QUANTITATIVE ANALYSIS OF 3D IMAGES FORMED USING RANGE COMPRESSED HOLOGRAPHY

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

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

By
Thomas Joseph Welsh V

Dayton, Ohio

December 2017
QUANTITATIVE ANALYSIS OF 3D IMAGES FORMED USING RANGE COMPRESSED HOLOGRAPHY

Name: Welsh V, Thomas Joseph

APPROVED BY:

David J. Rabb, Ph.D.
Advisory Committee Chairman
Graduate Faculty
Electro-Optics and Photonics
University of Dayton

Partha P. Banerjee, Ph.D.
Committee Member
Professor and Chair
Electro-Optics and Photonics
University of Dayton

Arnab K. Shaw, Ph.D.
Committee Member
Professor
Electrical Engineering
Wright State University

Robert J. Wilkens, Ph.D., P.E.
Associate Dean for Research and Innovation
Professor
School of Engineering
University of Dayton

Eddy M. Rojas, Ph.D., M.A., P.E.
Dean, School of Engineering
University of Dayton
ABSTRACT

QUANTITATIVE ANALYSIS OF 3D IMAGES FORMED USING RANGE COMPRESSED HOLOGRAPHY

Name: Welsh V, Thomas Joseph
University of Dayton

Advisor: David Rabb, Ph.D.

Range compressed holography is a technique that uses multiple two-dimensional (2D), single wavelength holograms in order to create a range compressed three-dimensional (3D) image of a scene. Typically, these range compressed 3D images are described in terms of system parameters such as SNR and resolution in each of the dimensions. While these quantify some aspects of the resulting 3D data product, the overall performance may only be qualitatively analyzed. A holistic metric is needed that encompasses these system parameters, as well as the nonlinear method of reconstruction of surfaces within volume noise. Representing the images as point clouds allows conventional point cloud metrics to be applied. The metric used is the Point Cloud Library’s fitness score, which calculates the mean squared Euclidean distance between the reconstructed point cloud and the reference point cloud. Two scenes were created. The first, a flat plate was chosen to test range precision only and in the other, a more complex scene was created including a vehicle on a flat surface to account for cross-range resolution impacts on the mean.
squared Euclidean distance. The range variances for surface reconstructions of the flat plate were measured for simulations and experiments and, due to the constant range being independent of cross-range, are equivalent to the mean squared Euclidean distance. The mean squared Euclidean distance was also found for the complex scene through simulations. The simulations and experiments varied signal photons, bandwidth, and speckle realizations to observe the impacts on image quality using a quantitative measurement. The purpose of this thesis is to use the metric to understand how each of the variables impacts image quality and determine the most signal photon efficient way to collect data for range compressed holography.
ACKNOWLEDGEMENTS

I would like to thank Dave Rabb for being my advisor, in addition to providing his patience and expertise throughout the process. Thank you to Dave Rabb, Partha Banerjee, and Arnab Shaw for being a part of my committee and the advice they provided. I would like to thank Jessica Knox and Maureen Crotty for helping edit my thesis. I would like to thank Jason Stafford for all of the additional help, advice, and guidance he provided through my research process. Thank you to Matt Dierking, Larry Barnes, and Andre Van Rynbach for providing helpful commentary through my research process.

This effort was supported in part by the U.S. Air Force through contract number FA8650-12-D-1377 and the University of Dayton Electro-Optics and Photonics Department (EOP). The views expressed in this article are those of the author and do not reflect on the official policy of the Air Force, Department of Defense, or the U.S. Government.
# TABLE OF CONTENTS

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

ACKNOWLEDGEMENTS ........................................................................................................ v

LIST OF FIGURES ................................................................................................................. viii

LIST OF ACRONYMS ............................................................................................................. xi

LIST OF SYMBOLS .............................................................................................................. xiii

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

CHAPTER 2 THEORY ............................................................................................................ 3

2.1. Holography .................................................................................................................. 3

2.2. Variance Bounds for 1D Range Compression ............................................................. 22

2.3. Quantitative Analysis ................................................................................................. 26

CHAPTER 3 SIMULATION METHODOLOGY .................................................................... 29

3.1. Range Compressed Holography Simulation ............................................................... 30

CHAPTER 4 EXPERIMENT .................................................................................................. 38

4.1. Experiment Design ..................................................................................................... 38

4.2. Methodology ............................................................................................................... 40

4.3. Noise Isolation .......................................................................................................... 41

4.4. Post Processing ......................................................................................................... 43

CHAPTER 5 RESULTS AND DISCUSSION ........................................................................ 44
5.1. Signal Photons ............................................................................................................ 44
5.2. Flat Plate Simulation and Experimental Data.......................................................... 45
5.3. Validation of Simulated Data..................................................................................... 56
5.4. Backhoe Results........................................................................................................ 56
5.5. Validation of Quality Metric ................................................................................... 73
CHAPTER 6 CONCLUSION ............................................................................................. 75
6.1. Original Contributions ............................................................................................. 77
6.2. Future Work ............................................................................................................. 78
BIBLIOGRAPHY ................................................................................................................ 79
APPENDIX A MATLAB Code to Simulate the Point Cloud Metric on a Flat Plate
   Target Using Range Compressed Holography............................................................. 83
   A.1. eigencorrection ....................................................................................................... 83
   A.2. objectpcd ................................................................................................................. 83
   A.3. imagepcdone .......................................................................................................... 84
   A.4. savepcd ................................................................................................................... 84
   A.5. Simulation of the Flat Plate Target ......................................................................... 88
LIST OF FIGURES

Figure 1: Twin images of a Fourier transformed hologram in the spatial frequency domain .......................................................... 8

Figure 2: 3D holographic ladar ......................................................................................................................... 11

Figure 3: Processing steps for applying the PGA method to 3D holographic ladar for correction of temporal frequency phase error .................................................................................. 14

Figure 4: Plotting the effects of peak intensity when zero padding is applied ........................................ 21

Figure 5: Simulation Methodology .................................................................................................................. 30

Figure 6: Reference Target Images .................................................................................................................. 31

Figure 7: First step in reconstructing RCH .................................................................................................... 33

Figure 8: A cross-range reconstructed dataset .............................................................................................. 35

Figure 9: Range compression of zero padded frequency dataset ................................................................ 36

Figure 10: Experimental setup diagram ........................................................................................................ 39

Figure 11: Experimental methodology diagram .............................................................................................. 40

Figure 12: The reference point cloud of the flat plate target ........................................................................ 45

Figure 13: Effects of speckle averaging on a flat plate target with 400 MHz bandwidth ......................................................... 47

Figure 14: Effects of speckle averaging on a flat plate target with 800 MHz bandwidth......................................................... 48
Figure 15: Effects of speckle averaging on a flat plate target with 1600 MHz bandwidth. ................................................. 49

Figure 16: Effects of speckle averaging on a flat plate target with 3200 MHz bandwidth. .................................................. 50

Figure 17: Effects of bandwidth on a flat plate target for 1 speckle average. .................. 51

Figure 18: Effects of bandwidth on a flat plate target for two speckle averages. .......... 52

Figure 19: Effects of bandwidth on a flat plate target for four speckle averages. ........ 53

Figure 20: Effects of bandwidth on a flat plate target for eight speckle averages. ........ 54

Figure 21: Effects of bandwidth on a flat plate target for sixteen speckle averages. ..... 55

Figure 22: The reference point cloud for the backhoe target. ...................................... 57

Figure 23: Effects of speckle averages on the backhoe target for 400 MHz bandwidth. 59

Figure 24: Effects of speckle averages on the backhoe target for 800 MHz bandwidth. 60

Figure 25: Effects of speckle averages on the backhoe target for 1600 MHz bandwidth. 61

Figure 26: Effects of speckle averages on the backhoe target for 3200 MHz bandwidth. 62

Figure 27: Effects of speckle averages on the backhoe target for 6400 MHz bandwidth. 63

Figure 28: Effects of speckle averages on the backhoe target for 12800 MHz bandwidth. 64

Figure 29: Effects of bandwidth on the backhoe target for one speckle average. .......... 66

Figure 30: Effects of bandwidth on the backhoe target for two speckle averages. ....... 67
Figure 31: Effects of bandwidth on the backhoe target for four speckle averages........ 68

Figure 32: Effects of bandwidth on the backhoe target for eight speckle averages....... 69

Figure 33: Effects of bandwidth on the backhoe target for sixteen speckle averages......71

Figure 34: Efficient use of signal photons for speckle averages and resolution of a
backhoe target.............................................................. 72
# LIST OF ACRONYMS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>Two-Dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>Three-Dimensional</td>
</tr>
<tr>
<td>CCD</td>
<td>Charged-Couple Device</td>
</tr>
<tr>
<td>CRLB</td>
<td>Cramer Rao Lower Bound</td>
</tr>
<tr>
<td>DC</td>
<td>Direct Current</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
</tr>
<tr>
<td>HVAC</td>
<td>Heating, Ventilation, and Air Conditioning</td>
</tr>
<tr>
<td>IDFT</td>
<td>Inverse Discrete Fourier Transform</td>
</tr>
<tr>
<td>In</td>
<td>Inch</td>
</tr>
<tr>
<td>InGaAs</td>
<td>Indium Gallium Arsenide</td>
</tr>
<tr>
<td>kd tree</td>
<td>k-dimensional Tree</td>
</tr>
<tr>
<td>Ladar</td>
<td>Laser Detection and Ranging</td>
</tr>
<tr>
<td>LO</td>
<td>Local Oscillator</td>
</tr>
<tr>
<td>M</td>
<td>Meter</td>
</tr>
<tr>
<td>MHz</td>
<td>Megahertz</td>
</tr>
<tr>
<td>mm</td>
<td>Millimeter</td>
</tr>
<tr>
<td>MLE</td>
<td>Maximum Likelihood Estimator</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Squared Error</td>
</tr>
<tr>
<td>MWDH</td>
<td>Multi-Wavelength Digital Holography</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>nm</td>
<td>Nanometer</td>
</tr>
<tr>
<td>PCL</td>
<td>Point Cloud Library</td>
</tr>
<tr>
<td>PGA</td>
<td>Phase Gradient Algorithm</td>
</tr>
<tr>
<td>Radar</td>
<td>Radio Detection and Ranging</td>
</tr>
<tr>
<td>RCH</td>
<td>Range Compressed Holography</td>
</tr>
<tr>
<td>SAL</td>
<td>Synthetic Aperture Ladar</td>
</tr>
<tr>
<td>SAR</td>
<td>Synthetic Aperture Radar</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>TX</td>
<td>Transmitter</td>
</tr>
<tr>
<td>μm</td>
<td>Micrometer</td>
</tr>
</tbody>
</table>
# LIST OF SYMBOLS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>Starting element of the uniform distribution</td>
</tr>
<tr>
<td>$b$</td>
<td>Ending point of the uniform distribution</td>
</tr>
<tr>
<td>$B$</td>
<td>Bandwidth</td>
</tr>
<tr>
<td>$c$</td>
<td>Speed of light</td>
</tr>
<tr>
<td>$d$</td>
<td>Distance between spatial domain plane and the spatial frequency domain plane</td>
</tr>
<tr>
<td>$df_x$</td>
<td>The differential of the $f_x$ axis for the spatial frequency domain</td>
</tr>
<tr>
<td>$df_y$</td>
<td>The differential of the $f_y$ axis for the spatial frequency domain</td>
</tr>
<tr>
<td>$dx$</td>
<td>The differential of the $x$ axis for the spatial domain</td>
</tr>
<tr>
<td>$dy$</td>
<td>The differential of the $y$ axis for the spatial domain</td>
</tr>
<tr>
<td>$dz$</td>
<td>Discrete step in precision</td>
</tr>
<tr>
<td>$F$</td>
<td>Fourier Transform</td>
</tr>
<tr>
<td>$F^{-1}$</td>
<td>Inverse Fourier Transform</td>
</tr>
<tr>
<td>$f_x$</td>
<td>Coordinate in the spatial frequency domain</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$f_y$</td>
<td>Coordinate in the spatial frequency domain</td>
</tr>
<tr>
<td>$g(x, y)$</td>
<td>Function in the spatial domain</td>
</tr>
<tr>
<td>$G(f_x, f_y)$</td>
<td>Function in the spatial Frequency domain</td>
</tr>
<tr>
<td>$h$</td>
<td>A hologram in the spatial domain</td>
</tr>
<tr>
<td>$k$</td>
<td>The amount of steps in range and time dimension</td>
</tr>
<tr>
<td>$l_0$</td>
<td>The wavefront as a function of the local oscillator wave in the spatial domain</td>
</tr>
<tr>
<td>$m$</td>
<td>Discrete step used for Fourier transforming a function for $x$ and $\xi$</td>
</tr>
<tr>
<td>$MSE$</td>
<td>The mean squared error</td>
</tr>
<tr>
<td>$n$</td>
<td>The amount of discrete steps in frequencies</td>
</tr>
<tr>
<td>$N$</td>
<td>Total number of frequency samples</td>
</tr>
<tr>
<td>$N_x$</td>
<td>Amount of discrete elements in the $x$ direction in the new confocal surface</td>
</tr>
<tr>
<td>$N_y$</td>
<td>Amount of discrete elements in the $y$ direction in the new confocal surface</td>
</tr>
<tr>
<td>$N_\xi$</td>
<td>Amount of discrete elements in the $\xi$ direction in the previous confocal surface</td>
</tr>
<tr>
<td>$N_\eta$</td>
<td>Amount of discrete elements in the $\eta$ direction in the previous confocal surface</td>
</tr>
<tr>
<td>$N_{padding}$</td>
<td>Amount of zero padding applied</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>$N_{range\ bins}$</td>
<td>Amount of total reconstructed range bins</td>
</tr>
<tr>
<td>$N_{speckle}$</td>
<td>Number of speckle averages</td>
</tr>
<tr>
<td>$N_x$</td>
<td>Amount of discrete elements in the x direction in the spatial domain</td>
</tr>
<tr>
<td>$N_y$</td>
<td>Amount of discrete elements in the y direction in the spatial domain</td>
</tr>
<tr>
<td>$p$</td>
<td>Discrete step used for Fourier transforming a function for y and $\eta$</td>
</tr>
<tr>
<td>$P$</td>
<td>The perfect plane wave directed at a defined angle</td>
</tr>
<tr>
<td>$p_1$</td>
<td>The reconstructed point cloud</td>
</tr>
<tr>
<td>$p_2$</td>
<td>The reference point cloud</td>
</tr>
<tr>
<td>$P_U(u)$</td>
<td>Probability distribution function</td>
</tr>
<tr>
<td>Pixel pitch</td>
<td>The distance between the center of a pixel to the center of the neighbor pixel</td>
</tr>
<tr>
<td>$r$</td>
<td>The wavefront as a function of the reflected wave in the spatial domain</td>
</tr>
<tr>
<td>$S(x_0, y_0, n\Delta\omega)$</td>
<td>Signal that contains cross-range and frequency diversity</td>
</tr>
<tr>
<td>$S_0(n\Delta\omega)$</td>
<td>A signal in the spatial frequency domain</td>
</tr>
<tr>
<td>$s_0(k\Delta t)$</td>
<td>A signal in the time domain</td>
</tr>
<tr>
<td>$SNR$</td>
<td>Signal to noise ratio</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$SNR_{RH}$</td>
<td>Signal to noise ratio of the reconstructed hologram</td>
</tr>
<tr>
<td>$T_{amb}$</td>
<td>Time ambiguity</td>
</tr>
<tr>
<td>$TI$</td>
<td>Summed total intensity from a spatial frequency domain twin image</td>
</tr>
<tr>
<td>$TN$</td>
<td>Summed total noise from a spatial frequency domain area that is the same size as a twin image but does not include either of the twin images or the DC</td>
</tr>
<tr>
<td>$\bar{u}$</td>
<td>The mean of the probability distribution function</td>
</tr>
<tr>
<td>$x$</td>
<td>Coordinate in the spatial domain</td>
</tr>
<tr>
<td>$y$</td>
<td>Coordinate in the spatial domain</td>
</tr>
<tr>
<td>$z$</td>
<td>Axis which is in line with optical axis</td>
</tr>
<tr>
<td>$z_{amb}$</td>
<td>Range ambiguity</td>
</tr>
<tr>
<td>$(x_0, y_0)$</td>
<td>A single location in the cross-range</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>The cosine of the angle between the propagation of the plane wave and the x-axis</td>
</tr>
<tr>
<td>$\beta$</td>
<td>The cosine of the angle between the propagation of plane wave and y-axis</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>The cosine of the angle between the propagation of the plane wave and z-axis</td>
</tr>
<tr>
<td>$\Delta f$</td>
<td>Step in frequency</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>Temporal spacing of samples</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$\Delta x$</td>
<td>The size of the discrete cross-range step in the new confocal surface</td>
</tr>
<tr>
<td>$\Delta z$</td>
<td>Range step defined by the temporal spacing of samples</td>
</tr>
<tr>
<td>$\Delta \xi$</td>
<td>The size of the discrete cross-range step in the previous confocal surface</td>
</tr>
<tr>
<td>$\Delta \omega$</td>
<td>The step between discrete angular frequencies</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Coordinates in one of the confocal planes</td>
</tr>
<tr>
<td>$\theta$</td>
<td>The angle of the LO to the optical axis to move the twin images into the center of a quadrant</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>The wavenumber, $\frac{2\pi}{\lambda}$</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Wavelength</td>
</tr>
<tr>
<td>$\xi$</td>
<td>Coordinates in one of the confocal planes</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>Variance of the probability distribution function</td>
</tr>
<tr>
<td>$\sigma^2_{CRLB}$</td>
<td>Variance of the Cramer Rao lower bound</td>
</tr>
<tr>
<td>$\sigma^2_{z,d}$</td>
<td>Variance of uniform distribution of a range bin</td>
</tr>
<tr>
<td>$\hat{\psi}_{ML}$</td>
<td>Maximum likelihood estimator</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

In 1948, Dennis Gabor discovered the topic of holography; however, the technology to produce holograms was not yet invented. Gabor did not have a coherent light source nor a technique to separate both reconstructed images from the direct current (DC) term [1]. Once lasers were invented, on-axis holograms were able to prove Gabor’s theory. Leith and Upatnieks introduced an off-axis holograph that solved the twin image issue that plagued Gabor’s original technique. Leith and Upatnieks’ hologram was also called the off-axis hologram due to the different optical axes of the reference beam and target’s wavefront [2].

The first surge in holographic research occurred when the laser was invented. This allowed holograms to be constructed within the laboratory. In the 1960s, B. Hildebrand and K. Haines discovered multi-wavelength holography, a precursor to Range Compressed Holography (RCH) [3]. Compared to the use of a single wavelength, multi-wavelength holography provided a greater range ambiguity to recreate a 3D scene. Due to the technology at the time, every additional wavelength required a new laser, further adding to the complex setup and reconstruction required for multi-wavelength holography.
The next notable surge in holography began with the invention of CCD imaging devices. These devices allowed the study of digital holography, which greatly simplified the creation and viewing of holograms. Digital RCH was demonstrated by J. Marron, T. Schulz, and K. Schroeder in the early 1990s [4 - 6]. A steady growth of RCH continued with J. Stafford, D. Rabb, B. Duncan, and M. Dierking in the 2010s. Stafford demonstrated a technique that achieves a high range resolution and cross-range by using synthetic apertures and range compression [7 - 13].
CHAPTER 2
THEORY

This section contains the theory needed to conduct RCH. It will start by going over Fourier transforms, followed by the theory of holographic imaging used for recording multi-frequency holograms. Next, it will demonstrate the corrections needed for reconstruction of experimental data due to errors in the recorded holograms. Then it will examine the effects up sampling has on the range dimension and impacts on range variance. Finally, the section will cover the theory of the quantitative analysis chosen for this thesis.

2.1. Holography

2.1.1. Introduction to Holographic Imaging

Fourier transforms are the foundation to many different disciplines, including holographic imaging. Fourier Optics uses a 2D Fourier transform to change the electro-magnetic field from one domain to a new domain. For Fourier Optics, the two domains of interest are the spatial and spatial frequency domains. To transform from the spatial domain to the spatial frequency domain, equation (1) is used. To transform back the inverse Fourier transform, equation (2) is used [14].
\[ G = F(g) = \int_{-\infty}^{\infty} g(x, y) e^{-j2\pi(f_x x + f_y y)} \, dx \, dy \] (1)

\[ g = F^{-1}(G) = \int_{-\infty}^{\infty} G(f_x, f_y) e^{j2\pi(f_x x + f_y y)} \, df_x \, df_y \] (2)

Where \(x\) and \(y\) are spatial domain coordinates, \(f_x\) and \(f_y\) are spatial frequency domain coordinates, and \(g\) and \(G\) are respectively functions in the spatial and spatial frequency domain. The Fourier transform is equivalent to a scaled Fresnel diffraction between confocal surfaces. The confocal surfaces are two spherical surfaces with a radius of curvature equal to the propagation distance. In order to use the confocal surface approximation the object phase was defined relative to a confocal surface and then an imaging configuration was assumed by using a lens that conjugates the phase curvature difference between the confocal surfaces associated with the propagation from the object and to the image. The Fourier transform is converted to a Fresnel propagation between confocal surfaces defined by the equations below [14].

\[ G = F(g) = \frac{e^{jkd}}{j\lambda d} \int_{-\infty}^{\infty} g(x, y) e^{-j2\pi \frac{x}{\lambda d}} (\xi x + \eta y) \, dx \, dy \] (3)

\[ g = F^{-1}(G) = \frac{e^{jkd}}{j\lambda d} \int_{-\infty}^{\infty} G(\xi, \eta) e^{j2\pi \frac{\xi}{\lambda d}} (\xi x + \eta y) \, d\xi \, d\eta \] (4)

Where \(\lambda\) is the wavelength, \(d\) is the distance between confocal surfaces, and \(\kappa\) is the wave number, \(\frac{2\pi}{\lambda}\). Continuous functions are very useful for theory; however, modern
computers and CCD cameras have allowed the discrete Fourier transforms to gain importance due to the increased efficiency, reduction in processing time, and discrete nature. The discrete Fresnel diffraction is described in equation (5) and the inverse Fresnel diffraction is described in equation (6) [14].

\[
F(g) = \frac{e^{j\kappa d}}{j\lambda d} \sum_{m=0}^{N_x-1} \sum_{p=0}^{N_y-1} g(\xi, \eta) e^{-\frac{j2\pi}{\lambda d}(\frac{mx}{N_x} + \frac{py}{N_y})}
\]  

\[
P^{-1}(G) = \frac{e^{j\kappa d}}{j\lambda d} \sum_{m=0}^{N_x-1} \sum_{p=0}^{N_y-1} G(x, y) e^{\frac{j2\pi}{\lambda d}(\frac{mx}{N_x} + \frac{py}{N_y})}
\]

where \(N_x\) and \(N_y\) are the amount of discrete elements in the transformed confocal surface, \(N_\xi\) and \(N_\eta\) are the amount of discrete elements in the first confocal surface, and \(m\) and \(p\) are integers for the amount of discrete steps. When a discrete Fresnel diffraction is applied, it converts the axis to the new confocal surface and changes the discrete step size. The spatial frequency domain’s discrete step size can be calculated by using equation (7) for any discrete transform.

\[
\Delta x = \frac{\lambda d}{N_x \Delta \xi}
\]

where \(\Delta x\) is the resolution at the new confocal surface and \(\Delta \xi\) is the resolution at the previous confocal surface.
2.1.2. Holograms

Holography is the study of recording the whole message, or recording all the information of the returned light. In terms of optics, the whole message is defined as being able to read both amplitude and phase, rather than solely the intensity of the recorded image. To gain the ability to record the amplitude and the phase, a Local Oscillator (LO) must be mixed with the reflected wavefront on the recording surface. This mixing allows the phase to be observed in a pure intensity recording in the form of spatial fringes. The spatial fringes have spatial frequencies that can be filtered to reconstruct the target. The equation for a hologram is displayed in equation (8).

\[
h(x, y) = |r(x, y) + lo(x, y)|^2
\]

\[
= |r(x, y)|^2 + |lo(x, y)|^2 + r^*(x, y)lo(x, y) + r(x, y)lo^*(x, y),
\]

where \( r \) is the reflected field, \( lo \) is the field of the LO and \( h \) is the recorded hologram. The ideal LO would have the same curvature as \( r(x, y) \) except for a tilt as described in equation (9) [14].

\[
P(x, y, z) = e^{j\frac{2\pi}{\lambda}z(\alpha x + \beta y)} * e^{j\frac{2\pi}{\lambda}yz}
\]

where \( P \) is the tilt between the reflected field and the LO, \( \alpha \) is defined as the cosine of the angle between the propagation of the plane wave and the x-axis, \( \beta \) is defined as the cosine angle between the propagation of plane wave and y-axis and \( \gamma \) is defined as the cosine of the angle of the plane wave and z-axis.
The hologram is a distribution of intensity. Therefore, it can be recorded by a CCD camera. To gain the amplitude and phase of the wavefront, the hologram needs to be reconstructed. To reconstruct an image plane hologram, the hologram was Fourier transformed to the pupil plane or spatial frequency domain. Then the 3rd or 4th term from equation (8) was cropped and masked. The crop was manually chosen by isolating one of the twin images, the 3rd or 4th terms from the equation, which is equivalent to filtering out the spatial frequencies that contain the target information. Next, the isolated term was Fourier transformed again to gain the reconstructed image in the image plane. Using this process the amplitude and phase information can be extracted. The other term could be inverse Fourier transformed to reconstruct the target because the twin images are mirrored complex conjugates of each other.

The first two terms in equation (8) are the DC components and will be in the center of the reconstructed images for on-axis holograms. The DC component of the reflected wave will be overpowered by the DC of the LO due to the significant difference in recorded power. The next two terms are the twin images, which are the reconstructions of the target. Both twin images contain the amplitude and the phase of the original target.

The Leith-Upatnieks hologram is also known as the off-axis hologram because the LO is offset from the optical axis. The offset is useful as it allows for the separation of the images from the DC. To correctly offset the LO for a digital system, the Nyquist sampling and the location of a reconstructed image must be considered. Assuming the small angle approximation, the angle at which the LO must be placed to center the frequency content
of the hologram, with respect to the CCD camera, is shown in equation (10) below. Where pixel pitch is the distance between the center of one pixel to the center of the neighboring pixel.

\[
\theta = \frac{\lambda}{4 \times \text{pixel Pitch}}
\]  

(10)

Figure 1: Twin images of a Fourier transformed hologram in the spatial frequency domain. The figure is an image of the twin images centered in the second and fourth quadrant from equation (8) in the spatial frequency domain. The image was scaled, so the DC did not dominate over the twin images. Either of the circles in the second or fourth quadrant could be used to reconstruct the hologram. In addition, the DC terms can be seen in the center of the figure.

This angle is in both the x and y directions causing the 3rd and 4th terms to be centered in their respective quadrants. This effect can be observed above in the spatial frequency domain of Figure 1. The image displays the correct placement of the LO with the two large, green circles representing the amplitude of the reflected field at the pupil. The reconstructed pupil fields will be mirror images about the origin, one the complex
conjugate of the other. Additionally, the first two terms of equation (8) are visible at the center of the image.

One of the more useful features of holograms is the ability to reconstruct them to display 3D images. A 3D reconstructed hologram’s main limitation is the system’s range ambiguity, which is the wavelength used to record the hologram. Holograms are often recorded using visible or IR light, restricting the range ambiguity from a fraction of a micron to a few microns. While unwrapping techniques have been proven to mitigate the range ambiguity of the holograms, the reconstructed hologram remains limited by the amount of reconstructed cross-range pixels. The unwrapping techniques would require unreasonable amounts of cross-range pixels to create a 3D reconstruction of a car. A technique developed to overcome the wrapping limits of holograms is Multi-Wavelength Digital Holography (MWDH). MWDH overcomes the range ambiguity limit through the use of a synthetic wavelength created from using two different frequency holograms [15]. MWDH was improved upon to use greater than two frequencies in a technique called RCH. The details of RCH are discussed in the next section.

2.1.3. Phase Gradient Algorithm (PGA) Method for Three-Dimensional Holographic Laser Detection and Ranging (Ladar) Imaging

Sections 2.1.3, 2.1.3.1, and 2.1.3.2 are a direct reprint with permission from Applied Optics, "Phase gradient algorithm method for three-dimensional holographic ladar imaging," Appl. Opt. 55, 4611-4620 (2016) [8].
2.1.3.1. Resolution and Ambiguity

We begin with a digital hologram image dataset, collected over some temporal frequency range, with equal spacing $\Delta \omega$ between the discrete frequencies. Furthermore, the complex image data is assumed to be arranged in increasing temporal frequency order and focused, or sharpened, in both cross-range dimensions. Sharpening eliminates cross-range phase aberrations; what remains is a piston-like phase uncertainty between neighboring frequency plane images. There are many excellent references for digital holography processing, and the method used to obtain the data is immaterial if the above conditions are met [16, 17]

Figure 2 shows the general principle of 3D holographic ladar. By using the relationship between temporal frequency and time, along with a simple coordinate transformation from time to range, a set of complex valued digital hologram images of sequential temporal frequency can be compressed via an IDFT to yield a full 3D dataset.

Note in Figure 2 the convention, used throughout this paper, of explicitly treating only temporal frequency $n\Delta \omega$ and range $k\Delta z$ as discrete variables. We acknowledge the discrete nature of the sampling sensor elements in the cross-range dimensions but assume them to be of consistent size and response, coherent with respect to each other, and of sufficient sampling period to be nearly continuous. In addition, for a particular frame, all cross-range samples are simultaneously collected. Temporal frequency, however, is assumed to have an unknown and random phase relationship among the samples. Furthermore, the frequency separation is user selectable.
Figure 2: 3D holographic ladar. Digital holograms collected at discrete radian temporal frequencies on the left are transformed by and IDFT to create the 3D image on the right.

Now, consider a signal $S_0(n\Delta\omega)$ from a single cross-range location $(x_0, y_0)$ of Figure 2. The relationship between $S_0(n\Delta\omega)$ and its time domain signal $S_0(k\Delta t)$ can be expressed in discrete form by use of the IDFT according to

$$s_0(k\Delta t) = \frac{1}{N} \sum_{n=1}^{N} S_0(n\Delta\omega)e^{j(2\pi)\frac{n}{N}k}$$

(11)

where $k\Delta t$ is the discrete temporal variable, $\Delta\omega$ is the frequency sampling interval, and the sum is over $N$ total frequency samples. Now the temporal spacing of samples $\Delta t$ is dependent on the total frequency range or bandwidth according to [18]

$$\Delta t = \frac{2\pi}{N\Delta\omega}$$

(12)

and is related to range by

$$\Delta z = \frac{c\Delta t}{2}$$

(13)
where \( c \) is the speed of light and the factor of \( \frac{1}{2} \) is due to the round trip travel of the pulse from the transceiver to the target and back. Equation (13) is the coordinate transform that, when applied to Eq. (11) produces the explicit relationship between the left and right datasets of Figure 2. Furthermore, inserting Eq.(12) into Eq. (13) gives the smallest achievable differential range measurement (range resolution), given by

\[
\Delta z = \frac{c\pi}{N\Delta \omega}
\]

Equation (14)

Note that \( N\Delta \omega / 2\pi \) is the total pulse bandwidth \( B \), allowing us to write

\[
\Delta z = \frac{c}{2B}
\]

which is identical to the form commonly found in synthetic aperture radar (SAR) and synthetic aperture ladar (SAL) literature [19]. Here, Eq. (15) is an exact expression of range resolution arising from the properties of the discrete Fourier transform (DFT). Another interesting property of Eq. (11) is that the IDFT of a signal regularly sampled in frequency results in a periodic function in time. Presuming the sampling locations are relatively closely spaced, the IDFT yields a signal whose ambiguity period \( T_{amb} \) is related to the frequency sample spacing by

\[
T_{amb} = \frac{2\pi}{\Delta \omega}
\]

Any time samples separated by \( T_{amb} \) will have the same value and will therefore be ambiguous. Again, relating time to range, we can find from Eq. (16),
\[ z_{amb} = \frac{c \pi}{\Delta \omega} \]  

where \( z_{amb} \) is the range ambiguity or the unambiguous range of the dataset. This sets a practical limit on the unaliased scene depth to be imaged for a given frequency separation and is again directly due to properties of the DFT.

So in theory, a set of digital hologram images collected at regularly spaced temporal frequencies can be range compressed with an IDFT to produce a 3D image. Due to the properties of the DFT, the separation of the frequencies has important impacts on the scene depth, while the total bandwidth affects range resolution. Of course, in the process of collecting actual data, noise with both amplitude and phase is often introduced, and simply applying an IDFT may no longer correctly compress the signal. The process to estimate and correct this noise is the subject of the next section.

2.1.3.2. Application of Phase Gradient Autofocus Algorithm

As detailed above, the steps to combine and transform a set of multi frequency digital hologram images into a 3D image are straightforward; however, differential phase aberrations across the frequency bins can lead to poor compression. These errors can be induced by temporal atmospheric effects as well as phase wander of the laser illumination source, or platform jitter. To correct the errors, we can exploit existing algorithms designed for SAL. Perhaps the best known of these is the phase gradient autofocus algorithm. The central assumption in the PGA method is that the field returning from a point target centered in the scene will manifest as a plane wave at the pupil of the sensor;
in other words, it will possess a phase which is flat across the entire synthetic aperture. The algorithm requires organizing the complex image data so that the brightest point targets are registered to the central cross-range bin and then windowed so that weaker targets’ contributions are minimized. It is designed to force any aberrations to be revealed in the synthetic aperture plane as deviations from the flat phase just described. Then, a phase correction can be estimated and refined as the algorithm is repeated. We seek to apply the PGA technique to 3D holographic ladar data, not for correcting cross-range phase aberrations but for phase errors across multiple temporal frequency images. Figure 3 shows the steps required to accomplish this.

Figure 3 shows the steps required to accomplish this.

**Figure 3: Processing steps for applying the PGA method to 3D holographic ladar for correction of temporal frequency phase error. Dashed boxes denote optional steps.**

A. Assemble Complex Image Plane Data Cube

We begin with a dataset arranged as shown in Figure 2, where a set of cross-range complex images of equal frequency separation are assembled into a complex data cube.
Within this dataset, a mean phase aberration between neighboring temporal frequencies exists that is assumed to be spatially invariant over the \((x,y)\) plane. This assumption requires that the temporal frequency steps are small enough that the speckle does not evolve appreciably between neighboring samples. Since all of the cross-range samples of each hologram are simultaneously recorded, each contains a copy of this common mode aberration along with any pixel specific noise inherent to the detection process. The PGA method, although designed for other applications, is well suited for data with these characteristics.

B. Apply a Cross-Range Target Support Mask

This is an optional step. Since each image already contains a 2D projection of the target whose cross-range extent is often readily delimited, a spatial support filter can be applied by masking around the desired cross-range target information. Note that this will not affect information in the range phase dimension. The support filter simply removes known cross-range clutter data from the algorithm.

C. Range Compress (1D IDFT)

Step 3 requires that the data cube be range compressed, pixel by pixel, via a 1D IDFT over \(n\Delta \omega\). The data is now entirely in 3D complex image space.

D. Centershift Brightest Range Values

Now, the brightest value in range of each cross-range pixel, assumed to be a high confidence measurement of a point target, is located and centershifted in range to the
central range bin. The result of this step yields a data cube whose intensity appears to be a single bright plane (the central range bin) in a 3D volume of noise and clutter. Centershifting eliminates phase corresponding to the depth dimension of the target’s 3D structure. The remaining phase profile of each pixel then contains registered common mode aberration content along with independent realizations of noise and clutter. With the target depth information removed, we can accurately estimate the aberration. With the many independent pixel realizations available, through averaging we can enhance the accuracy of the aberration estimate while simultaneously suppressing the noise.

E. Window in the Range Dimension

After the centershifting step, above, the other weaker pixels are effectively treated as noise or clutter. An optional window can therefore be applied around the central range plane. Just as with SAL data, it is important not to overly truncate the centered response while minimizing the contributions due to the weaker pixels. Windowing is performed to increase signal-to-clutter ratio; however, step 5 is designated optional since in holographic remote sensing applications clutter is often minimal along the range dimension. This occurs because, in most cases, the pulse will be completely reflected (neglecting target absorption effects) upon incidence with a target, leading to one strong return in each cross-range pixel. Also, recall the assumption that the data is focused in cross-range. If cross-range defocus is present, target content from different range locations can overlap in cross-range, making it appear as if multiple range bins within a pixel are occupied when in fact only one may contain true target information. In this case,
the apparent clutter in the data due to defocus can be mitigated by applying a range window. If the data is well focused and SNR is also adequate, then the benefit of windowing is minimal. As SNR decreases, windowing again becomes more effective as noise begins to contribute more significantly to the point response.

F. Decompress in Range (1D DFT)

For SAL, the preceding steps are designed such that any aberrations will now be manifest across the synthetic aperture as a deviation from an expected flat phase. For 3D holographic ladar, the phase deviations occur instead across $n\Delta\omega$. The windowed, centershifted data is decompressed in step 6 via a 1D DFT in order to allow us to estimate the phase error in subsequent steps.

G. Compute the Phase Gradient Estimate

In step 7, a phase gradient vector is estimated. There are a number of estimation methods (also called kernels) in the literature from which to choose. We selected the maximum likelihood estimator (MLE) $\hat{\psi}_{\text{ML}}$, which is compactly written as [20]

$$\hat{\psi}_{\text{ML}} = \angle \left( \sum_x \sum_y S(x, y, n\Delta\omega)S^*(x, y, (n+1)\Delta\omega) \right)$$

where the outer operator indicates the angle function and the double sum is over all of the detector array pixels. Equation (18) can be understood as the combination of three steps, the first of which is used to compute an $N$-element phase difference vector $S_0$ for each pixel, where the first sample is arbitrarily set to zero. For example, at some pixel $(x_0, y_0)$ a complex vector is computed,
where the data from each frequency bin is multiplied by its conjugated neighbor. The phase of this vector contains one realization of an estimate of the phase gradient over temporal frequency. Once this is done for all pixels, we have an ensemble of phase gradient estimates. To minimize the variance of the uncorrelated noise, an average is then calculated by performing a complex sum over all of the pixels. This is important because holographic ladar, like all coherent processes, is susceptible to speckle, and Eq. (18) appropriately weights the phase gradient measurement based on the amplitude of the samples.

Note that since the phase is the quantity of interest, the modulus scaling factor can be omitted from the averaging step of Eq.(18) After correcting the phase of the ensemble average, the final result of step 7 is a single N-element phase gradient vector, where again the first element is arbitrarily set to zero. Lastly, we reiterate that the phase gradient estimated here is over temporal frequency bins, unlike SAL, where the gradient is over cross-range aperture bins.

H. Integrate the Phase Gradient to Recover the Phase Aberration Estimate

Once the phase gradient vector has been calculated, the temporal frequency phase aberration estimate is retrieved by integrating the result of step 7.
I. Detrend

Any piston phase offset and linear phase trend in the phase aberration vector is now removed before its conjugate is applied to the complex image plane data. Failure to remove any residual linear phase trend may result in data that is circularly shifted in range. This has no effect on the algorithm accuracy, and if present can be removed once processing is completed. However, an algorithm exit decision may be based on a correction metric that benefits from detrended phase, leading to unnecessary algorithm iterations if the trend is not removed.

J. Apply Phase Correction to Complex Image Plane Data

At this point a single N-element vector containing the phase aberration estimate has been created. The conjugate of this estimate is then applied across every pixel of the data cube created in step 1.

K. Corrected?

The decision step implicitly includes a correction metric of the user’s choosing. This can be simple visual scrutiny, be based on residual phase error, or be image content driven (e.g., morphological analysis or image entropy minimization). In any case, once it has been determined that the algorithm has sufficiently converged, the loop is exited and the final 3D holographic ladar image is formed through a final IDFT step.
The steps outlined above will allow conventional PGA estimators to be directly applied to 3D holographic data. Application as described here will enable range compression when phase aberrations are present.


2.1.4. Range Precision

After the datasets have been corrected, the frequency dimension will need to be up sampled to increase range precision. Zero padding in the frequency dimension allowed RCH to improve sampling precision in the range dimension. The zero padding increased the probability of selecting the correct peak in the range dimension while also decreasing the size of each range bin. Zero padding does not increase the resolution of the system, but does allow for increased precision in the range dimension by reducing the range step size, as defined in the equation below [8].

\[
dz = \frac{c}{2 \cdot \Delta f \cdot N_{\text{padding}} \cdot N}
\]  \hspace{1cm} (20)

Where \(dz\) is the size of a range bin and \(N_{\text{padding}}\) is the amount of zero padding applied to the dataset. Increasing the sampling rate beyond Nyquist increased the probability of having a sample near the peak amplitude along the range dimension. Using a simulation, the amount of padding was statistically tested to calculate when an increase in zero padding would not significantly increase the amplitude of the peak sample. This
simulation was conducted because increasing the amplitude of the continuous signal peak increases the likelihood of finding the correct reconstructed surface. When the peak amplitude was found to be stabilized, the system was assumed to be saturated in the range sampling. The system results are displayed in Figure 4 below.

![Change in Peak from Range Padding Using 400 MHz Frequency](image)

**Figure 4:** Plotting the effects of peak intensity when zero padding is applied. The figure is the plot displaying the effects of peak intensity when zero padding is applied. The x-axis displays the ratio of up sampling to the amount of frequencies used in the dataset. The y-axis displays the percent of the peak intensity increase due to zero padding the array. This plot is only for four discrete frequency data, but eight, sixteen and 32 frequencies have similar plots with only a slight change in the upper bound.

In order to find the amplitude of the peak, the simulation (Figure 4) was conducted by using the average of 44,000 1D-frequency returns that were range compressed. This process was repeated with different amounts of zero padding. The peak amplitude was compared to the unpadded peak for varying amounts of zero padding. The Nyquist
sampling rate for the intensity along the range dimension corresponded to a value of two on the x-axis. The plot showed increased zero padding results as an increase of the peak value until it saturated at approximately eight times the number of frequencies followed by diminishing returns. To ensure the signal peaks were accurate and to limit digitization noise impact on the range variance, the processing used in this paper has a range padding sixteen times larger than the recorded dataset in the frequency dimension.

2.2. Variance Bounds for 1D Range Compression

To understand the results of the data, there must be an expected range of values. The expected values will be calculated from the assumption that the system will only vary in the range dimension and a results analysis will be conducted for the range variance. Picking the correct target to image can achieve this assumption.

2.2.1. Lower Bound

For a 1D RCH, the best possible result is the Cramer Rao Lower Bound (CRLB), a theoretical lower bound for variance that assumes the unbiased estimator for a fixed parameter. The CRLB is used as the lower bound in many different systems, including radio detection and ranging (radar). Ladar imaging techniques and Radar return techniques are nearly identical other than the wavelength used.

The CRLB used in this paper is for the variance of range estimation using linear frequency modulation, and was derived from “Principles of Modern Radar.” All of the conditions
“Principles of Modern Radar” puts on the CRLB are satisfied by RCH. The equation for the lowest variance is given in equation (21) [21].

$$\sigma_{CRLB}^2 = \frac{3c^2}{8\pi^2 \text{SNR} * B^2}$$  \quad (21)

Where SNR is the signal-to-noise ratio of the power of the returned waveform and $\sigma_{CRLB}^2$ is the lowest achievable variance. Richards states that when multiple events of white noise are combined together, the variance of the system will linearly reduce. RCH is limited by the Poisson noise of the LO; however, due to the large count of photons expectation of the LO, the central limit theorem allows it to be approximated as Gaussian white noise. This approximation of the LO having Gaussian white noise allows the CRLB to account for speckle realizations in equation (22), where $N_{\text{speckle}}$ is the number of speckle realizations used in the system [21].

$$\sigma_{CRLB}^2 = \frac{3c^2}{8\pi^2 \text{SNR} * N_{\text{speckle}} * B^2}$$  \quad (22)

2.2.1.1. 1D Analysis

The presented CRLB has a variance for only a single dimension range. Due to the discrete reconstructed range values, an uncertainty remains in the reconstructed data. Digitization leads to the binning of reconstructed points to a particular sample value; although, the actual location could be anywhere between the samples. The uncertainty
associated with the digitization of the range can be calculated as the standard deviation of a uniform distribution over a range bin, $\sigma^2_{z,d}$, as shown in equation (23).

$$\sigma^2_{z,d} = \frac{dz^2}{12}$$  \hfill (23)

The uniform distribution is attributed to the assumption that the target location has an equal probability of being anywhere within a range bin.

2.2.2. Upper Bound

The variance for the CRLB was given, providing the results for the best possible reconstruction. The worst case would be the reconstruction of a dataset, which was only noise. The reconstruction would result in an equal probability of any range bin being selected as the object location. The equal probability gives the system a uniform distribution of the range for every cross-range pixel. The variance of a noise-reconstructed dataset would be calculated by finding the variance of the uniform distribution.

The probability distribution for uniform distributed noise is one over the amount of unique values that are available, from $a$ to $b$.

$$P_U = \frac{1}{b - a}$$  \hfill (24)

$b$ is assumed to be larger than $a$. The first moment is then calculated using equation (25),

\[ \text{bias} = \frac{1}{b - a} \int_a^b x \, dx. \]
\[ \bar{u} = \int_{-\infty}^{\infty} u \ast P_U(u) du \]  \hspace{1cm} (25)

where \( P_U \) is the probability density function and \( \bar{u} \) is the first moment [22].

Next, the second moment is used to calculate the variance of the system using equation (26),

\[ \sigma^2 = \int_{-\infty}^{\infty} (u - \bar{u})^2 \ast P_U(u) du \]  \hspace{1cm} (26)

where \( \sigma^2 \) is the variance [22]. When plugging in the probability density function for equation (25), the result is shown in equation (27) below.

\[ \bar{u} = \int_{a}^{b} u \ast \frac{1}{b - a} du = \frac{b + a}{2} \]  \hspace{1cm} (27)

Plugging the results from equation (27) into equation (26) gives the variance of the uniform distribution to be

\[ \sigma^2 = \int_{a}^{b} \left( u - \frac{b + a}{2} \right)^2 \ast \frac{1}{b - a} du = \frac{1}{12} \ast (b - a)^2. \]  \hspace{1cm} (28)

Converting the results from equation (28) to be used by RCH gives

\[ \sigma_{\text{uniform distribution}}^2 = \frac{(dz \ast N_{\text{range bins}})^2}{12} = \frac{z_{\text{amb}}^2}{12} \]  \hspace{1cm} (29)

where \( N_{\text{range bins}} \) is the amount of total range bins.
2.3. Quantitative Analysis

The analysis technique that will be used to compare the reconstructed 3D images will be contained within the bounds when analyzing a 1D target. This quantitative analysis will be using a 3D point cloud function to evaluate the reconstruction of the 3D coherent imaging technique. The point cloud function will provide a unique evaluation method for RCH. Additionally, the numeric results will give quantitative results for RCH based on reconstruction parameters.

2.3.1. Point Cloud Library

The Point Cloud Library (PCL) is a C++ library that was developed as a large-scale, open source project that is used to analyze 2D and 3D images. The development of this library allows for collaboration between many global companies and educational institutes to develop and share different image filtering, displaying, producing, stitching, and quantifying metrics [23].

3D images were analyzed by using a point cloud metric to quantify the results of the reconstruction. This analysis was completed by using a PCL function called getFitnessScore. This function was selected as the chosen metric because it would calculate the mean squared error (MSE) between two different point clouds. The metric allows the reconstruction to be evaluated and returned with a single number to quantify how well the reconstructed image matched the reference image. Because most RCH work
has been compiled in MATLAB, using C++ was not desirable for the development of a quantitative metric.

To use the PCL’s function in MATLAB, a C++ program was developed. The program created would read in two point clouds, run the function, calculate the error in each of the dimensions for every point, and save all of the results. The program was then made into an executable allowing MATLAB to run the function from the PCL.

2.3.2. GetFitnessScore Function

The *getFitnessScore* function works by taking two registered point clouds and calculating the squared Euclidian distance between every point on the first point cloud to the closest point on the second point cloud. It then averages the results to supply the MSE.

\[
MSE = \left( (p_{1,x} - p_{2,x})^2 + (p_{1,y} - p_{2,y})^2 + (p_{1,z} - p_{2,z})^2 \right)
\]

(30)

where MSE is the mean squared error, \(p_1\) is the first point cloud, and \(p_2\) is the second point cloud. This allows the first point cloud to be the reconstructed hologram and compare it to the reference point cloud, a pristine point cloud of the target; therefore, giving a quantified metric for how well the reconstructed hologram represents the target. The *getFitnessScore* has the ability to calculate the MSE between two different size point clouds. By finding the closest point on the reference point cloud, it allows sparsely reconstructed 3D point clouds to be compared to complete reference point clouds [24].
The function uses the k-dimensional tree (kd tree) algorithm to find the closest point on the point cloud. A kd tree works by first choosing a point from the reconstructed point cloud and then finding the closest point on the reference point cloud. The closest point on the reference point cloud is found by first dividing its dimensions by the median of each of the respected dimension. These new sections are referred to as branches. The new branches are then divided by their respective medians and this process is repeated until the branches can no longer be divided by the median. The final sections are then called leaves. The point picked on the reconstruction is now compared to all the nearby leaves to find the closest point in the reference point cloud. The process is then repeated for every point on the reconstructed point cloud and then the results of the squared Euclidean distance are averaged [25].
CHAPTER 3

SIMULATION METHODOLOGY

This chapter goes over the methodology used to simulate the RCH in MATLAB. Figure 5 is a visual representation of the methodology used to simulate the quantification of RCH and Appendix A contains the code used to simulate one of the targets. This section will cover how the targets were created and how the simulated wavefronts propagated through the simulated optical system. This chapter will also cover how the simulation created holograms, how noise was added to the holograms, and how the simulation reconstructed the 3D images. Lastly, this chapter will cover how the algorithm was used to calculate the MSE of the reconstructed holograms.
Figure 5: Simulation Methodology. The figure displays the flow chart of the methodology of the simulated RCH. Each white box is an individual step in the system. The purple box corresponds to different laser frequencies needed for range-compression. Increasing the amount of frequencies used will increase the bandwidth and reduce the range bins. The green box relates how the system can create different speckle averages to reduce the amount of speckle noise. Creating a pristine target can be inside or outside of the green box, but was included in the green box for visual aesthetics. The final boxes on the right of the green box apply speckle averaging and surface reconstruction.

3.1. Range Compressed Holography Simulation

The first block in Figure 5 indicates the creation of the pristine targets. Two targets, a flat plate and a backhoe, were simulated and are displayed in Figure 6. The flat plate was chosen to isolate the effects of range noise on the MSE, as well as to be experimentally verified in the laboratory. Whereas, the backhoe was chosen to determine how limited cross-range affects the MSE for a realistic target.
Figure 6: Reference Target Images. The two images are pristine point cloud images of the two targets used for the simulation. The top image is the flat plate target used for simulations and experimental verification. The bottom image is the backhoe target used to see the effects of range diversity on the point cloud metric through simulation.

There were two different ways that the targets were created in the simulation. Using MATLAB, the range values for each cross-range location for the flat plate target were simulated using an array of zeros. The backhoe target was created by rendering a facet model to produce a pristine point cloud at a single observation angle. The assumptions
for RCH were that the target only returned light from a single angle and it did not significantly rotate while recording multiple speckle averages. The targets were then stored to be used as reference point clouds.

The next block in Figure 5 describes the simulation of surface roughness by adding real Gaussian white noise to the range values in the arrays. The roughness was created using the \textit{randn} function in MATLAB. The standard deviation of the Gaussian noise was set to 50 microns for the sixteen speckle realizations. Sixteen independent speckle realizations were simulated by generating random noise realizations for each target.

The next step was to simulate the optical system, starting with reflecting the Gaussian wave off the target (Figure 5). Assuming an effective point source transmitter, the Gaussian amplitude and wavefront were calculated at the target. For a particular cross-range pixel, the reflected amplitude and phase were determined as a function of the surface normal direction based on lambertian surface reflectivity.

Next, the imaging optics were simulated to propagate the reflected signal to the focal plane. The process to simulate the hologram involved propagating to the pupil plane and applying the optical aperture and limiting the wavefront which generated speckle on the reconstructed image. After which, the cropped wavefront was propagated and focused onto the image plane.

The following two blocks indicated in the methodology (Figure 5) created and added noise to the hologram. The hologram was created by combining the reflected wave at the sensor with the LO. The LO originated from an effective point source located next to the
optics, providing the correct angle with respect to the optical axis, as defined in equation (10). Real Gaussian noise was added to the hologram to simulate shot noise from the LO. The noise was dominated by the LO due to the assumption that it has large photon counts relative to the reflected signal. The noise was approximated as Gaussian based on the assumption that a large number of photo-events per pixel per hologram ensured the validity of the central limit theorem.

Figure 7: First step in reconstructing RCH. The figure displays the first step in reconstructing RCH. This step was to reconstruct each of the individual holograms by Fourier transforming, cropping, masking, and then Fourier transforming to produce the reconstructed complex image. The images shown were experimental data.

Reconstructing the complex fields from individual holograms and creating the dataset are the last two steps in the purple box of Figure 5. The reconstruction of a complex image is shown in Figure 7. First, the hologram was Fourier transformed to the spatial frequency domain. The absolute value of the field shows two circles in the center of their respective
quadrants and the central DC component. These circles are the images of the complex field at the pupil with one being the complex conjugate of the other. One of the images is cropped in the spatial frequency domain defined by the red circle. The image is then inverse Fourier transformed back to the spatial domain using equation (6). After the inverse Fourier transform, the hologram is now a reconstructed image. The reconstruction technique assumed that the spherical curvature of the LO and image were the same at the camera, so that there was a Fourier transform relationship between the pupil and image planes. The cross-range reconstructed image was then placed into the dataset, organized by the frequency used to record the hologram.

The hologram reconstruction can be repeated for N frequencies as demonstrated by the purple box in Figure 5. RCH requires holograms from multiple laser frequencies to be recorded in order to create a 3D image. The laser must be tuned to a different frequency for every hologram that is recorded.
Figure 8: A cross-range reconstructed dataset. The figure shows a cross-range reconstructed dataset. The N frequencies is the 3rd dimension with N different frequency recorded holograms.

All of the reconstructed complex fields are now placed into a dataset that is sorted in the 3rd dimension by the laser frequency used for each, illustrated in Figure 8. PGA is applied to the data cube to remove piston phase fluctuations between the various measurements. Then the data cube is symmetrically zero padded to sixteen times the original size of the range dimension. A fast Fourier transform is applied over the 3rd dimension to convert from the spatial frequency domain to the spatial domain.
Figure 9: Range compression of zero padded frequency dataset. The image displays how the dataset is zero padded and compressed in the frequency dimension.

Next, the system was sub-pixel shifted to set the mean range of the target to zero. The flat plate target sub-pixel shift was found by summing in cross-range to give a 1D-range return, which was then inverse Fourier transformed. Next, the 1D signal was up sampled by zero padding and Fourier transforming. The proper shift was determined by finding the distance between the peak location and the center of the array for the 1D up sampled range return. This range offset was then removed by applying the same phase ramp to the frequency realizations of each cross-range pixel, eliminating the difference between the peak location and the center array location. After the data cube was range compressed and shifted, it was multiplied by its complex conjugate to provide the intensity. The green block was repeated in Figure 5 to generate different speckle realizations. The surface roughness on the targets was unique for each of the N speckle realizations.
Finally, the remaining four blocks from the methodology chart were conducted. The data cube was averaged over the speckle realizations to reduce the speckle noise. The surface reconstruction technique used in this paper found the range location of the maximum intensity value for every cross-range pixel. Each location was then shifted so that the center location would have a range of zero, and was then multiplied by the range bin size defined in equation (20).

The reconstructed and reference images were converted from 2D arrays to point clouds by using the cross-range and range information. The next step converted the point clouds into files that the PCL could process using MATLAB code developed by Peter Corke [26]. The reconstructed image was quantitatively compared to the reference image by calculating the mean Euclidian distance of nearest neighbors using the PCL function. The PCL function was executed from MATLAB using the prepared point clouds and the metric files and results were recorded for each of the 3D reconstructions. The technique identified how bandwidth, speckle averages, and signal photon counts affected the image quality by calculating the amount of error present in the 3D reconstruction.
CHAPTER 4

EXPERIMENT

This chapter covers the experimental methodology that was used, the reason each component was chosen, and a review of how each contributed to the system as a whole. It will cover how the experiment was conducted and what system parameters were changed to affect the point cloud metric. Lastly, this chapter will cover the phase noise that was present in the recorded results, what was done to remove it, and what post processing was needed to correct the data.

4.1. Experiment Design

The optical setup of interest for this thesis was a holographic focal plane system with a frequency tunable, coherent illumination source. Figure 10 is a diagram of the system used in the experiment. The red triangles are an illustration of the illuminated light throughout the system. The system consisted of an Indium Gallium Arsenide (InGaAs) camera array, a tunable laser, a fiber splitter, a fiber variable attenuator, a lens, a transmitter (TX), a LO, and a target on a rotation stage.
Figure 10: Experimental setup diagram. The figure shows the experimental setup used for testing using a flat plate target in the laboratory.

The InGaAs camera was a large 1280 by 1024 array infrared imager which allowed for high cross-range detail in the reconstructed holograms. The laser’s central wavelength of 1545.4 nm was tuned to allow frequency diversity to record holograms necessary for range-compression. The 95/5 beam splitter was used to create the LO and TX from the same laser source. The TX was given 95% power due to the amount that was lost to the propagation and divergence of the TX beam and the divergence of the reflected wavefront. The variable attenuator was used to control the signal-to-noise ratio (SNR) of the recorded holograms. The target of interest for the experiment was a Permareflect reference surface with 99% reflectivity at 1.5μm. The flat surface was oriented such that the entire target was within a single range bin. The target was mounted on a rotation stage which was used to create independent speckle realizations.

The bare fibers were used for the transmitter and LO so that the beams hitting the camera and target filled the field of view. The LO was placed at an angle relative to the sensor to generate an off-axis hologram. This allowed for separation of the reconstructed images and the DC components in the spatial frequency domain.
4.2. Methodology

![Methodology Diagram]

Figure 11: Experimental methodology diagram. The figure is the methodology that was used to record the experimental data. The first three boxes were controlled by Labview while the last box was controlled manually.

Labview was used to automate the experimental collection of holograms. Labview VIs for the camera, rotational stage, and laser were previously developed by other researchers. In addition to these Labview interfaces with the equipment, a new VI was created to automate the collection of the holograms. The first step in the experiment was to power on the laser and tune it to the first frequency. Next, the propagation of light through the system created a hologram on the sensor array, which the sensor recorded and saved to a file. Then, the laser frequency was changed by 100 megahertz (MHz). The recording process was repeated until 32 unique frequency holograms were collected.

Labview was used to rotate the target stage to create a new speckle field for the next recorded dataset. A time delay was added after the rotation of the stage to allow the micro movements to stop before recording the next set of data. These micro movements were larger than the wavelength of the illumination source which caused phase errors in the experimental data. The system repeated the collection of a RCH dataset by recording the 32 frequencies for each of the sixteen unique speckle realizations.
After recording each of the sixteen speckle realizations for a single SNR then, the variable attenuator was adjusted until the desired SNR was achieved. The resulting SNR was measured by observing the spatial frequency domain of the hologram in real time and adjusting the variable attenuator while looking at the amplitude of the twin images. The power was summed from one of the twin images to determine the signal strength. Then an area equivalent to the twin images was summed to determine the noise power of the hologram. This area was from a different quadrant of the spatial frequency that only contained noise. Once both values were calculated, equation (31) was used to calculate the SNR of the hologram,

\[ SNR_{RH} = \frac{TI - TN}{TN} \]  

(31)

where \( SNR_{RH} \) was the SNR of the recorded hologram, TI was the sum of the intensity of a twin image in the spatial frequency domain, and TN was the calculated noise in a quadrant where the twin images were absent. Noise was subtracted from the twin image to isolate the signal.

4.3. Noise Isolation

The first set of 3D images had a range dependent phase curvature for 4 and 8 discrete frequency steps and a varied phase tilt for 16 and 32 discrete frequency steps in the reconstructed results that created a need for post processing correction. After observing many different reconstructions, the deviations were present in the reconstructed data.
One hypothesis for the appearance of the deviations in the reconstructed images were that components of the system were moving or breathing. To remedy this, the fibers were mounted and stabilized. The experiment was repeated to confirm the removal of the deviations due to the new mounts. However, there was little improvement of the reconstructed images. An alternate mitigation strategy was to reduce the amount of time taken to record the holograms. This was accomplished by first recording all the frequency components before generating a new speckle realization. A new laser was integrated into the system to more rapidly change frequencies. Unfortunately, short integration time did not significantly improve the results. Therefore, the deviations were not due to the breathing of the system.

The cause of the deviations were discovered to be significant airflow going through the optical system. The server in the laboratory and the heating, ventilation, and air conditioning (HVAC) system both aimed at the experiment’s optical axis, which caused a large amount of turbulence in the system that was not expected. The approach to remove the turbulence from the experimental setup was completed by enclosing the experimental system within a coffin. This coffin was an acrylic outer shell to block any atmospherics that was aimed at the optical axis. Before enclosing the system, the acrylic was tested to see if it was reflective in the infrared spectrum. The acrylic was reflective and to reduce the optical noise in the system from light reflecting off the acrylic, an infrared absorptive fabric was used to line the inside of the acrylic. Another set of data was taken with the coffin containing the experimental setup and the results showed a large improvement in removing the range bowl and varied phase tilt.
4.4. Post Processing

For the experimental setup there were additional errors that must be taken into account. Three corrections were applied to the experimental data to align the results with the simulated data: removal of the tilt, PGA (Chapter 2), and sub-pixel imaging correction. The rotation of the stage used to create speckle realizations also created a tilt in the reconstructed images. The tilt was calculated by observing the offset from the flat plate using the dataset with the highest SNR for individual speckle realization. The offset was used to correct for the tilt. The PGA for correcting the Permaflect reference flat plate used only G through K. Steps A through H were not needed because the target was already a flat plate.

The sub-pixel range correction was conducted by compressing the holograms after zero padding and summing in cross-range. Next, the data was uncompressed while maintaining the zero padding. To improve accuracy in locating the peak the 1D array was further up-sampled by Fourier transforming, zero padding, and inverse Fourier transforming. The location of the resulting peak relative to the center position then gave the desired sub-pixel shift that was then used on the experimental data to center the range returns.
CHAPTER 5

RESULTS AND DISCUSSION

This chapter covers the results that were collected from the simulation and the experiment. The results were produced for two targets, a flat plate and a backhoe. The flat plate represented a 1D range compressed target and the backhoe demonstrated cross-range and range diversity. The simulated and experimental results for the flat plate were compared to validate the simulation using the point cloud target. This chapter displays the results for the simulated backhoe to observe how cross-range and range diversity effected the point cloud metric.

5.1. Signal Photons

Signal photons per cross-range pixel are defined as the total amount of laser photons returning from the target and collected by the receiver across all reconstruction holograms. This definition of signal photons allowed the results to be compared for different bandwidths and varying amounts of speckle averaging in each of the reconstructed 3D images. The CRLB was redefined to account for the dependency on signal photons rather than SNR. For a shot noise limited image, the SNR for a single hologram is equivalent to the number of signal photons [27]. This allows the conversion of the CRLB as seen in equation (32).
\[ CRLB = \frac{3 \cdot c^2}{8 \cdot \pi^2 \cdot SNR \cdot N_{\text{speckle}} \cdot B^2} = \frac{3 \cdot c^2}{8 \cdot \pi^2 \cdot N_{\text{signal photons}} \cdot B^2} \] (32)

5.2. Flat Plate Simulation and Experimental Data

The simulation and experiment produced 3D images for bandwidths of 400 MHz, 800 MHz, 1600 MHz, and 3200 MHz in discrete steps of 100 MHz. For each bandwidth, speckle averaging was varied using one, two, four, eight, and sixteen independent realizations.

Figure 12: The reference point cloud of the flat plate target. The figure is an image of the reference flat plate target that was created for simulation and experimental testing. The target was 0.02164 meters wide with a range ambiguity of 1.5 meters.

Figure 12 displays the reference image used to create the simulated targets. This was also used as the reference point cloud to which the reconstructed holograms were compared for both simulated and experimental results. The target did not display interesting
qualitative results, but allowed for an analysis in a single dimension using the point cloud metric.

In the following plots, the number of speckle averages and the frequency bandwidths are varied independently. The circles correspond to experimental data points while the dashed line marks the trend line of the experimental points. In addition, the simulated data is shown by the solid lines of the same color as the corresponding experimental data points. The thick green lines signify the upper bound from the variance of a uniform distribution over the range ambiguity for the system and the thick pink lines is the CRLB for the largest bandwidth included in the specific plots.
Figure 13: Effects of speckle averaging on a flat plate target with 400 MHz bandwidth. The plot displays the effect that speckle averaging had on a flat plate target with a bandwidth of 400 MHz by using the point cloud metric to calculate the MSE. The circles are the experimental data points, with the dashed lines being the trend of the experimental data for each amount of speckle averages. The solid lines are the simulated data, last the different colors represent different amount of speckle averages used for reconstruction.

The first plot, figure 13, displays how speckle averaging effected the MSE of the reconstructed image with a bandwidth of 400 MHz. The results show that increasing the amount of speckles averages will lower the MSE. The other big thing to note on the plot is that for 16 speckle averages, the last point drops below the CRLB as well as the simulated data. One reason why this may have happened is that the experimental reconstructed data point could have had some cross-range speckle averaging which would cause the unbiased estimated of the reconstructed image to become a biased estimator. The CRLB has a condition that the estimator must be unbiased, when the
estimator is biased the variance can be lower. The reason why the simulated data drops below the CRLB is due to the bias in the estimator from center shifting the range returns and the digitization that biases the answer towards a range of zero, and as a consequence zero error.

Figure 14: Effects of speckle averaging on a flat plate target with 800 MHz bandwidth. The figure displays the results for different amounts of speckle averages for 800 MHz bandwidth for the flat plate target. The details of the plot are described in Figure 13.

Figure 14 displays how speckle averaging effected the bandwidth of 800 MHz. It can be seen that the experimental data agrees very well with the simulated data at lower speckle averages and starts to diverge as the amount of speckle averages increase. While the data points diverge, they still keep the same trend.
Figure 15: Effects of speckle averaging on a flat plate target with 1600 MHz bandwidth. This plot shows the results for the MSE of the flat plate target for 1600 MHz bandwidth. The figure is explained in detail in Figure 13.

The plot displays the results for how speckle averaging affected the MSE for 1600 MHz bandwidth. The interesting observation of the plot is that at the low signal photon region, the greater amount of speckle averages had a greater MSE than the lower speckle averages at the same amount of signal photons. This could have happened because the recorded holograms had a SNR of less than 1. When a hologram has a SNRs less than 1 the reconstruction of the hologram has different effects then a SNR greater than 1. This effect happened with the different amount of speckle averages because the greater amount of speckle averages had less amount of signal photons for each of the recorded holograms. This means that the SNR of each recorded hologram will go down as the amount of speckle averages increases.
Figure 16: Effects of speckle averaging on a flat plate target with 3200 MHz bandwidth. This is the final plot that follows Figure 13. This plot has a bandwidth of 3200 MHz instead of the 400 MHz bandwidth Figure 13 has displayed.

In figure 16 the plot displays how changing the amount of speckle averages for a bandwidth of 3200 MHz effected the MSE. When observing the lines on the plot it can be seen that the experimental data and simulated data both start to converge to a single MSE value. The experimental data had an unknown deviation of the flat plate assumption, which could have been a tilt on the object of resulting in 1.25mm of range change varying linearly as a function of cross-range location. The simulation was adjusted to have this amount of uncorrected tilt to better match the experimental results, giving the experiment and simulation have the same saturation value.
The next section of results will be for the target of the flat plate, but will be going over how different bandwidths effected the MSE of the system. The bandwidths that will be observed in the results are 400 MHz, 800 MHz, 1600 MHz, and 3200 MHz with 100 MHz step in frequency for each of the recorded holograms. The plot provided will display how bandwidths effect 1, 2, 4, 8, and 16 speckle averages.

![Mean Squared Error for 1 Speckle Average](image)

Figure 17: Effects of bandwidth on a flat plate target for 1 speckle average. The plot displays the effects that bandwidth had on a flat plate target for a single speckle average. The results were for how much MSE the reconstructed range compressed hologram had compared to the reference point cloud. The circles are experimental data points, the dashed lines are the trends for the experimental data. The solid lines are the simulated data and the colors corresponds to which bandwidth was used for the experimental and simulated data.

The figure above displays how increasing the bandwidth effected a RCH reconstruction with a single speckle average. It can be seen for a single speckle average that increasing
the bandwidth has no effect on the MSE unless the reconstruction has a high amount of signal photons. The CRLB for this plot and all of the plots to follow for the flat plate target is for the bandwidth of 3200 MHz.

Figure 18: Effects of bandwidth on a flat plate target for two speckle averages. The plot analyzes the effects of increasing bandwidth on two speckle averages for a flat plate target. The description to the figure is described in Figure 17.

The figure displays how bandwidth effected the MSE for the 3D images for two speckle averages. The results for two speckle averages displays that bandwidth does influence the MSE, which will improve the reconstructed 3D image. The results did not accomplish reaching the lowest variance of the CRLB, which means that the system speckle noise did not allow for the variance to achieve the lower bound.
Figure 19: Effects of bandwidth on a flat plate target for four speckle averages. The figure displays the results for how bandwidth effects the MSE of a flat plate target for four speckle averages. The readability is described in detail in Figure 17.

Now, the figure displayed is for how bandwidth effected the MSE for four speckle averages for a flat plate target. The simulated data at 3200 MHz has a close point to the CRLB, this point would give the best results over all the other simulated data results in this plot. The reason for this point being the best is from being closest to the CRLB with the least amount of signal photons used, allowing for the best reconstruction with the least amount of energy needed. The experimental data was also close to the CRLB.
Figure 20: Effects of bandwidth on a flat plate target for eight speckle averages. The figure shows the effects that bandwidth has on the MSE for a flat plate target for eight speckle averages. The key is described in Figure 17.

The plot displayed is for how different bandwidth effected the MSE for eight speckle averages. The result to note on the figure is that in the low signal photon region the larger bandwidths had a greater MSE than the smaller bandwidths at the same amount of signal photons. This effect may have happened because when reconstructing the surfaces with low amount of signal, the larger bandwidths have a greater number of uncorrelated noise samples. Having a greater number of noise samples means that the probability of picking a noise peak as opposed to a signal peak will increase, which will cause the MSE to increase.
Figure 21: Effects of bandwidth on a flat plate target for sixteen speckle averages. This is the last plot that was described by Figure 17. It has sixteen speckle averages for the reconstruction of the flat plate target.

The final plot of the flat plate target analyzes how bandwidth effected the MSE for sixteen speckle averages. To note how powerful the nonlinear reconstruction of finding the location of the peak intensity for every cross-range pixel with speckle averaging is, look at the lowest experimental signal photon for 3200 MHz bandwidth. This reconstruction is close to being on the CRLB with a very low amount of signal photons per recorded hologram; the 90 signal photons were split over 512 recorded holograms to reconstruct the experimental dataset. The results of this reconstruction showed that the system could reconstruct a target with less than one signal photon per recorded hologram cross-range pixel.
5.3. Validation of Simulated Data

As displayed in the nine figures above, the results of the flat plate experimental data and simulated data demonstrated that the simulation and experiment maintain the same trends. The values of the simulated and experimental data diverged due to an unknown deviation from the flat plate assumption for the experimental data; however, the simulated data continued to have the same trend while varying the amount of signal photons contained in the 3D reconstructed image. If the unknown deviation was removed, it would be likely that increasing the amount of speckle averages would decrease the MSE for a flat plate target, as well as increasing the bandwidth would reduce the MSE. If the unknown deviation was removed from the figures above, the results would likely follow the CRLB in the high signal photon region.

5.4. Backhoe Results

The second target, the backhoe, was cross-range diverse, range diverse, and cross-range limited. Therefore, the point cloud metric had dissimilar results from the previous target as it did not take into account the effects of cross-range or 3D errors. The backhoe simulation was conducted over bandwidths of 400 MHz, 800 MHz, 1600 MHz, 3200 MHz, 6400 MHz, and 12800 MHz while also covering 1, 2, 4, 8, 16 speckle averages for every bandwidths. The simulation will had a discrete step of 100 MHz.
Figure 22: The reference point cloud for the backhoe target. The picture is the reference target that was used for the backhoe point cloud. The cross range was 3 meters and the range ambiguity was 1.5 meter. The range dimension of the backhoe was reduced to keep the object within a reconstructed volume. This point cloud was used to calculate the MSE of the reconstructed range compressed holograms.

The target (Figure 22) was modified to fit within a single range ambiguity and the cross-range limits of the system for the allowance of a reconstruction that would work within the system. Using more than one range ambiguity would require unwrapping, which the problem being solved in this thesis is to find a holistic metric for RCH and find the most effective use of system parameters to reconstruct the hologram. Unwrapping is not a parameter, but a technique to reconstruct a hologram that has a range diversity greater than the system range ambiguity.
The backhoe target was simulated not only to understand the effects that cross-range and range diversity have on the MSE, but also to gather the most effective use of system parameters. The plots displayed in the next section will cover how changing the amount of speckle averages for a defined bandwidth effected the MSE for the backhoe target.
Figure 23: Effects of speckle averages on the backhoe target for 400 MHz bandwidth. This figure displays the effects speckle averaging had on the reconstruction of the backhoe target for 400 MHz bandwidth. The lines plotted are for different amounts of completed speckle averaging. The lowest observed MSE was found by looking at all of the MSE values for the reconstructed backhoe target over all bandwidths, speckle averages, and signal photons. The upper bound was uniform distribution with range ambiguity and the lower bound is the CRLB for 400 MHz bandwidth.

Figure 23 above contains the results of the backhoe target for four discrete frequencies. The plot displays the effects of speckle averaging on a range diverse target. The black line demonstrates the lowest observed MSE for all reconstructed holograms at all speckle averages, bandwidths, and signal photons. The lowest MSE was calculated from the
sixteen discrete frequencies and sixteen speckle averages. However, calculating the MSE did not reach the lowest observed limit, nor did it efficiently provide the best reconstruction for four discrete frequencies. Also, an evident trend was the continuous increase of speckle averages and continuous decrease to the lowest observed MSE.

Figure 24: Effects of speckle averages on the backhoe target for 800 MHz bandwidth. The figure has similar attributes as Figure 23, but has a bandwidth of 800 MHz.

This plot displays the effects of speckle averaging at 800 MHz bandwidth with eight discrete frequency steps. By comparing the results of this plot to the previous plot, it demonstrated a large decrease in the MSE when frequencies present were doubled. Also
evident was the effect of a continuous increase of speckle averages and the decreased error in the reconstructed image.

Figure 25: Effects of speckle averages on the backhoe target for 1600 MHz bandwidth. The figure is the same as Figure 23, but instead of speckle averaging on 1600 MHz bandwidth.

The results displayed in Figure 25 are the effects speckle averaging had at 1600 MHz bandwidth. The plot displays the lowest observed bound which occurred at sixteen speckle averages noted as the thin purple line in the plot.
Figure 26: Effects of speckle averages on the backhoe target for 3200 MHz bandwidth. The figure displays the effect of speckle averaging has for 3200 MHz bandwidth. The detailed description for the figure is explained in Figure 23.

Figure 26 displays the effects that speckle averaging had on the 3200 MHz bandwidths. An effect to note was an increase in speckle averages decreased the error in the reconstructed hologram until it reached the lowest observed error. A greater amount of speckle averaging would be needed to test whether additional speckle averages would reduce the error of the reconstructed images.
Figure 27: Effects of speckle averages on the backhoe target for 6400 MHz bandwidth. The figure displays the simulated results for viewing the effects of speckle averaging on the MSE with 6400 MHz bandwidths. The detailed description is in Figure 23.

Figure 27 represents the results which improved after increasing the amount of speckle realizations until a lower limit was reached. With the CRLB as the variance for 1D range return, it was noted that it is not the limiting factor on the system’s calculated error. The variance error calculated for the backhoe target was that of a 3D reconstructed hologram which causes a greater amount of error than the expected 1D reconstructed hologram. Therefore, the lowest observed limit of the system was limited by the cross-range of the system, not solely the range error present in the reconstructed holograms.
Figure 28: Effects of speckle averages on the backhoe target for 12800 MHz bandwidth. This is the last plot displaying how speckle averaging effects the MSE of the reconstructed backhoe. This plot displays the MSE for 12800 MHz bandwidth with a detailed description in Figure 23.

Figure 28 above displays the effects speckle averaging had on 12800 MHz bandwidth for a reconstructed 3D image. Opposite of expectations, yet noteworthy, was that the lower speckle averages provided a lower error in the low signal photon region than would a larger amount of speckle averages for the same amount of signal photons. However, the recorded holograms for these results had a SNR less than one, which, when compared to a SNR greater than one, can cause different effects on the reconstruction. Additionally,
with low signal photons and a low amount of speckle averages, the holograms used for RCH had a recorded hologram with a SNR greater than the higher speckle averages. This was because both systems would have the similar signal photon amounts and dividing those by additional recorded holograms would cause the SNR amounts of the individual recorded holograms to decrease.
Figure 29: Effects of bandwidth on the backhoe target for one speckle average. The plot displays the effects that bandwidth had on the reconstructed 3D images of the backhoe target for a single speckle average. The lines on the graph indicate different bandwidths used to reconstruct the target. The lowest observed MSE was found by observing all reconstructed MSEs over bandwidth, speckle averages, and signal photons to find the point with the smallest MSE. The uniform distribution was calculated over the range ambiguity and the CRLB using the largest bandwidth of 12800 MHz.

Figure 29 displays how different bandwidths effected the error in a single speckle reconstruction. Figure 17 has the same amount of speckle averages for the flat plate target, but does not have any significant change in MSE from increasing bandwidth. This is not the case for a single speckle realization for a range diverse, cross-range limited
There is a great effect for increasing the amount of bandwidth for a single speckle realization, range diverse, and cross-range limited target.

![Mean Squared Error for 2 Speckle averages](image)

**Figure 30:** Effects of bandwidth on the backhoe target for two speckle averages. The figure displays the MSE result for the reconstructed backhoe target for two speckle averages over many bandwidths. The lowest observed MSE is described in Figure 29.

Figure 30 shows the results for how bandwidth effected the backhoe target’s MSE for two speckle averages. The results for two speckle averages demonstrated that the system was not yet saturated by its limits and increasing the bandwidth by a factor of two exponentially decreased the MSE for the same amount of signal photons.
Figure 31: Effects of bandwidth on the backhoe target for four speckle averages. The figure displays the results for the reconstruction of the backhoe target for four speckle averages over many bandwidths with the lowest observed MSE described in Figure 29.

Figure 31 displays the effects that bandwidth had on a four speckle realization range diverse, cross-range limited target. The results of the plot show that there was a significant improvement for increasing the bandwidth to 1600 MHz, but the same amount of improvement was not noticeable when increasing the bandwidth past 1600 MHz. Also, the CRLB drifted away from the calculated error of the simulated target due to the cross-range resolution limitations while the CRLB is the variance for a 1D range returned target.
The CRLB was not an effective lower bound for 3D reconstructed images due to the effects of shifting from a 1D range to a 3D imaging object.

Figure 32: Effects of bandwidth on the backhoe target for eight speckle averages. The plot shows how the MSE was effected by changing the bandwidth for the backhoe target with eight speckle averages. The lowest observed MSE is described in Figure 29.

Figure 32 displays the results from eight speckle realizations with different amounts of bandwidth used for the simulation. The results showed that only the 400 MHz bandwidth did not have similar results as the other bandwidths. Therefore, it was of value to double the bandwidth to eight discrete steps, at minimum, so a low error could be achieved on...
the reconstructed 3D images. Also, at the low signal photon region, a lower bandwidth produced a lower error rather than a higher bandwidth at the same signal photon amount. A greater amount of discrete points chosen for the reconstructed surface in the larger bandwidth created this inverse effect at the low photon region; thus, permitting a greater probability of the incorrect surface to be chosen during the reconstruction.
Figure 33: Effects of bandwidth on the backhoe target for sixteen speckle averages. The final plot describes how bandwidth effected the backhoe target for sixteen speckle averages with the lowest observed MSE explained in Figure 29.

Figure 33 displays a plot for the effects of bandwidth on sixteen speckle averages for the backhoe target. The figure confirmed that the 400 MHz bandwidth would not work well unless the amount of signal photons is reduced to less than ten signal photons.

Through all of the figures a final statement could be concluded, but creating a concise plot to explain how speckle averaging and bandwidth can be effectively used would be a better result. This said plot was produced in the next figure.
Figure 34: Efficient use of signal photons for speckle averages and resolution of a backhoe target. The figure displays the most efficient RCH reconstructions for bandwidth and speckle averages by calculating how many signal photons are needed to gain a MSE twice the lowest observed MSE for every cross-range/range resolution and speckle averages. The cross-range/range resolution is a different way of defining bandwidth, with 0.125 equivalent to 400 MHz bandwidth, 0.25 equivalent to 800 MHz bandwidth, 0.5 equivalent to 1600 MHz bandwidth and continuing this trend until 4 is equivalent to 12800 MHz bandwidth. This plot has speckle averages on the x-axis and cross-range to range resolution on the y-axis. The lower amount of signal photons the better because the less signal photons needed the less laser power was needed to interact with the target for the same quality reconstructed hologram.

Figure 34 presents the necessary amount of signal photons for RCH reconstructions with a MSE equal to twice the lowest observed MSE for each speckle average and bandwidth, or cross-range per range resolution. This allowed for the discovery of the most signal photon efficient reconstruction for a particular target. A lower signal photon amount is desirable due to the need for less transmit energy in the production of an image. The
dark red in the figure means either the reconstruction took more than 250 signal photons to reach twice the lowest MSE observed value or it never reach twice the lowest MSE observed. The best cross-range to range resolution ratio was found to be 0.5. This corresponded to a bandwidth of 1600 MHz for the backhoe target reconstruction. The amount of signal photons it took for the reconstruction between seven to sixteen speckle averages was 46.7 to 56.5. While a 0.5 ratio of cross-range to range resolution gave the least amount of signal photons needed for reconstruction, an equal voxel or a 1 to 1 ratio of cross-range to range resolution reduced the range resolution by half with a small increase in the amount of signal photons needed to reconstruct 7 to 16 speckle averages, requiring between 56.5 and 68.3 signal photons. The equal voxel was expected to be the best reconstruction cross-range to range resolution, but most likely, it did not have the best reconstruction due to a lack of zero padding in the cross-range while zero padding in the range. The lack in cross-range zero padding may have caused the 0.5 cross-range to range resolution be the best reconstruction case.

5.5. Validation of Quality Metric

Based on the flat plate target results, the upper and lower bound were correctly chosen to represent the point clouds metric for the MSE of a 1D reconstructed target. This validated that the point cloud metric worked within the theoretical upper and lower variance bounds. The simulation and experimental data results had similar trends, despite an unknown deviation from the flat plate assumption for the experimental data. The deviation could have been a physical error produced in the laboratory that was
unaccounted for in the simulation. The flat plate target’s data indicated trends that confirm the simulation and experiment are equivalent to one another. The results produced from the backhoe target simulation revealed how the point cloud metric would produce results for a cross-range limited, range diverse target. The results do not follow the CRLB due to cross-range diversity that is unaccounted for in the CRLB definition based on a 1D radar range return.
CHAPTER 6

CONCLUSION

RCH has not had a holistic metric to analyze the reconstruction of a target. The previous ability to determine the quality of the reconstructed image was from system parameters like resolution in each of the directions and SNR. This thesis proposed the use of a point cloud metric to calculate a single value for the quantification of results from RCH. The point cloud metric was validated through simulations and experimental testing.

The first simulation was conducted on a flat plate target that allowed the achievement of the CRLB conditions for point cloud metric validation. The results of the simulated data for the flat plate target agreed with the CRLB at the mid-signal photon region. The low signal photon region did not reach the CRLB because the recorded holograms did not have enough SNR to achieve the lower bound. The low signal photon region had a MSE close to the uniform distribution. The results from the simulations’ high signal photon regions deviated below the CRLB, due to errors less than the size of a range bin. The last section of the simulated data displayed asymptotic behavior because the unknown deviation from the experimental data was added to the simulated results.

To validate the simulation, the simulated data was compared to the experimental data. The experimental data had an unknown deviation of the flat plate assumption and was offset from the simulated data. The results from the simulation and experimental data
concluded that the point cloud metric was accurate and provided a quantitative metric to compare reconstructed 3D images.

The backhoe target was simulated to analyze the effects of range and cross-range diverse target on the point cloud metric. The results from the simulation displayed that there is a MSE lower limit greater than the CRLB. The lowest observed error for the backhoe target, found by observing the amount of signal photons necessary to reach twice the lowest observed MSE, led to the discovery of the most efficient use of signal photons for a range diverse target.

This thesis has demonstrated the use of a point cloud metric as a holistic metric to quantify 3D images formed from coherent data. Using the point cloud metric, the simulated data and experimental data presented that, when reconstructing a 3D image for a target without range diversity, the system should continuously reduce the MSE for more speckle averages, more signal photons, and larger bandwidth. The results showed that reducing the size of the range bin, increasing speckle averages, and increasing signal photons allowed for greater accuracy in correct range bin choice to eventually zero the MSE. The range and cross-range diverse backhoe target had the lowest MSE for discovering the most efficient use of signal photons for the target: a bandwidth that results in a range resolution twice the size of the cross-range resolution with seven to sixteen speckle averages.
6.1. Original Contributions

- Found and applied a relevant point cloud metric to test on 3D coherent data
  - Used the `getFitnessScore` function from the PCL as a metric and applied the function to RCH.
- Applied upper and lower bounds for the performance of the point cloud metric
  - Found the variance of uniform distribution to be the upper bound for the 3D reconstructed RCHs.
  - Found the CRLB for 1D range radar returns for 1D RCH.
- Validated flat plate target through simulation and experimental data
  - Confirmed that the flat plate, a 1D range target, fit within the upper and lower bounds
  - The trends for the simulated and experimental data agreed leading to the use of other targets in MSE system estimation.
- Analyzed the most signal photon efficient way to form reconstructed surfaces for RCH as a function of bandwidth and speckle realizations
  - Found the most efficient reconstruction of a diverse target is having a range resolution twice that of the cross-range resolution with seven to sixteen speckle averages.
6.2. Future Work

- Develop a unique surface reconstruction technique for range compressed holography
- Apply the quality metric to different 3D coherent reconstruction techniques
- Apply the quality metric to different targets with varied characteristics to calculate the link budget for the most photon efficient reconstruction
- Find an analytical equation to describe the signal photons versus variance curve


[3] B. P. Hilderband and K. A. Haines, "Multiple-Wavelength and Multiple-Source


   Aperture Ladar," *OSA Publishing*. 


APPENDIX A

MATLAB Code to Simulate the Point Cloud Metric on a Flat Plate Target Using Range Compressed Holography

The code in the appendixes is for simulating a flat plate target using MATLAB. The first four sections are the functions made or found by other users. The 'getFitnessScore hologram for' is the C++ function that converted into an executable. The ft2 and ift2 functions are 2D Fourier and inverse Fourier transforms respectively. The ft2 and ift2 functions have correct shifting and inverse shifting to calculate the Fourier transform.

A.1. eigencorrection

function [correctedcube] = eigencorrect(datacube)
    amountoffreq = length(datacube(1,1,:));
    correctedcube=zeros(size(datacube,1),size(datacube,2),amountoffreq);
    % calculating the synthetic wavelengths
    synthcube = datacube(:,:,1:end-1).*conj(datacube(:,:,2:end));
    %calculating the phase difference between holograms
    phase(:,1,1) = angle(sum(sum(synthcube,1),2));
    phaser = ones(amountoffreq,1);
    correctedcube(:,1) = datacube(:,1);
    %applying a phase gradient correction
    for i=2:amountoffreq
        phaser(i,1) = phaser(i-1,1).*exp(1j.*phase(i-1,1));
        correctedcube(:,i) = datacube(:,i).*phaser(i,1);
    end
end

A.2. objectpcd
function [] = objectpcd(object,N,dxy)
    intout = 1;
    output = zeros(N.*N,3);
    [x, y] = meshgrid(-N/2:N/2-1);
    %converting the image to a point cloud
    for intx = 1:1:N
        for inty = 1:1:N
            output(intout,1) = x(1,intx).*dxy;
            output(intout,2) = y(inty,1).*dxy;
            output(intout,3) = object(intx,inty);
            intout = intout + 1;
        end
    end
    output = transpose(output);
    %saving the output as a .pcd file
    savepcd('object.pcd',output)
    clear output;
end

A.3. imagepcdone

function [] = imagepcdone(image,N,dxy)
    intout = 1;
    output = zeros(N.*N,3);
    [x, y] = meshgrid(-N/2:N/2-1);
    %converting the image to a point cloud
    for intx = 1:1:N
        for inty = 1:1:N
            output(intout,1) = x(1,intx).*dxy;
            output(intout,2) = y(inty,1).*dxy;
            output(intout,3) = image(intx,inty);
            intout = intout + 1;
        end
    end
    output = transpose(output);
    %saving the output as a .pcd file
    savepcd('image1.pcd',output)
    clear output;
end

A.4. savepcd

%SAVEPCD Write a point cloud to file in PCD format
%
SAVEPCD(FNAME, P) writes the point cloud P to the file FNAME as an
as a PCD format file.

SAVEPCD(FNAME, P, 'binary') as above but save in binary format. Default
is ascii format.

If P is a 2-dimensional matrix (MxN) then the columns of P represent the
3D points and an unorganized point cloud is generated.

If M=3 then the rows of P are x, y, z.
If M=6 then the rows of P are x, y, z, R, G, B where R,G,B are in the
range 0 to 1.
If M=7 then the rows of P are x, y, z, R, G, B, A where R,G,B,A are in
the range 0 to 1.

If P is a 3-dimensional matrix (HxWxM) then an organized point cloud is
generated.

If M=3 then the planes of P are x, y, z.
If M=6 then the planes of P are x, y, z, R, G, B where R,G,B are in the
range 0 to 1.
If M=7 then the planes of P are x, y, z, R, G, B, A where R,G,B,A are in
the range 0 to 1.

Notes::
- Only the "x y z", "x y z rgb" and "x y z rgba" field formats are currently
  supported.
- Cannot write binary_compressed format files
See also pclviewer, lspcd, loaddpcd.

Copyright (C) 2013, by Peter I. Corke

TODO
- option for binary write

function savepcd(fname, points, binmode)
  % save points in xyz format
  % TODO
  % binary format, RGB

  ascii = true;
  if nargin < 3
    ascii = true;

else
    switch binmode
    case 'binary'
        ascii = false;
    case 'ascii'
        ascii = true;
    otherwise
        error('specify ascii or binary');
    end
end

fp = fopen(fname, 'w');

% find the attributes of the point cloud
if ndims(points) == 2
    % unorganized point cloud
    npoints = size(points, 2);
    width = npoints;
    height = 1;
    nfields = size(points, 1);
else
    width = size(points, 2);
    height = size(points, 1);
    npoints = width*height;
    nfields = size(points, 3);
end

% put the data in order with one column per point
points = permute(points, [2 1 3]);
points = reshape(points, [], size(points,3))';

switch nfields
    case 3
        fields = 'x y z';
        count = '1 1 1';
        typ = 'F F F';
        siz = '4 4 4';
    case 6
        fields = 'x y z rgb';
        count = '1 1 1 1';
        if ascii
            typ = 'F F F I';
        end
end
else
    typ = 'F F F F';
end
siz = '4 4 4 4';
case 7
    fields = 'x y z rgba';
    fields = 'x y z rgb';
    count = '1 1 1 1';
    if ascii
        typ = 'F F F I';
    else
        typ = 'F F F F';
    end
    siz = '4 4 4 4';
end

% write the PCD file header
fprintf(fp, '# .PCD v.7 - Point Cloud Data file format\n');
fprintf(fp, 'VERSION .7\n');

fprintf(fp, 'FIELDS %s\n', fields);
fprintf(fp, 'SIZE %s\n', siz);
fprintf(fp, 'TYPE %s\n', typ);
fprintf(fp, 'COUNT %s\n', count);

fprintf(fp, 'WIDTH %d\n', width);
fprintf(fp, 'HEIGHT %d\n', height);
fprintf(fp, 'POINTS %d\n', npoints);

switch nfields
    case 3

    case 6
    % RGB data
    RGB = uint32(points(4:6,:)*255);
    rgb = (RGB(1,:)*256+RGB(2,:))*256+RGB(3,:);
    points = [ points(1:3,:); double(rgb)];

    case
case 7
  % RGBA data
  RGBA = uint32(points(4:7,:)*255);
  rgba = ((RGBA(1,:)*256+RGBA(2,:))*256+RGBA(3,:))*256+RGBA(4,:);

  points = [ points(1:3,:); double(rgba) ];
end

if ascii
  % Write ASCII format data
  fprintf(fp, 'DATA ascii
');

  if nfields == 3
    % uncolored points
    fprintf(fp, '%f %f %f
', points);
  else
    % colored points
    fprintf(fp, '%f %f %f %d
', points);
  end
else
  % Write binary format data
  fprintf(fp, 'DATA binary
');

  % for a full color point cloud the colors are not quite right in pclviewer,
  % color as a float has only 23 bits of mantissa precision, not enough for
  % RGB as 8 bits each

  % write color as a float not an int
  fwrite(fp, points, 'float32');
end

fclose(fp);
end

[26]

A.5. Simulation of the Flat Plate Target

% THIS SCRIPT WAS DEVELOPED TO IMPORT IMAGES OF THE FLAT PLATE TARGET TO
% SIMULATE THE RETURNED HOLOGRAM, RECONSTRUCT THE RANGE COMPRESSED
% HOLOGRAM OVER MANY DIFFERENT SNRS, BANDWIDTHS, AND SPECKLE AVERAGES,
%AND APPLY THE POINT CLOUD METRIC TO THE RECONSTRUCTED HOLOGRAMS.  
% Declaring all global constants  
c = 299792458;  
N = 4096;  
N_image = 1024;  
zo = 1.44;  
zO = 1.44;  
f = 0.5;  
waist_of_fiber = 6e-6;  
lambda_center = 1545.4e-9;  
[x, y] = meshgrid(-N/2:N/2-1);  
dxyS = 12.5e-6;  
averaging_num = 16;  
frequency_step = 0.1e9;  
totalsnrint = 1;  
totalsnrend = 2^14;  
reflectivity = 0.99;  
snrstep = 128;  
int2 = 1;  
zi = 0.8;  
amountoffreq = 4;  
totalsnr = logspace(log10(totalsnrint),log10(totalsnrend),snrstep);  
Title = 'Flat Plate';

% OF THE FLAT PLATE TARGET THAT ARE LOADED MUST HAVE A DIFFERENT,  
% BUT EQUAL DISTRIBUTION OF SURFACE ROUGHNESS  
% CREATING A PRISTINE TARGET  
z_scene = zeros(N,N,averaging_num);

% APPLYING SURFACE ROUGHNESS TO THE PRISTINE TARGETS  
for average = 1:1:averaging_num  
    z_scene(:,:,average) = z_scene(:,:,average)+50e-6*randn(size(z_scene(:,:,average)));  
end

%Calculating all of the cross-range step sizes  
[X, Y] = meshgrid(-N_image/2:N_image/2-1);  
dxyL = lambda_center.*zi/(N.*dxyS);  
dxyO = lambda_center.*zo/(N.*dxyL);  
dxyR1 = lambda_center.*zi./(N_image.*dxyS);  
dxyR = lambda_center.*(zo)./(N_image/2.*dxyR1);  
z = zeros(N,N,averaging_num);  
lambertian = zeros(size(nz));  

% calculating the reflectivity of the target  
for i = 1:1:averaging_num  
    [~,~,nz(:,:,i)] = surfnorm(z_scene(:,:,i)/dxyO);  
    lambertian(:,:,i) = reflectivity./pi.*nz(:,:,i);  
end

nz = [];  
circ = zeros(N,N);  
circ = circ + double(sqrt((x).^2+(y).^2)<(N_image/2+1));  
crop = double(sqrt((X+256).^2+(Y+256).^2)<129);  
ref_angle = lambda_center/4./12.5e-6;  
getfitstore = zeros(averaging_num,snrstep,amountoffreq);  
variancestore = zeros(averaging_num,snrstep,amountoffreq);  
for num_frequencies = 2.^((2:1:amountoffreq) + 1)
int99 = 1;
snrstart = totalsnrstart:1:snrstep;
  snrend = snrstep;
  snrlength = length(1:1:snrstep);
image_size = 512;
int = 1;
range_padding = num_frequencies.*16;
dz = c./(2.*range_padding.*frequency_step);
storage_cube = zeros(image_size,image_size,averaging_num,...
  snrlength);
lambda2 = zeros(1,1,num_frequencies);

%Calculating all of the frequencies that will be used for range
%compressed holography
for i = 1:num_frequencies
  lambda2(1,1,i) = c./(c./lambda_center + ... 
    (i - num_frequencies/2 - 1).*frequency_step);
end

%Calculating the Gaussian beam at the target
q_gauss_obj = 1/(1i.*pi.*waist_of_fiber.^2./lambda2+zO);
gauss_obj_waist = sqrt(-1.*lambda2./(pi.*imag(q_gauss_obj)));
scene_ill = abs(exp(-repmat((x.*dxyO),...
  [1 1 num_frequencies]).^2+repmat((y.*dxyO),...
  [1 1 num_frequencies]).^2)/repmat(gauss_obj_waist,...
  [N N 1]).^2).^2;
plane_wave2 = [ ];
for speckleloop = 1:1:averaging_num
  speckle_size = 1;
  reflected_wave = zeros(N,N,num_frequencies,speckle_size);

  %REFLECTING THE WAVE OFF THE TARGET
  for i = 1:1:speckle_size
    reflected_wave(:,:,i) = single(repmat(lambertian...
     (:,:,i+(speckleloop-1)),[1 1 num_frequencies]).*...
      scene_ill.*exp(1j*2*pi./repmat(lambda2,[N N 1])...
      .*2.*repmat(z_scene(:,:,i+(speckleloop-1)),...
      [1 1 num_frequencies])));
  end

  %PROPAGATE THE REFLECTED WAVE TO THE PUPIL PLANE
  z_propagated = ft2(reflected_wave);
  reflected_wave = [];
%CROP THE REFLECTED WAVE AT THE PUPIL PLANE
z_aperture = (z_propagated.*repmat(circ,...
    [1 1 num_frequencies speckle_size]));
z_propagated = [];

%PROPAGATE TO THE IMAGE PLANE
z_aperture2 = ft2(z_aperture);
z_aperture = [];
z_aperture3 = (z_aperture2((N/2-(N_image/2-1)):(N/2+N_image/2),(N/2-(N_image/2-1)):(N/2+N_image/2),:,:));
z_aperture2 = [];
z_aperture4(:,;1+(speckleloop-1)*speckle_size:...
    speckleloop*speckle_size) = z_aperture3;
plane_wave2(:,;1+(speckleloop-1)*speckle_size:...
    speckleloop*speckle_size) = 1e7.*exp(1i.*2.*pi./...
    (repmat(lambda2,[N_image N_image,1,speckle_size]).*...
    (repmat(X,[1 1 num_frequencies speckle_size]).*...
    dxyS.*ref_angle)+...
    (repmat(Y,[1 1 num_frequencies speckle_size]).*...
    dxyS.*ref_angle)+(sqrt(1-2.*ref_angle.^2).*zi)));
end

%COMBINE THE WAVEFRONT WITH THE LO TO CREATE A HOLOGRAM
hologram2= abs(z_aperture4 + plane_wave2).^2;
mu = mean(mean(abs(z_aperture4.*plane_wave2).^2));
z_aperture4 = [];
for average = 1:1:averaging_num
    re_image = zeros(image_size,image_size,range_padding,...
        snrlength);
    for avg_index = 1:1:average
        hologram = repmat(hologram2(:,;avg_index),[1,1,1,...
            snrlength]);
        plane_wave = plane_wave2(:,;avg_index);
        snr_index = 1;
        SNR_0 = [];
        SNR_0(1,1,:) = totalsnr(1,snrstart:snrend)/...
            num_frequencies/average;
sigma_noise(1,1,:) = repmat(mu(:,;avg_index),...
    [1 1 snrlength]).*repmat(numel(crop)/(SNR_0.*...
        sum(crop(:))),[1 1 num_frequencies 1]);
        noise = repmat(sqrt(sigma_noise),...
[N_image N_image 1 1]).*abs(randn(size(hologram)));

%%ADDING SHOT NOISE TO THE HOLOGRAM
hologram = hologram + noise;
noise = [];sigma_noise = [];

%%NEXT 8 LINES ARE THE CROSS-RANGE RECONSTRUCTION OF A
%%HOLOGRAM
intensity = ft2(hologram);
hologram = [];
intensity_crop = ift2(intensity.*repmat(crop,...
    [1,1,num_frequencies,snrlength]));
intensity = [];
z_image = (ft2((intensity_crop).*repmat(...
    ((plane_wave),[1,1,1,snrlength]))));
intensity_crop = [];plane_wave = [];
pixel_inc = 256;
z_image = ft2(z_image(N_image/2-pixel_inc+1:N_image/...2+pixel_inc,N_image/2-pixel_inc+1:N_image/2+... pixel_inc,:,:));

%%APPLYING EIGEN CORRECTION
%
for i = 1:snrlength
    z_image(:,:,i,:) = eigncorrect(z_image(:,:,i,:));
end

%%APPLYING SUB PIXEL CORRECTION
z_range_comp = abs(fftshift(fft(z_image,...
    range_padding,3),3)).^2;
rangelPR = sum(sum(z_range_comp));
rangelPR = fftshift(ifft(ifftshift(rangelPR,3),...[1,3],3));
rangelPR = abs(fftshift(fft(rangelPR,range_padding*...256,3),3));
 [~,b] = max(rangelPR,[]);3);
err=(b-(range_padding*256/2+1))*2*pi/range_padding/256;
for i=1:num_frequencies
    z_image(:,:,i,:) = z_image(:,:,i,:).*repmat(exp...
    (-1j*err*i), [ N_image/2 N_image/2 1 1]);
end

%%ZERO PADDING AND FOURIER TRANSFORMING OVER THE RANGE
%DIMENSION
    z_range_comp = abs(fftshift(fft(z_image,range_padding,3)... 
            ,3)).^2;
    z_image = [];

%ADDING THE INTENSITIES TOGETHER TO APPLY SPECKLE
%AVERAGING
    re_image = z_range_comp + re_image;
end

%DIVIDING BY THE AMOUNT OF SPECKLE AVERAGES TO FINISH
%SPECKLE AVERAGING
    re_image2 = re_image./average;

%PICKING OFF THE MAXIMUM LOCATION IN THE RANGE DIMENSION
%FOR EVERY CROSS-RANGE PIXEL
    [~,index] = max(re_image2,[],3);
    z_range_comp = [];re_image2 = [];

%CHANGING LOCATION TO RANGE
    index = (index-1-range_padding/2).*dz;
    storage_cube(:,:,int,:) = index;
    index = [];
    int = int + 1
end
    re_image = [];

objectpcd(zeros(1024),1024,dxyR)
    save(['image_cube_hologram ',num2str(num_frequencies),...
            'Frequencies SNR of ' num2str(snrstep),' averaging ',...
            num2str(average),' results ',Title,num2str(int99),...
            '.mat'],'storage_cube','-v7.3');
    results = zeros(averaging_num,snrlength);
    for avg = 1:1:averaging_num
        for snnr = 1:1:snrstep
            imagepcdone(storage_cube(:,:,avg,snnr),...
                        image_size-2,2.*dxyR)
        end
end

%CALCULATING THE POINT CLOUD METRIC
    system('getFitnessScore hologram for');
    fit = fopen('results hologram.txt');
    result=textscan(fit,'%f64');
    results(avg,snnr) = cell2mat(result);
fclose('all');
end
disp(avg)
end

%The 5.2e-7 is the experimental divation from the flat plate assumption
getfitstore(:,:,int2,int99) = results+5.2e-7;
average = 1:1:averaging_num;
index = 1;

%CALCULATING THE CRLB
for SNR_0 = totalsnrstart:1:totalsnrend
    BRMS = pi./sqrt(3).*frequency_step.*num_frequencies;
    variance = c.^2./(8*(SNR_0.*(average)).*BRMS.^2);
    variancestore(:,:,index,int2,int99) = variance;
    index = index + 1;
end
int99 = int99+1;
int2 = int2 + 1;
end
persnr = getfitstore;
save(['getFitnessScore results ', Title,' With ',... num2str(averaging_num),', averages ',num2str(num_frequencies)... ,' Frequencies',',persnr');
pervar = variancestore;
average = 1:averaging_num;
plotsnr = totalsnrstart:1:totalsnrend;
upperbound = zeros(length(plotsnr),1);
for i = plotsnr
    upperbound(i) = (dz.*range_padding)^2/12;
end

%PLOTTING THE RESULTS FOR THE POINT CLOUD METRIC OF A FLAT PLATE TARGET
for in = 1:2:(int2-1)
    for i = 1:1:averaging_num
        h = figure;
        loglog(plotsnr(1,round(totalsnr(1,1:snrend))),squeeze...
            (persnr(i,:,1)),plotsnr,squeeze(pervar(i,:,1)),plotsnr,...
            upperbound);
        set(gca,'FontSize',13)
xlhand = get(gca,'xlabel');
set(xlhand,'string','SNR times number of speckle realizations','fontsize',13)
ylhand = get(gca,'ylabel');
set(ylhand,'string','Variance','fontsize',13)
tlhand = get(gca,'title');
set(tlhand,'string',[num2str(2.^((in+1))))
    ' Frequencies, Speckle averaged ', num2str(i), ', ',...
    ' Uniform Distribution');
legend([num2str(2.^((in+1))), 'F gFS',], 'CRLB',...
    'Uniform Distribution');
ylim([10^-7 0])
saveas(h,['gFS for ',num2str(2.^((in+1))), ' freq ',num2str(i)...,
    ' averages ',Title], 'fig')
saveas(h,['gFS for ',num2str(2.^((in+1))), ' freq ',num2str(i),...
    ' averages ',Title], 'jpg')
end
end
BRMS = []; SNR = []; SNR_0 = []; crop = [];