NON ITERATIVE MULTI-APERTURE AND MULTI-ILLUMINATOR
PHASING FOR HIGH RESOLUTION COHERENT IMAGING

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

In Partial Fulfillment of the Requirements for
The Degree
Doctor of Philosophy in Electro-Optics

by
Jeffrey Read Kraczek

UNIVERSITY OF DAYTON
Dayton, Ohio
August 2017

Approved for Public Release
17-MDA-9238 (13 June 17)
NON ITERATIVE MULTI-APERTURE AND MULTI-ILLUMINATOR PHASING FOR HIGH RESOLUTION COHERENT IMAGING

Name: Kraczek, Jeffrey Read

APPROVED BY:

__________________________________________  ____________________________________________
Paul McManamon, Ph. D.                          Edward A. Watson, Ph. D.
Advisory Committee Chairman                      Committee Member
Technical Director, LOCI                          Distinguished Researcher Sensor Technologies
Electro-Optics and Photonics                     University of Dayton Research Institute

__________________________________________  ____________________________________________
Joseph W. Haus, Ph. D.                           Jack E. McCrae, Jr., Ph.D.
Committee Member                                Committee Member
Professor                                       Research Assistant Professor
Electro-Optics and Photonics                     Department of Engineering Physics
                                                  Air Force Institute of Technology

__________________________________________  ____________________________________________
Robert J. Wilkens, Ph.D., P.E.                   Eddy M. Rojas, Ph.D., M.A., P.E.
Associate Dean for Research and Innovation      Dean, School of Engineering
Professor                                       School of Engineering
ABSTRACT

NON ITERATIVE MULTI-APERTURE AND MULTI-ILLUMINATOR PHASING FOR HIGH RESOLUTION COHERENT IMAGING

Name: Kraczek, Jeffrey R.  
University of Dayton

Advisor: Dr. Paul McManamon

The maximum resolution of a multiple-input multiple-output (MIMO) imaging system is determined by the size of the synthetic aperture. The synthetic aperture is determined by a coordinate shift using the relative positions of the illuminators and receive apertures. Previous methods have shown non-iterative phasing for multiple illuminators with a single receive aperture for intra-aperture synthesis. This work shows non-iterative phasing with both multiple illuminators and multiple receive apertures for inter-aperture synthesis. Simulated results show that piston, tip, and tilt can be calculated using inter-aperture phasing after intra-aperture phasing has been performed. Use of a fourth illuminator for increased resolution is shown. The modulation transfer function (MTF) is used to quantitatively judge increased resolution. The analytic MTF is compared to an “experiment” MTF using the slant edge method. Though experimental proof has been elusive, experimental process has been described as well as some possible reasons why the experiment was unsuccessful.
ACKNOWLEDGMENTS

I would like to thank Paul McManamon for being the mentor that I needed. Paul believed that I could succeed and took the risk of letting me chart most of the course on my own. Ed Watson has listen to my rambling questions with the insight that I needed to move on when I couldn't think of anything else to try. Jim Zimnicki for coupling the light into all those fibers and joining me for the journey.

Most of all, Michelle. She has suffered through the entire process with me. She was there through three advisors, and eight long years. She has been there pushing behind, believing that someday would come, even when I couldn't see it myself.
# TABLE OF CONTENTS

ABSTRACT ...................................................................................................................................... iii

ACKNOWLEDGMENTS .................................................................................................................. iv

LIST OF FIGURES ........................................................................................................................ vii

LIST OF TABLES ........................................................................................................................ xi

CHAPTER 1  INTRODUCTION AND OBJECTIVES ....................................................................... 1
  1.1 Introduction ............................................................................................................................ 1
  1.2 Objectives .............................................................................................................................. 2
  1.3 Thesis Overview..................................................................................................................... 3
    1.3.1 Resolution ....................................................................................................................... 3
    1.3.2 Aperture Efficiency .......................................................................................................... 4
    1.3.3 Spatial Heterodyne .......................................................................................................... 4
  1.4 Dissertation Outline................................................................................................................ 8

CHAPTER 2  CLOSED FORM APERTURE PHASING THEORY ................................ ................ 10
  2.1 Introduction .......................................................................................................................... 10
  2.2 Intra-Aperture Phasing ......................................................................................................... 11
  2.3 Inter-Aperture Phasing ......................................................................................................... 14
  2.4 Enhanced Resolution by Adding Further Illuminators .......................................................... 17

CHAPTER 3  CLOSED FORM PHASING SIMULATIONS ............................................................ 20
  3.1 Aperture Array Builder.......................................................................................................... 20
    3.1.1 Using the Aperture Array Builder .................................................................................. 20
    3.1.2 Aperture Array Builder Outputs: Synthetic Pupils ......................................................... 24
    3.1.3 Aperture Array Builder Outputs: Modulation Transfer Function .................................... 25
  3.2 Aperture Array Considerations ............................................................................................. 29
  3.3 Phasing Simulation .............................................................................................................. 32
  3.4 Simulation MTF .................................................................................................................... 36
    3.4.1 Slant Edge MTF: Theoretical ........................................................................................ 37
    3.4.2 Slant Edge MTF: smramt3 ............................................................................................ 37
    3.4.3 MTF Comparisons ......................................................................................................... 40

CHAPTER 4  EXPERIMENTAL DESIGN AND SETUP .............................................................. 42
  4.1 Basic Equipment .................................................................................................................. 42
    4.1.1 Laser ............................................................................................................................. 42
    4.1.2 Fiber .............................................................................................................................. 43
    4.1.3 Cameras ........................................................................................................................ 47
LIST OF FIGURES

Figure 1: Example spatial frequency distribution of image plane spatial heterodyne image. .......................7

Figure 2: a: Single receive aperture, light green, with three illuminator apertures, magenta. b: Field overlap due to a single receive aperture and three illuminator cluster. .................................12

Figure 3: Difference between the wavefronts of two overlapped pupil fields. The fringed section in the middle is where the two fields overlap. .................................................................12

Figure 4: Plot of percent field overlap between two of the three overlapped pupil fields and corrected waves of aberration .............................................................................................................14

Figure 5: a: Example of a three illuminator and five aperture array. The red circles are unfilled receive apertures, the light green circles are filled receive apertures and the magenta circles are illuminators. b: Example of the field overlaps from the three illuminator and five aperture array shown on top. The light blue areas are where only one field is present, the yellow are areas of two fields overlapping, and the red is where three fields overlap. ......................16

Figure 6: Wavefront difference between two pupils coming from two different apertures. a: A Piston, tip, and tilt error remains. b: Piston error remains...............................................................17

Figure 7: a: Example of a four illuminator and five aperture array. The red circles are unfilled receive apertures, the light green circles are filled receive apertures and the magenta circles are illuminators. b: Example of the field overlaps from the four illuminator and five aperture array shown on top................................................................................................................ 18

Figure 8: MTF plots for the arrays shown in Figure 5 and Figure 7. The x-axis is in cycles per mm. a: Horizontal MTF for array shown in Figure 5. b: Horizontal MTF for array shown in Figure 7. ......19

Figure 9: Vertical MTF for the array shown in Figure 7.................................................................19

Figure 10: Example of prompts to select sub-apertures in hex array builder. a: Initial prompt with no sub-apertures selected. b: Second prompt with one sub-aperture selected.........................21
Figure 11: Sub-aperture array illuminators and prompts. a: One illuminator placed. b: Four
illuminators placed ............................................................................................................................... 23

Figure 12: Filled sub-aperture array with illuminator placement prompts with a three illuminator
cluster. a: First illuminator of the three illuminator cluster placed. b: Three illuminator
cluster placed with additional illuminator ........................................................................................... 24

Figure 13: Synthetic pupils. a: Synthetic pupil overlap map. b: Actual size of synthetic pupil to
show clearly how large the pupil is. ..................................................................................................... 25

Figure 14: Example full MTF plots. a: Full MTF plot calculated from the synthetic pupil shown in
Figure 13b. b: MTF plot without symmetry across the x or y axis ..................................................... 27

Figure 15: Two dimensional MTF plots. a: MTF in the horizontal dimension. b: MTF in the
vertical dimension ............................................................................................................................... 27

Figure 16: Analytic MTF comparison between using a unit amplitude synthetic pupil and an
aperture overlap valued synthetic pupil. a: MTF comparison for three illuminator cluster. b:
MTF comparison for three illuminator cluster plus additional illuminator ........................................ 29

Figure 17: Aperture array for use in closed form high resolution multi-aperture imaging. The
green circles are receive imagining apertures. The red circles are where further apertures
could be inserted. The smaller purple circles are illuminator apertures ........................................... 30

Figure 18: Overlapping fields from aperture array with multiple illuminators. The light blue areas
are places where field from only one illuminator is present, the yellow areas contain field
from two illuminators and the red area have field contribution from three illuminators ................. 31

Figure 19: Target used in simulation. ........................................................................................................... 33

Figure 20: 200 speckle realization averaged images formed from non aberrated pupil fields. (a)
Single aperture with one illuminator. (b) Four illuminator and five aperture array .......................... 34

Figure 21: 200 speckle realization averaged images formed from aberrated pupil fields. (a) Single
aperture with one illuminator. (b) Four illuminator and five aperture array ............................................. 35

Figure 22: Aberration corrected image from four illuminator and five aperture array ....................... 36

Figure 23: Comparison between a portion of the non aberrated image and the aberration
corrected image. The images have been rotated. (a) Non aberrated. (b) Aberration
corrected. ............................................................................................................................................. 36
Figure 24: Example usage of region of interest function, getroi. a: Initial prompte window. b: zoomed in version of figure to illustrate theselection of the ROI.

Figure 25: Comparisons of analytic and experimental MTF curves. (a) MTFs for the three illuminator five receive aperture system. (b) MTFs for the four illuminator five receive aperture system.

Figure 26: Diagram of coupling light into fiber with endcap.

Figure 27: Notional diagram of compact range.

Figure 28: a: Optical table arrangement for compact range and other experiments associated with this project. b: Optical tables with some equipment arranged on them.

Figure 29: a: Head-on view of IMAGE 2 testbed. b: Mostly unfilled side view of IMAGE 2 testbed. c: Filled side view of IMAGE 2 testbed.

Figure 30: Sub-aperture design. a: Zemax design parameters. b: Ray diagram of light passing through sub-aperture.

Figure 31: a: 1 to 5 beam splitting holographic grating. b: Illumination pattern of beam splitter. c: Breadboard layout of splitter in free-space to fiber implementation.

Figure 32: a: BNS liquid crystal polarization grating. b: Diagram of beam deflection.

Figure 33: Liquid crystal polarization grating beam patterns. a: No voltage applied. b: Voltage applied for 5° of beam deflection.

Figure 34: Breadboard layout of LCPG switch for free-space to fiber implementation.

Figure 35: Winford Engineering BRKAVH68FV1-R-FT breakout board.

Figure 36: NI PCIe 6321 pinout diagram.

Figure 37: Zaber rotational stepper motor with target mounted on the central axis of rotation.

Figure 38: Layout of spatial heterodyne experiment with transparent target.

Figure 39: a: Direct detection spatial heterodyne image. b: Processed spatial heterodyne image. Noise reduction is due to the spatial fringe noise being located outside of the crop area. Image fade is due to improper mode matching.
Figure 40: a: Pupil plane image of transparent target image shown in Figure 39a. b: Top left quadrant of pupil plane image. This section is used to create the spatial heterodyne image.

Figure 41: Single aperture spatial heterodyne imaging experiment using the compact range. a: Setup diagram. b: Picture of aperture assembly, illuminator, and LO.

Figure 42: Left: Direct detection image of target with LO also on the FPA. Right: Image of LO only.

Figure 43: Pupil plane image of Figure 42a. The green circle represents what would have been cropped if the pupil had not aliased across the frame. The red circle denotes the autocorrelation terms that are desired to be outside the pupil.

Figure 44: a: Block target. b: Spatial heterodyne image of target processed from the top left quadrant pupil in Figure 43.

Figure 45: Field shift due to illuminator shift.

Figure 46: a: Pupil plane image of fiber tip located in the entrance pupil. b: Pupil plane image of fiber tip located in the entrance pupil after image plane error phase removal.

Figure 47: Table vibration data from accelerometers placed on and around optical tables. This is low quality but it clearly shows strong vibrational peaks. a: Vibration data under ambient conditions. b: Vibration data with excitation from accelerometer hammer.

Figure 48: Illumination pattern from bear fiber tip.

Figure 49: Examples of pupils from different frames. The two circled areas show correlated speckles though the pupils are not correlated enough for registration.

Figure 50: Three scans through a Fabry-Perot interferometer.

Figure 51: Diagram of simple two illuminator spatial heterodyne experiment. The imaging part is a 2f to 2f system for 1 to 1 imaging.

Figure 52: Wavefront difference of two fields from the same aperture and illuminator taken one after the other.

Figure 53: Wavefront difference of two fields from the same aperture but different illuminator taken sequentially.

Figure 54: Chrome on glass target mounted in a two inch lens mount with a steel plate attached to the back.
LIST OF TABLES

Table 1: Efficiencies of power split into each beam for the 1 to 5 splitter. The farthest left (LLL) and farthest right (RRR) are > two orders of magnitude down from the desired 5 beams. ........................53

Table 2: Steering efficiencies for BNS liquid crystal polarization grating. ........................................................................55

Table 3: Zaber rotational stepper motor T-RSW60C-KT04U specifications .................................................................60
CHAPTER 1

INTRODUCTION AND OBJECTIVES

1.1 Introduction

Weight and space constraints are frequently placed on active EO systems, including ladar systems, but high resolution imaging requires large optics due to the physical limitations of imaging. Large optics are frequently impractical to use because an imaging system that uses them has to be deeper than a smaller lens to maintain the same f#. Large optics are also heavy and expensive. In order to achieve high resolution, methods have been developed to synthesize large apertures. This can be accomplished with multiple smaller apertures, or by moving one or more apertures, as is done with microwave synthetic aperture radar (SAR) systems. If a large sample of the pupil plane field can be captured, that sample can be Fourier transformed to create a high resolution image. SAR, does this to create higher resolution imagery than allowed with real beam radar. Because of the high frequency of lasers, we cannot directly measure the optical electro-magnetic field. Optical frequencies are orders of magnitude higher than frequencies which could be directly measured by any existing detectors. Current detectors can be used in heterodyne systems which use a local oscillator to beat against the return signal to measure the return optical field, including phase. This works because the detector measures the beat between the return signal and the local
oscillator rather than measuring the carrier frequency. The beat between the LO and the return signal creates spatial fringes in spatial heterodyne (a form of digital holography).

1.2 Objectives

High spatial resolution is of interest in many applications. A large aperture is required for high spatial resolution, but large optical elements are expensive and heavy. It has been shown that spatial heterodyne can be used to synthesize a large aperture using multi-aperture arrays. A problem with using multiple receive apertures is that the apertures are not perfectly aligned with each other requiring aperture phasing. One way to accomplish receive aperture phasing is by using iterative techniques which optimize a sharpness metric. This is computationally expensive and can take a considerable amount of time to accomplish, especially as the size of the aperture array gets larger, or the range gets longer. Three closely spaced illuminators, spaced closer than the aperture diameter of a single receive sub-aperture, can be used to solve in closed form for the intra aperture aberrations. Intra aperture phasing contributes to the amount of time it takes iterative phasing methods to complete aperture phasing. Closed form phasing should be much faster, especially for larger arrays. We believe that this approach may also be used for inter aperture phasing, removing the need for iterative phasing completely. Closed form aperture phasing should be significantly faster to solve than any iterative technique, especially in those cases where the iterative technique uses returns from the target, requiring the time for light to travel to and from the target to complete each iteration. It has also been shown that higher resolution can be achieved having multiple illuminators that are spaced farther apart. We would like to combine these techniques; obtaining higher resolution using a widely spaced illuminator with closed form phasing using 3 closely spaced illuminators in conjunction with a multi-aperture receive array. This aperture configuration can be called MIMO, multiple-input
and multiple-output, imaging. MIMO techniques have been significantly explored in the microwave region, but have not been widely explored in the optical region.

1.3 Thesis Overview

1.3.1 Resolution

There are several terms that need to be defined at this point. Image or spatial resolution has several definitions. When resolution is used, it will be referring to the spatial frequency response of a particular aperture. Generally this will be defined using the Modulation Transfer Function (MTF). The MTF is the modulus of the Optical Transfer Function (OTF) and it gives the expected spatial frequency response to an impulse, or point object, for a given system. The OTF for a given system as given by Goodman[1] is,

\[ H(f_x,f_y) = \frac{F\{h^2\}}{\int \int |h(u,v)|^2 dudv}, \]  

(1.1)

where \( h \) is the impulse response of the system. The MTF is the modulus of equation (1.1). There is an issue of how to compare MTF plots. Because the MTF is normalized to the area of the receive aperture, the MTF plot can be deceiving. The normalization makes it appear that a system with lower spatial frequency response seem like it has the same, or higher, frequency response then a system that actually has a higher response. This is due to the MTF being a measure of contrast at a specific frequency. This confusion is especially evident when using multiple apertures and illuminators because the MTF uses the area of the synthetic aperture for normalization. Using the MTF allows for a direct comparison of the frequency response cutoff, but only a general comparison of the relative shapes of the MTF plots. The system will respond better to certain
frequencies, but the contrast at those frequencies will be lower due to more light entering the system.

1.3.2 Aperture Efficiency

When using multiple apertures, there will be space between the apertures where no light is being collected. The ratio between the multi-aperture systems active area and the area of a monolithic aperture, with the same diameter as the receive aperture array diameter, is defined as aperture efficiency. An aperture efficiency of less than 100% will mean that certain spatial frequencies are not measured, or that the contrast on those spatial frequencies is diminished. When using multiple illuminators, the received fields must be stitched together in the frequency plane. This will sometimes cause fields from multiple apertures to overlap. An amplitude correction is needed so that field content is not counted multiple times[2]. This is called aperture weighting. The amount of aperture weighting can influence the high resolution, synthesized image. Using multiple illuminators, we could have an aperture efficiency larger than 1. The efficiency can be greater than one because it is based on the physical size of the hole made for the imaging optics. With multiple illuminators, the synthetic aperture can be larger than the size of the physical aperture.

1.3.3 Spatial Heterodyne

Standard imaging of light only measures intensity because the frequency of the light is too high for the electronics to measure the full field, which includes phase. Optical detectors are referred to as square law detectors, as the image received is the square of the magnitude of the field impingent on the sensor. Holographic techniques can be used to recover the amplitude and phase of the field incident on the detector. Holography was first discovered by Gabor in 1948[3]. As lasers were not invented yet and holography needs a certain degree of coherence of the light, it was not used much
in optics until the advent of the laser. In the 1960s, Leith and Upatneiks continued the development of holography[4][5][6]. One method that they developed was the foundation of what we call spatial heterodyne.

Digital holography was first employed by Goodman and Lawrence in 1967[7]. Capturing holograms digitally allowed for processing the received interferogram and outputting the hologram digitally without having to manually reconstruct the hologram with another light source. Since 1967, digital holography has had many applications. Further information can be found in the literature.

In a holographic system, the image field is mixed on a recording media with a reference beam, also known as a local oscillator, which creates an interference pattern. For digital holography, the LO and return image fields are separated in angle and are mixed on a detector array. The fields are separated in angle to produce fringes that capture the spatial variation in phase across the detector array. The resulting interference pattern is processed in the computer to retrieve amplitude and phase of the return field. Spatial heterodyne can be used in either image plane or pupil plane modes. Image plane mode uses imaging optics while pupil plane mode captures the fields without the use of imaging optics, though optics maybe used to demagnify the pupil plane field to match the size of the sampling array. For this project we will use image plane spatial heterodyne because a direct detection image of the object is formed without processing. The following derivation would be somewhat different if done in pupil plane mode.

Available detector arrays, are square law detectors, so the image received in spatial heterodyne is of the form

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

(1.2)
where $U_{\text{LO}}$ and $U_s$ are the LO and signal fields respectively. The reason this method is called spatial heterodyne is because we calculate phase variation in space, using a titled LO, $U_{\text{LO}}$, rather than in time. This phase tilt is of the form,

$$\frac{\beta}{e^{i\phi_{\text{LO}}}}$$

(1.3)

where $k$ is the wave number, $z$ is the distance traveled, and $\alpha_\text{o}$ and $\beta_\text{o}$ are the offset positions of the LO from the optic axis. The tilt causes interference fringes on the detector, which allow us to measure phase variations across the array. The beat frequency is spatial instead of temporal. We also want the field curvature to be matched to that of $U_s$.

For the LO and signal beams to mix efficiently, the two beams must be mode matched. This means that the quadratic phase curvature fronts of the two beams need to be equal. The field in the paraxial regime for the signal beam, assuming an ideal system, is of the form

$$U_s(u,v) = e^{i\Phi_s} \left( \frac{k_s}{\lambda z_s} \right)^2 e^{i\phi_s(u^2 + v^2)} U_s(u,v) \otimes \mathcal{F}[P(x,y)]$$

(1.4)

and the field for a point source LO beam is of the form

$$U_{\text{LO}}(u,v) = e^{i\Phi_{\text{LO}}} \left( \frac{k_{\text{LO}}}{f_{\text{LO}}z_{\text{LO}}} \right)^2 e^{i\phi_{\text{LO}}(u^2 + v^2)}.$$ 

(1.5)

For mode matching, we only care about the quadratic phase terms from the two equations and we want them to be equal,

$$e^{i\frac{k_{\text{LO}}(u^2 + v^2)}{z_{\text{LO}}^2}} = e^{i\phi_{\text{s}}(u^2 + v^2)},$$

(1.6)

where $k_{\text{LO}}$ and $k_s$ are the wavenumbers for each beam, $z_{\text{LO}}$ is the propagation distance from a point source, and $f_s$ is the focal length of the imaging lens. Simplifying the previous equation gives,
\[
\frac{1}{z_{LO}^2 \lambda_{LO}^2} = \frac{1}{f_s \lambda_s} \quad \text{or} \quad z_{LO} = \frac{f_s \lambda_s}{\lambda_{LO}} \tag{1.7}
\]

and if we assume that the wavelengths of the LO and signal beams are the same,

\[
z_{LO} = f_s. \tag{1.8}
\]

Returning to equation (1.2), with \( U_{lo} \) having the proper phase terms, we take the Fourier transform of the intensity image. This retrieves the amplitude and phase in the spatial frequency domain. Due to intensity being captured in the image plane, the Fourier transform gives the pupil plane field. The tilt from \( U_{lo} \) separates the \( \mathcal{F} \{ U_{lo}(u,v)U_{s}(u,v)^* \} \) and \( \mathcal{F} \{ U_{lo}(u,v)^* U_{s}(u,v) \} \) terms from the other two, and, since one is the complex conjugate of the other, the two terms are shifted in opposite directions. A tilt in the spatial domain becomes a shift in the frequency domain. The energy from the \( \mathcal{F} \{ |U_{lo}(u,v)|^2 \} \) and \( \mathcal{F} \{ |U_{s}(u,v)|^2 \} \) terms remain near the center. An example of the pupil plane field is shown in Figure 1. Given the proper value of the tilt terms, the image can be cropped to retain only \( \mathcal{F} \{ U_{lo}(u,v)U_{s}(u,v)^* \} \) or \( \mathcal{F} \{ U_{lo}(u,v)^* U_{s}(u,v) \} \) term.

![Figure 1: Example spatial frequency distribution of image plane spatial heterodyne image.](image-url)
The proper value of the tilt term can be determined based on camera pixel size $(dx)$, receiver diameter $(D)$, wavelength $(\lambda)$, offset of the LO from the receive aperture axis $(d)$, and the distance between the exit pupil and the camera $(z)\left[8\right]$. The size of the pupil is $\frac{D}{\lambda z}$. The offset of the pupil is $\frac{d}{\lambda z}$ along the same radial angle that the LO is from the receiver axis. The pupil must fit in $\frac{1}{2dx}$ or it will alias across the frame in the dimension that it exceeded the bounds.

We will choose to keep $\mathcal{F}\{U_o(u,v)U_s(u,v)\}$ and inverse Fourier transform it back to the spatial domain leaving us with $U_o(u,v)U_s(u,v)$. The crop removes the shift from the frequency domain meaning that the phase tilt is gone in the spatial domain. With proper mode matching, the quadratic wavefront phase curvatures from the two fields cancel. This is because the LO field is complex conjugated, and thus the sign on the quadratic phase curvature is reversed. If $U_o$ has uniform intensity, it becomes an amplifier. This is one of the benefits of using a heterodyne system.

1.4 Dissertation Outline

CHAPTER 2 will discuss closed form phasing for multi-aperture, multi-illuminator coherent imaging. Intra and inter-aperture phasing will be described, as well as increased resolution by adding illuminators. CHAPTER 3 will discuss simulations used to understand closed form phasing. It will cover the aperture array builder simulation, used for understanding the frequency response of different aperture arrays. Next, the closed form phasing simulations for a linear sub-aperture array, with the three illuminator cluster used for intra-aperture phasing and the fourth extended illuminator for extended resolution, will be presented. After that the slant edge MTF is introduced for comparing analytic MTF to experimental MTF. CHAPTER 4 discusses basic experiments and equipment used in this project. Much of this section is given over to why this specific
equipment was used and how to use it. CHAPTER 5 explains the application of the closed form phasing and what went wrong with the experiments. It finishes with what is believed to have been the reason the experiments did not turn out the expected results. CHAPTER 6 discusses the end results of the simulations and experiments, as well as giving some ideas where to go to move forward.
CHAPTER 2

CLOSED FORM APERTURE PHASING THEORY

2.1 Introduction

Large apertures are needed for high resolution imaging. For a monolithic aperture, the mass and volume increases approximately to the 2.7 power of the diameter of the primary optic. This results from the optical system getting larger in depth as well as getting larger in the other two dimensions. It does not increase in depth quite as much as in the other two dimensions. Attempts to increase optical resolution have turned to using multiple smaller apertures so the weight and volume would not increase as fast. For a constant f# optical system smaller diameter apertures are not as deep. One of the earlier attempts at combining the light from multiple apertures was performed by Meinel in 1970[9]. His design was to use six optical telescopes, combining the light by phasing all the telescopes to focus at the same point. Many others have since tried and succeeded at using mechanical means of phasing multiple telescopes to achieve higher resolution. More recently, Marron and Kendrick were able to use spatial heterodyne detection and digital image processing techniques to combine the fields from multiple apertures to achieve higher resolution images[10]. Image sharpness functions, pioneered by Muller and Buffington[11], were needed to achieve aperture synthesis. These methods are time consuming and often need to be adjusted dependent on the brightness of the target[12]. Rabb et. al. determined that, by using multiple illuminators,
aperture aberrations could be solved for and removed using a non-iterative method[13]. Using closely spaced illuminators, they were able to create a system of equations from the overlapping received fields. Instead of blindly searching, as done in the iterative approaches using the sharpness metrics, they were able to solve these systems of equations to correct for the sub-aperture aberrations. Using digital image processing techniques, it was shown that image resolution can also be increased by using multiple spatially separated illuminators[2]. This is not surprising since bi-static radar has been used in this capacity for years. Reference [2] also shows that the fields from different illuminators can be aligned in the pupil plane using a registration algorithm based off of speckle.

2.2 Intra-Aperture Phasing

Much of the theory for aperture phasing follows along with the paper published for this work.[14] Figure 2a shows an example of a multi-illuminator, single receive aperture system. The receive aperture is light green and the three illuminator cluster is magenta. The cluster is set to the vertices of an equilateral triangle with side lengths equal to the radius of the receive apertures. Figure 2b shows the measured received fields, referenced to a common coordinate system. The overlap regions of the received fields in the pupil plane due to the illuminator positions are shown in yellow and red. The outer edge of this image is the shape of the synthetic pupil. The synthetic pupil is where all the different fields are referenced to a common coordinate system before being used to create a high resolution image. As described in reference [2], a shift in the illuminator location has an equal result on the movement of the placement of the received field in the synthetic pupil. This is because a coordinate change must occur to compensate for different illuminator locations before the pupil plane fields can be coherently added. The synthetic pupil is built from the individual pupil fields being registered into the correct location in post processing.
Figure 2: a: Single receive aperture, light green, with three illuminator apertures, magenta. b: Field overlap due to a single receive aperture and three illuminator cluster.

Reference [15] describes how a linear system of equation is created and used to remove intra aperture phase errors. Figure 3 shows the difference of the wavefront of two overlapped pupil plane fields. The fringed section is where the two fields overlap. Using this overlap region is an extention of sheard coherent interferometric photography[16].

Figure 3: Difference between the wavefronts of two overlapped pupil fields. The fringed section in the middle is where the two fields overlap.

Since the fields overlap, taking the difference leaves only the wavefront error between the overlap sections. The phase wraps in the wavefront difference would be problematic except that each section, area between phase wraps, is segmented using
an edge detection algorithm. Each section has a constant unknown phase term added
to it in the system of equations. This effectively removes a complicated phase
unwrapping. Zernike polynomials are also used to create the system of equations.
Since Zernike polynomials are only orthogonal in certain geometries, the system needs
to be over defined. That means that there has to be overlap from the same receive
aperture and three different illuminators. Without at least two overlap sections there
would be multiple solutions to the system of equations. The wavefront difference
between two overlapping pupil plane fields is defined as

$$
\Delta W(m)\left(x_i^{(m)},y_i^{(m)}\right) = \alpha^{(m)} + \sum_k a_k \Delta Z_k^{(m)}\left(x_i^{(m)},y_i^{(m)}\right),
$$

(2.1)

where \(m\) is which fringe section is being used, \(x\) and \(y\) are the Cartesian coordinates, \(i\) is the pixel index, \(\alpha^{(m)}\) is the unknown phase constant added to each section, \(k\) is the index of the Zernike polynomial being solved for, \(a_k\) is the weight on the specific Zernike, and \(\Delta Z_k^{(m)}\) is the difference between shifted Zernike polynomials, or

$$
\Delta Z_k = Z_k(x+x_1,y+y_1) - Z_k(x+x_2,y+y_2)[15].
$$

Using all the pixels within the overlap section, except along the singularities formed by the phase wraps, a well-defined system is formed:
Once \( a^{(m)} \) and \( a_k \) are found using an inverse operation, the \( a_i \) s can be used with their assorted Zernike polynomials to remove the higher order phase aberrations. Though the matrix may not be truly invertable, Matlab uses a psudo-inverse operation that allows it to estimate the solutions for \( a^{(m)} \) and \( a_k \).

When using a closely spaced group of illuminators it is important that the space between the illuminators does not grow too large to prevent sufficient overlap of the received fields. Figure 4 is a plot of area of field overlap and waves of corrected phase aberration. The same phase was added to each field due to aperture aberrations using the Zernike polynomials for defocus, oblique astigmatism and vertical astigmatism. As the percent overlap decreases to around 12% overlap, the reliability of the phase estimates diminishes. Random phase noise was not added to these simulations so the needed level of overlap for an experimental system may be higher than this plot shows. Also, when the number of different phase aberrations increases, the process becomes weaker, requiring more overlap to get a good estimate of phase.

![Aberration Correction with Diminishing Overlap](image)

**Figure 4**: Plot of percent field overlap between two of the three overlapped pupil fields and corrected waves of aberration.

### 2.3 Inter-Aperture Phasing

The first and second order Zernike polynomials, piston, tip, and tilt, cannot be solved for in the method describe above because they do not show up in the wavefront
difference using the same receive aperture. This is because piston is a flat phase and there is no difference in that term across the aperture. Tip and tilt terms have a constant slopes across the aperture and therefore also cancel out when taking the difference of the two wavefronts from the same aperture.

A different method must be developed to correct for the inter-aperture tip-tilt-piston aberrations. Figure 5a shows a three illuminator and five aperture array. This is an example of a MIMO system that can be used for inter aperture closed form phasing. Figure 5b shows the overlap pattern for the system shown in Figure 5a. Each individual receive aperture provides coverage in the same pattern as shown in Figure 2. There is significantly less overlap between each aperture, but it still exists. Defining the wavefront difference differently here, as the wavefront error across the overlapped apertures is not the same. The wavefront difference is now defined as

$$\Delta W(x, y) = W_{e,i_1,j_1}(x, y) - W_{e,i_2,j_2}(x, y) \quad i_1 \neq i_2, j_1 \neq j_2,$$

(2.3)

where $i_1$ and $i_2$ are different illuminators and $j_1$ and $j_2$ are different receive apertures.
Assuming at this point that the higher order phase errors were removed using intra aperture phase correction. All that is left is the piston, tip, and tilt. Figure 6a shows an example of the wavefront difference between two different apertures. Using the same type of segmenting process as the intra aperture phase correction, the gradient in both dimensions is found across one of the segments. This can then be removed from the fields. At this point, all that is left is piston. Figure 6b shows an example of the wavefront difference after all other phase errors are removed. The value of the dark section is the piston phase. Both the piston and the tip and tilt corrections can be calculated with very few pixels. In a noisy system more pixels will need to be used, but as phase noise is generally zero mean Gaussian, using a moderate number of pixels should gain a fair estimate. Further research with phase noise added will be needed to
determine how this method works in noisy conditions. When correcting for these inter aperture phase errors, one aperture is set as a reference to which all others are corrected. The wavefront difference just shows only the relative difference between the phase errors on the two apertures.

Figure 6: Wavefront difference between two pupils coming from two different apertures. a: A Piston, tip, and tilt error remains. b: Piston error remains.

2.4 Enhanced Resolution by Adding Further Illuminators

Adding a fourth illuminator can increase the size of the synthetic pupil significantly. Figure 7a shows a four illuminator 5 receive aperture system. Figure 7b shows the shape of the synthetic pupil. The set of five circles on the left hand side comes from the fourth, widely separated, illuminator. Those fields do not need to be overlapped since they come from the same apertures that already had the intra and inter aperture phase aberrations removed. The one exception is that for a system where the exact distance between the other illuminators is unknown there is to be overlap between the farthest left field from the fourth illuminator and the farthest right field from the illuminator cluster. Reference [2] describes a way of using sub-pixel registration on the speckle in the pupil plane. For this simulation we assumed exact knowledge of all illuminators and receive. The approximate overlap needed to run the registration is therefore not determined.
Figure 7: a: Example of a four illuminator and five aperture array. The red circles are unfilled receive apertures, the light green circles are filled receive apertures and the magenta circles are illuminators. b: Example of the field overlaps from the four illuminator and five aperture array shown on top.

Figure 8 shows the theoretical modulation transfer function (MTF) in the horizontal dimension for the array in Figure 5 in part a and the array in Figure 7 in part b. Notice that the cutoff frequency for the three illuminator and five aperture array is about half the cutoff of the array with four illuminators and five apertures. This is because the synthetic pupil size is half as large in that dimension. The MTF in the vertical dimension shows a much lower cutoff, as seen in Figure 9. This is because the synthetic pupil is not much bigger in the vertical dimension than a single receive aperture.
Figure 8: MTF plots for the arrays shown in Figure 5 and Figure 7. The x-axis is in cycles per mm. a: Horizontal MTF for array shown in Figure 5. b: Horizontal MTF for array shown in Figure 7.

Figure 9: Vertical MTF for the array shown in Figure 7.
3.1 Aperture Array Builder

Many of the figures in this work were created using an aperture array simulator. This simulator was built to assess the analytic MTF of MIMO arrays with receive aperture set to a hex array. The process of using the array builder will be explained here with the standard version of the code being found in appendix A.1.

3.1.1 Using the Aperture Array Builder

The first block of code after the header contains a section labeled setup. In this section the parameters that are used to build a hex 19 aperture array are laid out. \( dx \) defines simulation pixel size, \( D_{\text{phys}} \) is the size of the sub-apertures, \( ds \) is dead space around each sub-aperture for structural considerations, and \( \text{ill}_s \) is structural considerations for illuminators. All of these parameters are defined in meters. The remaining variables in the setup section are not user defined except for \( n_{\text{ill}_\text{max}} \) which will be talked about later. The next section of code is used to find grid point locations for a hex array. The code is written based on a hex 19 array. The code is easily modified to go larger, though current computers will have memory issues dependent on the system parameters. The variable, \( \text{Num} \), is used to specify the number of sub-apertures across the center of the array. \( \text{Num} \) is the first variable defined in the Hex Center Finder section. \( \text{Num} \) must be an odd integer for this program to function properly.
When the program is started, it will take the user to the command window with the prompt, “How many sub-apertures will you choose? Enter a number 1 - 19 then press the enter key.” The 19 apertures is based on a hex 19 array. Once the user enters the desire number of receive apertures, a figure with a hex 19 array will come up with the prompt “Please click on desired sub-apertures”, as shown in Figure 10a.

![Figure 10: Example of prompts to select sub-apertures in hex array builder. a: Initial prompt with no sub-apertures selected. b: Second prompt with one sub-aperture selected.](image)

When the mouse is directed over the figure, a set of cross-hairs will appear. Use the cross-hairs to click within the red circles to choose the previously stated number of apertures. Once a circle is clicked on, it will turn green and the prompt will change to “Please choose next sub-aperture”, as shown is Figure 10b. If the position clicked on is not valid sub-aperture location, the prompt at the top of the figure will change to “Failed!!! Please click on desired sub-aperture” or, if the same sub-aperture is selected twice, “Failed!!! Please click on a different sub-aperture”. This will continue until the previously specified number of sub-apertures have been selected.

As discussed in section 2.2, intra-aperture phasing uses a three illuminator cluster. Once the sub-apertures are selected, the program will take the user back to the command window with the prompt, “Do you want a cluster of three closely spaced
illuminators? Enter y or n then press enter". The program was written so that it could be used for any type of MIMO array processing, not just the one described in this paper. If the user chooses anything other than y or n, the statement, "You chose poorly. You get a cluster of three closely spaced illuminators" appears in the command window. If the user chooses y, n, (for yes or no) or anything else, then a new prompt will appear stating” How many additional illuminators will you choose? Enter a number 0 - 4 then press the enter key.” If the user chooses something else, a new prompt will appear, stating "You chose poorly. Number of sub-aperture will be 0". The limit was set arbitrarily at 4. The program is written to not allow any more than that limit. The limit of maximum illuminators can easily be changed in the setup section. The variable n_ill_max defines the maximum number of illuminators. Though there are no obvious reasons n_ill_max cannot be changed, the program has not been tested with more than 4 additional illuminators. To clarify if you chose 1 then you will have a total of 4 illuminators, the array of 3 illuminators used for intra-aperture phasing, and one additional illuminator.

At this point in the program, the figure containing the hex array with the chosen sub-apertures will be displayed. If the user did not select the illuminator cluster, prompt on the figure will read, “Please choose the location for your first illuminator”. The cross-hairs will appear, and the user clicks on a spot within the figure. If that location is too close to one of the chosen sub-apertures, the prompt will change to “Failed!!! Please click farther from a filled sub-aperture”. This will happen with placement of any illuminator. Once the first illuminator placement is far enough away from the filled sub-apertures, the figure will have a magenta circle in that location and the prompt will change to “Please choose next illuminator location”, as shown in Figure 11a. The user will continue to place illuminators until the reach the number previously specified as shown in Figure 11b for an example of 4 illuminators.
Figure 11: Sub-aperture array illuminators and prompts. a: One illuminator placed. b: Four illuminators placed.

If the user chose the illuminator cluster, the prompt on the figure will be “Please choose the location for the first illuminator in the illuminator cluster”. Once the first illuminator placement is far enough away from the filled sub-apertures, the figure will have a magenta circle in that location and the prompt will change to “Please choose direction for second illuminator”, as shown in Figure 12a. The second illuminator is placed in the direction that the mouse is clicked rather than the location because the three illuminator cluster is set on an equilateral triangle with the side lengths being half that of a single sub-aperture. Placement can be tricky as the third illuminator in the cluster is automatically placed clockwise of a vector formed in the direction from the first to the second illuminators. Also, it is possible to cause the program to error out if the second or third illuminators are chosen in a way that either one ends up outside of the figure’s plottable area. This can be avoided by making sure the first illuminator is not too close to the edge of the plottable area. Once the second and third illuminators are placed, and if further illuminators were specified, the prompt will change to “Please choose next illuminator location”. The user will continue to place illuminators until they reach the
number previously specified as shown in Figure 12b for an example of the illuminator cluster with 1 additional illuminator.

![Figure 12: Filled sub-aperture array with illuminator placement prompts with a three illuminator cluster. a: First illuminator of the three illuminator cluster placed. b: Three illuminator cluster placed with additional illuminator.](image)

3.1.2 Aperture Array Builder Outputs: Synthetic Pupils

At this point, the user participation of the program is over. The program puts out 5 more figures. The first two are of the synthetic pupil shown in Figure 13. Figure 13a shows the synthetic pupil overlap map. This is useful in determining how the pupils will overlap in order to visibly judge if there will be enough for closed form phasing. Figure 13b shows the actual synthetic pupil. This is easier to see as the overlap sections do no confuse the user. The overlap sections are not used in the processed high resolution image, so the non-overlapped synthetic pupil is used for the calculation of the MTF.
3.1.3 Aperture Array Builder Outputs: Modulation Transfer Function

The next figure the program outputs is the full MTF plot. Though the MTF was mentioned, and the general form was given in (1.1), the method of finding the analytic form for a complex aperture will be described. Goodman[1] section 6.3.3 equation 6-30, gives the form of the diffraction limited OTF as

$$
\mathcal{H}(f_x, f_y) = \frac{\int \int P\left( x + \frac{\lambda z, f_x}{2}, y + \frac{\lambda z, f_y}{2} \right) P\left( x - \frac{\lambda z, f_x}{2}, y - \frac{\lambda z, f_y}{2} \right) dx dy}{\int \int P^2(x, y) dx dy}.
$$

(3.1)

Recall that the MTF is the modulus of the OTF. $P(x, y)$ is the pupil function, which in this case is the same as synthetic pupil. Note that the numerator of equation (3.1) is the auto correlation of the pupil function. The Fourier transform of the pupil function in a diffraction limited system is the impulse response. Using the auto-correlation theorem, the OTF of the system can be solved using equation(1.1), which will be given again as

$$
\mathcal{H}(f_x, f_y) = \frac{F\left\{ |h|^2 \right\}}{\int \int |h(u,v)|^2 du dv}.
$$

(3.2)
This leads to an easily programmed form of the MTF as

\[
MTF(f_x, f_y) = \frac{F^{-1}\left[|F\{P(x, y)\}|^2\right]}{\int_{-\infty}^{\infty} P^2(x, y) \, dx \, dy},
\]  

or, in words, the MTF is the normalized inverse Fourier transform of the squared modulus of the Fourier transform of the pupil function.

Figure 14 shows full MTF plots. Figure 14a shows the full MTF for the synthetic pupil shown in Figure 13b. Because of the placement of the sub-apertures and illuminators, the MTF has mirror symmetry across the x and y axis. Since the synthetic apertures coming out of this program are likely not to have this mirror symmetry, it is useful to have a visualization of the frequency response of the whole aperture rather than the usual one-dimensional plots. Figure 14b is an example of a full MTF calculated from a hex 19 aperture with the three illuminator cluster and two additional illuminators, the synthetic pupil for this MTF is not shown. Though the MTF is symmetric along any line passing through the center, this plot does not have the mirror symmetry across the x and y axis. A MTF plot created from only the x or y axis would not show the highest frequency cutoffs in this system. Using this program will help to design apertures with specific frequency responses, such as fully symmetric MTFs from multi-aperture arrays.
**Figure 14:** Example full MTF plots. a: Full MTF plot calculated from the synthetic pupil shown in Figure 13b. b: MTF plot without symmetry across the x or y axis.

Though the full MTF is good for an overall image of the systems MTF, one-dimensional MTFs are also useful in estimating system performance. Figure 15 shows two dimension MTF in the horizontal dimension (Figure 15a) and in the vertical dimension (Figure 15b). These plots give actual frequency cutoffs. The two cutoffs are significantly different due to the difference in the size of the synthetic pupil in their respective dimensions. The cutoff for the horizontal is nearly nine times that of the vertical MTF.

**Figure 15:** Two dimensional MTF plots. a: MTF in the horizontal dimension. b: MTF in the vertical dimension.
There were two options for the synthetic pupil to be used for calculating the MTF. The way this is considered is called aperture weighting. The question was whether to double count the overlap regions. This corresponds to whether the synthetic pupil was considered to be that found in Figure 13a or Figure 13b. Though reference [2] stated that it was necessary to count the overlap regions only once, it did not explain why, or the effect on the MTF. To confirm the necessity of normalizing the amplitude to 1 across the synthetic pupil and its effects on the MTF, the analytic MTF was calculated for both types of synthetic pupil. Figure 16a shows the MTF comparison of using the two different synthetic pupils for a five sub-aperture three illuminator cluster synthetic pupil and Figure 16b shows the comparison for the same aperture but with the additional illuminator, as seen in Figure 13. The MTF plots shown in Figure 16a shows little difference though the dips in contrast are deeper for the overlapped pupil. Figure 16b shows more discrepancy. The shape of the plots start to deviate from each other starting from the cutoff of the three illuminator system. Though this deviation is not large in most places, there is nearly a 10% decrease in the overlapped pupil from the unit amplitude pupil where the two MTFs deviate the furthest. As is expected, the two have the same cutoffs. As the MTF is a measure of contrast at a specific spatial frequency, it seems that having higher contrast would be desired. Though further exploration would be needed to confirm the need to count the overlap regions only once, it appears from the apertures used in this study that this is the case since the contrast is higher in all locations.
This program is easily expanded for additional usages, such as the study of synthetic pupil area to be counted in the MTF. Another experiment that was conducted using an expanded version of the program was to rotate the three illuminator cluster to explore the effect on overlap between two sub-apertures. This experiment showed that the optimal rotation, of the cluster for linear aperture arrays was to have two of the illuminators in line with the aperture array. This was the optimal since it maximized the overlap between two adjacent sub-aperture fields, without changing the frequency response significantly. The optimal positions for illuminators, maximizing or minimizing overlap, and maximizing fill factor are just some of the additional usages that could be added to the program’s shell. It would also be possible to remove the hex array shell to explore other aperture configuration. This could be useful in conformal array evaluations.

3.2 Aperture Array Considerations

In this work we seek to further enhance the work done by Rabb[13] and Gunturk[15]. The goal is to develop a multi-aperture, multi-illuminator system that uses
the non-iterative aperture phasing, here called closed form phasing, that also obtains the higher resolution shown by Rabb et. al. in reference [2].

The aperture array proposed for this work is shown in Figure 17. The green circles in the figure represent imaging apertures. To reduce cost for an eventual experiment, there will only be 5 imaging apertures instead of 19. The red circles are where imaging apertures would go given further resources for two dimensional high resolution imaging. The smaller purple circles represent illuminators. The three on the left are arranged for closed form intra-aperture phasing. The one on the right is for extended resolution.

![Aperture and Illuminator Array](image)

Figure 17: Aperture array for use in closed form high resolution multi-aperture imaging. The green circles are receive imaging apertures. The red circles are where further apertures could be inserted. The smaller purple circles are illuminator apertures.

This aperture array is significantly different than previous reported work. In reference [2], there were many illuminators spaced over about 15 cm, these illuminators were spaced 2.9 cm apart. They used field registration based on the cross-correlation of the speckle fields. The close spacing of the apertures facilitates registration through this
method. When using iterative techniques for phasing the received fields, larger spacing between imaging apertures degrades imaging aperture phasing. This is also true for closed form imaging with larger distances between illuminator apertures. If there is little, or no, overlap in the received fields, there is no way to do closed form phasing of the received fields from the different illuminators, as there is not enough information to use registration. The wide separation between the left and right illuminators may still cause problems if the fields are no longer correlated. It is expected, for the larger separation distance, about 15 cm for our array, that registering the fields from the different illuminator will be more difficult.

Having multiple receive apertures is also different from previous work. References [13] and [15] both had only one imaging aperture. We have developed in simulation a similar technique to the overlapping fields that works to synthesize the multi-aperture array as well as the multi-illuminator. With an appropriate aperture and illuminator spacing, overlapping fields from different apertures should allow closed form inter-aperture phasing of the full synthetic aperture array.

Figure 18 shows how the fields from the linear array of five sub-apertures with four illuminators would overlap. The light blue areas have field information from one illuminator, the yellow from two, and the dark red from three.

Figure 18: Overlapping fields from aperture array with multiple illuminators. The light blue areas are places where field from only one illuminator is present, the yellow areas contain field from two illuminators and the red area have field contribution from three illuminators.

There is significant overlap for the three closely spaced sub-apertures. The illuminators placed on the far side of the array only has overlap of fields with one of its
sub-apertures. We believe that this is enough overlap to facilitate registration of the right most illuminator’s apertures fields with the fields from the other illuminators. As long as there is not significant time for change between capturing the fields from the different illuminators, the sub-apertures will only have to be phased together once and then the fields can be registered as a set. If it takes too much time to capture the fields from all the illuminators, changes in the atmosphere could change the speckle needed for field registration and system movement may cause system aberrations to change. How much overlap is needed to register the fields from the fourth aperture needs to be studied.

3.3 Phasing Simulation

Simulations have been completed creating images using the synthesized aperture that results from combining the four illuminator and five aperture array using the non-iterative phasing.[14] Speckle averaging is not needed for this non-iterative phasing, but is used here for better visualization of the images and for increased accuracy of estimation of the experimental MTF. The simulations propagate 200 speckle realizations 300 m after reflecting from a portion of the ISO 12233 target, shown in Figure 19, to the pupil plane using the angular spectrum method from reference [17]. The ISO 12233 target is used because it was created with the intent of using the slant edge method of calculating the MTF from digital image data. For simplicity of the model, the target was illuminated with a unit amplitude plane wave and then the light propagated through vacuum to the pupil plane. The target had uniform random phase added to each pixel to simulate a rough target. No other noises were added to the system. The individual pupil fields were cropped from the pupil plane field based on a 48.3 mm aperture and illuminator placement, and then saved. The code for propagation is found in appendix A.2. Since reference [13] demonstrated the ability to register pupil
fields by registering the speckle, we do not include that here and instead assume the fields are received without misalignment.

The code for closed form phasing is found in appendix A2. The phasing code uses field information save from the propagation code. The fields were phased without adding any errors to obtain a baseline image as shown in Figure 20. It is clear, looking at the two sides of the figure, that Figure 20(a) containing a single aperture image does not contain the detail that the synthesized image has in Figure 20 (b).

Figure 19: Target used in simulation.
Up to five waves of phase using zeroth and first order Zernike polynomials and up to half a wave of phase using second through fourth order Zernike polynomials were added to the fields from each of the five apertures. These errors simulate aberrations due to the physical optical system. The phase errors were different for each of the apertures, but the same for each of the illuminators received through the aperture. The same phase aberrations were added to each of 200 speckle realization as it is assumed that there is no movement in the system. Figure 21 shows images of the ISO 12233 target formed from the aberrated pupil fields without any phase corrections. The single aperture image in Figure 21(a) is better than the synthesized image in Figure 21(b). This is not surprising since the aberrations are different on each aperture and tip-tilt was set to be the same for each realization of the same aperture.
Figure 21: 200 speckle realization averaged images formed from aberrated pupil fields. (a) Single aperture with one illuminator. (b) Four illuminator and five aperture array.

The non-iterative inter-aperture phasing described in this work as well as the intra-aperture phasing described by Gunturk were applied to the fields to remove the added phase aberrations.[15] Figure 22 contains an aberration corrected image. This image appears just as clean as the non-aberrated image shown in Figure 20. Figure 23 shows a comparison of the denser of the two horn targets from the non-aberrated and aberration corrected images. Although an image sharpness metric shows a difference between the two images visually, the two images appear identical.
3.4 Simulation MTF

The analytic MTF was described in section 3.1.3. To get a quantitative measure of the difference between simulation and theory, a way to measure the MTF using a specific image was desired. The method that was found is the slant edge MTF.
3.4.1 Slant Edge MTF: Theoretical

The method for finding the slant edge experimental MTF is described by Reichenbach[18]. He developed this method due to the finite sampling involved in digital cameras. Film cameras had continuous sampling, whereas CCDs and other digital imaging devices have pixels. Due to the digital nature of the pixels, under sampled frequencies can end up being aliased. When Reichenbach developed the slant edge MTF, it was common to use a knife edge scan to measure the edge spread function, and then use the edge spread function to calculate the MTF of an analog camera. This method did not work on the digital cameras due to the need for Nyquist sampling. Sampling frequencies above the Nyquist limit caused aliasing in the images. This aliasing confused the accuracy of MTF measurements. Using a slant edge allowed for a type of super resolution along the edge when multiple lines were extracted and processed to receive more intensity measurements as the image goes from light to dark along the edge. This allows for single image measurements of the MTF instead of having to scan a slanted knife edge as was required in the original method.

3.4.2 Slant Edge MTF: smrmat3

The MATLAB function sfrmat3 is used to calculate the slant edge MTF[19]. sfrmat3 allows for region of interest based MTF measurements, making it well suited for the ISO 12233 target. Since the function is being used, the process of using it is explained, though not how it under lying code works. An example of this code in use is found in the phasing simulation code located in appendix A.2. In the simulations, the image was allowed to move around the image plane due to tip and tilt phases in the pupil plane. As such, the region of interest (ROI) must be selected by hand. If the edge on the target was always in the same place, a ROI could be programeed rather than clicked on. sfrmat3 comes with several subsidiary functions that can be utilized in
conjunction with the main sfrmat3. The getroi function is one of these subsidiary functions that allows for the user to hand select a ROI. It was used independent of the sfrmat3 function because the same ROI needed to be used on the images from the three and four illuminator systems. Fortunately, the fields where all processed using the three illuminator cluster first, thus the images from the two configurations are located in the same location in the image array. The getroi function requires a single input of the image that an ROI is being extracted from. The outputs of the getroi function are an array that contains the ROI box and the pixel indices for the location of the initial right click as well as the pixel indices for the release location. The pixel indices make it simple to obtain the same location of a different image. In this example, the ROI was selected in the image created from the four illuminator synthetic pupil, but the pixel values were also used to obtain the same ROI from the three illuminator synthetic pupil.

When the getroi function is called, a figure will come up with a prompt saying “Select ROI”, as seen in Figure 24a, and directions in the command window stating, "Select ROI with right mouse button, no move = all". To select the region, the user must right click and hold in one location, drag to a second location, and then release the right mouse button. The user will see a dotted box formed while the right mouse button is held as shown in Figure 24b. The box should be in a location that allows for the slant edge to be completely within the ROI, meaning, when the ROI box is complete, the box it forms crosses the edge in only two locations. Both those crossings need to be on parallel lines of the box. The ROI should not be too close to an edge other than the one that the MTF estimate is being made from. The box is too close if another edge could be affecting the intensity of the target edge. The ROI box also needs to be sufficiently deep into the slant edge that the full contrast change has occurred. The sfrmat3 function automatically decides which orientation the user wants the MTF in. This is determined based on the length of the sides of the ROI box. The short side length is the dimension
that the MTF is estimated in. For an estimate in both dimensions, the ISO12233 target has slanted boxes and sets of three slanted edges that look somewhat like an elongated H. This will require more than one ROI.

In this simulation, the ROI was used to make obtaining the same ROI for two images easier, but it made the usage of the sfrmat3 a little more difficult. The function header for sfrmat3 contains usage information, but the way the function was used in this simulation is different than the header's examples. The function was called as “sfrmat3(1,dx,[]),a)”. The 1 means no gui interface is given, which was not desired because the ROI was needed for multiple MTF estimates. dx is the pixel spacing used to determine frequency spacing in the MTF plots. The way the function is written, dx is passed in millimeters. The function gives an option for windowing which was left blank by passing a null array with the [ ]. a is used by the function to represent the data located in the ROI region. The outputs of the sfrmat3 function that were used are status (which is not needed but is the first output), dat, which is the single sided MTF estimate, and nn2out which is the frequency spacing for dat.
3.4.3 MTF Comparisons

For quantitative comparison, the analytic MTF and the slant edge MTF are on the same plot. The analytic form of the MTF was discussed in section 3.1.3. The analytic MTF is plotted as a solid line with the slant edge as a dashed line. MTF are given in Figure 25 for both the three illuminator (Figure 25a) and four illuminator (Figure 25b) cases. The same data and region of interest are used for both, but the three illuminator version does not use the fields from the fourth illuminator when constructing the synthetic pupil.

![Figure 25: Comparisons of analytic and experimental MTF curves. (a) MTFs for the three illuminator five receive aperture system. (b) MTFs for the four illuminator five receive aperture system.](image)

The analytic and slant edge MTFs match reasonably well considering that the slant edge method was developed for photography, and not for a speckled image. The frequency cutoffs are almost identical. The three illuminator version closely overlaps the analytic MTF, but the four illuminator version falls away from its respective analytic MTF. This may be due to speckle size being closer to the size of the pixel. The pixel size is the same for both the three and four illuminator version, but the speckle size decreases.
as the synthetic aperture size increases. The smaller speckles interact adversely with slant edge MTF algorithm. How the slant edge MTF works in the presence of speckle is an area for further research.
CHAPTER 4

EXPERIMENTAL DESIGN AND SETUP

This chapter describes the equipment that was used and the experiments that were done in an attempt to demonstrate closed form phasing. Though the experiments did not lead to a functional experiment, it is hoped that inclusion of these experiments will be of benefit to the reader. Much of the time spent while working on this project was used trying to figure out why the speckle was changing in between subsequent frames taken with only a difference in time being the controlled variable. This guides the direction of the experiments conducted in this chapter.

4.1 Basic Equipment

4.1.1 Laser

The laser that was purchased for this project was a Cobolt Samba 05-01series laser with a center wavelength of 532 nm and max power of 1000 mW laser for $18,500. It is part # 90277 0532-05-01-1000-500 2014-09-05. This includes the laser head, CDRH controller, cable and PSU. A heat sink was separately purchased. Cobalt sold a 11141 heat sink with fans HS-04 2014-09-05 for $560. This laser has a linewidth of less than 1MHz, or a coherence length of 300 meters. This makes it well suited for the narrow line required for spatial heterodyne where the coherence length of the laser needs to be greater than the round trip distance from the illuminator to the receiver, or path length
matching must be added to the system. The length differences in our system are much less than 300 meters.

4.1.2 Fiber

Optical fibers have a maximum amount of power that can be handled before fiber damage occurs. There are also scattering based effects that can be detrimental to fiber operation. Thorlabs gives a tutorial on fiber damage that will be summarized here for the wavelength and amount of power that it is anticipated will be coupled into fibers[20]. This will be followed by allowable power levels to avoid worrying about Raman and Stimulated Brillouin scattering.

Single mode fibers give the mode field diameter (MFD) as a parameter. This is used as the effective beam diameter. For efficient coupling, a beam needs to be about 80% of the MFD. Area is thus defined by \( A = \pi (MFD \cdot 0.8)^2 \). The given Maximum power density for silica fibers is 1 MW/cm\(^2\). It is then suggested, due to impurities and inhomogeneities as well as dirt and other damage, that a practical value for power density is 250 kW/cm\(^2\). For a single mode fiber that operates at 532 nm, the MFD of the LMA-PM-10 fiber is 8.0±0.8 µm. Therefore,

\[
A = \pi (4 \cdot 0.8)^2 = 32.1699 \mu m ,
\]

with a damage threshold of

\[
2.5 \cdot 32.1699 = 80.4248 mW .
\]

It is probable that the fiber could safely handle more than this since the practical value was used rather than the maximum value.

Information in this section is retrieved from Agrawal's Nonlinear Fiber Optics, Chapter 8[21]. Raman scattering is detrimental because it changes the frequency of a portion of the light. This is a loss mechanism. A modified version of Agrawal's equation 8.1.13,
allows us to estimate the critical power for our system, or the power where the Raman scattered light begins to build up. The effective length,

\[ L_{\text{eff}} = \left[1 - \exp\left(-\alpha_pL\right)\right] / \alpha_p, \]

where \( \alpha_p \) is the fiber loss and \( L \) is the fiber length.

For the intended fibers, \( \alpha_p = 30 \text{dB/km} \) and a maximum fiber length of 3 m will be assumed. The Raman gain is a parameter we could not find for a wavelength of 532 nm and pure silica, but with GeO\(_2\) doping the Raman gain is (Appl. Phys. Lett. 102, 201107 (2013); doi: 10.1063/1.4807620), \( g_s = 2.9 \times 10^{-13} \text{ m/W} \). It is believed that this number is reasonable for pure silica fiber as well since Agrawal gives \( g_s = 1.3 \times 10^{-13} \text{ m/W} \) for a wavelength of 1.064 \( \mu \text{m} \) and states that \( g_s \) is inversely proportional to wavelength. With 532 nm being half that of 1.064 \( \mu \text{m} \) and \( g_s \) being slightly higher than twice, this assumption leads to a value that is little high, but a good enough approximation. Using the Raman gain value of 2.9 \( \times 10^{-13} \text{ m/W} \),

\[ L_{\text{eff}} = \left[1 - \exp\left(-1.3\right)\right] / 1 = 0.95, \]

\[ P_{\text{cr}}' \approx \frac{16A_{\text{eff}}}{g_sL_{\text{eff}}} = \frac{16 \times 32.1699E-12}{2.9E-13 \times 0.95} = 1.87 \text{ kW}. \]

This is well higher than the power available for this project.

Information in this section is retrieved from Agrawal’s Nonlinear Fiber Optics, Chapter 9[21]. The critical power for threshold gain for Brillouin is of the same form as the Raman scattering. The modified version of Agrawal’s equation 9.2.4 is,

\[ P_{\text{cr}}'' \approx \frac{21A_{\text{eff}}}{g_sL_{\text{eff}}}, \]
which will give a much different number than the Raman scattering as the Brillouin gain is much higher at $g_s \approx 5 \times 10^{-11}$ m/W. The allowable power in the chosen single mode fiber for SBS to not be significant is,

$$P_{cr} \approx \frac{21 A_{eff}}{g_s L_{eff}} = \frac{21 \cdot 32.1699 \times 10^{-12}}{5 \times 10^{-11} \cdot 95} = 10.8 W.$$ (4.8)

This is an order of magnitude higher than the full power from the Cobolt Samba that was purchased.

Power handling was the primary concern with the chosen wavelength. Because the single mode fiber power limit was under 100 mW and it was intended that power into fiber would be more than this, it was originally assumed that the solution for coupling higher power into fiber was photonic crystal fibers. Though the core sizes of the photonic crystal fibers are significantly larger than standard single mode silica fibers, thus increasing the amount of power that can be coupled into the fiber, using fiber endcaps enables single mode fibers to have a greater amount of power coupled into them than the photonic crystal fibers. Photonic crystal fibers with endcaps would be able to handle significantly more power than standard single mode fiber with endcaps, but for this project we do not need the extra capacity or cost.

The reason endcaps increase the amount of power that can be coupled into fiber is that much of the damage to fibers happens at the air-fiber boundary. To properly couple light into a fiber, the spot size at the fiber tip must be smaller than the core size. Since the required core size to keep single mode operation is proportional to the square of the wavelength, the spot size is much smaller for visible wavelengths as compared to IR wavelengths. As the power goes up the power density at the air-fiber interface passes the damage threshold at low powers. Fiber end caps allow for the spot size to be much greater at the air-fiber interface.
The endcap is a piece of fiber that is index matched to the fiber it is fused to. This endcap fiber is different because it does not have a core. The lack of a core allows for the light to continue focusing within the endcap till it reaches the actual fiber. Because the endcap and fiber are index matched, there is no hard boundary for the light to scatter off of when it reaches the fiber core. This lack of boundary allows for the higher power density and thus the damage threshold is greatly increased. The intended fiber had a numerical aperture of .12 with an endcap length of 330 µm and a core diameter of 3 µm. The numerical aperture is given by

\[ NA = n \sin \theta, \]

where \( n \) is the index of refraction, about 1.46, and \( \theta \) is the half acceptance angle. With these parameters, \( \theta = 4.7^\circ \). Using this angle, the beam acceptance radius at the air-endcap boundary is 28.7 µm. This gives an acceptance area of \( 2.6 \times 10^{-9} \, \text{m}^2 \). This compares to a core area of \( 7.1 \times 10^{-12} \, \text{m}^2 \). Having fiber endcaps gives an area increase of two orders of magnitude. Figure 26 shows how fiber end caps increase the capture area for a fiber.

![Diagram of coupling light into fiber with endcap.](image)

Since the endcaps allow for much higher power levels, various lengths of Coastal Connections P-FAnskFAnsk-3.8/125/ were purchased. These allowed for intensities greater than the entire output of the laser.

To couple light into the fibers ThorLabs’ Adjustable Focus FC collimators were used. The collimators can be easily attached to the ends of the fibers and consist of a
single aspherical lens in a spring-loaded stainless steel mount. Although, Thorlabs’ intended purpose of these adjustable focus collimators is to collimate the light exiting a fiber they work for focusing collimated light into a fiber as well.

5-degrees of freedom are needed for aligning the fiber tip and coupling the laser into the fiber. In order to couple light into the fibers, there needs be a way to control the orientation of the fiber, now fitted with the collimator. There needs to be control of fiber tip and tilt, as well as horizontal and vertical position. The adjustability of the collimator tip allows for focus adjustment. For compact purposes we will use ThorLabs 5-axis kinematic mounts for coupling light into the fibers. Although the end-capped fiber allows for higher power density going into the fiber core, when coupling light into these fibers it is required that the laser power is tuned down to well below the damage threshold.

To couple light into fiber using the fiber collimator tips, adjust the tip so that the focus lens almost falls out. This allows for the largest spot size on the fiber tip, thus the lowest power density. Adjust tip/tilt and the axial position until a maximum occurs on a power meter. Adjust the focus ring, but make sure there is still light coupled into the fiber, otherwise you will be required to start over. Due to spring control of the lens location, when the focus ring is adjusted, the lens orientation is also adjusted. This throws off previous alignment to some degree. Adjust tip/tilt and the axial position again. Repeat these steps until there is no more increase in power coupled into the fiber. Dependent on the fiber tip collimator used, coupling efficiency up to 75% was achieved.

4.1.3 Cameras

Five Point Grey Grasshopper 3 USB3 cameras were purchased for this project. These are GS3-U3-23S6M-C cameras (Sony IMX174 1/1.2" CMOS global shutter; mono; 1920 x 1200; 162 FPS; C-mount), at a cost of $1085. each. These cameras have a large well depth, >32,000 electrons, and can be gated down to 5 μsec. These cameras were purchased because of the low gate time and global shutter. The lower
gate time means that the data will not be affected as much by system movement. Global shutter cameras capture the data at the same time and then read the information out. Frequently cameras will have a rolling shutter which will continuously read out the data allowing for motion blur. The global operation is desirable in spatial heterodyne images so that the fringes caused by the interference of the signal and LO beams are not washed out.

4.2 Multi-Aperture Lab Setup

4.2.1 Compact Range and Optical Table Setup

It was intended that many of the experiments for this project would be conducted in the UD compact range. This is a modification of the compact range described in reference [22]. The compact range is a demagnification telescope which allows simulated longer ranges in the LOCI 100’ range hall. The de-magnification was set to 20x. The simulated range is the magnification squared times the actual separation, so, for example, a 2.5 meter distance becomes a simulated range of $2.5 * 20^2 = 1000$ meters. A notional diagram of the compact range is shown in Figure 27.

![Figure 27: Notional diagram of compact range.](image-url)
Figure 28a shows the configuration of the optical tables on which the compact range was re-built and Figure 28b shows a picture of the tables with some of the equipment set out on them. In order to eliminate the effect of vibrations, three optical tables were bolted together. The large mirror (24 inches (61 cm) in diameter), was placed in the center, side to side, of the right 8 foot long optical table. The large mirror is heavy, but this table has legs capable of handling 2000 pounds each. The mirror surface is placed approximately 4 feet from the end of the table. The tables were bolted together using 18 foot six inch long, five inch tall and wide steel I-beams. These are also heavy, but were balanced with most of the weight distribution on the central 8 foot optical table. Later the central table was dropped out due to issues with its stabilized legs. This meant that the weight was held up by the ends of the two remaining tables. The IMAGE 2 testbed is placed on the 12 foot optical table. For simulating ranges, the test target can be placed on the any of the three optical tables.
4.2.2 IMAGE 2 Testbed

A modified version of the IMAGE testbed was developed and is shown in Figure 29. This will be termed the IMAGE 2 testbed. The original was developed at UD by Miller[22]. Figure 29a shows the aperture array on the front of the IMAGE 2 testbed. 5 holes were chosen to be filled with receiver instrumentation to provide almost the same resolution in one dimension as full hex 19 array. In a real embodiment for two dimensional high resolution imaging a hex 19 with all 19 sub-apertures filled would be used. The optical design for the sub-apertures for the IMAGE2 testbed will be that of the original. Four illuminators have been added to the testbed in this iteration. The original design was set up for a single illuminator that was not connected to the holding structure. The IMAGE2 testbed has a rotatable 3 illuminator cluster as well as an adjustable fourth illuminator. Figure 29b shows a mostly unfilled side view of the
IMAGE2 testbed. Figure 29c shows the filled IMAGE 2 testbed. As can be seen from this image, there are a lot of fibers, cameras, and sub-apertures involved in this system.

![Image 2 testbed](image)

Figure 29: a: Head-on view of IMAGE 2 testbed. b: Mostly unfilled side view of IMAGE 2 testbed. c: Filled side view of IMAGE 2 testbed.

4.2.3 Sub-Aperture Design

The sub-aperture design is the same as the IMAGE testbed. There was some difficulty in locating the design, so it is added here. The tolerances for the telescope
were extremely loose. Any lens could be adjusted in between the first and last elements. When the first lens was used to refocus the system, the image size would be exactly the same. The Zemax design parameters are shown in Figure 30a. Figure 30b shows three set of rays from different angles passing through the sub-aperture.

![Figure 30: Sub-aperture design. a: Zemax design parameters. b: Ray diagram of light passing through sub-aperture.](image)

### 4.2.4 One to Five Free-Space Splitter

Five receive apertures require five LOs when processing images with spatial heterodyne. Though in fiber splitters are inexpensively available for near IR wavelengths, they are not as mature or cheap at 532 nm. As such, the LO power was split in free space using a holographic grating. This grating was purchased from Leon Glebov’s company, Optigrate. This grating is polarization maintaining. It is aligned by using a rotational stage. The splitter puts out seven spots but the two extra spot are
more than two orders of magnitude down from the 5 desired split beams. Table 1 gives the input power into the splitter, the output power into each beam, and the percentage of the input beam that is spread into each beam. The efficiency into each desired order is well within the quoted specifications and close enough to the desired 20% per beam. There are variations from a low of 16.8% to a high of 22%. The efficiencies can be adjusted by adjusting the angle of the holographic grating with respect to the optic axis. A slight bump of the grating will cause the efficiencies to change significantly, though fiber coupling is maintained. Figure 31a shows the holographic grating. Figure 31b shows the illumination pattern of the holographic grating. Note that the dots on either end are considerably dimmer. Figure 31c shows the breadboard layout for using the holographic grating to split the beam for insertion into fibers. The extra two dots are not fiber coupled.

Table 1: Efficiencies of power split into each beam for the 1 to 5 splitter. The farthest left (LLL) and farthest right (RRR) are > two orders of magnitude down from the desired 5 beams.

<table>
<thead>
<tr>
<th>Input (µW)</th>
<th>4148</th>
<th>Beam Position</th>
<th>LLL</th>
<th>LL</th>
<th>L</th>
<th>C</th>
<th>R</th>
<th>RR</th>
<th>RRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output (µW)</td>
<td>6</td>
<td>736</td>
<td>912</td>
<td>697</td>
<td>862</td>
<td>822</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Percentage</td>
<td>0.14%</td>
<td>17.74%</td>
<td>21.99%</td>
<td>16.80%</td>
<td>20.78%</td>
<td>19.82%</td>
<td>0.17%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.2.5 Free-Space Optical Switcher

A liquid crystal polarization grating (LCPG) was purchased from Boulder Nonlinear Systems (BNS) to be used as an optical switch. This LCPG was used to switch between different illuminators. Reference [23] contains information on this type of beam steering device, but a short description will be included here. This device has two liquid crystal devices that act as half-wave plates. The half-waveplates are able to switch in the µs regime. There are static polarization gratings that redirect the light into one of four beams based on the voltages supplied to the liquid crystal cells. To operate the LCPG, circularly polarized light is directed through the device. There is an arrow on the top of the LCPG to indicate the direction of propagation. The two sets of leads are connected to a waveform generator. Table 2 contains the switch voltages. The
switches should be driven with a 2kHz square wave with a peak-to-peak voltage of +/- V1 or +/-V2. The LCPG can handle a fair amount of power, but it has been damaged by power greater than .5W for a beam approximately 2mm in diameter. The LCPG can be translated to an undamaged location to allow it to function normally. The voltages for the device had to be re-optimized in order for the device to be brought back to peak steering efficiencies. The voltages found in Table 2 will work well enough for starting points for the re-optimization.

Table 2: Steering efficiencies for BNS liquid crystal polarization grating.

<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
<th>Spot location</th>
<th>Total power</th>
<th>Spot power</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>599 NA</td>
<td>457</td>
<td>0.747954</td>
</tr>
<tr>
<td>1.81</td>
<td>2.07</td>
<td>9.95</td>
<td>611</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.03</td>
<td>1.15</td>
<td>5.01</td>
<td>612</td>
<td>493</td>
<td>0.805556</td>
</tr>
<tr>
<td>1.81</td>
<td>1.21</td>
<td>-5.01</td>
<td>614</td>
<td>486</td>
<td>0.791531</td>
</tr>
<tr>
<td>1.03</td>
<td>1.96</td>
<td>-9.95</td>
<td>610</td>
<td>477</td>
<td>0.781967</td>
</tr>
</tbody>
</table>

Figure 32a shows the LCPG device. The blue and black wires coming off the bottom of the device are the leads for the liquid crystal half-wave plates. Figure 32b shows a diagram of beam deflection. The beams are deflected to -10°, -5°, 5°, and 10°. Examples of the beam pattern after the LCPG device are found in Figure 33. Figure 33a shows the illumination pattern when there is no voltage applied to the device. Figure 33b shows the illumination pattern for 5° of beam deflection. As can be seen, there is no single spot, but the majority of the power is sent to the desired location. Table 2 has the deflection efficiencies. Higher efficiency may be obtainable with the voltage optimization mentioned above. The two liquid crystal switches seem to be separable for optimization. Each switch location is optimized separately for the best results. Figure 34 shows the breadboard layout for the LCPG switch to be implemented for fiber coupling. There is a linear polarizer and quarter-wave plate before the LCPG switch. The polarizer makes sure that the polarization in as linear as it can be before entering the system. The
quarter-wave plate is used to make the light circularly polarized. Missing from the picture is another quarter-wave plate, after the switcher, to make the light linearly polarized again. Note that the negatively deflected and the positively deflected beams will be orthogonally polarized to each other. This is due to the effects of the LC half-wave plates inside of the switching device. Since the light was being coupled into fibers, the fast-axis of the fiber was oriented to the direction of polarization for each of the beams.

Figure 32: a: BNS liquid crystal polarization grating. b: Diagram of beam deflection.
Figure 33: Liquid crystal polarization grating beam patterns. a: No voltage applied. b: Voltage applied for 5° of beam deflection.

Figure 34: Breadboard layout of LCPG switch for free-space to fiber implementation.

4.2.6 Digital to Analogue Converter

A digital to analogue converter (DAC) was determined to be the best way to operate the BNS LCPG because of the need for controlling the voltages to drive the switching. A waveform generator could have been used, but it was desired that the LCPG device could be controlled in Labview. In order to get two analogue out channels that could put out a 2 KHz signal at the required voltages, the national instruments NI PCIe 6321 was chosen. Due to additional cost of the periphery equipment, only the DAC and cable were purchased from NI. The breakout board was purchased from Windford Engineering. The breakout board, part number BRKAVH68FV1-R-FT, was
compatible with the NI cable SHC68-68-EMP. The breakout board is found in Figure 35. The pinout diagram, originally found in the NI X series multifunction data acquisition data sheet[24], found in Figure 36, allows for wiring from the breakout board to the device.

Figure 35: Winford Engineering BRKAHV68FV1-R-FT breakout board.

Figure 36: NI PCIe 6321 pinout diagram.
The DAC was also useful for having digital out ports. Again using Labview in conjunction with the DAC, the cameras could be triggered in-sync with each other. This DAC allowed for complete automation of data collection for large numbers of realization in a very short amount of time.

4.2.7 Speckle Rotator

The last piece of equipment that was needed to fully automate data collection was a way to change the speckle realization. Originally, it was intended to rotate a plane of ground glass, which is a common practice. It was desired that this rotation be comparable in time scale to the camera gate time and the LCPG switch time. It was determined that the best device on the market for what was needed was the Zaber T-RSW60C-KT04U rotational stepper. This Zaber stepper does not work on the order µs, but with running five cameras the computer was not able to pull the data off fast enough anyway. The stepper does work in the ms regime which is actually where the rest of the devices can operate in conjunction with each other.

The ground glass idea for speckle realizations was determined to be a non optimum plan. Because the target was to be illuminated and the image received through the compact range, the light would have to pass through the glass twice. Since ground glass is a volume scatterer and would not be directly against the target, the different angles from the different illuminators would cause independent speckle realization rather than the overlapped speckle patterns needed for speckle registration. As such it was determined that rotating a target that was a diffuse scatterer was a better option. Rotating the target is the concept behind inverse synthetic aperture radar radar or inverse synthetic aperture lidar, so this implementation is similar to a real system. Target rotation allows for dependent speckle realizations for usage in speckle registration in the pupil plane, but locally independent speckle in the image plane. An
image of the stepper with a target mounted on the axis of rotation is shown in Figure 37. Specifications for the stepper are found in Table 3.

![Image of Zaber rotational stepper motor with target mounted on the central axis of rotation.](image)

**Figure 37: Zaber rotational stepper motor with target mounted on the central axis of rotation.**

**Table 3: Zaber rotational stepper motor T-RSW60C-KT04U specifications.**

<table>
<thead>
<tr>
<th>Specification (click for definition)</th>
<th>Value</th>
<th>Alternate Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microstep Size (Default Resolution)</td>
<td>0.0009375 degrees</td>
<td>16.362 urad</td>
</tr>
<tr>
<td>Integrated Controller</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Recommended Controller</td>
<td>X-MCB1 (48 V) Recommended</td>
<td></td>
</tr>
<tr>
<td>Range</td>
<td>360 degrees</td>
<td></td>
</tr>
<tr>
<td>Accuracy (unidirectional)</td>
<td>0.14 degrees</td>
<td>2.443000 mrad</td>
</tr>
<tr>
<td>Repeatability</td>
<td>&lt; 0.02 degrees</td>
<td>&lt; 0.349 mrad</td>
</tr>
<tr>
<td>Backlash</td>
<td>&lt; 0.08 degrees</td>
<td>&lt; 1.396 mrad</td>
</tr>
<tr>
<td>Maximum Speed</td>
<td>450 deg/s</td>
<td>75.0 rpm</td>
</tr>
<tr>
<td>Minimum Speed</td>
<td>0.000572 deg/s</td>
<td>9.983 urad/s</td>
</tr>
<tr>
<td>Encoder Type</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Maximum Continuous Torque</td>
<td>105 N-cm</td>
<td>148.7 oz-in</td>
</tr>
<tr>
<td>Maximum Centered Load</td>
<td>200 N</td>
<td>44.9 lb</td>
</tr>
<tr>
<td>Maximum Cantilever Load</td>
<td>410 N-cm</td>
<td>580.6 oz-in</td>
</tr>
<tr>
<td>Stage Diameter</td>
<td>60 mm</td>
<td>2.362 &quot;</td>
</tr>
</tbody>
</table>
4.3 General Experiments

4.3.1 Spatial Heterodyne: Transparent Target

Spatial heterodyne has been done for years. As part of the prep work for this project, spatial heterodyne imaging was accomplished in two different configurations. The first configuration was short range with a transparent target, layout shown in Figure 38. Every element in this design was free space. This made it more difficult to adjust the angle between the LO and signal beams.

![Figure 38: Layout of spatial heterodyne experiment with transparent target.](image)

The two images in Figure 39a show the direct detection image of the target with the LO present and Figure 39b shows the spatial heterodyne image. The spatial heterodyne image clearly has less noise, but it suffers from fading. The noise reduction is due to the spatial frequency content that made up the noise being outside of the crop region for the spatial heterodyne processing. The image fading is due to improper mode matching between the signal beam and the LO. This wasn't a great image due to the improper mode matching and the fact that the pupil plane image had no pupil as seen in Figure 40. Figure 40a shows the full pupil plane of the image shown in Figure 39a. Figure 40b shows only the top left quadrant of that image. Without a proper pupil, as
shown in Figure 1, it was hard to know what to crop out to create the spatial heterodyne image. Even without the proper pupil, the image in Figure 39b does show that the spatial heterodyne processing can reduce noise in an image and that certain features are much clearer in the processed image.

![Image](image1.png)

**Figure 39:** a: Direct detection spatial heterodyne image. b: Processed spatial heterodyne image. Noise reduction is due to the spatial fringe noise being located outside of the crop area. Image fade is due to improper mode matching.

![Image](image2.png)

**Figure 40:** a: Pupil plane image of transparent target image shown in Figure 39a. b: Top left quadrant of pupil plane image. This section is used to create the spatial heterodyne image.

### 4.3.2 Spatial Heterodyne: Reflective Target

Transparent targets are generally of little interest in the long range imaging systems that synthetic aperture ladar is used for because light is not reflected back to the receiver. The second spatial heterodyne imaging configuration was a reflective target
which was illuminated and imaged through the compacted range (discussed in 4.2.1). A setup diagram is found in Figure 41a and a picture of the aperture, LO, and illuminator, is shown in Figure 41b. There was not enough allowable room in between the aperture and the required location of the LO for a proper fiber mount. The LO is mounted in a pressure mount making it unsteady and difficult to aim.

Figure 41: Single aperture spatial heterodyne imaging experiment using the compact range. a: Setup diagram. b: Picture of aperture assembly, illuminator, and LO.

Figure 42a shows a direct detection image with LO present that was captured through the compact range and Figure 42b is an image of just the LO intensity. It seems that the highly visible fringes in both images are due to reflections within the cover glass of the camera as there were no other surfaces between the fiber tip from which the light was emitted and the detector array on which it was captured. The brighter object to the right of the target in Figure 42a is a reflection off of the eyepiece. The eyepiece was rotated until the reflection would not interfere in the image.
Figure 42: Left: Direct detection image of target with LO also on the FPA. Right: Image of LO only.

Figure 43 shows the pupil plane image of Figure 42. This is a fairly clean pupil plane image, though a piece of the pupil was aliased across the frame. The green circle represents what would have been ideal to crop to use for constructing the spatial heterodyne image had the pupil not been aliased across the pupil plane. The red circle shows the autocorrelation terms. It is desired that the red and green circles do not overlap. Much of the noise is encircled in the red circle. Part of the reason that the pupil ended up aliased across the frame is that we were trying to keep it out of the autocorrelation area. The length of the receiving telescope was also altered from the theoretical optimum causing the pupil to be enlarged. The other reason the pupil was aliased was that the holding structure for the LO was non-ideal, causing fine adjustment to be difficult. Given flexibility of LO placement, the LO could have been shifted to place the pupil fully in the top left quadrant and away from the auto-correlation noise.
Figure 43: Pupil plane image of Figure 42a. The green circle represents what would have been cropped if the pupil had not aliased across the frame. The red circle denotes the autocorrelation terms that are desired to be outside the pupil.

Figure 44a shows the block target used in this experiment and Figure 44b shows the spatial heterodyne image. The target was printed on standard copy paper using an ink jet printer. Some of the noise in the image may have been caused by poor contrast in the printing rather than poor contrast in the imaging system. The separated section on the left of the image was not visible in the direct detection image in Figure 42. The dark area in between is caused by non-uniform illumination of the target. Noise reduction is evident in this image as well as the image in Figure 39. The mode matching was much better in this image. There may be some loss of contrast from the direct detection image due to loss of the part of the field that was aliased across the frame. After this experiment was performed it was discovered that paper depolarizes incident illumination from reference [25] chapter 4.
Figure 44:  a: Block target.  b: Spatial heterodyne image of target processed from the top left quadrant pupil in Figure 43.
CHAPTER 5

EXPERIMENTAL PROCESS

5.1 Speckle Registration

5.1.1 Speckle Registration: Theory

In section 2.4, it was mentioned that the simulations would not include pupil field registration because it had previously been done by references [2][13]. In these references they describe how the fields shift when there are multiple separated illuminators. This field shift includes a shift in the speckle. A diagram of this field shift is shown in Figure 45. The red dot in the diagram is a specific piece of information. The dot moves in the opposite direction of the illuminator shift. To overlap the dot, the aperture field is moved in the same direction as the illuminator.
Speckle registration became the biggest challenge in this project once the simulations were done and real data was involved. The concept of registration is based on cross correlation. The formula for correlation is

\[
g(x_i, y_i) \otimes f(x_i, y_i) = \int_{-\infty}^{\infty} g^*(x_2, y_2) f(x_2 - x_i, y_2 - y_i) \, dx_2 \, dy_2. \tag{5.1}
\]

This is a computationally long process. Fortunately, the convolution theorem can be used on this to get

\[
g(x_i, y_i) \otimes f(x_i, y_i) = \mathcal{F}^{-1} \left\{ G(f_{x_i}, f_{y_i}) F^*(f_{x_i}, f_{y_i}) \right\}. \tag{5.2}
\]

Multiplying the Fourier transforms of the two functions and then taking the inverse Fourier transform is much more efficient than hard coding the cross correlation. To get sub pixel registration, the up sampled Fourier transforms are used. This runs into issues if registration on the order of a thousandth of a pixel is desired because the array needs to be up sampled by a thousand. Though upsampling by a thousand may be extreme, wanting registration to an hundredth of a pixel is reasonable. Guizar provides a solution to the up sampling problem by up sampling twice[26]. The first time is by a factor of two. By using this small up sampling factor, an estimate of the registration can be made. A smaller portion of the data is cropped and used with the full up sampling factor. This
makes it so the up sampling does not over run the memory of the computer and is much faster. Guizar also provided the code for performing sub pixel registration[26].

5.1.2 Speckle Registration: Usage

Implementation of his code is simple though it is necessary to read the instructions for full usage. To use the code with shifted pupil fields, fields from the same aperture but different illuminator, the approximate overlap needs to be known and the pupil fields need to be cropped so that the majority of the fields actually contain the same information. Even with the cropped fields, the speckle registration needs to be done using the intensity of the pupil fields instead of the fields. Field registration can be used if the pupil shift is fairly accurately known. In this study it seemed like the intensity registration was more precise.

Reference [2] describes their process of aperture phasing. Their methods for phasing are different than those described for this project, but their steps guided the approach used for this project. They also used pupil plane spatial heterodyne, rather than the image plane version used in this project. Their first step is a calibration step. Aperture specific aberrations are static and can be removed by creating a flat field across the image plane using spatial heterodyne. In previous work[2], this was done by creating a plane wave in the entrance pupil, but for this project this was achieved by placing a point source in the entrance pupil of the aperture. This was changed because the project from reference [2] used pupil plane imaging where as we used image plane imaging. Using spatial heterodyne processing, the field is obtained in the image plane. The inverse of the phase from that field can be multiplied against the pre-processed image to remove the phase aberrations that were calculated from that point source. Assume that the aperture phase error is described by $e^{i\phi}$. The equation for intensity on the camera is given in section 1.3.3, but with this phase error the equation becomes,
\[
\left| U_{lo} (u,v) + U_s (u,v) e^{i \phi_e} \right|^2 = \ldots \\
\left| U_{lo} (u,v) \right|^2 + \left| U_s (u,v) e^{i \phi_e} \right|^2 + U_{lo} (u,v) U_s (u,v)^* e^{-i \phi_e} + U_{lo} (u,v) U_s (u,v) e^{i \phi_e} 
\]  \hspace{1cm} (5.3)

Recall that \( U_{lo} \) is the LO field and \( U_s \) is the signal field in the image plane. When the inverse of the error phase is multiplied against this equation, the result is

\[
\left| U_{lo} (u,v) + U_s (u,v) e^{i \phi_e} \right|^2 e^{-i \phi_e} = \ldots \\
\left| U_{lo} (u,v) \right|^2 + \left| U_s (u,v) e^{i \phi_e} \right|^2 + U_{lo} (u,v) U_s (u,v)^* e^{-2i \phi_e} + U_{lo} (u,v) U_s (u,v) e^{i \phi_e} 
\]  \hspace{1cm} (5.4)

Using spatial heterodyne processing, the only term that is kept is \( U_{lo} (u,v)^* U_s (u,v) \). If \( U_{lo} (u,v) \) is flat in intensity and mode matched in phase, then \( U_s (u,v) \) (multiplied by LO amplitude) is all that is left in the image plane. This cannot actually eliminate all of the static phase error from an aperture because the calibration is created using a point source in the entrance pupil. This will only cover a small portion of the input element in a system that the entrance pupil and the input element are in the same plane. Even if this is not perfect, it does help. Figure 46a show a portion of the pupil plane when a fiber tip was placed close to the aperture entrance, which is also the entrance pupil. The fiber tip is significantly smaller than the pixel size in the pupil plane, but the spot many pixels across. Figure 46b shows the same image with image plane phase error removal. There are still some artifacts in the image, but the spot size has been reduced to a single pixel. Though it is hard to say exactly what aberrations this takes out, it is probably mostly defocus.
A second calibration step can be done by putting a point source in the far field as done by Rabb[2]. This will create a flat field at the aperture entrance. Similar processing theory and processing steps to that described for image plane error phase removal will create a phase that can be applied in the pupil plane. This blurred the images when implemented for this project. Solutions to this blurring were never explored and this calibration step was eliminated.

Rabb’s next step was to use a sharpness metric using Zernike polynomials to remove additional phase errors. This step is not used in this project because sharpness metrics are iterative methods, and Rabb did not use it in reference[13]. The point of the three illuminator cluster is to be able to calculate the phase errors using the intra-aperture phasing discussed in section 2.2. For this project this goes after Rabb’s step three.

For this project, Rabb’s steps three and four are combined. His step three is to place the apertures from a single illuminator in a common pupil plane. His step four is to use an optimization routine based on cross-correlation. This step sounds similar to the speckle registration, but optimization implies iterative. The way these two steps are
done for this project is different. Instead of placing the apertures from the same illuminators in a common image plane, the fields from the same aperture, but three illuminator cluster, are placed together using the speckle registration. Once the fields from the three illuminator clusters are registered, then intra-aperture phasing is performed.

Though the experimental part of this project broke down when trying to implement the inter-aperture phasing part, the next planned step would have been to register the fields from all five apertures and the three illuminator cluster together into a common pupil plane. After that the fields from the extra illuminator would have been registered using the one that overlapped and then use the now known aperture spacing for the non-overlapped fields. The reason that those spacings are now known is that the registration of the sets of three illuminator cluster fields should have the same spacing as the non-overlapped fields from their respective apertures.

5.2 What Went Wrong

Though the project stopped at the point where intra-aperture phasing could not be accomplished, there were many other problems that surfaced through the project. These will be addressed as they surfaced, so the topic of intra-aperture phasing failing will be covered last. Much of the driving force behind problem discovery was the inability to register the speckle. Frequently, this was the case when registering frames from the same illuminator and aperture with only a time delay between frames.

5.2.1 Table Vibration

Due to experience of the previous student working with the compact range [22], three optical tables were bolted together in hopes that the tables would move together. Even though measures were taken in our optical table design to not have vibration, (the company Etegent Technologies had done theoretical calculations to ensure stability)
there was a low frequency vibration in our data. The small vibration caused the laser to
dance on the target which prevented the returning beam and the LO from mixing well at
the camera, thus corrupting the received data from the cameras.

To investigate the vibrations further, Etegent was asked to take vibration data.
Multiple accelerometers were setup across all three tables to monitor vibrations. Data
sets were taken of the tables under normal lab settings and with forced tapping. Figure
47a shows the vibration data collected from accelerometers placed on every table and
optical elements under ambient conditions. The locations of the accelerometers included
multiple spots on each table, the three mirrors included in the simulated range, the
testbed and target platform, and the floor. This data includes lateral and axial vibrations.
The quality of Figure 47 is low, but peaks at certain frequencies are clearly visible.
Using an accelerometer hammer, over 30 locations were tapped to examine the
vibrations through the table. Figure 47b shows vibration data from accelerometers
placed in the same locations but with the forced tapping.
Figure 47: Table vibration data from accelerometers placed on and around optical tables. This is low quality but it clearly shows strong vibrational peaks. a: Vibration data under ambient conditions. b: Vibration data with excitation from accelerometer hammer.

Although the data contained in Figure 47 is somewhat messy and confusing, the two largest peaks on both charts come from the target platform. This means that the target platform was the largest factor causing vibrational issues. However, the second largest peak on both charts came from the middle table. Realizing there was a problem with the middle table caused the removal of the middle table from the system by unbolting it from the I-beams. The target platform was removed from the middle table and put on the end table with the large mirror, thus increasing the simulated range. It was determined that the middle table was not floating properly, so by removing it the two
remaining tables bolted together were able to float as they should. This solved many of the issues involving with vibration. We continued repeated efforts to secure all optical equipment to reduce movement effects.

5.2.2 Fiber Optics

Getting rid of the high amplitude vibrations helped with being able to track speckle movement visually, but did not yet allow for speckle registration. In an examination of the current setup, it was determined that the fiber collimator tips attached to the illuminators to illuminate the system needed to be removed. This was because the tips had an adjustment for variation of the fiber tip to lens distance. It was realized that the variation would make it appear that the illuminators were not located in the same plane as each other which is a requirement for the fields to shift due to illuminator location.

Also, the illumination spots on target were not uniform. Some of this non-uniformity was caused by the aspherical lens in the adjustable fiber collimation tips. Though these work great for fiber coupling into fiber and the non-uniformity was not visible when actually wanting collimated light, the fiber tip collimators have significantly aberrated wavefronts when used to control the divergence angle of the laser. These aberrations made it impossible to have uniform illumination across the target.

The variable fiber tip collimators were not the only thing causing aberrations in the wavefront. When the collimator tips were removed, it was discovered that the beam pattern was not uniform coming out of the fiber. Figure 48 shows the illumination pattern from a bare fiber tip. This airy pattern was disturbing because single mode fibers are supposed to act as spatial filters. The fibers were returned to Coastal Connections who couldn’t see any problem with the fibers. Once the fibers were returned it was determined that only some of the fibers had airy diffraction patterns. One side of a fiber might give a clean Gaussian diffraction pattern while the other gave the airy pattern. It
was concluded that the fiber endcaps on some of the fibers were either damaged or were long enough to clip the beam before it exited the endcap, though no evidence was discovered. Since flipping the fibers around fixed the problem, no further examination of the fibers was made.

Figure 48: Illumination pattern from bear fiber tip.

Though not directly a problem with the fibers, it is expedient to include that the polish type of the fibers needs to be taken into account. It was thought that getting angle polished fibers was the way to go. This was a mistake for several reasons. Fibers are angle polished to control back reflections. This was not an issue with this system because the laser allowed backscattered power without laser damage. Angle polishing fibers causes the beams to exit the fiber at an angle to the fiber axis. This made it more difficult to point the beams to the same locations. It also makes the beam somewhat elliptical. Endcapped fibers can also cause unanticipated effects. Due to the light diverging through endcap, several devices that rely on a standard distance from fiber tip to device fail. This was anticipated for the fiber collimator tips, hence the variable tips. This also makes butt-coupling non-functional because the fiber cores need to be extremely close so that the light does not have space to diverge to a larger size.
5.2.3 Laser Problems

Much of the frustration and time spent on this project was due to a laser that was not functioning to the required specifications. From the time the laser was received, it had a power fluctuation over a short amount of time. This made any measurements based on power level difficult because the laser would drift several mW over 10 to 20 minutes. This affected perceived coupling efficiencies because the power was measured at 15 mW when coupling was started, and could be as high as 18 mW when finished. For 12 mW of power exiting the power, that is the difference of 80% for the 15 mW input to 67% for an 18 mW input.

The real problem for this experiment was that the laser was causing speckle to change between frames, as seen in Figure 49. At first it was assumed that changes between frames in the speckles were due to system vibrations. Dependant on the data set, we could correlate speckle visibly, though registration failed. Two correlated speckle areas are marked in Figure 49. With all the vibration mitigation that had previously been done there wasn’t much else other than the table that could be seen as the source. At this point the experiment was moved to an actively stablized single table, but the experiment was changed to use only a single aperture and single illuminator. There was still a great enough difference between frames that registration would not work. Vibration was still a possible problem possibly due to the experiment being conducted on the fifth floor of an old factory building. One vibration source that was discovered was the fans attached to the heat sink for the laser. It was assumed that since product description required purchase of the heat sink for a narrow linewidth laser that it was built in a manner that it would not interfere with operation. When speaking with the company it was discovered that other researchers had found way to use the laser without the heat sink fans when using it for holography. They also said that the laser could be used for an estimated time of up to an hour without the fan on before the
laser heat became appreciable. Getting a laser designed without a cooling fan should be a consideration in the future.

![Figure 49: Examples of pupils from different frames. The two circled areas show correlated speckles though the pupils are not correlated enough for registration.](image)

Without the heat sink fan running, speckle change between frames was still present. The system was simplified so that only the laser, target, and camera were present. Looking at the speckle in such a simple system and with larger speckle due to spot size and distance, it was discovered that the speckle was changing shape. At this point the only real possibility was the laser.

The laser was set up in Fabry-Perot interferometer. This test was inconclusive due to precision of the instrument. It was indeterminable whether the trace on the oscilloscope was moving due to a drift in frequency or a drift in the trigger. It was unknown at the time that the dip in the amplitude change in the peak between scans could be accounted for by frequency drift.

Since the Fabry-Perot was inconclusive, a Michelson interferometer was set up instead. Because the coherence length of the laser was supposed to be around 300m, the arms of the instrument could not easily be made that long on a 12’ table. Instead, the beam was intentionally poorly collimated so the fringes on the camera would be
small enough to see. The interferometer breathed more than a full wave, so a box was
place over the interferometer in hopes that this would eliminate breathing due to air
movement. The laser still had to enter and exit the box, but both arms of the
interferometer were entirely in the box. At this point the interference spot was still
breathing so it was determined that contacting Cobolt for help was required. After
running two remote diagnostics and a remote recalibration, it was determined that the
laser should be sent back to manufacturer (Cobolt) for diagnosis of its coherence
properties and stability.

When the laser was sent back to Cobolt, the power fluctuation described at the
beginning of this section was their stated reason for building a new one. Once the new
laser was received, the speckle no longer morphed, the power was much more stable,
the interference spot from the Michelson interferometer only slightly breathed, and the
Fabry-Perot interferometer scans were stable. Figure 50 shows the stability of the trace
from three scan through a Fabry-Perot interferometer. The little peak in between main
peaks is where the scan reset.

![Three scans of a Fabry-Perot interferometer](image)

Figure 50: Three scans through a Fabry-Perot interferometer.
5.2.4 Intra-Aperture Phasing

To maintain simplicity, once the new laser was checked out on the single table, the laser was placed back into the single aperture, single illuminator version of the spatial heterodyne system, as seen in Figure 51. The wavefront difference of two sequential frames looked noisy, but was essentially blank, as seen in Figure 52. As such a second illuminator was added. The wavefront difference of two fields from different apertures was not clean enough for intra-aperture phasing.

Figure 51: Diagram of simple two illuminator spatial heterodyne experiment. The imaging part is a 2f to 2f system for 1 to 1 imaging.
Figure 52: Wavefront difference of two fields from the same aperture and illuminator taken one after the other.

Though the project failed at intra-aperture phasing, the error may have come in before that point. The speckle registration should have been good enough for the intra-aperture phasing to function, but this was not the case. The first step of the intra-aperture phasing was to register the fields from one aperture and the three illuminator cluster. Then, the wavefront difference of two overlapping fields can be taken. An example of the wavefront difference from two different illuminators is shown in Figure 53. There is clearly a fringe pattern seen in the image, but the speckle mismatch is great enough that the information is corrupted. There are several possibilities of what may have caused issues.
One of the difficult parts of this type of system is that the illuminators need to be the exact same distance from the target and essentially coplanar with the entrance pupil. The farther the target is away from the illuminators the exact distance shouldn’t matter as much due to the flatter curvature of the phase front. With the image and object distances being 2m, the phase curvature is still significant. These distances also make it more difficult to get the entrance pupil and the illuminators coplanar while having the target and the three illuminator cluster mount parallel to each other.

The distance can also be a problem due to speckle change dependent on the angle of illumination[27]. This is specific to the shorter table. Going back to the compact range would cause the angle much smaller and eliminate this problem. Since speckle is an interference phenomenon, some of its effects can be studied as fringe patterns. When illuminating a target from an angle, $\theta$, the path length difference will be

$$\delta = \frac{d}{\lambda} \sin \theta ,$$  

(5.5)
where \( d \) is the aperture width and \( \lambda \) is the wavelength. This gives the path length difference in waves. Every time this increases by one, a new fringe will be added to the pattern. The fringe rate is thus given by,

\[
\frac{d \delta}{d \theta} = \frac{d}{\lambda} \cos \theta . \tag{5.6}
\]

The angular extent of the target is, using the small angle approximation, \( \frac{a}{R} \), where \( a \) is the width of the aperture and \( R \) is the range from the target to the aperture. The rate of change of the fringes times the angular extent of the aperture will be the number of fringes across the aperture,

\[
N = \frac{d}{\lambda} \cos \theta \frac{a}{R} . \tag{5.7}
\]

The rate of change of the fringes across the aperture is,

\[
\frac{dN}{d \theta} = -\frac{d}{\lambda} \sin \theta \frac{a}{R} . \tag{5.8}
\]

For this system, the target is illuminated from multiple locations. The change in angle, \( \frac{b}{R} \), times the fringe rate at the aperture will give the fringe difference at the aperture,

\[
\Delta N = \frac{d}{\lambda} \sin \theta \frac{ab}{R^2} \tag{5.9}
\]

where \( b \) is the spacing between illuminators. \( \theta \) needs to be a number such that \( \Delta N \) is much less than 1. The target size can be equal to the image size because the imaging system is 2f to 2f and the image is collected in the image plane. Image plane spatial heterodyne makes it so any light that hits the aperture, but not the image sensor, is lost, thus any interference in the aperture plane where the light doesn’t make it to the sensor effectively never happened. Given \( \Delta N = .1 \), a wave length of 532 nm, a target diameter of 1 cm, a range of 2 m, an aperture diameter of 5 cm, and a illuminator spacing of 25 mm, then \( \theta = 17 \) mrad while the angular spacing of the illuminators is 12.5 mrad. Though
the illuminator spacing is less than the estimated limit of fringe movement, the two numbers are still close. If instead of solving for $\theta$, the same numbers are used to solve for $\Delta N$, using $\theta = \frac{b}{2R}$, which is the angle of the first illuminator, $\Delta N = 36.7 \times 10^{-3}$. This means for this experiment, the change of fringe size is .0367 of a fringe.

Treating the fringes as speckles, the former derivation was for how fast the speckles would change due to a target consisting of two point sources spaced a certain width apart. This will give the fastest that the speckles change. Point sources in between the two would cause slower changing speckle. This change in speckle may have caused the loss of the experimental ability to get the wavefront difference to cancel the speckle in the overlap region.

Another possible look angle problem is caused by depth in the target. The target is a chrome on glass ISO 12233 target as seen in Figure 54. It is mounted in a two inch lens mount. Because chrome on glass is too smooth to create the speckle statistics that were desired, a steel plate was clamped onto the back of the lens mount. The chrome side is facing the steel. The whole mount had originally been flipped with respect to the plate, but this caused the speckles to be uncorrelated between the two illuminators due to the space in between the chrome and the plate. Though the plate and chrome are much closer, less than 2 mm, there is still a possibility that the space is causing a different speckle pattern. The distance is small enough that the patterns are still correlated, but large enough that the different illumination angles cause marginally different patterns. As there is a shadow of the central horizontal bar visible in Figure 54, the possibility that the space is too large is moderate.
Figure 54: Chrome on glass target mounted in a two inch lens mount with a steel plate attached to the back.

The problem with getting a cleaner wavefront difference could be coming from one or all three, or none of these problems. Further research is needed to assess the reasons we are not obtaining speckle registration. Also the problem could possibly be solved if a suitable way to extract the fringe pattern in Figure 53 could be found.
CHAPTER 6

CONCLUSION

6.1 Simulation Conclusions

Theory and simulations have been done that show promising results that closed form phasing of a multi-aperture, multi-illuminator imaging ladar system with extended resolution is possible. These simulations show that the resolution of these ladar systems, according to the MTF, is near that of the analytical values with cutoffs being exactly that of the analytic results. Methods for measuring the experimental MTF in a speckled environment have been discussed. Extended resolution using well separated additional illuminators has been shown to be a possibility.

6.2 Experimental Conclusions

The experimental implementation of a multi-aperture, multi-illuminator ladar system did not come to fruition. This seemed to fail in the need for a wavefront difference from which the closed form solve could be implemented. Many subsidiary experiments were successfully achieved in the field of spatial heterodyne and general equipment that were novel solutions to problems that arose from using a wavelength of 532 nm. Many novel experimental approaches were developed to conduct this experiment at 532 nm.

6.3 Future Work

6.3.1 Future Work: Simulations

There are several improvements and steps forward that could be made to the simulations used in this work.
The propagation simulation could add location dependent target illumination that would simulate illuminators being in different locations. The current simulation assumed that speckle was invariant to the illuminator shifts. This assumption may be set to rest with this addition. Gaussian illumination would also address phase curvature on target. Adding phase curvature may also explain differences in the observed speckle patterns that prohibit the ability to achieve a proper wavefront difference.

Phase noise should be added to the phasing simulations. Phase noise would help assessments of needed overlap for intra and inter-aperture phasing as well as registration. With these assessments, a better design could be made for building the next generation of the IMAGE testbed.

6.3.2 Future Work: Experimental

Experiments should be done to identify why the wavefront difference between fields from two different illuminators was so corrupted. A diffusely scattering ISO 12233 target that did not have a space behind should be purchased. This would eliminate the possibility that different angles of illuminations are causing different speckle patterns due to the gap between the transitive target and the diffuse reflection. A longer range between the target and receiver could be used to make the angular extent of the aperture with respect to the target smaller. This would cause the speckle to change less dependent on illumination angle change. Making sure that the illuminator mount and target are parallel would remove any speckle variation due to different distances from the illuminators to the target.

The slant edge MTF was used to get MTFs in simulation, but was not explored in the lab. An experiment could be run to assess the validity of the slant edge MTF being used for speckled data. One particular experiment would be to judge the effect of speckle of on its ability to measure the MTF when the speckle gets smaller. Using
lenses with the same focal length, but of different size should allow for some conclusions to be made. The changing f# may cause problems with this assessment.
REFERENCES


[29] P. Fricker, “zernfun.m.”.
APPENDIX A

MATLAB Code

The code in this section contains functions that are not included, such as sfrmat3[19], dftregistration[28], and zernfun[29]. An attempt to include a version of all code written for this project will be made.

A.1. Aperture Array Builder Code

This code was discussed in section 3.1. A fair amount of the end of this code has been commented out. It involves using this code to calculate other MTFs for comparative purposes, as well as locating the angle of maximum cutoff frequency.

```matlab
%% Multi_trans_ap_array_and_MTFv2.m
% This simulation allows the user to define a pupil array with multiple
% illuminators by clicking on the displayed figure. It then displays
% the effective pupil with and without overlap and several MTF plots.
% Finally it displays the MTF of a circular pupil of equal area to the
% effective pupil.

% Instructions are given in the command window and in the title of the
% figures. Left click on the figure that matlab brings to the front when
% and how you are directed.

% Author: Jeffrey Kraczek
% University of Dayton
% 3/20/15, Last modified 4/17/15
%
%% Setup
% These are the initial parameters. dx D_phys and ill_scan be changed
dx = 100e-6; % Grid increment
D_phys = 1*22.9e-3; % Physical sub-aperture diameter [m]
ds = 5e-3; % Dead space
ill_s = 3e-3; % Clearence for illuminators from edge of apertures.
cs = ds + D_phys; %center to center spacing
p_cs = cs/dx; % Center to center spacing in pixels
n_ill_max = 4; %Maximum number of illuminators outside of illuminator cluster
%% Hex Center Finder
% Finds the centers for a hex array
```
Num=5; % Must be odd integer. Number of apertures in middle row.
Number of rows.
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(1,2);
subap_x_cen=zeros(NTotal,1);
subap_y_cen=zeros(NTotal,1);
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(1,:) = a1*u + a2*v; %Vector transformation from triangular coordinates
            subap_x_cen(q)=round(center(1,1)/dx); %x center
            subap_y_cen(q)=round(center(1,2)/dx); %y center
            q=q+1; %array index
        end
    else %bottom half of hex array
        for k=-N+i:Nfinal-1
            u=k;
            center(1,) = a1*u + a2*v;
            subap_x_cen(q)=round(center(1,1)/dx);
            subap_y_cen(q)=round(center(1,2)/dx);
            q=q+1;
        end
    end
end

%% Space builder
x_max = max(subap_x_cen) + p_cs; % Largest x pixel value
x_min = min(subap_x_cen) - p_cs; % Smallest x pixel value
y_max = max(subap_y_cen) + p_cs; % Largest y pixel value
y_min = min(subap_y_cen) - p_cs; % Smallest y pixel value
n = x_min:1:x_max; % Pixel numbers in x centered
m = y_min:1:y_max; % Pixel numbers in y centered
Nx = length(n); % X pixels
Ny = length(m); % Y pixels
% [nx ny] = meshgrid(n,m); % Cartesian grid
R_pix = round(D_phys/(2*dx)); % Radius of sub-aperture in pixels
[nxs, nys] = meshgrid(-R_pix:R_pix); % Small Cartesian grid for single sub-aperture
sub_ap = nxs.^2 + nys.^2 < R_pix^2; % fills sub-aperture, ones
ill_pix = ill_s / dx; %Space for illuminator in pixels
[nxill, nyill] = meshgrid(-ill_pix:ill_pix); % Small Cartesian grid for illuminators
sub_ill = nxill.^2 + nyill.^2 < ill_pix^2; % Fills illuminator, ones

%% Build Hex 19
Aperture = zeros(Ny,Nx); %Array that will contain full aperture
for count = 1:NTotal %Fills Aperture with sub apertures
    cxi = subap_x_cen(count) + 1 - x_min; %Center x pixel
    cyi = subap_y_cen(count) + 1 - y_min; %Center y pixel
    Aperture(cyi - R_pix:cyi + R_pix,cxi - R_pix:cxi + R_pix) = sub_ap;%->
% Makes a portion of the Aperture array = to sub_ap which is a single sub-aperture
end
figure
hex19 = gcf;  % gcf is a function that grabs the current figure for figure handles
imagesc(n,m,Aperture);
colormap jet
axis image
drawnow;  % Forces graphing

%% Choose sub_apertures
commandwindow;  % Brings command window to the front
n_aps = input('How many sub-aperture will you choose? Enter a number 1 - 19 then press the enter key.');
if isempty(n_aps) || n_aps < 1 || n_aps > 19  % Makes sure numbers within range
    n_aps = 2;
    fprintf('You chose poorly. Number of sub_aperture will be %d.
', n_aps)
end
cxsa = zeros(n_aps,1);  % Center in x of pick sub_apertures
cysa = zeros(n_aps,1);  % Center in x of pick sub_apertures
sapi = zeros(n_aps,1);  % Indicies of chosen subapertures
bool = zeros(n_aps,1);  % logical array for to check valid placement
message = 'Please click on desired sub-aperture';
CM = [0 0 0.563; 0.5 0 0; 0.563 1 0.438];  % array for new color map
for count = 1 : n_aps
    logical = 0;
    while logical == 0
        figure(hex19)
        imagesc(n,m,Aperture);
        axis image
        title(message)
        [cxsa(count),cysa(count)]=ginput(1);  % Gets pixel values from click
        cxsa(count) = round(cxsa(count));  % Rounds pixels to whole value
cysa(count) = round(cysa(count));
        logical = sqrt((cxsa(count)-subap_x_cen).^2 + (cysa(count)-subap_y_cen).^2) < R_pix;  % Indicates if chosen pixel is within a sub-aperture's area
        if sum(logical) == 0  % checks to see if click off of sub-ap
            beep
            message = 'Failed!!! Please click on desired sub-aperture';
        end
    ind = find(logical);  % finds index of chosen
    if sum(logical) ~= 0  % Records filled sub-ap
        for p= 1:n_aps
            bool(p)=sapi(p)==ind;
        end
        if sum (bool)  % Checks to makes sure didn't choose sub-ap twice
            logical = 0;
        end
    end
end

94
message = 'Failed!!! Please click on a different sub-aperture';
beep
end
end
end
sapi(count) = ind; %Records filled sub-ap

\[ \text{cx} = \text{subap_x_cen}(\text{sapi(count)}) + 1 - x_{\text{min}}; \] %center x pixel within
array
\[ \text{cy} = \text{subap_y_cen}(\text{sapi(count)}) + 1 - y_{\text{min}}; \] %center y pixel withing
array

Aperture(\text{cy} - R_{\text{pix}}:\text{cy} + R_{\text{pix}}, \text{cx} - R_{\text{pix}}:\text{cx} + R_{\text{pix}}) =\]
\[ \text{Aperture}(\text{cy} - R_{\text{pix}}:\text{cy} + R_{\text{pix}}, \text{cx} - R_{\text{pix}}:\text{cx} + R_{\text{pix}}) + \]
\[ \text{sub_ap}; \] % Mrks chosen sub-ap

\text{colormap}(\text{CM})
message = 'Please choose next sub-aperture';

\text{imagesc}(n,m,\text{Aperture});
\text{axis image}

%% Choose transmitters
\text{commandwindow}; % Brings command window to the front
% This section determines how many illuminators are desired
oneorthree = input('Do you want a cluster of three closely spaced illuminators? Enter y or n then press enter','s');
\text{Yes} = 'y'; \text{No} = 'n';
if \text{strncmpi(oneorthree,\text{Yes},1)}
    \text{cluster} = 3;
    message = 'Please choose the location for the first illuminator in the illuminator cluster';
elseif \text{strncmpi(oneorthree,\text{No},1)}
    \text{cluster} = 1;
    message = 'Please choose the location for your first illuminator';
else
    \text{fprintf('You chose poorly. You get a cluster of three closely spaced illuminators\n\n')}
    \text{cluster} = 3;
    message = 'Please choose the location for the first illuminator in the illuminator cluster';
end
\text{Input_phrase} = ['How many additional illuminators will you choose? Enter a number 0 - ', num2str(n_ill_max),', then press the enter key.'];
\text{n_ill} = \text{input(Input_phrase)};
\text{isempty(n_ill) || n_ill < 0 || n_ill > n_ill_max}
\text{n_ill} = 0;
\text{fprintf('You chose poorly. Number of sub-aperture will be %d.\n', n_ill)}
end
\text{cxill} = \text{zeros(n_ill+cluster,1)}; % Illuminator x coord
\text{cyill} = \text{zeros(n_ill+cluster,1)}; % Illuminator y coord
\text{CM} = [0 0 0.563; 0.5 0 0; 0.563 1 0.438; 1 0 1]; \% new colormap
\text{for count} = 1 : n_ill+1 %Illuminator placement
    \text{logical} = 1;
    \text{while sum(logical) ~= 0} %Checks to make sure illuminators in valid location
        \text{figure(hex19)}
imagesc(n,m,Aperture);
axis image
title(message)
[cxill(count),cyill(count)]=ginput(1); %gets pixel from
clicking on figure

cxill(count) = round(cxill(count)); %Rounds to whole pixel
cyill(count) = round(cyill(count)); %Rounds to whole pixel
logical = sqrt((cxill(count)-cxsa).^2 + (cyill(count)-cysa).^2) < R_pix + ill_pix; %->
if sum(logical) ~= 0    %informs illuminator within an invalid
area
    message = 'Failed!!! Please click farther from a filled
sub-aperture';
    beep
else   %Places illuminator within array Aperture
    cxi = cxill(count) + 1 - x_min;
cyi = cyill(count) + 1 - y_min;
    Aperture(cyi - ill_pix:cyi + ill_pix,cxi - ill_pix:cxi +
ill_pix) ...
    = Aperture(cyi - ill_pix:cyi + ill_pix,cxi - ill_pix:cxi +
ill_pix) + sub_ill*3;
end

if count == 1 && sum(logical) == 0 && cluster == 3  % 3 ill
cluster
    log = 1;
    while log ~= 0
        figure(hex19)
        imagesc(n,m,Aperture);
        axis image
colormap(CM)
        title('Please choose direction for second illuminator')
        [cxill(count+n_ill+1),cyill(count+n_ill+1)]=ginput(1);
%gets pixel value from clicking on figure
        a1 = atan2(cyill(count+n_ill+1) - cyill(count),cxill(count+n_ill+1) - cxill(count));  % angle between
first illuminator and second
        a2 = a1 + 60*pi/180;  %angle between first and third
illuminators
        cxill(count+n_ill+1) =
round(cxill(count)+R_pix*cos(a1));  %Finds value of 2nd ill
cyill(count+n_ill+1) =
round(cyill(count)+R_pix*sin(a1));
cxill(count+n_ill+2) =
round(cxill(count)+R_pix*cos(a2));  %Finds val of 3rd ill
cyill(count+n_ill+2) =
round(cyill(count)+R_pix*sin(a2));
        log = sqrt((cxill(count+n_ill+1)-cxsa).^2 +
(cyill(count+n_ill+1)-cysa).^2) < R_pix + ill_pix;
log2 = sqrt((cxill(count+n_ill+2)-cxsa).^2 +
(cyill(count+n_ill+2)-cysa).^2) < R_pix + ill_pix;
        log = sum(log)+sum(log2);   %Checks to see if ill ended
up in invalid areas
        if log~= 0 %Informs of invalid placement
            message = 'Failed!!! Please choose a direction
farther from a filled sub-aperture';
            beep
else %Places ills in array Aperture
cxi = cxill(count+n_ill+1) + 1 - x_min;
cyi = cyill(count+n_ill+1) + 1 - y_min;
Aperture(cyi - ill_pix:cyi + ill_pix,cxi - ill_pix:cxi + ill_pix) ...
= Aperture(cyi - ill_pix:cyi + ill_pix,cxi - ill_pix:cxi + ill_pix) + sub_ill*3;
cxi = cxill(count+n_ill+2) + 1 - x_min;
cyi = cyill(count+n_ill+2) + 1 - y_min;
Aperture(cyi - ill_pix:cyi + ill_pix,cxi - ill_pix:cxi + ill_pix) ...
= Aperture(cyi - ill_pix:cyi + ill_pix,cxi - ill_pix:cxi + ill_pix) + sub_ill*3;
end
end
end
message = 'Please choose next illuminator location';
colormap(CM)
end
figure(hex19)
imagesc(n,m,Aperture);
title('Aperture and Illuminator Array')
axis image
drawnow

%% Effective pupil plane
n_ill = n_ill+cluster;  %Number of illuminators
n_ps = n_aps*n_ill;     %Number of pupils
%(Effective number of sub_aps based on actual sub-aps and number of illuminators)
ap_cen_x = zeros(n_ps,1);
ap_cen_y = zeros(n_ps,1);
for p = 1 : n_aps  %pupil center pixels due to first illuminator
    ap_cen_x(p) = subap_x_cen(sapi(p));
ap_cen_y(p) = subap_y_cen(sapi(p));
end
for p = 2 : n_ill  %cycles through remaining illuminators to get pupil centers
difx = cxill(1) - cxill(p);  % x coordinate transform from changing ills
dify = cyill(1) - cyill(p);  % y coordinate transform from changing ills
for q = 1 : n_aps  %cycles through sub-aps to get pupil center pixels
    ap_cen_x(n_aps*(p-1)+q) = subap_x_cen(sapi(q)) - difx;
ap_cen_y(n_aps*(p-1)+q) = subap_y_cen(sapi(q)) - dify;
end
end
x_max = max(ap_cen_x) + p_cs;  %Largest x
x_min = min(ap_cen_x) - p_cs;  %Smallest x
y_max = max(ap_cen_y) + p_cs;  %Largest y
y_min = min(ap_cen_y) - p_cs;  %Smallest y
np = x_min:1:x_max;  % Pixel numbers in x centered
mp = y_min:1:y_max;  % Pixel numbers in y centered
Nxp = length(np);  % X pixels
Nyp = length(mp);  % Y pixels
pupil = zeros(Nyp,Nxp);  %Array for pupil plane
for count = 1 : n_ps  % Draws Pupil plane
cxi = ap_cen_x(count) + 1 - x_min;
cyi = ap_cen_y(count) + 1 - y_min;
pupil(cyi - R_pix:cyi + R_pix,cxi - R_pix:cxi + R_pix) = ...
pupil(cyi - R_pix:cyi + R_pix,cxi - R_pix:cxi + R_pix) +
sub_ap;
end
figure
pupil_plane = gcf;
imagesc(np,mp,pupil);
colormap(jet)
axis image
title('Synthetic Pupil with Overlap')
%% pupil plane normalization
% pupil plane normalization tells what the theoretical resolution will be
% for a system with multiple illuminators.
mask = pupil > 0;  % Finds where pupils overlap
pupil(mask) = pupil(mask)./pupil(mask);  % Normalizes overlap areas to 1
figure
imagesc(np,mp,pupil);
colormap(gray)
axis image
title('Synthetic Pupil')
p_area = sum(sum(pupil))*dx^2;  % Gets physical area to compare MTF of
% synthetic pupil to monolithic pupil of same area
%% Freeing up space
clear n m Nx Ny nx ny nxs nys sub_ap ill_pix nxill nyill sub_ill np mp
Aperture mask
%% Array MTF
[Ny,Nx] = size(pupil);
if Ny < Nx  % Check to see which dimension is larger to make a square array
  Ny = Nx;
else
  Nx = Ny;
end
nfy = Ny*2+1;  nfx = Nx*2+1;
Ipsf=abs(fftshift(fft2(pupil,nfy,nfx))).^2;  % Intensity point spread function
otf=fftshift(ifft2(Ipsf));  % Optical transfer function
m_otf=max(max(abs(otf)));
mtf=abs(otf)/m_otf;  % Modulation transfer function
clear Ipsf otf pupil
%% Makes shure rounding does nt move the center of the MTF from the
% center pixel
[vals,inds] = max(mtf);
[vals,indx] = max(vals);
[val,indy]=max(mtf(:,indx));
difx = indx - Nx - 1;
dify = indy - Ny - 1;
if difx ~=0 || dify ~=0
  mtf = imtranslate(mtf, [difx,dify]);
end
% %
\[ \text{fx} = (-\text{Nx:Nx})/(\text{nfx} \times \text{dx}); \quad \text{fy} = (-\text{Ny:Ny})/(\text{nfy} \times \text{dx}); \]

\begin{verbatim}
figure
imagesc(fx,fy,mtf)
axis image
colormap jet
title('Three Dimensional MTF')
figure
hor_mtf = mtf(Ny+1,Nx+1:nfx);
plot(fx(Nx+1:nfx),hor_mtf)
axis([0 max(fx) 0 1])
title('Horizontal MTF')
figure
vert_mtf = mtf(Ny+1:nfy,Nx+1);
plot(fy(Ny+1:nfy),vert_mtf)
axis([0 max(fx) 0 1])
title('Vertical MTF')
\end{verbatim}

% %% MTF Rotation and optimal directions by area under the MTF
% rot_mtf = mtf;
% tot_rot = 90;   %degrees of rotation
% rot_inc = 1;    %Increment of rotation
% max_a = 0;      % value of maximum area under MTF
% min_a = 1e12;   % value of minimum area under MTF
% countmx = 0;
% countmn = 0;
% % Long algorithm for coming up with the max and min. It was supposed to
% % find degenerate angles as well, but the areas are not exactly equal so it
% % does not work. Works to find an angle for both, so I didn't worry about
% % fixing it at this time
% for ang = 0:rot_inc:tot_rot
%     rot_mtf = imrotate(mtf,ang,'bicubic','crop');
%     sum_h = sum(rot_mtf(Ny+1,Nx+1:nfx));
%     sum_v = sum(rot_mtf(Ny+1:nfy,Nx+1));
%     if sum_v >= max_a || sum_h >= max_a %Checking for maximum area
%         if sum_v > sum_h    % checks for vertical and horizontal
%             s = sum_v;
%             m = rot_mtf(Ny+1:nfy,Nx+1);
%             angle = ang + 90;
%         else
%             s = sum_h;
%             m = rot_mtf(Ny+1,Nx+1:nfx);
%             angle = ang;
%         end
%     if s == max_a %checks to see if one of the two directions is a degenerate
%         countmx = countmx + 1;
%         anglemx(countmx) = angle;
%     else %if not degenerate sets the new max
%         max_a = s;
%         mtf_mx = m;
%         countmx = 0;
%         clear anglemx
%         anglemx(1) = angle; %Sets angle of max
%     end
\end{verbatim}
if sum_v <= min_a || sum_h <= min_a % Checking for minimum area
    if sum_v < sum_h % Checks to see which direction smaller
        s = sum_v;
        m = rot_mtf(Ny+1:nfy,Nx+1);
        angle = ang + 90;
    else
        s = sum_h;
        m = rot_mtf(Ny+1,Nx+1:nfx);
        angle = ang;
    end
    if s == min_a % Checks for degenerate
        countmn = countmn + 1;
        anglemn(countmn) = angle;
    else % If not degenerate sets new min
        min_a = s;
        mtf_mn = m;
        countmn = 0;
        clear anglemn
        anglemn(1) = angle; % sets new min angle
    end
end
end
figure
plot(fx(Nx+1:nfx),mtf_mx)
if countmx > 1
    tmx = sprintf('Maximum MTF, Angle = %d with %d Degeneracies',anglemx, countmx-1);
else
    tmx = sprintf('Maximum MTF, Angle = %d%c',anglemx,char(176));
end
title(tmx)
axis([0 max(fx) 0 1])
figure
plot(fx(Nx+1:nfx),mtf_mn)
axis([0 max(fx) 0 1])
if countmx > 1
    tmn = sprintf('Minimum MTF, Angle = %d with %d degeneracies',anglemn, countmn-1);
else
    tmn = sprintf('Minimum MTF, Angle = %d%c',anglemn,char(176));
end
title(tmn)
figure
mtfI = gcf;
imagesc(fx,fy,mtf)
axis image
fxmx = max(fx);  fxmn = min(fx);
colormap jet
hold on
plot(fx,fx*tan(anglemx(1)*pi/180),':g',fx,fx*tan(anglemn(1)*pi/180),'-k')
hold off
axis([fxmn fxmx fxmn fxmx])
title('MTF with Maximum and Minimum Slices Marked')
% legend(tmx,tmn)
% % Effective Monolithic Aperture MTF
% R_eff = sqrt(p_area/pi);  %Radius of Effective monolithic aperture
% R_epix = ceil(R_eff / dx);  %Radius in pixels
% ndx = (-R_epix:R_epix)*dx;  %physical vector
% [nx, ny] = meshgrid(ndx);
% mono = sqrt(nx.^2+ny.^2) < R_eff;  %fills the array with effective monolithic aperture
% Ipsf=abs(fftshift(fft2(mono,nfy,nfx))).^2; %Intensity point spread function
% otf=fftshift(ifft2(Ipsf));  %OTF
% m_otf=max(max(abs(otf)));   % max of OTF
% mtf_e=abs(otf)/m_otf;       % MTF
% fx = (-Nx:Nx)/(nfx*dx);
% fy = (-Ny:Ny)/(nfy*dx);
% figure
% plot(fx(Nx+1:nfx),mtf_e(Ny+1,Nx+1:nfx))
% title('MTF of Monolithic Aperture with Equal Area to Synthetica Pupil')
% figure
% imagesc(fx,fy,mtf_e)
% axis image
% colormap jet
% title('Three dimensionnal MTF of Monolithic Aperture')

A.2. Simulation: Closed Form Phasing

This code is broken up into separate functions. The two main functions that call the others will be given first, followed by the subsidiary functions.

Simulated propagation. This program requires an image and subap_centers_ill3.m.

\[
\begin{align*}
\text{lambda} &= 1.55e^{-6}; & & \% \text{Wavelength [m]} \\
L &= 300; & & \% \text{Target to pupil plane propagation distance [m]} \\
D_{\text{phys}} &= 48.3e^{-3}; & & \% \text{Physical sub-aperture diameter [m] 1" 22.9mm 2" 48.3mm with 2.5mm removed for connector ring} \\
cs &= 60e^{-3}; & & \% \text{Camera pixel pitch [m]} \\
f &= 0.750; & & \% \text{Focal length of imaging lens [m]} \\
A &= 256; & & \% \text{Number of camera pixels (assumed A-by-A dimensions)} \\
SR &= 10; & & \% \text{Number of speckle realizations} \\
n_aps &= 5; & & \% \text{Number of speckle realizations} \\
dx &= \frac{\text{lambda} \times f}{A \times \text{pixel pitch}}; & & \% \text{Effective pixel size for propagation simulation [m]} \\
ex_{\text{ill}} &= \{0, 6.25 \times A\}; & & \% \text{number of rows equal to number of extra illuminators} \\
\theta &= 180; & & \% \text{Rotations for illuminator cluster} \\
\text{Num} &= 3; & & \% \text{Must be odd integer. Number of apertures in middle row.} \\
\text{Number of rows.}
\end{align*}
\]
k = 2*pi./lambda;                        % Wavenumber

% 
[n_ill,~] = size(ex_ill); n_ill = n_ill + 3;
N = 6000;                               % Propagation simulation array
(assuming N-by-N count)
n = -N/2:1:N/2-1;                       % Cartesian grid
R_pix = round(D_phys/(2*dx));           % Radius of sub-aperture in pixels
% D = 2*R_pix - 1;                      % Diameter of sub-aperture
[nx,ny] = meshgrid(n);                  % Cartesian grid
nr = (nx.^2 + ny.^2).^0.5;              % Radial grid
pp = (nx.^2 + ny.^2)<R_pix^2;           % Centered sub-aperture pupil

function
border = .20;                          % Linear border fraction
b = (1-border)*(N/2);                   % Absorbing boundary width in pixels
supergauss = exp(-(nx.^2./(b^2)).^20 - (ny.^2./(b^2)).^20); % Absorbing boundary function
p = 1./(dx*N).*n;                       % Frequency space coordinates
[p_sqr,q_sqr] = meshgrid(p.^2);        % Frequency space Cartesian coordinate grid
rho = (p_sqr + q_sqr).^(0.5);          % Frequency space radial coordinate grid
clear p_sqr q_sqr

% Imager number
[subap_x_cen,subap_y_cen,array_info]=subap_centers_ill3('LINEAR',dx,cs,n_aps,Num,D_phys,ex_ill,theta);
subap_cen = [subap_x_cen; subap_y_cen];
if max(max(abs(subap_cen)))>max_pixel_center_coordinate,
    warning('Array size too large')
end
num_pup = length(subap_x_cen);

%%%%% Read in a USAF 1951 intensity target bitmap file %%%%%
imm = imread('iso12233aliased','png');
i_crop = imm(:,:,1);
%%%% Crop out portion of the target and place in center of NxN array
%%%%
i_crop = double(imm(351:1850, 351:1850,1));
imm_crop = abs(i_crop-254);
[nrow,ncol] = size(imm_crop);
I_0 = 2*zeros(N);
I_0(N/2+1-nrow/2:N/2+nrow/2,N/2+1-ncol/2:N/2+ncol/2) = imm_crop;
a_0 = sqrt(I_0);                % Take square root of intensity target

for field amplitude
    figure
    imagesc(100*dx*n,100*dx*n,abs(a_0).^2)               % Display target amplitude
    axis square
    colormap gray

102
Closed form phasing. This program requires Simulated_propagation_data.mat (created from the Simulated Propagation program), zernfun.m, and closed_form_solve2.m. It also requires getroi.m which is a subsidiary function of sfrmat3. The actual closed form
phasing sections are commented out so that the program is run in optimal phasing mode because the closed form phasing part takes too long to run.

tic
addpath('sfr3\sfrmat3_post')
load ('Simulated_propagation_data.mat')
number_of_speckle_realizations = 20; % Number of speckle realizations to compute
[data_length, num_pup, snaps] = size(pupil_data);
n_ill = n_ill;
n_aps = num_pup / n_ill; % Number of actual apertures
subap_x_cen = subap_cen(1,:);
subap_y_cen = subap_cen(2,:);
threei = 1;
if threei == 0;
    n_ill = 3;
    num_pup = num_pup - n_aps;
end
D = R_pix*dx*2; % Physical diameter of a sub-aperture [m]
alpha = -R_pix:1:R_pix-1;
[alpha,beta] = meshgrid(alpha); % Cartesian unit pixel grid
[theta,rad] = cart2pol(alpha,beta); % Radial pixel grid
clear alpha beta
pupil_mask = rad<R_pix; % Pupil mask in 2*R_pix square array
C = 2*R_pix;
%%%% Calculate necessary dimensions of composite array %%%%%
%%%% Note 2X size is sampling requirement when computing intensity from amplitude data
M = 2*(max(subap_y_cen) - min(subap_y_cen) + max(sum(pupil_mask,1))));
N = 2*(max(subap_x_cen) - min(subap_x_cen) + max(sum(pupil_mask,2))));
dimMax = max([N,M]); N = dimMax; M = dimMax;
m = -M/2:1:M/2-1;
n = -N/2:1:N/2-1;
%%%% Select a number of speckle realizations to use
if number_of_speckle_realizations<snaps,
temp = pupil_data(:,:,1:number_of_speckle_realizations);
    snaps = number_of_speckle_realizations;
else
    temp = pupil_data;
end
clear pupil_data
pupil_data = temp;
clear temp;

%% % Closed form phasing
%% % Calculates the m and n indices for the first Zth order Zernike

104
%%% polynomials and saves in the variables 'n_idx' and 'm_idx'

% Zorder = 5; % Calculates the first 5 orders
% n_idx = [];
% for k = 1:Zorder,
%     n_idx = [n_idx k*ones(1,k)];
% end
% n_idx = n_idx - ones(1,length(n_idx));
%
% m_idx = zeros(1,length(n_idx));
% for k = 2:length(n_idx),
%     if (n_idx(k) ~= n_idx(k-1))
%         m_idx(k) = -n_idx(k);
%     else
%         m_idx(k) = m_idx(k-1) + 2;
%     end
% end
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%
%%% Create the first Zth order Zernike polynomials
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% w = zernfun(n_idx,m_idx,rad(pupil_mask)/R_pix, theta(pupil_mask));
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%
[wlength, zernikes] = size(w);
%
% Creates phases to be added
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% piston = .25; tip = 3; tilt = 7; multiplier = .5;
% weight = [piston tip tilt multiplier*rand(1,zernikes-3)]';
% weight = [piston tip tilt]';
%
% sub_pupil_zern = zeros(C,C,zernikes);
% temp = zeros(C);
% strength=zeros(wlength,n_aps);
% for zz = 1:zernikes
%     temp(pupil_mask) = w(:,zz);
%     sub_pupil_zern(:,:,zz) = temp;
%     % strength(:,zz) = strength + w(:,zz)*weight(zz);
% end
% weight = zeros(n_aps,zernikes);
% weight(:,1) = rand(n_aps,1);
% weight(:,2:3) = 5*rand(n_aps,2);
% weight(:,4:zernikes) = multiplier*rand(n_aps,zernikes-3);
% for ap = 1:n_aps % This double loop adds the same aberrations
%    % This double loop adds the same aberrations
%     for zz = 1:zernikes
%         wc = weight(ap,zz);
%         strength(:,ap) = strength(:,ap) + w(:,zz)*wc;
%     end
%     for SR = 1:snaps
%         for ill = 1:n_ill
%             cur = ap + n_aps*(ill - 1); % Selects each field
%             % pupil_data(:,cur,1) = pupil_data(:,cur,1)...
%% 3rd order and higher zernike removal

% overlaps = 3;
% al=zeros(zernikes-3,n_aps);
% for ii = 1:n_aps
%     ii2 = ii+n_aps;    ii3 = ii+2*n_aps;
%     ap_nums = [ii ii2 ii3];%         [3 4]     [1 2 3]
%     [alph,~,~,~,num_sec] = ...
% closed_form_solve2(pupil_mask,pupil_data,C,subap_x_cen,subap_y_cen,ap_nums,1,zernikes,sub_pupil_zern,overlaps);
%     al(:,ii)= alph(num_sec+1:num_sec+zernikes-3);
%     remove = 0;
%     for zz = 4:zernikes
%         remove = remove + w(:,zz)*alph(num_sec+zz-3);
%     end
%     for SR = 1:snaps
%         for ap = ii:n_aps:num_pup
%             pupil_data(:,ap,SR) = pupil_data(:,ap,SR)...
%                 .*exp(-1i*2*pi*remove);
%         end
%     end
% al2=zeros(zernikes-3,n_aps);
% for ii = 1:n_aps
%     ii2 = ii+n_aps;    ii3 = ii+2*n_aps;
%     ap_nums = [ii ii2 ii3];%         [3 4]     [1 2 3]
%     [alph,~,~,~,num_sec] = ...
% closed_form_solve2(pupil_mask,pupil_data,C,subap_x_cen,subap_y_cen,ap_nums,1,zernikes,sub_pupil_zern,overlaps);
%     al2(:,ii)= alph(num_sec+1:num_sec+zernikes-3);
%     remove = 0;
%     for zz = 4:zernikes
%         remove = remove + w(:,zz)*alph(num_sec+zz-3);
%     end
%     for SR = 1:snaps
%         for ap = ii:n_aps:num_pup
%             pupil_data(:,ap,SR) = pupil_data(:,ap,SR)...
%                 .*exp(-1i*2*pi*remove);
%         end
%     end
% al=al'+al2';
% results = weight(:,4:zernikes)-al
% %% Piston tip tilt removal
% %area = zeros(n_aps-1,1); pistonc = zeros(n_aps-1,1); tipc = zeros(n_aps-1,1); tiltc = zeros(n_aps-1,1);
% overlaps = 1;
% [num_rows,~] = size(array_info);

%% phasing between rows

% if num_rows > 1
updown = flipud(sortrows(array_info,2));
%  This section takes the farthest left pupil in the non extended res
%  and
%  % phases the overlapping pupils from the rows directly above and below
%  for ii = 2:3
%    if updown(1,1) < updown(ii,1)
%      ap1 = updown(1,1)+2*n_aps;
%      ap2 = updown(ii,1)+n_aps;
%    else
%      ap1 = updown(1,1)+n_aps;
%      ap2 = updown(ii,1)+2*n_aps;
%    end
%    ap_nums = [ap1 ap2];
%  % [3 4] [1 2 3]
%  [alph,tipc,tiltc,area,num_sec] = ...
% closed_form_solve2(pupil_mask,pupil_data,C,subap_x_cen,subap_y_cen,ap_nums,1,zernikes,sub_pupil_zern,overlaps);
% pistonc = alph(1);
% for SR = 1:snaps
% for ap = updown(ii,1):n_aps:num_pup
%   pupil_data(:,ap,SR) = pupil_data(:,ap,SR)...
% .*exp(1i*2*pi*(w(:,1)*pistonc+w(:,2)*tipc+w(:,3)*tiltc));
% end
end
% This section phases the furthest left pupils from different rows to
% the one further out from the center, or longest row. This starts
% with the second to third rows.
% for ii = 2:2:num_rows-2
% for level = 0:1
%   if level == 0
%     ap1 = updown(ii,1)+2*n_aps;
%     ap2 = updown(ii+2,1)+n_aps;
%   else
%     ap1 = updown(ii+level,1)+n_aps;
%     ap2 = updown(ii+level+2,1)+2*n_aps;
%   end
%   ap_nums = [ap1 ap2];
% % [3 4] [1 2 3]
% [alph,tipc,tiltc,area,num_sec] = ...
% closed_form_solve2(pupil_mask,pupil_data,C,subap_x_cen,subap_y_cen,ap_nums,1,zernikes,sub_pupil_zern,overlaps);
% pistonc = alph(1);
for SR = 1:snaps
    for ap = updown(ii+level,1):n_aps:num_pup
        pupil_data(:,ap,SR) = pupil_data(:,ap,SR)...
.*exp(1i*2*pi*(w(:,1)*pistonc+w(:,2)*tipc+w(:,3)*tiltc));
    end
end
end

% phasing rows
for row = 1 : num_rows
    start = array_info(row,1);
    stop = array_info(row,1)+array_info(row,2)-2;
    for ii = start:stop
        ii2 = ii+n_aps+1;
        ap_nums = [ii ii2]; % [3 4] [1 2 3]
        [alph,tipc(ii),tiltc(ii),area(ii),num_sec] = ...
            closed_form_solve2(pupil_mask,pupil_data,C,subap_x_cen,subap_y_cen,ap_nums,1,zernikes,sub_pupil_zern,overlaps);
        pistonc(ii) = alph(1);
        for SR = 1:snaps
            for ap = ii+1:n_aps:num_pup
                pupil_data(:,ap,SR) = pupil_data(:,ap,SR)...
.*exp(1i*2*pi*(w(:,1)*pistonc(ii)+w(:,2)*tipc(ii)+w(:,3)*tiltc(ii)));
            end
        end
    end
% phasing
x = dx.*n;
 y = dx.*m;
[x,y] = meshgrid(x,y);                % Cartesian grid in physical units

% min_range_for_digital_focus_meters = (x(1,1).^2 - x(1,2).^2)/lambda % Minimum target distance for digital focusing
focus_factor = exp(1i*pi/(lambda*L).*((x.^2 + y.^2)); % Focusing term required for target at finite distance
clear x y
C = 2*R_pix;

%%% Analytic MTF

pupil = zeros(M,N);
pupil_3 = zeros(M,N);
sub_pupil = zeros(C);
sub_pupil(pupil_mask) = 1;
for cl = 1:num_pup
    pupil(M/2+1-C/2+subap_y_cen(cl):M/2+C/2+subap_y_cen(cl),N/2+1-C/2+subap_x_cen(cl):N/2+C/2+subap_x_cen(cl)) = ...
        pupil(M/2+1-C/2+subap_y_cen(cl):M/2+C/2+subap_y_cen(cl),N/2+1-C/2+subap_x_cen(cl):N/2+C/2+subap_x_cen(cl)) + sub_pupil;
    if cl <= num_pup - n_aps
        pupil_3(M/2+1-C/2+subap_y_cen(cl):M/2+C/2+subap_y_cen(cl),N/2+1-C/2+subap_x_cen(cl):N/2+C/2+subap_x_cen(cl)) = ...
        pupil_3(M/2+1-C/2+subap_y_cen(cl):M/2+C/2+subap_y_cen(cl),N/2+1-C/2+subap_x_cen(cl):N/2+C/2+subap_x_cen(cl)) + sub_pupil;
pupil_3(M/2+1-
C/2+subap_y_cen(cl):M/2+C/2+subap_y_cen(cl),N/2+1-
C/2+subap_x_cen(cl):N/2+C/2+subap_x_cen(cl))+sub_pupil;
end
end

[Ny,Nx] = size(pupil);
if Ny < Nx  %Check to see which dimension is larger to make a square
array
    Ny = Nx;
else
    Nx = Ny;
end

nfy = Ny*2+1;         nfx = Nx*2+1;
Ipsf=abs(fftshift(fft2(pupil,nfy,nfx))).^2; % Intensity point spread function
otf=fftshift(ifft2(Ipsf));  %optical transfer function
m_otf=max(max(abs(otf))));  %max of otf
mtf=abs(otf)/m_otf;
Ipsf=abs(fftshift(fft2(pupil_3,nfy,nfx))).^2; % Intensity point spread function
otf=fftshift(ifft2(Ipsf));  %optical transfer function
m_otf=max(max(abs(otf))));  %max of otf
mtf_3=abs(otf)/m_otf;

mask = pupil > 1;   %Finds where pupils overlap
pupil_norm = pupil;
pupil_norm(mask) = pupil(mask)./pupil(mask);
Ipsf=abs(fftshift(fft2(pupil_norm,nfy,nfx))).^2; % Intensity point spread function
otf=fftshift(ifft2(Ipsf));  %optical transfer function
m_otf=max(max(abs(otf))));  %max of otf
mtf_norm=abs(otf)/m_otf;

mask_3 = pupil_3 > 1;   %Finds where pupils overlap
pupil_3_norm = pupil_3;
pupil_3_norm(mask_3) = pupil_3(mask_3)./pupil_3(mask_3);
Ipsf=abs(fftshift(fft2(pupil_3_norm,nfy,nfx))).^2; % Intensity point spread function
otf=fftshift(ifft2(Ipsf));  %optical transfer function
m_otf=max(max(abs(otf))));  %max of otf
mtf_3_norm=abs(otf)/m_otf;

fx = (-Nx:Nx)/(dx*Nx); %Spatial frequency lable
fy = (-Ny:Ny)/(dx*Ny);
hor_mtf = mtf(Ny+1,Nx+1:nfx);
hor_mtf_norm = mtf_norm(Ny+1,Nx+1:nfx);
hor_mtf_3 = mtf_3(Ny+1,Nx+1:nfx);
hor_mtf_3_norm = mtf_3_norm(Ny+1,Nx+1:nfx);
figure
plot(fx(Nx+1:nfx),hor_mtf_3,fx(Nx+1:nfx),hor_mtf_3_norm)
axis([0 max(fx) 0 1])
title('Normalized Pupil MTF Comparison (3)')
figure
plot(fx(Nx+1:nfx),hor_mtf,fx(Nx+1:nfx),hor_mtf_norm)
axis([0 max(fx) 0 1])
title('Normalized Pupil MTF Comparison (4)')
figure
vert_mtf = mtf(Ny+1:nfy,Nx+1);
plot(fy(Ny+1:nfy),vert_mtf)
axis([0 max(fx) 0 1])
title('Vertical MTF')
clear Ipsf otf mtf mtf_norm
%%%%% Propagate unaberrated pupil data to the image plane with focus factor
%%%%% BEST CASE SCENARIO %%%%%
specksum = zeros(M,N);specksum_norm = zeros(M,N);
sub_specksum = zeros(M,N);
% piston=[0,1/12];
for count = 1:snaps,
    composite_pupil = zeros(M,N);
    for c1 = 1:num_pup
        sub_pupil = zeros(C);
        sub_pupil(pupil_mask) = pupil_data(:,c1,count);
        composite_pupil(M/2+1-
            C/2+subap_y_cen(c1):M/2+C/2+subap_y_cen(c1),N/2+1-
            C/2+subap_x_cen(c1):N/2+C/2+subap_x_cen(c1)) = ...
            composite_pupil(M/2+1-
            C/2+subap_y_cen(c1):M/2+C/2+subap_y_cen(c1),N/2+1-
            C/2+subap_x_cen(c1):N/2+C/2+subap_x_cen(c1))+sub_pupil;
    end
    image_plane =
        fftshift(fft2(fftshift(composite_pupil.*focus_factor)));
    specksum = specksum + (abs(image_plane)).^2;
    composite_pupil(mask)=composite_pupil(mask)./pupil(mask);
    image_plane =
        fftshift(fft2(fftshift(composite_pupil.*focus_factor)));
    specksum_norm = specksum_norm + (abs(image_plane)).^2;
    single_pupil = zeros(M,N);
    single_pupil(M/2+1-
            C/2+subap_y_cen(c1):M/2+C/2+subap_y_cen(c1),N/2+1-
            C/2+subap_x_cen(c1):N/2+C/2+subap_x_cen(c1)) = sub_pupil;
    sub_image_plane =
        fftshift(fft2(fftshift(single_pupil.*focus_factor)));
    sub_specksum = sub_specksum + (abs(sub_image_plane)).^2;
end
sharp_weight = 2;
sharpness = sum(sum(specksum.^sharp_weight));
[ill4,coords] = getroi(specksum);
ill4_norm = specksum_norm(coords(2):coords(4),coords(1):coords(3));
illap1 = sub_specksum(coords(2):coords(4),coords(1):coords(3));
[~, dat4, nn2out] = sfrmat3(1, dx, [],ill4);
[~, dat4_norm] = sfrmat3(1, dx, [],ill4_norm);
[~, datap1] = sfrmat3(1, dx, [],illap1);
figure
imagesc(1000*dx*n,1000*dx*m,angle(composite_pupil));
colormap gray
axis equal
title('Composite pupil phase (one speckle realization, dimensions in mm)')
horizontal_pixel_pitch = lambda*f/(N*dx);
vertical_pixel_pitch = lambda*f/(M*dx);
imagesc(1000*horizontal_pixel_pitch*n,1000*vertical_pixel_pitch*m,abs(image_plane).^2)
zoom(2)
colormap gray
axis square
title('Composite image intensity (one speckle realization, dimensions in mm)')
figure
imagesc(n,m,specksum_norm)
zoom(2)
colormap gray
axis square
title('Speckle average composite normalized image (4)')
figure
imagesc(n,m,specksum)
zoom(2)
colormap gray
axis square
title('Speckle average composite image (4)')
figure
imagesc(1000*horizontal_pixel_pitch*n,1000*vertical_pixel_pitch*m,sub_specksum)
zoom(2)
colormap gray
axis square
title('Speckle average single aperture image (dimensions in mm)')
specksum = zeros(M,N);
specksum_norm = zeros(M,N);
sub_specksum = zeros(M,N);
% piston=[0,1/12];
for count = 1:snaps,
    composite_pupil = zeros(M,N);
    composite_pupil_3 = zeros(M,N);
    for c1 = 1:num_pup - n_aps
        sub_pupil = zeros(C);
        sub_pupil(pupil_mask) = pupil_data(:,c1,count);
        composite_pupil(M/2+1-C/2+subap_y_cen(c1):M/2+C/2+subap_y_cen(c1),N/2+1-C/2+subap_x_cen(c1):N/2+C/2+subap_x_cen(c1)) = ...
            composite_pupil(M/2+1-C/2+subap_y_cen(c1):M/2+C/2+subap_y_cen(c1),N/2+1-C/2+subap_x_cen(c1):N/2+C/2+subap_x_cen(c1)) + sub_pupil;
        if mod(c1,n_aps) == 0
            composite_pupil_3(M/2+1-C/2+subap_y_cen(c1):M/2+C/2+subap_y_cen(c1),N/2+1-C/2+subap_x_cen(c1):N/2+C/2+subap_x_cen(c1)) = ...
                composite_pupil_3(M/2+1-C/2+subap_y_cen(c1):M/2+C/2+subap_y_cen(c1),N/2+1-C/2+subap_x_cen(c1):N/2+C/2+subap_x_cen(c1)) + sub_pupil;
        end
    end

image_plane = fftshift(fft2(fftshift(composite_pupil.*focus_factor)));
specksum = specksum + (abs(image_plane)).^2;
composite_pupil(mask_3)=composite_pupil(mask_3)./pupil_3(mask_3);
image_plane =
    fftshift(fft2(fftshift(composite_pupil.*focus_factor)));  
    specksum_norm = specksum_norm + (abs(image_plane)).^2;  
    sub_image_plane =
    fftshift(fft2(fftshift(composite_pupil_3.*focus_factor)));  
    sub_specksum = sub_specksum + (abs(sub_image_plane)).^2;
end
ill3 = specksum(coords(2):coords(4),coords(1):coords(3));  
ill3_norm = specksum_norm(coords(2):coords(4),coords(1):coords(3));  
ill3ap1 = sub_specksum(coords(2):coords(4),coords(1):coords(3));  
[~, dat3] = sfrmat3(1, dx,[],ill3);  
[~, dat3_norm] = sfrmat3(1, dx,[],ill3_norm);  
[~, dat3ap1] = sfrmat3(1, dx,[],ill3ap1);
figure
imagesc(n,m,specksum)  
zoom(2)  
colormap gray  
axis square  
title('Speckle average composite image (3)')
figure
imagesc(n,m,specksum_norm)  
zoom(2)  
colormap gray  
axis square  
title('Speckle average composite normalized image (3)')
figure
imagesc(n,m,sub_specksum)  
zoom(2)  
colormap gray  
axis square  
title('Speckle average single ap composite image (dimensions in pixels)')  
freq = dat4(:,1);
figure
plot(dat3(:,1),dat3(:,2),dat4(:,1),dat4(:,2),datap1(:,1),datap1(:,2),fx(Nx+1:nfx),hor_mtf,fx(Nx+1:nfx),hor_mtf_3,dat3ap1(:,1),dat3ap1(:,2))  
axis([0 freq(round(0.75*nn2out)),0,max(max(dat4(:,2)))]);
title('MTF')  
xlabel('Cycles / m')
figure
plot(dat3(:,1),dat3(:,2),dat3_norm(:,1),dat3_norm(:,2),fx(Nx+1:nfx),hor_mtf_3,fx(Nx+1:nfx),hor_mtf_3_norm)  
axis([0 freq(round(0.75*nn2out)),0,max(max(dat4(:,2)))]);
title('MTF(3)')  
xlabel('Cycles / m')
figure
plot(dat4(:,1),dat4(:,2),dat4_norm(:,1),dat4_norm(:,2),fx(Nx+1:nfx),hor_mtf,fx(Nx+1:nfx),hor_mtf_norm)  
axis([0 freq(round(0.75*nn2out)),0,max(max(dat4(:,2)))]);
title('MTF(4)')  
xlabel('Cycles / m')
figure
plot(dat3_norm(:,1),dat3_norm(:,2),fx(Nx+1:nfx),hor_mtf_3_norm)  
axis([0 freq(round(0.75*nn2out)),0,max(max(dat4(:,2)))]);
title('MTF(3) Normalized')  
xlabel('Cycles / m')
plot(dat4_norm(:,1),dat4_norm(:,2),fx(Nx+1:nfx),hor_mtf_norm).

axis([0 freq(round(0.75*nn2out)),0,max(max(dat4(:,2)))])).
title('MTF(4) Normalized')
xlabel('Cycles / m')
figure
plot(dat3(:,1),dat3(:,2),fx(Nx+1:nfx),hor_mtf_3).

axis([0 freq(round(0.75*nn2out)),0,max(max(dat4(:,2)))])).
title('MTF(3)')
xlabel('Cycles / m')
figure
plot(dat4(:,1),dat4(:,2),fx(Nx+1:nfx),hor_mtf).

axis([0 freq(round(0.75*nn2out)),0,max(max(dat4(:,2)))])).
title('MTF(4)')
xlabel('Cycles / m')

subap_centers_ill3.m. This code creates the sub-aperture centers used in Propagation simulation.

function
[ap_cen_x,ap_cen_y,array_info]=subap_centers_ill3(Array_Type,dx,cs,n_aps,Num,D_phys,ex_ill,theta)

%% subap_centers %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function creates two vectors that contain the center pixels for
% either a linear or hex array. Output vectors are in pixels, but
% based on
% physical dimensions
% Inputs:
% Array_Type        Chooses which array type is desired. Must be
% 'LINEAR'            or 'HEX'
% dx                Grid increment [m]
% cs                Center to center spacing [m]
% n_aps             Desired number of sub-apertures in linear array.
% Redefined
% in hex array
% Num               Center number of sub-aps in hex array, or used for
% determining center of triangular coords
% D_phys            Physical size of aperture
% ex_ill            Extra illuminators ****IF NONE use []**** Should
% be
% (x) in the second
% theta             Rotation angle of 3 illuminator cluster. 0 is 1
dot
%                  center to the left of two dots on a verticle
% line.
%                  30 is two dots on a horizontal line with one
%                  centered underneath
% Outputs:
% ap_cen_x       Center pixel values in the x dimension for effective
% pupils [Integer values around zero]
% ap_cen_y       Center pixel values in the y dimension for effective
% pupils [Integer values around zero]
% array_info     column 1 is the ap number for the first ap in the row. Column 2 is the number of aps in the row. Number of rows is the number of rows in the hex array.
%n_ill,~]=size(ex_ill);  % gets the number of extra illuminators
cluster = 3;                % Sets the number in the illuminator cluster. The code is not set up to change this number
n_ill = cluster + n_ill;    % Sets the total number of illuminators
r_cluster = D_phys / (2*sqrt(3)); % physical dimensions illuminator cluster radius
%% Sub-ap centers
Nfinal=(Num+1)/2;  % Number of apertures in the shortest row

al=cs*[1 0];  %Transfer vector from triangular coords.
a2=cs*[.5 (3^.5)/2];  %Transfer vector from triangular coords.
array_info = zeros(Num,2);  %column 1 is the ap number for the first ap in the row. Column 2 is the number of aps in the row. Number of rows is the number of rows in the hex array.
subap_x_cen= zeros(1,n_aps);    % X center pixel value
subap_y_cen = zeros(1,n_aps);   % y center pixel value
count = 1;   %Index for center
switch Array_Type
    case 'LINEAR'
        z = n_aps - Nfinal; % Works to create right spacing for sub_aps from triangular coords
        v = Num - n_aps;   % Used for creating right spacing for sub_aps for triangular coords
        for u = Nfinal - Num: z
            coords = a1*u+a2*v;  %Vector transmormation from triangular coordinates
            subap_x_cen(count) = round(coords(1,1)/dx);
            count = count + 1;
        end
        array_info = [1,n_aps];
    case 'HEX'
        NTotal=Num+2*sum(Nfinal:(Num-1));   %Total number of apertures
        n_aps = NTotal;
        center=zeros(NTotal,2);
        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(count,:)=a1*u+a2*v;
                    center(count,1)=round(center(count,1)/dx);
                    center(count,2)=round(center(count,2)/dx);
                    count=count+1;
                end
                num_row = i-1+Nfinal;  % number of aps in row
                first_ap = count-num_row;   % number of first ap in row
                array_info(i,:) = [first_ap,num_row];
            else  %bottom half of hex array
                num_row = i-1+Nfinal;  % number of aps in row
                first_ap = count-2*num_row;   % number of first ap in row
                array_info(i,:) = [first_ap,num_row];
            end
        end
end
for k=-Num+i:Nfinal-1
    u=k;
    center(count,:)=a1*u+a2*v;
    center(count,1)=round(center(count,1)/dx);
    center(count,2)=round(center(count,2)/dx);
    count=count+1;
end
num_row = Nfinal+Num-i;  % number of aps in row
first_ap = count-num_row;   % number of first ap in row
array_info(i,:) = [first_ap,num_row];
end

subap_x_cen=center(:,1)';
subap_y_cen=center(:,2)';
otherwise
    error('subap_centers_ill2:Array_Type','Array_Type must be a string of either LINEAR or HEX ')
end

% x_cen_span = max(subap_x_cen) - min(subap_x_cen);

cxill = zeros(n_ill,1);     % Illuminator x coord
cyill = zeros(n_ill,1);     % illuminator y coord
for count = 1 : cluster  % generates pixel values for the 3 illuminator cluster
    cxill(count) = round(r_cluster * cosd(theta + 120*(count - 1))/dx);
    %Gets x pix value
    cyill(count) = round(r_cluster * sind(theta + 120*(count - 1))/dx);
    %Gets y pix value
end
for ill = cluster + 1:n_ill
    cxill(ill) = ex_ill(ill-cluster,2);
    cyill(ill) = ex_ill(ill-cluster,1);
end
n_ps = n_aps*n_ill;     %Number of pupils
%(Effective number of sub_aps based on actual sub-aps and number of illuminators)
ap_cen_x = zeros(n_ps,1);   % x center of each pupil field for all illuminator recieve aperture pair
ap_cen_y = zeros(n_ps,1);   % y center of each pupil field for all illuminator recieve aperture pair
for p = 1 : n_aps  %pupil center pixels due to first illuminator
    ap_cen_x(p) = subap_x_cen(p);
    ap_cen_y(p) = subap_y_cen(p);
end
for p = 2 : n_ill  %cycles through remaining illuminators to get pupil centers
difx = cxill(p) - cxill(1); % x coordinate transform from changing ill
dify = cyill(p) - cyill(1); % y coordinate transform from changing ill
for q = 1 : n_aps  %cycles through sub-aps to get pupil center pixels
    ap_cen_x(n_aps*(p-1)+q) = subap_x_cen(q) + difx;
    ap_cen_y(n_aps*(p-1)+q) = subap_y_cen(q) + dify;
end
end
ap_cen_x = ap_cen_x';
ap_cen_y = ap_cen_y';

closed_form_solve2.m. This runs the closed form solve for both intra and inter-aperture phasing.

function [alph,xwaves,ywaves,area,num_sec] =
closed_form_solve2(pupil_mask,pupil_data,C,subap_x_cen,subap_y_cen,ap_nums,SR,zernikes,sub_pupil_zern,overlaps)
aps_used = length(ap_nums);  % number of aps being overlapped
if overlaps == 1  % 1 overlap is used for inter aperture phasing
    zern_used = 3;  % only 3 Zernikes are used with inter aperture phasing
    zern_start = 1;  % piston is first Zernike
    zern_stop = 3;   % having 3 Zernike terms gives piston tip and tilt
else
    zern_used = zernikes - 3;  %Any other number of overlap will be used for intra aperture phasing
    zern_start = 4;   % Piston tip and tilt cannot be corrected for in interaperture phasing so we start at the 4th Zernike
    zern_stop = zernikes;   % stop at the highest Zernike specified
end
sub_pupil = zeros(C,C,aps_used);  %This array holds the circular pupils. the ey are stored in a column vector in pupil_data
xcen = zeros(1,aps_used);  ycen = zeros(1,aps_used);  % x and y aperture center values this is only for the apertures that are being phased together.
for ii = 1:aps_used  % fills sub_pupil xcen and ycen
    temp = zeros(size(pupil_mask)); temp(pupil_mask) = pupil_data(:,ap_nums(ii),SR);  % converts column vector into circular pupil
    sub_pupil(:,:,ii) = temp;
    xcen(ii) = subap_x_cen(ap_nums(ii));  % assigns center pixel values
    ycen(ii) = subap_y_cen(ap_nums(ii));  % assigns center pixel values
end
shiftx = min (xcen);    shifty = min (ycen);  % These are used to make sure the pixel values fit within the array mat
xdif =zeros(1,overlaps);    ydif =zeros(1,overlaps);  % these store the information for how big mat needs to be. These are the differences between center pixel values
for ii = 1:overlaps  % defines which diffence to take based on the number of overlaps.
    switch ii
        case 1
            xdif(ii) = abs(xcen(1)-xcen(2));    ydif(ii) = abs(ycen(1)-ycen(2));
        case 2
            xdif(ii) = abs(xcen(1)-xcen(3));    ydif(ii) = abs(ycen(1)-ycen(3));
        case 3
xdif(ii) = abs(xcen(3)-xcen(2));  ydif(ii) = abs(ycen(3)-ycen(2));

otherwise
    error('closed_form_solve:overlaps','Switch statement error in finding center coords difference. FIX SOMETHING')
end
end
difx = max(xdif);  dify = max(ydif);  % finds the max number of
dif to create the mat array so it is large enough for all the different
apertures being phased
mat = zeros(dify+C,difx+C,aps_used);  %where the initial pupil data is
stored and processed
for ii = 1:aps_used  %fills mat with the approriate pupil information
    temp = zeros(dify+C,difx+C);
    temp(ycen(ii)-shifty+1:ycen(ii)-shifty+C,xcen(ii)-shiftx+1:xcen(ii)-shiftx+C) = sub_pupil(:,:,ii);
    mat(:,:,ii) = temp;
end
am = zeros(dify+C,difx+C,aps_used);  %Creates which holds the wavefront
differences
for ii = 1:overlaps  %switches between the diffent overlap cases to
    switch ii
        case 1
            am(:,:,ii) = mat(:,:,1) .*conj(mat(:,:,2));
        case 2
            am(:,:,ii) = mat(:,:,1) .*conj(mat(:,:,3));
        case 3
            am(:,:,ii) = mat(:,:,3) .*conj(mat(:,:,2));
        otherwise
            error('closed_form_solve:overlaps','Switch statement error in finding phase difference. FIX SOMETHING')
    end
    temp = angle(am(:,:,ii)); temp(temp == 0)=9;  %makes anything equal
to zero in am equal 9.  This assists in edge detection
    am(:,:,ii) = temp;  % finishes making am == 0 = 9
end
L = zeros(size(am)); NUM = zeros(1,overlaps);  %creates L which holds
the sectioned off wavefront difference mask areas after edge detection
and NUM which tels how many sections there are
dil = ones(3);  % Creates the dialation kernal
for over = 1:overlaps  %creates a mask for each section of the
    dife = edge(am(:,:,over),'canny','.3');  %edge detection
    dife = imdilate(dife,dil);  %dialation of edge detection to close
gaps, ie make complete sections.
    [L(:,:,over),NUM(over)] = bwlabel(imcomplement(dife));  %labels
each section so it can be easily used to create a mask for operationon
only the specific section
end
ywaves = 0; xwaves=0;  %initializing output variables that contain tip
and tilt
if overlaps == 1  %processing to find piston tip and tilt in waves
max_sec = 0; over_max = 0; sec_max = 0; % initializing
for over = 1 : overlaps %This loop finds the largest section to
use for an average gradient. This for loop is useless because the if
statement only allows overlaps ==1
    for sec = 2 : NUM(over) %starts at 2 because 0 is the outside
        sec_mask = L(:,:,over) == sec; %makes a mask the shape of
        the current section
            num_temp = sum(sum(sec_mask)); %Finds the number of pixels
            equal to 1 in the mask
                if num_temp > max_sec %Checks to see if this mask is the
                    max_sec = num_temp; %if mask is largest, saves the
                    over_max = over; % perpetuates the useless loop
                    sec_max = sec; % saves the number of the largest mask
                end
        end
    temp_sec = L(:,:,over_max)==sec_max; %recreateslargest mask
    temp_field = am(:,:,over_max); temp_field = temp_field .* temp_sec;
    % Wavefront diifference of only masked area
        [xgrad,ygrad] = gradient(temp_field); %Finds the gradient in both
        x and y for entire area
            temp_sec = imcomplement(temp_sec); %
            temp_sec = imdilate(temp_sec,dil); %
            temp_sec = imcomplement(temp_sec); %  Dialates the edge to get
            rid of gradient edge effects
                ywaves = mean(mean(ygrad(temp_sec)))*C/4/pi; xwaves =
                mean(mean(xgrad(temp_sec)))*C/4/pi; %gets average x and y gradients in
waves
sub_pupil(:,:,2)=sub_pupil(:,:,2).*exp(1i*2*pi*(xwaves*sub_pupil_zern(:,
,2)+ywaves*sub_pupil_zern(:,3)))); %removes tip and tilt
    temp = zeros(dify+C,difx+C);
    temp(ycen(2)-shifty+1:ycen(2)-shifty+C,xcen(2)-shiftx+1:xcen(2)-
shiftx+C) = sub_pupil(:,:,2);
    mat(:,:,2) = temp;
    am(:,:,1) = mat(:,:,1) .* conj(mat(:,:,2));
    temp = angle(am(:,:,1)); temp(temp == 0)=9;
    dife = imdilate(dife,dil);
[L(:,:,1),NUM(1)] = bwlabel(imcomplement(dife));
piston = mean(mean(temp(L(:,:,1)==2))); % % there should only be 1 or 2
sections left and the average will be the piston
end
num_sec = 0;
rows_matrix = 0;
for over = 1 : overlaps
    for sec = 2 : NUM(over)
        sec_mask = L(:,:,over) == sec;
        num_temp = sum(sum(sec_mask));
        if num_temp >20;
            rows_matrix = rows_matrix + num_temp;
        end
    end
end
118
num_sec = num_sec+1;
end
end
dk = 0;
rowindx = 0;
dW = zeros(rows_matrix,1);
phasing_matrix = zeros(rows_matrix,num_sec+zern_used);
for over = 1 : overlaps
    for sec = 2 : NUM(over)
        sec_mask = L(:,:,over) == sec;
        if sum(sum(sec_mask)) > 20;
            secindx = secindx + 1;
            temp_zern1 = zeros(dify+C,difx+C,zernikes);
            temp_zern2 = zeros(dify+C,difx+C,zernikes);
            switch over
                case 1
                    temp_zern1(ycen(1)-shifty+1:ycen(1)-shifty+C,xcen(1)-shiftx+1:xcen(1)-shiftx+C,:) = sub_pupil_zern;
                    temp_zern2(ycen(2)-shifty+1:ycen(2)-shifty+C,xcen(2)-shiftx+1:xcen(2)-shiftx+C,:) = sub_pupil_zern;
                case 2
                    temp_zern1(ycen(1)-shifty+1:ycen(1)-shifty+C,xcen(1)-shiftx+1:xcen(1)-shiftx+C,:) = sub_pupil_zern;
                    temp_zern2(ycen(3)-shifty+1:ycen(3)-shifty+C,xcen(3)-shiftx+1:xcen(3)-shiftx+C,:) = sub_pupil_zern;
                otherwise
                    temp_zern1(ycen(3)-shifty+1:ycen(3)-shifty+C,xcen(3)-shiftx+1:xcen(3)-shiftx+C,:) = sub_pupil_zern;
                    temp_zern2(ycen(2)-shifty+1:ycen(2)-shifty+C,xcen(2)-shiftx+1:xcen(2)-shiftx+C,:) = sub_pupil_zern;
            end
            zern_diff = temp_zern1 - temp_zern2;
            sec_vec = zeros (1,num_sec); sec_vec(secindx) = 1;
            temp = L(:,:,over);
            [lr,lc]=find(temp==sec);
            for ind = 1 : length(lr)
                rowindx = rowindx + 1;
                dW(rowindx) = am(lr(ind),lc(ind),over);
                zern_vec = zeros(1,zern_used);
                for zzz = zern_start:zern_stop
                    zern_vec(zzz-zern_start+1) =
                        zern_diff(lr(ind),lc(ind),zzz);
                end
                phasing_matrix (rowindx,:) = [sec_vec,zern_vec];
            end
        end
    end
end
if overlaps ==1
    alph = piston/(2*pi);
else
    alph = phasing_matrix\dW/(2*pi);
end
temp1 = zeros(dify+C,difx+C); temp2 = zeros(dify+C,difx+C);
temp1(ycen(1)-shifty+1:ycen(1)-shifty+C,xcen(1)-shiftx+1:xcen(1)-shiftx+C,:) = pupil_mask;
temp2(ycen(2)-shifty+1:ycen(2)-shifty+C,xcen(2)-shiftx+1:xcen(2)-shiftx+C,:) = pupil_mask;
tempmask = temp1&temp2;
overlap_area = sum(sum(tempmask));
fulla = sum(sum(pupil_mask));
area = overlap_area/fulla;

A.3. Experimental: Wavefront Difference

This program loads image data, parses the text from the image data, LO data, crops out the pupil data, and then computes the wavefront difference. The final section of code was an attempt to extract the fringes from the wavefront difference by filtering the noise from the data. This program uses dftregistration.

cam = 3;
computerpath = 'C:\Users\Jkraczek\Desktop\';%C:\Users\Me\Desktop\C:\Users\Jkraczek\Desktop\ E:\
folderpath = strcat(computerpath,'Large data\17 3 7\transmissive bright\');% Input folder path that folders of data are stored 'E:\Data\Run1\' 'C:\Users\Me\Desktop\Large data\12 22 15\Exp2 felt\'
n_ill = 4; %Number of illuminators
L = 1.1598e3; % Simulated length (approx 10'-4" from eye piece, then subtract the focal length of eye piece(250mm) and multiply by M^2(400))
c_ill = 1; % USER INPUT for which illuminator connects to the next camera
diam = .05; %aperture diameter
up = 3;
upsample = 1000; %upsampling for registration
home = pwd; % Saves the directory you are starting from
cd (folderpath); %Changes directory to folder with data
folders = dir('Cam*'); %Gives listings in directory that start with Cam. Folders in directory named Cam1 Cam2 Cam 3 ...
n_aps = length(folders); % Gives the number of folders that start with Cam
names = {folders.name}; %gets the names of the camera folders
filepath = char(strcat(folderpath,names(ccam))); %path to images within current folder
cd(filepath) %changes path to specific camera data
filelist = dir('*.bmp'); %creates a filist of only the .bmp files within the current directory specifically all images from a specific camera
data=imread (filelist(2).name); % loads an image
[rows,cols]=size(data); % Finds the size of the image
LO_path = strcat(computerpath,'Large data\LO 17 3 7\');
cd (LO_path);
LO_list = dir('*.bmp');
nLO = length (LO_list);
LO_AVG = zeros(size(imread(LO_list(1).name)));
for ii = 1:nLO
LO_AVG = LO_AVG + double(imread(LO_list(ii).name));
end
LO_AVG = LO_AVG / nLO;
da = 5.86e-6;       %Pixel pitch of Camera
lambda = 532e-9;       %Wavelength of light
% f = 499.9194e-3;        %Back focal distance of lens system
f = 1252.309e-3;        %Focal length of lens system
r_pix = 422;  %up = 3,422  up = 2,283 up = 1,141       %Radius in pixels
of the crop that is captured in the pupil plane
dx = diam / (2*r_pix);
pupil_shift = .0254 / dx;
r_reduce = 5;       %reduction in radius used to correlate a binary
circle with the pupil field in the pupil plane for automated location
of pupils independant of telescope design
r_quad = ceil((up*cols-10)/4);       %Sidelength in pixels for smaller
region to look for pupils
[X,Y]=meshgrid(-r_quad:r_quad);
circ = X.^2 + Y.^2 <= (r_pix-r_reduce)^2; %Binary circle used for cross
 correlation with pupil
pupil_mask = X.^2 + Y.^2 <= (r_pix)^2; %Binary circle used for saving
cropped pupil data
[M,N] = meshgrid(-up*rows/2:up*rows/2-1);
num_pup = 4;
signal = up*8;
ssgaus = 1 - exp(-(M.^4+N.^4)/signal^n);    %Super Gaussian
freq = fftshift(fft2(data,up*cols,up*rows));    %freq = freq .*ssgaus;
freqs = (freq(1:2*r_quad+1,1:2*r_quad+1));      %Pupil in upper left
quadrant
figure; imagesc(log(abs(freq).^2));  colormap gray;  axis image
figure; imagesc(log(abs(freqs).^2));  colormap gray;  axis image
SR = length(filelist) / n_ill;
num_pup = 4;
pupil_data = zeros(sum(pupil_mask(:)),num_pup,SR);       % Pupil plane
data with sub-apertures cropped out
filepath = char(strcat(folderpath,names(ccam)));  %File path for the
desired camera
filelist = dir('*.bmp');  %Changes path to the desired camera.
Adds originating folder to use functions
filelist = dir('*.bmp');
%     filename = char(strcat(filepath,filelist(2).name));
for jj = 1 : length(filelist)  %For loop to read each image in camera
folder,
% %This section searches through the file names to pull out
% camera number, speckle realization or set number, and switch
% position number
cam = strfind(filelist(jj).name, 'Cam');
set = strfind(filelist(jj).name, 'Set');
switch = strfind(filelist(jj).name, 'switch');
dot = strfind(filelist(jj).name, '.');
ncam = str2num(filelist(jj).name(cam + 3 : set-1));
nloop = str2num(filelist(jj).name(set + 3 : switch-1));
nswitch = str2num(filelist(jj).name(switch + 7 : dot-1));
% %End file identification section
if nswitch(ii) == 1
    if nswitch == 3 || nswitch == 4
        data = double(imread (filelist(jj).name)) - LO_AVG;  %load
image
121
freq = fftshift(fft2(data,up*cols,up*rows)); %Fourier transform to pupil plane
defreqs = (freq(1:2*r_quad+1,1:2*r_quad+1)); %Pupil in upper left quadrant
[output,~] = dftregistration(fft2(abs(freqs)),fft2(circ),1); %performs cross correlation. outputs the shift between the centered circle and the pupil
dif = output([3,4])'; %shift from correlation in pixels
mask = (X-dif(2)).^2 + (Y-dif(1)).^2 <= (r_pix)^2; %Creates a shifted circle mask to crop out the pupil
data1 = zeros(size(mask));
data1(mask) = freqs(mask);

defigure
%                         imagesc(abs(data1).^2)
%                         imagesc(angle(data1))
%                         imagesc(log(abs(data1).^2))
%                         imagesc(abs(freq.*ssgaus).^2)
%                         imagesc(abs(log(freq)))
%                         axis image
%                         colormap gray
%                         title(filelist(jj).name)
%                         drawnow
%                         pause(.75)
data1(:,nswitch,nloop) = freqs((X-dif(2)).^2 + (Y-dif(1)).^2 <= (r_pix)^2);
end

%% Preparing pupil data for registration
ext = 2; % Extention beyound anticipated size of apeture for shifting due to registration
C = 2*(r_pix + ext)+1;
[X,Y]=meshgrid(-r_pix-ext:r_pix+ext);
pupil_mask = X.^2 + Y.^2 <= (r_pix)^2;
[nrows,~]=size(pupil_mask);
Nrows = ifftshift(-fix(nrows/2):ceil(nrows/2)-1);
[Ncols,Nrows] = meshgrid(Nrows);
sub_pupil1 = zeros(nrows,nrows,SR); %This array holds the circular pupils. they are stored in a column vector in pupil_data
sub_pupil2 = zeros(nrows,nrows,SR); %This array holds the circular pupils. they are stored in a column vector in pupil_data
Loval = (X + pupil_shift).^2 + Y.^2 <= (r_pix - 2*r_reduce)^2; % Loval = (X + pupil_shift).^2 + Y.^2 <= (r_pix - 2*r_reduce)^2;
Roval = (X - pupil_shift).^2 + Y.^2 <= (r_pix - 2*r_reduce)^2;
sub_pupil1(:,3,ii) = temp.*Loval; % converts column vector into circular pupil
sub_pupil2(:,4,ii) = temp.*Roval;
end

%% Intensity Registration
difffx = zeros (1,SR);
difffy = zeros (1,SR);
for kk = 1:SR

\[ \text{register two real Inten} (\text{sub_pupil1}(::,kk), \text{sub_pupil2}(::,kk), \text{upsample}, \text{Nrows}, \text{Ncols}, \text{nrows}); \]
\[ \text{difffy}(kk) = \text{difff}(1); \]
\[ \text{difffx}(kk) = \text{difff}(2); \]
\end{verbatim}

\begin{verbatim}
xx = 1:SR;
figure
plot(xx,difffy,xx,difffx)
\end{verbatim}

\begin{verbatim}
%% Field Registration
difffy = round(difffy);    difffx = round(difffx);
for ii = 1:SR  % fills sub_pupil xcen and ycen
    Loval = (X + (difffx(ii))).^2 + (Y + (difffy(ii))).^2 <= (r_pix - 2*r_reduce)^2;    Roval = (X - (difffx(ii))).^2 + (Y - (difffy(ii))).^2 <= (r_pix - 2*r_reduce)^2;
    Loval = Loval & pupil_mask;   Roval = Roval & pupil_mask;
    temp = zeros(size(pupil_mask)); temp(pupil_mask) = pupil_data(:,3,ii);  % converts column vector into circular pupil
    sub_pupil1(:,:,ii) = temp.*Loval;
    temp = zeros(size(pupil_mask)); temp(pupil_mask) = pupil_data(:,4,ii);  % converts column vector into circular pupil
    sub_pupil2(:,:,ii) = temp.*Roval;
    temp1 = zeros(3*nrows,3*nrows);
    temp2 = zeros(3*nrows,3*nrows);
    temp1(nrows+1:2*nrows,nrows+1:2*nrows) = sub_pupil1(:,:,ii);
    temp2(nrows+1-1*round(difffy(ii)):2*nrows-1*round(difffy(ii)),nrows+1-1*round(difffx(ii)):2*nrows-1*round(difffx(ii))) = sub_pupil2(:,:,ii);
    [~,difff] = registertwoerealuneven(temp1,temp2,upsample);
    difffy(ii) = difffy(ii) + difff(1);
    difffx(ii) = difffx(ii) + difff(2);
end
xx = 1:SR;
figure
plot(xx,difffy,xx,difffx)

%% Preparing pupil data for Wavefront difference
for ii = 1:SR  % fills sub_pupil xcen and ycen
    Loval = (X + round(difffx(ii))).^2 + (Y + round(difffy(ii))).^2 <= (r_pix - 2*r_reduce)^2;    Roval = (X - round(difffx(ii))).^2 + (Y - round(difffy(ii))).^2 <= (r_pix - 2*r_reduce)^2;
    Loval = Loval & pupil_mask;   Roval = Roval & pupil_mask;
    temp = zeros(size(pupil_mask)); temp(pupil_mask) = pupil_data(:,3,ii);  % converts column vector into circular pupil
    sub_pupil1(:,:,ii) = temp.*Loval;
    temp = zeros(size(pupil_mask)); temp(pupil_mask) = pupil_data(:,4,ii);  % converts column vector into circular pupil
    sub_pupil2(:,:,ii) = temp.*Roval;
end
xx = 1:SR;
figure
plot(xx,difffy,xx,difffx)
\end{verbatim}
```matlab
% figure; imagesc(abs(temp)); axis image; colormap gray
temp = ifft2(fft2(temp) .* exp(1i*2*pi* (rem(difffy(ii),1)*Nrows/nrows + rem(difffx(ii),1)*Ncols/nrows))); % shifts by only the decimal difference
sub_pupil2(:,:,ii) = temp.*Roval;
end

%% Wavefront difference
shiftx = fix(min ([0,difffx])); shifty = fix(min ([0 difffy]));
difx = fix(max(abs([0 difffx]))); diffy = fix(max(abs([0 difffy])))); % finds the max number of dif to create the mat array so it is large enough for all the different apertures being phased
WaveDiff = zeros(3*nrows,3*nrows,SR);
for kk = 1:SR
    diffx = fix(difffx(kk)); diffy = fix(difffy(kk));
    temp1 = zeros(3*nrows,3*nrows);
    temp2 = zeros(3*nrows,3*nrows);
    temp1(nrows+1:2*nrows,nrows+1:2*nrows) = sub_pupil1(:,:,ii);
    temp2(nrows+1-1*round(difffy(ii)):2*nrows-1*round(difffy(ii)),nrows+1-1*round(difffx(ii)):2*nrows-1*round(difffx(ii))) = sub_pupil2(:,:,ii);
    figure
    imagesc(abs(temp1).^2)
    axis image
    colormap gray
    title('ap1')
    figure
    imagesc(abs(temp2).^2)
    axis image
    colormap gray
    title('ap2')
    % WaveDiff(:,:,kk) = angle(temp1 .* conj(temp2));
    % WaveDiff(:,:,kk) = angle(temp1 .* conj(temp2));
    % WaveDiff(:,:,kk) = angle(temp1) - angle(temp2);
    figure
    imagesc(abs(WaveDiff(:,:,kk)).^2)
    imagesc((WaveDiff(:,:,kk)))
    axis image
    colormap gray
end

aWaveDiff = mean(WaveDiff,3);
figure
imagesc(abs(aWaveDiff).^2)
axis image
colormap gray
faWaveDiff = fftshift(fft2(angle(WaveDiff(431-363:431+363,644-363:644+363))));
figure
imagesc((abs(faWaveDiff).^2)); title('Frequency');
axis image
```

124
colormap gray

[rr,cc] = size(faWaveDiff);
lvec = floor(rr/2);
[ccc,rrr] = meshgrid(-lvec:lvec);
RR = sqrt(ccc.^2+rrr.^2);
sig = 60;
nn = 16;
maskfa = exp(-((RR/sig).^nn));
mfaWaveDiff = faWaveDiff.*maskfa;
% mfaWaveDiff = faWaveDiff;
figure
imagesc(maskfa);title('super gaus');
axis image
colormap gray

figure
imagesc(log(abs(mfaWaveDiff).^2));title('Frequency & super gaus');
axis image
colormap gray

FWaveDiff = ifft2(ifftshift(mfaWaveDiff));
figure
imagesc(abs(FWaveDiff));title('Wave diff');
axis image
colormap gray
cd(home)