. \tag{2.84}
\]
The overbar, \( \overline{\cdot} \), denotes the ensemble average. \( t(r) \) is its own mean value as it is assumed to be a constant. The mean value of \( d(r) \) is zero because it is a Gaussian random variable centered at zero.

Next we find the statistical autocorrelation

\[
R_{u,u_o}(r_1,r_2) = \overline{t(r_1)d(r_2)\overline{t(r_2)}} = t(r_1)\overline{t(r_2)}d(r_2)\overline{d(r_2)},
\]

\[
R_{u,u_o}(r_1) = t(r_1)\overline{t(r_2)}R_{dd}(r_1,r_2).
\]

The \( R_{(.)} \) denotes a correlation and the \( \overline{\cdot} \) denote the complex conjugate. \( R_{dd}(r_1,r_2) \) can be written as \( R_{dd}(r_1-r_2) \) due to \( d(r) \) being a stationary variable, meaning the correlation depends on the differences in \( r \) and not the actual values. At this point we assume that \( R_{dd}(r_1-r_2) = \delta(r_1-r_2) \) as is consistent with the assumption that speckle, which the diffuser creates, is white noise.

\[
R_{u,u_o}(r_1,r_2) = t(r_1)\overline{t(r_2)}R_{dd}(r_1,r_2) = t(r_1)\overline{t(r_2)}\delta(r_1-r_2).
\]

The variance, \( \sigma_{(.)} \), of \( u_o(r) \) is unimportant here but will be included for completeness

\[
\sigma_{u_i} = \overline{U_o(r)}^2 - \overline{U_o(r)}^2 = R_{u,U_o} - 0 = R_{u,u_o}.
\]

2.6.2 Image Field Statistics

Moving to \( U_i(r) \), we will start with the mean value

\[
\overline{U_i(r)} = \overline{U_o(r)} \otimes h(r) = 0 \otimes h(r) = 0.
\]

Like \( t(r) \), \( h(r) \) is a constant and therefore its own mean value. As shown is Eq. (2.84), \( \overline{U_o(r)} = 0 \), thus \( \overline{U_i(r)} = 0 \). The variance of the \( U_i(r) \) is more interesting
\[ \sigma_i = \left| U_i(r) \right|^2 - \overline{U_i(r)}^2 = \overline{I(r)} - 0 = \overline{I(r)}. \] (2.89)

\( I(r) \) is the intensity image. The autocorrelation is more complicated.

\[ R_{w,w_i}(r_1,r_2) = U_o(r_1) \otimes h(r_1)U_o^*(r_2) \otimes h^*(r_2). \] (2.90)

We will expand this into the convolution integrals

\[ R_{w,w_i}(r_1,r_2) = \int_{-\infty}^{\infty} U_o(r_0)h(r_1-r_0)dr_0 \int_{-\infty}^{\infty} U_o^*(r_0^0)h^*(r_2-r_0^0)dr_0. \] (2.91)

\( U_o^*(r_0^0) \) can be pulled into the other integral since \( r_0^0 \) is independent of \( r_0 \). The h’s can be pulled out of the ensemble average

\[ R_{w,w_i}(r_1,r_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_{w,w_o}(r_0^0,r_0^0)h(r_1-r_0)h^*(r_2-r_0^0)dr_0^0 dr_0. \] (2.92)

Notice that Eq. (2.92) contains the autocorrelation of \( \overline{U_o(r)} \)

\[ R_{w,w_i}(r_1,r_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} R_{w,w_o}(r_0^0,r_0^0)h(r_1-r_0) \otimes h(r_2-r_0^0)dr_0^0 dr_0. \] (2.93)

We pull Eq. (2.93) back into symbolic convolution form to see the result

\[ R_{w,w_i}(r_1,r_2) = R_{w,w_o}(r_1,r_2) \otimes h(r_1) \otimes h^*(r_2). \] (2.94)

Expand \( R_{w,w_o}(r_1,r_2) \) using Eq. (2.86) but we only convert the first convolution into integral form

\[ R_{w,w_i}(r_1,r_2) = \int_{-\infty}^{\infty} t(r_0) t^*(r_1-r_0^2)h^*(r_2-r_0^2)dr_0 \otimes h^*(r_2). \] (2.95)

The delta function eliminates the integral

\[ R_{w,w_i}(r_1,r_2) = t(r_2) t^*(r_2)h(r_1-r_2) \otimes h^*(r_2). \] (2.96)

We write the second convolution in integral form
\[ R_{\mu_1,\mu_2}(r_1, r_2) = \int_{-\infty}^{\infty} t(r_0) t^*(r_0) h(r_1 - r_0) h^*(r_2 - r_0) dr_0 \]  

(2.97)

Now we assume that \( r_1 = r_2 = r \).

\[ R_{\mu_1,\mu_2}(r_1, r_2) = \int_{-\infty}^{\infty} t(r_0) t^*(r_0) h(r - r_0) h^*(r - r_0) dr_0 \]

\[ R_{\mu_1,\mu_2}(r_1, r_2) = \int_{-\infty}^{\infty} |t(r)|^2 |h(r)|^2 dr_0 \]  

(2.98)

The final form of \( R_{\mu_1,\mu_2}(r_1, r_2) \) is Eq. (2.98) rewritten in symbolic convolution form

\[ R_{\mu_1,\mu_2}(r_1, r_2) = |t(r)|^2 \otimes |h(r)|^2 \]  

(2.99)

Looking back at Eq. (2.89), the first, and only term with a value, is \( |\overline{U_i(r)}|^2 \) which is a condensed form of \( R_{\mu_1,\mu_2}(r_1, r_2) \)

\[ R_{\mu_1,\mu_2}(r_1, r_2) = |t(r)|^2 \otimes |h(r)|^2 = |\overline{U_i(r)}|^2 = \overline{I(r)} \]  

(2.100)

Because \( \overline{I(r)} \) is the average coherent intensity, we will rename it \( \overline{I_c}(r) \). Lowenthal\(^{10} \) gives the incoherent intensity as their equation 14, which we express as

\[ I_{IC}(r) = L(r) \otimes |h(r)|^2 \]  

(2.101)

Comparing Eqs. (2.100) and (2.101), we can see that \( |t(r)|^2 = L(r) \). This means that the average coherent intensity is equal to the incoherent intensity. At this point we want to reintroduce the \((x, y)\) and \((u, v)\) coordinate systems.

\[ I_{IC}(u, v) = \overline{I_c}(u, v) = |t(u, v)|^2 \otimes |h(u, v)|^2 \]  

(2.102)
An another important conclusion from this section is found in Eq. (2.102). This is the average intensity. This means that when we speckle average, we do so using the intensity images instead of the fields.

### 2.7 Transfer Functions

In an optical system, there are many ways to judge the performance of the system. Transfer functions are used to describe the spatial frequencies that are passed through the system. If we are only looking at the optics of a coherent system then we use the coherent transfer function (CTF). The CTF is equal to the pupil function. For an incoherent system, we use the optical transfer function (OTF), or the modulation transfer function (MTF). The OTF is equal to the normalized spatial autocorrelation of the pupil function. The MTF is the modulus of the OTF.

We would like to introduce the OTF into the average intensity given in Eq. (2.100). The transfer function is the Fourier transform of the impulse response. The impulse response was given in Eq. (2.31). We can see that the impulse response is essentially the Fourier transform of the pupil function

\[
h(u - M\xi, v - M\eta) = e^{jK(z_1 + z_2)}e^{jKz_2} \frac{Me^{jK(z_2^2 + v^2)}}{(\lambda z_2)^2} \mathcal{F}\left[ P(x, y) \right] \bigg|_{f_x = (u-M\xi), f_y = (v-M\eta)} \frac{z_2}{z_2^2} (2.31)
\]

Since we are working with the average intensity, which has had the signal and LO beams mixed, we can get rid of the quadratic phase term, if the mode matching condition has been met. The other terms in the impulse response are constants and will be given a symbol of c1 for this section

\[
h(u - M\xi, v - M\eta) = c_1 \mathcal{F}\left[ P(x, y) \right] \bigg|_{f_x = (u-M\xi), f_y = (v-M\eta)} \frac{z_2}{z_2^2} (2.103)
\]
The Fourier transform of the impulse response is called the transfer function, which is written as

\[ H(x, y) = \mathcal{F}\left[ h(u - M\xi, v - M\eta) \right] \],

or

\[ H(x, y) = \mathcal{F}\left[ \mathcal{F}^{-1}[P(x, y)] \right] \left[ \frac{f_x = (u - M\xi)}{z^2} \frac{f_y = (v - M\eta)}{z^2} \right] \] (2.104)

This function is easy to calculate

\[ H(x, y) = c1 \int_{-\infty}^{\infty} e^{-j2\pi \left( (\alpha f_x + \beta f_y) \right)} \int_{-\infty}^{\infty} P(\alpha, \beta) e^{-j2\pi \left( \alpha f_x + \beta f_y \right)} d\alpha d\beta df_x df_y \] (2.105)

We rearrange the order of integration as

\[ H(x, y) = c1 \int_{-\infty}^{\infty} P(\alpha, \beta) \int_{-\infty}^{\infty} e^{-j2\pi \left( (\alpha x + \beta y) \right)} df_x df_y d\alpha d\beta \] (2.106)

The integral in \((f_x, f_y)\) is given in Eq. (2.45), though in a different coordinate system

\[ H(x, y) = c1 \int_{-\infty}^{\infty} P(\alpha, \beta) \delta((\alpha + x), (\beta + y)) d\alpha d\beta \]

or simply

\[ H(x, y) = c1P(-x, -y) \] (2.107)

We will drop the negatives on the pupil function because pupils are generally circular, and circles are even functions. Also, most multi-aperture arrays are symmetrical through the vertical axis and therefore even

\[ H(x, y) = c1P(x, y) \] (2.108)

Taking the Fourier transform of a function twice only inverts the coordinates of the function. Using the transfer function \(H(x, y)\) we get the pupil function out an integral.
Using mathematical gymnastics, we can get the autocorrelation of the pupil function. We start by taking the Fourier transform and the inverse Fourier transform of the average intensity

$$
\mathcal{F}^{-1}\left[\mathcal{F}\left[\bar{I}(u,v)\right]\right] = \mathcal{F}^{-1}\left[\mathcal{F}\left[t(u,v)\right]^2 \otimes |h(u,v)|^2\right].
$$

(2.109)

By the convolution theorem, the Fourier transform of two functions convoluted is equal to the individual Fourier transforms of the two functions multiplied together.

$$
\mathcal{F}^{-1}\left[\mathcal{F}\left[\bar{I}(u,v)\right]\right] = \mathcal{F}^{-1}\left[\mathcal{F}\left[t(u,v)\right]^2\right] \mathcal{F}\left[|h(u,v)|^2\right].
$$

(2.110)

Also by the convolution theorem, the Fourier transform of two functions multiplied together is the convolution of the Fourier transforms of those two functions. We will only concern ourselves with the term containing the impulse response

$$
\mathcal{F}\left[|h(u,v)|^2\right] = \mathcal{F}\left[h(u,v)\right] \otimes \mathcal{F}\left[h^\ast(u,v)\right].
$$

(2.111)

From Eq. (2.104), we know the Fourier transform of the impulse response. Also, the Fourier transform of a complex conjugate is equal to complex conjugate of the Fourier transform with negative values of the coordinates

$$
\mathcal{F}\left[|h(u,v)|^2\right] = H(x,y) \otimes H^\ast(-x,-y),
$$

$$
\mathcal{F}\left[|h(u,v)|^2\right] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H^\ast(-\alpha,-\beta) H(x-\alpha,y-\beta) d\alpha d\beta.
$$

(2.112)

Use the variable substitution $p = -\alpha$ and $q = -\beta$. Note that the flip from the variable substitution of the limits of integration cancel with the negative signs introduced in the differentials

$$
\mathcal{F}\left[|h(u,v)|^2\right] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H^\ast(p,q) H(x+p,y+q) dp dq.
$$

(2.113)
We used the statistical autocorrelation starting in Eq. (2.85), but Eq. (2.113) is the form of the spatial autocorrelation. We will always say spatial for the spatial autocorrelation. We will use the * for the spatial autocorrelation

\[ \mathcal{F}\left[h(u,v)\right]^2 = H(x,y) * H^*(x,y). \quad (2.114) \]

Remember that the OTF is defined as the normalized spatial autocorrelation of the pupil function. The transfer function is given in Eq. (2.108). We will change \( c_1^2 \) to \( c_2 \) adding the normalization factor of the OTF, which is the maximum value of the OTF

\[ c_2 = c_1^2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |H(x,y)|^2 \, dx \, dy = c_1^4 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |P(x,y)|^2 \, dx \, dy \quad (2.115) \]

\[ \mathcal{F}\left[h(u,v)\right]^2 = c_1^2 P(x,y) * P^*(x,y) = c_2 OTF. \quad (2.116) \]

This changes the average intensity after speckle averaging given in Eq. (2.110)

\[ \mathcal{F}^{-1}\left[\mathcal{F}\left[I(u,v)\right]\right] = \mathcal{F}^{-1}\left[\mathcal{F}\left[I(u,v)\right]\right] c_2 OTF. \quad (2.117) \]

Using the convolution theorem again, we get

\[ I(u,v) = \mathcal{F}^{-1}\left[\mathcal{F}\left[I(u,v)\right]\right] = I(u,v)^2 \otimes c_2 OTF. \quad (2.118) \]
CHAPTER 3
MULTI-APERTURE SYSTEM THEORY

3.1 Pupil Considerations

3.1.1 Resolution Dependence on Pupil

We find the resolution of a single image is based on the impulse response found in Eq. (2.37) and its change due to convolution

\[
\tilde{h}(u,v) = e^{j(k_z z_f)} e^{jka_0} \frac{Me^{2j\pi z_f}}{|M|} \left( \lambda z_f \right)^2 \mathcal{F}\left[P(x,y)\right] \bigg|_{f_x = (u \zeta^2, v \zeta^2)}.
\]  

(2.37)

We will assume that the pupil function is a circle, where \( D \) is the diameter of the circle

\[
P(x,y) = \text{circ} \left( \frac{2\sqrt{x^2 + y^2}}{D} \right).
\]

(3.1)

The Fourier transform of a circle function is given as

\[
\mathcal{F}\left[ \text{circ} \left( \frac{2r_{x,y}}{D} \right) \right] = a \frac{2J_1(\pi Ds)}{\pi Ds}.
\]

(3.2)

where \( \mathcal{F} \) stands for the Bessel transform, which is the circularly symmetric version of the Fourier transform, and \( J_1 \) is the Bessel function of the first kind and first order. Now define \( s = \sqrt{f_x^2 + f_y^2} = \frac{1}{\lambda z_f} \sqrt{u^2 + v^2} \) and \( a = \pi \left( \frac{D}{2} \right)^2 \) to get the result
\[ \hat{h}_{uv} e^{jk(z_1+z_2)} e^{jkn\lambda_0} \left| M \right| \left( \frac{\lambda z_2}{2} \right)^2 \pi \left( \frac{D}{2} \right)^2 \frac{2J_1 \left( \frac{\pi Dr_{w,v}}{\lambda z_2} \right) (\lambda z_2)}{\pi Dr_{w,v}} . \]  

(3.3)

We simplify Eq. (3.3) to the form

\[ \hat{h}_{uv} e^{jk(z_1+z_2)} e^{jkn\lambda_0} \left( \frac{\lambda z_2}{2} \right)^2 J_1 \left( \frac{\pi Dr_{w,v}}{\lambda z_2} \right) \frac{1}{r_{w,v}} . \]  

(3.4)

The Bessel function is a decaying oscillatory function which has its first zero when the argument is equal to 3.83.

\[ \frac{\pi Dr}{\lambda z_2} = 3.83 , \]

or

\[ r = \frac{1.22\lambda z_2}{D} \]  

(3.5)

r, in Eq. (3.5), is the maximum value of \( r_{w,v} \) that fits the zero of the Bessel function. As D gets larger, r gets smaller, meaning, as the imaging lens gets bigger, the smaller the spot size. In other words we will be able to resolve finer features.

### 3.1.2 Pupil Size Constraints

The larger the pupil is, the more it costs and the greater the weight. For long distant imaging, the imaging lens needs to be larger than for close imaging due to the factor of \( z_2 \) in the numerator of Eq. (3.5).

### 3.2 Multiple Aperture Arrays

One of the ways to cut down on weight is to create an array of smaller sub-apertures. These arrays would have the same maximum spatial frequency, but will not necessarily contain the same intermediate frequency content as an equally sized
monolithic lens. There will be lower signal to noise at some spatial frequencies due to the fill factor necessarily being reduced by the gaps between the sub-apertures.

In section 2.7, we added the OTF into our imaging equation. We did this to show that relationship of the OTF to the imaging equation, but the true consequence is that we linked the MTF of the system to the imaging equations. Recall that the MTF is the modulus of the OTF. The MTF describes the spatial frequencies that can be found in the image based on the pupil array. To get the MTF we look back at Eqs. (2.115) and (2.116) rewritten here as

\[
c2 = c1^2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |H(x, y)|^2 \, dx \, dy = c1^4 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |P(x, y)|^2 \, dx \, dy, \tag{2.115}
\]

and

\[
\mathcal{F} \left[ |h(u, v)|^2 \right] = c1^2 P(x, y)^* P^*(x, y) = c2 \text{OTF} \tag{2.116}
\]

We redefine \( P(x, y) \) as

\[
P(x, y) = \sum_{i=1}^{q} P_i(x - x_i, y - y_i), \tag{3.6}
\]

where \( q \) is the total number of sub-apertures and \( i \) is the specific sub-aperture. The OTF is given as

\[
\text{OTF} = \frac{\sum_{i=1}^{q} P_i(x - x_i, y - y_i)^* \sum_{d=1}^{q} P_d(x - x_d, y - y_d)}{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left( \sum_{i=1}^{q} P_i(x - x_i, y - y_i) \right)^2 \, dx \, dy}. \tag{3.7}
\]

The MTF is the modulus of the OTF and is
MTF = \left| OTF \right| = \left| \sum_{i=1}^{g} P_i (x-x_i, y-y_i) * \sum_{d=1}^{g} P_d (x-x_d, y-y_d) \right| \left( \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left| \sum_{i=1}^{g} P_i (x-x_i, y-y_i) \right|^2 \, dx \, dy \right).

(3.8)

The code, to create and plot the multi aperture array MTFs, is found in the Appendix A.4. We will not discuss it because it follows Eq. (3.8), though it looks different. A section on pixel size and spatial frequency conversions from plane to plane is found in appendix B, as the results found there are used in the MTF code as well as in the propagation and processing codes.

We use 4 different pupil arrays in this thesis; 2 in-line sub-apertures, 3 in-line sub-apertures, a hex 7 array and a hex 19 array. We will show images for each pupil array of the 2D MTF, the a slice of the MTF, through the center, for the horizontal and vertical dimensions, and the MTF of the vertical and horizontal dimensions and the MTF of a monolithic aperture that has a diameter equal to that of the largest dimension of the pupil array. We will leave off the vertical MTF plot for the in-line arrays as there is little benefit. They will still be included in the combined MTF images. The horizontal and vertical parts of the combined MTF images are not strictly MTFs for the combined MTF plots. We normalized the horizontal and vertical MTFs with the same normalizing factor used for the monolithic aperture MTF so that we could relate the frequency response levels. When we originally plotted the strict MTFs together, we a saw higher frequency response in sections the vertical dimension’s MTF than in the monolithic MTF plot. Although technically correct due to normalization, the true frequency response of the monolithic aperture will always be greater than the frequency responses of the multi aperture array with the same diameter, because of fill factor loss. In the combined MTF
plots, the highest line is the MTF for the monolithic aperture and the horizontal slice peaks first. The vertical slice has fewer peaks than the horizontal slice.

Fig. 3-1: Top: 2D MTF of 2 in-line sub-apertures. Bottom Left: Horizontal slice of MTF for of 2 in-line sub-apertures. Bottom Right: MTF comparison plot of horizontal and vertical slices of the 2 in-line sub-apertures and a monolithic aperture with the same radius of the max dimension of the 2 in-line sub-aperture array.
Fig. 3-2: Top: 2D MTF of 3 in-line sub-apertures. Bottom Left: Horizontal slice of MTF for 3 in-line sub-apertures. Bottom Right: MTF comparison plot of horizontal and vertical slices of the 3 in-line sub-apertures and a monolithic aperture with the same radius of the max dimension of the 3 in-line sub-aperture array.
Fig. 3-3: Top Left: 2D MTF of hex 7 sub-aperture array. Top Right: Horizontal slice of MTF for a hex 7 sub-aperture array. Bottom Left: Vertical slice of MTF for a hex 7 sub-aperture array. Bottom Right: MTF comparison plot of horizontal and vertical slices of the Horizontal slice of MTF for a hex 7 sub-aperture array and a monolithic aperture with the same radius as the max dimension of the hex 7 sub-aperture array.
Fig. 3-4: Top Left: 2D MTF of hex 19 sub-aperture array. Top Right: Horizontal slice of MTF for a hex 19 sub-aperture array. Bottom Left: Vertical slice of MTF for a hex 19 sub-aperture array. Bottom Right: MTF comparison plot of horizontal and vertical slices of the Horizontal slice of MTF for a hex 19 sub-aperture array and a monolithic aperture with the same radius as the max dimension of the hex 19 sub-aperture array.

From the Combined MTF plot, we see that the lower fill factor in the sub-aperture arrays causes a decrease in frequency response. The difference in the maximum renormalized MTF values between the monolithic aperture and the multi-aperture cases is the ratio of their areas, which by definition is the fill factor.

3.3 Phase Errors

An issue that comes with multi-aperture arrays is the addition of phase errors. Multi-aperture arrays have all the possible problems with a single aperture, but now you have a lot of small sub-apertures. Spherical aberration is decreased as a lens gets
larger due to more of the light being in the paraxial region. The chance of different lens 
errors, like coma and astigmatism, is multiplied by having more lenses in the system.

3.3.1 Multiple Aperture Synthesis and Phase Errors

Using multiple aperture systems create issues not encountered in monolithic 
aperture systems, as the images from these arrays must be digitally arranged into a 
single image\textsuperscript{16}. Aperture synthesis is performed by post processing the multi-aperture 
images. This requires computer time and power to align the phases of all sub-apertures 
to within fractions of a wavelength. The process is further complicated for wavelengths in 
the optical regime because detector arrays do not exist that directly measure the optical 
phase. High speed detectors, detectors that work in the gigahertz range, have not been 
made into an array suitable for temporal heterodyne detection.

Through off-axis digital holography, we can measure the spatial phase incident 
on a single detector array. This is a somewhat misleading statement though. We know 
the point by point complex field values within a given sub-aperture, but this is not usable 
information by itself. The field is superimposed on top of noise in a way that cannot 
easily be recognized. When using a multi-aperture system, the phase relationships 
between sub-apertures are not measured.

These unknown phases turn into a problem for aperture synthesis. To 
understand this we rewrite Eqs. (2.41), (2.70), and (2.72)

\[
U_s(u,v) = \frac{1}{\lambda z_2^2} e^{j k (u^2 + v^2)} e^{j k(z_i + f_i)} e^{j k n \phi} U_g(u,v) \otimes \mathcal{F}[P(x,y)] \tag{2.41}
\]

\[
U_{LOps}(u,v) = e^{j k z_3} e^{j k z_i} e^{j k (a_i^2 + b_i^2)} \tag{2.70}
\]

and

\[
U(u,v) = U_s(u,v) U_{lo}(u,v)^* \tag{2.72}
\]
Insert Eqs. (2.41) and (2.70) into (2.72). It does not matter which spatial heterodyne LO approach we use here, so we will use the point source LO

\[ U(u, v) = \frac{1}{(\lambda z_2)} e^{\frac{ik}{2z_2}(u^2 + v^2)} e^{jk(z_2 + f_1)} e^{jkn_0} U_g(u, v) \]

\[ \otimes \mathcal{F} \left[ P(x, y) \right] e^{-j \frac{k}{2z_2}(u^2 + v^2)} e^{-jkz_1} e^{-j \frac{k}{2z_3}(\alpha_0^2 + \beta_0^2)} \]

We combine terms, assuming the mode matching condition is met

\[ U(u, v) = \left\{ \frac{e^{\frac{jk}{2z_2}P(x, y)}}{\lambda z_2^2 z_3} \right\} \left[ e^{jk(z_2 + f_1 + z_3)} e^{-j \frac{k}{2z_3}(\alpha_0^2 + \beta_0^2)} \right] U_g(u, v) \otimes \mathcal{F} \left[ P(x, y) \right] \]

The curved bracket section of Eq. (3.10), contains a non-essential amplitude and fixed phase terms that do not change. The square bracketed terms have values that can change, and together is the piston phase. For a single pupil, the phase terms disappear when you get the intensity. For a multi-aperture system, the terms after the curved brackets are different for each sub-aperture. Each aperture needs to have separate LOs so that the angle with the detector array is at least close for every sub-aperture. \( \alpha_0, \beta_0, z_1, \) and \( z_3 \) are all lengths that are almost impossible to get exactly equal for each sub-aperture.

For one aperture, our pupil function was a single circle that had several phase aberrations built into it. Piston is present, but it disappeared when the intensity was captured by the detector array.

With a multi-aperture array, we have a summation of the different sub-apertures. Since the most careful alignment of the sub-apertures will not get rid of all the phase errors, we have to capture the field using our off-axis digital holography system to obtain the spatial phase information of each low resolution image. Without the full field information, including phase, we cannot add the different sub-apertures in the pupil.
plane to obtain a high resolution image. Each sub-aperture is imaged in the image plane onto a separate detector array. We Fourier transform the individual sub-apertures to the pupil plane. The Fourier transform of the image is the field at the imaging lens, otherwise known as the pupil plane. We use the convolution theorem to perform the Fourier transform

$$\mathcal{F}[U(u, v)] = \left( e^{\frac{j2\pi}{\lambda} e^{jkz_0}} e^{\frac{j}{\lambda z_2^2 z_3}} \right) e^{jk(z_i + f_i)} e^{-jkz_i} e^{-\frac{k}{2z_i}(a_i^2 + b_i^2)} \mathcal{F}[U_g(u, v)] P(x - x_0, y - y_0) \quad (3.11)$$

We stitch the sub-apertures together in the pupil plane, aligning them by the physical separation distance

$$U(x, y) = \sum_{i=1}^{n} \left( e^{\frac{j2\pi}{\lambda} e^{jkz_0}} e^{\frac{j}{\lambda z_2^2 z_3}} \right) e^{jk(z_i + f_i)} e^{-jkz_i} e^{-\frac{k}{2z_i}(a_i^2 + b_i^2)} \mathcal{F}[U_g(u, v)] P_i(x - x_i, y - y_i) \quad (3.12)$$

Now that we have the stitched together fields, we inverse Fourier transform the stitched pupil fields back to the image plane

$$U(u, v) = \sum_{i=1}^{n} \left( e^{\frac{j2\pi}{\lambda} e^{jkz_0}} e^{\frac{j}{\lambda z_2^2 z_3}} \right) e^{jk(z_i + f_i)} e^{-jkz_i} e^{-\frac{k}{2z_i}(a_i^2 + b_i^2)} U_g(u, v) \mathcal{F}[P_i(x - x_i, y - y_i)] \quad (3.13)$$

This looks simple in analytical equation, but the images we get from the detector arrays are not easily turned into equation. As stated before, the field in the pupil plane is a point by point complex spatial field due to the use of a detector array. The analytical equations show that there are many complex portions superimposed on top of each other. There are also parts of the field that we have not described yet.

The missing parts of the field are found in $P_i(x - x_i, y - y_i)$. We assumed that the pupil’s shape is a circle. There are phase terms built into the pupil function
\[ \mathcal{F} \left[ P_i(x-x_i, y-y_i) \right] = \mathcal{F} \left[ P_i(x, y) \right] e^{j \frac{1}{2\pi d} (x u + y v)} \]  

(3.14)

A translation in a function is transformed through a Fourier transform into a phase tilt.

There are other phase terms, known as phase errors because they alter the received image. There are two categories of phase errors. There are static phase errors and temporal phase errors.

### 3.3.2 Static Phase Errors

Static phase errors are errors that are inherent to a particular sub-aperture. These errors do not change between different images. With perfect conditions, these are the only phase errors present.

There are two types of statics phase errors, sub-aperture misalignment and lens aberrations. Sub-aperture misalignment comes from the detector array being shifted off the optic axis, or the entire sub-aperture being pointed so that the object is not quite on the sub-aperture’s optic axis. These show up as tip and tilt in the pupil plane, which correspond to a shift in the image at the image plane

\[ P_i(x, y) = \text{circ} \left( \frac{2r}{d} \right) e^{j \frac{1}{2\pi d} (u x + v y)} \]  

(3.15)

Lens aberrations come from imperfect lens construction. These are problems like astigmatism, coma, and spherical aberration. For this thesis we will assume that we have none of these phase errors. They could be added in a similar fashion to the tip and tilt, using the corresponding Zernike polynomials, which will be discussed in section 3.4.1.

Unlike piston, which is also a phase error, other phase errors are not removed by finding the intensity for a single aperture system. These errors cause the image to look different then the object, due to the lens errors, and in a different place in the image, due to the sub-aperture misalignment errors. These errors do not affect speckle averaging.
3.3.3 Temporal Phase Errors

Temporal phase errors change between every image. There are two types of temporal phase errors. The first type is atmospheric. We will ignore this error, as it is a complicated nightmare of its own. The second type is systematic change. This is due to movements of individual parts of the system. Most of the time, these movements are caused by vibration. Serious vibrations, or large high speed vibrations, can cause the image to washout. Slower vibrations give us temporal tip, tilt and piston. Looking back at Eq. (3.15), we added these temporal phase errors. The subscript “j” denotes a different speckle realization. Many of the terms have subscripts of “ij” because they are different for every sub-aperture and speckle realization

\[ P_{i,j}(x, y) = \text{circ} \left( \frac{2r}{d} \right) e^{j \frac{1}{\lambda (z_{2i} + \Delta z_{2i})} \left[(u_{i} + u_{j}) x + (v_{i} + v_{j}) y \right]} \] (3.16)

Piston error changes everything. Though we had some static piston error, shown in Eq. (3.10), the problem with temporal piston is that it can change \( z_1 \), \( z_2 \) and \( z_3 \). Up until this point we have assumed \( z_2 = f_s \), and that it was easy enough to align that it would not make a difference. With the addition of temporal piston error, we will rewrite \( f_s \) as \( z_2 \)

\[ U(u, v) = \sum_{i=1}^{n} \left( \frac{jz}{\lambda^3 z_1^2 z_2} \right) e^{j \beta(z_{1i} + \Delta z_{1i} + z_{2i} + \Delta z_{2i} - z_{1j} - \Delta z_{1j})} e^{-j \frac{\lambda}{2} \left[(\alpha_i + \Delta \alpha_i)^2 + (\beta_i + \Delta \beta_i)^2\right]} \]

and the irradiance is

\[ G(u, v) \otimes \mathbb{G} \left[ \text{circ} \left( \frac{2r}{d} \right) e^{j \frac{1}{\lambda (z_{2i} + \Delta z_{2i})} \left[(u_{i} + u_{j}) x + (v_{i} + v_{j}) y \right]} \right] e^{j \frac{\lambda}{2} \left[(\alpha_i + \Delta \alpha_i)^2 + (\beta_i + \Delta \beta_i)^2\right]} \] (3.17)
As deduced from Eq. (3.17), the field for one single speckle realization of a multi-aperture system is complicated. This is why there are more steps to post processing than just stitching the pupils together.

### 3.4 Eliminating Phase Errors

Since the most careful alignment of the sub-apertures will not get rid of all the phase errors, we capture the field using our off-axis digital holography system and get the spatial phase information on the image. We Fourier transform the individual sub-apertures to the pupil plane. We stitch the sub-apertures together in the pupil plane, aligning them by the physical separation distance. Then the combined complex spatial field is inverse Fourier transformed to back to the image plane.

Though this is already a significant amount of processing, much more is required. When we initially get the combined field at the image plane, the image is a blurred low resolution image. For a single pupil with temporal phase errors, the image might be slightly fuzzy depending on how much the system was vibrating. As the number of sub-apertures and speckle realizations increase, the image becomes unrecognizable if no other processing is performed.

#### 3.4.1 Zernike Polynomials

We can remove the phase error from the system using weighted Zernike polynomials\(^\text{17}\). These polynomials form an orthonormal basis set over a unit circle. Each phase error has one or more Zernike polynomials that describe the same effect.
We can multiply our pupil field by the proper weights on the proper polynomial term, or terms, to eliminate that aberration.

### 3.4.2 Sharpening Metric

We have to find the proper weights to eliminate those terms. Sharpening algorithms are used with the image to do this. The phase correction is multiplied against the individual sub-aperture field in the pupil plane. All the sub-apertures are Fourier transformed back to the image plane. We find the speckle averaged intensity and use the sharpening metric function to look at the change. The sharpening metric that we used was first described by Muller and Buffington\(^\text{11}\). We use the metric that they termed \(S_5\)

\[
S_5 = \iiint I^n \, dx \, dy; \quad n \leq 2 \text{ and } n \neq 1. \tag{3.19}
\]

For values of \(n\) greater than 1, \(S_5\) is maximized, and for values of \(n\) less than 1, it is minimized. Values of \(n\) greater than 1 make bright areas brighter, and values of \(n\) less than 1 make dark areas darker\(^\text{18}\). Using this metric the algorithm keeps going back and forth from, the pupil plane complex field to the image plane intensity, until the value is optimized.

### 3.4.3 Intra Aperture Corrections

We eliminate the intra aperture phase errors first. These are the static phase errors that do not change between speckle realizations, and that do not depend on any other aperture. Defocus, astigmatism, coma and spherical aberration are the most common of these phase errors. For intra aperture errors, we make the same corrections to all the speckle realization, from one sub-aperture, in the pupil plane field. Each speckle realization is individually propagated to the image plane and the intensity is found. With the intensity, we can speckle average and use the sharpness metric. Once
we have the value of the sharpness metric, we go back to the pupil plane field and start over. This continues until the metric is optimized. We move on to the next sub-aperture and repeat until they are all corrected. Even though the same correction is made to each speckle realization, we use all of them so that the speckle does not impede the sharpening process. The time that this part of the post processing takes is heavily based off the number of pupils.

3.4.4 Inter Aperture Corrections

The inter aperture aberrations are the tip, tilt, and piston phase errors. We correct the static tip and tilt separately from the temporal tip and tilt. These phase errors are based on the other sub-apertures. We choose one sub-aperture that we do not adjust, and base all the other sub-apertures on its position. Because this is a static phase error, we use the same process as for the intra aperture errors.

The temporal tip and tilt are processed differently than the static tip and tilt. We focus on each speckle realization, optimizing each pupil. The exception is that we choose one speckle realization of one sub-aperture that we base all the others on. The temporal phase errors take much more time to process due to the loops being much larger.

The piston is removed in a similar fashion to the temporal tip and tilt. There are two big differences. Piston is a relative phase. It wraps around modulo two pi. This makes it somewhat easier to remove, because even in a brute force method of sharpening, it limits the number of possibilities.

Reconsider Eq. (3.18)
The piston disappears when the field of a single sub-aperture is turned into intensity. This is true for the other phase errors, but the piston does not affect the image quality with a single aperture. Tip and tilt move the images from the different sub-apertures and result in a blurred image with speckle averaging, just as intra aperture phase errors distort the image. The error correction for the intra aperture phase errors is applied to every speckle realization from a sub-aperture. The static tip and tilt are also corrected for the stack from a sub-aperture. The temporal tip and tilt require a speckle realization specific change, but it is still just moving the images. Piston requires a speckle realization specific change. Each speckle realization is completely independent from the others. One of the sub-apertures is used as a reference, while the others are changed. Local minimums multiply as the number of sub-apertures is increased. Piston sharpening takes the longest time and is especially computationally time intensive when the number of sub-apertures is increased.
CHAPTER 4

APERTURE SYNTHESIS SIMULATIONS

Numerical simulations are performed to quantify the savings in computational time by measuring the piston phase. These simulations are very valuable because it is possible, after comparing the different simulations, to estimate the time saved by not running the sharpness algorithms to obtain piston phase. The savings are based on the number of sub-apertures phased together.

4.1 Simulation Mechanics

4.1.1 Propagation Simulation

The simulations that we used were created by Miller, which we adjusted according to our own requirements. We will only include code that was written for this thesis; it is found in the appendices. We did not simulate spatial heterodyne to get the
field in the image plane, but used an alternative method of getting the complex field. We numerically propagate the entire field from the object plane to the pupil plane. We began with an intensity object, in this case a diffuse USAF-1951 resolution chart, shown in Fig. 4-2. To model the field at this diffuse target, we take the square root of the object intensity, making it the field, and apply a random phase (uniformly distributed from 0 to $2\pi$) to each pixel. The random phase produces speckle, as it simulates a rough object. The field was then propagated 100 meters, using a wavelength of 1550 nm, to the pupil plane using angular-spectrum propagation, as described by Schmidt$^{20}$. In the pupil plane, the complex-valued field was cropped to keep only the data inside of the sub-apertures. This is the field that we would have had after cropping, when using spatial heterodyne. We repeated this propagation with a different random phase applied to the object for each speckle realization, obtaining a statistically independent pupil plane each time. We chose 60 speckle realizations based on prior subjective evaluation of coherent image quality. Sixty realizations is the number used in Marron and Kendrick$^{13}$. We believe that 60 phase realizations are sufficient to provide imagery that closely resembles the incoherent image. The data that was created in the propagation simulation was usable in any of the variations on the processing code.

![Fig. 4-2: USAF-1951 resolution chart gray scale bitmap used in the propagation simulation.](image-url)
4.1.2 Pupil Arrays

In section 3.2, we mentioned that the fill factor of multi aperture arrays was lower than that of a monolithic aperture. Section 7.4 of Goodman\textsuperscript{21} discusses how imaging is an interferometric process. When we have places of no return in our receiving aperture, due to an array fill factor less then unity, that information is lost. We still get the image, but some spatial frequency information is lost in the mid-spatial frequency regions. This will reduce the signal to noise at those spatial frequencies, or could even eliminate information at some spatial frequencies. The eliminated spatial frequencies turn into an average intensity over those pixels. When the multi-aperture array is uneven in size, the dimension that is smaller will have a lower maximum spatial frequency.

We used several different pupil arrays. We used this code to process data from 2 in-line sub-apertures, 3 in-line sub-apertures, a hex 7 array and a hex 19 array. We ran many tests using 2 sub-apertures placed in a horizontal dimension to determine feasibility and workability. We expanded to 3 sub-apertures, also in the horizontal dimension, once everything seemed to be working correctly. Then we moved to Hexagonal arrays. All of our pupils were 22.9 mm in diameter.

4.1.3 Hex Arrays

We created the hex array to maximize fill factor. The hex array is the best approximation to a full circular aperture that is practical. We based the sub-aperture spacing and pixel count on the Sensors Unlimited SU320kts-1.7RT/RS170 which has a 53 mm square casing. The actual code is in Appendix A.1, though to use it you will have to add the variable dx. It is commented, but we will explain it here.

A hex array is based on a triangular lattice, see Fig. 4-3. A hex array has an odd number of elements across its longest dimensions. The code for the hex array requires only the number of sub-apertures desired in the arrays longest dimension. We chose the
longest row be in the horizontal dimension. You could choose it to be in any direction. The number of elements in the longest row is called num in the code. The number of apertures from corner to corner, cutting the hexagon in half, must be the same. The number of rows is equal to the number of elements in the center row to make this work. Each row gets shorter by one element as it moves away from the center. The parameter, Nfinal, gives the number of elements in the shortest rows. Finding Nfinal allows for a simple calculation of the total number of elements in the array.

![Diagram of hex array aperture centers.](image)

There is a simple transfer from triangular coordinates to Cartesian coordinates, using two basis vectors. The basis vectors, labeled in the code as \( a_1 \) and \( a_2 \), shown in Fig. 4-3, allow us to easily use for loops with integer values in triangular coordinates and then convert to irrational values in the Cartesian plane. cs is the physical distance, in meters, between the centers of two adjacent sub-apertures. It is the length of vectors \( a_1 \) and \( a_2 \)

\[
a_i = cs \left\{ 1, 0 \right\}, \tag{4.1} \\
a_2 = cs \left\{ \frac{1}{2}, \frac{\sqrt{3}}{2} \right\}. \tag{4.2}
\]
We used the pixel size in the pupil plane, $dx$, and the number of pixels to $cs$. To clarify, a coordinate in the triangular coordinate system $(m, n)$ is multiplied by these vectors.

The $m$ coordinate is unchanged, due to it being equal to the $x$ coordinate. The $n$ coordinate is a tilted version of the $y$ coordinate. Thus we have the vector relation:

$$m \cdot a_1 + n \cdot a_2.$$  

If $n = 1$ this causes a shift up by $\frac{\sqrt{3}}{2}$ and over by $\frac{1}{2}$.

Within the framework of the triangular lattice, creating the hex array is simple. The goal with this code is to fill a matrix with sub-aperture center positions. We want to be able to call a specific sub-aperture, $q$. We start a “for” loop in $i$. This goes from 1 to the num, or the number of rows. The first line in the loop tells what row it is on, $n$, the center row being zero. $n$ goes from $N_{final}$ to negative $N_{final}$. Then there is an “if else” statement.

This statement determines whether the program is working on the top or bottom half of the array. In the top half of the array, the number of elements to the left of the axis is a constant for all the rows. The same is true for the right side of the axis for the bottom half of the array. There are nested “for” loops, in the “if else” statement, that create the coordinates for the specific elements. The nested “for” loop is in the $m$ coordinate. Once we have the coordinate for an element, we multiply the first dimension, $m$, by $a_1$, and add the second dimension, $n$, multiplied by $a_2$. As $a_1$ and $a_2$ are vectors, this is matrix addition. This information is stored in the variable “center”, which has $q$ rows, one for each sub-aperture, and two columns, one for the $x$ coordinate and one for the $y$ coordinate. The final step is to place the center of each sub-aperture on the numerical grid. This is done by rounding the center position of each sub-aperture to the nearest numerical grid point. The rounding causes a slight variation in the sub-aperture
spacing, but at most this is half a grid point in any direction. At the end of each loop in m, we increment q to go to the next sub-aperture.

In the rest of the paper we will refer to the hex arrays, by the number of apertures in the entire array. A hex array with 3, 5 or 7 sub-apertures in the center row is a hex 7, 19, or 37.

4.2 Processing Simulations: General

We studied many versions of processing code to find the best way to deal with correcting the piston phase. The many different versions will be given different names to differentiate between the different variations. Pay attention to the names of the sub section to get the name of the code, or image, discussed in that section.

4.2.1 Adding Aberrations

The bulk of our simulations were completed using a variation of the code that Miller\textsuperscript{22} used for their real imaging data. Due to the code being written for noisy data, we had to add phase errors to different simulations discussed here with exception to the section on perfect images.

The simulated data had static and temporal phase errors added to each sub-aperture to better replicate experimental images. The static errors included in the simulations were defocus and fixed tip and tilt phases at each sub-aperture. The defocus was inherently included because the propagation distance to the pupil plane was not long enough for the Fourier transform to focus the image in the image plane using Fraunhofer diffraction. The tip and tilt phases were uniformly distributed in the range [-4, 4] waves and were the same for every speckle realization of a particular sub-aperture. We allowed 8 waves of phase because these are larger alignment errors. At a wavelength of 1550 nm, 8 waves is only 12.4 μm, which is easily reached in misaligning a whole sub-aperture.
The temporal aberrations were temporal tip and tilt, and piston phases. These were uniformly distributed in the range \([-0.5, 0.5]\) waves, and were different for every speckle realization of every sub-aperture. The temporal tip and tilt phases are up to 1 wave because they are due to small system vibrations. Piston phase wraps around over 1 wave for which this interval allows the full spectrum of change. For our sharpening metric, we used $n = 0.5$ (see Eq. (3.19)). Though this was probably the wrong choice for a primarily dark target, it seemed to work fine.

### 4.2.2 Removing Aberrations

We will explain the order of steps used in the code written Miller. The program loaded the data from the specified file created by the propagation simulation. It created the required Zernike polynomials that were needed to remove the phase errors. The Zernike polynomial creation code is found in Fricker. We then used the Zernike polynomials to add the phase errors using the code found in Appendix A.2. We will label the steps in the processing code so that we can reference the different steps in the different variations of the code.

The first processing step, step 1, sharpened each sub-aperture image by correcting the defocus error. Identical focus corrections were applied to each speckle realization of a particular sub-aperture. All sharpening was done using MATLAB’s optimization toolbox. The fminunc function was specifically used. In step 2, the program sharpened the image for static tip and tilt errors. In step 3, the temporal tip and tilt were corrected. In step 4, the piston phase error was removed.

The corrections that the program makes are different each time because the added aberrations were different each time. Following this process we could determine the statistical properties of the simulations. We visually inspected each resulting image to see that the simulation created the desired high resolution image.
4.3 Processing Simulations: Specific

4.3.1 Perfect Image Simulation

The data from the propagation simulation gave us the complex field, at the pupil plane. This is the same data we would have with the spatial heterodyne method to recover the complex field by Fourier transforming the real data from the image plane. One variation on the program took the data from the propagation simulation and added the same focus correction to every speckle realization. The altered complex pupil data was Fourier transformed to the image plane to get an unaberrated image that is diffraction limited. We will refer to these as our perfect images because they are as good as you can get based on the number of speckle realizations and the sub-aperture array geometry. These images were formed using the code with no sharpening algorithm.

4.3.2 General Processing Simulation

![Flowchart of general processing simulation.](image)

This is the basic program and all the others are variations of this one. This program loads data created from the propagation simulation. The needed Zernike polynomials are created and then used to add the phase aberrations, then the phase aberrations are removed through the 4 sharpening algorithm steps. Processing data is then saved. We saved the time that it took the entire program to run, the time it took the
program up to step 4, the number of iterations that the sharpening algorithm went through to remove piston, and the number of times the function to be optimized was called during piston sharpening.

This simulation program was the most useful one. We used this code to process data from 2 sub-apertures, 3 sub-apertures, a hex 7 and a hex 19.

4.3.3 Piston Phase Measured Processing Simulation A

![Flowchart of piston phase measured processing simulation A.](image)

In this simulation, we added all the phase errors as in the general processing simulation, but we immediately removed the piston before the program got to step 1. We removed the piston using the Zernike polynomial for piston and the saved piston information. This code is found in Appendix A.3. This was intended to simulate the piston being measured while taking the data. The piston phase error was still applied even though it was removed immediately after. This was done to simulate the processes of loading the piston error data. Also, we did not want to change the base time that the simulations ran. In a way we treated the phase error addition section of the code as part of loading the data. We still ran through all 4 sharpening steps. We expected this simulation to speed up all 4 steps. We ran this simulation for 2 and 3 sub-apertures.
4.3.4 Piston Phase Measured Processing Simulation B

This is the same simulation as simulation A, but the piston was removed after step 3. This simulation was most done to see if the sharpening of the other parts messed up the piston phase after it had been corrected. We were mainly looking at the total time and the number of iterations that the piston sharpening went through. We ran this simulation for 2 and 3 sub-apertures.

4.4 Simulation Results

We based the quality of our images off which sets of bars were clearly resolvable, and how much better the images appeared than the single 1 inch approximate diameter aperture. The processing simulation would not work with a single aperture, so the single aperture images were created using the perfect image simulation.

4.4.1 Two Sub-aperture Pupil Array Results

In Fig. 4-2 the USAF-1951 resolution chart is shown. We ran the propagation simulation with this target and two horizontally aligned sub-apertures. The output of the propagation simulation gave us 60 speckle realizations of speckled data for the pupil plane. When we put this in the perfect image simulation, we get the right side of Fig. 4-7. The left side of Fig. 4-7 is the same image, but for one aperture.
Fig. 4-7: Left: Perfect image from a single aperture. Right: Perfect image from a 2 aperture array.

It is clear that more bars are resolvable in the 2 aperture perfect image, than in the single aperture. The additional resolution only appears in the vertical bars. There are slight improvements in the horizontal bars, but this is due the increased resolution in the horizontal dimension due to vertical elements in the horizontal bars being sharper. Note that the increased resolution brings along a higher pixel count in the horizontal dimension. This is due to there being more elements in the inverse Fourier transform, from the pupil plane to the image plane, because the composite pupil is larger.

We will illustrate the images after different steps of the general processing simulation. On the left of figure Fig. 4-8, we have the speckle averaged, aberrated data from one of two sub-apertures. On the right, we have the same sub-aperture, but with the focus correction added. This is where the other intra aperture corrections would be made also, but we did not add them.
The bars are unresolvable in both images for all but the largest sets. There is a noticeable improvement after the intra aperture correction, but it is still less than can be resolved by a single aperture. This makes sense as these are single sub-aperture images. With the intra aperture corrections made, we look at the composite image of the 2 sub-apertures.

As mentioned before, the tip and tilt cause a shift in the image. This is clearly shown in the left of figure Fig. 4-9. The right side has the static tip and tilt corrected. We
can see a little of the increased resolution from the blur getting smaller in the horizontal dimension, most noticeably in the large square. There are no more sets of bars that can be resolved.

![Temporal aberration (tip/tilt) corrected composite image](image1)

![Temporal aberration (piston) corrected composite image](image2)

**Fig. 4-10:** Left: Temporal tip/tilt corrected image for 2 sub-apertures. Right: Piston corrected for 2 sub-apertures.

Fig. 4-10 has the temporal tip and tilt corrected image on the left and the piston corrected image on the right. The image with the temporal tip and tilt corrected appears very similar to the single aperture perfect image given in Fig. 4-8. There is a little bit more definition in some of the unresolved bars, but not enough to say that they are resolvable. The piston corrected image clearly has more resolvable bars. All the set of bars that are resolvable in the 2 sub-aperture perfect image, right of Fig. 4-8, are resolvable in the piston corrected image.

We will now switch to the piston phase measured processing simulation A. The images before the static tip and tilt are corrected will not be affected by the added piston measurement because piston is an inter aperture phase error. It only applies to the composite images. We will also leave out the static tip and tilt corrected image, as it still has heavy blurring from the temporal tip and tilt.
Fig. 4-11 has the temporal tip and tilt corrected image on the left and the piston phase corrected image on the right. Looking at the two images it is hard to see any difference between the two. This is good, because there shouldn’t be any difference. It shows that we are justified in removing the piston phase errors at the beginning of the simulation, and that step 4 is not required when we measure the piston.

We will leave out all images from the piston phase measured processing simulation B because they look exactly like the processing simulation’s images. The main motivation for doing this simulation was to see if correcting the piston phase here would help speed up the piston sharpening section if the measurements were not exact. We saw that this was not the case for this simulation, in fact it took it longer to sharpen, so we discontinued using this simulation. The piston phase measured processing simulation A showed a decrease in the piston sharpening section.

All simulation data was averaged over 25 independent sets of sixty speckle realizations, unless otherwise noted. We can see from Table 4-1 that the time it takes to optimize the piston error out of the images is a significant portion of the simulations. The piston phase measured processing simulation
A took a shorter amount of time in the piston sharpening portion, but the time difference was minimal. This means that that the sharpening algorithm does not converge faster depending on how close the value is to optimal when it starts. The standard deviation was significantly higher, though we do not understand why. It is not odd that the piston phase measured processing simulation B took longer to sharpen the piston phase because the time to remove piston explicitly was included into the sharpening.

Table 4-1: Simulation time data for the three processing simulations with 2 sub-apertures. Times are in seconds.

### 2 Apertures

<table>
<thead>
<tr>
<th>General Processing Simulation</th>
<th>Simulation Time</th>
<th>Simulation Time w/o Piston Sharpening</th>
<th>Piston Sharpening</th>
<th>Percent of Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>230.78</td>
<td>198.09</td>
<td>32.69</td>
<td>14.19%</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>10.94</td>
<td>10.68</td>
<td>1.51</td>
<td>0.86%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Piston Phase Measured Simulation A</th>
<th>Mean</th>
<th>Simulation Time</th>
<th>Simulation Time w/o Piston Sharpening</th>
<th>Piston Sharpening</th>
<th>Percent of Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>227.70</td>
<td>198.52</td>
<td>29.18</td>
<td>12.82%</td>
<td></td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>14.43</td>
<td>13.27</td>
<td>4.30</td>
<td>1.73%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Piston Phase Measured Simulation B</th>
<th>Mean</th>
<th>Simulation Time</th>
<th>Simulation Time w/o Piston Sharpening</th>
<th>Piston Sharpening</th>
<th>Percent of Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>234.60</td>
<td>200.63</td>
<td>33.97</td>
<td>14.52%</td>
<td></td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>12.44</td>
<td>12.52</td>
<td>1.23</td>
<td>0.96%</td>
<td></td>
</tr>
</tbody>
</table>

### 4.4.2 Three Sub-aperture Pupil Array Results

The 3 sub-aperture array is like the 2 sub-aperture array as the sub-apertures are aligned horizontally. We still only get increased resolution in the horizontal dimension. Fig. 4-12 is a perfect image of the 3 sub-aperture array. We can resolve more sets of bars in this than we could in the 2 sub-aperture perfect image.
Fig. 4-12: Perfect image of 3 sub-aperture array.

We will forego images of the single sub-apertures. Fig. 4-13 has the composite image of the processing simulation for a 3 sub-aperture array on the top left, the static tip and tilt corrected image on the top right, the temporal tip and tilt corrected image on the bottom left and the piston corrected image on the bottom right. The composite image is less recognizable than the two sub-aperture version. The tip and tilt images look similar to the 2 sub-aperture array's images. The piston corrected image, like the perfect image, has several more sets of bars resolvable.
Fig. 4-13: Images from the processing simulation with 3 sub-aperture array. Top Left: Composite image. Top Right: Static tip/tilt corrected image. Bottom Left: Temporal tip/tilt corrected image. Bottom Right: Piston corrected image.

The two images from the piston phase measured simulation A, Fig. 4-14, look identical. This verifies our hypothesis.
The fraction of time in the simulations that is devoted to piston sharpening is only somewhat higher, but it is going up. The mean simulation times were almost identical. This shows that, for the 3 sub-aperture simulations, taking a piston measurement shortens the time that it takes the piston sharpening section to run, but not enough to shorten the total processing time if the piston sharpening portion is still included.

Table 4-2: Data for the 3 sub-aperture array

<table>
<thead>
<tr>
<th>3 Sub-apertures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General Processing Simulation</strong></td>
</tr>
<tr>
<td>Simulation Time</td>
</tr>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>Standard Deviation</td>
</tr>
<tr>
<td><strong>Piston Phase Measured Simulation A</strong></td>
</tr>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>Standard Deviation</td>
</tr>
</tbody>
</table>
4.4.3 Hex 7 Sub-aperture Pupil Results

The perfect image, found in Fig. 4-15, is the same as the 3 sub-aperture array in the horizontal dimension, but the vertical dimension gains increased resolution as well. The resolution is not as good in the vertical dimension. The edge to center distance of the hex 7 array for the horizontal dimension is 36.87 mm and it is 33.42 mm in the vertical dimension. Using Eq. (3.5), this corresponds to a diffraction limited spot size of 2.56 μm and 2.83 μm in the horizontal and vertical dimension. The difference is not large, but it is significant. Though it is hard to tell in Fig. 4-15, the contrast on the vertically aligned 3 bars is better than the horizontally aligned set. When Fig. 4-15 is compared to Fig. 4-7, we can again see that the pixel numbers have increased. These are close enough in the two dimensions that it is hard to tell that the horizontal dimension is larger.

Fig. 4-15: Perfect image of hex 7 array.

The hex 7 array images show the same trends that the other arrays showed. A hex 7 array gives an even more unrecognizable composite image until aberrations are corrected. The static and temporal tip and tilt corrected images are about the same as the images for the other arrays. The piston corrected high resolution image is very
similar to the perfect image. The piston phase measured simulation A images are indistinguishable in detail.
Fig. 4-16: Top Left: Composite image of hex 7 array from the general processing simulation. Top Right: static tip/tilt corrected image. Mid Left: Temporal tip/tilt corrected image. Mid Right: Piston corrected. Bottom Left: Temporal tip/tilt corrected from piston phase measured simulation A. Bottom Right: Piston corrected.
The piston phase measured simulation A data for the hex 7 array only had 7 sets of speckle realizations. The numbers look consistent with the previous arrays. The percent of the simulation devoted to piston sharpening increased again.

Table 4-3: Time data for hex 7 array.

<table>
<thead>
<tr>
<th>Hex 7</th>
<th>General Processing Simulation</th>
<th>Piston Phase Measured Simulation A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Simulation Time</td>
<td>Simulation Time w/o Piston Sharpening</td>
</tr>
<tr>
<td>Mean</td>
<td>4858.83</td>
<td>3771.19</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>273.10</td>
<td>260.964</td>
</tr>
<tr>
<td>Mean</td>
<td>4713.10</td>
<td>3879.20</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>239.99</td>
<td>234.47</td>
</tr>
</tbody>
</table>

4.4.4 Hex 19 Sub-aperture Pupil Results

We have fewer pictures of the hex 19 due to memory constraints. Fig. 4-17 has the perfect image for a hex 19 array on the left and the piston corrected image from the general processing simulation on the right. The images have all the same resolvable features. Since the images for the other arrays coincide with each other, we think that those for the hex 19 would too.
Fig. 4-17: Left: Perfect image of hex 19 array. Right: Piston corrected image of hex 19 array from general processing simulation.

The data found in Table 4-4 shows that for greater numbers of sub-apertures, the greater the benefit found from measuring piston phase. The piston phase measured processing simulation A was only run once for this array, so those are the actual times rather than the averages. Since we were looking for the time savings from not including the piston sharpening in the general processing simulation, we did not run the piston phase measured processing simulation A more than once. In that single example the discrepancy between having a measured piston and running the sharpening metric for the piston takes much less time.

Table 4-4: Time data for hex 19 array.

<table>
<thead>
<tr>
<th>Hex 19</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>General Processing Simulation</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Simulation Time</td>
</tr>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>Standard Deviation</td>
</tr>
<tr>
<td>Piston Phase Measured Simulation A</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Time</td>
</tr>
</tbody>
</table>
CHAPTER 5

CONCLUSION

5.1 Simulation Conclusions

Using a code written for experimental images, we have shown that measuring piston will decrease processing time significantly. The percentage of savings grows as the number of sub-apertures grows. The processing times recorded in the Tables in Chapter 4 show that we should look at the percent saving from the general processing simulation and the simulation time without piston sharpening for an approximate time for the images to processes when a piston measurement is used. The processing time numbers are listed in Table 5-1.

Table 5-1: Savings and approximate processing time by sub-aperture array.

<table>
<thead>
<tr>
<th>Array</th>
<th>Percent Savings</th>
<th>Approximate Processing Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 sub-aperture</td>
<td>14.19%</td>
<td>198.51</td>
</tr>
<tr>
<td>3 Sub-aperture</td>
<td>17.44%</td>
<td>533.21</td>
</tr>
<tr>
<td>Hex 7</td>
<td>22.44%</td>
<td>3879.20</td>
</tr>
<tr>
<td>Hex 19</td>
<td>32.31%</td>
<td>46220.62</td>
</tr>
</tbody>
</table>

5.2 Future Work

5.2.1 Measuring Piston

A significant task based on this work is to actually create a system that will measure piston. Until this is accomplished, the work in this thesis is only theoretical.
The benefits of a piston measurement would have to base time saving versus cost of additional complexity in both equipment and personal hours to keep the system aligned. The temporal heterodyne theory in this thesis can be used to complete piston phase measurements. Additional theory will most likely be needed as problems arise.

5.2.2 Processing Optimization

Much of the benefit for measuring piston might be negated with a more efficient optimization algorithm. The time that it took to run the piston sharpening portion of the simulation was extreme, especially for the simulations where the piston was already measured. The percentage of time might be decreased significantly with a better algorithm. We did not spend time on figuring out how the fminunc function in MATLAB went about optimizing the sharpness metric. Understanding fminunc, or writing our own optimization algorithm might allow us to forget piston measuring completely.

The sharpness metric is also something that we need to study further. Fineup and Miller\textsuperscript{18} said that dark images should use an \( n \) greater than 1. We used an \( n \) of .5. Not only should we look into optimal values of \( n \), but also into other sharpening metrics.
REFERENCES


9 J. Rigden and E. Gordon, in Proc of the Institute of Radio Engineers (1962), pp. 2367-2368.


APPENDIX A

MATLAB CODE

A.1 Hex Array Building Code

Num=3; % Must be odd integer. Number of apertures in middle row.
Nfinal=(Num+1)/2; % Number of apertures in the shortest row
NTotal=Num+2*sum(Nfinal:(Num-1)); %Total number of apertures
cs=dx*140; %Center to center spacing for apertures. dx is pixel size in pupil plane.
a1=cs*[1 0]; %Transfer vector from triangular coords.
a2=cs*[.5 (3^.5)/2]; %Transfer vector from triangular coords.
center=zeros(NTotal,2); %Index for center
for i=1:Num
n=(Num-1)/2-(i-1); %Initial (i=1) Brings to top of hex
if i<(Num+1)/2 %top half of hex array
    for m=-Nfinal+1:i-1
        center(q,:)=a1*m+a2*n;
        center(q,1)=round(center(q,1)/dx);
        center(q,2)=round(center(q,2)/dx);
        q=q+1;
    end
else %bottom half of hex array
    for m=-Num+i:Nfinal-1
        center(q,:)=a1*m+a2*n;
        center(q,1)=round(center(q,1)/dx);
        center(q,2)=round(center(q,2)/dx);
        q=q+1;
    end
end
end

A.2 Phase Error Addition Code

%% Creates phases to be added
initial=randn(snaps,num_pup), %random number array (- halfwave to halfwave)
tip=8*rand(1,num_pup)-4; %random number vector (-4 waves to 4 waves
tilt=8*rand(1,num_pup)-4; %random number vector (-4 waves to 4 waves
for pupcount = 1:num_pup
    for snapcount = 1:snaps
        pupil_data(:,pupcount,snapcount) = pupil_data(:,pupcount,snapcount)*
            .exp(1i*2*pi*(w(:,1)*piston(snapcount,pupcount)+w(:,2)*
                (tip(pupcount)+(rand(1)-.5))+w(:,3)*(tilt(pupcount)+(rand(1)-.5))));
    end
end

A.3 Direct Piston Removal Code

%% Piston reversal
for pupcount = 1:num_pup
    for snapcount = 1:snaps
        ab_pupil_data(:,pupcount,snapcount) = ab_pupil_data(:,pupcount,snapcount)*
            exp(-1i*2*pi*w(:,1)*piston(snapcount,pupcount));
    end
end

A.4 Aperture Array MTF Code

lambda = 1.55e-6; % Wavelength [m]
D_physical = 22.9e-3; % Physical sub-aperture diameter [m]
pixel_pitch = 25e-6; % Camera pixel pitch [m]
f = 0.750; % Focal length of imaging lens [m]
A = 256; % Number of camera pixels (assumed A-by-A dimensions)
dx = lambda*f/(A*pixel_pitch); % Effective pixel size for propagation simulation [m]
cs = dx/140;
% to Use hex array, Comment out this section

% To use horizontal array, Comment out this section

% Imager number
Num=5; % Must be odd integer. Number of apertures in middle row.
Nfinal=(Num+1)/2; % Number of apertures in the shortest row
NTotal=Num+2*sum(Nfinal:(Num-1)); % Total number of apertures
a1 = cs*[1 0];  % Transfer vector from triangular coords.
a2 = cs*[.5 (3^.5)/2];  % Transfer vector from triangular coords.
center = zeros(NTotal, 2);
q = 1;  % Index for center
for i = 1:Num
    v = (Num-1)/2-(i-1);  % Initial (i=1) Brings to top of hex
    if i < (Num+1)/2  % top half of hex array
        for k = -Nfinal+1:i-1
            u = k;
            center(q,:) = a1*u + a2*v;
            center(q,1) = round(center(q,1)/dx);
            center(q,2) = round(center(q,2)/dx);
            q = q + 1;
        end
        else  % bottom half of hex array
            for k = -Num+i:Nfinal-1
                u = k;
                center(q,:) = a1*u + a2*v;
                center(q,1) = round(center(q,1)/dx);
                center(q,2) = round(center(q,2)/dx);
                q = q + 1;
            end
        end
    end
end

subap_x_cen = center(:,1)';
subap_y_cen = center(:,2)';

%%% Setup for vectors, variables and function. Aperture centers found previously.

R = round(D_physical/(2*dx));  % Radius of sub-aperture in pixels
circ = @(r) (abs(r)<=R);  % Function for creating circles
Rf = (round((Num-1)*cs/dx)+2*R)/2;
N = round((Num-1)*cs/dx)+2*R+10;  % array size in horizontal dimension in pixels. +10 to account for rounding slop
M = round((Num-1)*cs*(3^.5/2)/dx)+2*R+10;  % array size in vertical dimension in pixels.
n = -3*N:1:3*N-1;  % Creates integer vectors. The size is large to show the contained resolution.
m = -3*M:1:3*M-1;  % n and m are in pixel count with zero in the center.
ns = size(n);  % number of elements in n
ms = size(m);  % number of elements in m

[nx ny] = meshgrid(n,m);  % Cartesian grid
Aps = zeros(ns,ms);  % Aperture Array Preallocation
for ap = 1:length(subap_x_cen)
    r = hypot(nx-subap_x_cen(ap), ny-subap_y_cen(ap));
    Aps = Aps + circ(r);
end
%% Monolithic ap Setup
h=hypot(nx,ny);
Apf=(abs(h)<=Rf);

%% Plots Intensity psf
imagesc(Aps)
Ipsf=abs(fftshift(fft2(Aps))).^2; % intensity psf

%% monolithic ap psf
Ipsff=abs(fftshift(fft2(Apf))).^2;

%% Plots Full MTF
otf=fftshift(fft2(Ipsf)); % Non normalized OTF
m_otf=max(max(abs(otf))); % Maximum value of the magnitude of
the OTF,
mtf=abs(otf)/m_otf; % Normalized MTF
dfx=dx/(lambda*f); % Conversion from pixel size to spatial frequency
dfy=dx/(lambda*f); % Conversion from pixel size to spatial frequency
FY=m*dfy; % Creates vectors that corresponds to the spatial
frequencies from the pixel numbers
FX=n*dfx;
figure()
imagesc(FX,FY,mtf)
zoom(3)
xlabel('{f}_X cycles/m')
ylabel('{f}_Y cycles/m')
title('2D MTF of Hex 19 Sub-Aperture Array')
colormap(gray)

%% Monolithic ap Full MTF
otff=fftshift(fft2(Ipsff)); % Non Normalized OTF of Monolithic ap
m_otff=max(max(abs(otff))); % Normalization factor for Monolithic ap
MTF. Used for Combined plots also.
mtff=abs(otff)/m_otff; % MTF

%% Plots Horizontal MTF
mtfx=mtf(3*M+1,3*N+1:6*N);
% Creates horizontal slice of the full MTF.
fx=0:dfx:(length(mtfx)/3-1)*dfx;
% Creates frequency numbers corresponding to the mtfx
figure()
plot(fx,mtfx(1:length(mtfx)/3))
xlabel('{f}_X cycles/m')
ylabel('MTF')
title('Horizontal MTF of Hex 19 Sub-Aperture Array')

%% plots vertical MTF
mtfy=mtf(3*M+1:3*M,3*N);
% Creates vertical slice of the full MTF
fy=0:dfy:(length(mtfy)/3-1)*dfy;
% Creates frequency numbers corresponding to the mtfy
figure()
plot(fy,mtfy(1:length(mtfy)/3))
xlabel('{f}_Y cycles/m')
ylabel('MTF')
title('Vertical MTF of Hex 19 Sub-Aperture Array')

%% Monolithic ap MTF
mtffx=mtff(3*M+1,3*N+1:6*N);
% Creates 1d MTF slice of the Monolithic MTF
figure()
% plot(fx,mtffx(1:length(mtfx)/2))
% Plots the Horizontal and Verticle MTFs together with part of the full aperture MTF on top
figure()
mtfx=mtfx\*m\_otf/m\_otff; \%Renormalizes \( m_{tfx} \) to match frequency response with mono ap
mtfy=mtfy\*m\_otf/m\_otff; \%Renormalizes \( m_{tfy} \) to match frequency response with mono ap
plot(fx,mtfx\((1:length(mtfx)/3)\),fy,mtfy\((1:length(mtfy)/3)\),fx,mtffx\((1:length(mtffx)/3)\))
xlabel('f cyles/m')
ylabel('Renormalized MTF')
title('MTF comparison Normalized to a Monolithic Aperture')
APPENDIX B

PIXEL SIZE AND CORRESPONDING FREQUENCY CONVERSION BETWEEN PLANES

B.1 Pixel Size and Frequency Conversion From the Pupil Plane

A discrete Fourier transform, to go numerically from the \((x, y)\) plane or the pupil plane to the focal plane \((\alpha, \beta)\) goes to the \(f_x, f_y\) image plane first. The change in pixel pitch between planes for a physical system is

\[
df_x = \frac{1}{Nd_x},
\]

where \(d_x\) is the pixel pitch in the pupil plane and \(N\) is the number of pixels in the pupil plane. To mathematically go between the two planes, a scaled Fourier transform is used. A normal Fourier transform is used to change from the spatial plane to the spatial frequency plane. The scaling factor, which puts \(f_x, f_y\) into physical coordinates is

\[
f_x = \frac{\alpha}{\lambda f},
\]

where \(\alpha\) is the coordinate in
the image plane, $\lambda$ is the wavelength of light, and $f$ is the focal length of the lens. Taking the derivative, $df_x = \frac{d\alpha}{\lambda f}$. We now have two equations for $df_x$, so $df_x = \frac{d\alpha}{\lambda f} = \frac{1}{Nd_x}$. This makes $d\alpha = \frac{\lambda f}{Nd_x}$, which is the pixel size in the focal plane.

The same process is used to go back to the object plane. Because $f_x \sim f_{xx}$ with all the other variables are the same except $f$, $d\xi = \frac{\lambda R}{Nd_x}$. $R$ is the propagation distance.

**B.2 Pixel Size and Frequency Conversion From the Pupil Plane**

The process is different if the data is captured in the focal plane instead of the pupil plane. The pixel size in the $(f_{\alpha}, f_{\beta})$ coordinates is $df_\alpha = \frac{1}{Nd_\alpha}$. From the pupil plane method above, it is known that $dx = \frac{\lambda f}{Nd_\alpha}$ but $N$ is now the number of pixels in the detector array located in the focal plane.

To find the pixel pitch in the object plane is a similar process. When computing the discrete Fourier transform, the array to be transformed is frequently zero padded. This protects against aliasing. The size of the new array is the same as the size the transformed array was. A typical value used in zero padding is twice the array.
To go from the pupil plane to the object plane, \( df_x = \frac{1}{Gd_x} \), where \( G \) is the size of the new array. Similar to the previous, \( f_x = \frac{\alpha f}{\lambda R} \) and \( df_x = \frac{d\xi}{\lambda R} \), so \( d\xi = \frac{\lambda R}{Gd_x} \). It is of note that, \( \frac{NR}{gf} \) is the multiplication factor and that if no zero padding is used the multiplication factor is \( \frac{R}{f} \). This makes sense as the magnification of the imaging system should be \( \frac{R}{f} \).

This same process can be used going from the object plane to the focal plane with the difference being the definition \( d\alpha = \frac{Gf d\xi}{NR} \).

**B.3 Application of Pixel Size and Frequency Conversion for MTF Plots**

For the MTF plots, we started in the Image plane using a camera with \( A \) pixels and pixel size \( d\alpha \). We will derive the results using one dimension because the other dimension follows the same steps. From Section B.2, the conversion to the pupil plane is \( dx = \frac{\lambda f}{Ad\alpha} \). The physical size of the pupils that formed our array is known, so we used \( dx \) to find the pixel numbers that were required to build our aperture array. \( dx \) was used to get the center spacing, \( cs \), in physical coordinates where needed. The OTF can be described as the Fourier transform of the intensity psf. The intensity psf is the square of the scaled Fourier transform to propagate the pupil array to the image plane. If we were returning to the image plane, we would use \( d\alpha \), but we found that the plot of the intensity psf was heavily pixilated, so we zero padded the pupil array to 6 times its size.

This makes \( d\alpha' = \frac{\lambda f}{Gdx} \), where \( G \) is the size of the array. One of the definitions of \( df_{\alpha'} = \frac{1}{Gd\alpha'} = \frac{dx}{\lambda f} \). \( df_{\alpha'} \) is the pixel size, or more appropriately the frequency resolution.
in the Fourier plane connected to the image plane. With $\Delta f_{\alpha}'$, we can plot the Fourier transform with proper values at each frequency component.

We can check that the correct frequency values were used by finding the maximum frequency of the monolithic array, $f_{\alpha'} \max = \frac{D}{\lambda f'}$. This is a recognized formula, a form of which can be found just below Equation 6-32 in Goodman.\textsuperscript{14}