

APPLICATION OF COMPUTED TOMOGRAPHY FOR  
MEASURING THREE DIMENSIONAL REFRACTIVE INDEX  
INHOMOGENEITY

by  
Brian Lee Stamper

---

Copyright © Brian Lee Stamper 2004

A Dissertation Submitted to the Faculty of the  
COMMITTEE ON OPTICAL SCIENCES (GRADUATE)

In Partial Fulfillment of the Requirements  
For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2004

UMI Number: 3145133

Copyright 2004 by  
Stamper, Brian Lee

All rights reserved.

### INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

**UMI**<sup>®</sup>

---

UMI Microform 3145133

Copyright 2004 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against  
unauthorized copying under Title 17, United States Code.

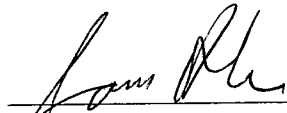
ProQuest Information and Learning Company  
300 North Zeeb Road  
P.O. Box 1346  
Ann Arbor, MI 48106-1346

The University of Arizona ®  
Graduate College

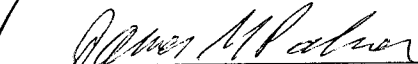
As members of the Final Examination Committee, we certify that we have read the  
dissertation prepared by Brian L. Stamper

entitled Application of Computed Tomography for Measuring Three  
Dimensional Refractive Index Inhomogeneity

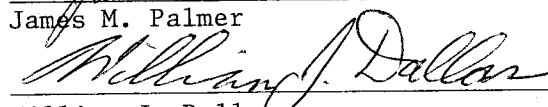
and recommend that it be accepted as fulfilling the dissertation requirement for the  
Degree of Doctor of Philosophy

  
\_\_\_\_\_  
James H. Burge

July 7, 2004  
\_\_\_\_\_  
date

  
\_\_\_\_\_  
James M. Palmer

7 July 2004  
\_\_\_\_\_  
date

  
\_\_\_\_\_  
William J. Dallas

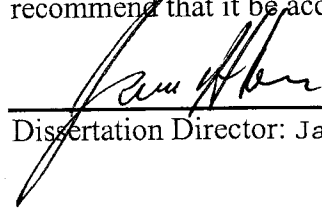
7 July 2004  
\_\_\_\_\_  
date

\_\_\_\_\_  
date

\_\_\_\_\_  
date

Final approval and acceptance of this dissertation is contingent upon the  
candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and  
recommend that it be accepted as fulfilling the dissertation requirement.

  
\_\_\_\_\_  
Dissertation Director: James H. Burge

7/13/04  
\_\_\_\_\_  
date

## STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

SIGNED: Brian Lee Stamps

## ACKNOWLEDGMENTS

I was able to focus on research the last two years of my studies thanks to the Graduate Student Researchers Program. Funded by NASA, this award was managed by JPL to encourage graduate studies in support of NASA objectives.

I worked on many projects before deciding that the topic presented here gave me the best opportunity to lead interesting research. Many people were a huge help on my earlier projects, not the least of which are Roger Angel, Tom Connors, Brian Duffy, and Marty Valente. Much of the success of those projects was due to their help, insight, and after hours help conducting experiments.

As my advisor, Jim Burge has helped me roll with the punches so well in the course of this degree that I can't imagine working for any other professor. He managed to find the perfect balance between providing help when needed, yet not looking over my shoulder at every turn. The responsibilities that I've been allowed to take on have boosted confidence in myself and my abilities to lead people and conduct independent research.

I would like to thank Bill Dallas for taking time to help me work through problems. Time he often didn't have. His insight, programming skills, and commiseration in large part kept me going until finishing this dissertation.

Many thanks to my parents for their unwavering support over my entire educational career. Without their continuing support and confidence in my abilities to reach this goal, I would never have succeeded. This degree is as much theirs as it is mine.

## TABLE OF CONTENTS

LIST OF FIGURES . . . . .	<b>8</b>
LIST OF TABLES . . . . .	<b>11</b>
ABSTRACT . . . . .	<b>12</b>
<b>CHAPTER 1. INTRODUCTION . . . . .</b>	<b>13</b>
1.1. Manufacturing Errors in Optical Glass . . . . .	13
1.2. Power Spectral Density Applied to Index Variations . . . . .	14
1.2.1. Analysis . . . . .	14
1.2.2. Image Degradation . . . . .	15
1.3. Applications . . . . .	16
1.3.1. Telescope Correctors . . . . .	16
1.3.2. Lithography . . . . .	17
1.3.3. Prism Spectrometers . . . . .	17
1.4. Measurement limitations . . . . .	17
1.5. Improved method . . . . .	18
<b>CHAPTER 2. COMPUTED TOMOGRAPHY . . . . .</b>	<b>20</b>
2.1. Basic Reconstruction Theory . . . . .	20
2.2. Advanced Methods . . . . .	21
2.2.1. Filtered Backprojection . . . . .	22
2.2.2. Summation Image . . . . .	22
2.2.3. Direct Inversion . . . . .	24
2.2.4. Iterative Methods . . . . .	25
<b>CHAPTER 3. APPLIED CT FOR MEASURING BULK INDEX VARIATION . . . . .</b>	<b>27</b>
3.1. Fourier Synthesis Simulations . . . . .	27
3.1.1. Two Dimensional CT . . . . .	27
3.1.2. Filling the reconstruction array . . . . .	30
3.1.3. Filtered reconstruction . . . . .	32
3.1.4. Linear Interpolation . . . . .	32
3.1.5. Gridding . . . . .	32
3.1.6. Limited Angle . . . . .	33
3.1.7. Region of Interest . . . . .	34
3.1.8. Object Contrast . . . . .	36
3.1.9. Effect of sharp change in values . . . . .	37
3.2. Circular Harmonic Decomposition . . . . .	37

TABLE OF CONTENTS—*Continued*

3.2.1.	Filling Zero Projections . . . . .	38
3.2.2.	Projection Smoothing . . . . .	39
3.2.3.	Windowing . . . . .	40
3.3.	Measurement Quality . . . . .	42
3.3.1.	Root Mean Square . . . . .	42
3.3.2.	Correlation . . . . .	43
3.3.3.	Number of Projections . . . . .	47
3.3.4.	Signal to Noise Ratio . . . . .	49
CHAPTER 4.	EXPERIMENTAL . . . . .	<b>51</b>
4.1.	Sample Geometry . . . . .	52
4.2.	Background compensation . . . . .	53
4.3.	Multiple axis information . . . . .	53
4.4.	Demonstration . . . . .	54
4.5.	Result Comparison . . . . .	56
4.5.1.	Visual . . . . .	56
4.5.2.	Numerically . . . . .	56
4.5.3.	Initial Reconstructions . . . . .	57
4.5.4.	Final Reconstructions . . . . .	57
4.6.	Error Analysis . . . . .	59
4.6.1.	Ray Bending . . . . .	59
4.6.2.	Accuracy . . . . .	60
CHAPTER 5.	CONCLUSIONS . . . . .	<b>61</b>
5.1.	Future Work . . . . .	62
5.1.1.	Statistical model . . . . .	62
5.1.2.	Multi-axis measurements . . . . .	62
5.1.3.	Lens design modelling . . . . .	62
5.2.	Future Limited Angle Solutions . . . . .	63
5.2.1.	Prisms . . . . .	64
5.2.2.	Water Bath . . . . .	65
5.2.3.	Cylindrical Cavity . . . . .	65
5.2.4.	Multi-sided Detection . . . . .	65
5.2.5.	Augmented Projections . . . . .	65
APPENDIX A.	PROGRAMS . . . . .	<b>67</b>
A.1.	main.pro . . . . .	67
A.2.	object.pro . . . . .	72
A.3.	stdcompare.pro . . . . .	74
A.4.	mask.pro . . . . .	76

TABLE OF CONTENTS—*Continued*

A.5. display.pro . . . . .	77
A.6. createindex.pro . . . . .	77
A.7. read.pro . . . . .	80
A.8. reproject.pro . . . . .	81
APPENDIX B. HISTORY OF X-RAY COMPUTED TOMOGRAPHY . . . . .	<b>83</b>
APPENDIX C. HARDWARE COMMANDS . . . . .	<b>85</b>
C.1. Sample Commands . . . . .	85
REFERENCES . . . . .	<b>87</b>



## LIST OF FIGURES

FIGURE 2.1. Three different objects can appear the same if only two projections are used. Additional information is needed (i.e. more projections) to uniquely reconstruct the object. Notice that the diagonal projections are different. . . . .	21
FIGURE 2.2. Single projection being backprojected. . . . .	23
FIGURE 2.3. Phantom object shown with three projections. . . . .	24
FIGURE 2.4. Image formed by summing the backprojections to form a reconstruction. . . . .	24
FIGURE 3.1. Geometry for making a single measurement, or projection. The integration along the projection axis is perpendicular to the rotation axis.	28
FIGURE 3.2. Rotation of sample between projections as needed for computed tomography. After each projection, the sample is rotated by a preset angle.	29
FIGURE 3.3. At left is a projection array made from individual measurements. At right is the resulting Fourier array after transforming each row of the projection array, and placing it into a blank array based on their corresponding angle. All blank locations will be filled in with the nearest non-zero value. . . . .	30
FIGURE 3.4. The Fourier array resulting when a nearest neighbor approach is used to fill each location. . . . .	31
FIGURE 3.5. A sample with a limited range of angles available over which a projection can be made. The cones represent angles over which we can make measurements. . . . .	34
FIGURE 3.6. Degradation when limiting the measurement angle. From left to right we have a total view angle of $180^\circ$ , $120^\circ$ , $60^\circ$ , $20^\circ$ . . . . .	34
FIGURE 3.7. Effect of limiting the measurement aperture. The left is a full reconstruction while the right image has a region of interest imposed on the measurement. . . . .	35
FIGURE 3.8. Increase in angular projections available when a smaller test beam is used. A large beam on the left allows fewer rotation angles where the smaller beam on the right permits a larger maximum angle. . . . .	35
FIGURE 3.9. Our reconstruction for an object contrast equal to one is on the left. In the center we see the effect of decreasing the object contrast to $\frac{1}{2}$ , and on the right of decreasing to $\frac{1}{5}$ . No noise has been added to the object.	37
FIGURE 3.10. The left plot shows blank projections that are replaced by approximations at right. . . . .	40
FIGURE 3.11. The left plot shows the reconstruction without using a windowing function. The plot at right is multiplied by a Hanning window. . . . .	41

LIST OF FIGURES—*Continued*

FIGURE 3.12. RMS is given comparing single and dual axis placement of the projections. The dual axis method increases the accuracy even when the same number of projections as the single axis are used. . . . .	43
FIGURE 3.13. The original object is shown at left. As we go right, we have images with limited angles of $60^\circ$ , $40^\circ$ , and $20^\circ$ . . . . .	43
FIGURE 3.14. At left, the original object is shown as created with a Gaussian filter width of 40 pixels for a 256 pixel array. The center image is the reconstruction with a correlation coefficient of 0.9814, and the right image is the difference. Only a very small component of the reconstruction is error. . . . .	46
FIGURE 3.15. At left, the original object is shown as created with a Gaussian filter width of 40 pixels for a 256 pixel array. The center image is the reconstruction with a correlation coefficient of 0.8566, and the right image is the difference. A 120 degree limited angle constraint has been imposed. . . . .	46
FIGURE 3.16. Correlation change with spatial scale are shown. The object is 256 pixels wide. We can lower the spatial content by filtering with a Gaussian. The width given is the width of the Fourier space filter. . . . .	47
FIGURE 3.17. Correlation as a function of the total number of projections used in the reconstruction. . . . .	48
FIGURE 3.18. Signal to noise change with spatial scale is shown. . . . .	49
FIGURE 3.19. Signal to noise change with spatial scale is shown. A smaller amount of data is shown to emphasize crossing points for $SNR = 1$ . . . . .	50
FIGURE 4.1. Double pass configuration used for transmission testing of glass samples. . . . .	52
FIGURE 4.2. Some problems appear for large angles of incidence. At large angles, total internal reflections occurs for the transmitted rays, while rays missing the front surface are scattered out of the test system. . . . .	53
FIGURE 4.3. A cube glass sample is shown at left, and a thin glass plate is shown at right. . . . .	54
FIGURE 4.4. Interferogram at left has been rotated $15^\circ$ clockwise from where the face is perpendicular to the interferometer. At right, the interferogram has been rotated $15^\circ$ counter-clockwise. . . . .	55
FIGURE 4.5. Interferogram at left has been rotated $15^\circ$ clockwise from where the face is perpendicular to the interferometer. At right, the interferogram has been rotated $14^\circ$ counter-clockwise. . . . .	55
FIGURE 4.6. A slice perpendicular to the rotation axis at its midpoint from the 3D reconstruction for the cube glass sample is on the left. The slice shown at right is also perpendicular to the rotation axis a quarter of the way from the top. . . . .	57

LIST OF FIGURES—*Continued*

- FIGURE 4.7. From left to right we show projections at a rotation angle of one degree, five degrees, and  $10^\circ$  made from a three dimensional reconstruction. The top row shows surface maps created from interferograms made through the sample. On bottom is the corresponding projection for the same angle of rotation. . . . . 58
- FIGURE 4.8. A projection from our data cube is shown, and three slices are shown. Each slice is a projection through one third of the data cube. The images correspond to a projection angle of one degree. . . . . 59
- FIGURE 5.1. Extreme portions of the glass are covered by projecting through a prism. Grazing incidence is shown with an entrance angle of  $\theta_1$ , and the decreased angle of  $\theta_2$ . . . . . 64
- FIGURE 5.2. Sinogram built from two detector information. . . . . 66

## LIST OF TABLES

TABLE 1.1.	Comparison of select medical and refractive variables. . . . .	19
TABLE 3.1.	Calculation time versus the number of projections. . . . .	48
TABLE B.1.	Chronological overview of x-ray computed tomography. . . . .	84

## ABSTRACT

Manufacturers of optical glass strive to make a product that is homogeneous, isotropic, and free of any bubbles or mechanical strain. Glass used in forming images is very good, but the process of mixing the constituent materials, and melting them into a glass is limited. The index of refraction varies based on the lack of uniformity remaining after the manufacturing process. Transmitted wavefronts will have errors due to this inhomogeneity.

The most common method currently used to quantify the homogeneity of a glass sample is to measure in one direction through the glass. Variations along the test axis are integrated resulting in loss of positional information in this direction. Homogeneity is then reported by using the peak-to-valley wavefront error reducing the three dimensional nature of glass to a single value. Not only have we lost the longitudinal information, but we have also lost any knowledge of the transverse texture of the sample.

We present in this research a method for retrieving three dimensional information about the inhomogeneity of a glass test piece. Computed tomography provides a well developed methodology for constructing a three dimensional measurement from two dimensional data. Common interferometric measurements, or projections, taken at multiple angles has sufficient information to estimate the full three dimensional structure of the test piece.

Important differences from computed tomography used for medical diagnoses are explored. Refraction at the interfaces of the sample limits the number of angles over which projections can be made. The angular distance between projections also influences the accuracy of the reconstructed object.

## Chapter 1

# INTRODUCTION

Homogeneity is the most important requirement for glass intended to form images[1]. Refractive index inhomogeneities are inherent errors remaining from the manufacturing process. Even though a great deal of effort is exerted in making very homogeneous, isotropic optical glasses, this perfection is never quite achieved[2, 3]. Mixing mechanics is the dominate process for this phenomenon as several materials come together simultaneously. Component ingredients of a glass will be in particulate form initially before melting into an amorphous conglomerate. The grains of each material will vary in size to some degree, and may have a different average particle size than the other ingredients. The result is a variation of index with position in the final product. A purely homogeneous piece of glass will allow light to propagate without any change in direction as it traverses the glass. Small changes in index will bend the light during its journey through the glass, culminating in degradation of the final image.

### 1.1 Manufacturing Errors in Optical Glass

In addition to random phase errors introduced by inhomogeneous glass, other manufacturing errors can be more dramatic. Striae and bubbles cause sharp changes in the phase as a wavefront encounters them in a glass element where the inhomogeneity has a much smoother variation in index. Striae is where a part of the glass has a local difference in index compared to the surrounding glass [4]. A small string of internal glass may have a different index, or it may be a layer of the glass that has a different index. Methods for measuring and quantifying the errors will be presented.

## 1.2 Power Spectral Density Applied to Index Variations

The Power Spectral Density (PSD) can provide information on the spatial fluctuations in the measured quantity. Taking the PSD of a phase map can give the size of the variations that are the source for errors present in the glass. This has been applied to the measure of the surface roughness and figure for optical glass and other materials [5, 6, 7]. The mid-spatial frequencies, in which glass usually falls for surface quality, is very well suited to PSD measurements [8]. This frequency range falls between the regions normally measured by figure errors and surface roughness [9].

### 1.2.1 Analysis

We want to characterize an entire random process by its spectral distribution. Inhomogeneity is treated as a random variation of refractive index with position. We will outline the steps to obtain the PSD in terms of averages over the entire random process [10]. We begin by showing one definition of the Fourier Transform:

$$U(\nu) = \int_{-\infty}^{\infty} u(x)e^{i2\pi\nu x} dx, \quad (1.1)$$

where  $x$  is our distance variable. Spatial frequency is represented by  $\nu$ , and we see that the area under  $|U(\nu)|^2$  is equal to the total energy in  $u(x)$ . This is expressed in the form:

$$\int_{-\infty}^{\infty} |u(x)|^2 dx = \int_{-\infty}^{\infty} |U(\nu)|^2 d\nu \quad (1.2)$$

so that the energy can be written:

$$\mathcal{E}(\nu) = |U(\nu)|^2 \quad (1.3)$$

and has the units of energy per unit frequency with  $\mathcal{E}(\nu)^2$  being the energy spectral density (ESD). If the function of interest is not Fourier transformable, but has finite

average power, then we can truncate the function to a finite period in space. If the ESD is then normalized in space, the normalized energy spectrum becomes:

$$G_{\mathbf{x}}(\nu) = \frac{|U(\nu)|^2}{\mathbf{X}} \quad (1.4)$$

and has the units of power per unit frequency. The power spectral density is then realized as the normalized energy spectrum tends to the limit of an infinite distance period,

$$G_{\mathbf{x}}(\nu) = \lim_{\mathbf{x} \rightarrow \infty} \frac{|U(\nu)|^2}{\mathbf{X}}. \quad (1.5)$$

In practice,

$$G(\nu) = \Re\{U(\nu)\}^2 + \Im\{U(\nu)\}^2 \quad (1.6)$$

where this is just the modulus squared of the complex fourier transform. The PSD is then straightforward to calculate using any computer method.

Once a reconstruction of the test sample is calculated, we can take a closer look at subsections within the full volume. Smaller volumes within the larger data set can be compared to find the best portion of a sample. Using the PSD on smaller sections of the glass would differentiate the best suited section of the glass sample to be used for the application.

### 1.2.2 Image Degradation

Once many inhomogeneity measurements have been taken of a single glass (e.g. BK7, ULE, etc.), a PSD model can be made that represents a typical sample from that glass. Such a model can then be used to determine the necessary tolerances on an element of the modelled glass being used in an imaging system. The advantage is that the inhomogeneity is now built into our tolerances during the design process, instead of being a surprise after the system is built.



A specific band of frequencies may be much worse for an imaging system than other spatial frequencies, and efforts to improve quality could be focused in that area. Correlating spatial frequencies of error in the glass to the effect at different wavelengths will provide additional information as well.

## 1.3 Applications

Inhomogeneity in glass can cause a significant error in any application where light must traverse several centimeters of glass. Even though the surfaces may have figures better than  $\frac{\lambda}{20}$ , the internal inhomogeneity can still cause several waves of error if the glass contributes an erratic phase distribution to the wavefront. Some specific examples are discussed where this effect is known to be a problem.

### 1.3.1 Telescope Correctors

A telescope will often be refitted for duties that it wasn't designed for during initial construction. Often, this is a very cost effective way to do new science, and is a great way to take advantage of an existing installation. An example of this may be to convert a Cassegrain system for duty at prime focus. Larger fields can be viewed this way for sky survey work. Aberrations corrected by the secondary now have to be handled by a lens assembly which for a *90 inch* telescope would require *60 cm* diameter lenses each of which may be several inches thick.

While the thickness can introduce variations along the transmitted direction, the lateral dimension is also a problem. A transmissive optic as large as *60 cm* may have very different index profiles at different locations in the field of view. Thus, different parts of the object may be presented with refractive index that varies with position. Resulting wavefronts can contain up to several waves of error across the field of view.

### 1.3.2 Lithography

Lithography systems today work at very short visible wavelengths [11] pushing below 100 *nm*. Few glasses transmit well in this part of the spectrum leading to lens systems with only one or two glasses being used. Over a finite bandwidth, this requires much more glass to correct for dispersion and still achieve the desired optical performance. The amount of glass used in lithography systems can be large, often being measured in feet of glass that the light must propagate through.

### 1.3.3 Prism Spectrometers

Prism based spectrometers use a large glass prism as the dispersing element in the system. Once again, the large thickness of glass can cause unexpected errors due to inhomogeneity alone. Accurately testing the glass to be used could avoid wasting a very costly and possibly lengthy polishing process.

## 1.4 Measurement limitations

Currently, interferometry is used to analyze the refractive index variations in which measurements are integrated along the test beam. Any changes along the optical axis are averaged along the length of the glass piece. It is common to use index matching fluid and polished windows of known surface figure to reduce the error during refraction of the test beam into the sample. Inhomogeneity is then specified using a single number - the peak to valley change found within the glass sample. Not only have we lost the longitudinal information, data about the texture of the two transverse axes is also gone. An inclusion, or defect, may have a small volume, but will be shown to extend the length of the sample along the test axis. The magnitude of the index change caused by the inclusion has been averaged with the other values along the test axis.

A series of interferograms is taken along this direction such that the surface errors of the sample can be subtracted from the measurement. Windows of known surface quality are used with index matching fluid so that the only unknown contributor to the wavefront variation is the change of refractive index as a function of position within the sample. Homogeneity is then reported by using the peak-to-valley wavefront error reducing the three dimensional nature of glass to a single value. We want to measure the homogeneity, but retain information along the test axis, building up a three dimensional profile of the sample.

Having three dimensional information on the refractive index is the first step in compensating for imaging errors contributed by the inhomogeneity of the glass. A glass block could be tested before being made into a lens to ensure that no section has a large index variation. The measured inhomogeneity can be used to determine a boundary for the best quality image that can be made. The power spectral density (PSD) will provide the spatial scale at which those errors will be introduced into the imaging system. If a large sample population is available, a statistical model of these errors could be mapped for a particular type of glass.

## 1.5 Improved method

Computed tomography is a technique that uses multiple measurements to provide three dimensional information on the test sample, solving the problem of integration along the test axis. The same interferometric data, that currently integrates one dimension, will be used to gather the necessary projections for a glass sample. However, many more measurements are to be taken as the glass is rotated. The set of optical path data will be used to build a three dimensional array that corresponds to the refractive index at any point in the bulk sample of glass.

This is significantly different than computed tomography as applied in the medical industry and comparison is given in Table 1.1. Important differences arise when the

TABLE 1.1. Comparison of select medical and refractive variables.

	<b>Medical</b>	<b>Refractive Index</b>
<b>Limited Angle</b>	Imaged Through $180^\circ$	$60^\circ$ or less
<b>Angular Step</b>	$\sim 0.33^\circ$	$1.00^\circ$ or more
<b>MTF-Filtering</b>	Associated PSF	Flat Response Required
<b>Ray Bending</b>	Negligible	Refraction at Interface

methodology is applied using visible wavelengths with a refracting object. Medical scanners take evenly spaced measurements while rotating about the patient. Transmissivity for x-ray wavelengths is then measured. For a refracting sample such as glass, only a few angles can be measured before the geometry limits the angular extent of the measurement. The result is large blank spaces in the measurements needed to reconstruct the object.

## Chapter 2

# COMPUTED TOMOGRAPHY

Tomographic techniques provide a methodology to rebuild three dimensional information using two dimensional projections. This information is lost when taking a single measurement, but can be restored such that the magnitude and location of index changes within the sample can be determined. Here we outline the idea behind computed tomography and give several methodologies. Detailed procedures are given on the specific approaches used and the motivation behind them.

### 2.1 Basic Reconstruction Theory

Tomography techniques involve reconstructing a picture of the object by measuring the transmission of the object. Data are taken with the sample rotated through many different angles. The rotation angle is recorded for each measurement, and will identify the angle at which that measurement was taken. The most common method today is known as Computed Tomography (CT). More accurately, it is called computed trans-axial tomography. In medical imaging where people are the objects of importance, the source is commonly x-rays or gamma rays. Similar methods are used for emission tomography [12]. The reconstructed plane is parallel to the beam of rays, and three dimensional images are made one slice at a time [13].

The first implementation of tomography was a longitudinal reconstruction. This analog method is most commonly referred to as classical tomography [14], and is still in use today. Related approaches such as tomosynthesis [15, 16], and Ectomography [17] will not be discussed here.

In its simplest form, line integrals are taken for all straight lines through the region of measurement [18, 19, 20, 21]. At first glance, one would think that just taking the

perpendicular projections would be sufficient, but that is not the case. Figure 2.1 shows the ambiguity when just the orthogonal projections are used [12]. Medical applications of tomography measure the density of an object based on absorption information while we want to measure small changes in the optical path [22].

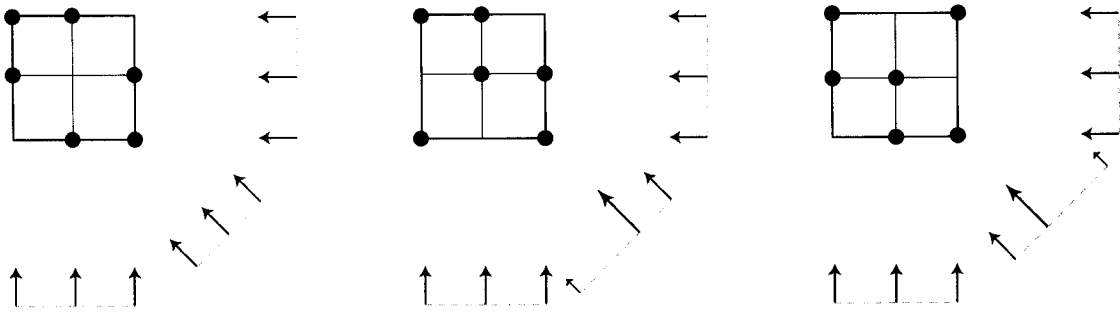


FIGURE 2.1. Three different objects can appear the same if only two projections are used. Additional information is needed (i.e. more projections) to uniquely reconstruct the object. Notice that the diagonal projections are different.

In practice each line integral is over a finite area. Each line becomes a strip that has a finite width. Instead of having a line of infinitesimally small points, we have a strip made up of pixels with a length and width larger than zero [23]. Two problems are now intertwined. The infinite number of intensity points along the line are lumped together in discreet buckets, and there is a local blurring caused by the width of the strip. Compensation for the local blurring has been referred to as restoration while building an image from projections is called reconstruction. For purposes presented here, restoration will be ignored as we focus on techniques for reconstruction.

## 2.2 Advanced Methods

In the early 1970's Hounsfield [21] demonstrated a practical application of computed tomography techniques. This spurred work on specific algorithms, many of which dealt with a particular geometry for making the projections. Differences between the

early algorithms is subtle as even small gains in computing time were very important.

The projection array, or sinogram, is a very good diagnostic tool. It visually can clarify if projections are oriented and organized correctly. Fourier transforms, filtering, and other operations can be done at this step in the algorithm. Importance is then placed on the accuracy and validity of the projection array.

### 2.2.1 Filtered Backprojection

We start by defining the process of backprojection. It is just the reverse of the projection operation [24]. The one dimensional projection is smeared across the reconstruction array such that the projection is copied along one dimension within the two dimensional array as illustrated in Figure 2.2. If a small number of projections are available, it is advantageous to smear a single projection across several angles. Trial and error is necessary to determine the most effective angular spread. The recovered original object is less blurry if each one dimensional transform is filtered before the backprojection [25]. The simplest inverse filter will be discussed further in the next section.

### 2.2.2 Summation Image

A summation image is formed if we have multiple projections that have been backprojected onto the reconstruction array. Figure 2.3 demonstrates three projections taken for a simple object. To backproject we uniformly smear the projection across the reconstruction plane corresponding to the original angle used to make the projection. Figure 2.4 shows how the backprojections overlap more completely at the center of the array. When the values are added at each location to form a new reconstruction, a summation image is formed. Often these operations are lumped together in the term backprojecting.

Concentration at the center causes a spike to form that is inversely proportional

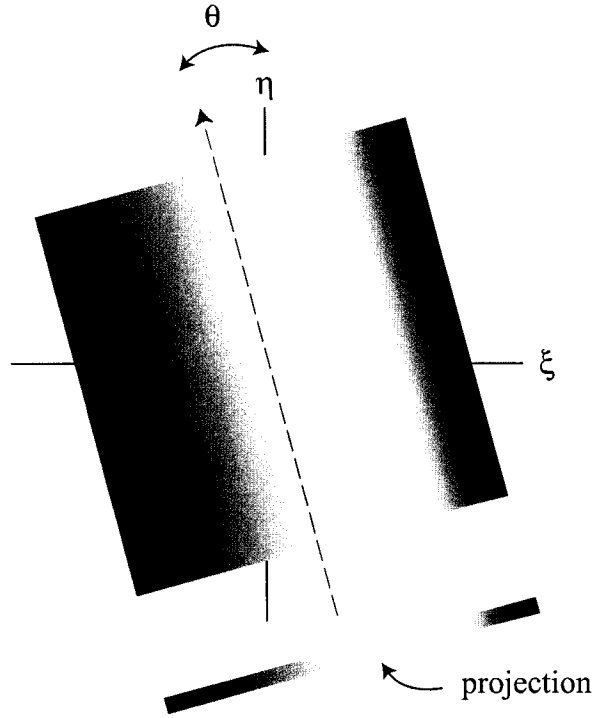


FIGURE 2.2. Single projection being backprojected.

in magnitude to the distance from the center of the array. Filtering the summation compensates for the large central spike. The most common filter is to divide by  $\frac{1}{\rho}$ , or simply multiply by  $\rho$  in frequency space. For this method of *filtered backprojection*, where  $\rho$  is the distance from the center to any point in our reconstruction array. We can think of the summation image as having a point spread function (psf) of  $\frac{1}{r}$  with a Fourier transform equal to  $\frac{1}{\rho}$ . To properly compensate in the frequency domain we multiply by  $\rho$  as mention above. Our frequency response is then flat so that we have uniform reconstruction with respect to spatial frequency.

Filtered backprojection, is often referred to as rho-filtered layergram reconstruction [26, 27]. Note that the filtering step can be done before or after the backprojection. We choose to follow Barrett here in keeping the concept of summation and filtering of the backprojections separate [12].



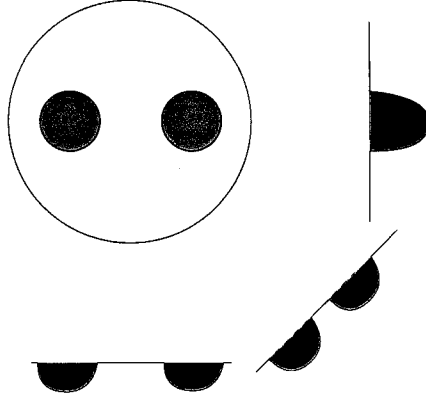


FIGURE 2.3. Phantom object shown with three projections.

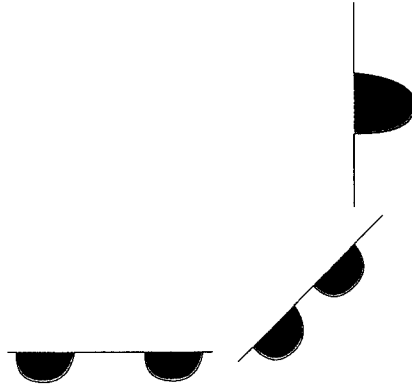


FIGURE 2.4. Image formed by summing the backprojections to form a reconstruction.

### 2.2.3 Direct Inversion

Direct Inversion simply inverts the projection matrix directly. Starting with a projection matrix,  $p_i$ , where

$$p_i = \sum_j w_{ij} f_j \quad (2.1)$$

and we want to solve for the solutions  $f_j$ . Rows are indexed by  $i$ , and the columns of the projection matrix are indexed using  $j$ . Inverting the matrix  $w_{ij}$  as shown below

$$f_j = \sum_{i=1}^J (w^{-1})_{ij} p_i \quad (2.2)$$

will provide a solution. Noise or errors in the data may keep this method from having a solution. It is also likely that  $w_{ij}$  may have a prohibitively large number of elements. Often,  $ij$  in which case the inverse of the matrix does not exist. A psuedo-inverse, or singular value decomposition, however, may exist [28]. Iterative methods described next provide a solution without actually inverting the matrix.

#### 2.2.4 Iterative Methods

Iterative methods are an alternative to the more analytic approach previously mentioned. Some metric is used as a measure of how close the current reconstruction iteration is to the original object. A weighted least-squares method is shown here [29]:

$$\chi^2(\mathbf{X}) = \frac{\sum_{km} (\sum_{ij} F_{ij}^{km} X_{ij} - p_{km})^2}{\sigma_{km}^2} \quad (2.3)$$

$$= (\mathbf{X} \cdot \mathbf{M}\mathbf{X}) - 2(\boldsymbol{\nu} \cdot \mathbf{X}) \quad (2.4)$$

where  $\mathbf{X}$  is the intensity matrix that we are trying to reconstruct. The parameter  $p_{km}$  is the projection taken at the rotation angle  $m = \theta$ , and array location  $k$ ;  $\sigma_{km}$  is the uncertainty in the measurement of  $p_{km}$ ;  $X_{ij}$  is the intensity at pixel location  $i, j$  to be reconstructed; and  $F_{ij}^{km}$  represents the model used to describe the intensity distribution across each pixel and also contains any attenuation compensation. Equation 2.4 is shown rewritten in matrix form.

Once the problem has been properly defined, the task is narrowed to finding an efficient method of calculation [30]. Often, a few iterations are run to improve noise, but rarely to convergence. The nature of the problem determines the best approach, but these issues will not be discussed here.

Algebraic Reconstruction Techniques (ART) treat each projection as an equation in a set of simultaneous equations [31, 32]. The unknowns are represented by each of the pixels in the projection. The number of equations is almost always different than the number of unknowns, but iterative techniques have been developed to overcome this difficulty in solving the equations [33]. In this case, a set of equations can be written as follows:

$$A^{m \cdot n} x^n = b^m \quad (2.5)$$

A set of  $x$  values are then found:

$$x^{k+1} = x^k + A_i(b_i - A_i^\lambda x^k) \|A\| \quad (2.6)$$

where the value of  $\lambda$  is the relaxation coefficient, generally between 0 and 2, and controls how fast the routine converges. The projection being acted upon is indexed by  $i$ . When  $\lambda$  is very small we have a method equivalent to a conventional least squares approach, which can also be applied to the image after the reconstruction to increase detail [34].

## Chapter 3

# APPLIED CT FOR MEASURING BULK INDEX VARIATION

## 3.1 Fourier Synthesis Simulations

Shift invariance has given the Fourier Transform a dominant role in the reconstruction of an image from projection data [35]. Successive images can be registered by removing any shift, in the test beam, caused by the hardware. We will describe the two dimensional procedure for ease of understanding. To expand to three dimensions only requires taking several two dimensional measurements at once. The measurements are just stacked one on top of the other to build up the third dimension of our projection array. Of course, this implies that the layers are far enough apart that they don't interact with each other. In cases where the layers interact, a full three dimensional approach must be taken [36, 37, 38]. This is a large field of study, and will not be detailed here. Three dimensional reconstructions for purposes here will be made from a series of two dimensional reconstructions.

### 3.1.1 Two Dimensional CT

The test beam, from a collimated source, goes through a plane, or slice, of the sample. The irradiance detected exiting the sample is one projection as shown in Figure 3.1.

We then build up an array of projections, each at a different angle as shown in Figure 3.2. Each row in the matrix corresponds to a specific projection angle. The length of the columns is dependent on the number of projections taken, and the resolution in angle.

Each projection is an integration through the test piece. This can be represented by Equation 3.1 as a summation through the sample where  $\phi$  is the angle rotated in

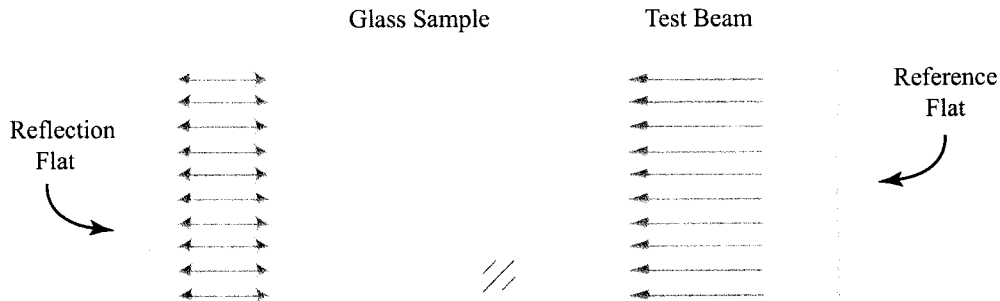


FIGURE 3.1. Geometry for making a single measurement, or projection. The integration along the projection axis is perpendicular to the rotation axis.

$x$ , and  $y$ :

$$p(x, \phi) = \int_{-\infty}^{\infty} g(x, y, \phi) dy. \quad (3.1)$$

Each projection in our array will correspond to a specific angle of rotation, and is placed into an array where the row index relates to the rotation angle, and the column index is the spatial location within the projection. We then perform a one dimensional Fourier transform row by row as shown in Equation 3.2:

$$P(\xi, \phi) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y, \phi) e^{-2\pi i(\xi x + 0y)} dx dy = G(\xi, 0, \phi). \quad (3.2)$$

All Fourier transforms discussed are based on the following formula:

$$F(\xi, \eta) = \int_{-\infty}^{\infty} f(x, y) e^{-2\pi i(\xi x + \eta y)} dx. \quad (3.3)$$

A new two dimensional array is built by taking each row of our transformed array, and placing it at the proper angle in a blank array. The row is placed at the equivalent angle as recorded during the measurement. Each row is placed on top of the previous row, but at its own corresponding angle illustrated in Figure 3.3. Overlapping cells are replaced with the most recently placed projection.

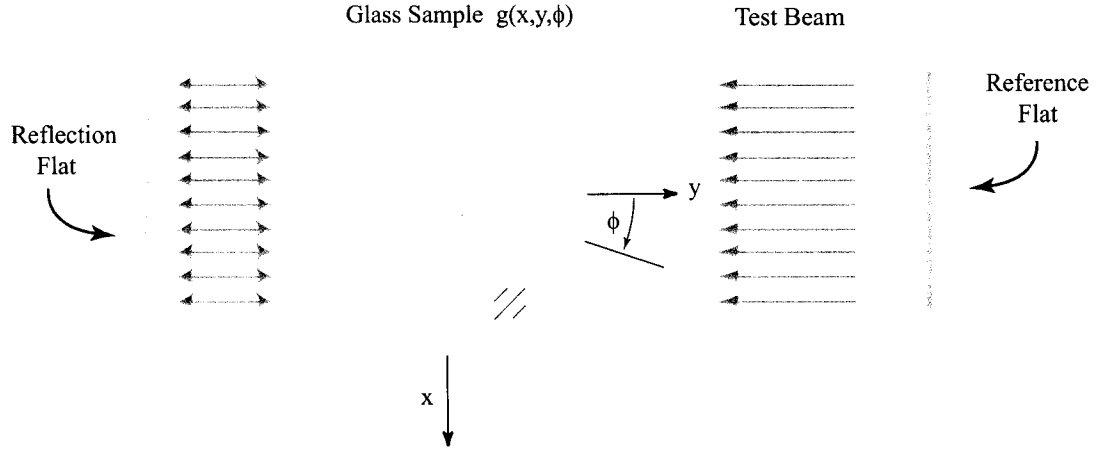


FIGURE 3.2. Rotation of sample between projections as needed for computed tomography. After each projection, the sample is rotated by a preset angle.

This new array,  $G(\xi, \eta)$ , is a sampled version of the two dimensional Fourier transform of our object, but some data points are zero since we don't have samples at an infinite number of angles.

We can illustrate the validity of taking the one dimensional transform of the projections, and using them to build a two dimensional array that will be inverse transformed. Start with the expanded form of  $G(\xi, \eta)$ :

$$G(\xi, \eta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) e^{-2\pi i(\xi x + \eta y)} dx dy. \quad (3.4)$$

The substitution,  $\xi = \rho \cos \phi$  and  $\eta = \rho \sin \phi$ , can be made to get Equation 3.5

$$G(\rho \cos \phi, \rho \sin \phi) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) e^{-2\pi i(\rho x \cos \phi + \rho y \sin \phi)} dx dy. \quad (3.5)$$

Setting  $\phi = 0$  in Equation 3.2 and Equation 3.5, we can see the similarity:

$$P(\xi, 0) = \int_{-\infty}^{\infty} g(x, y) e^{-2\pi i(\xi x)} dx \quad (3.6)$$

$$G(\rho, 0) = \int_{-\infty}^{\infty} g(x, y) e^{-2\pi i(\rho x)} dx \quad (3.7)$$

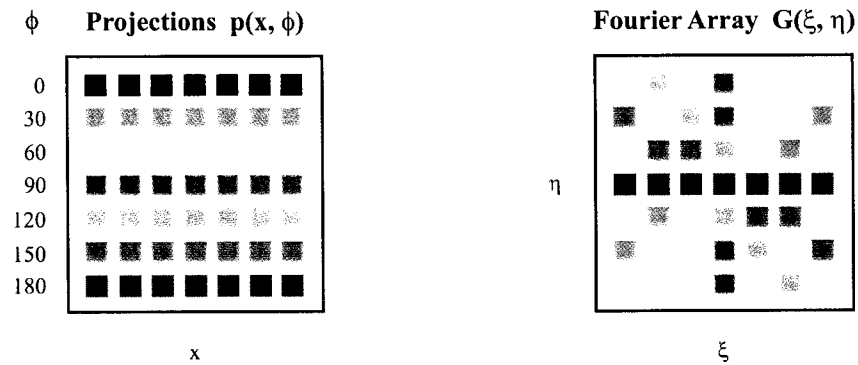


FIGURE 3.3. At left is a projection array made from individual measurements. At right is the resulting Fourier array after transforming each row of the projection array, and placing it into a blank array based on their corresponding angle. All blank locations will be filled in with the nearest non-zero value.

In fact they are equivalent operations. In Equation 3.6 the sample is rotated, and in Equation 3.7 the linear array has been rotated an equal but opposite angle before being placed into the two dimensional Fourier array.

### 3.1.2 Filling the reconstruction array

Sections of the two dimensional Fourier array will have blank values. We must fill in the rest of these points in the array, or they will adversely bias the inverse transform for the reconstruction. Methods to compensate for zero values will be discussed later in this chapter.

A simple way to fill blank values is to step through the Fourier array. Every time a blank is found, just fill it with the value to the left in the same row. This works, but causes a noisy reconstruction. A slight improvement is to find the nearest value in the same row. Better to find the absolute closest non-zero value, such as with a nearest neighbor sampling, regardless of its direction from the blank location.

Both steps can be done at once using a true nearest neighbor approach working from the Fourier array back to the projection information. Step through each location

in the Fourier array, calculate the distance from the center,  $\rho$ , and the associated angle,  $\phi$ . Find the nearest  $\phi$  value in the projection array, then find the  $\xi$  value of that projection closest to  $\rho$ , and place that value in the Fourier Array. Keep in mind that here,  $\rho$  can be negative. This stems from the fact that only rotation through  $180^\circ$  is used to completely fill the two dimensional transform space. Only stepping through half of the  $2D$  array is necessary. Since we have a projection value for this point, and its complimentary point directly across the center, why not fill both points now. So for every point we step through, we fill it, and its complimentary point resulting in  $G(\xi, \eta)$  as in Figure 3.4. Perform the  $2D$  inverse Fourier Transform on the final array to reconstruct the object.

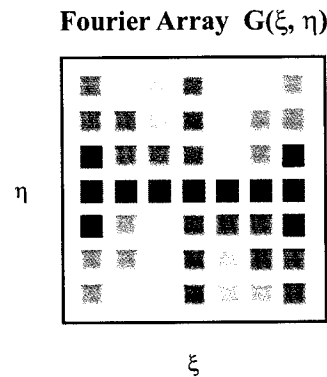


FIGURE 3.4. The Fourier array resulting when a nearest neighbor approach is used to fill each location.

The two methods differ in the direction that you fill the Fourier array. The first works from the projection array. Each projection is placed, adding the values that overlap,  $\rho$  filtering, then inverse transforming. We choose to work in the opposite direction by starting at the blank Fourier array. We then find the best value from the projections for each location. Then inverse transform to get the reconstructed object.



### 3.1.3 Filtered reconstruction

When the next projection in sequence is written to the Fourier array, it may overlap locations that already have non-zero values from a previous projection. The simplest approach is to overwrite the value each time a new projection overlaps a previously written location. This works very well, and if coupled with a nearest neighbor method, is almost the same as our current method. A small improvement in accuracy, and complexity, adds the values that overlap. As new projections are added into the Fourier array, the central location becomes a large value. A  $\rho$  filter is used to adjust the values back to the correct order of magnitude[12, 39]. We simply multiply each value by its distance from the center.

### 3.1.4 Linear Interpolation

Our specific method of reconstruction fills the array by interpolating between projections in the angular coordinate. The two closest projections are found based on the angle from horizontal of that array location. The two values are then averaged and weighted by the angular distance from the two projections. We still do a true nearest neighbor in the radial coordinate.

### 3.1.5 Gridding

The nature of sampling the object during tomography produces a projection array that is defined in polar coordinates. Gridding is one way to improve accuracy when returning from a polar grid to a cartesian array [40], and has the same accuracy as filtered backprojection. Essentially, a convolution is used to more fully fill the reconstruction array. The convolution window is often a two dimensional Gaussian function. The spatial content corresponding to index variations is heavily weighted at lower frequencies. A density weighting function is used to compensate such that the spatial frequency response is flat.

### 3.1.6 Limited Angle

Our biggest problem, and a huge hit on accuracy, occurs when the number of projections are limited in angular extent. Geometry dictates how many projection angles are valid for a given refractive media. No convenient geometries are available in glass to allow all angles.

In medical tomography there are no refractive properties of the subject to complicate the measurement. Time limitations may force a limited angle problem, or a dose limitation constraint. The heart, or other moving objects, will limit the amount of time available for making projections [41]. Even if the geometry allows any angular position, the time over which those projections must be made again introduces this error.

Using visible light, and a sample that is not circularly symmetric, will not allow  $360^\circ$  of rotation. A subset of measurements must be used as shown in Figure 3.5 for our square geometry. This is known as the limited angle problem, and can reduce the amount of information available to make an accurate reconstruction. The decrease in information that directly transfers to the Fourier array is not as disastrous for our object geometry as one might assume[42]. Many other sample types don't give good results with this error. An example of how the reconstruction degrades is shown in Figure 3.6.

Extrapolating into the missing projection area can improve the reconstruction if done intelligently. Using *a priori* information can be beneficial and is one approach [43, 44, 45]. An assumption is made that the index will vary slowly over the sample. Such foreknowledge aids us in choosing the best algorithm. Much of this work falls under the topic of superresolution [46], or analytic continuation.

Fourier transform iterative methods can improve the reconstruction as well. If feedback is provided on the finite extent of the object, the projections can be improved [47], and the noise in the reconstruction reduced. This is one method of

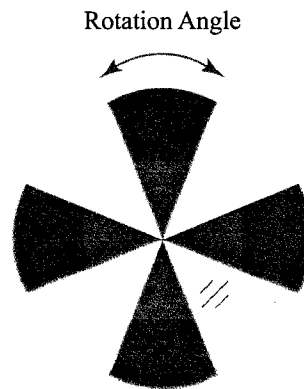


FIGURE 3.5. A sample with a limited range of angles available over which a projection can be made. The cones represent angles over which we can make measurements.



FIGURE 3.6. Degradation when limiting the measurement angle. From left to right we have a total view angle of  $180^\circ$ ,  $120^\circ$ ,  $60^\circ$ ,  $20^\circ$ .

regaining a portion of the large amount of information lost from the limited number of projections.

### 3.1.7 Region of Interest

When the available measurement aperture is smaller than the extent of the sample, only a portion of the sample can be measured[48, 49] for each projection. The test setup can often be changed to avoid this problem, but in the spirit of using instrumentation that is at hand, we would expect to see region of interest errors. Figure 3.7 shows how a reconstruction can degrade. A small feature in the sample can rotate into the measurement beam, and back out of it so that only a fraction of the projec-

tions have measured the feature in question. This can be compounded further as a limited angle is probably required if a region of interest problem is present.

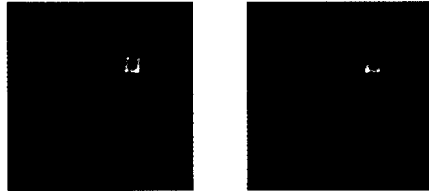


FIGURE 3.7. Effect of limiting the measurement aperture. The left is a full reconstruction while the right image has a region of interest imposed on the measurement.

We can also benefit from this phenomenon. If we limit the extent of our measurement beam, a larger range of angles can be measured. A small center region can be reconstructed more accurately at the expense of information closer to the object's boundaries. Figure 3.8 shows how the maximum angle of rotation increases as the test beam decreases in size.

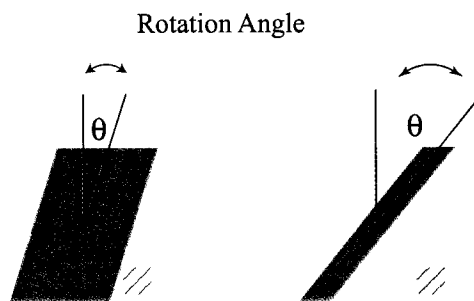


FIGURE 3.8. Increase in angular projections available when a smaller test beam is used. A large beam on the left allows fewer rotation angles where the smaller beam on the right permits a larger maximum angle.

If we consider a sample of size,  $s$ , and a test beam with radius,  $r$ , we can calculate the maximum projection angle. Equation 3.8 gives a simple relation to find the beam radius for a given projection angle.

$$r = \frac{s}{2} * (1 - \tan(\theta)) * \cos(\theta) \quad (3.8)$$

At  $45^\circ$  we lose exactly half of the beam. To get any useful data from a larger angle would require multi-sided detection. Some potential ways of increasing the angle will be discussed under *Future Work*.

### 3.1.8 Object Contrast

Applications where the density is measured require only that the aperture in question be bigger than the sample. Using an interferometer such as a laser fizeau that is double pass can introduce unwanted information in the measurement. When the aperture is bigger than the sample, the portion of the beam that doesn't intersect the sample is reflected by the reference flat and becomes part of the measurement. The optical path contributed by the flat must be removed from the measurement.

Simulations so far have used a background of zero, and an internal feature having a magnitude equal to one. Object contrast is then equal to one, and we get the reconstruction as before. If we change the contrast of the object to  $\frac{1}{2}$ , or  $\frac{1}{5}$ , then the feature of interest starts to fade into the background of the reconstruction as shown in Figure 3.9. This decrease in contrast of the reconstruction is much closer to the internal structure of our object type. The measured refractive index will result from the reconstruction of our sample.

Revised algorithms have eliminated this problem, and are given in Appendix A. Padding the input by increasing each dimension to three times the original size, and adding linear interpolation eliminates the artifacts shown in Figure 3.9 down to a contrast of below  $\frac{1}{100}$ .

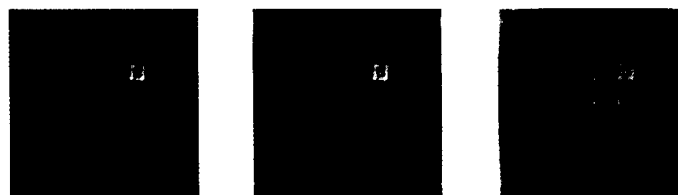


FIGURE 3.9. Our reconstruction for an object contrast equal to one is on the left. In the center we see the effect of decreasing the object contrast to  $\frac{1}{2}$ , and on the right of decreasing to  $\frac{1}{5}$ . No noise has been added to the object.

### 3.1.9 Effect of sharp change in values

If values in the Fourier Array vary by a large amount pixel to pixel, this approach still leaves a step between the two projections used. Artifacts result that are a function of the high spatial frequencies created by the sharp transition. The magnitude of such artifacts presents a problem if they are larger in magnitude than the spatial frequencies of interest.

As mentioned before, large changes in index over a small distance will not be discussed here. Such a problem does arise in practice as in the case of measuring the index profile of a fiber optic [50, 51, 52], or capillary tube [53].

## 3.2 Circular Harmonic Decomposition

By decomposing the object circularly, we can provide much better correction of the limited angle problem. In Fourier space the lack of a small range of angles shows up as a blank wedge. Using a nearest neighbor routine in Cartesian coordinates to fill in the missing area fills the blank spaces, but leaves a sharp transition where the values of two adjacent projections meet. Some interpolation methods reduce the effect of sharp transitions, but introduce distortions if based on a Cartesian coordinate system. Since the missing projections are based on the angular coordinate in the Fourier domain, it makes sense to use a method that is based on the rotational nature of the missing

data. A method of circular decomposition and filtering will be presented next.

### 3.2.1 Filling Zero Projections

Blank projections can occur in the data in very different ways. The limited angle problem, as described before, can show up as a set of opposing wedges in the reconstruction array. For large areas, it will introduce large errors to interpolate or select values from adjacent projections. We would only expect to reach a few angles into the blank space before the values became unrelated to the data. This would decrease the sharp transition that could introduce high spatial frequency artifacts into the reconstruction by making a smooth shift to the empty space. This can be accomplished by extrapolation, or using a Fourier technique to smooth the data into the blank areas. Smoothing methods will be discussed in more detail later.

Zero projections can also occur by having the available projections separated by a large angle. A projection array built up from  $180^\circ$  range of projections may have zeros at every other angle. This could be an experimental constraint, or an intentional decrease in data to lower the time needed for analysis.

To simplify the model, we treat the glass as if no gradients in index are present such that the index is slowly varying compared to our sampling frequency. The slowly varying object is mapped to a rectilinear projection array. We then gain an advantage when we start our polar reconstruction routine. At the center of the new array, the missing projection values cease to be a problem. Depending on the sparseness of the projection array, within a known radius the array values are not blank, and the missing angle problem disappears for these low spatial frequencies.

For values of  $\rho$ , in our reconstruction array, that are not angularly oversampled, linear interpolation along the theta coordinate can help. Instead of choosing the closest value out of the projection array, pick the two closest values in  $\theta$ . To find the correct value to place in the two dimensional Fourier array, do a linear interpolation

weighted by the distance between the two valid points.

Three dimensional reconstruction from an incomplete data set can be overcome [41]. Using available prior information, and iterative techniques, a reasonable reconstruction can be created. In a method like this, two dimensional data is used to build the three dimensional reconstruction.

### 3.2.2 Projection Smoothing

Each projection is written into an array by labelling the row according to the rotation angle. Such an arrangement is exactly the starting point to calculate the circular harmonics. Before the reconstruction starts, we take the one dimensional Fourier transform in the  $\theta$  coordinate. Simply calculate the transform along the columns instead of the rows. The Fast Fourier Transform (FFT) built into the IDL<sup>1</sup> programming environment is the specific tool used for these operations.

The Fourier Series is one method to fill in areas left blank in experimental data [54, 55]. In general the Fourier Series of the projection would be approximated by Equation 3.9 and Equation 3.10 [56]:

$$g(r, \theta) = \sum_{M=-\infty}^{\infty} g_m(r) e^{im\theta} \quad (3.9)$$

where the coefficients are given:

$$g_m = \frac{1}{2\pi} \int_0^{2\pi} g(r, \theta) e^{-im\theta} d\theta. \quad (3.10)$$

The center columns of the transformed projection array represent the lower frequencies, and are sampled more than the outer columns as a result of rotating the glass sample about its center. The smoothing can be done by either a Gaussian filter, or by any method that decreases the contribution from high spatial frequencies. Just

---

<sup>1</sup>Research Systems Inc.



multiply the transformed columns by an appropriate filter, then inverse transform. The new projection array is now ready for use in reconstruction.

Figure 3.10 shows one example where the circular harmonics were calculated by transforming the columns, filtering, then inverse transforming. Here, we replace the nonzero rows with our original data. An approximation is now in place where blank projections were previously located. Only high spatial frequencies have been rebuilt in this example. No improvement was measured in reconstructions of our objects. The objects are primarily of low spatial frequency content to more closely match the slow variations typical in glass.

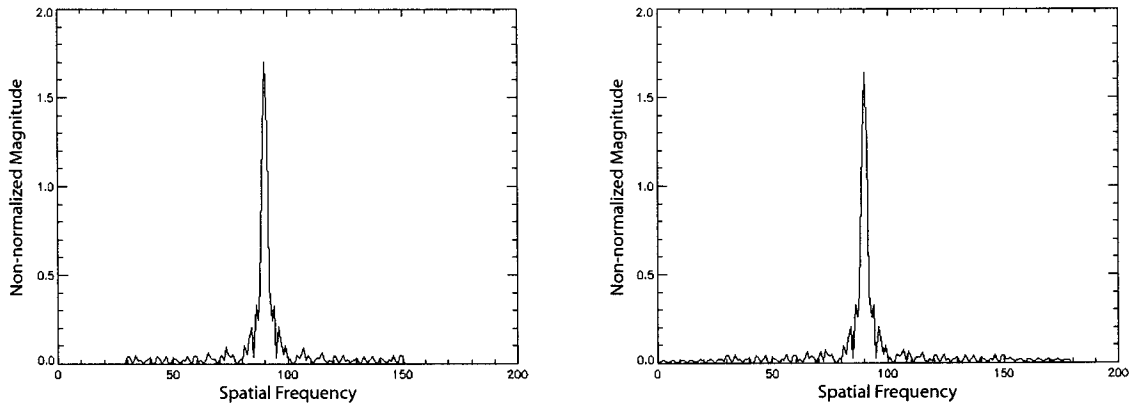


FIGURE 3.10. The left plot shows blank projections that are replaced by approximations at right.

### 3.2.3 Windowing

Filtering can take place at two likely steps in the process of reconstruction. The circular harmonics can be manipulated as discussed above, or the final two dimensional array can be filtered before the inverse transform is performed. Filtering the final array is more effective since the act of reconstruction is also a sampling process. Any filtering done before, will be compromised by the reconstruction process, and limit the resulting accuracy.

Artifacts in the resulting image can be created by sharp gradients as a result of the missing data. Even with a combination of nearest neighbor and linear interpolation, the gradients aren't completely smooth. Applying a Gaussian filter, or Hanning window [57], rounds off the remaining sharp transitions.

Figure 3.11 shows an example of applying a Hanning window. Pictured is a two dimensional reconstruction immediately before performing an inverse Fourier transform. Notice the high frequency contributions in the corners of the array. These are extrapolated from our projections. This increases the noise in the reconstruction resulting in high frequency artifacts. As mentioned earlier, our objects are taken to be smooth and slowly varying. Any high frequency components are not physically representative in our model.

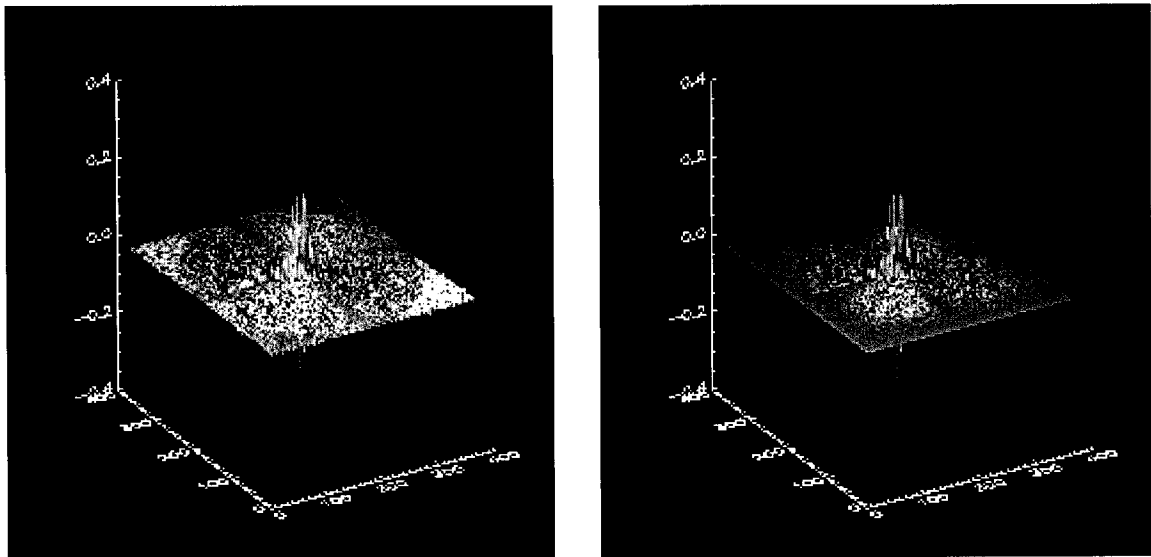


FIGURE 3.11. The left plot shows the reconstruction without using a windowing function. The plot at right is multiplied by a Hanning window.

### 3.3 Measurement Quality

It is important to have a metric with which to grade the quality of our algorithm. We strive to quantify how closely the reconstructed image is to the original object. Implementations of the root mean square (RMS) difference and the cross correlation are discussed next. Numerical and Fourier methods of calculating the cross correlation are given.

#### 3.3.1 Root Mean Square

Now we calculate the RMS of the difference between our original object and the resulting reconstruction. We get a measure of how much noise and other information is introduced by the reconstruction process. Subtract the two bias-corrected images by finding the difference after subtracting their respective means. Calculate the RMS of the resulting array. This should give some measure of how close the reconstruction is to the original. However, at a large enough limited angle, or sparseness of the projection angles, the reconstruction gets artificially flattened. The RMS value of the difference will then asymptotically approach some value. The RMS is calculated in the following way:

$$RMS(x) = \sqrt{\frac{\sum_1^N (x - \bar{x})^2}{N}} \quad (3.11)$$

Where  $N$  is the total number of points used. Figure 3.12 shows the improvement when the projections are split. For the dual axis case, the projections are split as if they had been taken through two different sets of parallel faces. We have the same number of total projections as in the single axis, but they are in two perpendicular sections instead of one. Angular sampling is better in the case of splitting the projections into to separate sections.

We can see that more projection angles is better. Using the dual axis method, 60

degrees total of available angles produces a useful reconstruction. At 40 degrees we can visually recognize features of the object, but our confidence in the accuracy goes down. Figure 3.13 shows an object with full angular sampling, and at 60 degrees and 40 degrees. Here the full sampling consist of 360 projections traversing 180 degrees.

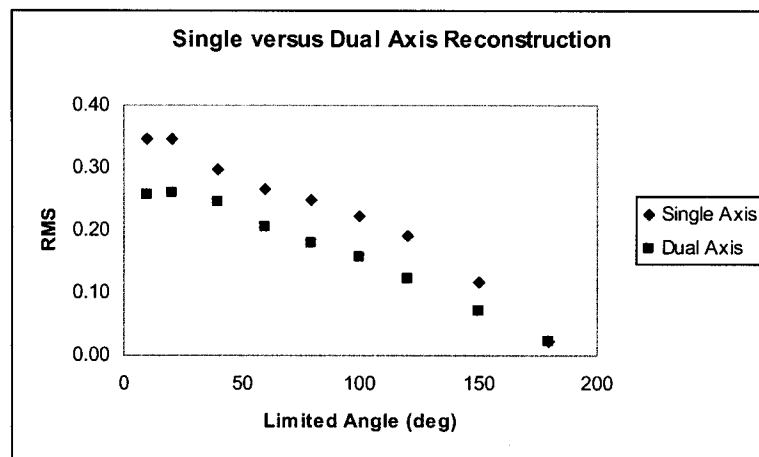


FIGURE 3.12. RMS is given comparing single and dual axis placement of the projections. The dual axis method increases the accuracy even when the same number of projections as the single axis are used.

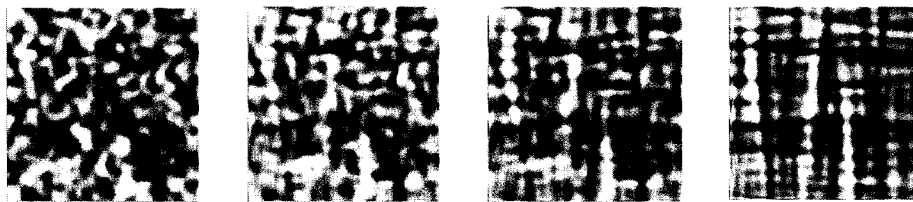


FIGURE 3.13. The original object is shown at left. As we go right, we have images with limited angles of  $60^\circ$ ,  $40^\circ$ , and  $20^\circ$ .

### 3.3.2 Correlation

The correlation is used to quantify how accurately our reconstruction recreates the object. Correlation gives a measure of how much information is lost by measuring

the covariance of our two arrays. This is the ratio of the observed covariance divided by the highest possible covariance. When the observed covariance is equal to the largest possible covariance, the correlation will have a value of 1.0, indicating a perfect match. In the context of imaging, the highest possible covariance is the magnitude of the autocorrelation which is the normalization factor mentioned above, and discussed below.

Equation 3.12 shows a numerical method for calculating the cross correlation. When the correlation is 1.0, our object is perfectly returned. A perfect reconstruction is never achieved as the pixels have a finite size. If each element in our Fourier Array was infinitesimally small, and our interpolation was perfect, then an exact replica of the object would be returned. The jitter from the finite pixel size introduces a small error. We have to pick a value to put into each array location, and it may not be the ideal value for that location. Blur is visible in the reconstruction when the nearest neighbor is used, but with linear interpolation in the angular coordinate the effect isn't visible.

$$C = \frac{\sum_{i=1}^N (x_i \cdot y_i)}{\sqrt{\sum_{i=1}^N (x_i)^2} \cdot \sqrt{\sum_{i=1}^N (y_i)^2}} \quad (3.12)$$

Equation 3.13 shows the normalized complex cross correlation.

$$\gamma_{fh}(x) = \frac{\int_{-\infty}^{\infty} f(x)h^*(x - \alpha)dx}{\sqrt{\int_{-\infty}^{\infty} |f(x)|^2 dx} \sqrt{\int_{-\infty}^{\infty} |h(x)|^2 dx}} \quad (3.13)$$

In analytic form, the correlation is a projection, or dot product, in a Hilbert space. This complex vector space requires that any function,  $f$ , be square integrable and finite [30, 58]. Equation 3.14 defines the requirements for any function on the real interval  $[a, b]$ .

$$\int_a^b |f(x)|^2 dx < \infty. \quad (3.14)$$

If it is required to find the correlation in Fourier space, the calculation is easily done. The correlation is the product of the transform of the first function and the complex conjugate of the second function's transform [59, 60]. The result is Equation 3.15.

$$\gamma_{FH}(\xi) = \frac{F(\xi)H^*(\xi)}{\sqrt{\int_{-\infty}^{\infty} |f(x)|^2 dx} \sqrt{\int_{-\infty}^{\infty} |h(x)|^2 dx}}. \quad (3.15)$$

The normalization factor is the same for both calculations. Normalization of the complex cross correlation is done relative to the maximum value of the unshifted self correlation, more commonly referred to as the autocorrelation where  $f(x) = h(x)$ . Equality of the square integrals, for the special case of the autocorrelation, is given by Parseval's theorem shown in Equation 3.16

$$\int_{-\infty}^{\infty} |f(x)|^2 dx = \int_{-\infty}^{\infty} |F(\xi)|^2 d\xi \quad (3.16)$$

Using the Fourier method, we separate out the component of the reconstruction attributed to noise from the component that represents the input object. The idea is to calculate the cross correlation of the output with the known input to find the amount of the object present in the output. A scaled version of the input is the first component needed in the comparison, and the scale factor is given by the maximum value of the inversely transformed cross correlation. The second component is the resulting reconstruction. Figure 3.14 shows an example where 540 projections were used with no limited angle constraint imposed. In this case, only single axis reconstruction is used. We find the correlation value, then multiply it by the object. The reconstruction is subtracted, and the result is the error remaining after the reconstruction has taken place.

Figure 3.15 shows an example where 540 projections were used when  $120^\circ$  limited angle constraint is imposed resulting in a correlation coefficient of 0.8566. In this case, only single axis reconstruction is used.

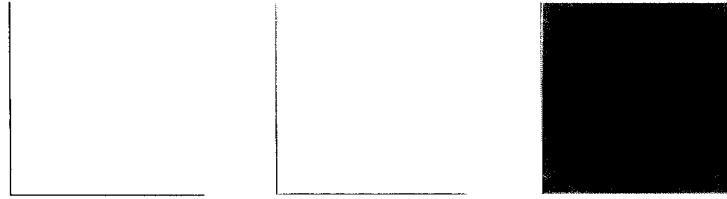


FIGURE 3.14. At left, the original object is shown as created with a Gaussian filter width of 40 pixels for a 256 pixel array. The center image is the reconstruction with a correlation coefficient of 0.9814, and the right image is the difference. Only a very small component of the reconstruction is error.

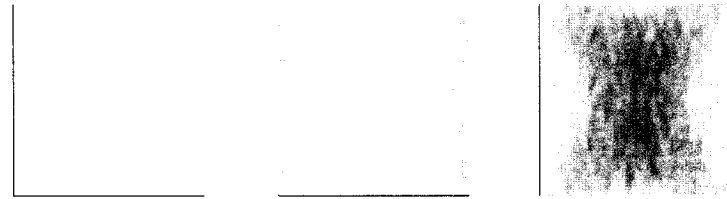


FIGURE 3.15. At left, the original object is shown as created with a Gaussian filter width of 40 pixels for a 256 pixel array. The center image is the reconstruction with a correlation coefficient of 0.8566, and the right image is the difference. A 120 degree limited angle constraint has been imposed.

We can see some change in the correlation as the spatial scale of the object changes. Objects with lower spatial frequency content are less tolerant at severe limited angles, such as restrictions to less than  $20^\circ$ . Figure 3.16 shows the decrease in reconstruction accuracy. The lower frequency objects are made by applying a smoothing function of a specific width in frequency space.

Error bars are present on the object data to give a feel for the variation in the data. We choose to use the built in standard error calculation used by Excel as shown in Equation 3.17

$$\text{Standard Error} = \sqrt{\frac{\sum_{i=1}^m \sum_{j=1}^n x_{ij}}{(n_x - 1)(n_x)}} \quad (3.17)$$

where  $i$  and  $j$  are the array locations, and  $n_x$  is the number of elements in the array.

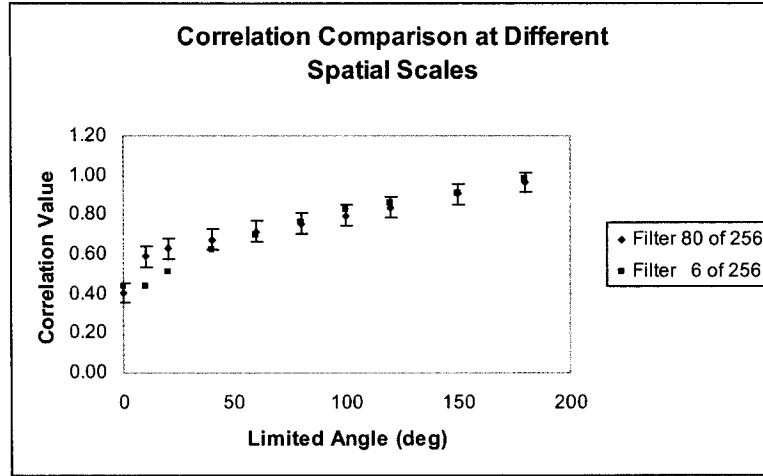


FIGURE 3.16. Correlation change with spatial scale are shown. The object is 256 pixels wide. We can lower the spatial content by filtering with a Gaussian. The width given is the width of the Fourier space filter.

### 3.3.3 Number of Projections

We start with a square grid with  $N^2$  total points, and the physical spacing between points is  $\Delta x$ . The radius of the array can be defined by drawing a line from the center of the array to the center of one edge. Label the distance as  $R$ , and we see that  $R = \frac{1}{2}N\Delta x$ . An angle,  $\Delta\theta$  can now be defined such that  $R\Delta\theta = x$ . By solving for  $\Delta\theta$ , and substituting for  $R$ , we can show that  $\Delta\theta = \frac{2}{N}$ . To get full projection coverage by sampling over 180 degrees, we can see the number of angular samples needed in Equation 3.18.

$$n_p = \frac{\theta}{\Delta\theta} = \frac{\pi}{2/N} = N\frac{\pi}{2} = 1.57 \cdot N \quad (3.18)$$

For an array that is 256 by 256, we should have 402 angular samples. The correlation value is then 0.980. If we only use 180 projections we still have a correlation value that is 0.975. Even going to 540 projections, the correlation value only increases to 0.981, but the total calculation time is markedly increased as shown in



TABLE 3.1. Calculation time versus the number of projections.

Number of Projections	Time (sec)	Correlation Value
540	315.240	0.9814
450	254.065	0.9811
360	197.721	0.9806
180	130.504	0.9748
90	71.845	0.9528
60	56.106	0.9331
30	42.701	0.8528
20	37.594	0.7836

Table 3.1. Figure 3.17 shows the correlation as the number of projections is changed. One projection per degree results in 180 projections, and we can see that using more projections only offers a small increase in accuracy. We use 360 projections to ensure good reconstruction in our analysis. A cutoff of 60 degrees of total projection angle is reasonable, and provides the minimum information needed for most uses.

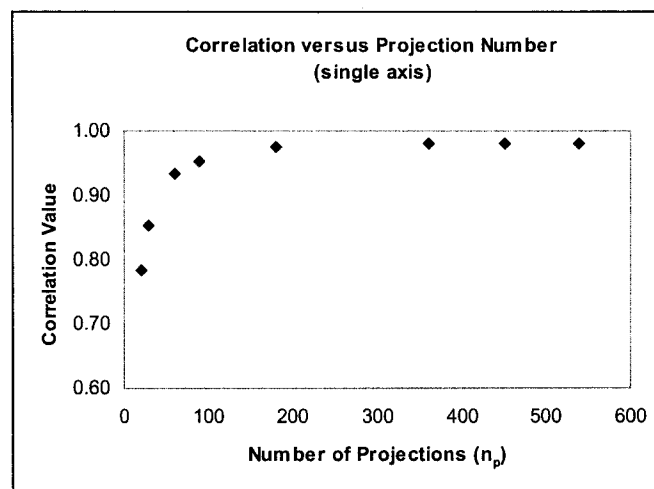


FIGURE 3.17. Correlation as a function of the total number of projections used in the reconstruction.

### 3.3.4 Signal to Noise Ratio

We can use the Signal to Noise Ratio (SNR) to characterize the quality of our reconstructions as well. A ratio of the standard deviations is used to calculate the SNR. The signal is the output of our system, or the reconstruction, and is the best estimate of our original object. To get the noise, we subtract a scaled version of the object that represents the portion of the output that is correlated to the object from the output. Equation 3.19 shows the fractional form of the SNR.

$$SNR = \frac{\sigma_{signal}}{\sigma_{noise}} \quad (3.19)$$

We expect the SNR to change as the spatial content of the object changes. Spatial content of the object is varied by multiplying by a filter in Fourier space whose width is given in the chart. A filter of width, 6, will smooth the object much more than a filter of width, 80. Figure 3.18 illustrates the different trends as the object's spatial scale is varied.

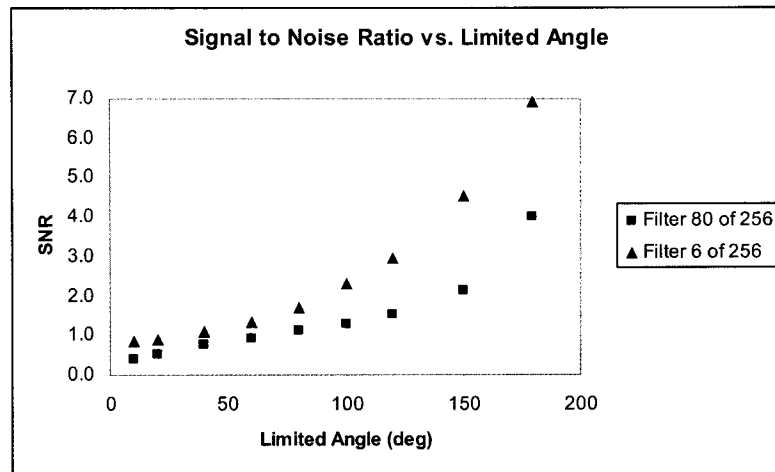


FIGURE 3.18. Signal to noise change with spatial scale is shown.

If we expand the region around  $SNR = 1$ , we can better see how the signal to noise varies with the spatial content of the object. Figure 3.19 shows that coarse

objects are more tolerant to a smaller limited angle. The coarsest object doesn't drop below  $SNR = 1$  until  $30^\circ$ . An object with more high frequency information must have nearly  $45^\circ$  to get the same quality reconstruction. Objects with the highest frequencies used in our analysis, require a minimum of  $60^\circ$  of projections to reach the same level of output.

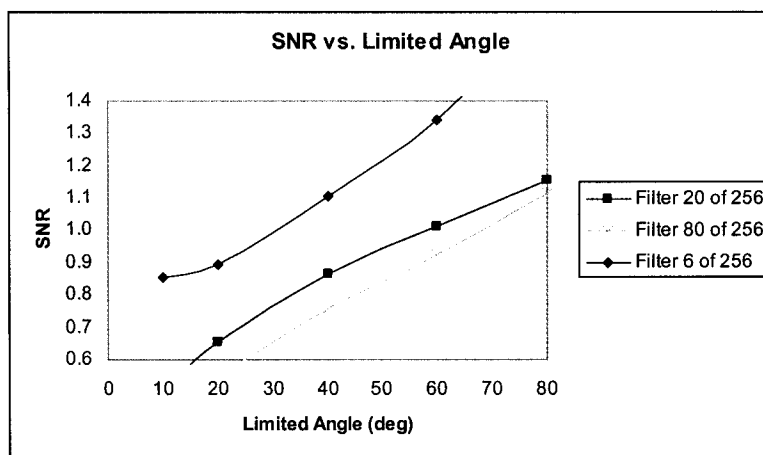


FIGURE 3.19. Signal to noise change with spatial scale is shown. A smaller amount of data is shown to emphasize crossing points for  $SNR = 1$

A signal to noise ratio of one was chosen as a breakpoint since the noise is equal to the signal at that point. The reconstruction below that point has become little better than a guess, and does not increase our confidence that the object has been reconstructed. Systems that are limited so as to produce such low quality output should be modified to increase the information available for analysis. We present the SNR as an additional metric for quantifying the quality of our tomographic reconstruction.

## Chapter 4

# EXPERIMENTAL

In practice, we calculate the wavefront surface and use that as the optical path (phase difference) through the sample. This is the optical path difference(OPD) of the test beam through the glass. When we make a measurement, we are finding  $\Delta\phi$  in equation 4.1:

$$\text{Phase Object} = e^{i\Delta\phi} \tag{4.1}$$

$$= e^{i \int n ds} . \tag{4.2}$$

The difference in phase is calculated from the interferogram using phase shift interferometry [61]. The phase map becomes one projection in the set of projections used in the tomographic analysis.

As a method of checking the measurement technique, medical scanners use a simulated object of varying densities. More accurately, it will have some known variation in transmission at the operating wavelength in x-rays, and is called a phantom. It is much more difficult to make a phantom for our use as the geometry of the sample will affect our measurements due to refraction of the test beam. If we keep the same geometry, but decrease one dimension of our cube, we can eliminate one degree of freedom in order to check our method. A good sample is a microscope slide. It should be possible to reconstruct it as a small portion of the field of view. Another possibility is to use an opaque object with a complex shape, such as a bulk letter 'C'.

Reconstructions will give some feedback on the surface quality, although not as accurate as direct interferometry of the surface. Various errors will be discussed later in Chapter 4, and show that the surface errors alone create deviations smaller than our pixel size.

## 4.1 Sample Geometry

The test setup is a laser Fizeau<sup>1</sup> used in a double pass transmission configuration as shown in Figure 4.1. Such a setup is typical for an optics shop, but a single pass system will be considered in the future. The detector has a maximum resolution of 512 by 480, but we normally use 256 by 240 to keep the time needed for data analysis manageable. A six inch cross section gets mapped to 256 by 240 pixels. Each pixel then represents a  $.59\text{ mm}$  by  $.625\text{ mm}$  area of measurement.

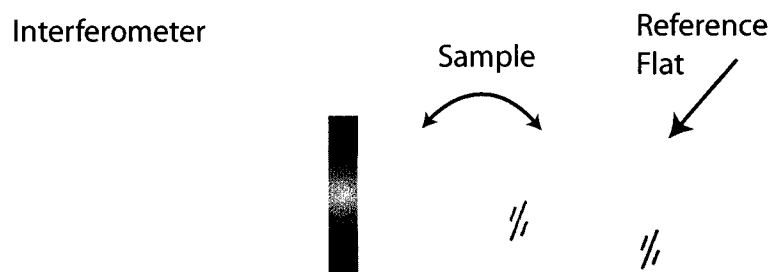


FIGURE 4.1. Double pass configuration used for transmission testing of glass samples.

As the sample is rotated, Figure 4.2 shows how large angles from perpendicular cause TIR at the receding surface and scattering at the approaching surface. Ray bending isn't a problem if the surface pair has good parallelism. A displacement will occur as the ray exits the glass which then shifts it back to its original angle on the second pass through the sample. The deviation in the output angle is a result of the homogeneity change we want to measure. The change appears in our interferogram as an optical path variation.

At large angles, the beam as it goes through the sample will be foreshortened. This dimension can be scaled to compensate for the smaller volume of the sample that is measured. The beam size as it hits the detector will be too large, and we can

<sup>1</sup>WYKO 6000 six inch laser Fizeau interferometer operating at  $\lambda = 632.8\text{ nm}$

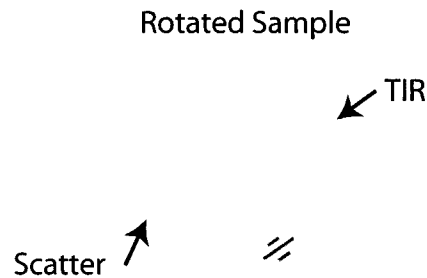


FIGURE 4.2. Some problems appear for large angles of incidence. At large angles, total internal reflections occurs for the transmitted rays, while rays missing the front surface are scattered out of the test system.

scale one axis of the output data to accurately reflect the volume of glass that was tested.

## 4.2 Background compensation

Background information can be present in the measurement when rays are returned into the system, but did not go through the test piece. Here, we define the background as the portion of the measurement that doesn't go through the sample at all. This can be eliminated by taking a null measurement at the start if the setup allows for it. Subtracting the null from each measurement eliminates the error.

## 4.3 Multiple axis information

The limited angle problem creates such a huge hit in accuracy that anything we can do for an improvement pays big dividends. If we look at a cube of glass, the limited angle shows up by limiting the measurement to a finite number of angles over which a projection is possible. This pair of parallel surfaces presents a classic limited angle geometry [43]. We can include additional angles in our reconstruction by incorporating measurements through the other set of parallel surfaces. Fewer projections are zero, and the reconstruction increases in accuracy.

Moving to three dimensions, each set of parallel surfaces can provide two sets of limited-angle data. For one set of surfaces, we average the two data sets in an intelligent way. A total of six sets of the original limited angle information is now available for use in our reconstruction array. By rotating about two different axes, we double the information acquired about the sample, and reduce the error in the measurement.

#### 4.4 Demonstration

The experimental setup is shown in Figure 4.3. Some care must be taken to securely mount such a large piece of glass especially when doing the vertical rotations. Rotation stages that can handle the weight of large samples were used.

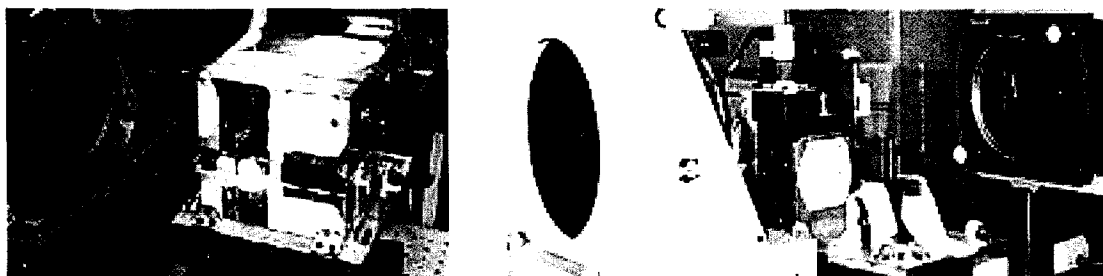


FIGURE 4.3. A cube glass sample is shown at left, and a thin glass plate is shown at right.

Typical interferograms are shown in Figure 4.4. The left image is rotated  $15^\circ$  clockwise, and the right image is rotated  $15^\circ$  counter-clockwise. At zero rotation, the incident face of the sample is perpendicular to the aperture test beam from the interferometer.

Figure 4.5 shows interferograms from transmission through a different pair of faces. Foreshortening of the sample can be seen where the vertical sides have moved inward from the left and right of the measurement.

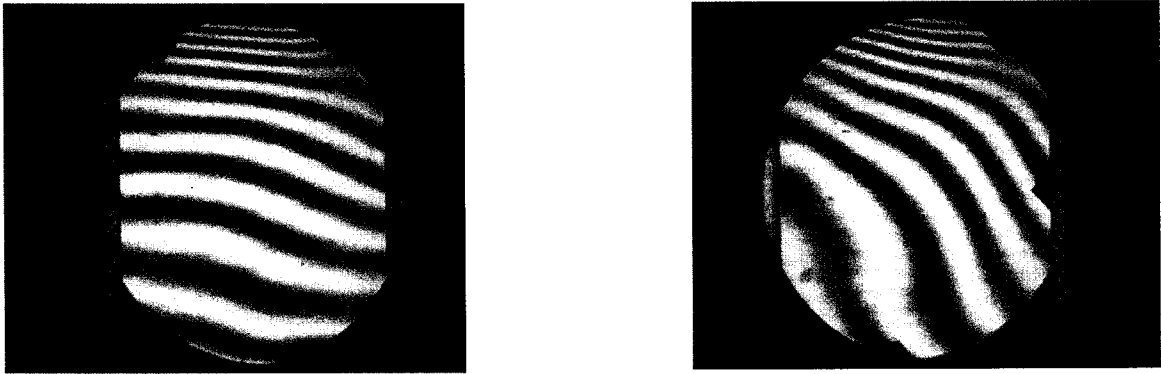


FIGURE 4.4. Interferogram at left has been rotated  $15^\circ$  clockwise from where the face is perpendicular to the interferometer. At right, the interferogram has been rotated  $15^\circ$  counter-clockwise.



FIGURE 4.5. Interferogram at left has been rotated  $15^\circ$  clockwise from where the face is perpendicular to the interferometer. At right, the interferogram has been rotated  $14^\circ$  counter-clockwise.



## 4.5 Result Comparison

There are two different ways of comparing images, visual and numerically. The eye is very sensitive to changes in brightness and contrast. A simple calculation can often overlook some changes spatially or in gray level. The content of the image will determine how the reconstruction should be compared to the original object.

### 4.5.1 Visual

When comparing two images visually, the perceived difference may catch the eye and overwhelm a more subtle artifact [62]. The idea is to process the reconstructed image based on the spatial distribution of the gray scales in the original image. A simplified method of standardizing the two images is to use a selected histogram as a reference. Equal probability quantization [63] is the common method, but does not factor in the spatial arrangement of the gray values.

We can also adjust the reconstructed image by making both means zero to remove any bias. Take the ratio of standard deviations to force the processed image to have the same magnitude variation as the original. This is one way of scaling the processed image. Then add the mean of the original back into the processed image. In essence, the brightness and contrast has now been matched to the original image allowing for a much better visual comparison.

### 4.5.2 Numerically

The Root Mean Square (RMS) is often used to fit data, do estimates, and measure errors. It is a good first analysis on how well two data sets compare [64]. More detailed analysis information concerning the quantification of errors is given in section 3.3.

### 4.5.3 Initial Reconstructions

Preliminary results have severe artifacts present that have a much higher magnitude than the features of interest. Figure 4.6 shows the lines that seem to overpower the any other information in the reconstructions. The artifacts have the same shape as those seen in Figure 3.6 that arise as a result of having a limited angle data set. A limited data set in Fourier space is two long and thin wedges, and zero at every other location. The steep transition these wedges create are responsible for the line shape of the artifacts.

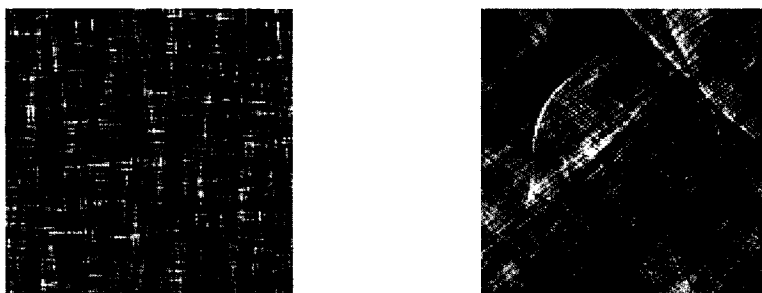


FIGURE 4.6. A slice perpendicular to the rotation axis at its midpoint from the 3D reconstruction for the cube glass sample is on the left. The slice shown at right is also perpendicular to the rotation axis a quarter of the way from the top.

### 4.5.4 Final Reconstructions

Surface maps, as shown in Figure 4.7, are made from interferograms in our double pass test system. Wavefront tilt is removed from all of the measurements. Tilt along the vertical axis of our interferograms can result from a small error in the position of the sample. Current measurements are made by rotating in azimuth, but at present the elevation isn't changed. As a result, the vertical direction is parallel to the reference surface at all times. Small deviations from perfectly parallel will introduce tilt that we know isn't contributed by the homogeneity of the glass. The trade-off

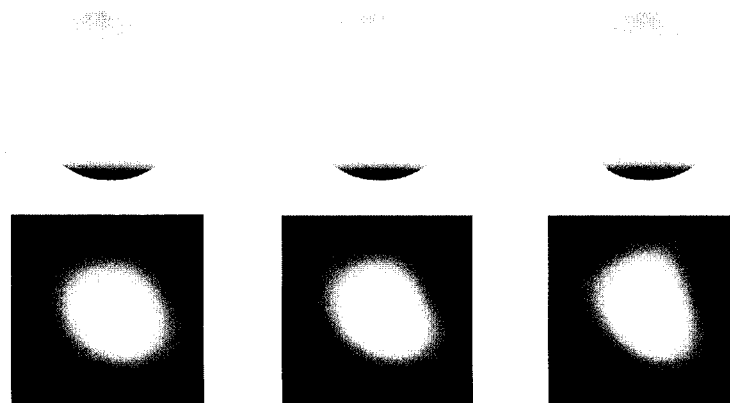


FIGURE 4.7. From left to right we show projections at a rotation angle of one degree, five degrees, and  $10^\circ$  made from a three dimensional reconstruction. The top row shows surface maps created from interferograms made through the sample. On bottom is the corresponding projection for the same angle of rotation.

is a tilt ambiguity in the projections that will carry over into the three dimensional reconstruction.

High spatial frequency noise is minimized in the reconstruction by filtering the three dimensional projection data before it is processed. The raw data cube is smoothed by multiplying the Fourier transform of the data with a three dimensional Gaussian.

Figure 4.8 illustrates the variation of structure within the reconstruction. The full projection through the reconstruction is shown for a rotation angle of one degree. Smaller sections are shown that only integrate through a portion of the data set. For this example, the reconstruction is split into thirds.

Angular extent for the above case is limited to  $30^\circ$ . Referring to Figure 3.16, our example falls at a correlation value of 0.60. This is not a very high confidence level for the reconstruction, but more projection angles are needed to improve the accuracy. A value of 1.00 for the SNR is found by inspection of Figure 3.19. Noise is potentially as large a part of the reconstruction as the signal.

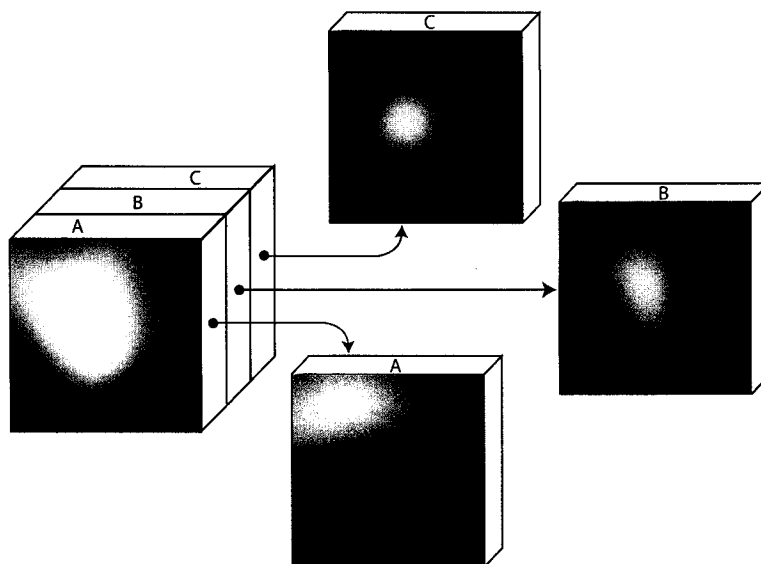


FIGURE 4.8. A projection from our data cube is shown, and three slices are shown. Each slice is a projection through one third of the data cube. The images correspond to a projection angle of one degree.

## 4.6 Error Analysis

Our tomographic routines treat a three dimensional object as a set of two dimensional objects that are stacked vertically. Any interaction between the layers is assumed to be small, and is ignored for our calculations. Additional information can be gathered by rotating the sample so that planes are in the perpendicular direction.

### 4.6.1 Ray Bending

One possible source of error is refraction of the measurement beam as it goes through the glass sample. First, we look at what happens if a beam enters the glass at a location where the index is different than the average index by one part in a thousand,  $\Delta n = .0001$ . Using Snell's law and propagating through six inches of glass, we find that the ray has moved by  $.005390 \text{ mm}$  in our detection plane. This is much smaller than the mapped size of our pixel as calculated earlier. The index would need to

change by more than .01 before a ray would walk onto the adjacent pixel.

Another error is encountered when a surface error is present. If there is a local tilt at the surface, the ray will propagate through the glass at an angle different than the rest of the rays entering the surface. If we choose a conservative surface slope created by a  $\frac{\lambda}{4}$  depression 25 *mm* away, we get a ray deviation of 3.96  $\mu m$ .

If one of the two parallel faces is tilted, forming a wedge shape, this will introduce a global shear to our interferogram. It is not part of the wavefront that we are interested in measuring. The global tilt will be removed before the analysis, so this error will not be in the processed measurement.

#### **4.6.2 Accuracy**

We subtract the reconstruction from the original and calculate the RMS of the difference. When a large limited angle is present, the reconstruction is artificially flattened, and the RMS number can stabilize at a number smaller than the peak error.

## Chapter 5

# CONCLUSIONS

The initial phase of an effort to improve optical glass inhomogeneity testing has been discussed. Using readily available interferometric tools and techniques is a primary goal of this research. Eliminating the need to use index matching fluid, whether to attach windows to a sample or as a bath[65, 66], is another important goal. Use of common equipment and a simple algorithm would make inhomogeneity measurements more accessible. It would allow a specific piece of glass to be analyzed to insure internal features aren't incompatible with the final use of the glass. This could save many hours of labor grinding and polishing surfaces on a lens from glass that will not meet the intended goal.

Tomographic reconstruction has the advantage of recreating information along the measurement axis that is lost during single measurement techniques. A simple direct Fourier method of reconstruction has been presented in detail. Many different methods of reconstructing an image using tomographic techniques are available[67, 68]. Only a small portion of known algorithms have been compared here.

We have also presented four ways to quantify the accuracy of the reconstruction. They include the root mean square (RMS) of the error between the reconstruction and the original object, the numerical correlation, the complex cross correlation in Fourier space, as well as the signal to noise (SNR) ratio. Each method is a valid measure of the quality of reconstruction.

## **5.1 Future Work**

### **5.1.1 Statistical model**

It is our desire that this method of measurement be used to gather data on glasses of interest. Data collection is simplified, and if several different measurements of a glass type are taken, then typical behavior for that glass can be cataloged. As a useful tool, we can use a measurement of a specific piece of glass to aid us in its use, but more data is necessary to build a predictive general model.

### **5.1.2 Multi-axis measurements**

Our models and measurements are based on using one axis of rotation for tomography that reconstructs one plane at a time. We have shown that the addition of new projections taken through the orthogonal set of faces, adds significantly to the accuracy of the result. If we expand the measurements to include two orthogonal axes of rotation through each set of faces, we expect an additional improvement in the accuracy of the reconstruction.

How best to take advantage of the additional data is left as a future exercise. Whether averaging, three dimensional convolution, or another approach will yield optimum results is unclear at this time. Information from the additional rotation axes is the next method to try for dealing with the severe limited angle problem.

### **5.1.3 Lens design modelling**

A broader application would be a statistical model used in a lens design code. The statistics of a glass, or even a particular batch of a single glass, could be derived from many measurements using the power spectral density as a way to quantify each data set. Tolerances could then be applied in the design code based on the measured statistics.

The effects of inhomogeneity on image quality can be simulated by defining a PSD and applying that to one element in an optical design. The sensitivity of each parameter is checked and is probably best done by using a Monte Carlo approach[69]. Things may be further complicated since multiple inhomogeneity configurations could be generated based on a single PSD.

Here the case would consist of a statistical model for the PSD made from actual measurements. Multiple versions of an optical element would give the same PSD. Each element would be put into the design to provide a statistical likelihood of creating the sought after image quality. Each of those results could be compared to see how variations in homogeneity affect the image quality, even when they have the same PSD.

It would be of use to know how the different spatial frequencies interact with specific tolerances. For instance, one spatial scale may be more affected by tilt while another might be more sensitive to a decenter of that element. Other factors, such as wavelength, may be affected. The application in question would determine what bandwidth from the PSD is important, as well as other factors that can be constrained by the design.

## 5.2 Future Limited Angle Solutions

Here is a list to date of possible future solutions to the limited angle problem. These solutions may be impractical at this time, but are valid nonetheless. The first four are hardware solutions that will increase the complexity of our current system. We have purposely avoided the increased difficulty, but the required accuracy may warrant the additional effort.

1. hardware–prism faces
2. hardware–water bath



3. hardware—cylindrical core in larger square block of glass
4. hardware—multiple sided detection
5. software—augmented projections

### 5.2.1 Prisms

The thought here is to put prism wedges on the face through which the test beam is being transmitted. Once the angle of incidence is nearly grazing, a prism would be put in contact with the incident surface. It will also be necessary at the exiting surfaces to use prisms when TIR occurs at the interior surfaces.

This method requires prisms to be available that are at least as big as the entrance face of the sample. Putting the prism in contact with the face of the sample would also require index matching fluid. The prism reduces the entrance angle of rays incident on the sample.

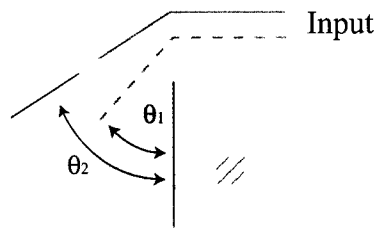


FIGURE 5.1. Extreme portions of the glass are covered by projecting through a prism. Grazing incidence is shown with an entrance angle of  $\theta_1$ , and the decreased angle of  $\theta_2$ .

Using prisms can increase sampling of the glass at its extreme locations. The corners of the cube, for example, may not be sampled at all using just the faces of the cube. Detectors at multiple faces will be required.

### 5.2.2 Water Bath

Here we could submerge the entire sample in water. This may be just as messy as an oil bath of index matching fluid, but it would be more safe in most cases. The increase in index over air would help increase the available range of angles, but would not eliminate the problem.

This would require a method for rotating the sample while in the bath. Previous work indicates that a change of index of the bath with temperature can be a problem, and would require monitoring.

### 5.2.3 Cylindrical Cavity

Here we core out a cylindrical cavity from a larger block of glass. The sample is then placed into the cavity. Less matching fluid is required, but the problem of rotating the sample is still present. This also provides a known surface for the test beam to enter since the boundary provided by the larger block of glass is stationary.

### 5.2.4 Multi-sided Detection

One way to deal with transmission through multiple faces of the sample is to put a detector at each face. The bookkeeping will be a significant issue. A single projection would be built up from information detected on multiple sides. This would require the registration of the two images to be correct. Processing may be needed to interpolate a wavefront from each detector, and then correctly register the information to build a projection. Figure 5.2 shows how the information may be split between two detectors.

### 5.2.5 Augmented Projections

We can also guess the values of blank projections based on what we have for the measured projections. Extrapolation based on the known projections will help, but

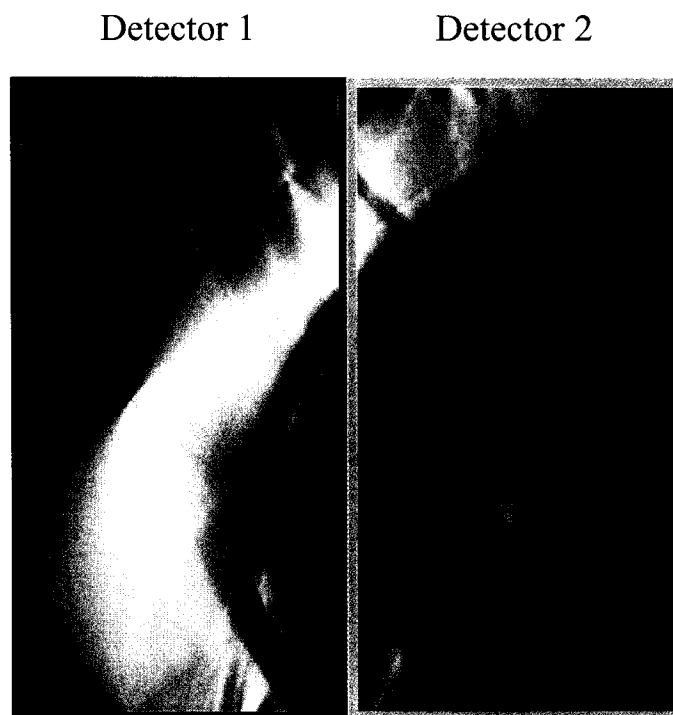


FIGURE 5.2. Sinogram built from two detector information.

the extrapolation distance is typically two or three pixels. The small distance over which extrapolation remains accurate is insufficient even for moderate limited angle problems.

Just filling the blank projections with the average value of the image should improve the reconstruction. The more intelligent the guess, the better the reconstruction should be.

## Appendix A

# PROGRAMS

The Integrated Development Language (IDL)<sup>1</sup> was used to develop our tomographic techniques. All post processing of data, tomographic algorithms, and graphical renderings were done in IDL as well.

### A.1 main.pro

```

;*****
; chd_main2                                Brian Stamper
;*****
; Date: 4/9/04    Modified: 6/10/04    IDL 6.0
;*****
; added sub 1 degree resolution in the proj. matrix
;*****

timein = systemtime(1)

;--Color Table Issues device, decompose=0

;--Variables n_data = 256L
;--width of our data sets--can also use long(256)
w = 360          ;--number of steps--projections
ang = 180.0/w    ;--angle between projections to cover 180 degrees
lim_ang = 180    ;--total range of angles wanted
b = 96.0         ;--width of Gaussian filter
sparse = 1       ;--courseness of projections used
source = 1       ;--when 1, we make a new object, otherwise we load a $
previously saved object

;--define new width so that our data is inscribed inside a circle that
;--is inscribed in the array inside the matrix to be rotated; this
;--will avoid clipping; round up in determining size
n = 3L * n_data

;--Define new arrays to house values
object2 = complexarr(n,n)
object = fltarr(n,n)

```

---

<sup>1</sup>Research Systems Inc.

```

Input2 = fltarr(n,n)
projection_rl = complexarr(n,w+1)
;--The projection array is rectilinear in rho & theta
projection_rl2= complexarr(n,w+1)
proj_vert = complexarr(n/2, (w+1)*2)
InputFT = complexarr(n,w)
;--Fourier Transform of the input array
InputFT_filtered = complexarr(n,w)
reconstruct = complexarr(n,n)

if source eq 1 then begin
  ;--Call routine to define test object
  object_pro, object, n_data, n, 7
  ;--object: 1=random, 2=constant; 3=block; 4=gradient; 5=pic
  ;--          6=block centered; 7=smoothed random
  SAVE, FILENAME = 'object768_zeromean_d.sav', object;, n_data, n
endif else begin
  RESTORE, 'object768b.sav'
endif

;offset = (n/2)-(n_data/2)
;object = object-mean(object[offset:(n-offset-1), offset:(n-offset-1)])
;object = object - mean(object)

;--Now sample the data into a rectilinear coordinate system by rotating then
;--putting the x axis into the 'projection' array with theta=0 at y=0
;--(bottom row) Must do the rotation from the original each time, or it will
;--get increasingly blurry after multiple rotations
for k=0, w do begin
  Input2 = ROT(object, k*ang, cubic = -0.5, missing=0)
  projection_rl[* , k] = complex(0.0, total(Input2, 2)) / 768 ;/256
endfor

;--Limit the projection angle
for t=0, w do begin
  proj_angle = ang * t
  ;--single axis
  if proj_angle gt (90+(lim_ang/2)) or proj_angle lt (90-(lim_ang/2)) then begin
    ;--double axis
    ;if ((proj_angle gt (90+(lim_ang/2))) and (proj_angle lt $
    (180-(lim_ang/2)))) or ((proj_angle lt (90-(lim_ang/2))) and $
    (proj_angle gt (0+(lim_ang/2)))) then begin
      projection_rl[* ,t] = 0.0
    endif
  endif
endfor

;--Make the projection array more course if sparse ne 1 then begin
proj_angle2 = indgen(w+1)
zeros = where((proj_angle2 mod sparse) ne 0)

```

```

    projection_rl[*, zeros] = 0.0
endif

;--Now take 1D FFT along r to prepare for reconstruction
InputFT_r = shift(FFT(shift(projection_rl, n/2, 0), dimension=1), n/2, 0)

;--To reconstruct the Object work back from the Image into the system
;--Input variables to this procedure: n, anglestep, num_proj, I3
;--Output: Image
center=n/2 for i=0, n-1 do begin
    for k=0, n-1 do begin
;for i=172, 192 do begin ;
        for k=172, 192 do begin
            rho=sqrt((i-center)^2 + (k-center)^2)
            ;--ATAN is valid from -pi to pi if given two arguments
            theta=atan(k-center, i-center) * (180.0/!DPI)
            rho_prime=round(rho)
;            rho_prime=rho
                if rho_prime ge center then rho_prime = center - 1
                ;--First, let's get rid of negative angles
                ;--since our projections start at zero and only go up
                if (theta lt 0) then theta = theta + 360.0

                ;--there is no 180 projection, but we can get that data from the low
                ;--half of the zero projection shouldn't need this now that we are
                ;--keeping the 180th projection
                ;if theta eq 180 then theta = 0

                ;--Now we have to do the nearest neighbor by pulling the closest
                ;--number from the projection matrix. In the projection matrix,
                ;--layers are different angles.

                ;--Attempt to catch theta>179, but <180
;            if (theta gt 179.0) and (theta lt 180.0) then begin
                ;??? how best to deal with theta 179.???

                projnumraw = theta/ang
                projnum1 = floor(projnumraw)
                projnum2 = ceil(projnumraw)
                ;--Better if used subroutine here to reconstruct with
                ;--theta=0 rather than 180
;            if (projnum2 eq w) then projnumraw = projnumraw - 1 ;;
                projnum1 = floor(projnumraw)
;            if (projnum2 eq w) then projnumraw = projnumraw - 1 ;;
                projnum2 = ceil(projnumraw)
                if (theta gt 180.0) then begin
                    projnumraw = round((theta - 180.0)/ang)
                    projnum1 = floor(projnumraw)
                    projnum2 = ceil(projnumraw)
                end
            end
        end
    end
end

```

```

;;          if projnum2 eq w then projnumber = projnumber - 1

          ;--Now loop through the third dimension along the rotation axis
;          for j=0, n-1 do begin
;              Image[j, i, k] = I3[j, (center - rhoprime), projnumber]

          ;--We want to make a weighted linear interpolation.  To start we
          ;--will calculate the weighting coefficients.
          if (projnum1 eq projnum2) then begin
              reconstruct[i,k] = InputFT_r[(center - rhoprime), projnum1]
          endif else begin
              weight1 = ((projnum2-projnum1)-abs(projnum2-abs(theta/ang)))/$
              (projnum2-projnum1)
              weight2 = ((projnum2-projnum1)-abs(projnum1-abs(theta/ang)))/$
              (projnum2-projnum1)
              reconstruct[i, k] = weight1*InputFT_r[(center - rhoprime), $
              projnum1] + weight2*InputFT_r[(center - rhoprime), projnum2]
          ;reconstruct[i, k] = 1.0
          endelse

;          endfor
          endif else begin
;          for j=0, n-1 do begin
;              ;--Here we write the point across from the center
;              Image[j, i, k] = I3[j, (center + rhoprime), projnumber]
;              if (projnum1 eq projnum2) then begin
;                  reconstruct[i,k] = InputFT_r[(center + rhoprime), projnum1]
;              endif else begin
;                  weight1 = ((projnum2-projnum1)-abs(projnum2-abs(theta/ang)))/$
;                  (projnum2-projnum1)
;                  weight2 = ((projnum2-projnum1)-abs(projnum1-abs(theta/ang)))/$
;                  (projnum2-projnum1)
;                  reconstruct[i, k] = weight1*InputFT_r[(center + rhoprime), $
;                  projnum1] + weight2*InputFT_r[(center + rhoprime), projnum2]
;              ; reconstruct[i, k] = 1.0
;              endelse

;          endfor
          endelse

          endfor
      endfor

      ;mask_pro, mask, maskfiller, n, 364, n_data
      ;reconstruct = reconstruct * HANNING(n, n, ALPHA=0.5) ;mask
      Output = shift(FFT(shift(reconstruct, n/2, n/2), /inverse), n/2, n/2)

      ;--Separate routine to run comparison based on standard deviation

```

```

stdcompare_pro, object, output, n_data, n, imag_modified

;--Separate routine to run display windows
display_pro, object, output, reconstruct, imag_modified, projection_rl, $
InputFT_r, InputFT_r2, n, w, InputFT_filtered
;display_pro, object, output, reconstruct, projection_rl, InputFT_r, $
InputFT_r2, n, w, InputFT_filtered

;print, 'Sparse multiplier ', sparse
;print, 'RMS difference is', rms

;--Correlation
;--The scale of objconj and reconstruct are verydifferent.
object2 = dcomplex(0.0, object)
;;objectfft = shift(FFT(shift(object2, n/2, n/2)), n/2, n/2)
;;objconj = conj(objectfft)

;;corr = objconj * reconstruct
;;outcorr = (shift(FFT(shift(corr, n/2, n/2), /inverse), n/2, n/2)) * $
768.0^2 / (total(object2 * conj(output)))

;--Now do autocorrelation
;;corr2 = objconj * objectfft
;;outcorr2 = (shift(FFT(shift(corr2, n/2, n/2), /inverse), n/2, n/2)) * $
768.0^2 / (total(object2 * conj(object2)))

;--Cross Correlation
out1 = total(conj(object2) * output) / (sqrt(total(object2^2)) * $
sqrt(total(conj(output)^2)))
print, 'Correlation Value ', max(out1)

;;print, 'FFT Correl. Value ', max(outcorr)

out6 = (max(out1) * imaginary(object2)) - imaginary(output)
window, 31, xsize=768, ysize=768 tvscl, out6

;--standard deviation of each component: object, output, difference
print, sqrt(total((object - mean(object))^2) / (768.0^2 - 1.0))
print, sqrt(total((imaginary(output) - mean(imaginary(output)))^2) / $
(768.0^2 - 1.0))
print, sqrt(total((out6 - mean(out6))^2) / (768.0^2 - 1.0))

timetotal=system(1)-timein
print, 'Time to calculate ', timetotal

END

```



## A.2 object.pro

```

;*****
; PROCEDURE object                      Brian Stamper
;*****
; Date:10/22/03      Modified: 12/8/03      IDL 6.0
;*****
; Routine to create object
;*****

;--object, n_data is the size of the array of interest, n is padded array size,
PRO object_pro, object, n_data, n, object_choose

cmparr = fltarr(n_data,n_data)
;--Array used to set complex portion of input array
cmparr_full = fltarr(n,n) object = fltarr(n,n)

;--Make the object pure phase by designating an array for the complex portion
;--object_choose determines which type of object
if (object_choose eq 1) then begin
  seed=SYSTIME(/SECONDS)
  for i=0, n_data-1 do begin
    for j=0, n_data-1 do begin
      RandomNumber=RANDOMU(seed, /DOUBLE)
      cmparr[i,j] = 0.5 + (RandomNumber * 0.1)
    endfor
  endfor
endif

;--Constant object
if (object_choose eq 2) then begin
  cmparr = 1.00
endif

;--Sine object
if (object_choose eq 8) then begin
  for i = 0, n_data-1 do begin
    cmparr[i,*] = sin(2 * !DPI * i / 64)
  endfor
endif

;--Block object
if (object_choose eq 3) then begin
  cmparr[80:90, 80:90]=1.0
endif

;--Gradient object if (object_choose eq 4) then begin

```

```

    cmparr = dindgen(n_data,n_data)                ;gradient object
    ;--Scale to 1.0 so that the large numbers don't cause integer problems
    ;--in my analysis
    cmparr = cmparr / max(cmparr)
endif

;--Picture object (2D picture)
if (object_choose eq 5) then begin
    cmparr = READ_BMP ('C:\Research\IDL files\Bulk\Project3D\2dtest\flowers.bmp')
endif

;--Centered block object
if (object_choose eq 6) then begin
    cmparr[122:134, 122:134]=1.0
endif

;--smooth random object
if (object_choose eq 7) then begin
    seed=SYSTIME(/SECONDS)
    for i=0, n_data-1 do begin
        for j=0, n_data-1 do begin
            RandomNumber=RANDOMU(seed, /DOUBLE)
            cmparr[i,j] = 0.0 + (RandomNumber * 01.0)
        endfor
    endfor
    ;--now smooth the array
    c = 20.0
    obj_filter = fltarr(n_data,n_data)
    for m=0, n_data-1 do begin
        for p=0, n_data-1 do begin
            obj_filter[m, p] = EXP(-!DPI*(((m-n_data/2)/c)^2 + $
                ((p-n_data/2)/c)^2))
        endfor
    endfor
    obj_fft= shift(FFT(shift(cmparr, n_data/2, n_data/2)), n_data/2, $
        n_data/2)*obj_filter
    cmparr = shift(FFT(shift(obj_fft, n_data/2, n_data/2), /inverse), $
        n_data/2, n_data/2)
endif

;--subtract mean to make a higher contrast reconstruction
;cmparr = cmparr - mean(cmparr)

;--pad input so that it fits inscribed circle of rotating matrix
offset = (n/2)-(n_data/2)
cmparr_full[offset:(n-offset-1), offset:(n-offset-1)]=cmparr
object = cmparr_full

```

END

### A.3 stdcompare.pro

```

;*****
;PROCEDURE stdcompare          Brian Stamper
;*****
; Date: 2/17/04      Modified: 4/12/04      IDL 6.0
;*****
; Routine to measure quality of reconstruction
;*****

;--object, n_data is the size of the array of interest, n is padded array size,
PRO stdcompare_pro, object, output, n_data, n, imag_modified

;--When using a circular mask, need to make sure that the output is masked as well
;roi = n_data
;mask = fltarr(n,n)
;maskfiller=fltarr(n,n)
;mask_pro, mask, maskfiller, n, 256, n_data
;output = output * mask

;--Better metric than pure rms of difference array
;--Analyze just the central section
pad_step = round(n-n_data)/2  imag_small = fltarr(n_data, n_data)
;obj_small = hist_equal(object[pad_step:n-1-pad_step, pad_step:n-1-pad_step])
obj_small = object[pad_step-1:n-1-pad_step, pad_step-1:n-1-pad_step]
;obj_small = object[54:309, 54:309]
imag_small= sqrt(((imaginary(output[pad_step-1:n-1-pad_step, $
pad_step-1:n-1-pad_step])/n_data)^2 + (real_part(output[pad_step-1:
n-1-pad_step, pad_step-1:n-1-pad_step])/n_data)^2)
; ; ; ; imag_small=imaginary(output[pad_step-1:n-1-pad_step, $
pad_step-1:n-1-pad_step])/n_data;^2
;--Find the mean of each array (reconstruction and original)
xbar_obj_small = mean(obj_small)
xbar_imag_small= mean(imag_small)

;--Set both means to zero
obj_meanzero = obj_small - xbar_obj_small  imag_meanzero = imag_small - $
xbar_imag_small
;--Now find Standard deviations of both and calculate ratio--first find variances
objvarsum = 0.0
imagvarsum = 0.0
for i=0, n_data-1 do begin
  for j=0, n_data-1 do begin
    objvarsum_temp = ((obj_meanzero[i,j] - xbar_obj_small)^2)
    objvarsum = objvarsum + objvarsum_temp

    imagvarsum_temp = ((imag_meanzero[i,j] - xbar_imag_small)^2)
    imagvarsum = imagvarsum + imagvarsum_temp
  
```

```
        endfor
    endfor
    std_ratio = sqrt(objvarsum / (n_data^2 - 1)) / sqrt(imagvarsum / $
(n_data^2 - 1)) imag_modified = (imag_meanzero * std_ratio) + xbar_obj_small

;window, 23, xsize=400, ysize=400, title = 'imag_mod'
;tvsc1, imag_modified

;--Now subtract the zero mean arrays and calculate the rms
diff=fltarr(n, n) rms = 0.0
;pad_step = round(n-n_data)/2
;diff = obj_small - (imag_modified * std_ratio)
diff = obj_meanzero - (imag_meanzero * std_ratio)
xbar = total(total(diff, 1))/n^2
for p=0, n_data-1 do begin
    for q=0, n_data-1 do begin
        rmstemp = sqrt((diff[p,q] - xbar)^2/n^2)
        rms = rms + rmstemp
    endfor
endfor

print, 'RMS is ', rms

END
```

## A.4 mask.pro

```

;*****
;PROCEDURE mask                                Brian Stamper
;*****
; Date: 6/23/03      Modified: 4/20/04      IDL 6.0
;*****
; Make a circular mask to apply to data when needed
;*****

;--n is the size of the array, and roi is the size of the region of interest
PRO mask_pro, mask, maskfiller, n, roi, n_data

;--Define arrays
mask_radius = fltarr(n, n) mask_region = fltarr(n, n) reg_int = roi

;--First, define mask_radius using inscribed circle (based on n)
for i = 0, n-1 do begin
  for j = 0, n-1 do begin
    mask_radius[i,j] = sqrt((i-n/2.0)^2 + (j-n/2.0)^2)
  endfor
endfor

;--Mask to limit data to inscribed circle
mask_pass = where(mask_radius le (roi/2 - 0.0))
mask_block = where(mask_radius gt (roi/2 - 1.0))
mask(mask_pass) = 1.0 mask(mask_block) = 0.0

;--Mask to simulate Region of Interest
;mask2_pass = where(mask_radius le (reg_int/2 - 0.0))
;mask2_block = where(mask_radius gt (reg_int/2 - 1.0))
;mask_region(mask2_pass) = 1.0
;mask_region(mask2_block) = 0.0

;mask = mask_region
maskfiller(mask_pass) = 0.0
maskfiller(mask_block) = 0.5

END

```

## A.5 display.pro

```

;*****
; PROCEDURE display                                Brian Stamper
;*****
; Date: 10/22/03      Modified: 10/22/03      IDL 6.0
;*****
; Routine to make display windows
;*****

;--object, n_data is the size of the array of interest, n is padded array size,
PRO display_pro, object, output, reconstruct, imag_modified, projection_rl, $
InputFT_r, InputFT_r2, n, w, InputFT_filtered

;--Show Random phase
window, 0, xsize=n, ysize=n, xpos = 0, ypos = 0, title = 'Object'
tvsc1, object

;--Plot Function with random phase (Imaginary part of Phase1)
window,1, xsize=n, ysize=w, title = 'rectilinear/polar sampling'
tvsc1, Imaginary(projection_rl)

window,4, xsize=n, ysize=n, title = 'Im: image'
;shade_surf, imaginary(Output)
tvsc1, imaginary(output)

window,5, xsize=n, ysize=n, title = 'Amplitude'
tvsc1, sqrt((real_part(Output)^2 + Imaginary(Output)^2))

window,8, xsize=n/1.4142, ysize=n/1.4142, title = 'Imag_modified'
tvsc1, imag_modified

END

```

## A.6 createindex.pro

```

;*****
; Bulk 1 3D                                Brian Stamper
;*****
; Date: 11/6/02      mod: 12/11/02      IDL 5.6
;*****
; Simulate a cube of glass and its statistics in 3D
; Treats random fluctuation as index directly
;*****

```

```

;--Procedure Heading
;PRO CreateIndex

timein=systemtime(1)

;--Color Table Issues
device, decompose=0

;--Variables--must make n=100.0 NOT 100 or when I use n/2 later,
;it automatically converts n/2 to an integer i=0 h=0 n=50.0 w=12
;number of projections

;--Define new array to house values Ind1 = fltarr(n, n, n)
TransferFunction = fltarr(n, n, n) I1 = fltarr(n, n) I2 =
fltarr(n, n, w) plane = fltarr(n, n) proj_matrix = fltarr(n, n, w)

;--Create random data set from a uniform distribution
;--The seed is based on the system time so that it changes each time
;the program is ran
seed=SYSTIME(/SECONDS) for i=0,n-1 do begin
  for j=0,n-1 do begin
    for k=0,n-1 do begin
      RandomNumber=RANDOMU(seed, /DOUBLE)
      Ind1[i,j,k]=RandomNumber
    endfor
  endfor
endfor

;--Calculate FFT of index fluctuations
Index_fft=Shift(FFT(Ind1), n/2, n/2, n/2)

;--Now make a transfer function (filter) to smooth the index
;--We use a 3D Gaussian for now xnot=n/2
;Can use n/2 here if n=100.0
earlier b=8 d=12 g=30 for i=0,n-1 do begin
  for j=0,n-1 do begin
    for k=0,n-1 do begin
      TransferFunction[i,j,k]=Exp(-!DPI*((i-xnot)/b)^2)*Exp(-!DPI*$
      ((j-xnot)/d)^2)*Exp(-!DPI*((k-xnot)/g)^2)
    endfor
  endfor
endfor

;--Calculate FFT of Transfer Function
;--No longer need this since we apply the TF directly in Fourier space
;G=Shift(FFT(TransferFunction), 50, 50)

;--Multiply FFT_index * FFT_TransferFunction to get result and

```

```

;--take PSD H=Index_fft*TransferFunction

;--Kill DC spike in result to keep it from dominating the inverse FFT
;--I do this by thresholding the entire array H_filter = H for
p=0,n-1 do begin
  for q=0,n-1 do begin
    for r=0,n-1 do begin
      if (H_filter[p,q,r]) gt (0.03) then begin
        H_filter[p,q,r]=0.0
      endif
    endfor
  endfor
endfor

;Hinverse = FFT(H, /inverse) Hinverse = FFT(H_filter, /inverse)

;--Use SAVE & RESTORE to output the final matrix
;--I want the array and n
SAVE, FILENAME = 'test1.sav', Hinverse, n

;--PSD created by making DC spike in the FFT equal to zero
H_psd_filter=abs(H_filter)^2 + imaginary(H_filter)^2

;--Show Random phase
;window,0, xsize=400, ysize=320, TITLE = 'Index'
;shade_surf, Ind1[*,*,n/2], color=0, background=255

window,6, xsize=400, ysize=320, TITLE = 'Inverse of filtered FFT'
shade_surf, Hinverse[*,*,n/2];, color=0, background=255

END

```



## A.7 read.pro

```

;*****
; Read Data                               Brian Stamper
;*****
; Date: 6/9/03                             IDL 5.6
;*****
; Read in Ascii columnar data (x, y, value) from WYKO
; Stretch to compensate for .83 aspect ratio
; Flip vertically since WYKO 6000 gives data upside down
;*****
;--Color Table Issues
device, decompose=0

;--This reads the first part of the WYKO files
; the surface map usually with Tilt removed

Intensity = READ_ASCII('C:\Research\WYKO Data\1jul03\processed\n1deg3.asc', $
DATA_START=13, num_records = 245760, Missing_value=0.0, delimiter=' ')

;--Define new array to house values
I2 = fltarr(1,245760)

;--Copy values from field within anonymous structure that was created during read
I2 = Intensity.Field1[2, *]

;--Convert back to matrix form
I3 = Reform(I2, 480, 512)

;--Transpose the array to make it right side up again
I3 = Transpose(I3)

;--Now I need to take care of the fact that the aspect ratio is 0.83
;--instead of 1.0
I4 = CONGRID(I3, 308, 240, /CENTER, CUBIC=-0.5)
I4 = I4[26:281, *]

;--Now make the orientation correspond to real life. This is a transpose and rotate
I4 = Rotate(I4, 7) ;* (1.0 / max(I4))

;--Show input file
window,0, xsize=256, ysize=240, TITLE = 'Data'
tvsc1, I3
window,1, xsize=256, ysize=240, TITLE = 'Stretched & Flipped Data'
tvsc1, I4

END

```

## A.8 reproject.pro

```

;*****
; Read Data                               Brian Stamper
;*****
; Date: 7/8/04                             IDL 6.0
;*****
; Program to take projection of a reconstructed
; three dimensional data set
;*****

Restore, 'test1_output.sav'

data = output
ang = 10
n=150
rotated = fltarr(n,n,n)

;--make filter c = 7.0
obj_filter = fltarr(n,n,n)
for m=0, n-1 do begin
  for p=0, n-1 do begin
    for q=0, n-1 do begin
      obj_filter[m, p, q] = EXP(-!DPI*(((m-n/2)/c)^2 + ((p-n/2)/c)^2 $
        + ((q-n/2)/c)^2))
    endfor
  endfor
endfor

data_ft = shift(fft(shift(data, 75, 75, 75)), 75, 75, 75)

;--filter data
data_ft_filt = data_ft * obj_filter
;--inverse
data_out = shift(fft(shift(data_ft_filt, 75, 75, 75), /inverse), $
75, 75, 75)

;--rotate array before taking new projection
for j=0, n-1 do begin
  out_rot = rot(reform(imaginary(data_out[j, *, *]), n, n), ang, $
  cubic = -0.5, missing=0)
  rotated[j, *, *] = out_rot
endfor

;--now make new projection
proj = total(rotated[*,*,*], 3)

```

```
;;--now filter to smooth it
;;proj_ft = shift(fft(shift(proj, 75,75)), 75, 75)
;;--multiply by previously calculated FFT of filter
;;out_ft = proj_ft * obj_filter ;--take inverse
;;out_proj_filt = shift(fft(shift(out_ft, 75, 75), /inverse), 75, 75)

;;--to view
window, 1, xsize=150, ysize=150
tvsc1, rebin(imaginary(proj[50:99, 50:99]), 150,150)
window, 2, xsize=150, ysize=150
tvsc1, rebin(real_part(proj[50:99, 50:99]), 150,150)
;tvsc1, rebin(sqrt(real_part(out_proj_filt[50:99, 50:99])^2 + $
imaginary(out_proj_filt[50:99, 50:99])^2), 150,150)
```

END

## Appendix B

# HISTORY OF X-RAY COMPUTED TOMOGRAPHY

Computed tomography got off to a slow start, but has gained ground rapidly in the last three decades. Table B.1 gives the key chronology of computed tomography. This information is largely based on an overview given by Kalender in 1993 [70].

TABLE B.1. Chronological overview of x-ray computed tomography.

<b>1895</b>	W. C. Röntgen discovers a new type of rays, which he names 'x-Strahlen.' these rays today are called x-rays or Röntgen rays, in honor of its discoverer.
<b>1917</b>	J. H. Radon develops the mathematical basis for calculation of planar images of a distribution from measurements of its line integrals [18].
<b>1963</b>	A. M. Cormack describes a procedure for the calculation of the distribution of x-ray absorption in the human body [19].
<b>1972</b>	G. N. Hounsfield and J. Ambrose carry out the first clinical trials with x-ray computed tomography
<b>1973</b>	First publications on x-ray computed tomography [21, 71]
<b>1974</b>	60 clinical installations of head CT scanners
<b>1975</b>	First whole body computer tomograph in clinical use [72]
<b>1979</b>	Award for Medical Nobel Prize to Hounsfield and Cormack
<b>1983</b>	First clinical trials with an electron beam CT [73]
<b>1989</b>	First clinical trials with a spiral CT [74]
<b>1993</b>	Approximately 23,000 clinical CT installations
<b>2004</b>	GE unveils first clinical Volume Computed Tomography System.

## Appendix C

# HARDWARE COMMANDS

Below are the necessary simple commands for operating the rotation stage. They use the Compumotor *Motion Planner* software included with the Parker rotation stages.

### C.1 Sample Commands

Just to give a quick example of the commands needed to move the stage, an example is given below.

```
> A 10,10  
> V 3,3  
> D 1000,1000  
> GO 11
```

The controller receives the commands from the computer and then applies the appropriate signals to the motors. The appropriate step size must be downloaded to the controller, which for our setup is 25,000 steps per revolution. The rotation stages have a gear reduction of 90:1, and must be taken into account to calculate the rotation angle. Proper electrical current must also be applied to the motors, and our system uses 1.51 Amps as required by the motor model and having the windings wired in series.

The commands for acceleration, velocity, and distance have the motor location after the command. For example, D a,b would be the distance, in motor steps, to move motor a, and motor b.

- A is acceleration measured in revolutions per second per second.

- V is velocity in revolutions per second.
- D is the distance in motor steps to move.
- GO is the command that tells the controller to execute the previous commands.  
In the above example, both motors would actuate. The GO command does not have the comma separating the motor identifications so we have GO ab.

## REFERENCES

- [1] T. S. Izumitani, *Optical Glass*, American Institute of Physics, 1986.
- [2] M. Cable, "Principles of glass melting," in *Processing I*, D. R. Uhlmann and N. J. Kreidl, eds., *Glass: Science and Technology* **2**, ch. 1, pp. 1–44, Academic Press, Inc., 1984.
- [3] R. Brückner, "Properties and structure of glasses and melts versus preparation," in *Structure, Microstructure, and Properties*, D. R. Uhlmann and N. J. Kreidl, eds., *Glass: Science and Technology* **4A**, ch. 2, p. 84, Academic Press, Inc., 1990.
- [4] H. Eisermann, U. Lange, H. Loch, and G. Weidmann, *Homogenizing and Conditioning*, ch. 3, pp. 165–192. Springer, 2002. Section 3.1 The Intensity of Mixing Processes.
- [5] J. M. Bennett and L. Mattsson, *Introduction to Surface Roughness and Scattering*, Optical Society of America, second ed., 1999.
- [6] A. Duparré, J. Ferre-Borrull, S. Gliech, G. Notni, J. Steinert, and J. M. Bennett, "Surface characterization techniques for determining the root-mean-square roughness and power spectral densities of optical components," *Applied Optics* **41**, pp. 154–171, 2002.
- [7] C. J. Walsh, A. J. Leistner, and B. F. Oreb, "Power spectral density analysis of optical substrates for gravitational-wave interferometry," *Applied Optics* **38**, pp. 4790–4801, 1999.
- [8] J. K. Lawson, C. R. Wolfe, K. R. Manes, J. B. Trenholme, D. M. Aikens, and R. E. English, Jr., "Specification of optical components using the power spectral density function," *Proc. SPIE* **2536**, pp. 38–50, 1995.
- [9] R. E. Parks, "Optical specifications and tolerances for large optics," *Proc. SPIE* **406**, pp. 98–105, 1983.
- [10] J. W. Goodman, *Statistical Optics*, John Wiley & Sons, Inc, 1985.
- [11] C. Wagner, W. Kaiser, J. Mulken, and D. G. Flagello, "Advanced technology for extending optical lithography," in *The 25th Annual International Symposium on Microlithography*, *Proc. SPIE* , 2000.
- [12] H. H. Barrett and W. Swindell, *Radiological Imaging*, Academic Press, 1981.
- [13] R. A. Robb, *Three Dimensional Biomedical Imaging: Principles and Practice*, VCH Publishers, Inc., first ed., 1995.



- [14] B. G. Ziedes des Plantes, "Eine neue method zur differenzierung in der roentgenographie," *Acta Radiologica* **13**, pp. 182–192, 1932.
- [15] D. G. Grant, "Tomosynthesis a three-dimensional imaging technique," *IEEE Trans. on Biomedical Engineering* **BME-19**(1), pp. 20–28, 1972.
- [16] G. Lauritsch and W. H. Hlärer, "A theoretical framework for filtered backprojection in tomosynthesis," *Proc. SPIE* **3338**, pp. 1127–1137, 1998.
- [17] P. Edholm, G. Granlund, H. Knutsson, and C. Petersson, "Ectomography a new radiographic method for reproducing a selected slice of varying thickness," *Acta Radiologica* **21**, pp. 433–442, 1980.
- [18] J. Radon, "Über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten," *Berichte über die Verhandlungen der Königlich Sächsischen Gesellschaft der Wissenschaften zu Leipzig* **69**, pp. 262–277, 1917.
- [19] A. M. Cormack, "Representation of a function by its line integrals, with some radiological applications," *Journal of Applied Physics* **34**, pp. 2722–2727, 1963.
- [20] A. M. Cormack, "Representation of a function by its line integrals, with some radiological applications. II," *Journal of Applied Physics* **35**, pp. 2908–2913, 1964.
- [21] G. N. Hounsfield, "Computerized transverse axial scanning (tomography): Part I description of system," *British Journal of Radiology* **46**, pp. 1016–1022, 1973.
- [22] S. Cha and C. M. Vest, "Tomographic reconstruction of strongly refracting fields and its application to interferometric measurement of boundary layers," *Applied Optics* **20**, pp. 2787–2794, 1981.
- [23] R. N. Bracewell, *Image Reconstruction in Radio Astronomy*, ch. 3, p. 84. Springer-Verlag, 1995.
- [24] J. Hsieh, *Image Reconstruction*, ch. 3. SPIE Press, 2003. Section 3.6 The Filtered Backprojection Algorithm.
- [25] H. H. Barrett and K. J. Myers, *Foundations of Image Science*, Wiley Series in Pure and Applied Optics, John Wiley & Sons, Inc., first ed., 2004.
- [26] P. R. Smith, T. M. Peters, and R. H. T. Bates, "Image reconstruction from finite numbers of projections," *J. Phys. A: Math., Nucl. Gen.* **6**, pp. 361–382, 1973.
- [27] R. H. T. Bates and T. M. Peters *N. Z. J. Sci.* **NS-21**, pp. 883–896, 1971.

- [28] E. W. Weisstein, "Singular value decomposition," *MathWorld—A Wolfram Web Resource*. <http://mathworld.wolfram.com/SingularValueDecomposition.html>.
- [29] T. F. Budinger and G. T. Gullberg, *Emission Computed Tomography*, ch. 5, p. 201. Springer-Verlag, 1995.
- [30] H. H. Barrett and K. J. Myers, *Iterative Methods*, ch. 15.4. Wiley Series in Pure and Applied Optics, John Wiley & Sons, Inc., first ed., 2004.
- [31] Y. Censor and S. Zenios, *Parallel Optimization*, Oxford University Press, 1997.
- [32] R. Gordon, R. Bender, and G. Herman, "Algebra reconstruction techniques (art) for three-dimensional electron microscopy and x-ray photography," *J. Theor. Biol.* **39**, pp. 471–181, 1970.
- [33] R. Gordon, "A tutorial on ART (algebraic reconstruction techniques)," *IEEE Trans. NS-21*, pp. 78–93, 1974.
- [34] M. J. Lahart, "Local image restoration by a least-squares method," *J. Opt. Soc. Am.* **69**, pp. 1333–1339, 1979.
- [35] K. T. Smith, "Reconstruction formulas in computed tomography," in *Computed Tomography, Proc. of Symposia in Applied Mathematics* **27**, pp. 7–23, 1982.
- [36] P. Gilbert, "Iterative methods for the three-dimensional reconstruction of an object from projections," *Journal of Theoretical Biology* **36**, pp. 105–117, 1972.
- [37] A. B. Goncharov, *Three-Dimensional Reconstruction of Arbitrarily Arranged Identical Particles Given Their Projections*. American Mathematical Society, first ed., 1990.
- [38] S. Lanzavecchia, P. L. Bellon, and M. Radermacher, "Fast and accurate three-dimensional reconstruction from projections with random orientations via Radon transforms," *Journal of Structural Biology* **128**, pp. 152–164, 1999.
- [39] R. N. Bracewell, *Two-Dimensional Imaging*, Prentice Hall, 1995.
- [40] J. I. Jackson, C. H. Meyer, D. G. Nishimura, and A. Macovski, "Selection of a convolution function for fourier inversion using gridding," *IEEE Trans. Medical Imaging* **10**, pp. 473–478, 1991.
- [41] E. Payot, F. Prêteaux, Y. Troussset, and R. Guillemaud, "3D reconstruction from incomplete fourier spectra: An extrapolation approach," in *Statistical and Stochastic Methods for Image Processing, Proc. SPIE* **2823**, pp. 160–173, 1996.

- [42] M. Y. Chiu, H. H. Barrett, R. G. Simpson, C. Chou, J. W. Arendi, and G. R. Gindi, "Three-dimensional radiographic imaging with a restricted view angle," *JOSA* **69**, pp. 1323–1333, 1979.
- [43] T. Sato, S. J. Norton, M. Linzer, O. Ikeda, and M. Hirama, "Tomographic image reconstruction from limited projections using iterative revisions in image and transform spaces," *Applied Optics* **20**, pp. 395–399, 1980.
- [44] K.-C. Tam and V. Perez-Mendez, "Tomographical imaging with limited-angle input," *J. Opt. Soc. Am.* **71**, pp. 582–592, 1981.
- [45] A. M. Darling, T. J. Hall, and M. A. Fiddy, "Stable, noniterative object reconstruction from incomplete data using *a priori* knowledge," *J. Opt. Soc. Am.* **73**, pp. 1466–1469, 1983.
- [46] A. Papoulis, "A new algorithm in spectral analysis and band-limited extrapolation," *IEEE Trans. circuits Syst.* **CAS-22**, pp. 735–742, 1975.
- [47] K.-C. Tam and V. Perez-Mendez, "Limited-angle three-dimensional reconstructions using Fourier transform iterations and Radon transform iterations," *Optical Engineering* **20**, pp. 586–589, 1981.
- [48] R. M. Lewitt, "Processing of incomplete measurement data in computed tomography," *Medical Physics* **6**, pp. 412–417, 1979.
- [49] R. N. Bracewell and S. J. Wernecke, "Image reconstruction over a finite field of view," *JOSA* **65**, pp. 1342–1346, 1975.
- [50] W. Górski and M. Kujawińska, "Three-dimensional reconstruction of refractive index inhomogeneity," *Proc. of 1st Int. Symposium on Process Tomography* , pp. 105–117, 2000.
- [51] W. Górski and M. Kujawińska, "Three-dimensional quantitative reconstruction of asymmetrical refractive index inhomogeneity," in *Photonics in information processing, Proc. 14th School of Optoelectronics* , pp. 59–60, 2000.
- [52] A. M. Zysk, J. J. Reynolds, D. L. Marks, P. S. Marney, and S. A. Boppart, "Projected index tomography," *Optics Letters* **28**(9), pp. 701–703, 2003.
- [53] P. Espinasse, "Combined techniques map materials," *Optical Engineering Magazine* **3**(7), p. 7, 2003.
- [54] R. Haberman, *Elementary Applied Partial Differential Equations*, Prentice-Hall, Inc., second ed., 1987.

- [55] S. Yakowitz and F. Szidarovszky, *An Introduction to Numerical Computations*, Macmillan Publishing Company, second ed., 1989.
- [56] J. W. Goodman, *Introduction to Fourier Optics*, McGraw-Hill, second ed., 1996.
- [57] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, "Fourier and spectral applications," in *Numerical Recipes in C: The art of scientific computing*, ch. 13, pp. 537–558, Cambridge University Press, second ed., 1992.
- [58] J. J. Gray, *The Hilbert Challenge*, Oxford University Press, first ed., 2000.
- [59] J. D. Gaskill, *Linear Systems, Fourier Transforms, and Optics*, John Wiley & Sons, Inc., 1978.
- [60] R. N. Bracewell, *The Fourier Transform and Its Applications*, The McGraw-Hill Companies, Inc, 2000.
- [61] J. E. Greivenkamp and J. H. Bruning, "Phase shifting interferometers," in *Optical Shop Testing*, D. Malacara, ed., ch. 14, John Wiley & Sons, Inc., second ed., 1992.
- [62] W. J. Dallas and W. Mauser, "Preparing pictures for visual comparison," *Applied Optics* **19**, pp. 3586–3587, 1980.
- [63] R. W. Conners and C. A. Harlow, "Equal probability quantizing and texture analysis of radiographic images," *Computer Graphics and Image Processing* **8**, pp. 447–463, 1978.
- [64] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, "Modelling of data," in *Numerical Recipes in C: The art of scientific computing*, ch. 15, pp. 656–706, Cambridge University Press, second ed., 1992.
- [65] H. Suhara, "Interferometric measurement of the refractive-index distribution in plastic lenses by use of computed tomography," *Applied Optics* **41**, pp. 5317–5325, 2002.
- [66] J. M. De Freitas and M. A. Player, "Ultrahigh precision measurements of optical heterogeneity of high quality fused silica," *Applied Physics Letters* **66**, pp. 3552–3554, 1995.
- [67] D. W. Sweeney and C. M. Vest, "Reconstruction of three-dimensional refractive index fields from multidirectional interferometric data," *Applied Optics* **12**, pp. 2649–2664, 1973.
- [68] D. Verhoeven, "Limited-data computed tomography algorithms for the physical sciences," *Applied Optics* **32**, pp. 3736–3754, 1993.

- [69] G. H. Smith, *Practical Computer-aided Lens Design*, William-Bell, Inc., first ed., 1998.
- [70] W. A. Kalender, *X-ray Computed Tomography - State of the Art*, pp. 10–27. SPIE–The International Society for Optical Engineering, 1993.
- [71] J. Ambrose, “Computerized transverse axial scanning (tomography): Part II clinical application,” *British Journal of Radiology* **46**, pp. 1023–1047, 1973.
- [72] R. S. Ledley, G. Di Chiro, A. J. Luessenhop, and H. L. Twigg, “Computerized transaxial x-ray tomography of the human body,” *Science* **186**, pp. 207–212, 1974.
- [73] D. P. Boyd and M. J. Lipton, “Cardiac computed tomography,” *Proc. of IEEE* **71**, p. 298, 1983.
- [74] W. A. Kalender, W. Seifler, E. Klotz, and P. Vock, “Spiral volumetric CT with single-breathhold technique, continuous transport, and continuous scanner rotation,” *Radiology* **176**, pp. 181–183, 1990.