

DEVELOPMENT OF AN ADAPTIVE X-RAY RESEARCH FACILITY AND A
DIRECT SILICON BONDED REFERENCE OBJECT FOR COMPUTED
TOMOGRAPHY

by

Jeffery D. Thousand

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Mechanical Engineering

Charlotte

2019

Approved by:

Dr. Stuart T. Smith

Dr. Naiquan Zheng

Dr. Christopher J. Evans

Dr. Glenn D. Boreman

©2019
Jeffery D. Thousand
ALL RIGHTS RESERVED

ABSTRACT

JEFFERY D. THOUSAND. Development of an Adaptive X-ray Research Facility and a Direct Bonded Silicon Reference Object for Computed Tomography (Under the direction of DR. STUART T. SMITH)

Our ability to harness the power of X-ray radiation has led to many important scientific discoveries. Two common applications for X-rays are crystallography and imaging through objects (tomography). Facility design is a critical component in optimizing a system's performance for an application and insuring operator safety. In the first part of this dissertation, the design, construction, and performance of an adaptive X-ray research facility is discussed, with emphasis placed on development as a future X-ray interferometry system. Performance is evaluated from thermal measurements, measuring characteristics of detectors, measuring characteristics of the X-ray generator, and by demonstrating successful diffraction and reflectivity experiments.

One of the fastest growing fields in metrology is part measurement using industrial X-ray computed tomography (CT). The ability to calibrate the system for accurate measurements to date has relied on physical reference objects. A novel, direct silicon bonded reference object for X-ray CT was developed and is discussed in the second part of the dissertation. This reference object is constructed by bonding four etched silicon pieces from a single wafer. Within each reference object are two sets of patterns grouped into linear and curved features, with features ranging in size from 5 μm to 1200 μm . The standalone silicon object measures $4 \times 4 \times 2 \text{ mm}^3$, or $6 \times 6 \times 4 \text{ mm}^3$ when encapsulated in a silica and epoxy protective shell. X-ray images of the reference object were taken, reconstructed into a 3D model, and features were extracted. A different part, cut from the

same wafer, was measured by coherence scanning interferometry (CSI) to perform a simple comparison study to CT data. Two features were extracted from the CT and CSI data, one with an edge constructed by a cubic polynomial (asymmetric cubic), and the other having a smooth, continuous curved edge (chain-link). CSI measurements were within $0.7\text{ }\mu\text{m}$ of the nominal design, while CT measurements were up to $3.1\text{ }\mu\text{m}$ from nominal. Two algorithms were used to measure the cubic curve, both programmed to minimize the tangent-normal residuals. One algorithm calculated best fit coefficients and the other used the nominal coefficients. The 1σ deviations for the CSI data were with $0.12\text{ }\mu\text{m}$ and $0.14\text{ }\mu\text{m}$ respectively, and for the CT data were $0.32\text{ }\mu\text{m}$ and $0.39\text{ }\mu\text{m}$ respectively. With the chain-link feature, circles were best fit to the outer edges. Similar results were seen with 1σ deviations of $0.13\text{ }\mu\text{m}$ for the CSI data and $0.4\text{ }\mu\text{m}$ for the CT data. Use of a high-resolution X-ray CT system with smaller voxel sizes should improve the CT results.

DEDICATION

This work is dedicated to my parents and grandma.

ACKNOWLEDGMENTS

It has taken a long time to get to this point, and with that, many people have my eternal gratitude. While it is difficult to put most in some sort of order, at the top of the list is my PhD advisor Dr. Stuart Smith. His patience, encouragement, professionalism, and insight are invaluable and admirable. Having the opportunity to be mentored by Dr. Smith and work on the various programs in his lab has been an amazing experience.

I would like to thank Dr. Nigel Zheng for his mentorship. My journey towards a graduate degree began shortly after I started undergraduate research in Dr. Zheng's group. Under his guidance, I learned about the use of motion capture for studying biokinematics and how to develop computer programs that control mechanical systems.

I also want to thank Dr. Glenn Boreman, Dr. Lou Deguzman, Dr. Christopher Evans, Mr. John Hudak, Dr. Robert Hudgins, Dr. Jimmie Miller, Dr. Herminso Villarraga-Gómez, and Mr. Scott Williams. Without them, my journey may have ended before I could see it through to a satisfactory conclusion. Their sharing of knowledge, advice, assistance, and simply taking some time to chat are cherished.

And to my friends, thank you for your support. Whether it came as sincere encouragement or banter, this major milestone in my life has been positively influenced by your involvement.

TABLE OF CONTENTS

LIST OF FIGURES	xii
LIST OF TABLES	xxi
LIST OF ABBREVIATIONS	xxiii
CHAPTER 1: INTRODUCTION	1
1.1 History of X-rays	1
1.2 X-ray Interferometry History and Applications	2
1.3 Reference Objects for X-ray Computed Tomography	7
1.4 Research Objectives and Dissertation Outline	13
1.5 Summary of Research Goals	14
CHAPTER 2: A FACILITY FOR X-RAY METROLOGY	16
2.1 Principle of X-ray Operation	16
2.1.1 Generation	16
2.1.2 Absorption	20
2.1.3 Detection	21
2.1.4 Interferometry	23
2.2 Facility Design	28
2.2.1 X-ray Enclosure	29
2.2.2 X-ray System and Control	32
2.2.2.1 Source Subsystem	32
2.2.2.2 Specimen Subsystem	35
2.2.2.3 Detector Subsystem	38
2.2.2.4 System Controls	39
2.2.3 Water Chiller Acoustic Damping Enclosure	42
2.2.4 Crystal Aligner	43
CHAPTER 3: FACILITY PERFORMANCE STUDY	46
3.1 Thermal Sensors and Stability	46
3.2 Radiation Detectors	49
3.2.1 Commercial Ion Chamber Meter	49

3.2.2 Geiger-Müller Tube	51
3.2.3 Commercial X-ray Camera	56
3.2.3.1.1 Integer Overflow	57
3.2.3.1.2 Image correction	58
3.2.4 Custom X-ray Camera	62
3.2.4.1.1 Digital Camera	65
3.2.4.1.2 Optical Components	69
3.2.4.1.3 Scintillators (Ce:YAG and Ce:LuAG)	71
3.3 X-ray Source Performance	73
3.4 Diffraction Experiment	76
3.5 Reflectivity Measurements	78
CHAPTER 4: DEVELOPING AN X-RAY REFERENCE OBJECT	80
4.1 Principles of Microfabrication	80
4.1.1 Photolithography	80
4.1.2 Silicon Etching	83
4.1.2.1 Etching Study	85
4.1.3 Silicon Bonding	90
4.1.3.1 Bonding Study	93
4.2 Reference Object Design	94
4.2.1 Geometric Features	95
4.2.1.1 Linear Features	97
4.2.1.2 Curved Features	98
4.2.2 Protective Packaging	99
4.3 Reference Object Fabrication	101
4.3.1 Wafer Etching	102
4.3.2 Direct Silicon Bonding	107
4.3.3 Final Assembly	113
CHAPTER 5: PARAMETER EXTRACTION OF MEASURED DATA	116
5.1 Data Collection	116
5.1.1 X-ray CT	116
5.1.2 Coherence Scanning Interferometry	118

5.2 Parameter Extraction Algorithm	118
5.2.1 Data Type Preparation	120
5.2.1.1 CT Image	120
5.2.1.2 CSI or Confocal	120
5.2.2 Feature Detection and Edge Extraction	122
5.2.3 Spatial Measurement	124
5.2.4 Step Height	126
5.3 Reference Object Evaluation and Measurements	126
5.3.1 General Observations	127
5.3.2 Extracted Feature Measurements	130
5.3.2.1 Asymmetric Cubic	131
5.3.2.2 Chain Link	142
CHAPTER 6: CONCLUSIONS	147
6.1 X-ray Facility	147
6.2 CT Reference Object	149
CHAPTER 7: FUTURE WORK	153
7.1 X-ray Facility	153
7.2 CT Reference Object	155
REFERENCES	157
APPENDIX A: SUPPLEMENTARY FIGURES	177
APPENDIX B: INCLINOMETER FOR X-RAY ROTARY TABLE	183
APPENDIX C: REFERENCE OBJECT POLYNOMIAL SIMULATION	185
C.1 Overview	185
C.2 Algorithm	185
C.3 Comparison to Actual Data	189
APPENDIX D: SECONDARY INTERFEROMETER	193
D.1 Equipment	193
D.2 Mounting	193
D.3 Machining	195

D.4 Unmounting and Cleaning	198
D.5 Silicon Etching	198
D.6 Part Drawings	200
D.7 Additional Images	203
APPENDIX E: MATLAB® CODES	204
E.1 Main Measurement Scripts	204
E.1.1 CSI and Confocal: ‘Artefact_Surface_Metrology_v5.m’	204
E.1.2 CT Images: ‘Artefact_Image_Metrology.m’	208
E.1.3 Create Average Image: ‘LoadMultipleImages.m’	211
E.2 Simulation Scripts	212
E.2.1 ‘Run_APS.m’	212
E.2.2 Circle Simulation: ‘ArtefactCircleSim.m’	214
E.3 Functions	218
E.3.1 ‘autolevel.m’	218
E.3.2 ‘ArtefactPolySim.m’	219
E.3.3 ‘circfit.m’	223
E.3.4 ‘DeleteInnerPts.m’	224
E.3.5 ‘distmeas.m’	225
E.3.6 ‘GaussFilter.m’	226
E.3.7 ‘histpeaks.m’	227
E.3.8 ‘maxdist.m’	228
E.3.9 ‘normdist.m’	229
E.3.10 ‘openfile.m’	230
E.3.11 ‘planeft.m’	237
E.3.12 ‘polyminddev.m’	239
E.3.13 ‘ReadMetroProFile.m’	241
E.3.14 ‘rotatedata.m’	246
E.3.15 ‘sumsquare.m’	247
E.4 Shape Scripts	248
E.4.1 ‘asymmetric_rules.m’	248
E.4.2 ‘asymmetric_meas3.m’	251
E.4.3 ‘barbell_rules.m’	254

E.4.4 ‘barbell_meas.m’	257
E.4.5 ‘chain_rules2.m’	259
E.4.6 ‘chain_meas3.m’	262
E.4.7 ‘rectangular_rules.m’	265
E.4.8 ‘rectangular_meas.m’	267
E.4.9 ‘secondorder_rules.m’	268
E.4.10 ‘secondorder_meas3.m’	271
E.4.11 ‘thirdorder_rules.m’	274
E.4.12 ‘thirdorder_meas3.m’	277
E.4.13 ‘triangle_rules.m’	280
E.4.14 ‘triangle_meas.m’	282
E.5 Supporting Scripts	284
E.5.1 ‘shapeorient.m’	284
E.5.2 ‘stepmeas.m’	286

LIST OF FIGURES

Figure 1.1. Examples of X-ray interferometer configurations. From left to right are Laue-case, Bragg-case, and Laue-Bragg-Laue. The incident blade is the splitter, followed by the mirror(s), and analyzer blade.	4
Figure 1.2. Becker and Stümpel's skew-symmetric X-ray interferometer with a hinge at the center.	6
Figure 1.3. Influential factors in CT measurement presented as an Ishikawa diagram.	8
Figure 1.4. CAD drawings of X-ray CT reference objects that have been found in the literature. (a) CT tree with ruby spheres attached to CFRP rods and mounted to a CFRP bar. (b) Ball plate with ruby spheres attached to a CFRP plate. (c) Tetrahedral stack of ruby spheres. (d) PTB calotte cube. (e) PTB hole plate. (f) Cactus standard. (g) Step gauge. (h) Step cylinder. (i) Fiber gauge with glass fiber optic cables of varying heights. (j) Pan flute gauge made from glass cylinders mounted on a carbon fiber support.	11
Figure 1.5. Interface structural resolution reference objects produced by QRM GmbH. (a) Standard model silicon chip size is $5 \times 5 \text{ mm}^2$ and is mounted inside a 20 mm diameter, 40 mm long cylinder filled with air. (b) Nano model silicon chip size is $3 \times 3 \text{ mm}^2$ mounted to a plastic base that is 5.2 mm diameter and 19 mm long. Images from QRM GmbH's website [106].	13
Figure 2.1. Electron from the E-beam collides with a core electron in the K shell, resulting in an electron from a higher energy shell to drop down.	18
Figure 2.2. X-ray spectrum of a molybdenum target at 35 kV. The line widths are not to scale.	19
Figure 2.3. Molybdenum mass attenuation coefficient with labeled absorption edges [109].	19
Figure 2.4. Schematic of a modern X-ray generator.	20
Figure 2.5. Silicon and lead mass attenuation coefficient with the $K\alpha$ characteristic emission marked [111,112].	21
Figure 2.6. An x-ray photon enters the Geiger-Müller tube and creates an ion pair. As the electron accelerates towards the anode wire, more ion pairs form and an avalanche of electrons generates a measurable pulse of voltage.	22
Figure 2.7. Principle of operation for a (a) Mach-Zehnder interferometer compared to an (b) X-ray interferometer.	24

Figure 2.8. A cubic crystal with lattice constant a and examples of planes in (111) and (220) orientations.	25
Figure 2.9. To calculate the Bragg angle, the spacing between the cubic crystal's planes d_{hkl} about one orientation and the wavelength of incident light k need to be known.	26
Figure 2.10. Displacement of the analyzer blade along axis z gives a direct measurement of the crystal plane spacing d_{hkl} . Signals recorded by detector 1 and 2 are opposite.	27
Figure 2.11. When X-ray radiation passes through a medium, the intensity of the signal is attenuated, and the phase is shifted forward. Modified from [116].	27
Figure 2.12. The enclosure is constructed from interlocking steel frames, lead, wood, and leaded acrylic. Painted lead lines the major frame components and the wood. Windows are held in place by expanding adjustable feet. The floor of the system is lined with lead and protected by a plastic film.	30
Figure 2.13. Computer aid drawing of the X-ray enclosure with dimensions in inches.	31
Figure 2.14. Configuration of components for X-ray interferometry studies.	32
Figure 2.15. Components of the source subsystem.	33
Figure 2.16. Configuration of the X-ray generator. When assembled, the vacuum tube (right) slides into the port (left).	34
Figure 2.17. The (a) X-ray generator and its capillary optic for beam focusing. Add-ons are (b) a laser aligner that slides onto the capillary optic and (b) filters that snap onto the optic window.	35
Figure 2.18. Multiple component setups of the specimen subsystem. Left is for interferometry, and the middle and right setups are for reflectivity.	36
Figure 2.19. V-groove kinematic mount for the X-ray interferometer.	37
Figure 2.20. Single crystal X-ray interferometer.	38
Figure 2.21. Components of the camera and GM tube subsystem.	39
Figure 2.22. X-ray facility connection schematic configured for X-ray interferometry.	41
Figure 2.23. Water chiller and sound damping enclosure.	43
Figure 2.24. Crystal aligner design for interferometer manufacturing.	44

Figure 3.1. Temperature sensor circuit diagram.	47
Figure 3.2. Temperature comparison between the enclosure and lab. The enclosure filters the lab's high frequency temperature oscillations.	48
Figure 3.3. Temperature readings within the enclosure showing increased heating around the X-ray source when turned on. The X-ray was turned off after 80 min. Opening the enclosure caused the drops at 100 min and 112 min.	48
Figure 3.4. Fluke Biomedical 451B-RYR ion chamber survey meter.	49
Figure 3.5. Indicated dose by the commercial dose meter to calculate the detector response time constants. X-ray shutter was closed at 463 min to measure the decay time constant and opened at 534 min to measure the rise time constant.	50
Figure 3.6. Simplified Geiger-Müller circuit.	51
Figure 3.7. Effect of different R1 resistances on the voltage spike.	52
Figure 3.8. Effect of the Geiger-Müller circuit's R2 resistance on the peak voltage.	52
Figure 3.9. Effect of the Geiger-Müller circuit's R2 resistance on the decay time constant.	53
Figure 3.10. Geiger-Müller circuit diagram to convert voltage spikes into a TTL signal.	54
Figure 3.11. Graph of the count rate compared to the comparator's threshold voltage. The voltage decreased in subsequent tests as the organic quench of the Victoreen 6306 tube was destroyed.	55
Figure 3.12. Scanning the voltage into the comparator using a CompactRIO analog out to determine the optimum threshold voltage and showing the output consistency of the inorganic tube.	56
Figure 3.13. The branded Photonic Science camera that used a scintillator coated fiber bundled directly bonded to the CCD.	57
Figure 3.14. Integer overflow of the camera causing a ringing effect in the image.	58
Figure 3.15. Fixed-pattern noise in an image of the X-ray beam. Speckling is caused by photo response non-uniformity and the graying of the background is the dark signal non-uniformity.	59
Figure 3.16. Picture of a (a) tuning fork, (b) the raw X-ray image, and (c) the post processing with image corrections. Images from [123].	60

Figure 3.17. (a) Sinogram of the quartz tuning fork raw image at image row 277 and (b) the fixed-pattern noise corrected sonogram. Each column is the row's image as the object is rotated.	61
Figure 3.18. Inverse Radon transform created from the (a) raw sonogram (b) and the fixed-pattern noise corrected sinogram. Image is a cross-sectional view of a quartz tuning fork. Images from [123].	62
Figure 3.19. Assembled in-line style X-ray camera with a custom mount attached to a rotatory stage that is attached to a two-axis motorized YZ stage. In this setup, leaded glass protects the CCD from radiation damage.	63
Figure 3.20. Exploded solid model of the periscope X-ray camera.	64
Figure 3.21. Mean pixel value and the standard deviation of the difference in bias frames with 2 min delay between capture.	66
Figure 3.22. Mean pixel value verses variance. Slope is the camera's gain factor.	67
Figure 3.23. Mean pixel value and the standard deviation of the difference in bias frames with no additional delay between captures.	68
Figure 3.24. Tilt in the bias frame.	69
Figure 3.25. Dark signal non-uniformity removal comparison between a constant and plane. Dark frame exposure time was 50 s.	69
Figure 3.26. Comparison of the custom camera to the Bede Scientific camera.	70
Figure 3.27. Leaded glass attenuation test of the custom camera.	71
Figure 3.28. Intensity comparison between the uncoated 150 μm and the 350 μm scintillator before and after coating with aluminum and silicon dioxide.	73
Figure 3.29. Output of the x-ray source as at different power settings.	74
Figure 3.30. Amplitude of the spot size at different positions for the camera along the beam axis.	75
Figure 3.31. The full-width half-maximum spot size at different positions for the camera along the beam axis.	76
Figure 3.32. X-ray diffractometer data of a spare piece of silicon cut from the interferometer with an intensity peak at approximately 28.4° .	77
Figure 3.33. Successful diffraction from the (111) and (220) planar orientations. Diffracted spot is indicated by the left arrow and transmitted spot is the right arrow.	77

Figure 3.34. Diffraction from a tilted (111) plane. Diffracted spot is indicated by the left arrow and transmitted spot is the right arrow. Angle θ is approximately 19.5° .	78
Figure 3.35. X-ray reflection curve for tungsten carbide. Data provided by Jacob Cole.	79
Figure 4.1 Photolithographic process and photoresist types.	81
Figure 4.2 Isotropic and anisotropic etching.	83
Figure 4.3 Deep reactive-ion etching using the Bosch process where a layer of C_4F_8 protects the side walls during the SF_6 etch.	84
Figure 4.4 Physical disadvantages to performing DRIE. (a) Sidewall undulations from the Bosch switching. (b) Aspect ratio dependent etching. (c) By-products re-depositing on surface.	85
Figure 4.5. The STS Advanced Silicon Etcher used for deep reactive-ion etching.	86
Figure 4.6 (Left) Paper template design with location tracking numbers. (Right) Confocal image of etched rectangular features that were measured.	87
Figure 4.7 Confocal measurements of the step heights for the baseline etch.	88
Figure 4.8 Contact angles for hydrophilic and hydrophobic surfaces.	91
Figure 4.9 Reference object design with a protective shell. Modified from [160].	94
Figure 4.10. Linear features containing only straight lines. Dark areas represent etched sections.	96
Figure 4.11. Curved features containing at least one polynomial edge. Dark areas represent etched sections.	96
Figure 4.12. Epoxy curing clamp assembly. Modified from [160].	101
Figure 4.13. Reference object fabrication process: (a) protective thermal oxide layer is grown; (b) photoresist is spin coated onto wafer; (c) photoresist is exposed and developed; (d) wafer is etched; (e) wafer is diced into square wafers; (f) four square wafers are stripped of oxide, cleaned with 'Piranha' and oxygen plasma, stacked, and clamped; (g) stacked wafers are annealed twice and diced into individual references; (h) glass and epoxy protective shell is built around the silicon reference.	102
Figure 4.14. Brewer Science Cee 1100FX hotplate used for the dehydration, soft, and post-exposure bakes.	104
Figure 4.15. Brewer Science CEE 200 spin coater for adding photoresist.	104

Figure 4.16. Quintel Ultraline Q 4000 for transferring the mask design into the photoresist using ultraviolet light.	105
Figure 4.17. Pressure sensitive dicing saw tape with grip-ring and the Micro Automation 1100 dicing saw for cutting wafers.	107
Figure 4.18. Wafer boat and handle for handling the square silicon wafers.	108
Figure 4.19. STS Advanced Oxide Etcher used for oxygen plasma cleaning prior to bonding.	109
Figure 4.20. V-groove mechanical aligner and the steel clamp.	110
Figure 4.21. Programmable vacuum furnace.	111
Figure 4.22. A bonded stack secured to a carrier wafer and ready for dicing. Modified from [160].	112
Figure 4.23. Reference object carrier made of PTFE used for cleaning fused silica and bonded silicon reference objects prior to final assembly.	113
Figure 4.24. Epoxy clamp halves assembled and the syringes in the filling process. Needle for the 10 mL syringe is not shown.	114
Figure 4.25. UVO-Cleaner Model 42 used to cure the UV epoxy.	115
Figure 5.1. (Left) X-ray source with reference object and (right) ISO50 thresholding example. Thresholding is used to differential between different materials and from the background. In the example, all voxels left of the ISO50 line are background and all voxels right of the ISO50 line are material.	117
Figure 5.2. Flowchart of the feature extraction and measurement algorithm.	119
Figure 5.3. Areas targeted by padding are the (a) edge ringing artifact seen in the laser confocal microscope data and (b) the etched surface curvature discussed later in section 5.2.4 on step height measurement.	121
Figure 5.4. Example of positive feature padding that increases the area with zeros.	122
Figure 5.5. Histogram comparison showing the leveling results. An offset has been added to the pre-level and first level for clarity.	122
Figure 5.6. Orienting the asymmetric cubic shape. (a) Detecting the origin indicated by the star icon and a second corner point. (b) Translate rotate, detect a third corner, and extract two edges. (c) Align the bottom edge to the x-axis and to extract remaining corners and edges.	123

Figure 5.7. Minimization of the deviations when using the (a) nominal line and (b) best fit calculated line. Data is from a CSI measurement of an asymmetric feature.	125
Figure 5.8. SEM images showing the scalloped sidewall profile of the periodic squares. Modified from [160].	127
Figure 5.9. A comparison of the (a) cyanoacrylate and (b) direct bonding between layers of silicon. Modified from [160].	129
Figure 5.10. Silicon surface reconstruction (a) without a protective shell and (b) with the protective shell. Modified from [160].	129
Figure 5.11. An (a) early reference object using microscope glass and (b) a final version with end plates cut from a silica wafer.	130
Figure 5.12. Images of the largest asymmetric feature that shows the contrast difference between the (a) reconstructed X-ray CT data, (b) CSI intensity data, and (c) CSI phase data. CSI data taken with 20× objective. Modified from [160].	131
Figure 5.13. CSI intensity image showing four boundaries of the feature and the binary edge superimposed onto the image following one of the inner boundaries.	132
Figure 5.14. View of a (a) cross-sectional through the CT reconstructed volume to show the (b) cavity shape and step height. Modified from [160].	133
Figure 5.15. Spatial dimensions and cubic fit coefficients of the asymmetric figure. Modified from [160].	134
Figure 5.16. CT spatial and curve fitting measurements for the individual slices.	134
Figure 5.17. Residuals of the measured CSI and CT data from nominal and fitted polynomial lines [160].	135
Figure 5.18. Dimensional measurement results for the four stitching algorithms.	137
Figure 5.19. Comparison of the tangent-normal deviations as measured from the best fit cubic curve for the four stitching algorithms.	138
Figure 5.20. Intensity and phase images using the ‘Cartesian’ stitching algorithm.	139
Figure 5.21. Intensity and phase images using the ‘Overlay’ stitching algorithm.	140
Figure 5.22. Intensity and phase images using the ‘XY Adjust’ stitching algorithm.	141
Figure 5.23. Phase images using the ‘Adaptive Adjust stitching algorithm.	142

Figure 5.24. Radial measurements of the chain-link arcs.	143
Figure 5.25. Chain link plots of the (a) CSI and (b) X-ray CT data.	144
Figure 5.26. Residuals of the (a) CSI and (b) X-ray CT data from the fitted arcs.	146
Figure A.1. First attempt at a kinematic mount for the interferometer. The flaw in a single-clamp design was as the clamp was engaged, the vertical plate rotated.	177
Figure A.2. Comparison of the beam amplitude at different distances from the X-ray source.	177
Figure A.3. Comparison of the FWHM spot sizes at different distances from the X-ray source showing differences in voltages and current do not significantly affect the cone angle.	178
Figure A.4. Photograph of the temperature sensor circuit. The circuit was constructed to accommodate up to eight LM35 sensors one time.	178
Figure A.5. The LM35 sensor was coated with a brushed-on water resistant polymer after heat shrinking the electrical contacts.	179
Figure A.6. Photograph of the GM Tube sensor circuit.	179
Figure A.7. Geiger-Müller tube with a 3D printed mount for attachment to the in-line camera configuration.	180
Figure A.8. Plot of the mass attenuation coefficient for silicon, silica, and air.	180
Figure A.9. Exploded solid model showing the components for the original in-line configuration.	181
Figure A.10. Histogram of the dark frames after removing tilt using the same tilt profile.	182
Figure B.1. Custom inclinometer with kinematic mount.	184
Figure C.1. (a) High density compared to (b) low density data and their effect on simulating pixelated data from a camera.	186
Figure C.2. (a) The shaded area is the feature with the red line indicating the feature's edge. (b) Theoretical line converted into pixelated data. (c) Pixelated data from edge detection missing points on the inside edge of the feature's edge.	186
Figure C.3. Superimposing the noise profile (top) normal to the tangent of the original line.	189
Figure C.4. Plot of the asymmetric cubic curve's tangent normal distances.	190

Figure C.5. Plot of the quadratic curve's tangent normal distances.	191
Figure C.6. Plot of the symmetric cubic curve's tangent normal distances.	192
Figure D.1. Bridgeport Harig 612 surface grinder used to cut the silicon pieces into secondary interferometers.	193
Figure D.2. Components shown are (a) 1-2-3 blocks, (b) optical wax, (c) and the hot plate necessary for mounting and securing silicon piece to be machined.	194
Figure D.3. Using a 1-2-3 as the sample holder and another block as a datum plane for squaring the silicon piece to the side.	195
Figure D.4. Surface grinder with clear plastic shielding to contain the flood coolant.	197
Figure D.5. Silicon test piece cut into the approximate size of the final part. Dimensions of the blades and feed rate are critical to avoid breaking the silicon blades.	198
Figure D.6. Progression of cuts through the actual silicon piece to form the interferometers. Two interferometers were produced one block and the second block was preserved for future use. A scrap part was mounted behind the interferometers for support during machining.	198
Figure D.7. Dimensions in inches of the silicon block that would be used to create the extra interferometers.	200
Figure D.8. Locations of the first cuts using a 0.063 inch diamond wheel.	200
Figure D.9. The second cut splitting the piece into two. The width of each part is about 16.2 mm and the length is 44 mm.	201
Figure D.10. Facing the sides and cutting the interferometer blades into the block. Blades are 1.5 mm thick and the diamond wheel used is 0.263 inches thick.	201
Figure D.11. Cutting the piece once again in half using the 0.063 inch diamond wheel, creating two secondary interferometers.	202
Figure D.12. Additional images showing the single part being split into the four potential interferometers.	203
Figure D.13. Additional images following the final cuts that produce the blades before the final cut that separates the part into two interferometers.	203
Figure D.14. Additional images of the part after cutting in half to yield two interferometers.	203

LIST OF TABLES

Table 1.1. Summarized description of reference objects found in literature. See Figure 1.4 for CAD model representations.	10
Table 1.2. JIMA test resolution models	13
Table 2.1. CompactRIO chassis modules.	40
Table 3.1: Component values used in the Geiger-Müller circuit shown in Figure 3.10.	56
Table 3.2: Atik 428EX technical specifications [124].	65
Table 4.1. Baseline recipe for silicon etching with the ASE.	86
Table 4.2. Processing pressure etch results for 15, 20, 25, and 30 mTorr.	89
Table 4.3. SF ₆ flow rate etch results at 120, 130, and 140 cm ³ ·min ⁻¹ measured using the confocal microscope.	90
Table 4.4. Bonding attempt order of operations.	94
Table 4.5. Feature names and their initials.	95
Table 4.6. Linear features and their nominal dimensions.	97
Table 4.7. Curved features and their nominal dimensions.	99
Table 4.8. Polynomial coefficients for the full-scale features. $y = Ax^3 + Bx^2$	99
Table 4.9. Spin coater settings for Microposit S1813 G2 photoresist.	104
Table 4.10. Silicon DRIE (passivate and etch) and photoresist strip settings.	106
Table 4.11: Oxygen plasma activation settings using the STS AOE system.	109
Table 4.12: Programmed furnace settings using for direct bonding.	111
Table 5.1. Nikon XT H 225 ST scan settings.	117
Table 5.2. Standard deviations of the tangent-normal residuals as measured by the nominal coefficients and coefficients calculated by a best fit cubic curve.	137
Table 5.3. Centroid-to-centroid distance measurements of the chain link.	145
Table 5.4. Standard deviations of the chain link's arcs.	145

Table C.1. Logic to determine whether the point falls inside or outside the feature.	187
Table C.2. Comparison of 1σ standard deviation values between CSI data and simulation.	190

LIST OF ABBREVIATIONS

ADC	– Analog-to-digital converter
ADU	– Analog-to-digital unit
AFM	– Atomic force microscopy
AOE	– Advanced oxide etcher
ASE	– Advance silicon etcher
BBB	– Triple Bragg
BLB	– Bragg-Laue-Bragg
CAD	– Computer-aided design
Ce:YAG	– Cerium-doped yttrium aluminum garnet
Ce:LuAG	– Cerium-doped lutetium aluminum garnet
CFRP	– Carbon-fiber reinforced polymer
CMM	– Coordinate measurement machine
CMOS	– Complementary metal–oxide–semiconductor
CMS	– Coordinate measurement system
CSI	– Coherence scanning interferometer
CT	– Computed tomography
CTE	– Coefficient of thermal expansion
DI	– Deionized
DC	– Direct current
DRIE	– Deep reactive-ion etching
FCC	– Face-centered cubic
FPGA	– Field programmable gate arrays
FWHM	– Full-width half-maximum
GdOS:Tb	– Terbium-doped gadolinium oxysulphide
GM	– Geiger-Müller
HF	– Hydrogen fluoride
ICP	– Inductively coupled plasma
IPA	– Isopropyl alcohol
ISR	– Interface structural resolution
ISO	– International Organization for Standardization
JIMA	– Japan Inspection Instruments Manufacturers' Association

LBL – Laue-Bragg-Laue

LBBL – Laue-Bragg-Bragg-Laue

LLL – Triple Laue

NA – Numerical Aperture

NBS – National Bureau of Standard (see NIST)

NDT – Non-destructive testing

NIST – National Institute of Standards and Technology (formerly NBS)

PDMS – Polydimethylsiloxane

PET – Polyethylene terephthalate

PMMA – Polymethyl methacrylate

PTB – Physikalisch-Technische Bundesanstalt

PTFE – Polytetrafluoroethylene

PVD – physical vapor deposition

RIE – Reactive-ion etching

SC1 – Standard clean 1 (aka RCA1)

SC2 – Standard clean 2 (aka RCA2)

STR – Surface topography repeatability

STS – Surface Technology Systems (now known as SPTS Technologies)

TMAH – tetramethylammonium hydroxide

XRD – X-ray diffractometer

CHAPTER 1: INTRODUCTION

1.1 History of X-rays

X-ray systems have been around for a little more than a century now and their impact on science have transformed the world as we see it, or rather, transformed our understanding of the world we cannot see. The physics of X-rays that allow us to see what cannot be seen are a double-edged sword. On the one hand, the ionizing radiation is destructive to biological tissue, polymers, and electronics. Notable victims of radiation induced erythema, illnesses, and possibly death include Thomas Edison's assistant Clarence Dally, two-time Nobel Prize winner Marie Curie, and highly revered X-ray crystallographer Rosalind Franklin [1]. The effect of radiation on polymers can be degradation or accelerated polymerization [2–5]. With electronics, ionizing radiation can produce voltage spikes and added noise. The increase in current associated with the voltage spike leads to reading errors and permanent damage [6,7]. Aerospace and nuclear reactors employ radiation hardening techniques to reduce glitching and permanent damage from the radiation [8].

On the other hand, X-rays are invaluable to scientific research. The first X-ray image was that of Anna Bertha Ludwig's hand, the wife of Wilhelm Röntgen in 1895 [9]. With this newfound knowledge, John Hall Edwards demonstrated the benefits of X-rays for medical application by recording a radiograph image of a needle inside a hand. Around 1906, physicist Charles Barkla discovered that exposing elements to X-rays, characteristic X-rays were emitted. The birth of X-ray crystallography occurred in 1912 when Max von Laue and fellow researchers were successful in their diffraction of X-rays by crystals.

Following in von Laue's footsteps, the father-son team of William Henry Bragg and William Lawrence Bragg formulated Bragg's law that links the reflection of X-rays to the spacing of crystal planes [10]. Around the same time, X-ray generator design received a substantial boost in intensity with the introduction of the Coolidge tube, patented by William D. Coolidge and capable of operating upwards of 100,000 volts [11]. In 1915, William Duane and Franklin Hunt formulated the law that relates frequency of emitted X-rays to the energy of the x-ray generator [12]. Two years later, Coolidge filed US patent 1,215,116 for his invention of the rotating anode X-ray tube that improves heat distribution [13]. During the 1920's, the dangers of X-ray radiation started to become more widely known and lead to formation of the International Committee on Radiological Protection [14]. X-ray powder diffractometer was unveiled in 1935 by Le Galley, with the first commercial powder diffractometers sold in the USA by Philips starting in 1947. Then in 1953, one of the most widely recognized scientific achievements, the discovery of DNA's double helix as reported by Francis Crick and James Watson [15]. The foundation for X-ray computed tomography began in 1959 with William H. Oldendorf building a prototype system that rotated the X-ray source and detector around an object, and then reconstructed the image [16,17]. Ulrich Bonse and Michael Hart introduced the single-crystal X-ray interferometer in 1965 that produced phase-contrast images [18,19].

1.2 X-ray Interferometry History and Applications

In the first publication about an X-ray interferometer, Bonse and Hart discussed their construction of an interferometer from a single-crystal silicon machined to have three flat and parallel blades cut with respect to a crystal plane. The incident beam on the first blade splits into two by way of Laue reflection. The beams diverge towards the second

blade, the mirror, and using the same principle create two more beams that converge onto the third blade, the analyzer. Within the analyzer blade, the converging waves are superimposed to form constructive and destructive interference. Additional theory for X-ray interferometry is provided in section 2.1.4.

Bonse and Hart's interferometer was made from single-crystal silicon and cut to use the (220) crystal plane for diffraction. In their experimental, they slid a wedge of polymethyl methacrylate (PMMA) into the path of a converging X-ray beam between the mirror and analyzer blades. Taking a topographic image of the output from the analyzer blade, the first X-ray phase-contrast image showed a distinct fringe pattern relating the wedge thickness and phase [18]. A modified version of the interferometer increased the beam's path, allowing for thicker parts to be measure. In the increased path version, they noted the sensitivity of the device to bending from the mounting forces [19]. The next interferometer designs by the pair focused on Bragg reflection or combined Bragg-Laue [20,21]. Their opinion of the Bragg-case interferometers is that since the energy is diverted, rather than absorbed as with the Laue-case, their use may be more suitable for neutron systems. For combined Laue-Bragg versions, they preferred the LBBL configuration rather than the BLB and LBL because of relaxed requirements and believed with refinement it could exceed the capability of previous interferometer designs.

Single-crystal, monolithic interferometers are limited in field of view when used for phase-contrast imaging by the size of the crystal ingot. Becker and Bonse partly overcame this problem by designing the skew-symmetric two-crystal interferometer which splits the middle blade into two offset parts [22]. Momose and others built upon the skew-symmetric design and managed to increase the field of views to up to areas of $60 \times 30 \text{ mm}^2$

[23,24]. While the manufacturing difficulty of skew-symmetric interferometers is roughly the same as an LLL design, skew-symmetric do require higher precision in mounting and relative rotational alignment to maintain a stable optical beam path [25]. As for the fringe resolution of the system, Momose *et al* determined that thinning the analyzer blade increased the interference pattern resolution [26,27].

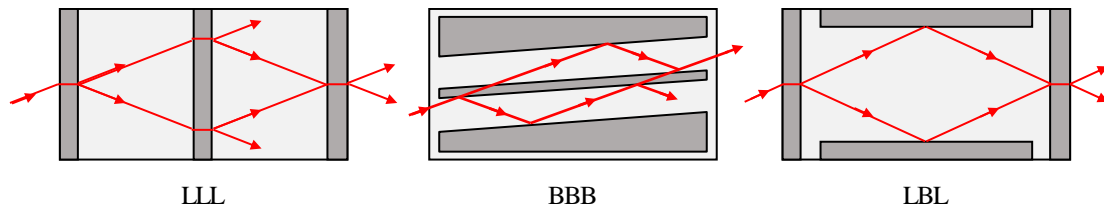


Figure 1.1. Examples of X-ray interferometer configurations. From left to right are Laue-case, Bragg-case, and Laue-Bragg-Laue. The incident blade is the splitter, followed by the mirror(s), and analyzer blade.

These modifications are important because one of the benefits to X-ray interferometry is that sensitivity to phase-contrast can be upwards of 1000 times higher than absorption-contrast [28]. Biomedical has invested the most into phase-contrast imaging, where low absorption materials are abundant. Examples of biomedical materials imaged are cancerous masses, blood cells, fat tissues, brain, liver, and hearts [23–25,28–30]. Radiographic images are not the only images that have been taken of these samples as Momose *et al* have combined phase-contrast with computed tomography to image soft tissues for reconstruction into three-dimensional renders [25,29–32]. While the LLL-case interferometer is still used from time-to-time for phase-contrast imaging, much of that research has switched to phase-grating (or Talbot-Lau) interferometers [33–38]. Unlike the interferometer designs discussed so far, phase-grating interferometers can operate with polychromatic, cone-beam X-rays.

Shortly after Bonse and Hart published their research on X-ray interferometers, another application discussed by Hart was to use the device for measuring displacement. The ångström ruler, as Hart called it, functions by translating the third blade and counting of moiré fringes which directly relate to the crystal parameters [39]. In engineering metrology, the first coupling of the X-ray interferometer to another optical interferometer was by Deslattes at the National Bureau of Standards (renamed to the National Institute of Standards and Technology in 1988) [40]. By coupling Fabry-Perot and X-ray interferometers, Deslattes and Henins made absolute measurements of the silicon lattice spacing, and the Cu $K\alpha_1$ and Mo $K\alpha_1$ characteristic emission lines in terms of a 633-nm He-Ne stabilized laser [41]. Combined optical and X-ray interferometer systems have since been built by Basile *et al* and Yacoot *et al* [42–45]. In terms of non-optical based systems coupled to an X-ray interferometers, there are scanning probes and atomic force microscopes [46–51]

A practical application of X-ray interferometers is the calibration of transducers with traceability to silicon lattice parameters. The first usage as a metrological reference was suggested by Chetwynd *et al* where an interferometer was coupled to the head of a Rank Taylor Hobson Talystep with an inductive displacement transducer [52]. Feasibility of measuring translations down to 10 pm was assessed with results reported by Bowen *et al* [53,54]. Following calibration of an inductive transducer, displacement transducers that have also been calibrated by an X-ray interferometer include linear variable differential transformers (LVDT), capacitance gauges, and optical interferometers [55–59]. As the transducers can have ranges into the micrometers (and beyond), alternative monolithic X-ray interferometer designs with extended range have been evaluated by Chetwynd *et al*.

[60]. At both the University of Warwick and Physikalisch-Technische Bundesanstalt (PTB), long-range interferometers capable of moving beyond their 10 μm range target have successfully constructed [50,61].

Angular sensitivity promotes the idea of an X-ray interferometer as a high precision angle-measuring device [62–66]. Demonstrated by Becker and Stümpel, by displacing and rotating the second mirror and analyzer blades of their skew-symmetric interferometer relative to the splitter and first mirror (see Figure 1.2), a modulation in the beam intensity and a 10 nrad periodicity were measured [62,63]. Chetwynd *et al* took a different route to measuring angle by machining a notch hinge below the analyzer blade of their LLL monolithic interferometer. Although the hinge produced a center of rotation 35 mm from the center of the analyzer blade, this offset was not found to cause any practical problems. Although limited to a 6 mm wide image from the beam collimation, they were able to show that at zero load there was about 20 nrad misalignment of the blades. Using dead-weights, a near-linear relationship between torque and change of pitch was shown, with a 2 g load producing over 500 nrad of rotation and about 16 fringes [66].

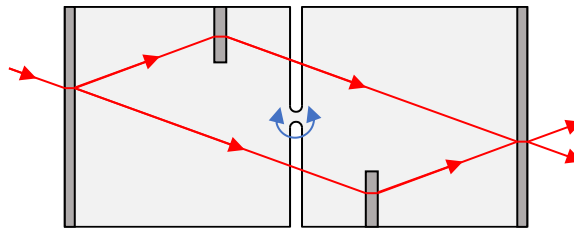


Figure 1.2. Becker and Stümpel's skew-symmetric X-ray interferometer with a hinge at the center.

1.3 Reference Objects for X-ray Computed Tomography

The first X-ray computed tomography (CT) scanner was built for medical imaging by Godfrey Hounsfield around 1969. Two short years later and after continual development, a system built by Hounsfield was installed in London's Atkinson-Morley Hospital. Although limited to 80×80 pixels, the scanner was able to scan a layer in under five minutes and reconstruct the cross-section of a patient's brain in seven minutes [67–69]. With the technology improving, some of the earliest attempts at non-destructive testing (NDT) for industrial applications emerged in the 1980's and as a dimensional measurement tool by around 1991 [69]. Industrial CT has since exploded into the metrology scene as a trending topic of research. This interest in industrial CT is built from the capability of reconstructing both external and internal geometries for measuring simultaneously material and dimensional quality using millions of data points. Two applications where X-ray CT is sometimes the only measurement system available are in additive manufacturing and composites where internal structures, voids, and porosity are common and can adversely influence structural integrity [70–74].

Because of the complexity of these systems, X-ray CT measurement is not without its difficulties. By using an Ishikawa diagram, the influencing factors can be separated into five distinct groups: CT system, data processing, workpiece, environment, and operator. Within the CT system are the X-ray source, detector, and positioning system. Data processing is 3D reconstruction, threshold determination for surface generation, feature fitting methods, and data corrections such as filtering and scaling. Influences by the workpiece are material composition, dimensions, geometry, and surface finish. Environmental influences are temperature, humidity, and vibration. And finally, the

influences from the operator are their experience, workpiece fixturing, and choosing appropriate scan settings (e.g. X-ray source settings, magnification, number of projections, and image averaging). Others have broken down the influential factors in more-or-less the same way [67,75–79]. With so many influential factors affecting quality of measurement, quantifying uncertainties with traceability to national standards is a formidable task [80–83].

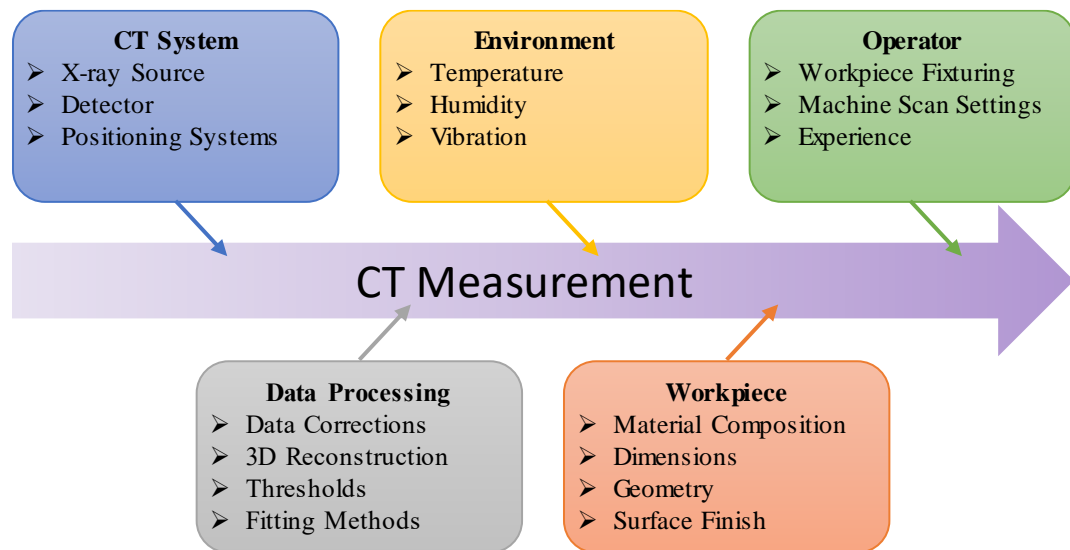


Figure 1.3. Influential factors in CT measurement presented as an Ishikawa diagram.

In order to qualify and verify the performance of a CT system, manufacturers and operators have relied on reference objects. Reference objects are physical items that have been qualified themselves with uncertainties as measured by a calibrated instrument; typically, an industrial coordinate measuring machine. There are a few design principles that should be followed when building a reference object. Choosing a dimensionally stable material is first and foremost. The material should not exhibit creep, have a low coefficient of thermal expansion (CTE), and should be protected from corrosion. Absorption should be considered, as a high-absorption material will require more power and higher x-ray

energies for imaging. For parts that are larger than a few centimeters, materials like steel or tungsten are not good choices because of their high absorption. Metals such as aluminum and titanium are good materials as they are creep-resistant, have low CTE, form hard oxides on their surfaces that can protect them from further corrosion, and have low absorption.

A common feature found on reference objects are ruby, steel, or ceramic precision spheres. With precision spheres, the influence on the centroid calculation by surface determination algorithms and beam hardening can be minimized. At the same time, knowledge of the sphere's diameter and form can help in testing the surface determination procedure [67]. Most approaches involve mounting spheres onto rods or plates constructed from carbon-fiber reinforced polymer (CFRP) [67,69,76,84–89]. A design consisting of four spheres stacked into a tetrahedron has also been demonstrated [90]. Reference objects not incorporating precision spheres are the calotte plate and cube [91–93], multi-material cube with holes [94], hole plates [95], fiber gauges [75,96], cactus standard [97,98], step gages [99,100], step cylinders [92,95,101–104], and pan flutes [105]. Descriptions of reference objects mentioned are summarized in listed in Table 1.1 with CAD models provided in Figure 1.4. By volume, the smallest reference object listed with traceability from a CMM was a tetrahedral stack (Figure 1.4c) using 0.5 mm spheres. The largest part by volume found in the literature was a step cylinder (Figure 1.4h) of major diameter 220 mm, 160 mm tall. In terms of the part with the smallest traceable dimension, the glass fibers used in the fiber gauge (Figure 1.4i) were 125 μm diameter with heights between 350 μm to 700 μm .

Table 1.1. Summarized description of reference objects found in literature. See Figure 1.4 for CAD model representations.

Object Name	Material and Dimensions
CT Tree	Ø 3 mm ruby spheres, CFRP rod lengths from 16 mm to 40 mm
Ball Plate	Ø 5 mm ruby spheres, 10 mm pitch, 2 mm thick CFRP plate
Tetrahedral Stack	Ø 0.127 mm or larger ruby spheres, glued together or set into a holder
Calotte Cube	Titanium, 10 mm edge lengths, Ø 0.8 mm calottes
Hole Plate	Steel, $6 \times 6 \times 1 \text{ mm}^3$, Ø 1 mm holes Aluminum, $48 \times 48 \times 8 \text{ mm}^3$, Ø 4 mm holes
Cactus Standard	Aluminum, 45 mm edge lengths, 5 mm horizontal surface distances
Step Gauge	Variety of materials, 2 mm wide steps, 4 mm pitch
Step Cylinder	Aluminum, Ø 40 mm to 220 mm, 160 mm tall
Fiber Gauge	Ø 125 µm glass fibers, 350 µm to 700 µm lengths
Pan Flutes	Ø 1.9 mm glass cylinders, 2.5 mm to 12.5 mm lengths, CFRP mount

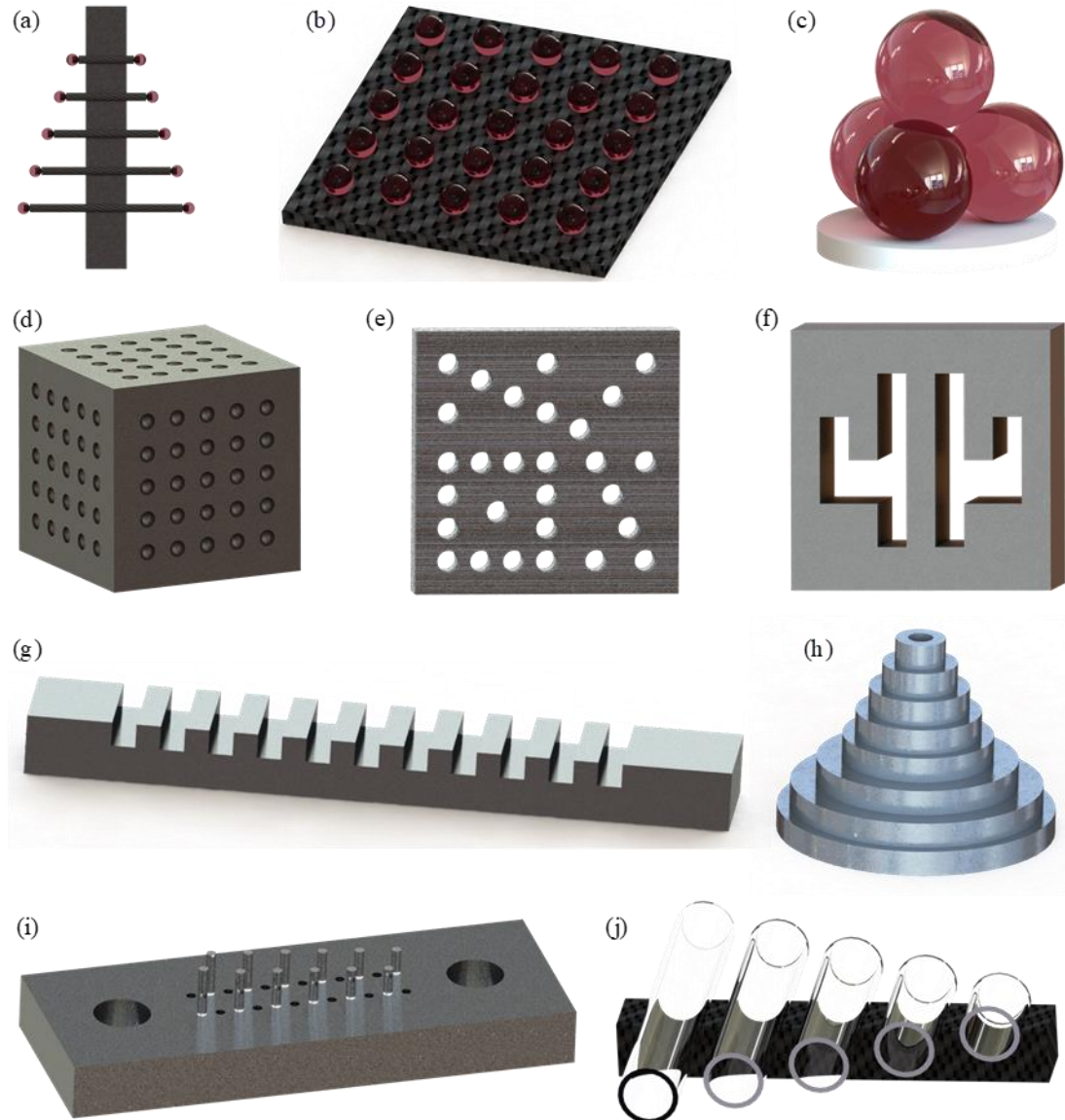


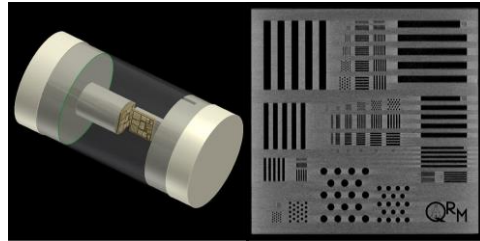
Figure 1.4. CAD drawings of X-ray CT reference objects that have been found in the literature. (a) CT tree with ruby spheres attached to CFRP rods and mounted to a CFRP bar. (b) Ball plate with ruby spheres attached to a CFRP plate. (c) Tetrahedral stack of ruby spheres. (d) PTB calotte cube. (e) PTB hole plate. (f) Cactus standard. (g) Step gauge. (h) Step cylinder. (i) Fiber gauge with glass fiber optic cables of varying heights. (j) Pan flute gauge made from glass cylinders mounted on a carbon fiber support.

The reference objects described so far are simple designs, having either been cut from a monolithic piece of material or assembled with multiple materials using adhesives. With adhesive bonding, the mounted parts are subject to potential creep and will need to be revalidated over time. For the monolithic parts, their manufacturing processes limit their smallest feature size in the hundredths of millimeters. While these types of reference objects are suitable for calculating scale, penetration length, developing methods beam-hardening correction, or testing surface determination, they are not designed to measure the interface structural resolution. The interface structural resolution (ISR) is defined as the minimum distance measurable between two parallel surfaces [67]. For ISR measurements, the reference objects are silicon with grooves that have been etched or deposited with a metal (e.g. gold, tungsten). ISR patterns are commercially manufactured by QRM GmbH and by the Japan Inspection Instruments Manufacturers' Association (JIMA) [106–108]. QRM GmbH produces two variants of their ISR reference object with shapes limited to lines and circles. The larger (Figure 1.5a) of the two offerings is a $5 \times 5 \text{ mm}^2$ silicon substrate with etched patterns that range in size from $5 \text{ }\mu\text{m}$ to $150 \text{ }\mu\text{m}$ and depths between $80 \text{ }\mu\text{m}$ to $120 \text{ }\mu\text{m}$. For protection, two substrates are mounted inside a 20 mm diameter plastic cylinder or inside an 8 mm diameter resin casting, 40 mm long. The second version, called the 'BarPattern Nano' (Figure 1.5b), is a $3 \times 3 \text{ mm}^2$ silicon substrate with $1 \text{ }\mu\text{m}$ to $10 \text{ }\mu\text{m}$ patterns and depths between $5 \text{ }\mu\text{m}$ to $15 \text{ }\mu\text{m}$. The 'Nano' has two substrates mounted to a plastic base and is protected inside of a plastic cylinder, 5.2 mm diameter and 19 mm long. JIMA approach to building ISR reference objects is to deposit gold or tungsten onto a silicon substrate. Patterns are limited to lines and offered in models CT-01, RC-04, RC-05, or (RC-02B) resolutions. Specifications for each JIMA model is provided in Table 1.2.

Table 1.2. JIMA test resolution models

Model	Base	Size (mm ³)	Absorption Lines	Protection Film	Mount
CT-01	Si	$0.8 \times 7 \times 0.2$	Au, 3 μm to 7 μm	N/A	Acrylic, $\varnothing 2 \text{ mm} \times 40 \text{ mm}$
RC-04	Si	$5 \times 5 \times 0.015$	W, 0.1 μm to 10 μm	PET	Al, $30 \times 40 \times 5 \text{ mm}^3$
RC-05	Si	$8 \times 8 \times 0.2$	Au, 3 μm to 50 μm	Al vapor film	Al, $30 \times 40 \times 5 \text{ mm}^3$
RC-02B	Si	$5 \times 5 \times 0.06$	W, 0.4 μm to 15 μm	Polycarbonate	Al, $30 \times 40 \times 5 \text{ mm}^3$

(a) QRM Micro-CT BarPattern



(b) QRM Micro-CT BarPattern Nano

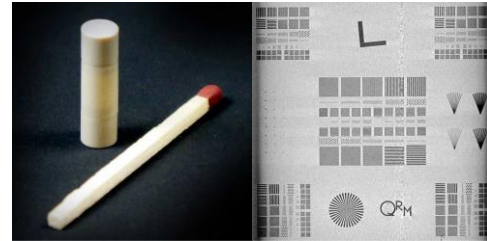


Figure 1.5. Interface structural resolution reference objects produced by QRM GmbH. (a) Standard model silicon chip size is $5 \times 5 \text{ mm}^2$ and is mounted inside a 20 mm diameter, 40 mm long cylinder filled with air. (b) Nano model silicon chip size is $3 \times 3 \text{ mm}^2$ mounted to a plastic base that is 5.2 mm diameter and 19 mm long. Images from QRM GmbH's website [106].

1.4 Research Objectives and Dissertation Outline

From phase-contrast imaging to displacement measurement, X-ray interferometers have proven useful as a high precision, traceable measurement device. Benefits of phase-contrast imaging have to date been focused on biomedical applications, work that has been often more qualitative than quantitative. Phase-contrast imaging of objects for measurement is an area of research that has yet to establish itself, despite the growth in popularity of industrial X-ray CT measurement research. A couple of applications where such a tool could be used is non-destructive measurement of thin-films and of sub-millimeter parts with low absorption, such as those produced by multiphoton additive

manufacturing. By building an adaptive X-ray facility, advancements in interferometry are possible, in addition to other areas of research that require an X-ray micro-source beam.

The second half of the dissertation is on the development of a new reference object for the assessment and calibration of X-ray systems. Where the new reference object differs from others that were described earlier is the inclusion of engineered cavities, produced via direct bonding of etched silicon wafers.

The arrangement of the dissertation is as follows. Chapter 2 of the dissertation describes the construction of the facility. Chapter 3 summarizes the performance of the facility and its components. Chapter 4 discusses background of the manufacturing processes, describes the features of the proposed reference object, and contains the successful manufacturing methodology. In Chapter 5, results of a study comparing coherence scanning interferometer data to X-ray computed tomography reference objects is presented. Chapter 6 summarizes the major conclusions from both bodies of work. Finally, Chapter 7 suggests potential avenues for future work.

1.5 Summary of Research Goals

The research goals are summarized as the following:

1. Build an enclosure for the X-ray generator and auxiliary components that satisfies safety regulations set forth by the university's Environmental Health and Safety Department and any state or federal agencies.
2. Develop a control system for the X-ray system instruments.
3. Evaluate the performance of the auxiliary instrumentation and X-ray generator.

4. Design and fabricate a three-dimensional X-ray computed tomography reference object with engineered cavities.
5. Create an algorithm for extracting feature parameters from X-ray reference object data from various tools for a comparison study.

CHAPTER 2: A FACILITY FOR X-RAY METROLOGY

In this chapter, the theory of X-ray generation, absorption, detection, and interferometry are discussed in section 2.1. In section 2.2, the construction of the X-ray facility is described which includes the enclosure, X-ray system and controls, the acoustic damping enclosure for the water chiller, and the start of a crystal aligning device for interferometer manufacturing.

2.1 Principle of X-ray Operation

Before discussing the design and building of the X-ray research facility, knowledge of the generation, detection, and diffraction of X-rays for interferometry is informative. The first subsection details the physics of generating X-ray photons and how this applies to a modern X-ray generator and is followed by a subsection on X-ray absorption. To illuminate the principles behind a custom x-ray camera discussed in section 3.2.4, section 2.1.3 reviews the detection of X-rays with methods limited to those used within the facility. These detection methods are Geiger-Müller tubes, an ionization chamber, and crystal scintillator. As the original purpose of the facility was for X-ray interferometry studies, the final subsection covers crystallography and diffraction.

2.1.1 Generation

Between the ultraviolet and gamma ray wavelengths is the spectrum of electromagnetic radiation referred to as X-rays. X-rays span from 10 nm to 0.01 nm with energies between 100 eV and 100 keV. Energies above 10 keV are sometimes called hard X-rays and those below called soft X-rays. X-ray output is in two forms, bremsstrahlung and characteristic emission. The continuous spectrum called the bremsstrahlung radiation,

or “braking radiation”, is the byproduct of an electron being deflected by the atomic nucleus and losing kinetic energy. As energy is conserved, a photon of equivalent energy to the kinetic energy loss is created. Bremsstrahlung radiation’s minimum wavelength is calculated using the Planck-Einstein relation (Equation 1.1)

$$E = hv = \frac{hc}{\lambda} \quad (1.1)$$

where h is Planck’s constant, v is frequency, c is the speed of light, and E is the electron charge given in eV. When solved for the minimum wavelength λ_{min} , the relationship is referred to as the Duane-Hunt Law. With respect to the Bede Scientific microsource used in the facility, at the maximum voltage of 50 kV, λ_{min} is 25 pm [12].

Characteristic emission involves ejecting an electron from the inner shell following a collision with an incident electron of higher energy than the binding energy. Outer shell electrons try to fill the vacancy and create a photon with an energy level equal to the difference of the binding energies. The classification for the emission is called Siegbahn notation. In this notation, the inner shell name is listed first with a subscript describing the difference in principal quantum number. For the case where an electron from the L shell fills a K shell vacancy, the emission line is called K_{α} and often has the strongest intensity of the emissions. Within the K_{α} emission are two slightly different characteristic emissions, $K_{\alpha 1}$ and $K_{\alpha 2}$. A diagram of the characteristic emission process is shown in Figure 2.1. When an electron moves from the M_2 or M_3 shell to fill a vacancy in the K shell, the emission line is called K_{β} .

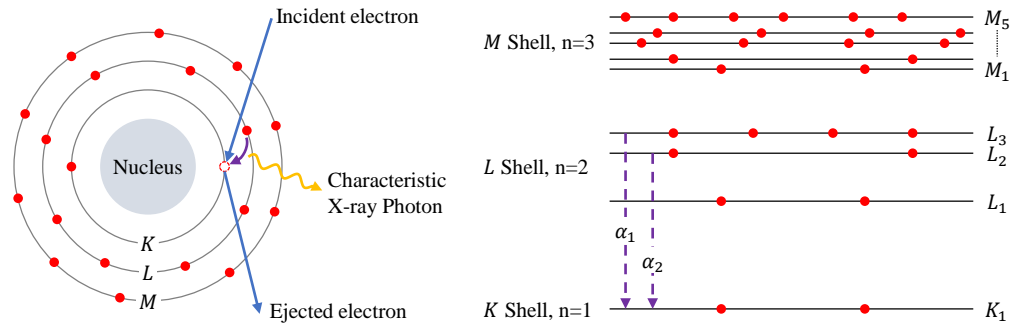


Figure 2.1. Electron from the E-beam collides with a core electron in the K shell, resulting in an electron from a higher energy shell to drop down.

An X-ray generator with a molybdenum target operating at 35 kV has a spectrum similar to the one shown in Figure 2.2. From the diagram, it is easy to locate the characteristic emissions for K_{α} at slightly larger than 0.07 nm and K_{β} near 0.065 nm with intensity spikes exceeding the bremsstrahlung continuum. To determine the binding energies, a plot of the mass attenuation coefficient shows absorption edges for each line within a shell. Figure 2.3 is a plot of the mass attenuation coefficient for molybdenum with arrows marking the K and L lines. The binding energies are 20 keV for K_1 , 2.866 keV for L_1 , 2.625 keV for L_2 , and 2.520 keV for L_3 . Taking the difference between K_1 and L_3 yields $K_{\alpha 1}$, and between K_1 and L_2 yields $K_{\alpha 2}$ of the characteristic emissions. Therefore, the energy of $K_{\alpha 1}$ is 17.480 keV at 0.07093 nm wavelength and the energy for $K_{\alpha 2}$ is 17.375 at 0.07136 nm wavelength. The wavelengths correspond to the characteristic emission spikes of the spectrum diagram from Figure 2.2.

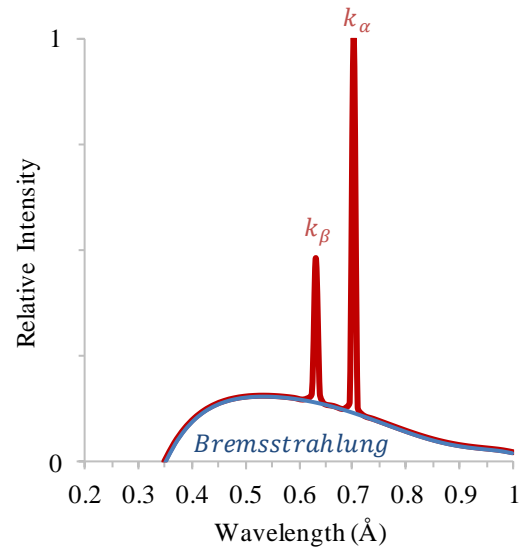


Figure 2.2. X-ray spectrum of a molybdenum target at 35 kV. The line widths are not to scale.

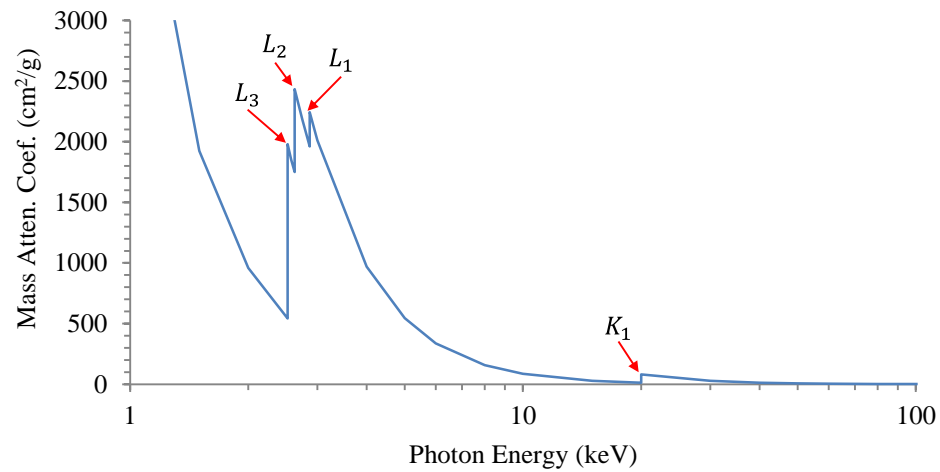


Figure 2.3. Molybdenum mass attenuation coefficient with labeled absorption edges [109].

While radioactive decaying material (e.g. uranium) can be used as a source for X-ray emission, an X-ray generator is better for safety, controllability, and higher emissions. When Wilhelm Röntgen discovered X-ray radiation in 1895, it was while working with a Crookes tube. The Crookes tube is a cold cathode ray tube that generates electrons and ions

under partial vacuum by ionizing the residual air with a high voltage DC power supply. The high potential drives the electrons towards the device's anode and glass wall, producing X-ray photons in the process.

Modern day X-ray generators, such as the Bede Scientific Microsource, are closely related to hot cathode tubes. A potential between the cathode and anode drives the stream of electrons towards the anode target, passing through a magnetic field focusing lens. As the electrons interact with the anode target, X-ray photons are created and exit the casing through a low absorption beryllium window. Majority of the energy introduced to the target is converted into heat that is removed by backside water cooling. A schematic showing the typical components and operation is provided in Figure 2.4. Additional focusing can be employed through X-ray capillary optics which relies on reflection [110].

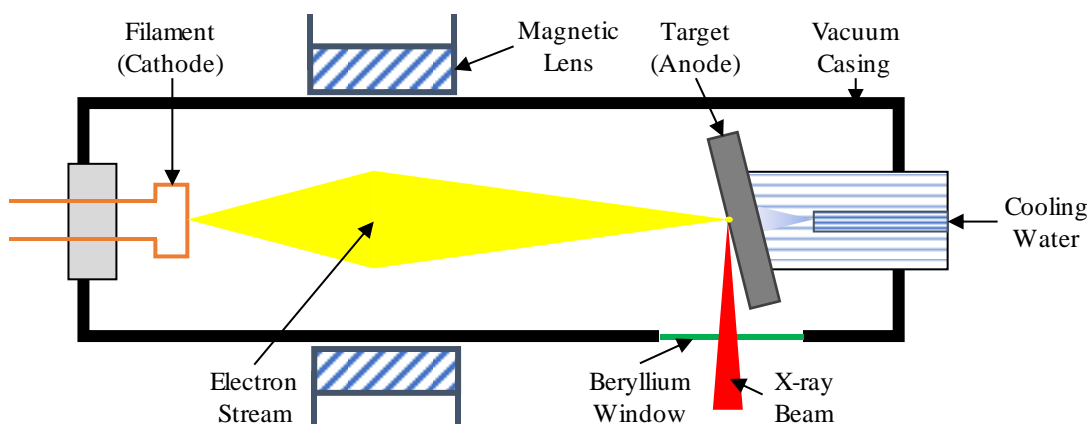


Figure 2.4. Schematic of a modern X-ray generator.

2.1.2 Absorption

The product of the mass attenuation coefficient (μ/ρ) and density (ρ) yield the linear attenuation coefficient (μ). To give perspective, a graph comparing silicon and lead is presented in Figure 2.5. Lead's high density makes it a useful component in X-ray

shielding whereas silicon's low density can be a challenging material to inspect. For lead, the mass attenuation coefficients at 17.5 keV and 50 keV are approximately $120.99 \text{ cm}^2\cdot\text{g}^{-1}$ and $8.04 \text{ cm}^2\cdot\text{g}^{-1}$. Silicon's mass attenuation coefficients for the same photon energies are respectively $6.83 \text{ cm}^2\cdot\text{g}^{-1}$ and $0.44 \text{ cm}^2\cdot\text{g}^{-1}$. By knowing a material's linear attenuation coefficient and thickness, the attenuation of intensity is computed from

$$I = I_0 e^{-\mu x} \quad (2.1)$$

Where:

I is final intensity

I_0 is the initial intensity

μ is the linear attenuation coefficient in cm^{-1}

x is the thickness in cm

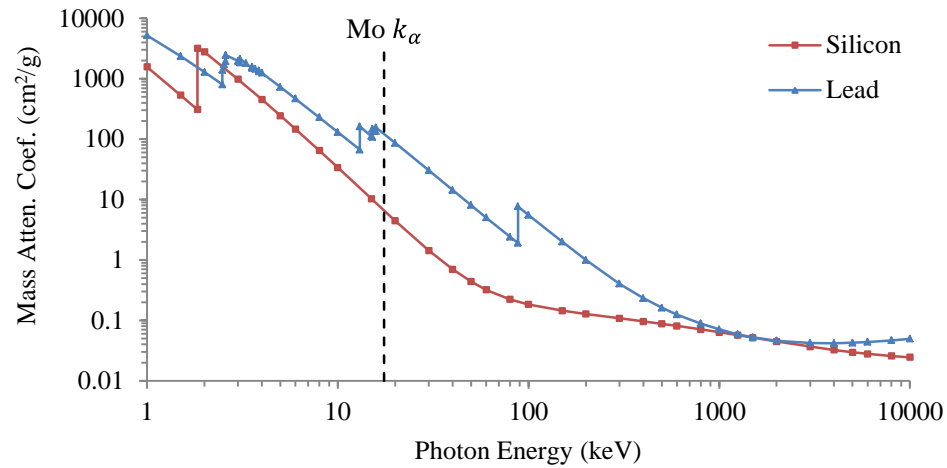


Figure 2.5. Silicon and lead mass attenuation coefficient with the $K\alpha$ characteristic emission marked [111,112].

2.1.3 Detection

There are two types of gas-based detection methods used in the project for dose measurement: an ionization chamber and a Geiger-Müller tube. Another type of detector

is a crystal scintillator used for imaging with a digital camera. Starting with the Geiger-Müller tube (Figure 2.6), the device is sealed tube containing a low-pressure gas mixture. The wall of the tube acts as a cathode and inside the tube is an anode wire. A high voltage potential, on the order of hundreds of volts, is applied to the tube. When an X-ray photon enters the gas, molecules become positively ionized by giving off free electrons. These positively charged ions flow towards the negatively charged cathode wall while the free electrons flow towards the positively charged anode wire. As the free electrons move closer to the anode wire, the increasing electric field strength accelerates the electron. Collisions with more gas particles occur, generating new ion pairs and creating an avalanche of electrons. This interaction happens within microseconds and produces a measurable pulse of voltage [113].

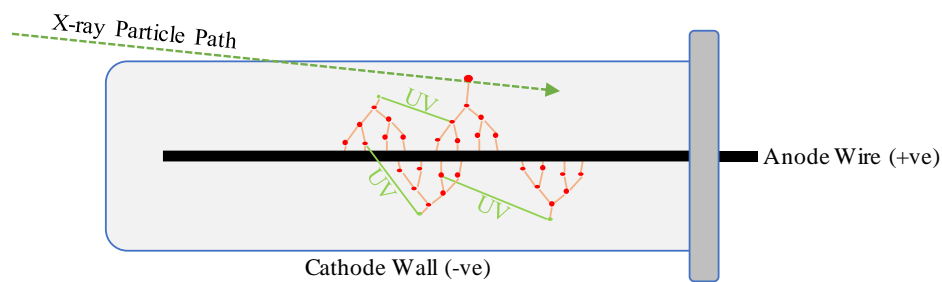


Figure 2.6. An x-ray photon enters the Geiger-Müller tube and creates an ion pair. As the electron accelerates towards the anode wire, more ion pairs form and an avalanche of electrons generates a measurable pulse of voltage.

An ionization chamber operates in a similar manner to the Geiger-Müller tube. A chamber is filled with a gas and has an anode and cathode electrode. Ionization chambers are configured as either a tube with a coaxial wire or as parallel plates. Interaction between an X-ray photon and gas particle forms an ion pair, with each moving towards the electrode of opposite polarity. Where the ionization chamber differs from a Geiger-Müller tube is

that the free electron is not accelerated to the high energy necessary for the avalanche effect. Instead of a pulse output, a continuous current between femtoamperes to picoamperes is measured [113].

For imaging by camera, one method uses a scintillating crystal to convert the X-ray photon energy into visible light via the photoelectric effect. When an X-ray photon enters the crystal and its energy is absorbed, an electron in the valence band is excited into the exciton or conduction band. Excitons moving through the crystal lattice are captured by a dopant added to the crystal and are de-excited in the form of scintillating light. The dopant is chosen for its emission wavelength and the common dopant found in X-ray scintillators is cerium that emits between 530 nm to 550 nm. This wavelength falls into the green to green-yellow spectrum where digital cameras are sensitive. Thickness of the crystal scintillators impacts absorption and resolution. Thinner scintillating crystals can yield a higher imaging resolution than thicker crystals, but require longer exposure times to compensate for the reduction in absorbed radiation that is converted to visible light.

2.1.4 Interferometry

Interferometry is a procedure that superimposes electromagnetic waves and analyzes the interfering light. One type of interferometer is a Mach-Zehnder, which has commonality to the X-ray interferometer as depicted in Figure 2.7. Collimated light passes through a beam splitter, that splits the light into two equal beams. The two beams then interact with separate mirrors before merging at a second beam splitter and entering detectors. Intensity of the light as measured by the detector that varies by the difference in phase between two beams. Light waves perfectly in phase constructively interfere and yield

the highest intensity while any difference in phase will cause destructive interference, yielding a lower intensity.

For operation of an X-ray interferometer, the incident collimated X-ray beam first passes through a splitter blade to create two identical beams. These two beams pass through effectively another splitter called the mirror. The two converging beams are superimposed as they pass through the third blade, the analyzer, before entering a detector. To understand how the beam is split by the blades, construction of the X-ray interferometer is critically important and requires background information on crystal planes and Bragg reflection.

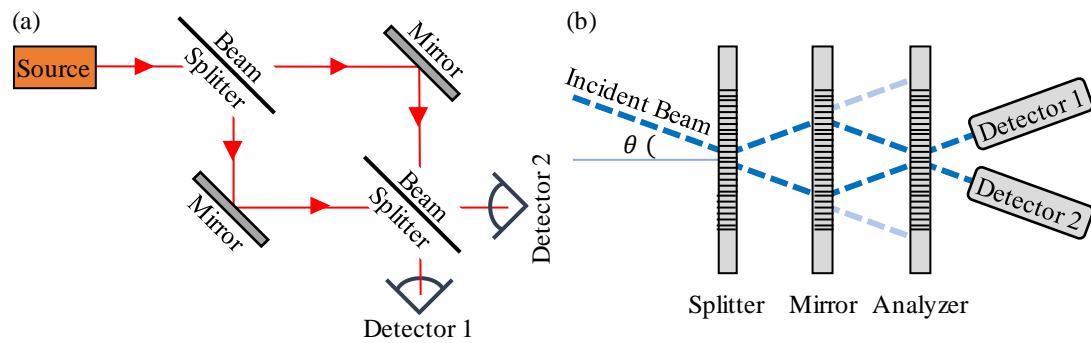


Figure 2.7. Principle of operation for a (a) Mach-Zehnder interferometer compared to an (b) X-ray interferometer.

Plane orientation for a crystal structure is described with Miller indices. A Miller index contains values that indicate where a plane intersects a given axis on a unit cell. These values are given as their reciprocal, can be negative, and are written (hkl) to represent a plane and $[hkl]$ to represent a direction. For instance, a plane that intersects the midpoint of the x-axis and y-axis, but not through the z-axis is given the Miller index (220). The two planes used for a silicon monolithic interferometer are (111) and (220) planes. A depiction of the cubic structure and the planes are shown in Figure 2.8.

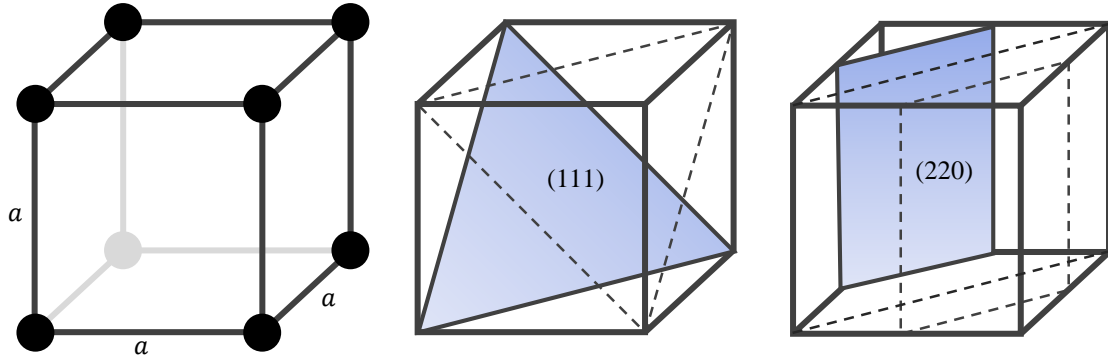


Figure 2.8. A cubic crystal with lattice constant a and examples of planes in (111) and (220) orientations.

For an X-ray photon to reflect off a crystal, the conditions must satisfy Bragg's law. Bragg's law is the angle in which a beam constructively interferes following reflection as depicted in Figure 2.9 [10]. Given that the X-ray interferometer for the facility is constructed from single crystal silicon with a diamond face centered cubic crystal structure (diamond FCC), part of the cubic crystal family, the inter-planar spacing can be calculated using Equation (2.2) [114]. The variables of Equation (2.2) are the planar spacing d_{hkl} , the three integers (hkl) from the Miller indices, and the lattice constant a . Calculating the angle θ from Bragg's law, Equation (2.3), where λ is the wavelength of light and n is an integer multiple. Not every reflection is possible. For the case of silicon that is diamond FCC, allowed reflections exist for two cases. One case is when (hkl) is all odd and the other case is when (hkl) are all even and $h + k + l = 4$. Forbidden reflections are when (hkl) is a mix of odd and even values or are all even with $h + k + l \neq 4n$.

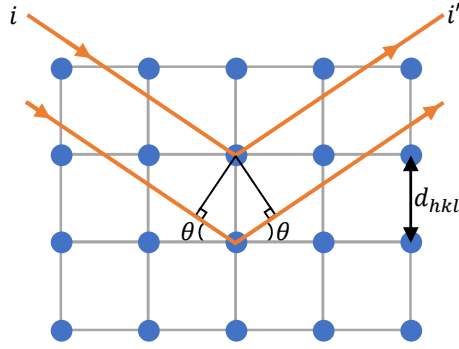


Figure 2.9. To calculate the Bragg angle, the spacing between the cubic crystal's planes d_{hkl} about one orientation and the wavelength of incident light k need to be known.

$$d_{hkl} = \frac{a}{\sqrt{h^2 + k^2 + l^2}} \quad (2.2)$$

$$n\lambda = 2d_{hkl} \sin(\theta) \quad (2.3)$$

One utilization of the X-ray interferometer is to measure the displacement of the analyzer blade as it is displaced along axis z of Figure 2.10. As the blade moves and the signal is recorded by the two detectors, each fringe is equal to spacing d_{hkl} between the crystal planes [39]. The other way to use the interferometer is to induce a phase shift by placing a specimen in one of the reflected beam paths depicted in Figure 2.11 [18]. As the beam passes through the specimen the intensity is attenuated and the beam's phase is shifted forward. This phase shift is related to the material's electron density as calculated by Equation 2.4 [115].

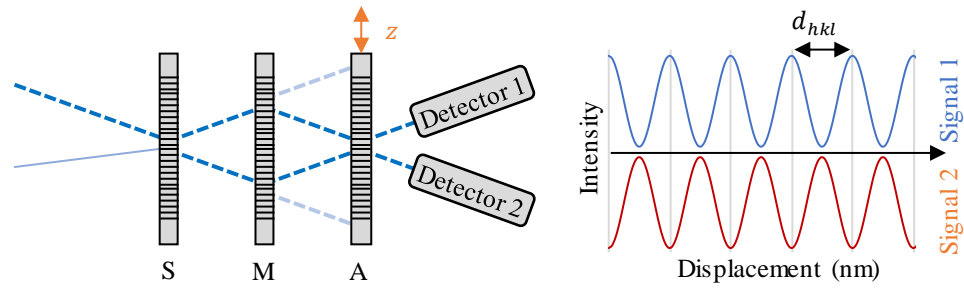


Figure 2.10. Displacement of the analyzer blade along axis z gives a direct measurement of the crystal plane spacing d_{hkl} . Signals recorded by detector 1 and 2 are opposite.

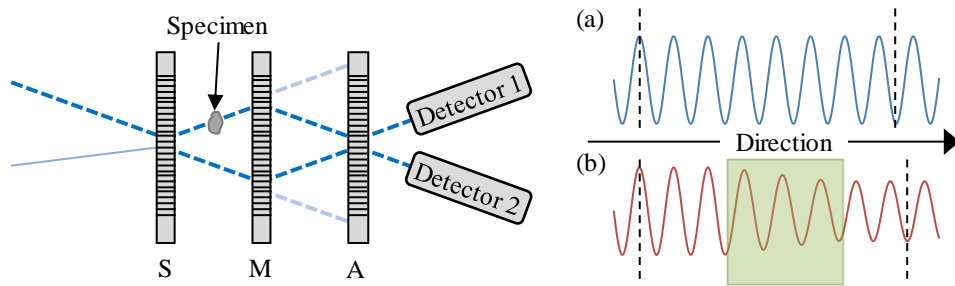


Figure 2.11. When X-ray radiation passes through a medium, the intensity of the signal is attenuated, and the phase is shifted forward. Modified from [116].

$$\phi(x, y) = -r_0\lambda \int \rho(x, y, z)dz \quad (2.4)$$

Where:

λ is the wavelength of the radiation

r_0 is the classical electron radius

$\rho(x, y, z)$ is the electron number density

Theoretically the electron number density ρ_e is given by Equation 2.5 with unit of electrons per length. This equation assumes a uniform material density ρ_m and that the number of electrons matches the atomic scattering factor f_1 . Additional components of the

equation are Avogadro's constant N_A and the molar mass M . Taking the product of Equation 2.5 and the specimen's thickness z gives the number of electrons.

$$\rho_e = \rho(x, y, z) = \frac{\rho_m N_A f_1}{M} \quad (2.5)$$

$$\phi(x, y) = -r_0 \lambda \rho_e z \quad (2.6)$$

$$I = I_1 + I_2 + 2\sqrt{I_1 I_2}(\cos \phi) \quad (2.7)$$

By substituting Equation 2.5 into Equation 2.4, the phase shift becomes a function of thickness z (Equation 2.6). Using the phase shift from Equation 2.6 in Equation 2.7 determines the output intensity for two beams of intensity I_1 and I_2 with equal frequency [117].

2.2 Facility Design

The initial goal of the X-ray facility was to build a system for X-ray interferometry, however that was later changed to build a multi-functional facility that easily adapts to growing needs. For user safety, an enclosure to house the components had to be designed, built, and inspected by the university's environmental health and safety department. After building the enclosure, multiple motion stages, temperature sensors, radiation detectors, and the X-ray generator were assembled inside the enclosure. Numerous programs were developed in National Instrument's LabView software to build a control system. Auxiliary components that were produced include a kinematic mount for the interferometer, an acoustic damping enclosure for the X-ray generator's water chiller, and development was started on a crystal aligner intended for manufacturing interferometers.

2.2.1 X-ray Enclosure

Multiple configurations of the enclosure were brainstormed, ranging from a large shed that would encapsulate an optical table ($2.44 \times 1.25 \times 0.9 \text{ m}^3$) to a multi-chamber design. In the multi-chamber design, one chamber of the enclosure would sit on the optical table and a separate chamber would contain the generator. To guide the design process, some design constraints were ultimately drafted that include:

- the enclosure is large enough to contain the X-ray generator, camera, and interferometer,
- could be assembled on an isolation vibration table,
- has lead lining on all frames and walls,
- no direct path for an X-ray photon to escape,
- contain leaded acrylic windows for monitoring system status,
- and all controllers sit outside the enclosure.

With user safety being of the utmost importance, it is necessary to use enough lead shielding to absorb at least 99.9% of the X-ray radiation. Using the data from Figure 2.5 and applying Equation 2.1, a minimum thickness for 99.9% absorption from a 50 keV X-ray photon is 0.75 mm. With 1.6 mm thick lead sheeting, only a single layer is necessary. The leaded acrylic window is 30% lead by weight and based on the manufacturer's lead equivalency chart, a 22 mm thick window has a lead equivalency of 1 mm [118].

A choice was made to partition the $72 \times 36 \text{ in}^2$ leaded acrylic into four pieces. Two windows for the set of doors and a two more windows for the left and right sides of the enclosure. The left and right windows were cut as $36 \times 22.5 \text{ in}^2$ and the door windows cut

as $26 \times 18 \text{ in}^2$. Rectangular steel window frames were constructed from steel 90° angle, 0.125 inch thick, with 1 inch legs. Welded to the window frames are four adjustable swivel feet that when expanded, secure the window frames inside the door and enclosure frames.

The main enclosure frames are constructed from 2 inch leg, 0.25 inch thick, steel 90° angle. Excluding the window frames, there are nine interlocking steel frames as shown in the Figure 2.12. When assembled (Figure 2.13), the vertical frames are constrained by the top and bottom. These vertical frames were also designed to interlock in a way that prevents them from collapsing inward. To assist in assembly, the top frame has weld-on handles at the front and back. Doors are mounted to the front frame by lift-on hinges.

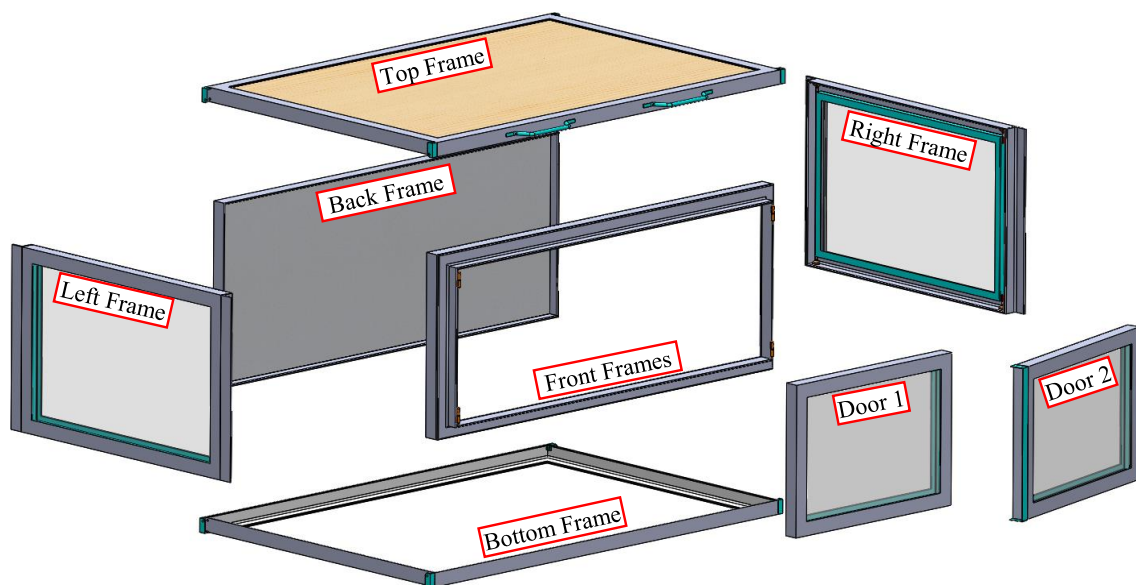


Figure 2.12. The enclosure is constructed from interlocking steel frames, lead, wood, and leaded acrylic. Painted lead lines the major frame components and the wood. Windows are held in place by expanding adjustable feet. The floor of the system is lined with lead and protected by a plastic film.

The lead sheets are adhered to the steel frames and wood panels using high strength Gorilla® brand adhesive. Water-cooling and cables are routed through two large holes cut into the back panel with a lead-lined plastic ‘dryer hood’ vent fitted. Additional safety

measures include covering all exposed lead with white paint for identification, a lock for the doors, a lead sheet covered by a plastic film for the floor of the enclosure, and four tie-rods to fasten the top and bottom frame together. Overall outer dimensions of the enclosure are 1.7 m wide, 1.2 m deep, and 0.7 m tall. Powder coating the steel frames and bluing the tie-rods was implemented to protect the frames from corrosion.

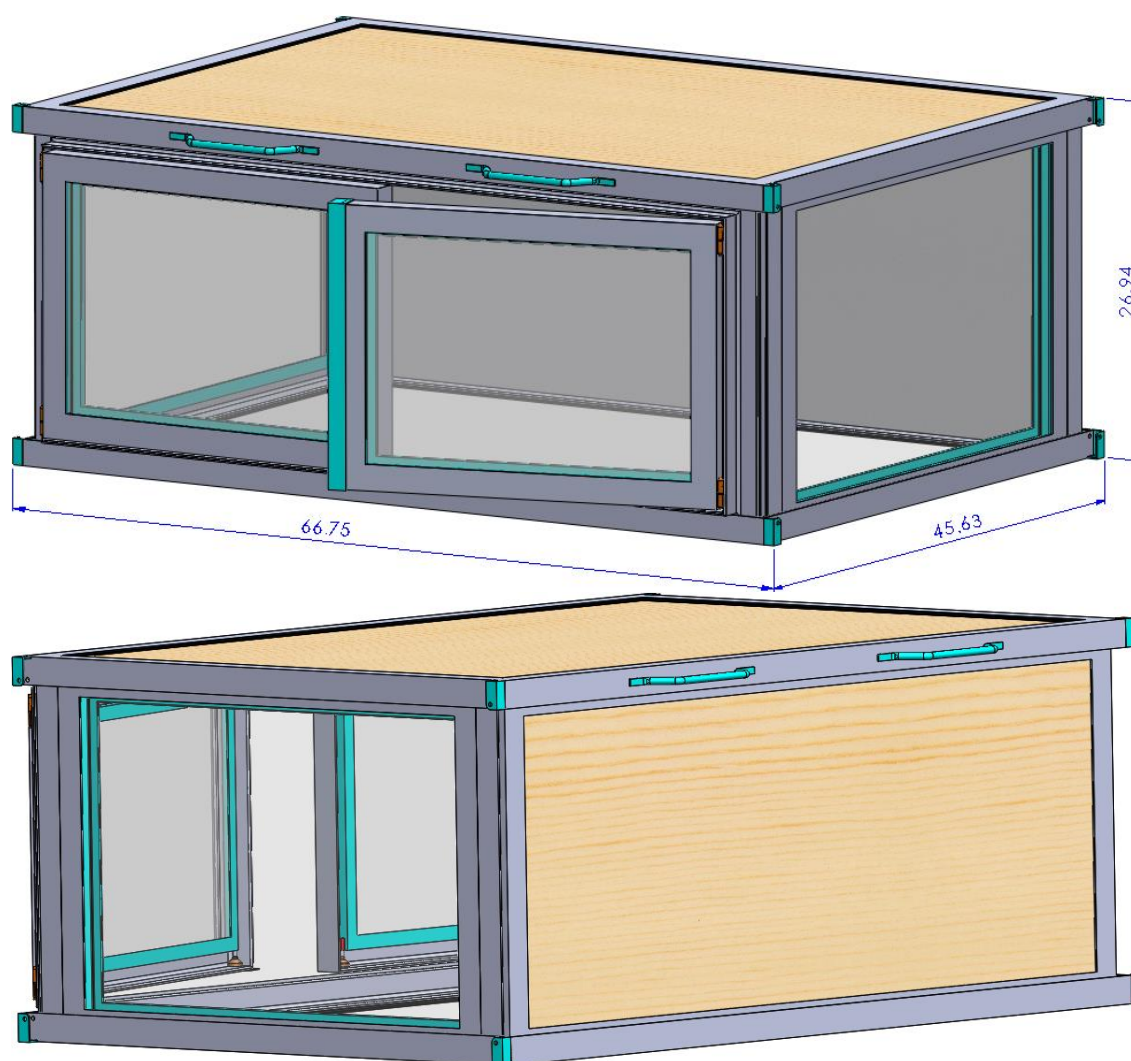


Figure 2.13. Computer aid drawing of the X-ray enclosure with dimensions in inches.

2.2.2 X-ray System and Control

Within the enclosure are the X-ray generator, multiple linear and rotary motion stages, and a variety of detectors. The system can be broken into three subsystems, starting from left to right, these subsystems are the source, specimen, and detector. Although the facility is designed to be highly reconfigurable, order of the subsystems will not change. Reconfiguration is primarily limited to the specimen subsystem where the item of interest, such as the interferometer, is positioned. Position of the subsystems is shown in Figure 2.14, with the specimen subsystem setup for interferometry.

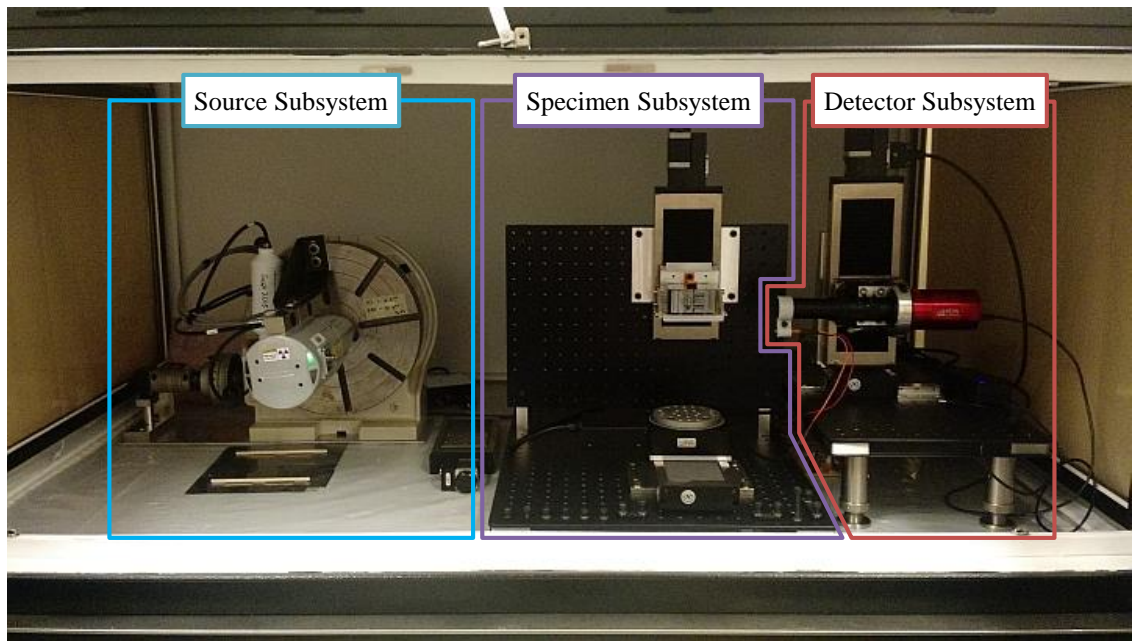


Figure 2.14. Configuration of components for X-ray interferometry studies.

2.2.2.1 Source Subsystem

On the left side of the enclosure is the source subsystem (Figure 2.15). The X-ray generator is mounted to a motorized rotary table for control of the incident beam angle. Instead of rotating the interferometer, the X-ray generator is rotated until diffraction.

Automation of the rotary table is provided by a NEMA-17 stepper motor with a 90:1 gearbox to increase the torque and angular resolution. A full rotation of the stepper motor yields 0.04° of rotation of the rotary table. Backlash in the rotary table was measured as approximately 0.1° or 2.5 rotations.

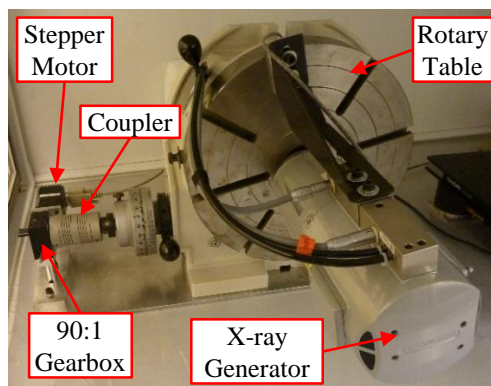


Figure 2.15. Components of the source subsystem.

The Bede Scientific X-ray generator is a microsource that operates similar to the design described in Section 2.1. An electron beam is generated inside a vacuum tube and is focused by an electromagnetic lens. The target of the e-beam is a piece of molybdenum located near a beryllium window. These elements are shown Figure 2.16. When the generator is assembled, the e-beam tube slides into the tube port. Control of the electromagnetic lens is handled by a control box (not shown) with options to adjustments focus, stigmatism, horizontal deflection, vertical deflection, and two different aligns.

A couple of add-ons made to mount to the X-ray generator are a laser aligner and filter (Figure 2.17). Aligning components is difficult without some way to see where the X-ray beam travels. To make alignment easier and faster, one of the earliest add-ons was a tip-tilt flexure with laser that slides onto the capillary optic. Laser diodes are not guaranteed to output co-axial to their housing. To account for the off-axis alignment, the laser housing

is installed in the tip-tilt flexure. When the flexure is slid onto the capillary optic, the flexure can be adjusted using the four screws until the laser beam is co-axial to the X-ray beam. The second add-on is an X-ray filter built from a printed plastic case with a filter window. For attenuating the bremsstrahlung with photon energies above 20 keV, a 25 μm thick molybdenum sheet is installed as one filter. Another material recommended as a filter for molybdenum targets is zirconium [119]. Unlike a molybdenum filter with the K-edge at 20 keV, a zirconium filter's K-edge is 17.997 keV and would significantly filter the K_β characteristic radiation.

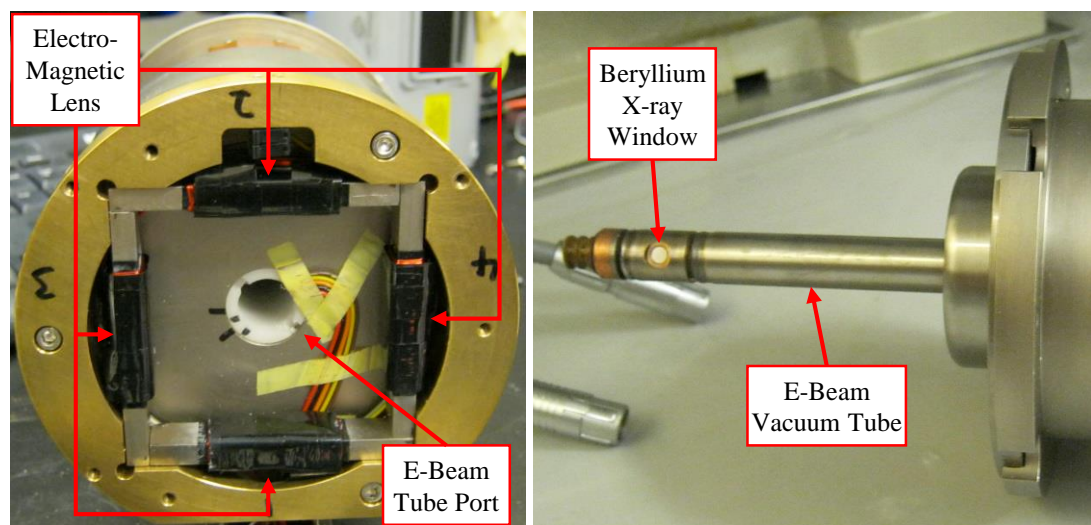


Figure 2.16. Configuration of the X-ray generator. When assembled, the vacuum tube (right) slides into the port (left).

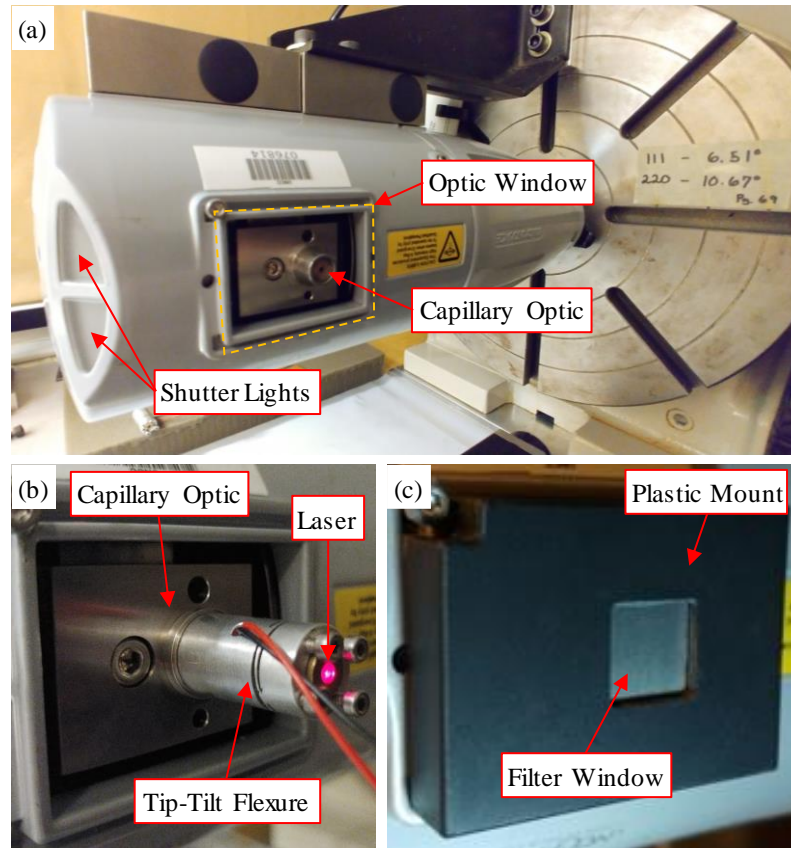


Figure 2.17. The (a) X-ray generator and its capillary optic for beam focusing. Add-ons are (b) a laser aligner that slides onto the capillary optic and (b) filters that snap onto the optic window.

2.2.2.2 Specimen Subsystem

In the specimen subsystem are two optical breadboards, motorized stages, and a specimen holder. For interferometry experiments, this subsystem includes the interferometer and interferometer mount. On the breadboard table, there can be multiple linear stages that move a rotary stage with a specimen mounted in a holder into and out of the beam. With one setup, by rotating a specimen in small angle increments and taking X-ray pictures, we can perform computed tomography imaging. Also attached to the breadboard are mounts that hold a breadboard wall. Mounted to breadboard wall is a linear stage that moves up and down. On this stage, one setup would move the kinematic mount

with the X-ray interferometer. Another setup is one for reflectivity studies that includes a pinhole aperture for reducing the beam spot size.

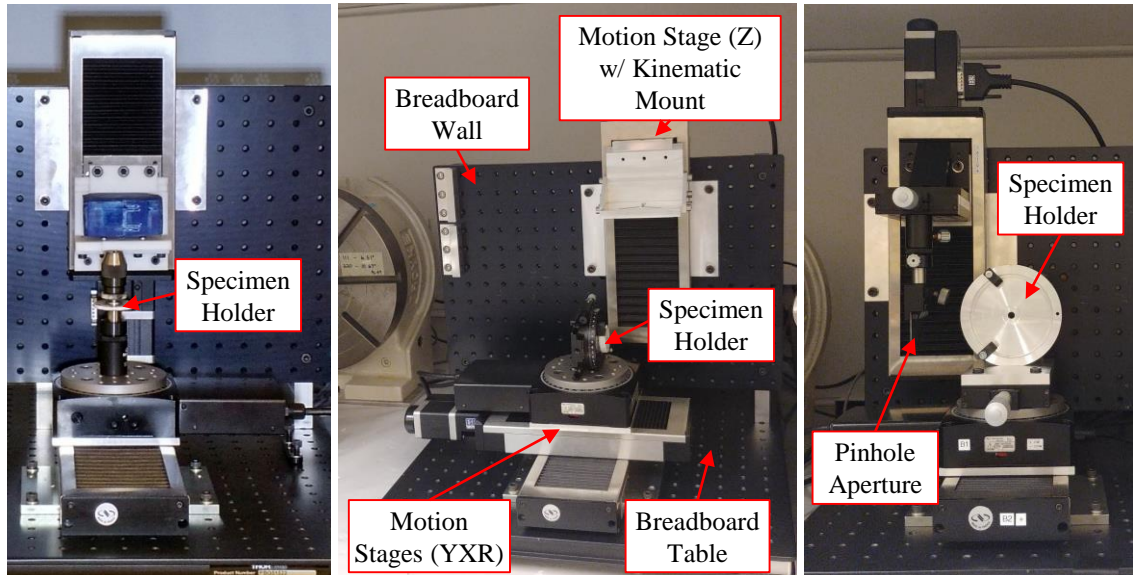


Figure 2.18. Multiple component setups of the specimen subsystem. Left is for interferometry, and the middle and right setups are for reflectivity.

As the silicon interferometer is extremely fragile, its mount is designed to provide the following features:

- repeatable kinematic placement,
- a method for securing the interferometer,
- the interferometer is removable without tools,
- and the mount can be translated longitudinally (i.e. left and right).

The first attempt at a design was a tongue-and-groove single-clamp. After prototyping with printed plastic parts, a problem was discovered early on when using the single-clamp design. As the clamp was tightened, the tongue and groove acted as a pivot

point that pushed the bottom of the mount outwards. This failed design is shown as Figure A.1 in Appendix A.

Building upon that failure, the second design used clamp brackets at the top and bottom [120]. Testing the printed prototype was a success and the design was remanufactured from aluminum (Figure 2.19). With this new design, instead of pulling the vertical plate towards the stage, the top clamp uses three set screws with conical ends to push the vertical plate downwards and inwards into an identical bottom clamp. The same set screws were added to the bottom bracket and rotational leveling to the mount. On the vertical plate there are three threaded holes that can be used to secure the interferometer with a copper leaf spring or for mounting auxiliary items. An example of an auxiliary item would be an actuator for the interferometer's flexure. Hole tolerances on the side plates are enough to provide leveling for the mounting plate. Repeatability is achieved by the three v-grooves cut into the mounting plated where truncated steel spheres epoxied to the bottom of the interferometer will sit.

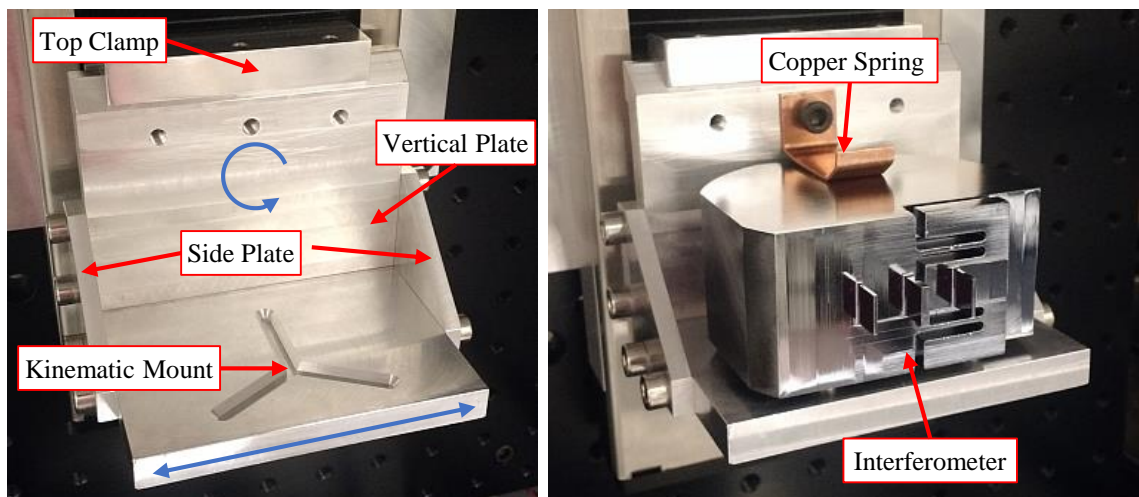


Figure 2.19. V-groove kinematic mount for the X-ray interferometer.

The X-ray interferometer (Figure 2.20) is constructed from a monolithic block of single crystal silicon. Machined into the silicon by way of creep feed grinding are the mirror, splitter, and analyzer blades. Blades were cut with respect to the [111] and [220] directions. Thickness of the splitter and analyzer blades are 1 mm while the mirror is 2 mm. Remaining dimensions of the blade are 8 mm along the [220] and 10 mm along the [111] planes. Spacing between the blades is 10 mm. Translation of the analyzer blade is accomplished with a leaf spring flexure.

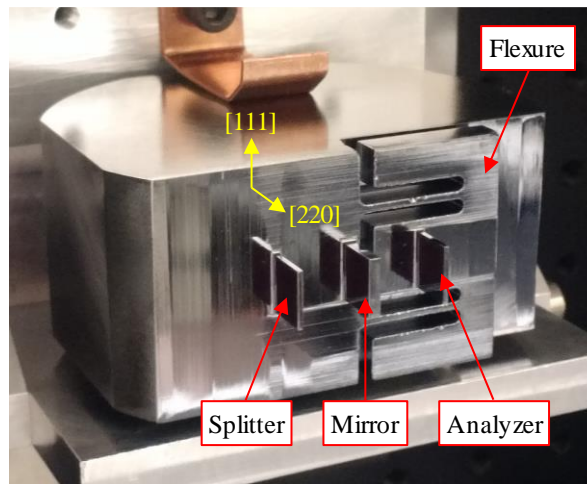


Figure 2.20. Single crystal X-ray interferometer.

2.2.2.3 Detector Subsystem

The third and final subsystem provides mounts and alignments for the detector (Figure 2.21) and is located at the right side of the enclosure. Mounted to a breadboard table are two stages in a YZ configuration. On the Z-axis stage is a mount for the X-ray camera. A turning mirror protects the camera's sensor by removing the camera from the X-ray beam's path. Conveniently located on the bottom of the turning mirror is a threaded

hole. This was the perfect mounting location for the Geiger-Müller detector as it makes switching between the two detectors effortless.

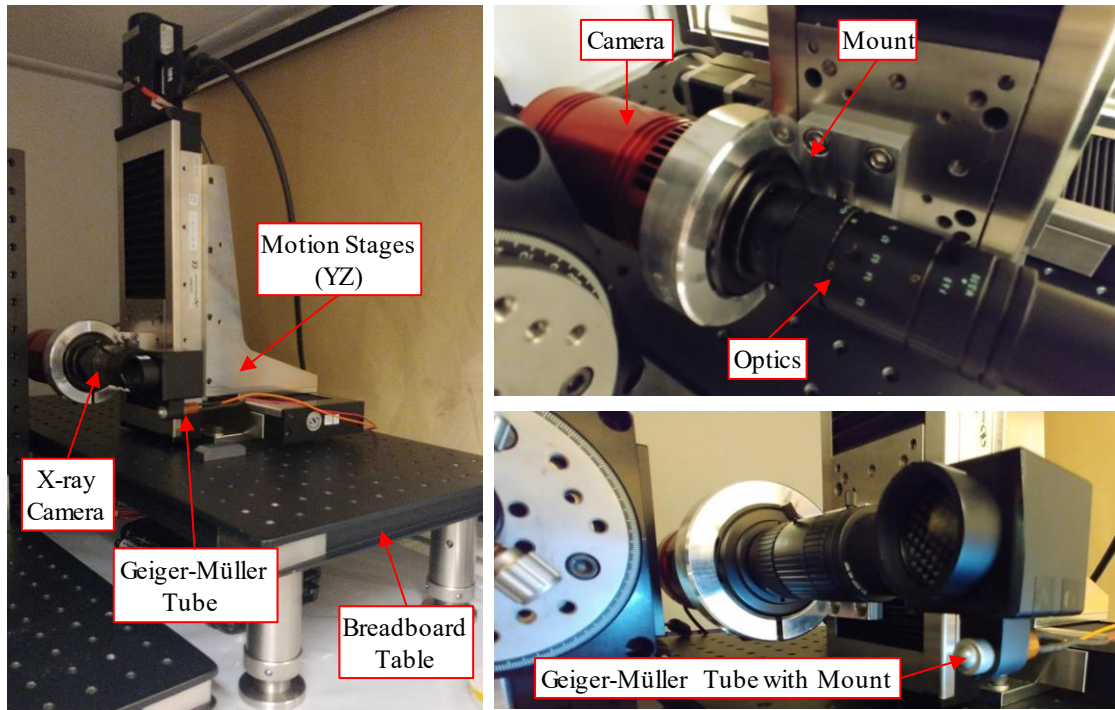


Figure 2.21. Components of the camera and GM tube subsystem.

2.2.2.4 System Controls

The brain of the system is a host computer running Microsoft's Windows 7 operating system where the system's controls are programmed using National Instruments' LabView software. The host computer directly connects to the X-ray microcontroller, an NI CompactRIO, and the X-ray camera. The CompactRIO (model cRIO-9022) has a 533 MHz real-time processor and attaches to an 8-slot chassis (model cRIO-9114). Included with the chassis is a Xilinx Virtex-5 LX50 FPGA. Inserted into the chassis are digital I/O, analog in, analog out, and serial communication modules. Module models, their function, number of channels, performance, and quantity are listed in Table 2.1.

Modules currently in use are one NI-9215 that reads four LM35 temperature sensors, an NI-9401 for the Geiger-Müller counter, and a NI-9870 for serial communication with the rotary table's stepper motor and Newport® motion stages. Unfortunately, the X-ray microsource controller cannot be connected to the CompactRIO and must be programmed to use Microsoft's ActiveX framework from LabView to work with the manufacturer's software. Newport® stages are connected to two model ESP301 motion controllers. Each ESP301 can control up to three stages. One controller manages the three stages of the specimen subsystem and the other manages the detector subsystem's two stages. For a connection schematic, refer to Figure 2.22. Not included in the schematic are the interlocks for the X-ray microsource's water cooling flow monitor and the enclosure's safety switches monitored by the X-ray controller. The cRIO's FPGA is utilized in this system to communicate via serial to the ESP301 controllers, read the temperature sensors, and read the Geiger-Müller tubes.

Table 2.1. CompactRIO chassis modules.

Model	Function	Channels	Performance	Qty.
NI-9263	Voltage Output	4	± 10 V, 16-Bit, $100 \text{ kS} \cdot \text{s}^{-1} \cdot \text{ch}^{-1}$	1
NI-9215	Voltage Input	4	± 10 V, 16-Bit, $100 \text{ kS} \cdot \text{s}^{-1} \cdot \text{ch}^{-1}$	2
NI-9423	Digital	8	24 V, 100 μs sinking input only (I)	1
NI-9401	Digital	8	5 V/TTL, 100 ns, bi-directional (I/O)	1
NI-9870	Serial	4	14 bit·s ⁻¹ to 921.6 kbit·s ⁻¹ baud rate	1

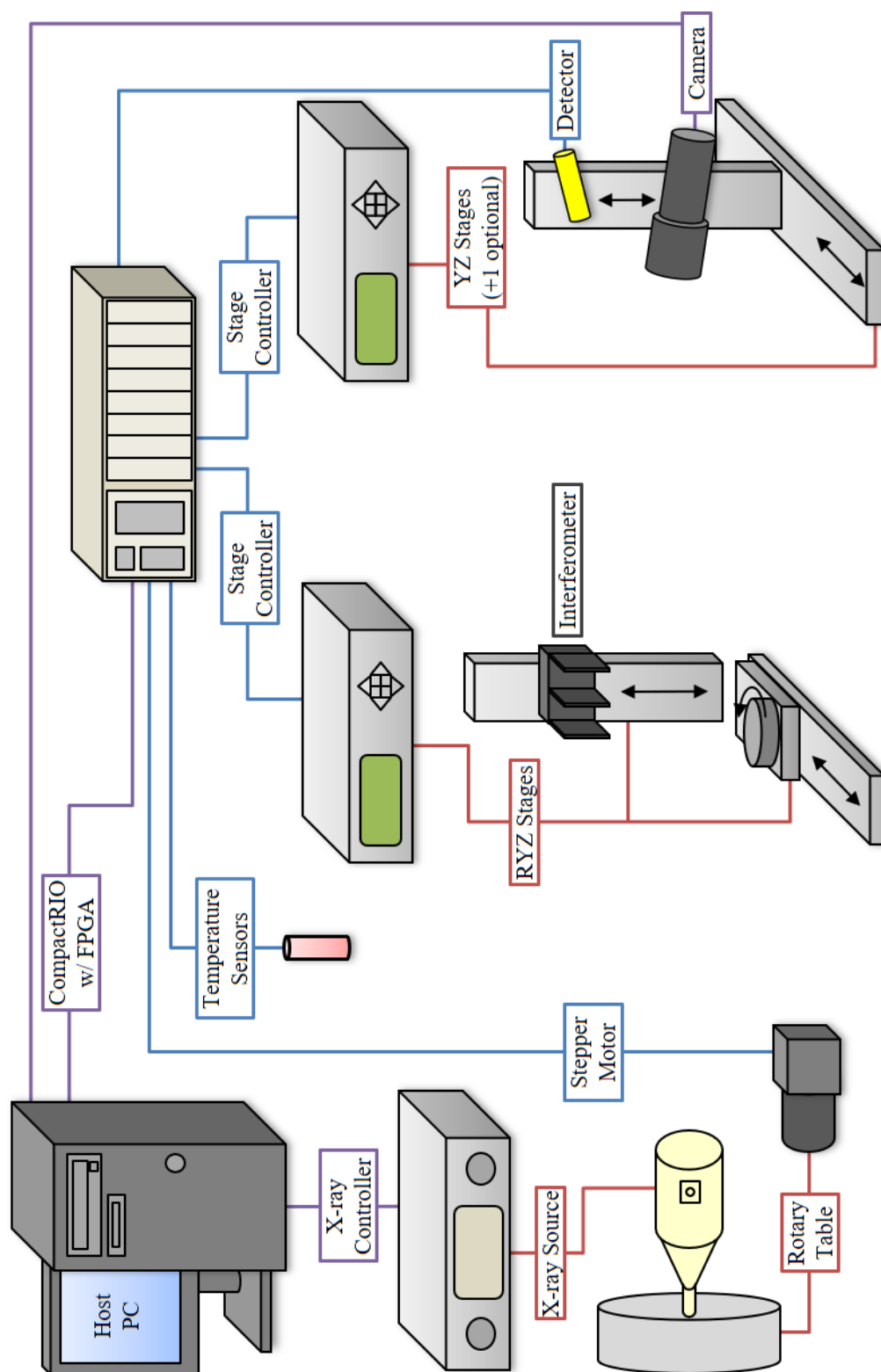


Figure 2.22. X-ray facility connection schematic configured for X-ray interferometry.

2.2.3 Water Chiller Acoustic Damping Enclosure

Originally the water chiller was located next to the isolation table with the X-ray system due to limited lab space. This setup leads to two problems, heat and noise. Knowing that majority of the noise comes from the chiller's fan and pump, their noise could be reduced by placing the chiller inside an acoustic damping enclosure (Figure 2.23). The enclosure measures $51 \times 27 \times 36.5 \text{ in}^3$. It is constructed from 1 inch square T-slot aluminum extrusions and bolted to the outside of the enclosure are drywall panels with a viscoelastic polymer for damping (QuietRock, model 500). A set of 9.5 inch panels are used for the front and back faces with an 8 inch gap between the panels. The gap between the panels provides access for the water lines, the power switch, and allows for air flow. Inside the enclosure is sound damping foam adhered to the panels and interlocking foam mats for the flooring. Another source of noise comes from the vibration of thin sheet metal cosmetic panels and the distilled water tank's lid. To further reduce noise, the sheet metal panels were removed, and mass was added to the tank's lid. Measuring the damping performance, a Radioshack® sound decibel meter (model 33-4050) was placed 1 m from the side of chiller. Without the enclosure, the sound level is 67 dB and by adding the enclosure it is reduced to 59 dB. For perspective, that is the difference between a household vacuum cleaner and a conversation between people 1 m apart. The chiller and its enclosure has since been relocated to an adjoining lab.

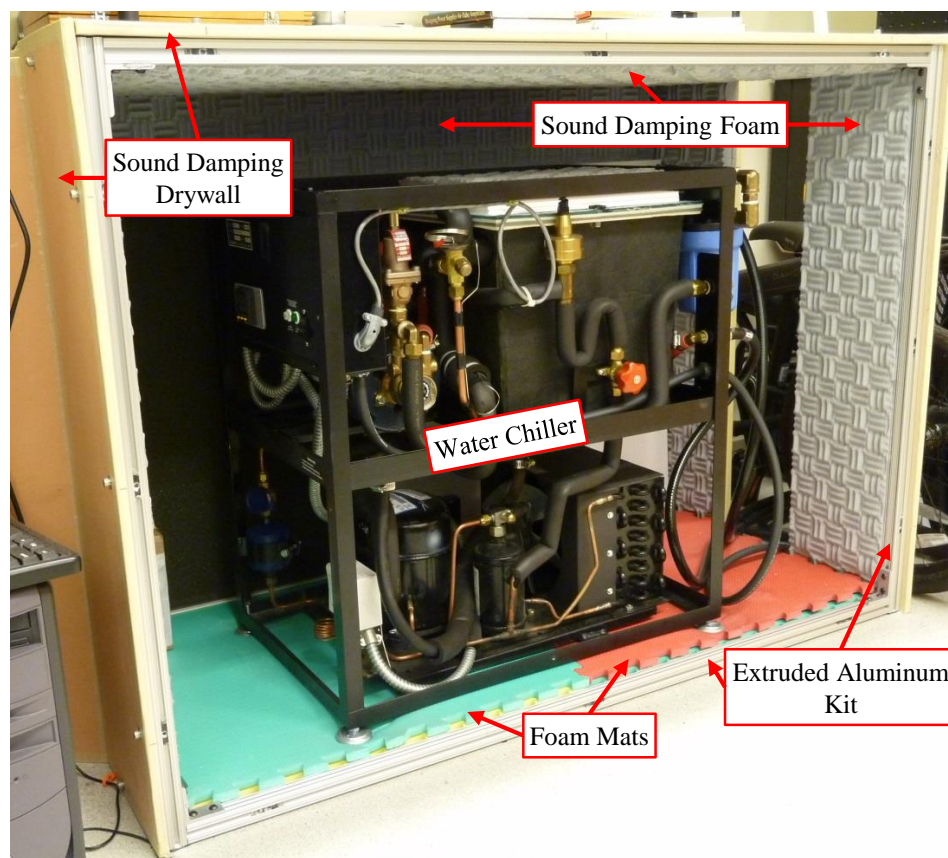


Figure 2.23. Water chiller and sound damping enclosure.

2.2.4 Crystal Aligner

A desired capability of the X-ray facility is to one day manufacture interferometers of different crystal orientations and materials. For manufacturing to be a possibility, the crystal aligner is shown in Figure 2.24 (constructed with the help of Tyler Harper).

The aligner functions by mounting a crystal blank onto the crystal holder. It is critical that when mounted, the axis of the precision cylinder is perpendicular to within a couple degrees of the desired crystalline plane for diffraction. The aligner is meant to refine the angle and therefore its range is limited. The next step is to position the aligner until the X-ray diffracts off the crystal. When diffraction is achieved, the X-ray camera or Geiger-Müller tube records the diffracted X-ray's intensity.

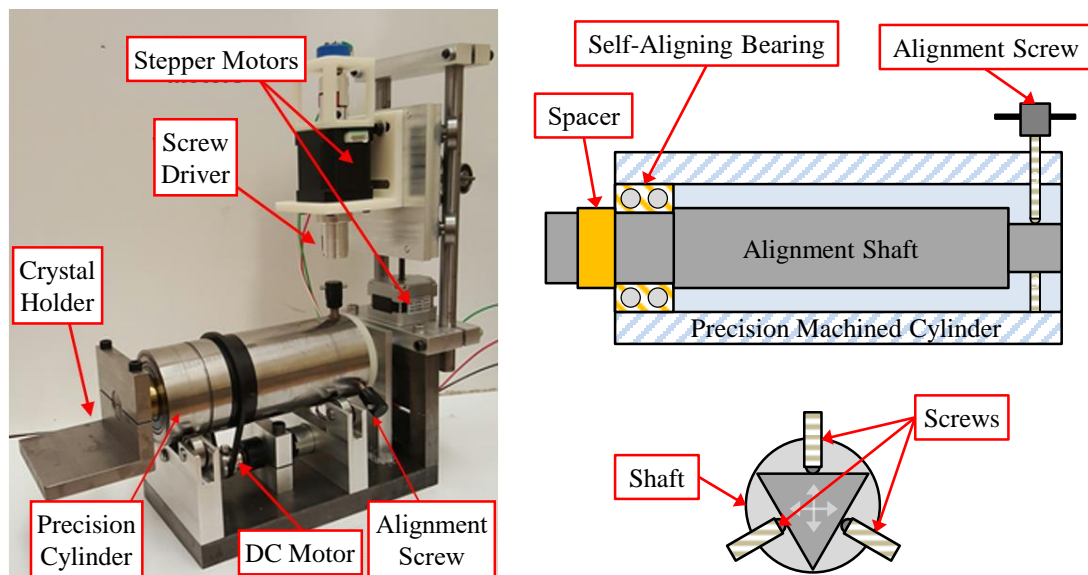


Figure 2.24. Crystal aligner design for interferometer manufacturing.

The precision cylinder sits on four bearings so that it can be smoothly rotated. A DC motor is attached to the precision cylinder with a rubber belt and as it rotates, a potentiometer tracks the cylinder's rotation. By recording the diffracted X-ray intensity relative to cylinder position, an oscillation in X-ray intensity indicates a misaligned crystal. The goal is to adjust the crystal's angle until the X-ray intensity is constant. To adjust the crystal's angle, the holder is affixed to an alignment shaft that runs through a self-aligning bearing to the back of the precision cylinder. At the back, the shaft has been machined into an equilateral triangle. Connected to the precision cylinder are three fine-threaded alignment screws holding the alignment shaft in place. These screws are used to adjust the crystal's angle.

In the initial design, a screwdriver engages an adjustment screw by moving up and down with a motorized linear stage. An algorithm determines the order of the screws, direction, and number of turns. With the alignment shaft length at 5.5 inches long and the

pitch of the adjustment screw at $80 \text{ thread} \cdot \text{in}^{-1}$, a standard NEMA 17 stepper motor with 200 steps yields a theoretical angular adjustment resolution of $0.001^\circ \cdot \text{step}^{-1}$.

CHAPTER 3: FACILITY PERFORMANCE STUDY

System monitoring is integral in understanding the performance and limitations. Monitoring occurs via temperature sensors and X-ray detectors, such as Geiger-Müller counters and X-ray cameras. This chapter discusses the performance of the sensors, detectors, and the X-ray generator, as well as the displaying results from diffraction and reflection experiments.

3.1 Thermal Sensors and Stability

Temperature is one of the most important physical quantities in studying performance for precision systems. Determining which type of temperature sensor to implement is based on range, accuracy, and cost. Operating a temperature-controlled lab, we can assume the temperature will not go below 20 °C. However, the maximum temperature around a system is influenced by heat generated from all the electronics, primarily the x-ray source. It is assumed that the combined energy output of all the components will not elevate the temperature beyond 50 °C. At least four temperature sensors, with an option to add up to four more, can be placed inside and outside the enclosure. One setup could be to monitor the temperature in each of the three subsystems and additional monitoring outside the enclosure.

The LM35 is an integrated-circuit temperature sensor that has been precision trimmed. Technical specifications for the LM35 are an accuracy to within 0.5 °C at 25 °C, non-linearity of ± 0.25 °C, operating range from -55 °C to 150 °C, and 10 mV·°C⁻¹ output. The voltage is read by the CompactRIO NI-9215 module at 0.3 mV·bit⁻¹, which places the resolution at 0.03 °C. Input noise of the module is 1.2 LSB_{rms} or 0.036 °C. A capacitor to

filter high frequency noise and a non-inverting operation amplifier with $10\times$ gain are connected to the LM35 sensors as shown in Figure 3.1.

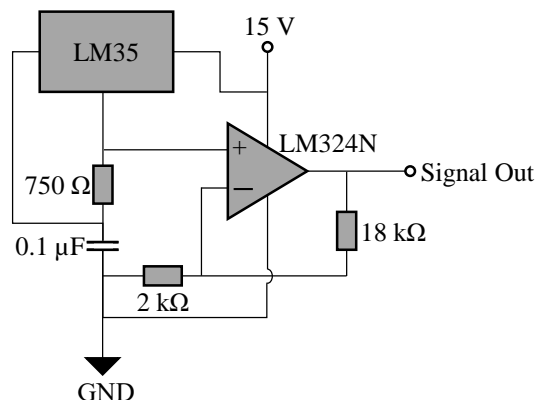


Figure 3.1. Temperature sensor circuit diagram.

Previous monitoring of the lab space showed an oscillation in temperature. The cause of this oscillation is the controller for the air conditioning system, referred to as a bang-bang or on-off controller. Three temperature sensors were placed inside the x-ray enclosure and the fourth outside the chamber. Temperature was recorded every minute for 1500 min. Each reading is the average of 64 measurements that were taken with a 1 ms interval. In Figure 3.2, the average of the three enclosure sensors is compared to the lab's temperature. Although the oscillation is present in the lab, the enclosure acts as a low pass filter. The cause of the temperature dropping from 400 min to 600 min was not determined.

As previously mentioned, the sensors can be strategically placed to record the three subsystems. Shortly after recording was started, the X-ray generator was turned on to 50 kV and 0.5 mA with the shutter opened. As seen in Figure 3.3, within the first 5 min the temperature near the x-ray source increased by nearly 1°C , and near the specimen and camera by 0.5°C . From 5 min to 80 min, temperature increases more gradually. At 80 min,

the shutter was closed, and the X-ray generator turned off. Drops in temperature at 100 min and 112 min were caused by opening and closing the enclosure doors. Similar results were seen during a previous study with a commercial X-ray system [83].

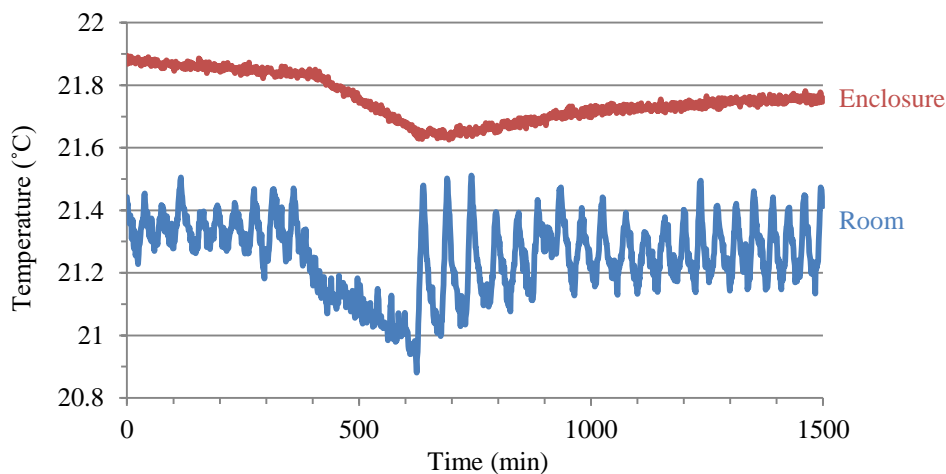


Figure 3.2. Temperature comparison between the enclosure and lab. The enclosure filters the lab's high frequency temperature oscillations.

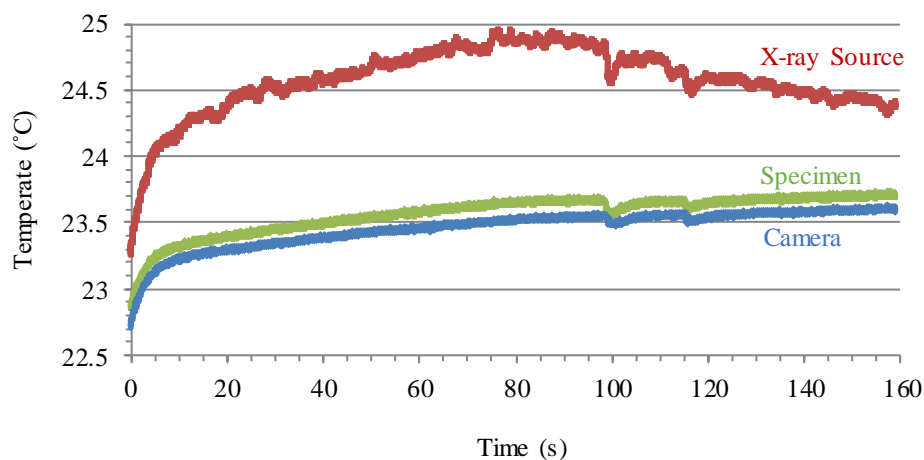


Figure 3.3. Temperature readings within the enclosure showing increased heating around the X-ray source when turned on. The X-ray was turned off after 80 min. Opening the enclosure caused the drops at 100 min and 112 min.

3.2 Radiation Detectors

Different detectors are available depending on the needs of the user. In cases where a single numerical output is preferred, there is choice between a commercial ion chamber meter calibrated to output equivalent dose or a Geiger-Müller tube for counts·s⁻¹. For imaging, a Bede Scientific camera was first used, however this was replaced with a custom-built camera. The following sections explore the performance of these detectors.

3.2.1 Commercial Ion Chamber Meter

The Fluke Biomedical, model 451B-RYR, ion chamber survey meter is a commercial radiation meter displayed in Figure 3.4. The meter is designed to read beta radiation above 100 keV, and gamma and X-ray radiation above 7 keV. Dosage readings are given in Roentgen equivalent man (rem) or Sievert (Sv) with a maximum operating range of 500 mSv·h⁻¹. The time response is dependent on the operating range, with a fastest response of 2 s for 0 mSv/h to 500 mSv/h. The detector's slowest response of 8 s is for the range 0 to 50 µSv/h. Precision is given as 10% of the reading [121].

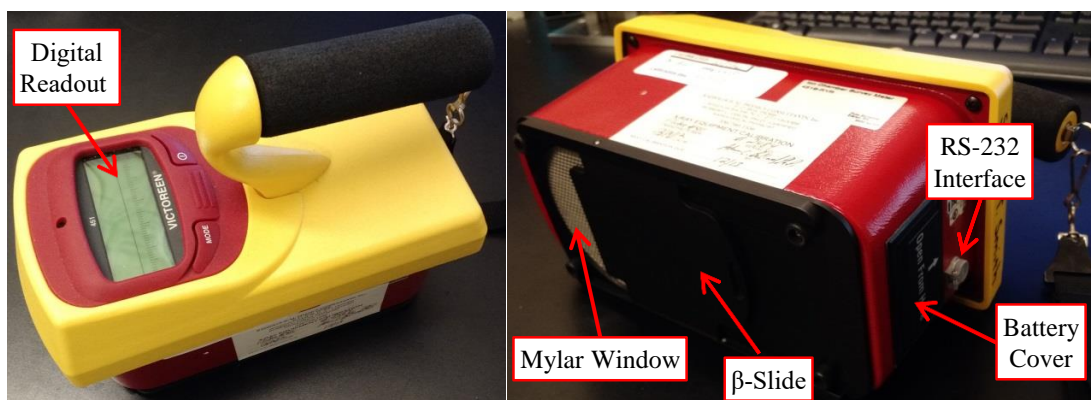


Figure 3.4. Fluke Biomedical 451B-RYR ion chamber survey meter.

To test the X-ray source's maximum output and calculate the time constant of the detector, the detector was connected to the computer using an RS-232 cable. Data was logged by an add-on script in Microsoft® Excel. The beta window was placed in the open position with the window facing towards the generator approximately 400 mm from the generator. Data was logged every second for 500 s. The first 40 s were recorded with the X-ray generator turned on and with the X-ray shutter closed. When the shutter was opened, a rapid increase in dosage is seen within 10 s, followed by a slow increase to $320 \mu\text{Sv}\cdot\text{h}^{-1}$. To estimate the detector's response time, the shutter was closed after 450 s and reopened about 60 s later.

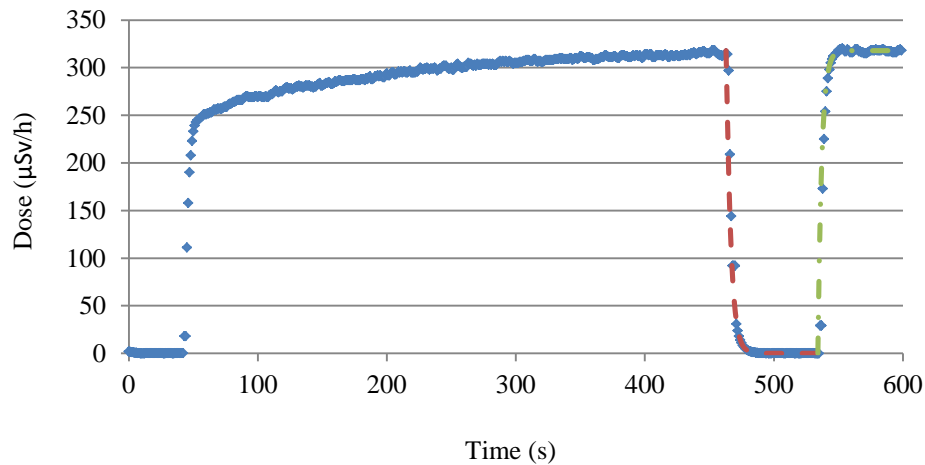


Figure 3.5. Indicated dose by the commercial dose meter to calculate the detector response time constants. X-ray shutter was closed at 463 min to measure the decay time constant and opened at 534 min to measure the rise time constant.

The detector's data is plotted in Figure 3.5. For actual dose, the indicated dose can be divided by approximately 0.82 per the 451B meter's user manual. Decay time of the detector is calculated from Equation (3.1) and rise time is calculated from Equation (3.2). Closing the shutter yields a decay time constant (τ) of 3.5 s, amplitude (A) of $318 \mu\text{Sv}\cdot\text{hr}^{-1}$, and an offset time (t_0) of 463 s. When the X-ray shutter was reopened, the rise time

constant is 3 s, amplitude 318 $\mu\text{Sv}\cdot\text{h}^{-1}$, and offset time 534 s. Taking the larger of the two values yields a conservative estimate of the time constant at 3.5 s.

$$y = Ae^{\left(\frac{t_0-t}{\tau}\right)} \quad (3.1)$$

$$y = A\left(1 - e^{\left(\frac{t_0-t}{\tau}\right)}\right) \quad (3.2)$$

3.2.2 Geiger-Müller Tube

Without having the ability to adjust the settings of the ion chamber detector, it was decided to build a detector with a faster response time. For the detector a Geiger-Müller tube was chosen. The simplest circuit to operate and sense X-rays with a Geiger-Müller tube is shown in Figure 3.7. It consists of a high voltage power supply connected through resistor R1 to the tube's anode, and the cathode connects to ground through resistor R2. Output is measured across resistor R2 as a series of voltage spikes.

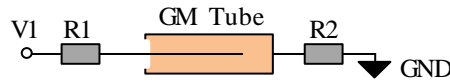


Figure 3.6. Simplified Geiger-Müller circuit.

To understand the impact resistor R1 and R2 contribute towards the voltage spike, different resistances were tested. Using a Victoreen model 6306 organic Geiger-Müller tube with 950 V supplied, resistor R1 was tested at 8 and 10 $\text{M}\Omega$, while resistor R2 was set as a 1 $\text{M}\Omega$ resistor. From Figure 3.7, an increase in the R1 resistance reduced the peak voltage from 1.2 V to 1 V, but provided a more consistent peak value. Looking at the voltage spike at 5 ms, the decay time constant is indiscernible.

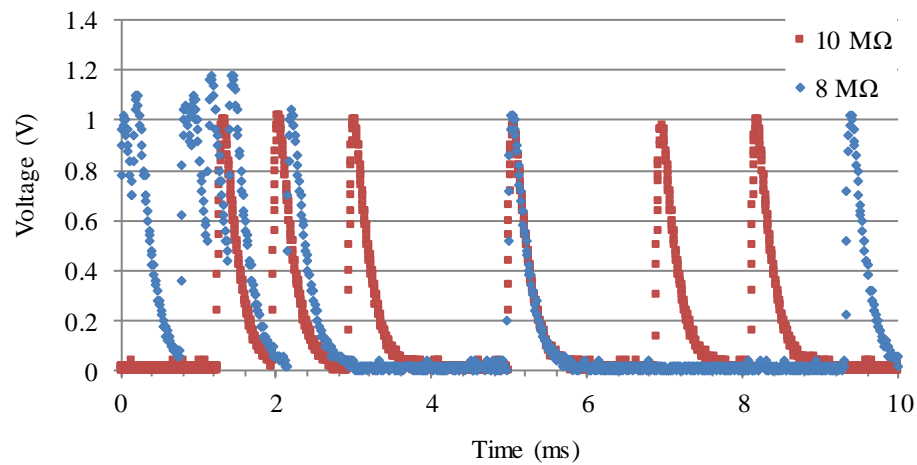


Figure 3.7. Effect of different R1 resistances on the voltage spike.

Setting resistor R1 to 10 MΩ, resistor R2 was tested with resistances of 250 kΩ, 500 kΩ, and 1000 kΩ. Following Ohm's law, an increase in resistance should yield an increase in voltage. Peak voltages were measured as 0.65 V for 250 kΩ, 0.84 V for 500 kΩ, and 1.0 V for 1000 kΩ (Figure 3.8). As for the decay time constant, Figure 3.9 shows that an increase in resistance reduces the time constant. The resultant time constants are 0.11 ms for 250 kΩ, 0.15 ms for 500 kΩ, and 0.2 ms for 1000 kΩ.

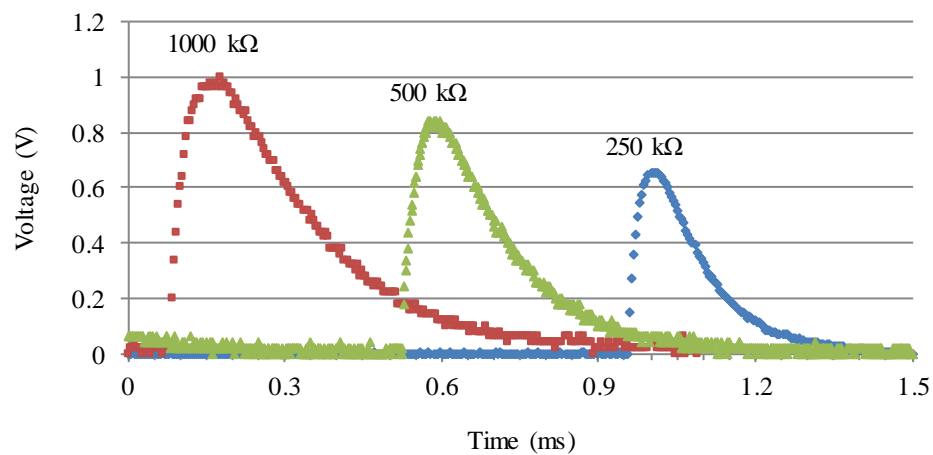


Figure 3.8. Effect of the Geiger-Müller circuit's R2 resistance on the peak voltage.

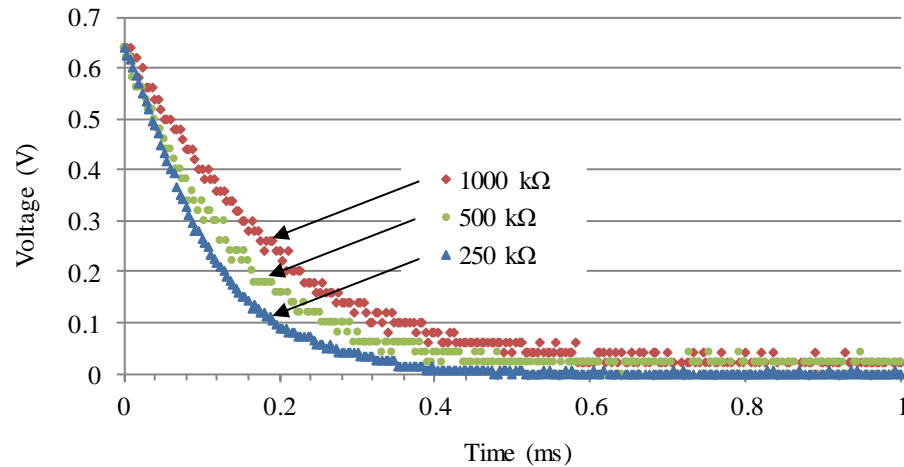


Figure 3.9. Effect of the Geiger-Müller circuit's R_2 resistance on the decay time constant.

The next issue dealt with was handling the conversion of the voltage spikes into a 5 V TTL signal that can be read by the CompactRIO's digital I/O frequency counter. The first idea was to tie the Geiger-Müller output signal into a monostable 555 circuit for minimum pulse width control. However, a monostable 555 circuit triggers when the input signal is pulled low, which is opposite of the Geiger-Müller's voltage spike. To remedy the inverted input trigger issue, a LM339N quad comparator was added before the monostable 555 stable to act as a signal inverter. This also enabled setting of a voltage threshold, activating only when the voltage spike exceeds a specified voltage. A diagram of the finished Geiger-Müller circuit can be seen in Figure 3.10.

As discussed, the monostable 555 circuit operates when the input voltage is pulled low. This occurs by causing the discharge pin (pin 7) to deactivate and the capacitor C_1 to charge. To calculate the minimum pulse width, Equation 3.3 is a mathematical model of a charging capacitor. When the charge in capacitor reaches two-thirds of the source voltage, Equation 3.4, the threshold pin (pin 6) is enabled to discharge the capacitor. The time t is

solved as shown in Equation 3.5 to Equation 3.8 by substituting the time constant τ with the product of resistor R5 and capacitor C1 [122]. In the circuit, R5 and C1 were set to 3.9 k Ω and 7 nF, yielding a minimum pulse time of 30 ms.

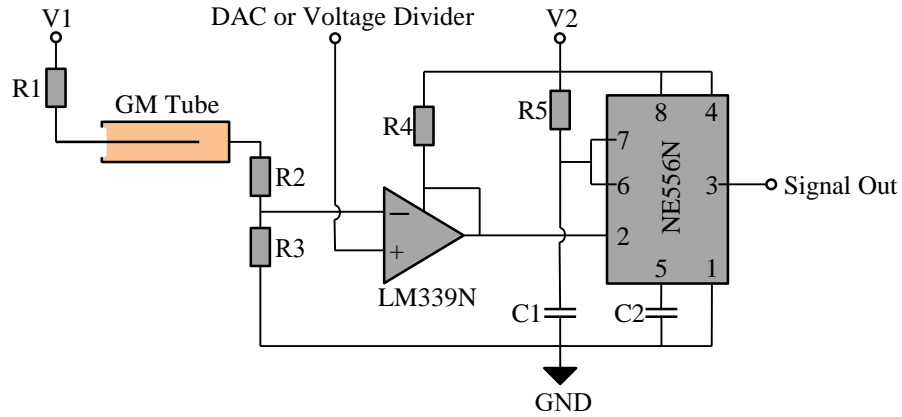


Figure 3.10. Geiger-Müller circuit diagram to convert voltage spikes into a TTL signal.

$$V_c = V_s \left(1 - e^{\left(\frac{-t}{\tau}\right)} \right) \quad (3.3)$$

$$\frac{V_c}{V_s} = \frac{2}{3} = 1 - e^{\frac{-t}{\tau}} \quad (3.4)$$

$$\frac{1}{3} = e^{\left(\frac{-t}{\tau}\right)} \quad (3.5)$$

$$\ln \frac{1}{3} = \ln 1 - \ln 3 = \frac{-t}{\tau} \quad (3.6)$$

$$\ln 3 = \frac{t}{\tau} = \frac{t}{R_5 C_1} \quad (3.7)$$

$$t = R_5 C_1 \ln(3) \cong 1.1 R_5 C_1 \quad (3.8)$$

To find the optimal threshold voltage for the comparator, analog out on the CompactRIO was connected into the LM339N's noninverting pin. Operating the X-ray

source at 40 kV and 0.5 mA and with the tube voltage at 950 V, the threshold voltage was swept from 0.1 V to 0.5 V to record the counts·s⁻¹. At the start, the highest count rate was around 3200 counts·s⁻¹ at 0.3 V. To assure confidence in the threshold voltage value, multiple tests were performed. A problematic pattern emerged showing a continuous drop in the threshold voltage to obtain peak output as seen in Figure 3.11. Investigating the properties of the Victoreen 6306 tube, it was discovered that the tube is filled with argon and an organic quench. Over time, the interactions within the tube break down the organic quench until the tube has reached an end of life. This was the cause of the voltage threshold decay and the Victoreen 6306 had to be replaced with an inorganic tube.

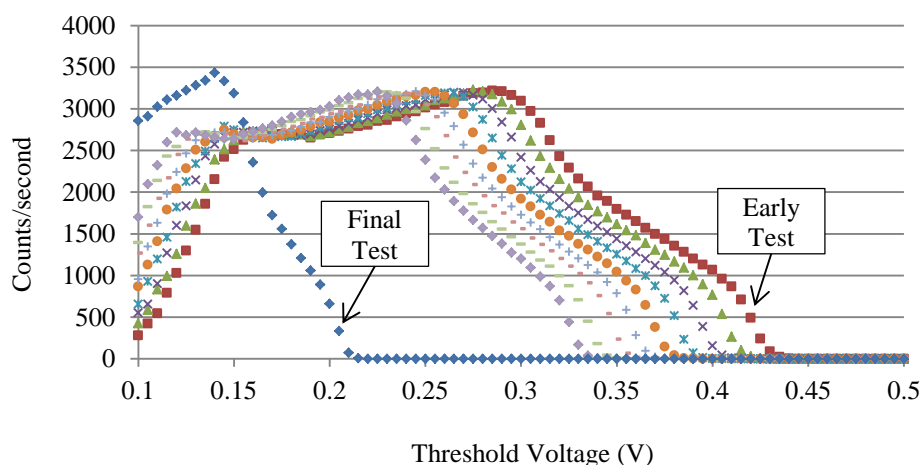


Figure 3.11. Graph of the count rate compared to the comparator's threshold voltage. The voltage decreased in subsequent tests as the organic quench of the Victoreen 6306 tube was destroyed.

Switching to an inorganic tube (manufacturer and model unknown), the source voltage was reduced to 400 V. Resistance for R2 was retested while R1 was left as 10 MΩ. With a 500 kΩ resistor, the voltage spiked exceeded 5.5 V with a decay time constant of 0.18 ms. For a 250 kΩ resistor, the output was under 4 V with a time constant of 0.08 ms. For the threshold test with the inorganic tube, the CompactRIO analog output was swept

from 0.2 V to 1.9 V. Results of this sweep are shown in Figure 3.12 for different X-ray power settings. Between 0.8 V and 1.1 V produced consistent outputs regardless of X-ray generator power. Therefore, for simplicity a 1 V threshold was chosen. The components used in the finished board are provided in Table 3.1 with additional photographs in the appendix.

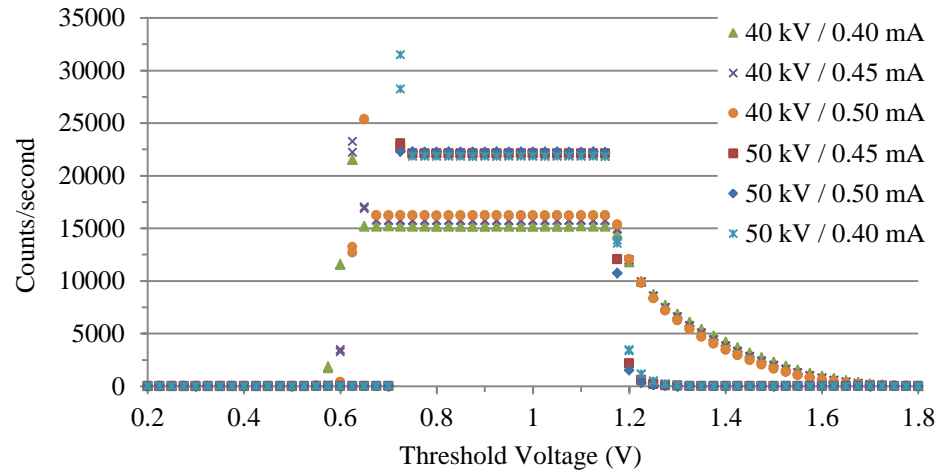


Figure 3.12. Scanning the voltage into the comparator using a CompactRIO analog out to determine the optimum threshold voltage and showing the output consistency of the inorganic tube.

Table 3.1: Component values used in the Geiger-Müller circuit shown in Figure 3.10.

R1	R2	R3	R4	R5	C1	C2
10 M Ω	240 k Ω	51 k Ω	15 k Ω	3.7 k Ω	7 nF	10 nF

3.2.3 Commercial X-ray Camera

The camera (Figure 3.13) purchased with the X-ray microsource was manufactured by Photonic-Science and rebranded for Bede Scientific Instruments Ltd. The camera used a terbium-doped gadolinium oxysulphide (GdOS:Tb) coated fiber optic bundle epoxy bonded to the CCD to transfer the scintillated light for a 1:1 imaging (i.e. no magnification).

The camera output provided 1000 pixels by 1000 pixels with a pixel size of about $10\text{ }\mu\text{m}$ square and grey scale resolution of 10-bit. Capture rate was a default at $30\text{ frames}\cdot\text{s}^{-1}$ and data was transmitted over a camera link interface via a National Instruments® PCI-1428 image acquisition card.



Figure 3.13. The branded Photonic Science camera that used a scintillator coated fiber banded directly bonded to the CCD.

3.2.3.1.1 Integer Overflow

An issue with the Bede Scientific camera was that when the X-ray generator was set at 50 kV and 0.5 mA, the camera experienced integer overflow even though the camera supposedly came with a 12-bit ADC for 4096 grey levels. Integer overflow is when the numerical value exceeds the maximum possible stored value and the counter resets to the minimum value and starts counting again. With the camera outputting an unsigned 10-bit integer, the values range from 0 to 1023. When the counter reaches 1023, the next value which would be 1024 is rolled over and recorded as 0. A value of 1025 would be 1 and so forth. An example of this effect is shown in Figure 3.14.

Three methods to handle overflow are reducing exposure time, data unwrapping, and lowering the X-ray intensity either with the generator settings or filtering. Unfortunately, at the highest intensity the exposure time could not be reduced to prevent overflow. For data unwrapping, dealing with a specimen obstructing the beam could prove

difficult. Therefore, the simplest solution was to reduce the intensity of the X-ray beam by lowering the generator's voltage and current. With an unsaturated image, the noise of the image was reduced by taking the average of multiple images. A benefit to averaging is additional resolution can be extracted from the data.

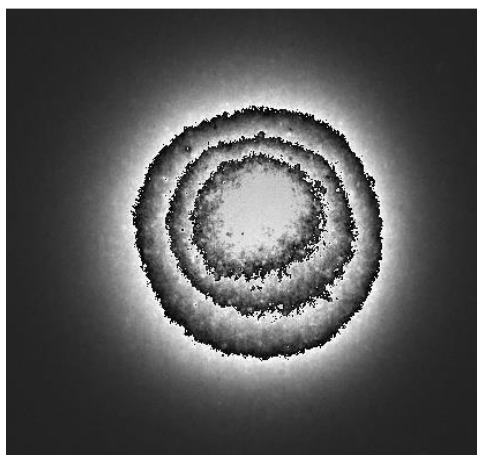


Figure 3.14. Integer overflow of the camera causing a ringing effect in the image.

3.2.3.1.2 Image correction

Systematic noise of digital cameras is called 'fixed-pattern noise' and comes in two forms, both visible with the Bede Scientific camera. Most obvious of the two is the speckling pattern seen in Figure 3.15 caused by photo response non-uniformity. Photo response non-uniformity is the variation in excitement from light from pixel to pixel. In other words, each pixel's gain or scaling factor. The second form of fixed-pattern noise component is dark signal non-uniformity caused by dark current, creating an offset from zero. Even when a sensor is not excited by light, electrons are thermally generated in the sensor, creating dark current. Although the role of the dark current offset is not as

noticeable on the image quality when compared to photo response non-uniformity, the reduces the camera's total range before saturation.

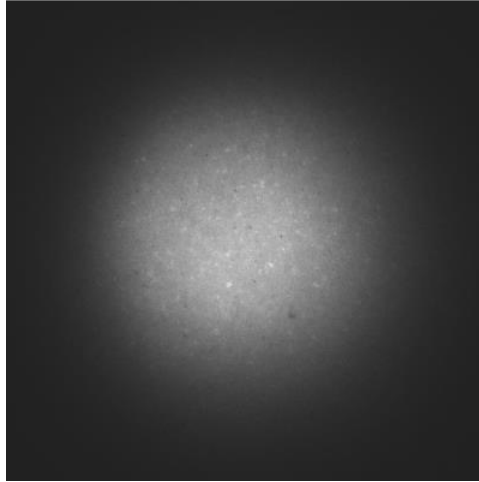


Figure 3.15. Fixed-pattern noise in an image of the X-ray beam. Speckling is caused by photo response non-uniformity and the graying of the background is the dark signal non-uniformity.

In photography, compensating for the fixed-point noise will improve image quality. This compensation is referred to as ‘flat-field correction’. Creating the flat-field corrected image requires a raw image, a dark frame, and the flat frame . A raw image is the actual photographic image of the object or in the case of X-ray imaging, the projection image. The dark frame is an image taken without light exposure and exposure time equal to the raw image. Opposite of the dark frame is a light frame, which is an image taken with uniform light and with an exposure time equal to the raw image. The flat-field corrected image is calculated using

$$Corrected = \frac{Raw - Dark}{Flat - Dark} \quad (3.9)$$

Correcting for photo response and dark signal non-uniformities involved removing the X-ray capillary focusing optic and placing the camera 200 mm from the X-ray generator. Removing the capillary optic allowed the full sensor to be exposed at the same time. With the X-ray generator off there were 1000 images captured, and the mean value for individual pixels were saved to a spreadsheet to form a dark frame. Turning the X-ray's on, another 1000 images were captured below the saturation level with dark signal non-uniformity removed. Since the default exposure time was always used, a singular mean value was calculated, and each pixel was divided by that mean value to generate a spreadsheet with scaling factors. The difference before and after correction is shown in Figure 3.16 with the object of interest being a quartz tuning fork. Removal of the X-ray beam profile is not always wanted, hence the alternative approach towards flat-field correction. For CT imaging where the beam profile should be removed, this is easily accomplished by taking an unimpeded image of only the beam profile.

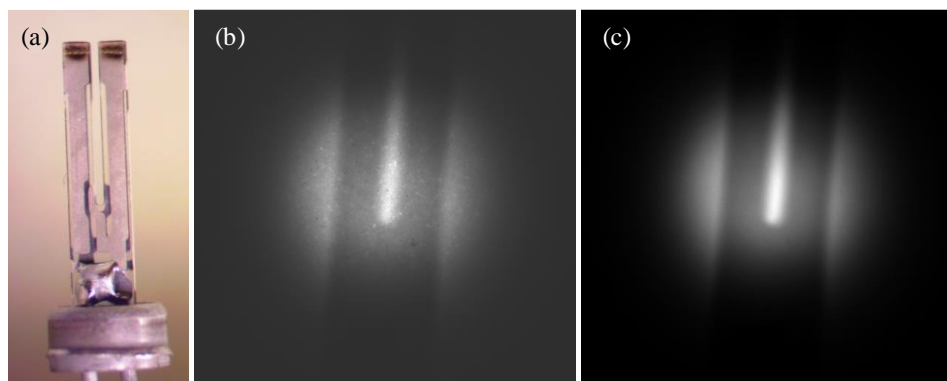


Figure 3.16. Picture of a (a) tuning fork, (b) the raw X-ray image, and (c) the post processing with image corrections. Images from [123].

Another way to see image quality improvement from the corrections is by creating a sinogram and performing the inverse Radon transform. The process is how a reconstructed cross-section of an object is formed for CT. To create a sinogram, an object

is rotated along an axis normal to the axis of the X-ray beam that is captured at different angles of this rotation. The same row number from each image is extracted and added to a matrix with respect to the capture angle (Figure 3.17). From the uncorrected sinogram image, a banding effect caused by speckling is clearly visible. After flat-field correction, the same sinogram shows the band artifacts have been significantly reduced.

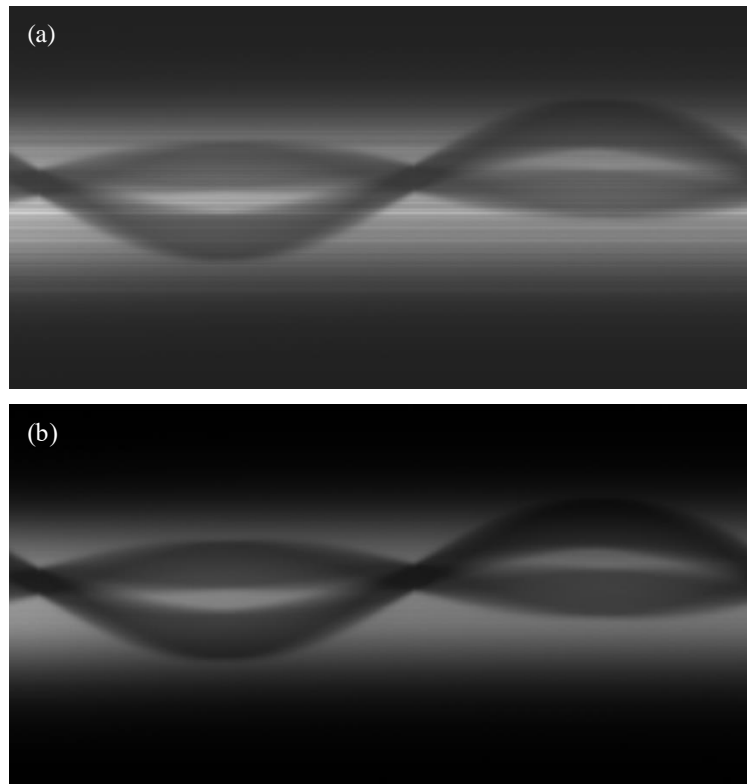


Figure 3.17. (a) Sinogram of the quartz tuning fork raw image at image row 277 and (b) the fixed-pattern noise corrected sonogram. Each column is the row's image as the object is rotated.

When the inverse Radon transform is performed on the sinograms of Figure 3.17, reconstructed cross-sectional images are shown in Figure 3.18. Without flat-field correction, the banding is converted into ringing artifacts within the reconstructed images. While the correction process is not perfect, Figure 3.18 further demonstrates the significance that applying correction has on discriminating features of an object.

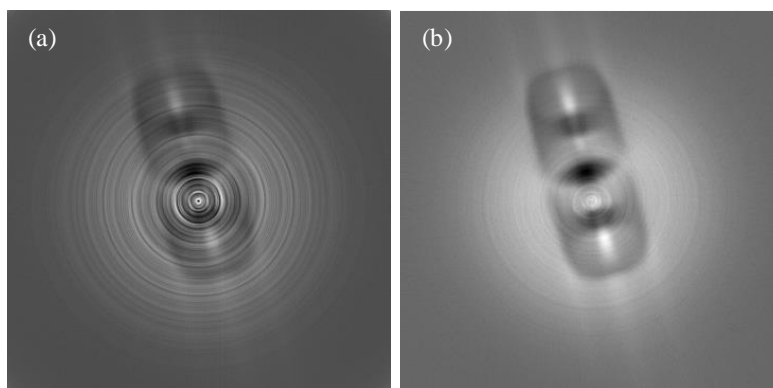


Figure 3.18. Inverse Radon transform created from the (a) raw sonogram (b) and the fixed-pattern noise corrected sinogram. Image is a cross-sectional view of a quartz tuning fork. Images from [123].

3.2.4 Custom X-ray Camera

When the Bede Scientific camera's fiber bundle debonded from the sensor, there was a choice to either repair or replace. Taking into account the age of the Bede Scientific camera and its estimated repair cost, the decision was made to build an X-ray camera. Going the custom route ensures full control and flexibility with upgrading components and component optimization for the application. Design requirements were that the performance should match or exceed the Bede Scientific camera and the first camera build should cost less than the Bede Scientific repair cost.

In this section, the three components determining X-ray camera performance are evaluated. These components are the digital camera, optics, and scintillator. Auxiliary components that are necessary for assembly are the lens tubs for setting path lengths and blocking ambient light, and adapters to convert the various thread sizes. A carbon fiber window protects the scintillator crystal from damage and blocks ambient light. Carbon fiber was chosen for its high strength and low electron density, making it nearly transparent

to X-rays. A picture of the first camera design is shown in Figure 3.19 and an exploded diagram showing the component layout is provided in Figure A.9 of Appendix A. An exploded image of the second version of the camera system with a turning mirror is shown in Figure 3.20. Information related to changing from leaded glass to a turning mirror is discussed in subsection 3.2.4.1.3 on optical components.

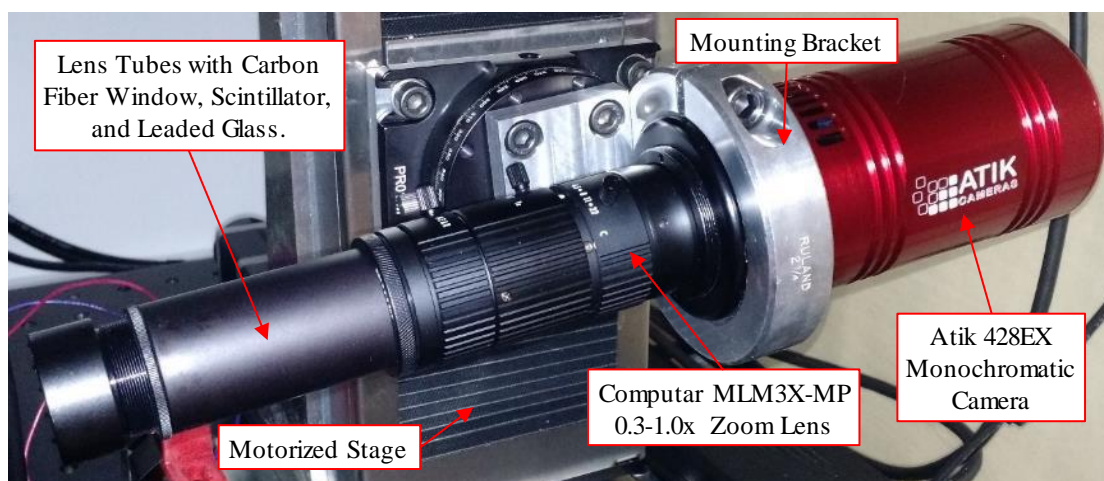


Figure 3.19. Assembled in-line style X-ray camera with a custom mount attached to a rotatory stage that is attached to a two-axis motorized YZ stage. In this setup, leaded glass protects the CCD from radiation damage.

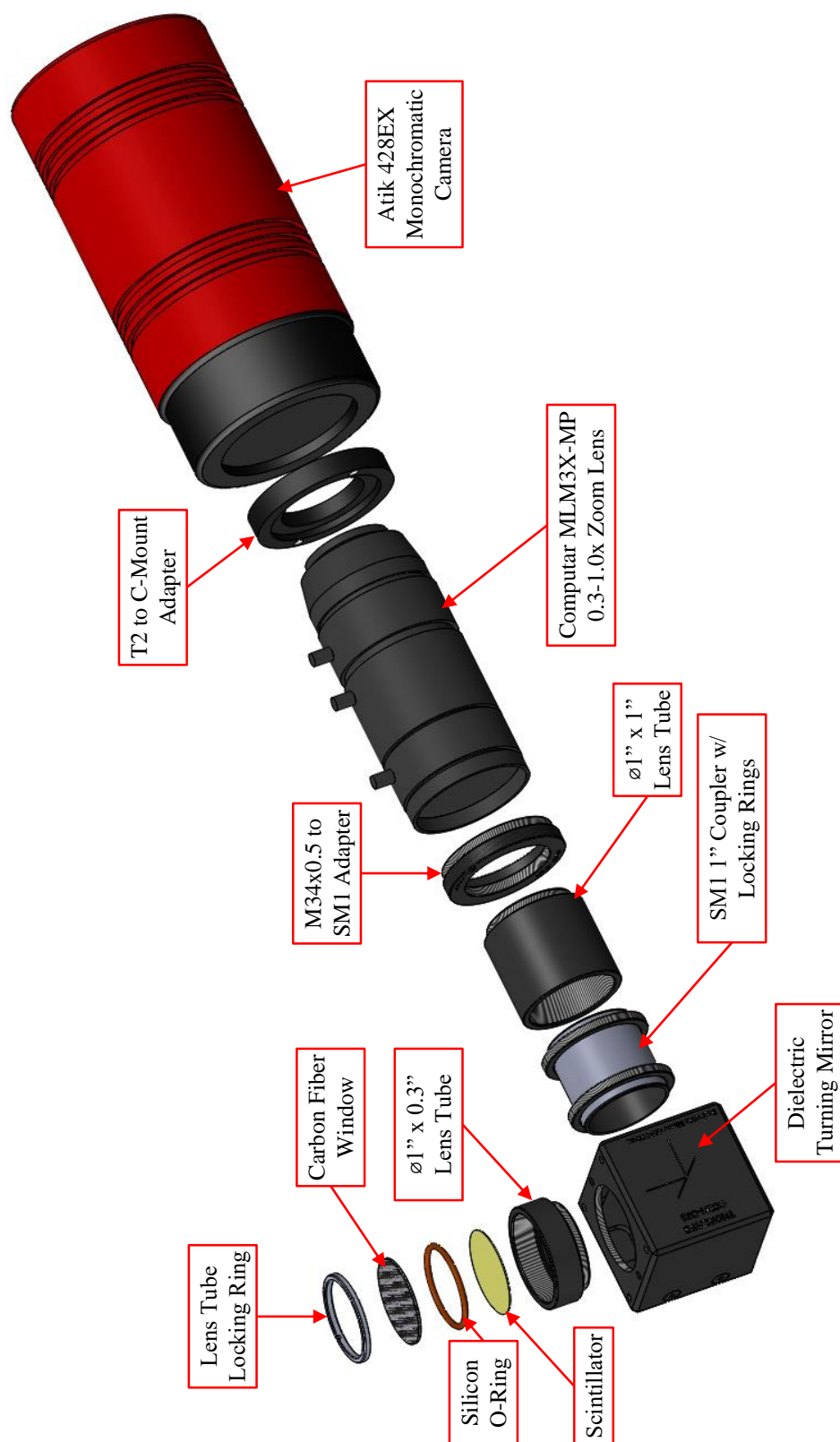


Figure 3.20. Exploded solid model of the periscope X-ray camera.

3.2.4.1.1 Digital Camera

In this section, the digital camera technical specifications and performance characteristics are discussed. Digital camera technical specifications that were picked for the custom build are a 14-bit or greater analog-to-digital converter (ADC), low readout noise, a high quantum efficiency sensor, and low dark current. CMOS sensors have higher readout noise than CCD, narrowing the search to CCD-based digital cameras. Quantum efficiency is the sensitivity of a sensor and is given as the percentage of incident photons at a wavelength that will produce charge carriers. The higher the quantum efficiency at a wavelength equal to the scintillators output, the better. By thermoelectrically cooling the sensor, dark current is reduced producing a large usable grey level range. Taking these factors into account, the digital camera purchased is the Atik 428EX with technical specifications listed in Table 3.2. The Sony ICX674 CCD sensor that is used in this camera has one of the highest quantum efficiencies for 550 nm wavelength light at 75%, a wavelength commonly matched to the light output by X-ray scintillators.

Table 3.2: Atik 428EX technical specifications [124].

Sensor	ADC	Resolution	Pixel Size	Exposure Time
Sony ICX674 ALG CCD	16-bit	1932 x 1452 pixels	4.54 μm	1 ms to ∞
Dark Current	Cooling	Readout Noise	Full Well	Gain Factor
0.0002 $\text{e}^- \cdot \text{s}^{-1}$ at -10°C	$\Delta T = -25^\circ\text{C}$	5 e^-	$\sim 17,500 \text{ e}^-$	0.27 $\text{e}^- \cdot \text{ADU}^{-1}$

As discussed, flat-field correction improves image quality by reducing the fixed-pattern noise. Out of the frames used for flat-field correction, the dark frame provides information about a digital camera's offset from dark current, hot pixels, and readout noise.

A bias frame is a dark frame captured with zero-length or the shortest exposure time. Unlike the dark frame, the major source of noise in a bias frame is the readout noise, given as the standard deviation of the signal. To measure the readout noise, the thermoelectric cooler was set to $-10\text{ }^{\circ}\text{C}$ and 100 bias were captured with a 1 ms exposure, 2 min delay between captures, and saved as uncompressed TIF images. Measuring the mean pixel value and standard deviation of different bias frames, when plotted verses time shows a time dependency (Figure 3.21). The change in mean pixel value is less than one analog-to-digital unit (ADU), staying within 322 ADU. Subtracting consecutive bias frames and taking the standard deviation, the deviation starts at 58 ADU and increases to about 64 ADU at steady state.

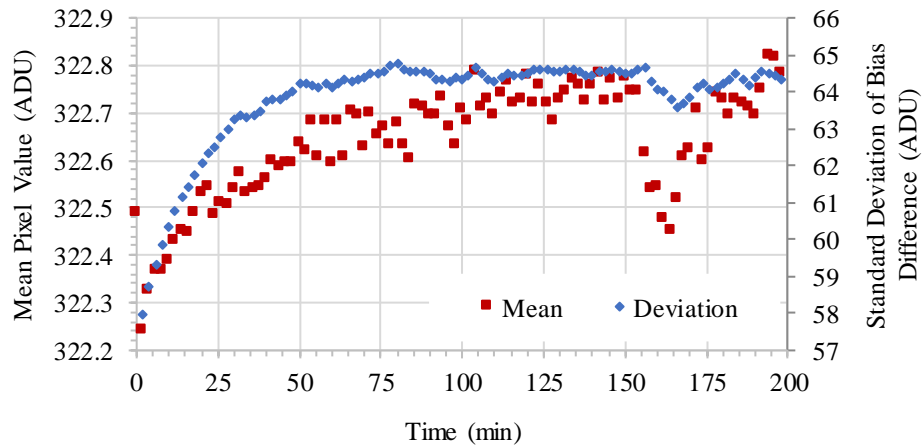


Figure 3.21. Mean pixel value and the standard deviation of the difference in bias frames with 2 min delay between capture.

Taking the steady state standard deviation as 64 ADU and applying Equation 3.10 yields a readout noise of 12.2 [125]. However, this assumes the gain factor provided by the camera manufacture is accurate. Atik Camera's gain factor of $0.27\text{ e}^{-}\cdot\text{ADU}^{-1}$ is calculated by dividing the full well value (17,500 e^{-}) by the maximum output value of the ADC

(65535 ADU). For a more accurate gain factor, pairs of light frame images were captured with 1 to 5 s exposures in 1 s increments and then 10 to 50 s exposures in 10 s intervals without oversaturation. For each pair, an average and a difference image are created. The standard deviation of the difference image is calculated and then converted to variance using Equation 3.11. From Figure 3.22, each point on the graph is the mean value taken from an averaged image and the slope of the line yields the camera's gain factor, approximately $0.25 \text{ e}^- \cdot \text{ADU}^{-1}$. Readout noise at the measured gain factor is 11.3 e^- .

$$\text{Readout Noise} = \frac{(\sigma_{\text{Bias Diff Image}}) * \text{Gain}}{\sqrt{2}} \quad (3.10)$$

$$\text{Variance}_{\text{Diff Image}} = \frac{(\sigma_{\text{Light Diff Image}})^2}{2} \quad (3.11)$$

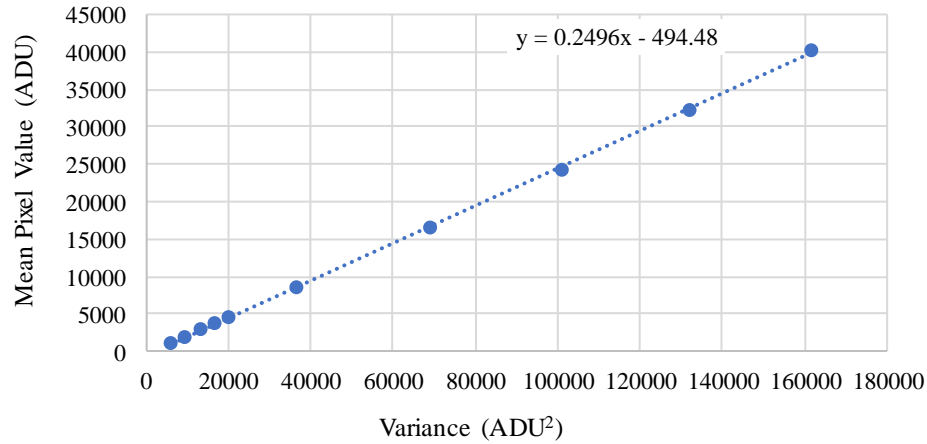


Figure 3.22. Mean pixel value verses variance. Slope is the camera's gain factor.

Directly after capturing bias frames with two-minute delays between captures, 100 bias frames were captured without adding wait time. The point of this test was to determine if capture rate impacted noise. By the time the images were captured, both the mean pixel

value and deviation of the bias difference had increased by 1 ADU. Results of the quick bias capture tests are shown in Figure 3.23. At image capture number 53, both the mean and deviation measurements suddenly drop. A similar drop is seen at 166 min during the 2 min delay test. An exact cause for this drop is unknown, but with the thermoelectric cooling in use to reduce the dark current, the cooling controller is the likely culprit.

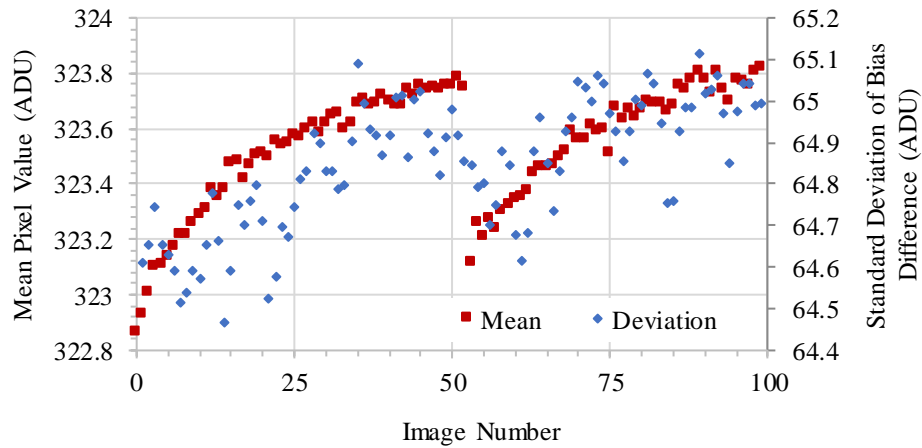


Figure 3.23. Mean pixel value and the standard deviation of the difference in bias frames with no additional delay between captures.

Fitting a plane to average bias frame without the delay indicates an inherent tilt in the pixel value (Figure 3.24). Slope of the tilt is along the vertical axis and is 7.5 ADU from minimum to maximum value. No significant tilt is present along the horizontal axis. To determine if the dark frame changes with time, dark frames were captured with exposure times of 1, 10, 20, 30, 40, and 50 s. Every dark frame's mean value was within $323.7 \text{ ADU} \pm 0.05 \text{ ADU}$ for a 3σ deviation. With the dark frame mean value consistently within 323 ADU, capturing a dark frame is not necessary. Instead, the bias plane from Figure 3.24 or subtraction of a constant can replace the dark frame. A histogram, Figure 3.25, shows the

50 s exposure dark frame as corrected using each method. Standard deviation using constant value correction was 50.559 ADU and plane correction was 50.518 ADU.

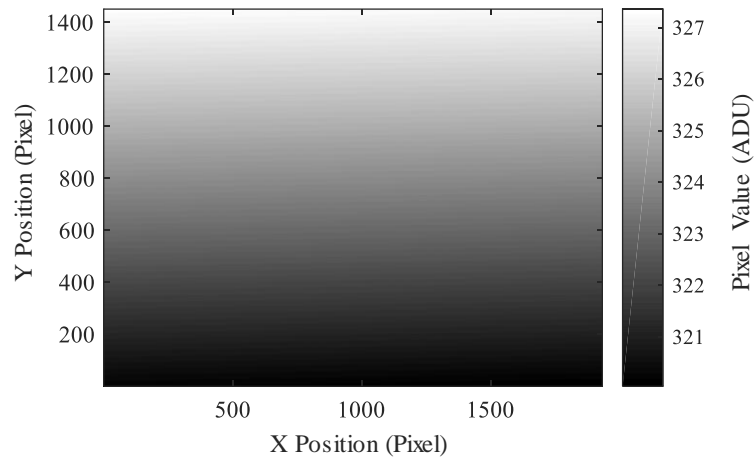


Figure 3.24. Tilt in the bias frame.

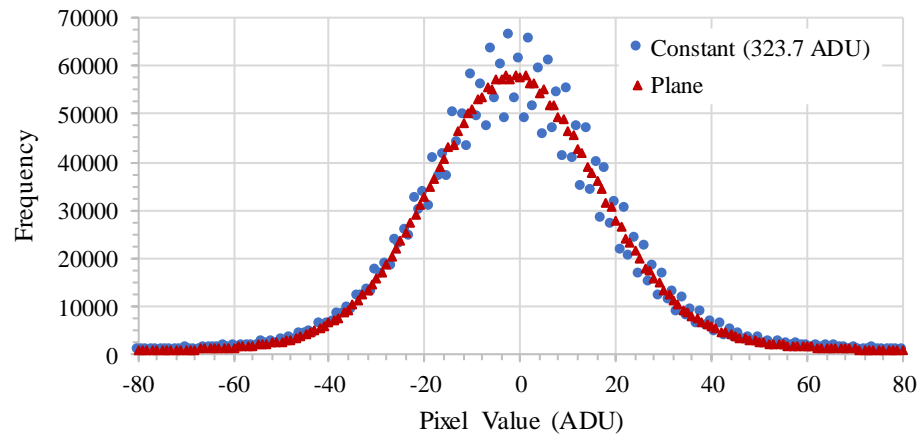


Figure 3.25. Dark signal non-uniformity removal comparison between a constant and plane. Dark frame exposure time was 50 s.

3.2.4.1.2 Optical Components

A desired optics feature for the custom camera is adjustable magnification. Adjustable magnification to a lower zoom would help while aligning the camera to the X-

ray beam and imaging multiple diffracted or reflected spots. Switching to high magnification could then be made for high resolution captures. The Computar MLM3X-MP zoom lens offers 0.3X to 1X magnification with the benefit that it maintains the working distance through its zoom range. Spot size difference between the Bede Scientific camera and the custom camera is shown in Figure 3.26. At 0.3x magnification zoom set for the custom camera, the spot size of the beam is nearly identical between the cameras. Included in Figure 3.26 is a visual representation of the new camera's 1.0x magnification compared to the 0.3x magnification and the Bede Scientific camera.

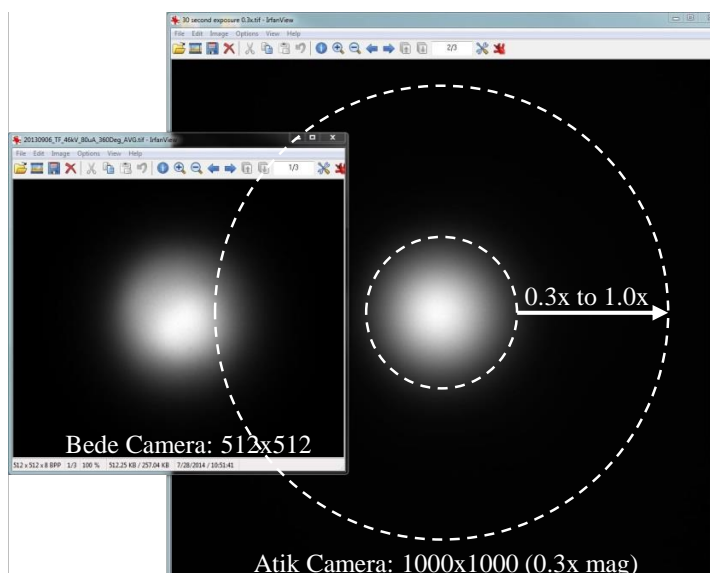


Figure 3.26. Comparison of the custom camera to the Bede Scientific camera.

Part of the optical system included protecting the sensor from X-ray radiation damage. In the first version of the camera setup, this protection came from a 5 mm thick piece of leaded glass placed between the lens and scintillator. The second iteration of the camera setup replaced the leaded glass with an E02 dielectric coated turning mirror (Thorlabs CCM1-E02) that reflects more than 99% of average light between 400 nm to 750

nm wavelength. Reasoning for switching from leaded glass to a turning mirror is that upon testing, the leaded glass attenuated about 12.5% of the light. Measuring this attenuation was accomplished by generating light with a mobile phone set to display only green light. With a lens tube attached to the lens, light from the environment was blocked and the source-to-sensor distance was kept constant by setting the lens tube directly onto the screen. Three tests were performed with each test acquiring 10 images, alternating between no glass and glass. A mean image was constructed from each test and then the mean and standard deviation of the images were calculated. Mean bit value with glass was 34,983 and without the glass it was 39,948. Standard deviation of the bit value for each image was 4% of the mean. Individual testing results are reported in Figure 3.27.

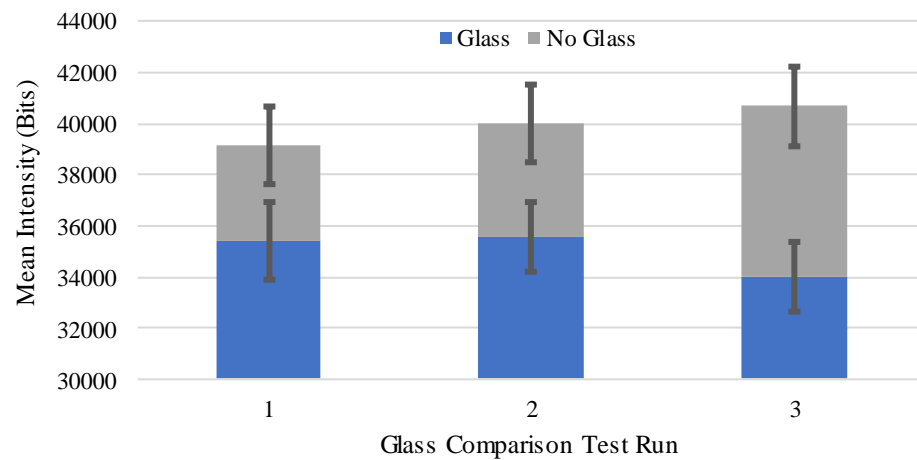


Figure 3.27. Leaded glass attenuation test of the custom camera.

3.2.4.1.3 Scintillators (Ce:YAG and Ce:LuAG)

The final component of the camera setup is the scintillator. First purchased was a 0.35 mm thick cerium-doped yttrium aluminum garnet (Ce:YAG) purchased from SPI Supplies. Peak emission wavelength is 550 nm for Ce:YAG, similar to the wavelength for

peak quantum efficiency of the camera sensor. Unfortunately, there is an inverse relationship between maximum resolution and maximum absorption. A thicker crystal absorbs more X-ray radiation than a thinner crystal, but the resolution is reduced. Since the goal of the facility is adaptability, a second Ce:YAG crystal with a 0.15 mm thickness was also purchased from SPI Supplies. The third scintillator purchased is a 0.05 mm thick, cerium-doped lutetium aluminum garnet (Ce:LuAG) from CRYTUR, Ltd. Its peak emission wavelength is 535 nm and the benefit it has over Ce:YAG is that at the same thickness, Ce:LuAG absorbs more X-ray photons for conversion to visible light [126]. Graphs supplied by CRYTUR show a 50 μm thick Ce:LuAG screen has a 70% absorption while the Ce:YAG screen has 52% absorption at 20 keV. At 40 keV, Ce:LuAG has 18% absorption and Ce:YAG is 11%. Unlike the Ce:YAG crystals, the Ce:LuAG was purchased with a 100 nm aluminum coating and a silicon dioxide protective coating on one face. This was suggested by a CRYTUR engineer to increase the capturable light by 1.8 times that of an uncoated scintillator. One last difference is that since Ce:LuAG scintillator is thinner, the crystal's diameter is 20 mm and was mounted to an aluminum support ring. Clear field of view is 18 mm.

To test the aluminum coating concept, the uncoated 0.35 mm scintillator was setup in the camera with the X-ray settings at 50 kV and 0.5 mA, and with the molybdenum filter attached. Camera exposure time was 30 s with 11 images captured and saved as TIF files. The 0.35 mm scintillator was then taken to the Grigg Hall clean room and coated with aluminum and silicon dioxide using an e-beam evaporator (Lesker PVD 75). With the same X-ray and camera settings, the freshly coated scintillator was retested, as was the 150 μm scintillator. In MATLAB®, the sets of TIF images were loaded and a mean image

generated. Summing along the columns, the row with the peak value was saved and plotted, Figure 3.28. Peak intensity with the coating was 1.9 times greater than without the coating. For the different thicknesses, the 350 μm thick scintillator generated 1.1 times more captured light than the 150 μm thick scintillator.

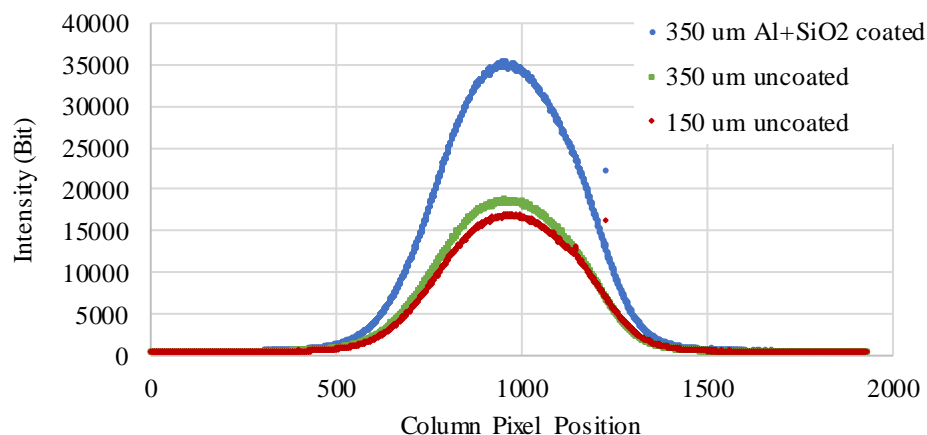


Figure 3.28. Intensity comparison between the uncoated 150 μm and the 350 μm scintillator before and after coating with aluminum and silicon dioxide.

3.3 X-ray Source Performance

Evaluating the X-ray generator's performance involved measuring the X-ray intensity for different power settings and measure the beam's divergence angle. Using the custom Geiger-Müller sensor, the microsource voltage was swept in 2 kV increments from 20 kV to 50 kV and the current was incremented with 0.1 mA steps from 0.1 mA to 1 mA. Voltage-current combinations exceeding 25 W are prevented by the microsource. Voltages from 26 kV and lower at all current settings reported a count rate below $500 \text{ counts}\cdot\text{s}^{-1}$. Beginning at 30 kV and 0.1 mA, the output exceeds $1000 \text{ counts}\cdot\text{s}^{-1}$ and continues to increase until 50 kV and 0.5 mA where it exceeds $22,500 \text{ counts}\cdot\text{s}^{-1}$ (Figure 3.29). As the count rate nears 18,000 to 20,000 $\text{counts}\cdot\text{s}^{-1}$, the Geiger-Müller tube's ability to quench the

electron avalanche may be at its limit. This could account for the non-linearity of the 46 kV and 50 kV lines.

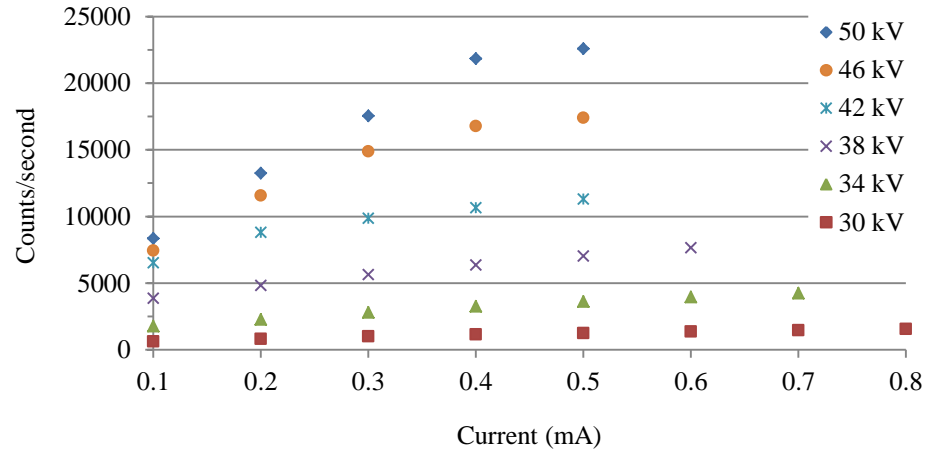


Figure 3.29. Output of the x-ray source as at different power settings.

Critical to the generator's ability as a source for interferometry is the beam divergence angle (i.e. cone angle). To measure divergence angle, the Bede Scientific camera was mounted to an XYZ stage system and moved the camera collinear to the beam. In the proximal position, different power settings were adjusted until the camera registered similar amplitudes. For the three measurements, their starting amplitude values were 716 bits at 43 kV, 723 bits at 44 kV, and 709 bits at 45 kV. The X-axis was moved in 5 mm increments from 0 mm (relative proximal position) to 140 mm and back to 0 mm, adjusting the Y- and Z-axis to keep the spot centered to the camera. After correction, the distance moved was 143.8 mm. At each position, the average beam's amplitude, average full-width half-maximum (FWHM) spot size, and all stage positions were recorded. Using the stage positions, the total distance for each move was computed and plotted against amplitude loss (Figure 3.30). The linearity of the three amplitude plot lines was measured as 0.995 using R^2 .

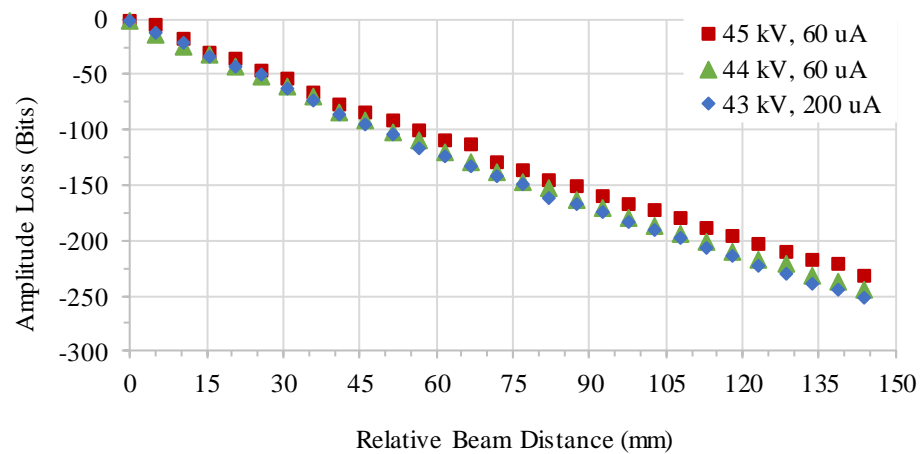


Figure 3.30. Amplitude of the spot size at different positions for the camera along the beam axis.

Beam spot size was measured at the FWHM. Spot sizes relative to distance for the three power settings are shown in Figure 3.31. From the plots, two relationships can be seen. First, the linearity of the change in spot size as measured using R^2 for the three plot lines was 0.997. Second, with nearly identical amplitudes, a higher voltage has a smaller FWHM spot size. FWHM measurements were fitted with best fit lines and a theoretical spot size at the proximal and distal locations was calculated. By assuming the spot sizes are two circles within the same cone, the divergence angle from the cone's centerline is calculated. Divergence angles are 0.049° for 43 kV, 0.047° for 44 kV, and 0.043° for 45 kV.

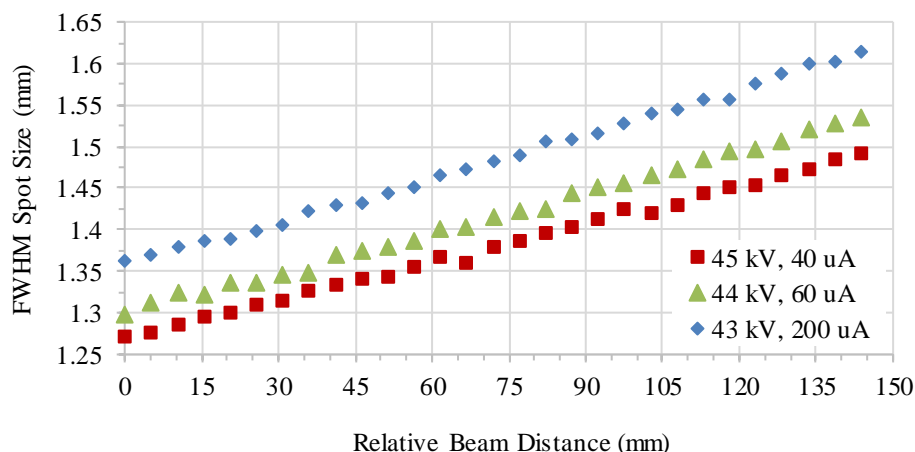


Figure 3.31. The full-width half-maximum spot size at different positions for the camera along the beam axis.

3.4 Diffraction Experiment

As one end-goal is to prove the facility can be used for X-ray interferometry, the ability to diffract through single crystal silicon had to be demonstrated. To accomplish this, an extra piece of silicon saved from the original interferometer was tested on the PANalytical X-ray diffractometer (XRD) located in Grigg Hall to verify the (111) plane's orientation. The XRD has a copper target with a $K_{\alpha 1}$ and $K_{\alpha 2}$ wavelength of 0.154 nm and was set to 45 kV and 40 mA. Scan step size was 0.02° with 249 points collected from 25.95° to 30.93° . The plot in Figure 3.32 shows diffraction occurring at 28.4° for the 2θ angle. From Bragg's law, setting λ to 0.154 nm and θ to 14.2° , the planar spacing was calculated at 0.313 nm. This confirmed we were diffracting through the (111) planes.

With knowledge of the spare silicon's orientation, the spare piece was then tested in the X-ray facility. The piece was mounted to a motorized rotary stage and the target face was aligned parallel to the X-ray beam. Since the microsource has a molybdenum target at about half the wavelength of the XRD's copper target, the Bragg angle is reduced to 6.4° .

Diffraction of the same (111) face was in fact experienced with the sample rotated to 6.4° . On the adjacent side the crystal orientation, the orientation was supposedly cut along the (220) planes. After rotating 90° to the new face, diffraction of the (220) orientation was seen at 10.5° . Pictures of the diffracted spots on a phosphorescent screen are shown Figure 3.33.

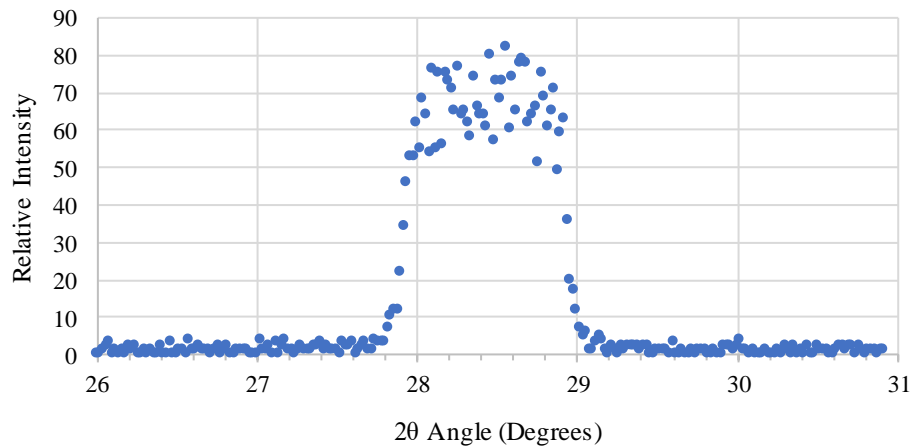


Figure 3.32. X-ray diffractometer data of a spare piece of silicon cut from the interferometer with an intensity peak at approximately 28.4° .

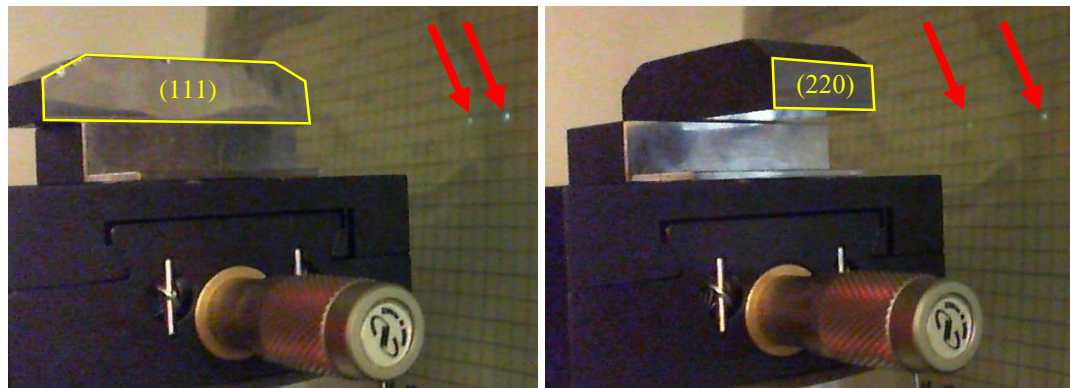


Figure 3.33. Successful diffraction from the (111) and (220) planar orientations. Diffracted spot is indicated by the left arrow and transmitted spot is the right arrow.

After finding the (111) and (220) diffractions, the silicon was rotated with the (111) planes facing downwards and the (220) planes perpendicular to the beam. The diffractions planes in the new orientation were detected as (111), however with a small shift in the vertical direction as well as viewed in Figure 3.34. This vertical component is explainable by looking normal to the (220) plane and setting one of the (111) planes in the horizontal position. The resulting orientation produces tilted (111) planes. The angle between the (111) tilted plane and a line perpendicular to the horizontal plane is 19.5° .

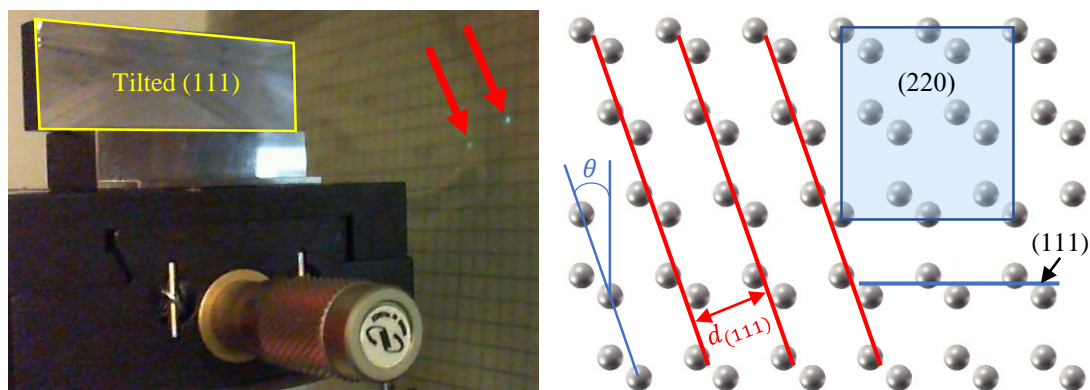


Figure 3.34. Diffraction from a tilted (111) plane. Diffracted spot is indicated by the left arrow and transmitted spot is the right arrow. Angle θ is approximately 19.5° .

3.5 Reflectivity Measurements

A recent study started in the X-ray facility is reflectivity. X-ray reflections occur at very small angles, less than 1° , with lower electron density materials requiring an ever-smaller angle. In Figure 3.35, the facility successfully measured reflectivity from a polished tungsten carbide block. An unavoidable issue in reflectivity measurements is that at the start, the surface area will be too small to capture a full beam's width. This is demonstrated by the increasing intensity from 0° to 0.12° . As the angle continues to increase, surface area for the X-rays to reflect increases, corresponding to an increase in reflected intensity.

A small pinhole aperture has since been implemented between the source and the specimen, reducing the beam spot size with intent of improving the reflection data for small angles.

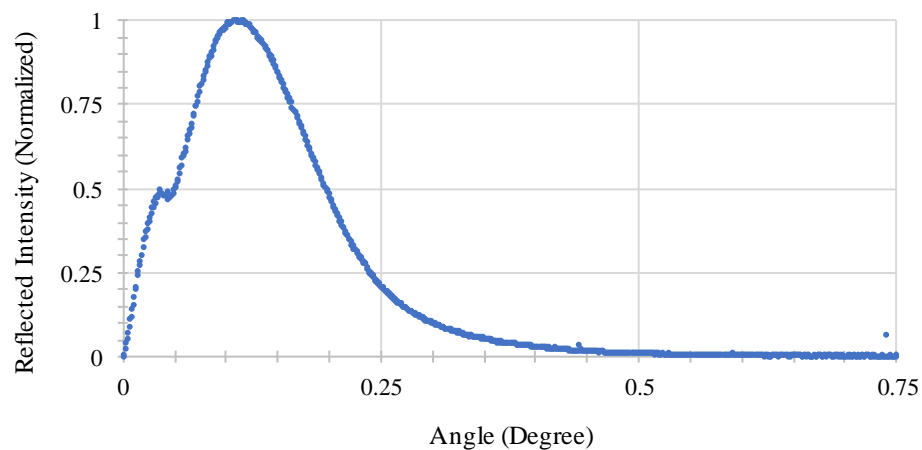


Figure 3.35. X-ray reflection curve for tungsten carbide. Data provided by Jacob Cole.

CHAPTER 4: DEVELOPING AN X-RAY REFERENCE OBJECT

To verify the calibration of an X-ray system, a reference object with known features is a helpful tool. A common method is to utilize machines (CMM), also called coordinate measurement systems (CMS), to measure a part and use the reported values as a reference for the X-ray scan [76]. The problem with relying on a CMM is the inability to measure internal voids, and as the size of the parts decreases, the difficulty of measuring with physical probes increases. With X-ray computed tomography systems increasing in popularity, there will be a growing need for a calibrated reference object for sub-millimeter measurable features. The goal of the reference object project is to develop a manufacturing process to produce measurable internal cavities and outline a path towards traceability. Topics discussed in this chapter are the design of the reference object, the reactive-ion etching and direct bonding fabrication processes, and the complete manufacturing methodology of the reference object.

4.1 Principles of Microfabrication

The microfabrication processes used in the construction of the silicon reference are photolithography, reactive-ion etching, and silicon bonding. In this section, background is provided for each, and results from etching and bonding studies are discussed.

4.1.1 Photolithography

Photolithography is the microfabrication process developed to transfer the pattern of a photomask to photoresist. Photomasks are generally made from soda-lime or quartz glass to allow ultra-violet (UV) light to pass through to expose the photoresist. Parts of the photomask making up the pattern to be transferred are coated with chrome to block the UV

light. The basic photolithographic steps are substrate cleaning, photoresist application, UV exposure, and developing (Figure 4.1).

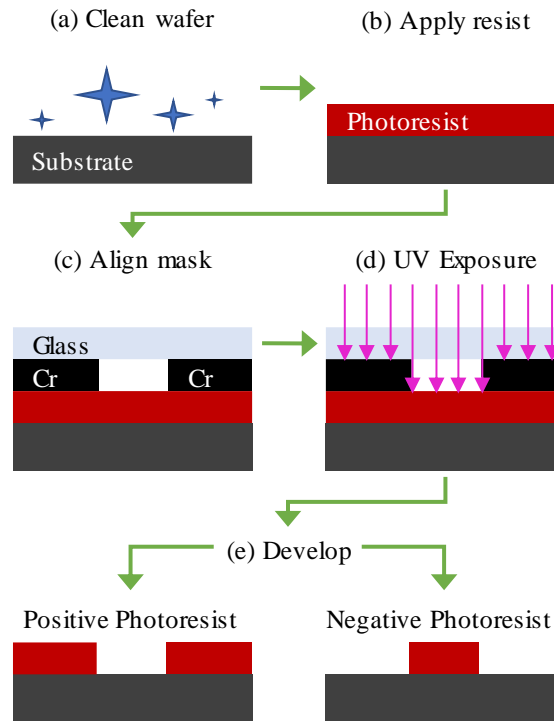


Figure 4.1 Photolithographic process and photoresist types.

Cleaning the surfaces of organic and inorganic material is done typically with wet chemicals involving a solution containing hydrogen peroxide, or with solvents like isopropyl alcohol, acetone, and methanol. One peroxide-based cleaning method is the standard clean 1 (SC 1 or RCA 1) and standard clean 2 (SC 2 or RCA 2). SC 1 is a mixture of NH_4OH , H_2O_2 , and H_2O in a 1:1:5 ratio that removes organic contaminants and SC 2 is HCL , H_2O_2 , and H_2O mixed as a 1:1:6 ratio for removing metallic contaminants [127–129]. These solutions are heated to 75 °C before submersing the wafers for 10 min. The second most common wet cleaning method is to use ‘Piranha’, a combination of sulfuric acid (H_2SO_4) and hydrogen peroxide in 2:1 to 4:1 ratios. ‘Piranha’ is an exothermic reaction

that is effective at removing organic contaminants, photoresist, and other visible residues [128].

After wet cleaning followed by rinsing of the substrates with DI water, the substrate surfaces are prepared for the photoresist. First, the water is removed by blow drying or spin drying with dry nitrogen gas, and sometimes baking. To improve the adhesion of the photoresist and avoid features washing away, the surfaces can be altered with adhesion promoters. Wafers are coated on one side with a liquid photoresist using a spin coater. Spinning the wafer at a high speed forces the photoresist to evenly spread across the surface. The type of photoresist can be either positive or negative, determined by their solubility after UV light exposure. If the photoresist becomes soluble when exposure, it is positive photoresist, and if it becomes insoluble, then it is negative photoresist. Choosing photoresist depends on the application as differences exist beyond their reaction to UV light, such as adhesion, cost, developer type, minimum feature size, undercutting, and more [130]. A soft bake of the coated wafers reduces the photoresist's solvent content. Without soft baking, photoresist can stick to the photomask, causing damage to the photomask or corrupting the pattern transfer.

The last part of the photoresist lithography is developing. An exposed wafer can be post-exposure baked to reducing standing waves caused by constructive and destructive interference at the edges of photoresist features. Washing away the soluble photoresist is done with either a sodium hydroxide or tetramethylammonium hydroxide (TMAH) solution, the latter more commonly used today because it is metal-ion free.

4.1.2 Silicon Etching

A common method to precisely remove material from silicon wafers is by wet or plasma (dry) etching. Wet etching involves submersing the wafer into a liquid etchant, such as hydrofluoric acid for silicon dioxide, or hydrofluoric acid with nitric acid (HNO_3) for silicon where nitric acid acts as a surface oxidizer that is then removed by the hydrofluoric acid [131]. While the advantage of wet etching is a low entry cost, the etchants are usually isotropic which prevents the formation of large aspect-ratio features. Anisotropic wet etchants generally have etching rates that vary with respect to the crystal orientation [132]. For instance, etching $\{100\}$ silicon wafers can produce a trapezoid shape etch with $\{111\}$ sidewalls. Depictions of the etching behaviors are shown in Figure 4.2.

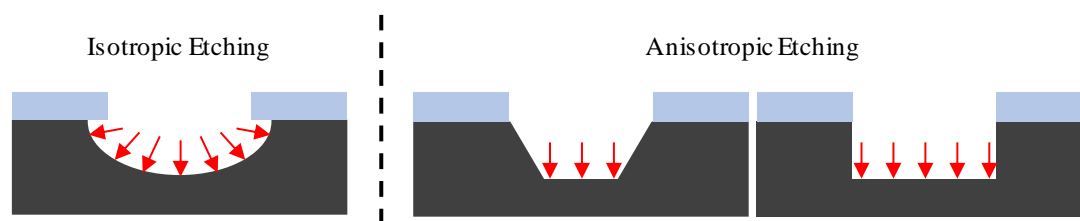


Figure 4.2 Isotropic and anisotropic etching.

Dry etching options are plasma etching, ion etching, and reactive-ion etching. Plasma etching is an isotropic etch that involves generating a plasma of neutrally charged atoms and free radicals to etch the substrate. Ion etching (or milling) is anisotropic and etches by using the kinetic energy of ions accelerated into the substrate to remove material. Reactive-ion etching (RIE) uses a combination of chemical reaction and kinetic energy. Ions from the plasma are accelerated onto a wafer that is placed between two electrodes with an electric potential difference. The process is primarily anisotropic, but some undercutting does occur. A modified version of RIE used to manufacture higher aspect

ratio features with nearly vertical sidewalls is deep reactive-ion etching (DRIE). The Bosch version of DRIE (Figure 4.3) works by switching between coating the surfaces with a passivating layer of octafluorocyclobutane (C_4F_8) and then etching with sulfur hexafluoride (SF_6) [133]. RIE is predominantly anisotropic, with the ions removing the passivating layer at the bottom of the trenches and etching into substrate before the passivating layer on sidewalls has been etched away.

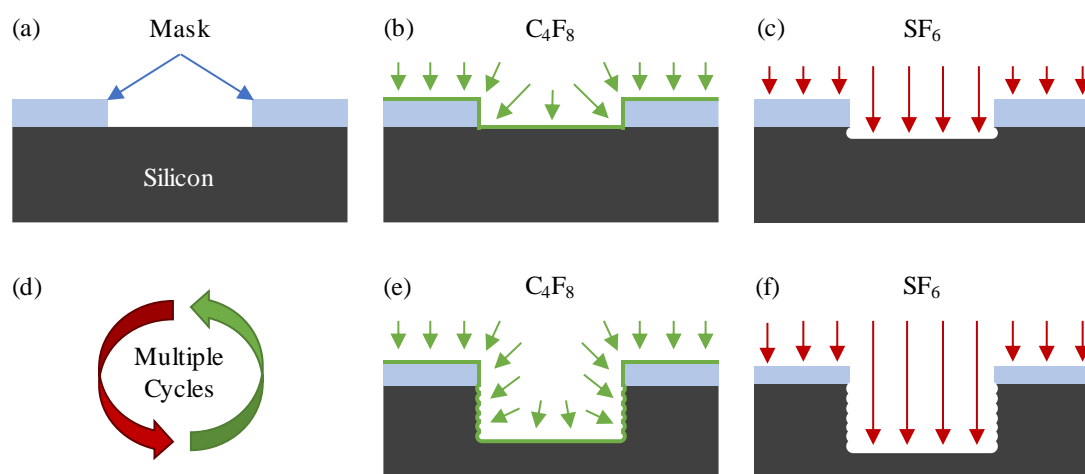


Figure 4.3 Deep reactive-ion etching using the Bosch process where a layer of C_4F_8 protects the side walls during the SF_6 etch.

Disadvantages to dry etching start with, first and foremost, the high costs to purchase and maintain the etching system. Also, some of the processing gases (such as C_4F_8) are corrosive to the equipment and toxic to users. Regarding the substrate, drawbacks are undulations in the sidewall (scalloping), non-uniformity of the etching rate over the wafer, aspect ratio dependent etching, and possible re-deposition of non-volatile by-products that can create a structure called ‘black grass’ or ‘black silicon’ [134–136]. Although black grass is considered a disadvantage for the reference object application,

research has shown that the needle-like structures improve the efficiency of light to electricity in the solar cell and photodetectors industries [137,138].

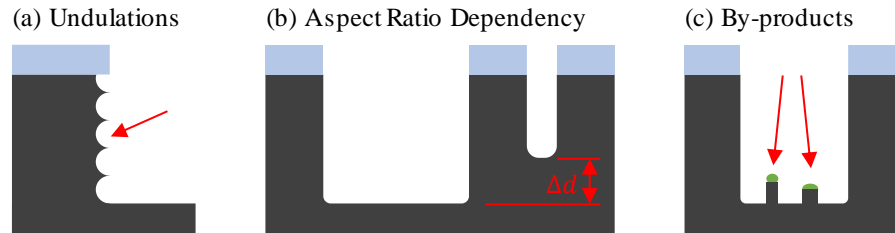


Figure 4.4 Physical disadvantages to performing DRIE. (a) Sidewall undulations from the Bosch switching. (b) Aspect ratio dependent etching. (c) By-products re-depositing on surface.

4.1.2.1 Etching Study

Studies were performed using a Surface Technology Systems (STS) Advanced Silicon Etcher (ASE) for deep reactive-ion etching to characterize the response for select processing parameters. Process parameters for Bosch DRIE (Figure 4.5) are the number of cycles, etching time, passivation time, coil power, platen power, frequency, platen chiller temperature, process pressure, and gas flow rates (O_2 , SF_6 , C_4F_8 , and Ar). A baseline recipe was created with the parameters listed in Table 4.1. Responses to changing the parameters are the etch rate, mask selectivity, wafer etching non-uniformity, etch profile angle, aspect ratio dependency, etched trench surface roughness, and sidewall roughness.

To avoid investing a significant amount of time and resources, two responses chosen as being critical for reference object manufacturing are the etched surface roughness and non-uniformity. Referring to a trend chart developed by Berkeley Microfabrication Laboratory, non-uniformity is impacted most by the processing pressure, SF_6 flow rate, and the helium backside cooling the cools the wafer. Surface roughness is

impacted by the processing pressure and platen temperature [139,140]. Because the platen temperature could not be changed through the software, only the processing pressure and SF_6 flow rates were evaluated.

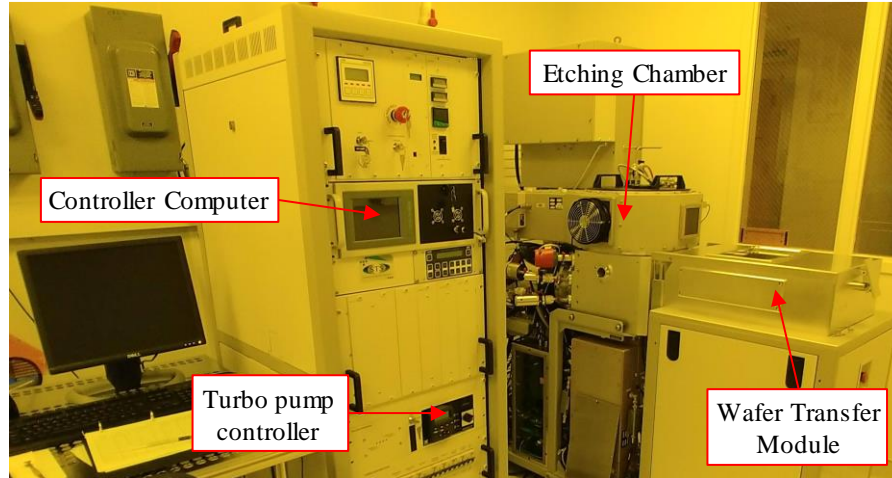


Figure 4.5. The STS Advanced Silicon Etcher used for deep reactive-ion etching.

Table 4.1. Baseline recipe for silicon etching with the ASE.

	Passivation	Etch	
Cycle Times	7	9	s
Pressure	20	20	mTorr
C_4F_8 Flow	85	0	$\text{cm}^3 \cdot \text{min}^{-1}$
SF_6 Flow	0	130	$\text{cm}^3 \cdot \text{min}^{-1}$
O_2 Flow	0	13	$\text{cm}^3 \cdot \text{min}^{-1}$
Ar Flow	0	0	$\text{cm}^3 \cdot \text{min}^{-1}$
Coil Power	600	600	W
Platen Power	0	30	W
Bias Frequency	0	13.56	MHz
Platen Chiller	20	20	$^{\circ}\text{C}$

A photomask filled with $530 \mu\text{m}$ by $200 \mu\text{m}$ rectangles was used to determine the etch rate, non-uniformity, and surface roughness while resulting from changes in the

etching parameters. A paper template with nine $5 \times 5 \text{ mm}^2$ cutout squares was used to assist with measuring each wafer in approximately in the same locations. From center, four cutout areas are 15 mm away and the other four are 30 mm away. A diagram of this template and a confocal image of the rectangular feature are shown in Figure 4.6. .

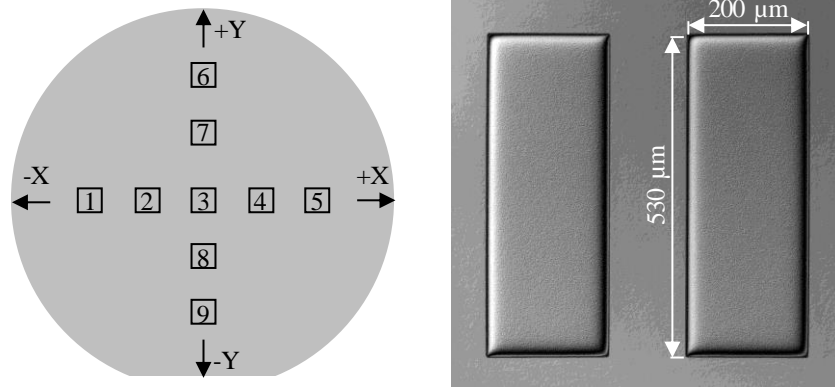


Figure 4.6 (Left) Paper template design with location tracking numbers. (Right) Confocal image of etched rectangular features that were measured.

Measurements of the etched features were obtained using an Olympus® LEXT™ OLS4000 confocal microscope and imported into MATLAB® where a script was written to automate analysis (Figure 4.7). Settings for the confocal measurements were 20× objective, 0.6 NA, $640 \times 640 \text{ μm}^2$ field of view, $0.625 \text{ μm} \cdot \text{pixel}^{-1}$ spatial sampling, and data acquisition averaging set to '4'. The first etched wafer using the baseline etching parameters and cycled 45 times had a mean etch depth of 23.04 μm and mean etch rate (E_{avg}) of $512 \text{ nm} \cdot \text{cycle}^{-1}$. Minimum etch rate (E_{min}) was $500 \text{ nm} \cdot \text{cycle}^{-1}$ near the wafer's center and maximum etch rate (E_{max}) was $527 \text{ nm} \cdot \text{cycle}^{-1}$ at 30 mm from center. Using Equation 4.1, the etch non-uniformity was calculated at 2.7%.

$$NU(\%) = \frac{E_{max} - E_{min}}{2E_{avg}} \quad (4.1)$$

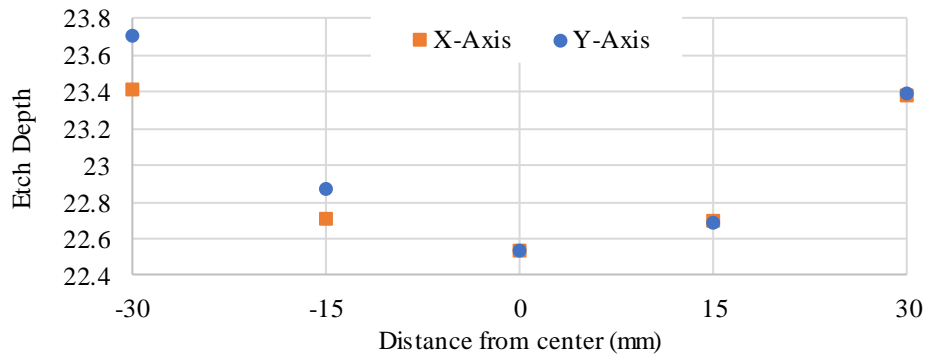


Figure 4.7 Confocal measurements of the step heights for the baseline etch.

Processing pressure was tested first and was evaluated at 15, 20, 25, and 30 mTorr. With all other parameters set to their baseline value, the trend found is that an increase in processing pressure leads to an increase in etch rate. This increase in etch rate comes with an increase in non-uniformity and roughness. To confirm that the clean room's confocal microscope can be trusted for quick measurements, the 15 mTorr and 20 mTorr wafers were also measured by a Zygo® ZeGage™ Pro coherence scanning interferometer (CSI). Settings for the ZeGage™ Pro were the 20× objective with 0.4 NA, $417 \times 417 \mu\text{m}^2$ field of view, $0.407 \mu\text{m} \cdot \text{pixel}^{-1}$ spatial sampling, and data acquisition averaging set to '4'. Filtering applied to the data was a low pass 'Convolution' that averaged a 3×3 matrix. Data was opened in MATLAB® where a plane was fitted to a region of the polished surface for leveling. Rectangular regions of the etched surfaces were also fitted with planes to measure the step height. Data within the rectangles were used in the fitting was used to calculate the RMS roughness. Results for the processing pressure experiments as measured by the confocal and CSI are provided in Table 4.2. Surface topography repeatability (STR) for the ZeGage™ Pro CSI was measured as less than 1.4 nm using a Zerodur optical flat with the same settings previously described. The confocal's STR was measured as 9 nm using a

polished silicon wafer. For all measurements except for the RMS roughness, the results are similar. While the roughness value measured by the two instruments is significantly different, both results indicated that a lower pressure yields a smoother etched surface.

Table 4.2. Processing pressure etch results for 15, 20, 25, and 30 mTorr.

	ZeGage CSI		LEXT Confocal				
Pressure	15	20	15	20	25	30	mTorr
Mean Depth	19.44	23.05	19.31	23.04	26.9	27.6	μm
Mean Etch Rate	432	512	429	512	598	613	$\text{nm}\cdot\text{cycle}^{-1}$
Δ Etch Rate	16	25	18	26	35	44	$\text{nm}\cdot\text{cycle}^{-1}$
Non-uniformity	1.9	2.4	2.1	2.6	2.9	3.6	%
RMS Roughness	4.2	7.5	20.1	30.2	50.1	63.3	nm

Since the 15 mTorr process pressure had a lower non-uniformity and roughness results than the baseline, the 15 mTorr processing pressure was retained when evaluating the SF_6 flow rate. With the baseline set at $130 \text{ cm}^3\cdot\text{min}^{-1}$, the flow rates tested were $120 \text{ cm}^3\cdot\text{min}^{-1}$ and $140 \text{ cm}^3\cdot\text{min}^{-1}$. For the flow rate test, one wafer was processed with 45 cycles at $130 \text{ cm}^3\cdot\text{min}^{-1}$ and the other three wafers were processed at 49 cycles. Confocal microscope measurements of the flow rate study are listed in Table 4.3 and show that the difference between $120 \text{ cm}^3\cdot\text{min}^{-1}$ and $130 \text{ cm}^3\cdot\text{min}^{-1}$ is not significant. However, increasing to $140 \text{ cm}^3\cdot\text{min}^{-1}$ appears to improve the roughness and non-uniformity. A common trend between the pressure and flow rate studies is that reducing the etching rate causes a reduction of the surface roughness and non-uniformity.

Table 4.3. SF₆ flow rate etch results at 120, 130, and 140 cm³·min⁻¹ measured using the confocal microscope.

Property	Values				Unit
SF ₆ Flow Rate	130	120	130	140	cm ³ ·min ⁻¹
Cycles	45	49	49	49	-
Mean Depth	19.27	20.15	20.88	20.36	μm
Mean Etch Rate	428	424	426	416	nm·cycle ⁻¹
Δ Etch Rate	17	17	18	16	nm·cycle ⁻¹
Non-uniformity	2.0	2.0	2.1	1.9	%
RMS Roughness	23.0	23.4	22.9	18.3	nm

4.1.3 Silicon Bonding

While many bonding techniques have been developed, two popular methods are intermediate-layer and direct bonding [141]. Intermediate-layer bonds can be polymer adhesives, creating solder-like bonds using low temperature glass or a eutectic system, or thermo-compression that creates a bond between two thin metal films [141–148]. Direct bonding is a method of bonding where no intermediate layer is used and that the bond is mainly van der Waals forces and hydrogen bonds [127,141,149]. In the development of the reference object, bonding with an intermediate-layer was not tested since there is a risk of the intermediate-layer appearing in the X-ray data, either by adding significant thickness between the silicon layers, or as an apparent change in density from silicon.

Direct silicon bonding methods are divided into hydrophilic and hydrophobic. The contact angle describing the two surfaces is measured as drawn in Figure 4.8. Hydrophilic surfaces have a low contact wetting angle (< 5°) made possible by the native oxide layer on the silicon surface that forms silanol groups. Silanol groups are silicon-oxygen-hydrogen connections (Si-O-H). In contrast, hydrophobic surfaces have a high contact

wetting angle ($> 90^\circ$) and are created by replacing the oxide surface with mostly hydrogen terminated (Si-H) and some fluoride terminated (Si-F) bonds.

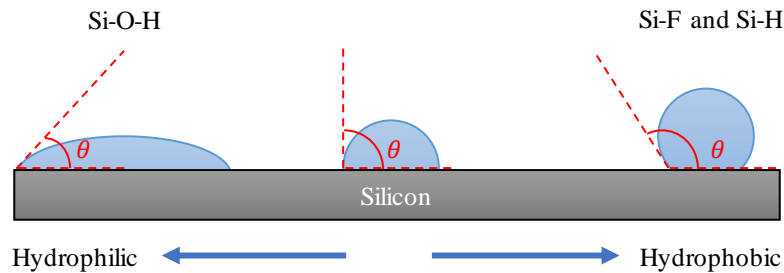


Figure 4.8 Contact angles for hydrophilic and hydrophobic surfaces.

Contributing to the bond strength are surface roughness, flatness, and cleanliness [127,149]. Improving the bonding performance starts with using the highest quality of polished wafer. When choosing between prime, test, mechanical, or reclaimed grade wafers, the prime grade has the tightest tolerances on surface roughness, cleanliness, and flatness. The prime grade double-sided polished wafers used in this study were purchased from University Wafer and are quoted to have a mean roughness (Ra) below 0.5 nm and thickness variation below 1 μm for a 100 mm diameter wafer [150]. For flatness or warp of the wafer, the contacting areas are increased by clamping the substrates together [151].

The next contributing factor to the success of bonding is cleanliness. Research shows that a particle of dimension of 1 μm will prevent bonds from forming within a 10 mm diameter area for wafers with thickness greater than 200 μm [127,151]. As particle size decreases, there is a critical height where the radius of the unbonded area converges to the same order as the particle size, around 100 nm [127]. To remove the various contaminants from the wafer surfaces, wet and dry processes have been developed.

Wet cleaning solutions are used to remove contaminants. The two commonly used solutions are the standard clean 1 and standard clean 2 or ‘Piranha’ as described in section 4.1.1 on photolithography. High purity chemicals and deionized water must be used for cleaning and rinsing. Contaminants trapped within or on the native oxide layer are removed by stripping the native oxide with HF. Following an HF dip, extra care must be taken for hydrophobic silicon to prevent recontamination as the surface becomes highly attractive to particulates. Buffered hydrofluoric with ammonium fluoride should be avoided when removing the oxide surface as it can increase surface roughness and thereby reduce bonding energy [127,151].

Dry cleaning options are gas-phase, plasma, and cryogenic aerosols [152–154]. Gas-phase and cryogenic aerosols options are not readily available at UNC Charlotte, and therefore, focus is placed on plasma systems. In the section discussing etching, the systems used for plasma etching can perform photoresist stripping, residue removal, and surface conditioning [127,128,149,153,155]. Similar to the plasma etching of silicon, the mechanisms behind plasma cleaning can be both physical and chemical. The gases used for photoresist stripping and residue removal are oxygen, argon, nitrogen, and hydrogen. Research also shows that plasma treatment can improve the bond’s interface energy [149,155,156].

The final key to a strong bond is to increase the bond energy by annealing the silicon wafers after the surfaces have been placed in contact. Between 200 °C and 800 °C, hydrophilic bond energies stay nearly constant at $1.3 \text{ J}\cdot\text{m}^{-2}$. Hydrophobic bond energies increase exponentially and by 600 °C the hydrophobic bond energies have surpassed the hydrophilic. By 700 °C, the hydrophobic bond energy exceeds $2.7 \text{ J}\cdot\text{m}^{-2}$. It is not until 800

°C that hydrophilic bond energies start increasing again and this is attributed to the oxide layer becoming viscous enough to fill gaps [127]. At 1100 °C, hydrophilic and hydrophobic bond energies match the strength of bulk silicon [127,157,158]. Plasma-treating the surfaces and then annealing can have bond energies over $2 \text{ J}\cdot\text{m}^{-2}$ at 300 °C to 400 °C [127]. At lower bonding temperatures, increasing the annealing time will increase the bond energy [127].

4.1.3.1 Bonding Study

While developing the bonding process, many approaches were tested with some shown in Table 4.4. Deionized water (DI) and nitrogen were used for the rinsing and drying operations. Methods were tested multiple times, although some tested more than others depending on their initial indications of bonding. The first attempts were made using methods 1 through 4 with stacks of six or eight 4 mm square wafers all starting with ‘Piranha’ cleaning. These wafers were sandwiched between quartz discs with more than 200 g of tungsten on top and annealed at 1000 °C for 1.5 hr (the furnace step in Table 4.4) [159]. With little to no success, the methodology was significantly changed. These changes were; increasing the diced silicon pieces to 17.2 mm squares, limiting stack size to four layers, using a steel clamp to increase pressure to the stack, and a two-step annealing process. With the clamp, the first anneal was tested with temperatures between 400 °C and 600 °C, and the second anneal was to 1000 °C without the clamp. Methods 1 through 4 were retried before ultimately being abandoned. Of the remaining methods, the methods that produced fully intact bonded parts that were able to be diced were methods 6 and 8. From a potential of sixteen reference objects formed by the larger square wafers, method 6 yielded five bonded references and method 8 yielded nine.

Table 4.4. Bonding attempt order of operations.

Step	Bonding Order of Operations							
	1	2	3	4	5	6	7	8
1	Piranha	Piranha	Piranha	Piranha	HF Dip	HF Dip	Piranha	HF Dip
2	Rinse	Rinse	Rinse	Rinse	Rinse	Rinse	Rinse	Rinse
3	Dry	Furnace	HF Dip	HF Dip	Piranha	Piranha	HF Dip	Piranha
4	Furnace		Rinse	Rinse	Rinse	Rinse	Furnace	Rinse
5			Dry	Furnace	Dry	Furnace		Dry
6			Furnace		Furnace			Plasma
7								Rinse
8								Dry
9								Furnace

4.2 Reference Object Design

Designing the reference object involved the geometric feature patterns and the protective packaging. The geometric features are separated into patterns consisting of linear features and curved patterns. Silicon reference objects are built from four silicon wafers layers, with each layer having one type of pattern etched into it. They are bonded so that the outward facing surfaces contain features. Since the intent of the reference object is for X-ray CT systems, an added layer of protection is added via a shell made from glass and epoxy. Figure 4.9 shows a solid model representation of the design.

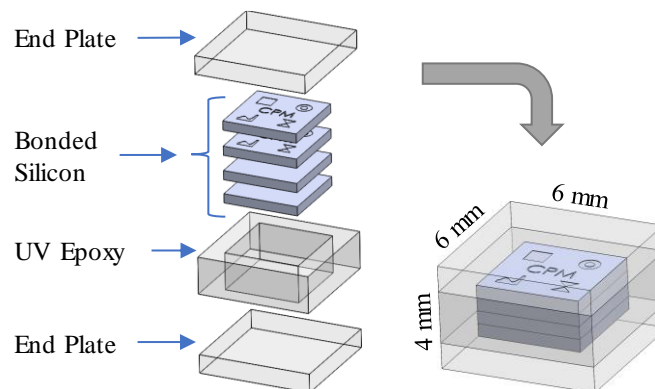


Figure 4.9 Reference object design with a protective shell. Modified from [160].

4.2.1 Geometric Features

Two patterns used in the reference object were drafted using Autodesk® AutoCAD. Within each pattern are features that fall under the category of linear or curved. Linear patterns contain only shapes drawn with straight lines and curved patterns have at least one edge formed by a polynomial. Both patterns are $4 \times 4 \text{ mm}^2$ regions and include a 1 mm long scale bar and the UNCC crown. On the photomask used for lithography, the patterns are spaced 0.3 mm apart to provide a path for dicing the wafer without damaging the patterned areas. CAD drawings of the linear and curved patterns are shown in Figure 4.10 and Figure 4.11 respectively. Features within each pattern are given a name and initials for tracking measured data, see Table 4.5. Majority of the features come in different scaled sizes and to further help with tracking, the largest feature is designated as 1, with the integer increasing as the size decreases. For instance, the largest of the four mirrored triangles is ‘MT1’ and the smallest is ‘MT4’.

Table 4.5. Feature names and their initials.

Feature Name	Initials	Feature Name	Initials
Variable Lines	VL	Quadratic	QC
Periodic Lines	PL	Symmetric Cubic	SC
Periodic Squares	PS	Asymmetric Cubic	AC
Concentric Squares	CS	Chain-link	CL
Barbells	BB	Scale Bar	SB
Mirrored Triangles	MT	UNCC Crown	UNCC
Siemens Star	SS		

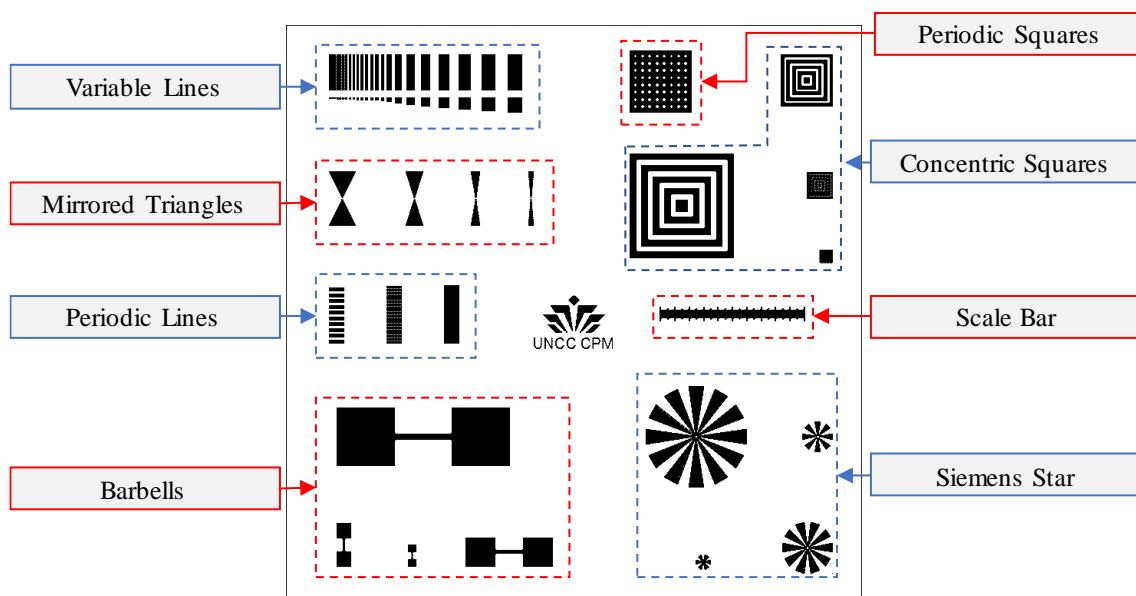


Figure 4.10. Linear features containing only straight lines. Dark areas represent etched sections.

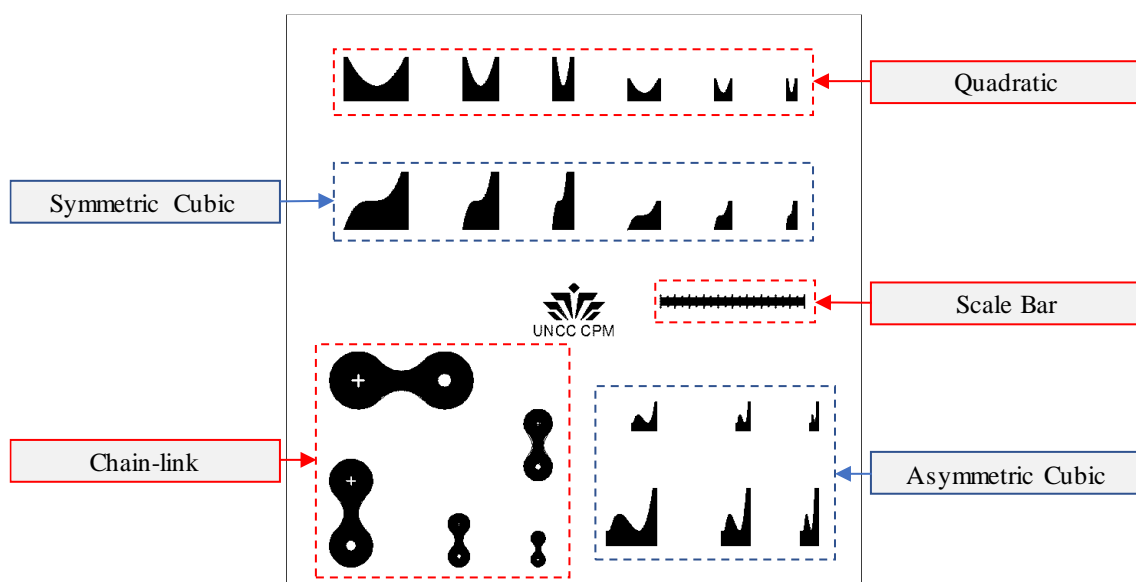


Figure 4.11. Curved features containing at least one polynomial edge. Dark areas represent etched sections.

4.2.1.1 Linear Features

Within the linear pattern are seven features unique to that pattern, listed in Table 4.6. Variable line features are a single pair of rectangular and square shapes. From line widths of 5 μm to 20 μm the size increases in 1 μm steps and from 20 μm to 100 μm the size increases in 10 μm steps. Periodic lines are rectangles with equivalent line widths and spacing. The concentric squares start with an etched center square and alternate between etched lines and unetched surfaces. A single set of periodic squares exist with the area surrounding the squares etched. Barbells are formed by two squares linked together with a rectangle that is of equivalent length as the squares, but 10% of the width. The last sets of features are composed of triangles, one set with two opposing triangles called ‘mirrored triangles’ and the other consisting of a polar array of triangles known as a ‘Siemens Star’.

Table 4.6. Linear features and their nominal dimensions.

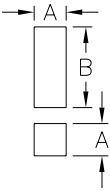
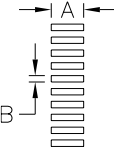
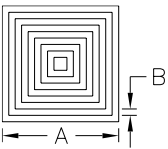
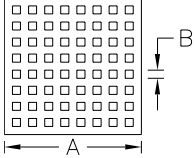
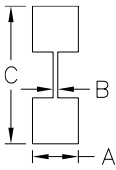
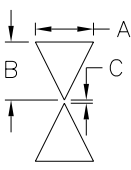
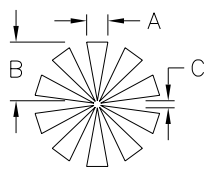
Variable Line	Periodic Line	Concentric Squares	Periodic Squares
			
A = 5:1:20, 20:10:100 μm B = 250 μm	A = 100 μm B = 5, 10, 20 μm	A = 720, 360, 180, 90 μm B = 40, 20, 10, 5 μm	A = 425 μm B = 25 μm

Table 4.6 (continued).

Barbell	Mirrored Triangle	Siemens Star
		
A = 400, 200, 100, 50 μm B = 40, 20, 10, 5 μm C = 1200, 600, 300, 150 μm	A = 180, 120, 60, 30 μm B = 180 μm C = 10 μm	A = 100, 60, 40, 30 μm B = 340, 165, 95, 45 μm C = 20, 20, 20, 10 μm Spokes: 12, 10, 8, 6

4.2.1.2 Curved Features

The second set of patterns are four sets of features that contain a second- or third-degree polynomial as an edge, shown in Table 4.7. Unlike the linear features, these features are unique to this reference object design. Features designated as ‘quadratic’ or ‘cubic’ are made in three sizes that come as both full-scale and half-scale versions, and the curves are connected at the ends by straight edges to be used as datums. Polynomial coefficients and ranges are listed in Table 4.8. Half-scale versions are not included in Table 4.7 or Table 4.8. The fourth shape is called the chain link and is a continuous curve. Inside the major arcs are cross and circle fiducials. Shapes with polynomial edges were created because curves are sensitive to distortions. Some anticipated sources of distortion come from the detector, geometric alignment, beam hardening, artifacts and compensation, source blur, drift (thermal, focal spot), and errors in the mechanical systems [69,102,161–163]. Origins of some distortions, such as distortion found in flat-panel detectors, have yet to be determined [164].

Table 4.7. Curved features and their nominal dimensions.

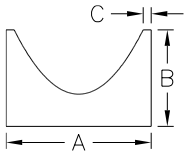
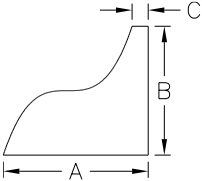
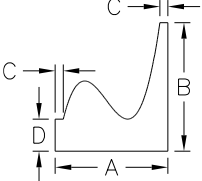
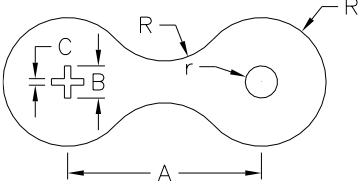
Quadratic	Symmetric Cubic	Asymmetric Cubic	Chain Link
			
A = 450, 250, 150 μm B = 300 μm C = 25 μm	A = 450, 250, 150 μm B = 400 μm C = 50 μm	A = 350, 200, 125 μm B = 400 μm C = 25 μm D = 100 μm	R = 200, 150, 100, 75, 50 μm r = 50, 37.5, 25, 18.75, 12.5 μm A = 600, 450, 300, 225, 150 μm B = 100, 75, 50, 37.5, 25 μm C = 20, 15, 10, 7.5, 5 μm

Table 4.8. Polynomial coefficients for the full-scale features. $y = Ax^3 + Bx^2$

Feature	Range	A	B
Quadratic	[-200, 200]	0	0.005
	[-100, 100]	0	0.01
	[-50, 50]	0	0.02
Symmetric Cubic	[-200, 200]	2.5e-5	0
	[-100, 100]	2.0e-4	0
	[-50, 50]	1.6e-3	0
Asymmetric Cubic	[-200, 100]	1.0e-4	0.02
	[-100, 50]	8.0e-4	0.08
	[-50, 25]	6.4e-3	0.32

4.2.2 Protective Packaging

Damage and contamination of the silicon used for reference object can occur during handling, cleaning, and mounting. To prevent damage, and therefore a reduction in the effectiveness of the reference object, a protective packaging (or shell) encases the silicon reference. Surrounding the silicon in an epoxy is undesired as the epoxy would fill the externally exposed features. The solution is to add end plates, preferably made from fused silica glass, that sandwich the silicon. To secure the end plates in place, UV cure epoxy (Norland Optical Adhesive 61) fills the gap along the side of the silicon. The epoxy is

chosen for its transparency, adhesion to glass and silicon, uncured material can be removed by acetone and isopropyl alcohol, and it is resistant to most solvents after cure. If removing the cured epoxy is required, some solvents and photoresist strippers are available that can remove the epoxy without damaging the silicon reference.

To construct the protective shell, a combination mold and clamp was fabricated (Figure 4.12). The clamp design consists of three plastic printed frames (top, center, and bottom), 3.18 mm thick quartz discs, a molded PDMS insert, 1 mm thin PDMS seals, and fasteners. The center frame component of the clamp is intentionally designed 0.05 mm thinner than the PDMS insert and finished reference object. This was done so that when the clamp is fastened, the clamping force is transmitted through the end plates and the silicon reference. By transmitting the force through the end plates and silicon, epoxy is prevented from contaminating the silicon-glass interface. Holes in the sides of the center frame guide two syringe needles for piercing the PDMS insert. One syringe is used to create a small vacuum while the second syringe filled with epoxy is injected, minimizing trapped air bubbles.

PDMS is chosen as the mold material for three reasons; its compliance, it does not form a bond with the UV epoxy, and is transparent to the UV light. To create the PDMS insert and seals, Sylgard 184 (Dow Corning) is mixed in a 10:1 by weight ratio of base to curing agent. After mixing, the PDMS is degassed in a vacuum chamber before pouring into 3D printed plastic molds. Curing is accelerated by baking at 70 °C for at least 1.5 hr.

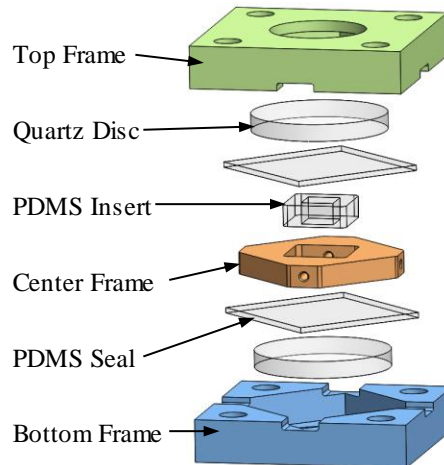


Figure 4.12. Epoxy curing clamp assembly. Modified from [160].

4.3 Reference Object Fabrication

Fabrication of the reference object can be divided into three distinct phases. A brief outline of the process is illustrated in Figure 4.13. Wafer etching is the first phase that includes; wet cleaning, thermal oxide growth, lithographic processes, wafer etching, and dicing. Next is the direct bonding phase where the wafers undergo wet cleaning, oxygen plasma, stacking, annealing, and dicing. The last phase is the final assembly where the protective shell composed of glass end plates and epoxy are fabricated around the silicon reference.

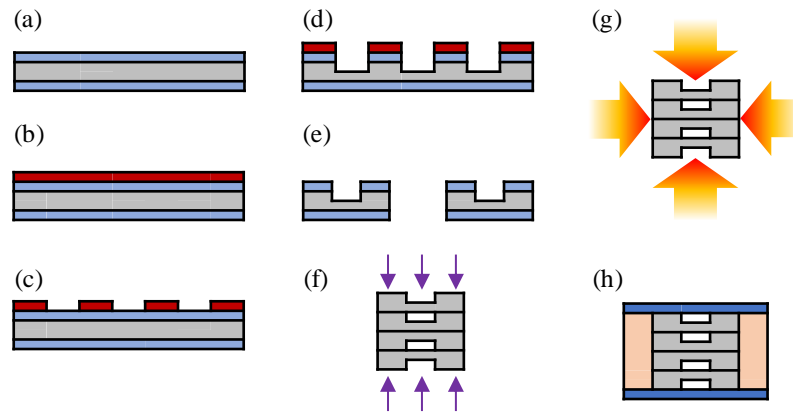


Figure 4.13. Reference object fabrication process: (a) protective thermal oxide layer is grown; (b) photoresist is spin coated onto wafer; (c) photoresist is exposed and developed; (d) wafer is etched; (e) wafer is diced into square wafers; (f) four square wafers are stripped of oxide, cleaned with ‘Piranha’ and oxygen plasma, stacked, and clamped; (g) stacked wafers are annealed twice and diced into individual references; (h) glass and epoxy protective shell is built around the silicon reference.

4.3.1 Wafer Etching

The wafer etching phase is the first of the three fabrication phases. In this section the methodology for cleaning, growing a thermal oxide layer, lithographic processes, silicon etching, and dicing are described in detail.

Throughout the manufacturing process, a 3:1 mixture of ‘Piranha’ clean is used to remove biological contaminants from the silicon wafers. Inside an ISO Class 5 clean room located in Grigg Hall, 100 mm diameter and 0.5 mm thick prime grade silicon wafers are cleaned with ‘Piranha’ in a quartz container for 15 min. The ‘Piranha’ solution is lightly agitated to dislodge air bubbles from the wafer surfaces. Wafers are then transferred to second quartz container with continuous flowing DI water for 3 min. Individually the wafers are removed from the DI water and dried with a nitrogen air gun before being storing in the original container.

The container was sealed with tape and wrapped inside multiple bags and transported to the clean room located in UNCC's Cameron Hall where a protective layer of silicon dioxide is thermally grown. This part of the process was completed by the clean room manager John Hudak and involved an HF dip to strip the native oxide, DI water rinse, and drying using a multi-wafer spin drier. Wafers are loaded onto a quartz boat and processed in an oxidation furnace for 30 min at 1000 °C. After oxidation the wafers are returned to their container, resealed, and brought back to the ISO 5 clean room. One wafer was randomly selected, and a 40 nm oxide thickness was measured by a JA Woollam VASE® ellipsometer.

In preparation for lithography, wafers were cleaned in 'Piranha', DI rinsed, and nitrogen air dried. One at a time, the wafers were immersed for 30 s in SurPass 4000 (DiscChem, Inc.), a photoresist adhesion promoter, before another DI water rinse and nitrogen dry. To further improve photoresist adhesion, wafers are subjected to a dehydration bake using a Brewer Science Cee 1100FX hotplate (Figure 4.14). The hot plate temperature is set at 115 °C and with the timer set to 60 s. All subsequent uses of the hot plate are performed with the same temperature and time settings.

After the wafer cools, it is centered onto the vacuum chuck of a Brewer Science Cee 200 spin coater (Figure 4.16). Positive photoresist (Dow Electronic Materials Microposit S1813 G2) is poured from a beaker onto the center of the wafer until a 50 mm diameter puddle forms, careful to avoid air bubbles. The spin coater is programmed to run the process listed in Table 4.9. Step one of the program rotates the wafer slowly to spread the photoresist across the entire wafer surface. The second step spins the wafer at a higher

RPM to create a uniform layer of 1.3 μm thickness. Coated wafers are soft baked on the hot plate.

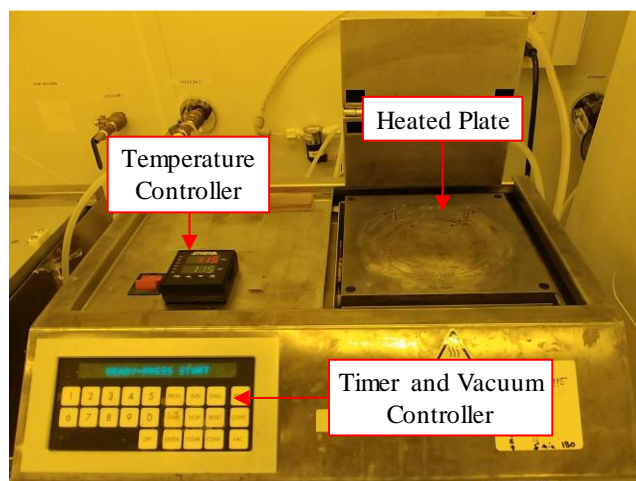


Figure 4.14. Brewer Science Cee 1100FX hotplate used for the dehydration, soft, and post-exposure bakes.

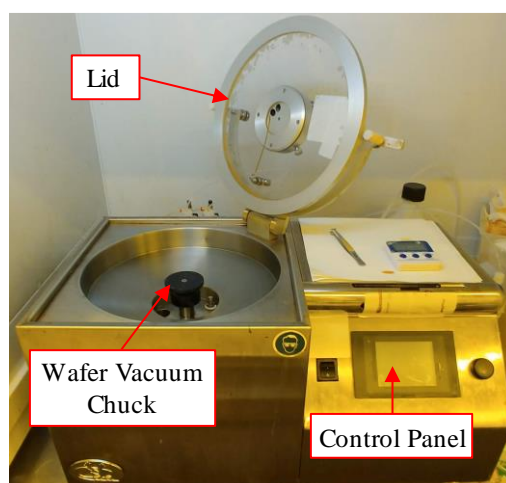


Figure 4.15. Brewer Science CEE 200 spin coater for adding photoresist.

Table 4.9. Spin coater settings for Microposit S1813 G2 photoresist.

Step	Accel ($\text{rpm}\cdot\text{s}^{-1}$)	Run Time (S)	Speed (RPM)
1	100	10	900
2	1000	60	3800

The photomask was manufactured by Martin Photomask, Inc. and is made from soda lime glass with chrome patterns. The contact mask aligner (Quintel Ultraline Q-4000) shown in Figure 4.16 uses UV light exposure to transfer the pattern of photomask into the photoresist. Before exposing wafers, the UV lamp intensity measured at $11 \text{ mJ}\cdot\text{cm}^{-2}\cdot\text{s}^{-1}$ using a meter optimized for 365 nm wavelength. Printing mode on the aligner was set to ‘pressure contact’, which uses the piston to lift the chuck holding the wafer into contact with the photomask. The operating mode was set to ‘expose pressure’ which applies pressure between the chuck and the wafer, forcing the wafer to conform to the photomask.

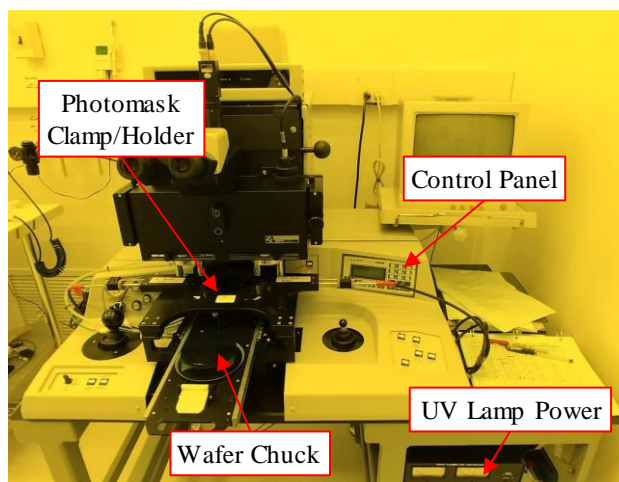


Figure 4.16. Quintel Ultraline Q 4000 for transferring the mask design into the photoresist using ultraviolet light.

The photomask is loaded onto the aligner and locked in place via vacuum. A wafer is placed on a vacuum chuck and is up raised to align the plane of the wafer surface to the photomask. Although the photoresist datasheet provides a dosage value for estimating exposure time, for a more accurate feature replication, dosage tests were previously carried out to find an optimal exposure time of 4 s. After exposure the wafers are post-exposure baked on the hot plate. With the positive photoresist, the exposed photoresist is removed by immersing the

wafer in a TMAH developer (Dow Electronic Materials Microposit MF-319). Agitating the developer is required and after 60 s, the wafer is rinsed with DI, dried with nitrogen, and stored.

As discussed in section 4.1.2.1, etching the silicon was carried out using the Surface Technology Systems Advanced Silicon Etcher (STS ASE, model 52143) that is located in the ISO Class 6 clean room bays. To create vertical walls, the ASE is operated in switching mode to perform the DRIE Bosch process that swaps between passivating and etching. After the etching recipe is finished, a second plasma recipe using oxygen strips away the majority of the remaining photoresist. Settings for the etching, passivation, and oxygen strip are given in Table 4.10.

Table 4.10. Silicon DRIE (passivate and etch) and photoresist strip settings.

Parameter	Passivate	Etch	Strip	Unit
Cycles	27		1	-
Cycle Time	7	9	180	s
Chamber Pressure	12	12	45	mTorr
C ₄ F ₈ Flow	85	0	0	cm ³ ·min ⁻¹
SF ₆ Flow	0	130	0	cm ³ ·min ⁻¹
O ₂ Flow	0	13	45	cm ³ ·min ⁻¹
Coil Power	600	600	300	W
Platen Power	0	30	0	W
Bias Frequency	0	13.56	13.56	MHz

Outside the clean room, wafers are diced using a Micro Automation 1100 saw. To hold the wafer while being diced, a piece of blue pressure sensitive dicing tape is held taut by two press-fit plastic grip-rings and the wafer is affixed near the center of the tape on the unetched side. Refer to Figure 4.17 for pictures of the blue tape, grip-rings, and dicing saw. A 0.1 mm thick resin-bonded diamond dicing wheel, rotating at 28000 rpm, is zeroed

relative to the wafer vacuum chuck by touching off on a metal plate. After loading the silicon wafer, the 0.3 mm streets etched into the wafer are aligned to the virtual lines of the camera alignment system. A test cut is made with a $5 \text{ mm} \cdot \text{s}^{-1}$ feed rate in an area outside the usable area of the wafer. This test cut is for adjusting the virtual lines to the edges of the cut and for measuring the width of cut. Moving the wafer to the first saw street, it is aligned to one edge of the street and then centered to make the first cut. Subsequent cuts are made every 17.2 mm before the wafer is rotated 90° and diced. A total of sixteen $17.1 \times 17.1 \text{ mm}^2$ wafers are cut from a single 100 mm diameter wafer. The wafers are left attached to the blue tape and returned to the clean room.

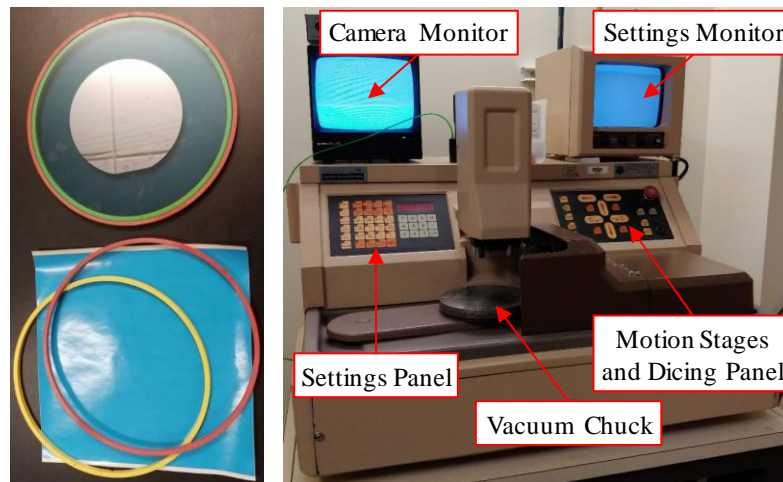


Figure 4.17. Pressure sensitive dicing saw tape with grip-ring and the Micro Automation 1100 dicing saw for cutting wafers.

4.3.2 Direct Silicon Bonding

The second phase of the reference object fabrication is the direct bonding of square wafers produced in the previous section. Within this section, preparation of the wafers for bonding using wet and plasma cleaning processes, the stacking and clamping of layers, thermal annealing, and dicing operations discussed in detail.

In the ISO class 5 clean room, four of the square wafers are loaded into a wafer handling boat (Figure 4.18) and their protective thermal oxide is stripped by diluted HF for 60 s. After rinsing the parts in DI wafer for 2 min, they were visually inspected by looking for water droplets attached to the surface. If no evidence of water on the surface was seen, then the surface was hydrophobic and ready for the next step. If water droplets were seen attached to the surface, then the surface was still hydrophilic and the oxide stripping process was repeated. To clean and form a thin oxide layer, the wafers were put in ‘Piranha’ for 15 min, rinsing with DI water for 2 min, and dried with nitrogen.

Using a different etching system, the STS Advanced Oxide Etcher (ASE, model MP0563) in the ISO class 6 clean room, the square wafers were placed onto a 100 mm silicon wafer that acted as a carrier. An oxygen plasma clean using the settings in Table 4.11 was performed on the exposed surface. After flipping the square wafers over, the oxygen plasma clean was repeated on the opposite side of the part before returning the pieces to the wafer boat. One last rinse in DI water for 2 min and the square wafers are ready for bonding.

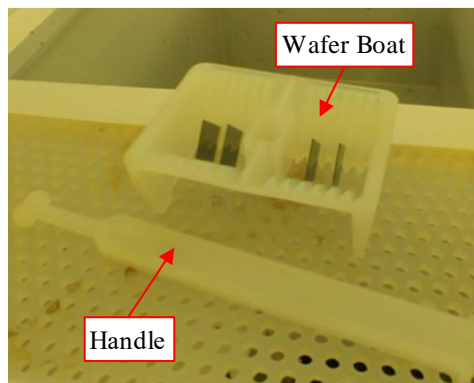


Figure 4.18. Wafer boat and handle for handling the square silicon wafers.

Table 4.11: Oxygen plasma activation settings using the STS AOE system.

Setting	Value	Unit
Time	60	s
Coil Power	400	W
Platen Power	100	W
O ₂ Flow	30	cm ³ ·min ⁻¹
Pressure	50	mTorr
RF Frequency	13.56	MHz

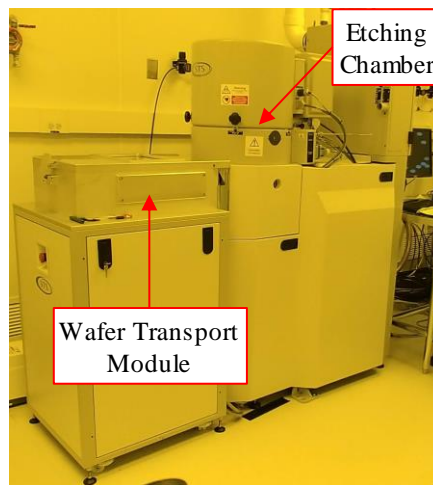


Figure 4.19. STS Advanced Oxide Etcher used for oxygen plasma cleaning prior to bonding.

A $17 \times 17 \times 1 \text{ mm}^3$ fused silica piece is set as a base layer onto a plastic v-groove mechanical aligner, Figure 4.20. Using cleaned tweezers and touching only a small amount of the wafer near a corner, the square wafers are removed from the DI wafer, blown dry, and oriented into the aligner. Wafers are oriented with the first two layers feature-side down and the last two feature-side up. When the last wafer is placed, a second fused silica layer is placed at the top of the stack and the stack is lightly compressed to increase the number of bonds that can help keep the stack aligned as it is transferred to the steel clamp. Carefully the stack is placed into the steel clamp, Figure 4.20, and a truncated steel sphere is centered

on top of the stack. Two small v-blocks are pinched together around the stack reduce misalignment as the 1/4-20 inch screw is torqued to 1.4 N·m.

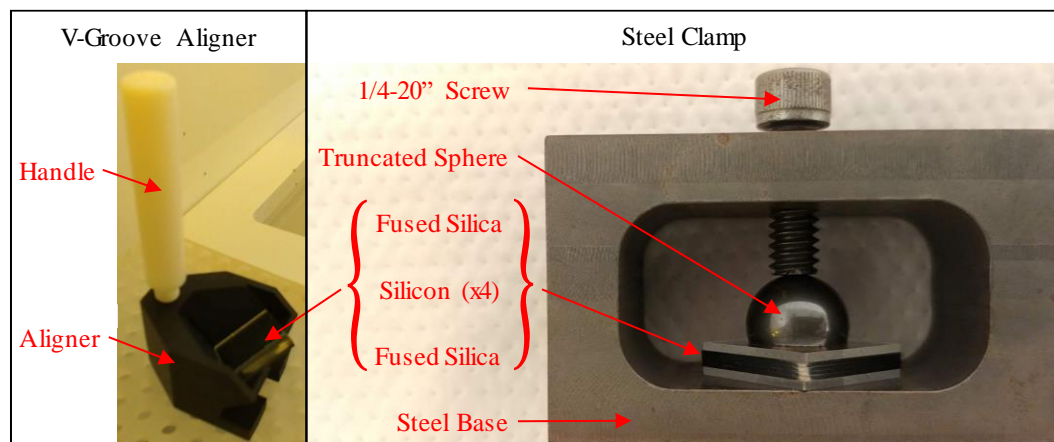


Figure 4.20. V-groove mechanical aligner and the steel clamp.

A four-step programmable vacuum furnace (Ney Centurion Q50, see Figure 4.21) anneals the silicon to strengthen the bond. The programmable steps in order are; a low temperature drying time, a temperature ramp up, a high temperature hold time, and cool down time. Provided in Table 4.12 is a complete list of settings and the two annealing programs. The vacuum is forced to continuously run by setting the vacuum stop temperature higher than the high temperature and disabling the vacuum stop time. A 400 °C anneal is run first with the parts clamped. After cooling down, the silicon and fused silica are removed from the clamp and returned to the furnace for a second anneal at 1100 °C.

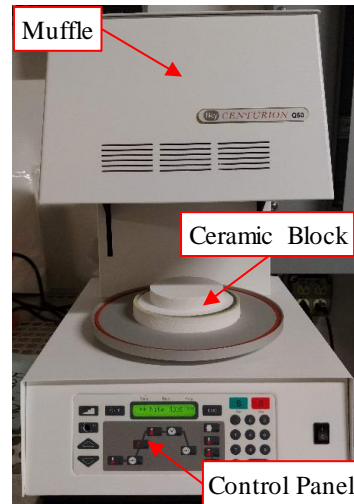


Figure 4.21. Programmable vacuum furnace.

Table 4.12: Programmed furnace settings using for direct bonding.

Settings	Prog. 1	Prog. 2	Unit
Low Temperature	100	100	°C
Drying Time	0	0	min
Ramp Rate	25	50	°C·min ⁻¹
High Temperature	400	1100	°C
Hold Time	90	90	min
Cooling Time	90	90	min
Vacuum Pull	125	125	°C
Vacuum Level	740	740	mm Hg

Unlike dicing of the 0.5 mm thick and 100 mm diameter wafer, the bonded stack does not have adequate surface area to safely be held by blue tape alone. To solve this problem, a carrier wafer and the bonded stack are heated on a hot plate to 110 °C and both coated on one side with a soluble wax (Figure 4.22). The waxed side of the bonded stack is then set onto the carrier wafer. After cooling, the carrier wafer can be attached to the blue tape. Using a smaller collet with the 0.1 mm thick blade previously used exposes more

blade necessary to cut through the bonded stack. Rotation speed is set to 28000 rpm and feed rate is set as $0.2 \text{ mm}\cdot\text{s}^{-1}$, the slowest the machine allows. Test cutting near the edge of the carrier wafer is performed to set the position of the virtual lines on the monitor. The stack wafer is rotated to align an outer edge with the virtual lines and then moved to the first cutting position. Cuts are made through the center of three streets, the part is rotated, and the perpendicular streets are cut to separate the sixteen potential bonded references from each other. To remove the references from the carrier wafer, the carrier wafer is pulled off the blue tape and returned to the hot plate. Once the wax is melted, each silicon reference is removed and placed in an acetone bath to dissolve the wax. Successful parts are dried, stored in a membrane case, and returned to the clean room for the next phase in the reference object fabrication. The size of each silicon reference object is $4.2 \times 4.2 \times 2 \text{ mm}^3$.

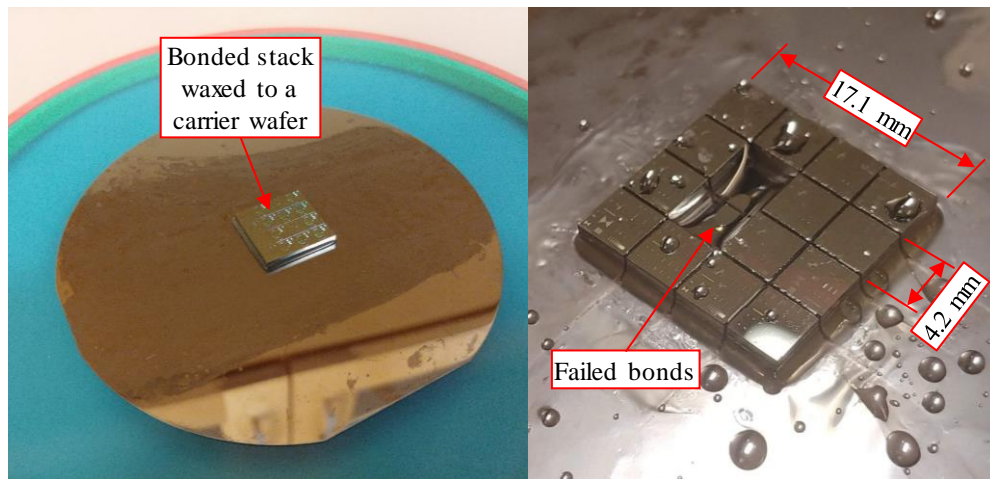


Figure 4.22. A bonded stack secured to a carrier wafer and ready for dicing. Modified from [160].

4.3.3 Final Assembly

The final phase is assembly of a protective shell made from epoxy and glass that is constructed around the silicon reference. Discussed in this section are the cleaning procedures, clamp preparation, and epoxy filling and curing procedures.

A bonded silicon reference and two glass end plates are ultrasonically cleaned with acetone for 5 min and then with isopropyl alcohol. The silicon part is placed into the PTFE carrier shown in Figure 4.23 and the oxide layer is stripped with diluted HF for 1 min and washed with DI water for 2 min. Adding two glass end plates to the carrier, the parts are cleaned in ‘Piranha’ for 15 min.

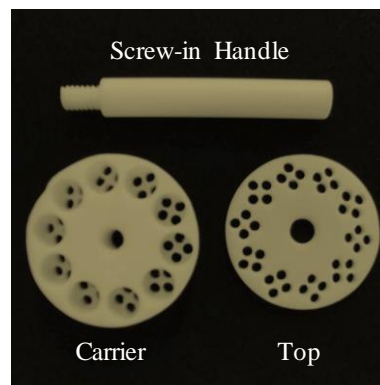


Figure 4.23. Reference object carrier made of PTFE used for cleaning fused silica and bonded silicon reference objects prior to final assembly.

While the silicon and end plates are being cleaned by the ‘Piranha’ solution, the clamp shown in Figure 4.12 and is prepared as shown in Figure 4.24. The bottom half of the clamp is assembled up to the PDMS insert. Two syringe needles of diameter 1.27 mm (18-gauge) are inserted through opposing ends of the center frame to pierce through the PDMS insert. Piercing the PDMS with the needles may cause the needles to become clogged. After removing the syringe needles from the PDMS, the blockage can be removed

by pushing in the plunger of the syringe. A 1 mL syringe extracts 0.5 mL of epoxy from the epoxy bottle and is then held with the needle pointing upwards to pull in air to create an air bubble. Air trapped within the epoxy is dislodged by tapping the side of the syringe. When all visible air bubbles have merged with the air below the needle, the plunger is pushed to evacuate the air out until a drop of epoxy emerges from the needle. With the epoxy syringe primed and the 10 mL syringe's plunger pushed in, they are reinserted into the center frame and PDMS insert.

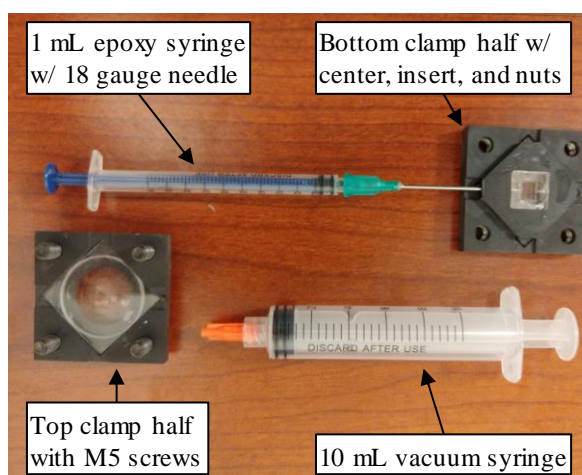


Figure 4.24. Epoxy clamp halves assembled and the syringes in the filling process. Needle for the 10 mL syringe is not shown.

Once the glass end plates and silicon reference are finished in the ‘Piranha’ solution, they are rinsed in DI water for 2 min and blown dry with nitrogen. A silica end plate is picked up with tweezers and pushed into the PDMS insert until it is seated against the PDMS seal at the bottom. Next, the bonded silicon reference is picked up by its sides using tweezers to avoid contaminating or damaging the two faces with features. The bonded reference is placed feature-side down onto the silica end plate and centered within the PDMS insert. Pressing the needles of the syringes against the sides of the silicon helps

prevent the silicon reference from shifting as the second silica end plate and the rest of the clamp are assembled. After fastening the four nuts and bolts of the clamp, the plunger of the 10 mL vacuum syringe is pulled to create a vacuum. Slowly pushing in the plunger of the epoxy syringe fills the cavity surrounding the silicon reference. As the syringes are slowly pulled from the clamp, epoxy continues to be pushed in until the needle ends are halfway removed from the PDMS insert. Curing the epoxy takes 2 hr in the UV exposure system (UVO-Cleaner Model 42, see Figure 4.25).



Figure 4.25. UVO-Cleaner Model 42 used to cure the UV epoxy.

CHAPTER 5: PARAMETER EXTRACTION OF MEASURED DATA

5.1 Data Collection

Following the manufacturing of the reference object, to evaluate its use in practice, it was necessary to measure it with an X-ray CT system and compare the data to features on the wafer prior to assembly that were measured by coherence scanning interferometry (CSI). The first two sections of this chapter discuss the scanning parameters for the CT and CSI systems. An algorithm for extracting the feature of interest from a set of data for measurements is described. Ending the chapter are some general observations about the reference object construction and CT reconstruction, and examples of measurements taken from the two instruments (CT and CSI).

5.1.1 X-ray CT

Without an industrial X-ray CT system at the college, a reference object was stored in membrane containers and shipped to Dr. Herminso Villarraga-Gómez at Nikon Metrology. Using a Nikon XT H 225 ST system with a Perkin Elmer 1621 detector, three reference objects were scanned with 2880 projections and settings listed in Table 5.1. Nikon's CT Pro 3D software was used for the Ram-Lak filtering and tomographic reconstruction. The grayscale voxel data is output from Nikon's software and imported into VGStudio MAX 3.2 for automatic surface determination using a global ISO50 (or ISO-50%) threshold. ISO50 thresholding takes the histogram of the voxel grayscale values and sets a threshold half way between two peaks, with the peaks representing materials and background [98]. See Figure 5.1 for a picture of the reference object in front of the X-ray source and an example of ISO50 thresholding.

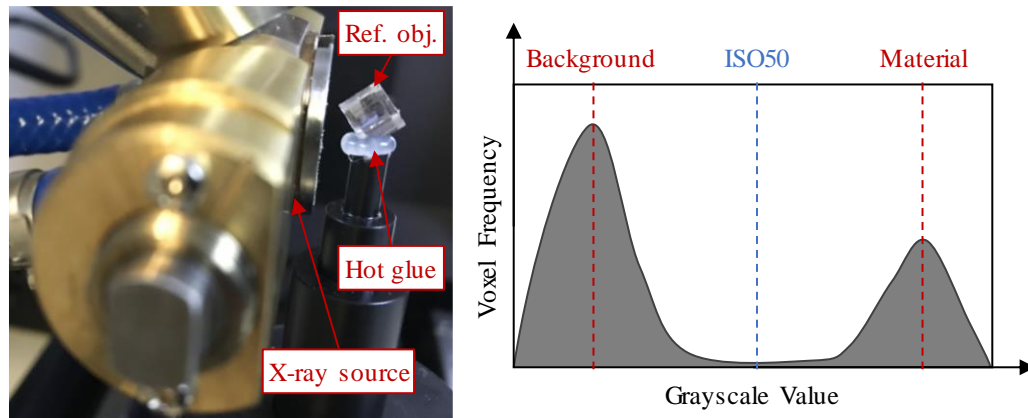


Figure 5.1. (Left) X-ray source with reference object and (right) ISO50 thresholding example. Thresholding is used to differential between different materials and from the background. In the example, all voxels left of the ISO50 line are background and all voxels right of the ISO50 line are material.

Table 5.1. Nikon XT H 225 ST scan settings.

V (kV)	I (μ A)	Fs (μ m)	Vx (μ m)	SOD (mm)	SDD (mm)	It (ms)	A	G (dB)
80	47	3	3.8	15	815	708	4	5

V = tube voltage, *I* = tube current, *Fs* = focal spot size, *Vx* = voxel size, *SOD* = source-to-object distance, *SDD* = source-to-detector distance, *It* = integration time in milliseconds, *A* = averaged images, *G* = gain.

To capture data in a format that can be processed through an algorithm developed in MATLAB®, the cross-sectional images of a region of interest containing patterned features was saved as a TIF image with a scale bar. Based on the image's superimposed scale bar, spatial sampling was measured as $0.47 \mu\text{m} \cdot \text{pixel}^{-1}$. For each set of features, eleven cross-sectional images were taken equidistant apart from the polished surface at the top of the feature to the etched surface at the bottom of the feature. Of the eleven images, some of the images captured closest to the top and bottom surfaces had to be excluded from measurements as the feature extraction was complicated by the density gradient. The step height measurement made within VGStudio used the built-in tools to generate two least

squares fit planes, one for the etched surface and the other for the polished surface, within the boundary of the feature being evaluated.

5.1.2 Coherence Scanning Interferometry

Measurements using the Zygo® Nexview™ CSI are from the same wafer used to assemble the reference object measured with the CT system. While measurements are from different parts these patterns from a similar radial distance from the wafer center. Measurements of the curved feature were taken with the 20× and 50× objectives, both paired with the 1× tube lens. Numerical aperture (NA) for the objectives are 0.40 and 0.55 respectively, providing optical resolutions of 0.71 μm and 0.52 μm using the Abbe criteria (Equation 5.1) and λ of 570 nm. Spatial sampling sizes are 0.408 μm·pixel⁻¹ and 0.164 μm·pixel⁻¹. Because some features exceed the size of the field of view, features may require stitching to be captured in full and this will affect the results. During acquisition of data without stitching, an average of three images for each measurement point was used to reduce noise. When acquiring stitched data, no averaging was applied. The four stitching algorithms supplied in the Zygo® Mx™ software (version 7.2.0.10) that were tested are ‘XY Adjust’, ‘Cartesian’, ‘Overlay’, and ‘Adaptive Adjust’. Data from the CSI is saved as a DAT file and imported into MATLAB® for analysis.

$$Resolution = \frac{\lambda}{2 \times NA} \quad (5.1)$$

5.2 Parameter Extraction Algorithm

The software of choice for performing the feature detection and parameter extraction is MATLAB®. Scripts are written to import the data from the different

instruments and process the data with the least amount of user interaction so that the results are repeatable. At the start, processing either the CSI or confocal height data requires the top surface to be leveled or the CT image requires input from the user to calculate the spatial sampling. After converting to a binary image, the data is processed the same way with only the step height measurement neglected for CT data. A flowchart of the algorithm is provided as Figure 5.2. Discussion of the algorithm is focused on the curved features.

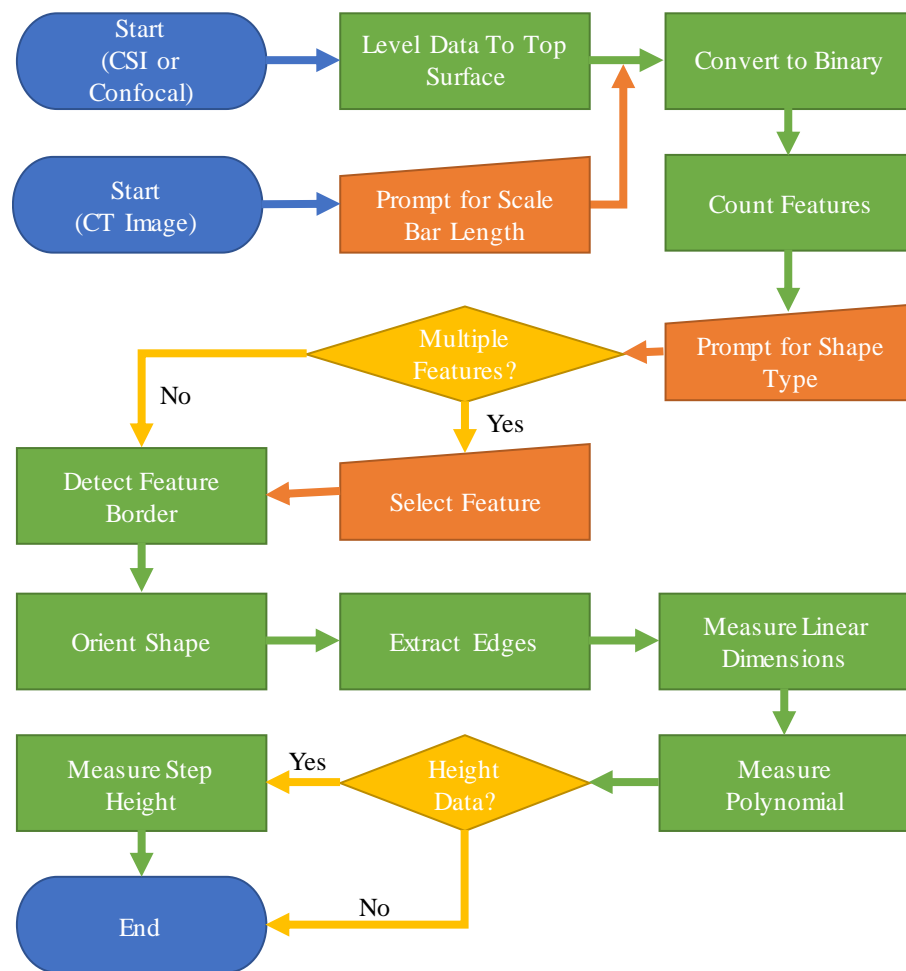


Figure 5.2. Flowchart of the feature extraction and measurement algorithm.

5.2.1 Data Type Preparation

To get to the point where the different data types are processed through the same code, the starting script is different. Image data obtain from the CT reconstructions requires some help to calculate the spatial sampling and determine if an average of multiple images is used. Height data, such as that from CSI and confocal systems must have the top surface leveled. In this section the start of each script through to the feature select is described.

5.2.1.1 CT Image

For processing CT images the first step of the script prompts the user to select one or more images, if multiple images are selected an average of the images is generated. The script then measures the length of the software scale bar superimposed on the TIF image in pixels, and ask the user to input the scale bar length for conversion to spatial sampling. Similar to the ISO50 method discussed in section 5.1.1, a histogram of the grayscale values is created. The midpoint between the two largest peaks is calculated. Pixels with grayscale values greater than the midpoint are converted into values of one representing the silicon material. Where the feature's cavities exist, the pixels have a grayscale values less than or equal to the midpoint and are converted to zero. A binary image is now formed, and the script continues to the common code for feature detection and orientation.

5.2.1.2 CSI or Confocal

When the CSI or confocal script is started, the data first goes through a leveling process. The leveling process is performed in two steps, a coarse level and a fine level. The function 'autolevel.m' is called to perform the leveling by creating a histogram of the data and finding the midpoint between the two largest peaks. A binary image is created where the height values less than or equal to the midpoint are zero (etched silicon surface) and

values greater than the midpoint are one (polished silicon surface). Zero values are then changed to NaN (not a number) and element-wise multiplication of the NaN-one matrix and height matrix is performed. The zeros of the binary image are replaced with NaN because NaN values are ignored during the least squares plane calculation. From the non-NaN values, coefficients of a least squares fit plane are calculated to create a plane that is removed from the height matrix. Refer to 'plane_fit.m' in the appendix for the least squares plane fit code. The coarsely leveled height data is sent through the leveling function a second time for fine leveling. In some optical measurements, ringing artifacts on the edge of the feature are seen (Figure 5.3). To prevent the ringing artifact from influencing the leveling process, after the binary image is created and before converting zeros to NaN, the area with zeros are positively padded (enlarged) by 10 pixels, see Figure 5.4 for an example. Padding should be adjusted to the smallest distance from the edge that removes ringing. Histogram results before leveling and after each leveling step are shown in Figure 5.5. At this point the binary image is ready for feature detection and orientation.

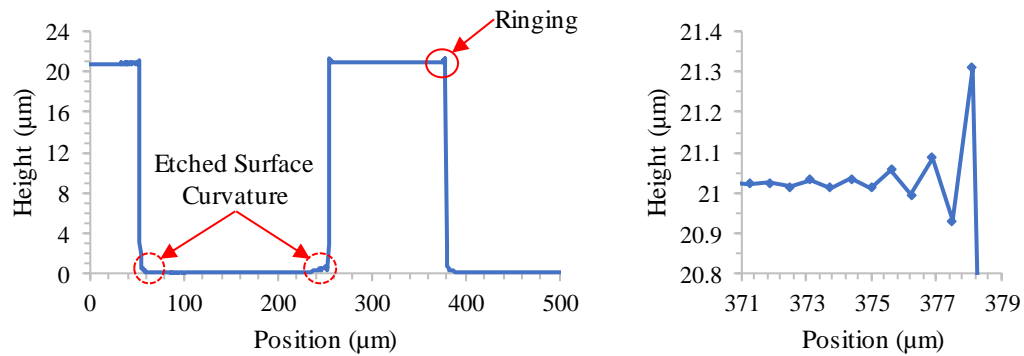


Figure 5.3. Areas targeted by padding are the (a) edge ringing artifact seen in the laser confocal microscope data and (b) the etched surface curvature discussed later in section 5.2.4 on step height measurement.

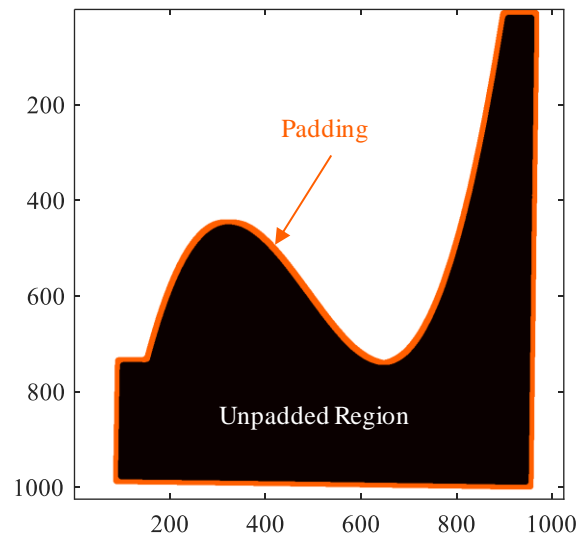


Figure 5.4. Example of positive feature padding that increases the area with zeros.

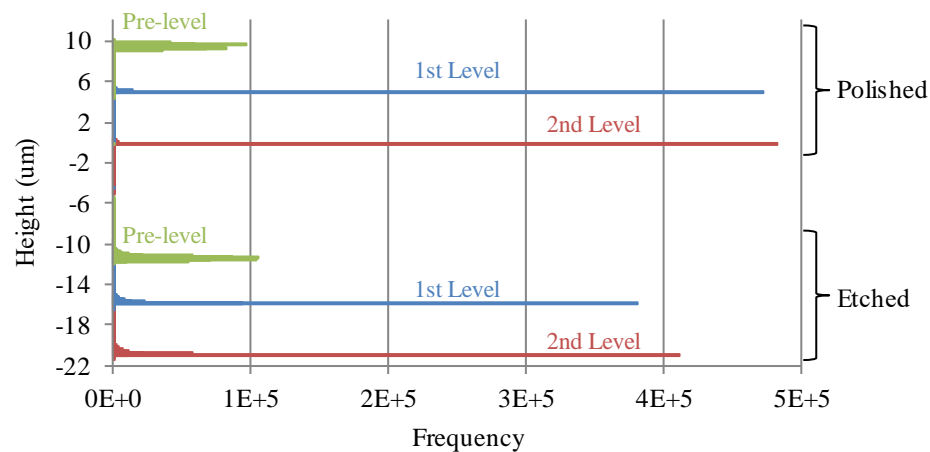


Figure 5.5. Histogram comparison showing the leveling results. An offset has been added to the pre-level and first level for clarity.

5.2.2 Feature Detection and Edge Extraction

Feature detection starts by counting the number of features using the function 'bwlabel'. After the user to select the shape's name from a list, if multiple features are detected, the user picks the feature of interest from the image. Features not selected are

removed from the binary image and then the binary image is processed through the 'bwboundaries' function to detect the feature boundary.

The second- and third-order features of the curved pattern set (Table 4.7) are oriented by first locating the bottom left and top right corner. These two points are found by looking for the furthest two points in the data, Figure 5.6(a). Translating the feature, one point is set on the origin, designated by the star icon in Figure 5.6, and the feature is rotated to set the second point on the x-axis. After checking if the feature needs to be mirrored, a third corner at the bottom left of the shape is located. Two straight edges are extracted from the three corner points, and the feature's bottom edge is determined, fitted with a line, and rotated onto the x-axis. The position of the three corners is refined and any remaining corners are found. With all corner points located, the edges connecting the corners are extracted and truncated to 90% their length. Refer to Figure 5.6 for a graphical representation of the orient and extraction process.

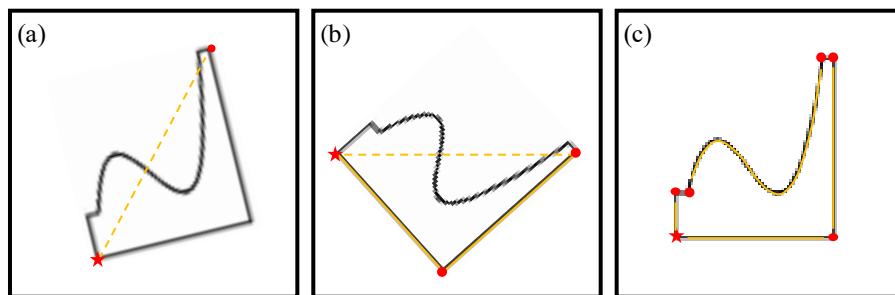


Figure 5.6. Orienting the asymmetric cubic shape. (a) Detecting the origin indicated by the star icon and a second corner point. (b) Translate rotate, detect a third corner, and extract two edges. (c) Align the bottom edge to the x-axis and to extract remaining corners and edges.

Orienting the chain link feature is performed a different way due to the lack of corners. Within the feature are two fiducials, a circle and a plus sign. These fiducials are

detected and translated to set the plus sign fiducial at the origin. The feature is rotated to set the circle fiducial on the x-axis. After translating a second time to move the center of the feature to the origin, the four arc edges are extracted without truncating.

5.2.3 Spatial Measurement

Spatial measurements are in the form of linear dimensions, circle fitting, and second- or third-order polynomial fitting. Straight edges are fitted with least squares fit lines. Distances are measured between parallel lines by drawing a tangent-normal from the side with more data points to the midpoint intersect of the opposing line fit. To measure the radius of the chain link feature, circles are fit using least squares to each of the four arcs. Distances of the data points from the fit circle's centroid are measured.

Second- and third-degree polynomials are measured to determine their fit coefficients and deviation from the fit line and the nominal line. To perform the fit, a MATLAB® function was written called 'polymindv' with inputs for the x and y vectors, the nominal coefficients, degree of polynomial, and a Boolean variable that dictates whether to calculate and use best fit polynomial coefficients or the nominal coefficients. Tangent-normal distances are first measured from the raw data to the theoretical curve for calculating the standard deviation. Next, the script uses MATLAB®'s built-in Procrustes superposition function, without scaling, to determine a linear transform of the data points to fit a theoretical curve. This process is repeated until the difference in standard deviation between the next-to-previous and the current transformations are less than 0.1 nm. Once the fitting algorithm finishes, the root-mean-square (RMS) and the standard deviation of the tangent-normal distance should be close to each other, and often identical values. In Figure 5.7, the number of iterations to minimize the standard deviation for a nominal and

a best fit line are shown. While fitting to the nominal coefficients requires fewer iterations, the difference between the RMS and standard deviation at the start are further apart than for the best fit line. The reason the best fit line takes more loop iterations to reach a minimum is related to the degrees of freedom of the fit. When calculating the best fit line, the coefficients are variables that can change every iteration of the loop while the nominal line's coefficients stay constant.

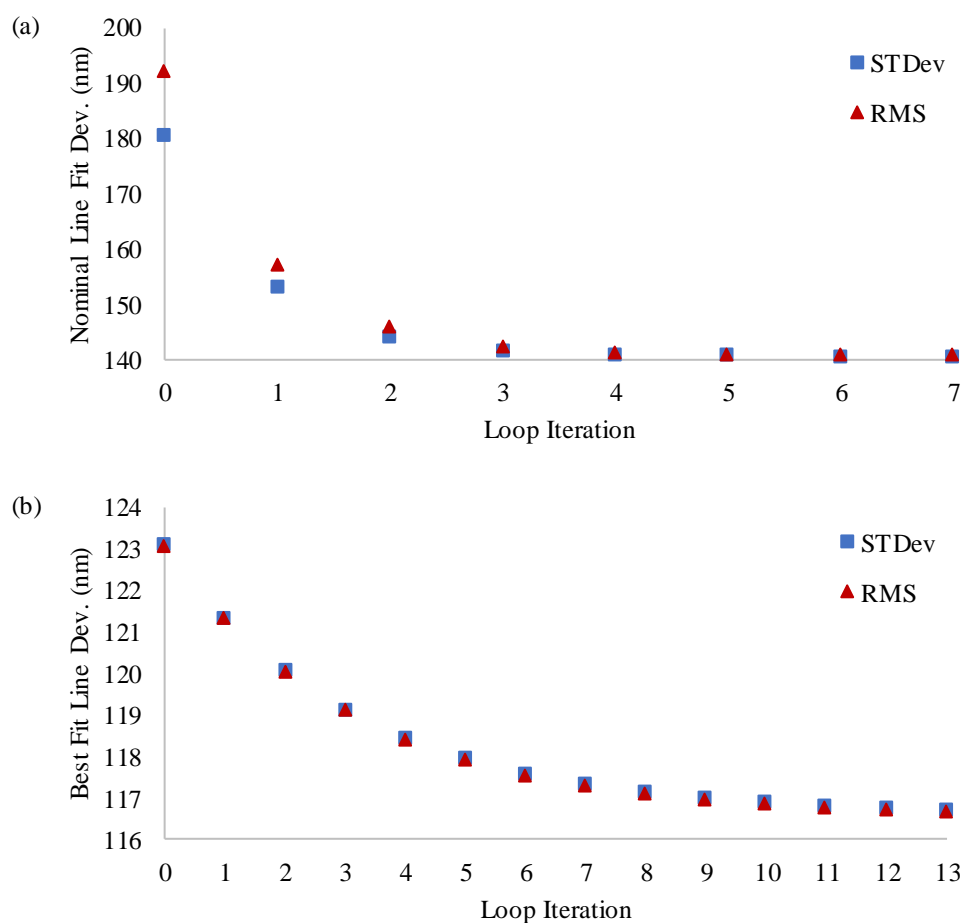


Figure 5.7. Minimization of the deviations when using the (a) nominal line and (b) best fit calculated line. Data is from a CSI measurement of an asymmetric feature.

5.2.4 Step Height

Measuring the step height of the CSI or confocal data is the final part of the algorithm. The polished top surface was leveled to a zero mean and the etched region is negative. An inverse of the binary image is created, setting the top surface to zero and etched surface to one. Recall that the leveling process positively padded the feature to remove the ringing artifact (Figure 5.3). When measuring the step height, negative padding is used to reduce the size of the binary image's feature, thereby avoiding the ringing artifact. Another reason is that the near the edges at the bottom of the etched feature, the etching rate is slower and produces a curved surface instead of a flat, as seen in Figure 5.3. Although the padding length is set to 15 pixels, the padding length should be optimized based on the ringing and etched surface curvature near the edges. After element-wise multiplication of the step height's binary map and the leveled data, values less than zero are used to calculate the mean step height and standard deviation.

5.3 Reference Object Evaluation and Measurements

Having successfully created the reference object manufacturing process, its quality was evaluated visually, with scanning electron microscopy, and by X-ray computed tomography. From the same X-ray CT data, features were extracted to illustrate how the reference object can be compared to measurements made by a calibration tool. In this study, the CSI data used in the comparison is taken from the same feature from different diced wafer. However, the features on the pattern are cut from the same wafer as used for the reference object.

5.3.1 General Observations

To see the results of the sidewall profile, a part was viewed using a JEOL SEM. From the images of the periodic squares shown in Figure 5.8, the sidewall appears to be nearly 90° vertical. Having a vertical sidewall is important for the comparison between CT and CSI data since the measurands are different. The CSI measures the top and bottom surfaces to determine spatial dimensions, while the CT data measures from the sidewalls. By having an undercut sidewall, the CT spatial measurements would be larger than the CSI.

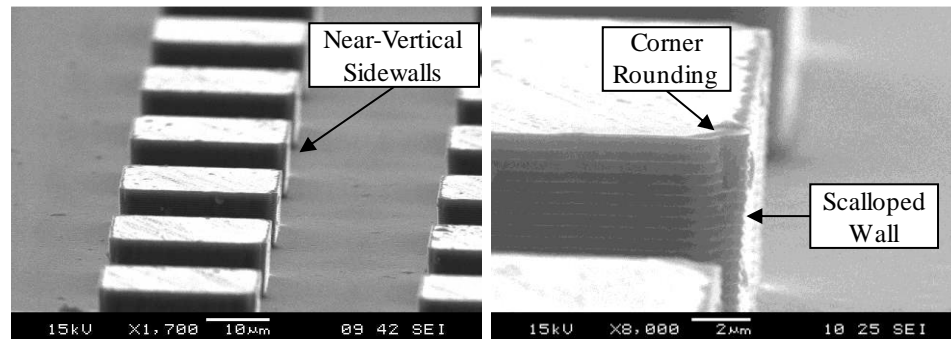


Figure 5.8. SEM images showing the scalloped sidewall profile of the periodic squares. Modified from [160].

Inspecting one of the squares in more detail, the scalloping of the sidewall caused by the switching between passivation and etching is observed. Refinement of the etching parameters would reduce the scalloping effect. However, since the voxel size of the CT data is $3.8\text{ }\mu\text{m}$, the scallops within a voxel are averaged. Another effect that can be seen is rounding of the feature's corner. This rounding is a byproduct of the corner causing interference during the exposure process. To reduce the corner rounding, the geometry of the features would need to be redesign on the photomask using optical proximity correction techniques to work with the mask aligner's UV wavelengths.

The process for direct bonding of silicon took a couple of years to yield successfully bonded assemblies. During this development a test part was constructed by dipping silicon layers into cyanoacrylate glue to act as a bonding medium. While using an adhesive is faster and less expensive, the quality of the reference object is reduced as observed in CT measurements. Figure 5.9 shows the interface between two silicon layers in which adhesive appears in the X-ray CT image as splotchy regions of varying contrast and partially filled etched cavities. A cross-section of the layers shows a visible gap between the layers, even though force was applied to the object during construction. As expected, the directly bonded reference object has a uniform layer interface and no visible separation between the bonded silicon layers.

The adhesive reference object was fabricated before the protective shell was designed. This object provides a baseline for the determination of the silicon's external surfaces. Surface determination is limited to differences in the attenuation of the X-ray by the materials. When two materials attenuate the X-ray signal at nearly the same rate, the contrast is low. Attenuation rates for silicon and glass are closer than to air (Figure A.8 in Appendix A), and by adding the glass end plates to the reference object, the surface reconstruction is complicated. Figure 5.10 shows the result of this increased difficulty in material determination. While the silicon with an air interface appears smooth, the silicon with the glass interface appears significantly rougher.

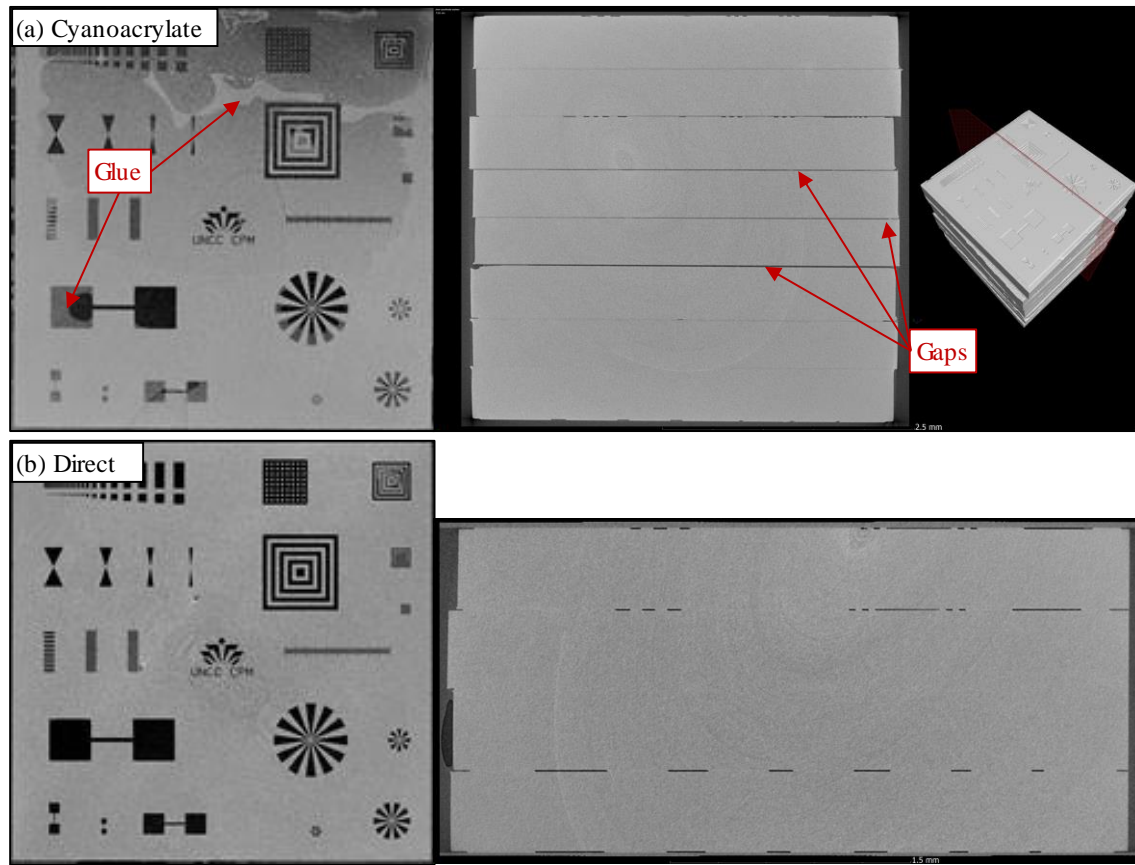


Figure 5.9. A comparison of the (a) cyanoacrylate and (b) direct bonding between layers of silicon. Modified from [160].

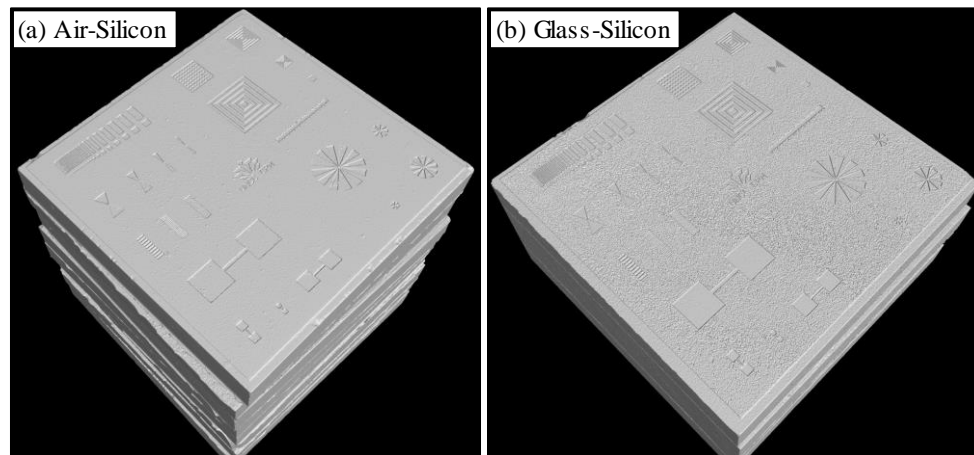


Figure 5.10. Silicon surface reconstruction (a) without a protective shell and (b) with the protective shell. Modified from [160].

Testing materials and clamp designs was part of the final assembly process. The main problem that plagued early clamp designs was not being able to fill the gap between the glass plates. Even though the designs contained a port for venting, air bubbles were created near the borders as shown in Figure 5.11. The addition of the second syringe to create a small vacuum helped with removing the trapped air. For the end plates, diced microscope glass slides were first tried. Although the microscope slides appeared clear during the assembly, after curing the UV epoxy in the oven, the glass became cloudy. The cause of this cloudiness was not investigated, however switching to a silica wafer solved the clouding problem.

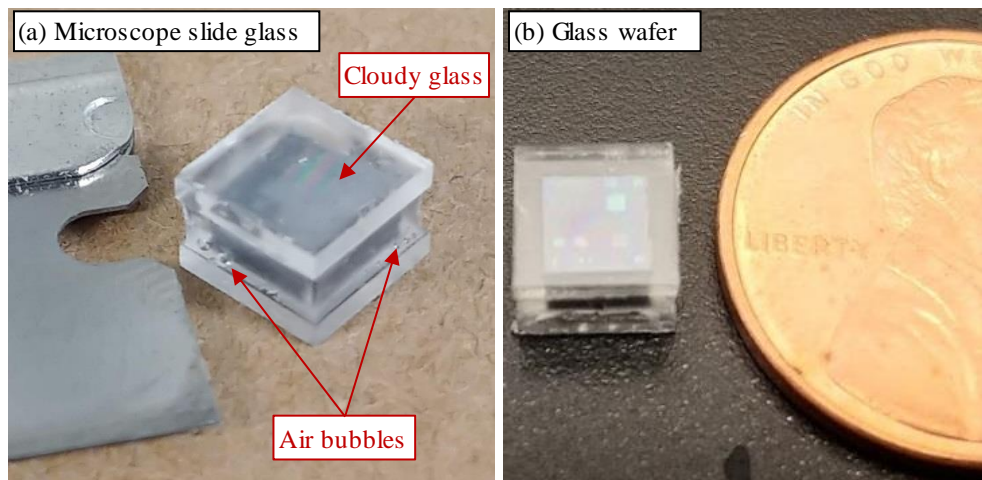


Figure 5.11. An (a) early reference object using microscope glass and (b) a final version with end plates cut from a silica wafer.

5.3.2 Extracted Feature Measurements

The goal of this section is to demonstrate measurements that can be performed on the reference object features. Neither the CT or CSI are provided with measurement uncertainties, and, because the features measured are different parts from the same wafer, this comparison is not a performance evaluation of the tool. Measurements provided are

linear dimensions between edges, curve fitting, and step heights. The largest asymmetric cubic figure is evaluated using CT data and CSI data using the 20 \times and 50 \times objectives. The second feature evaluated is the chain link using CT data and 20 \times objective CSI data.

5.3.2.1 Asymmetric Cubic

Contrast is an essential component for defining a feature's border. Data with a lower contrast appears to have less distinct, noisier, edges that will impact line and curve fitting. As evident in Figure 5.12, the CT data feature edges appear blurry compared to the CSI phase. CSI images are taken with the 20 \times objective. Although the CSI intensity data is not used in the measurements, it shows the data is acquired by the camera using two analog-to-digital converters to increase acquisition speed. Looking closer at border of the feature, there are four dark edges. After converting the CSI phase data to a binary image using the algorithm described in section 5.2, it is shown in Figure 5.13 that the edge of the binary image falls on an inner edge of the intensity data.

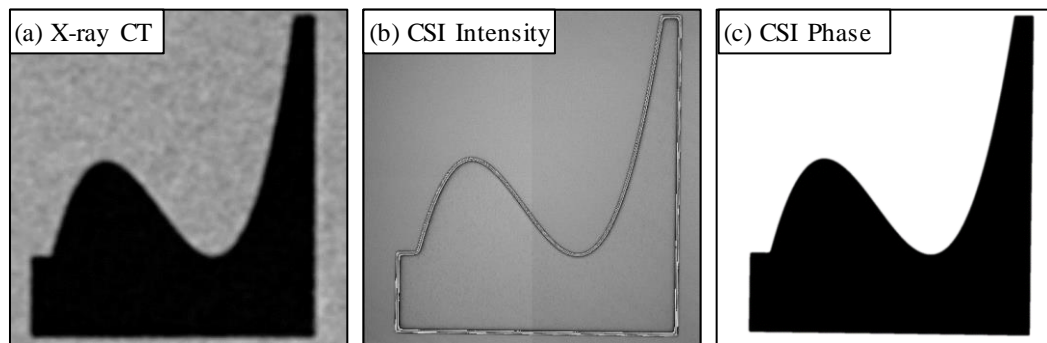


Figure 5.12. Images of the largest asymmetric feature that shows the contrast difference between the (a) reconstructed X-ray CT data, (b) CSI intensity data, and (c) CSI phase data. CSI data taken with 20 \times objective. Modified from [160].

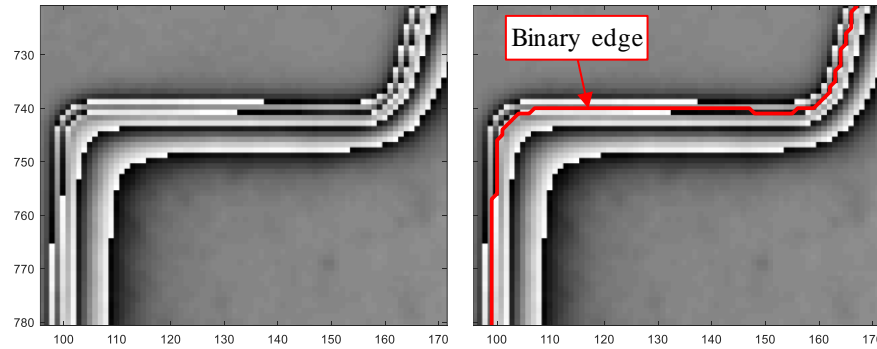


Figure 5.13. CSI intensity image showing four boundaries of the feature and the binary edge superimposed onto the image following one of the inner boundaries.

Measuring the step height with the CT data was handled within VGStudio. In VGStudio, with the material definition set to automatic to use ISO50 thresholding, planes were fit to the etched and polished surfaces using least squares fitting. A plane representing the etched surface was constructed from 1042 data points and one for the original polished surface had 1051 data points. The mid perpendicular step height was $9.76\ \mu\text{m}$ as shown in Figure 5.14. With a $3.8\ \mu\text{m}$ voxel size, the geometry has to be predicted by three voxels. To improve the CT measurement, a system with a smaller focal spot size must be used to reduce the blur and voxel size. For comparison, the CSI's measured step height using the MATLAB® algorithm was $8.96\ \mu\text{m}$.

Spatial measurements from the CSI and CT data are included in Figure 5.15. With the $20\times$ objective and $1\times$ tube lens, CSI measured linear dimensions are less than a micrometer from the nominal dimensions. For a traceable measurement of the spatial dimensions, a tool such as a metrology atomic force microscope is recommended for this measurement. Linear measurements from CT data measure larger yet, however the values are within a single voxel of both the nominal and CSI measurement. Coefficients of the cubic polynomial fit show the CSI is closer to the nominal than the CT data. Once again,

switching to a CT system with reduced blur and a smaller voxel size, the spatial measurements and curve fitting might improve agreement with the CSI and nominal values.

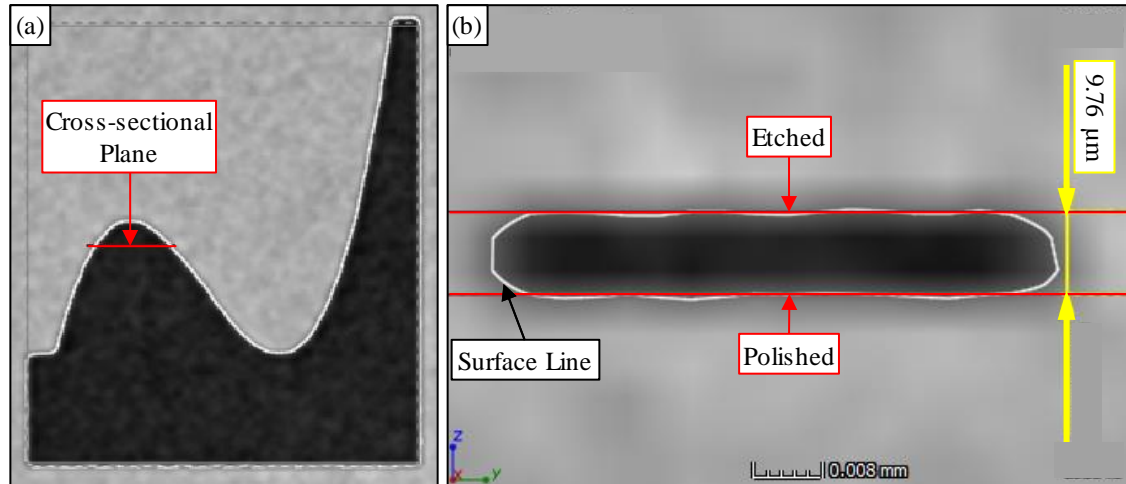


Figure 5.14. View of a (a) cross-sectional through the CT reconstructed volume to show the (b) cavity shape and step height. Modified from [160].

Besides measuring the CT data as an average of multiple slices, the individual slices were evaluated (Figure 5.16). When graphing the single slice measurements, as expected the curve of the sidewalls that was seen in Figure 5.14 is present. Slices closer to the top or bottom surface planes yield linear measurements closer to nominal, but the curved edge becomes more distorted.

The last set of measurements are the tangent-normal residuals of the measured data from the nominal line and line of best fit (see Figure 5.17). Coefficients for the best fit line are provided in Figure 5.15. Standard deviations of the CSI data are 0.14 μm from the nominal line and 0.12 μm from the fitted line. For the CT data, the standard deviations are 0.39 μm and 0.32 μm respectively. Both curve fitting methods show the CSI data is about 36.5% of the CT. Since the data is recorded as discrete pixels, similarities can be seen around positions -100 μm and 0 μm.

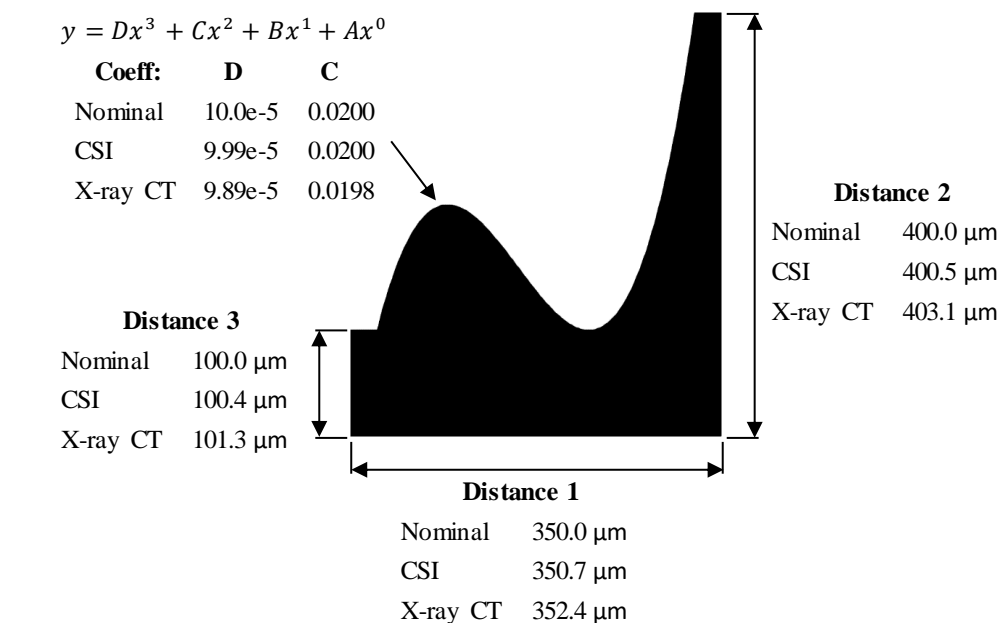


Figure 5.15. Spatial dimensions and cubic fit coefficients of the asymmetric figure. Modified from [160].

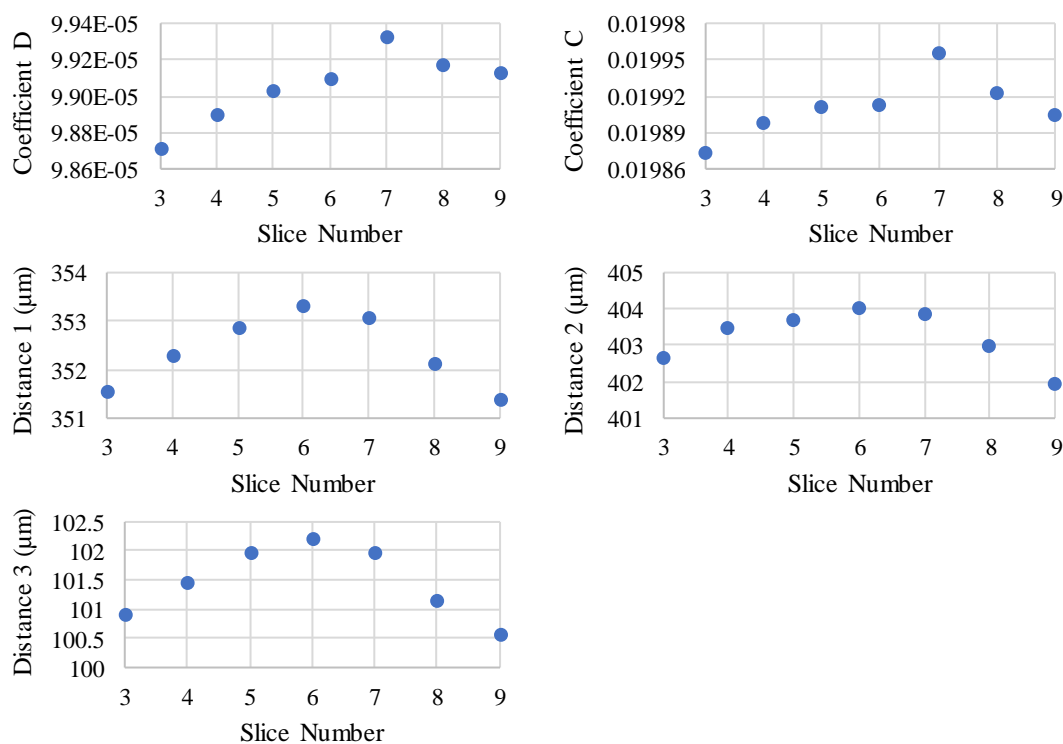


Figure 5.16. CT spatial and curve fitting measurements for the individual slices.

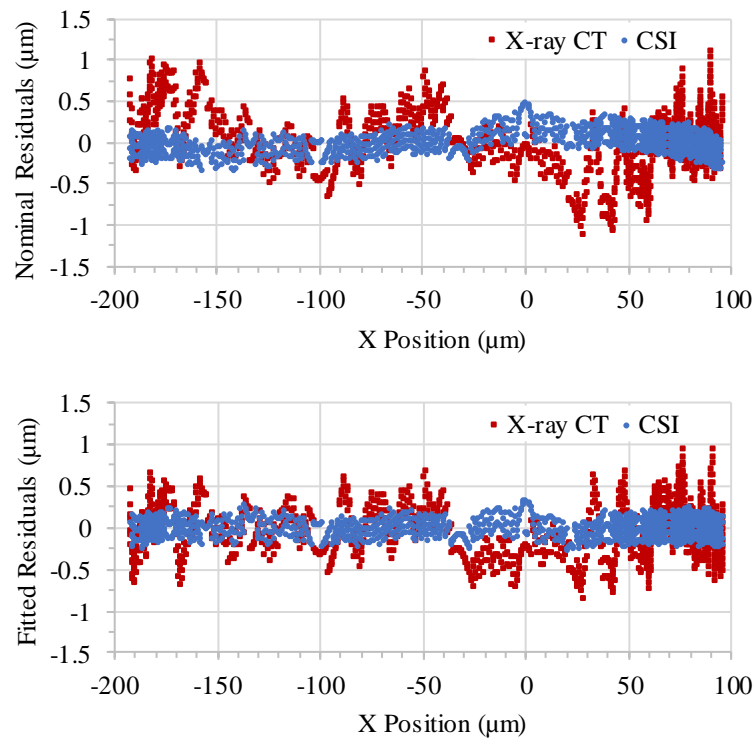


Figure 5.17. Residuals of the measured CSI and CT data from nominal and fitted polynomial lines [160].

The CSI offers stitching multiple scans into an image to provide an increased field of view. By utilizing the stitching function, it was possible to measure the same asymmetric cubic feature using higher magnification. With the 50 \times objective and 1 \times tube lens, the spatial sampling rate is 0.164 $\mu\text{m}\cdot\text{pixel}^{-1}$ with a 164 μm square field of view. To scan the full 350 \times 400 μm^2 asymmetric cubic feature, the stitching parameters were chosen to scan a 3 \times 3 matrix with 20% overlap. Stitching algorithms offered are ‘XY Adjust’, ‘Cartesian’, ‘Overlay’, and ‘Adaptive Adjust’. For testing the algorithms, ‘Overlap Merge’ was set to ‘Blend’, ‘Sample Mode’ was set to ‘None’, and ‘Global Correction’ was enable for ‘XY Adjust’.

Results of the linear dimensions and curve fit coefficients are shown in Figure 5.18. The algorithm that produced an image with linear dimensions closest to the nominal values was 'Adaptive Adjust'. Next closest results were the 'Cartesian' and 'Overlay' methods with results that were nearly identical. 'XY Adjust' had the largest deviation from nominal in the 'Distance 1' measurement at $0.7\text{ }\mu\text{m}$ and was also the only algorithm in 'Distance 3' to exceed the nominal even though the 'Distance 2' was identical to the 'Adaptive Adjust' measurement. As for the best fit curve coefficients, the 'Adaptive Adjust' algorithm once again had values closest to the nominal.

Tangent-normal residuals of the stitched curves were measured for their standard deviation (Table 5.2) and those residuals were plotted (Figure 5.19). Based on the curve fitting results, both the 'XY Adjust' and 'Adaptive Adjust' algorithms performed well. The smallest standard deviation of the residuals was measured as $0.065\text{ }\mu\text{m}$ using the 'Adaptive Adjust' stitched scans. Both the 'Cartesian' and 'Overlay' performed the worst with every measurement exceeding $0.210\text{ }\mu\text{m}$, over twice as large as the worst value measured by the 'Adaptive Adjust' and 'XY Adjust'. Taking into account the previously discussed linear measurements, neither the 'Cartesian' nor 'Overlay' algorithms are recommended at this time for measuring these types of features.

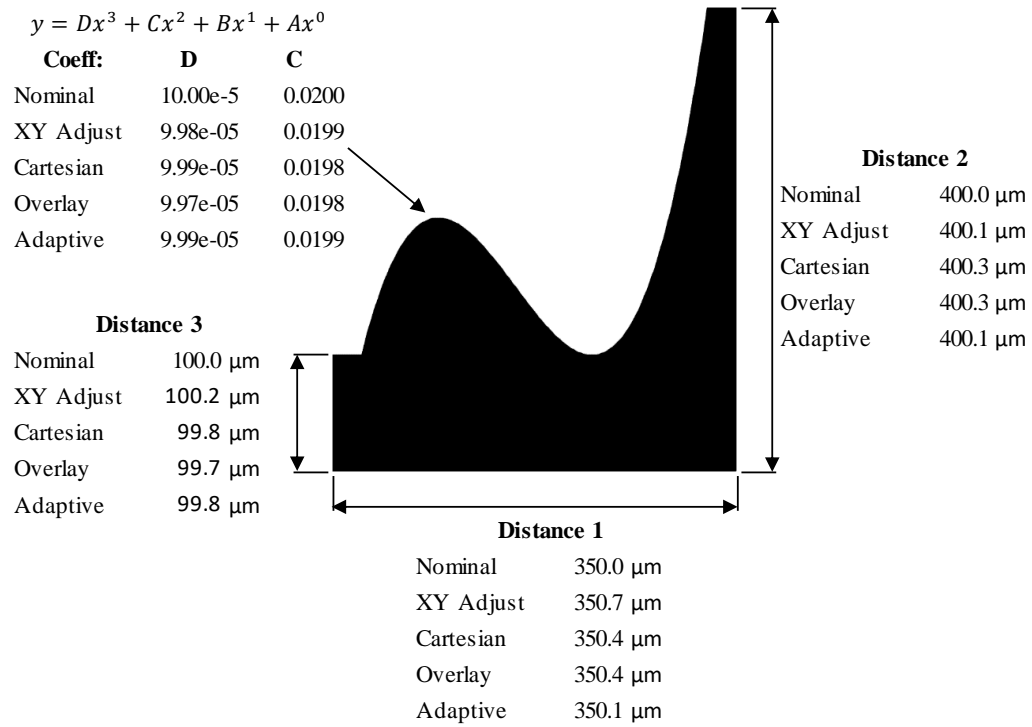


Figure 5.18. Dimensional measurement results for the four stitching algorithms.

Table 5.2. Standard deviations of the tangent-normal residuals as measured by the nominal coefficients and coefficients calculated by a best fit cubic curve.

Algorithm	Standard Deviation (1σ)	
	Nominal (μm)	Best Fit (μm)
XY Adjust	0.079	0.070
Cartesian	0.214	0.234
Overlay	0.246	0.242
Adaptive Adjust	0.090	0.065

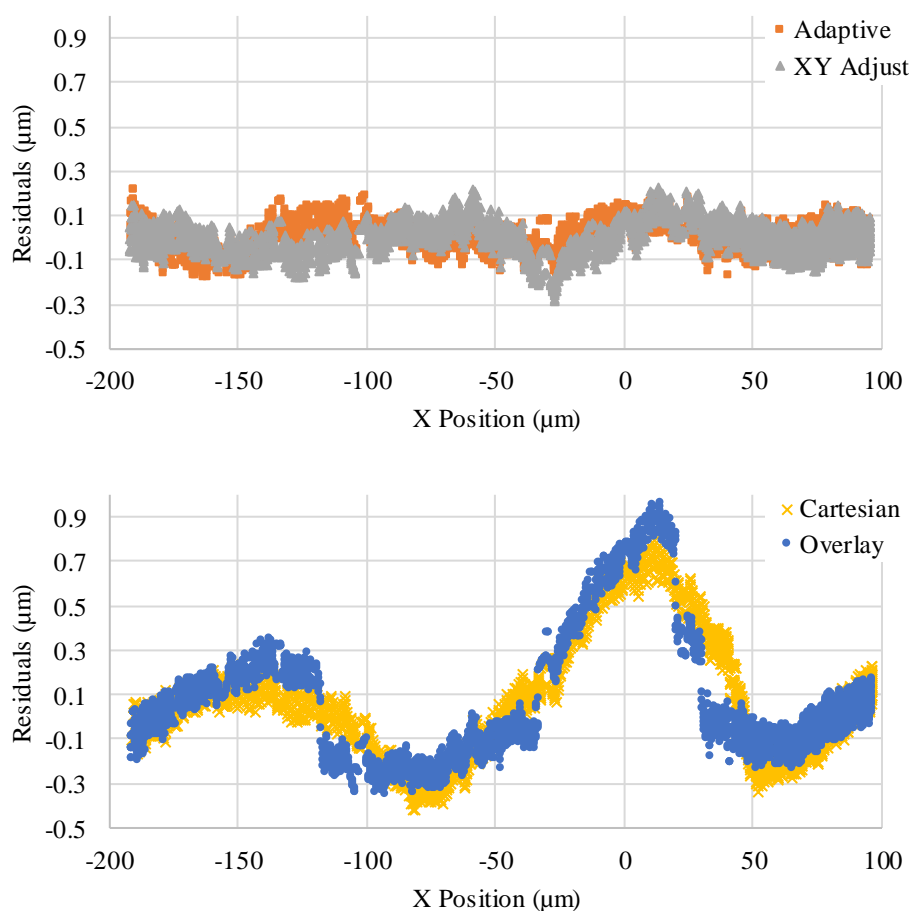


Figure 5.19. Comparison of the tangent-normal deviations as measured from the best fit cubic curve for the four stitching algorithms.

Intensity and phase images were evaluated to try to understand why the ‘Cartesian’ (see Figure 5.20) and ‘Overlay’ (see Figure 5.21) algorithms yield significantly different standard deviations of the residuals for the cubic curves. Intensity and phase images for the ‘XY Adjust’ are provided in Figure 5.22, and only the phase images shown in Figure 5.23 is generated for ‘Adaptive Adjust’ stitching. In the intensity data, near the local minima of the cubic curve, there is a section of the curve where multiple scans merge. Upon closer inspection of the section, it is clearly seen that the curved line in ‘Cartesian’ and ‘Overlap’ stitched images are not optimally aligned. In contrast, there are no apparent misalignments

using the ‘XY Adjust’. With the phase image, misalignment of the line in the merged section is not as easily discerned.

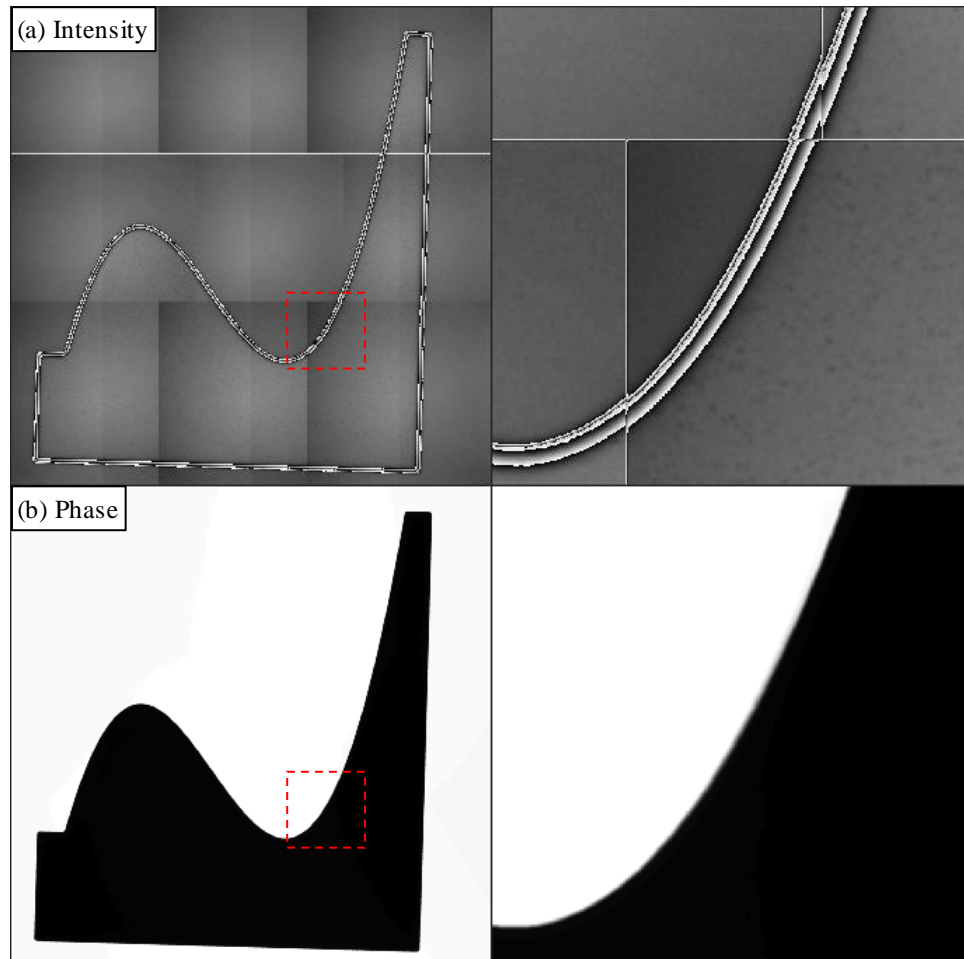


Figure 5.20. Intensity and phase images using the ‘Cartesian’ stitching algorithm.

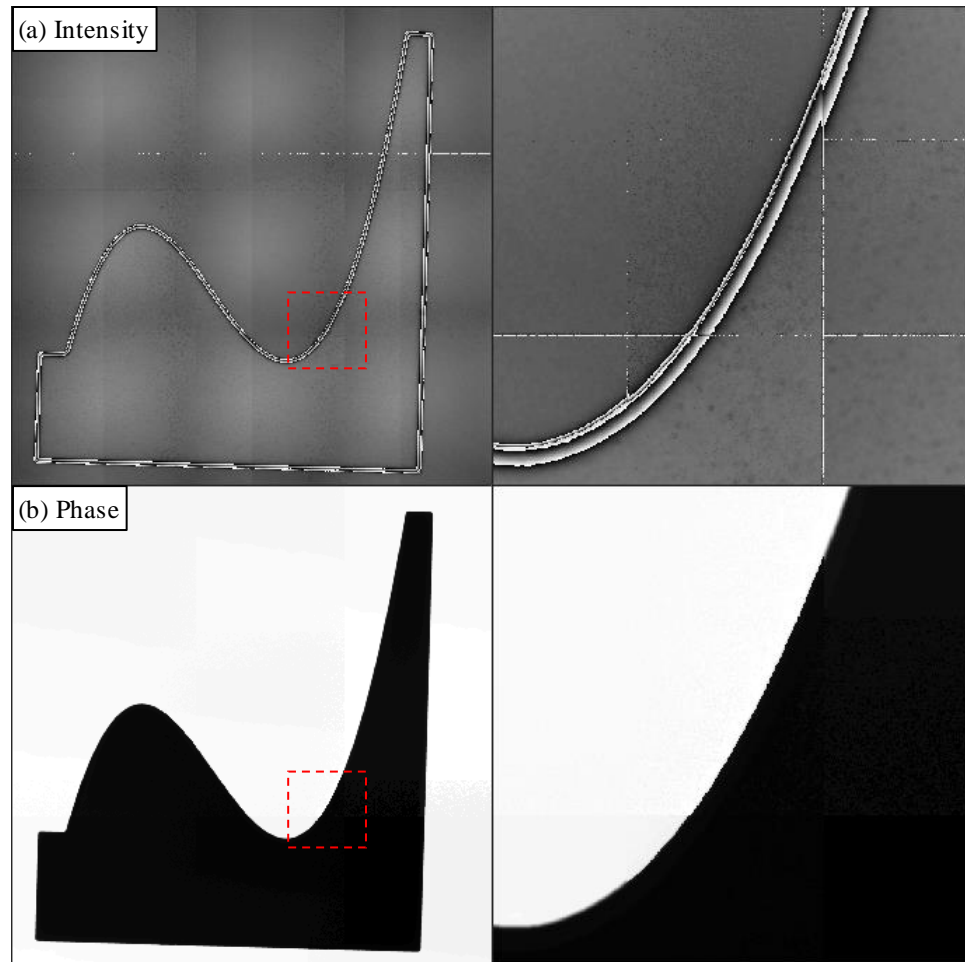


Figure 5.21. Intensity and phase images using the 'Overlay' stitching algorithm.

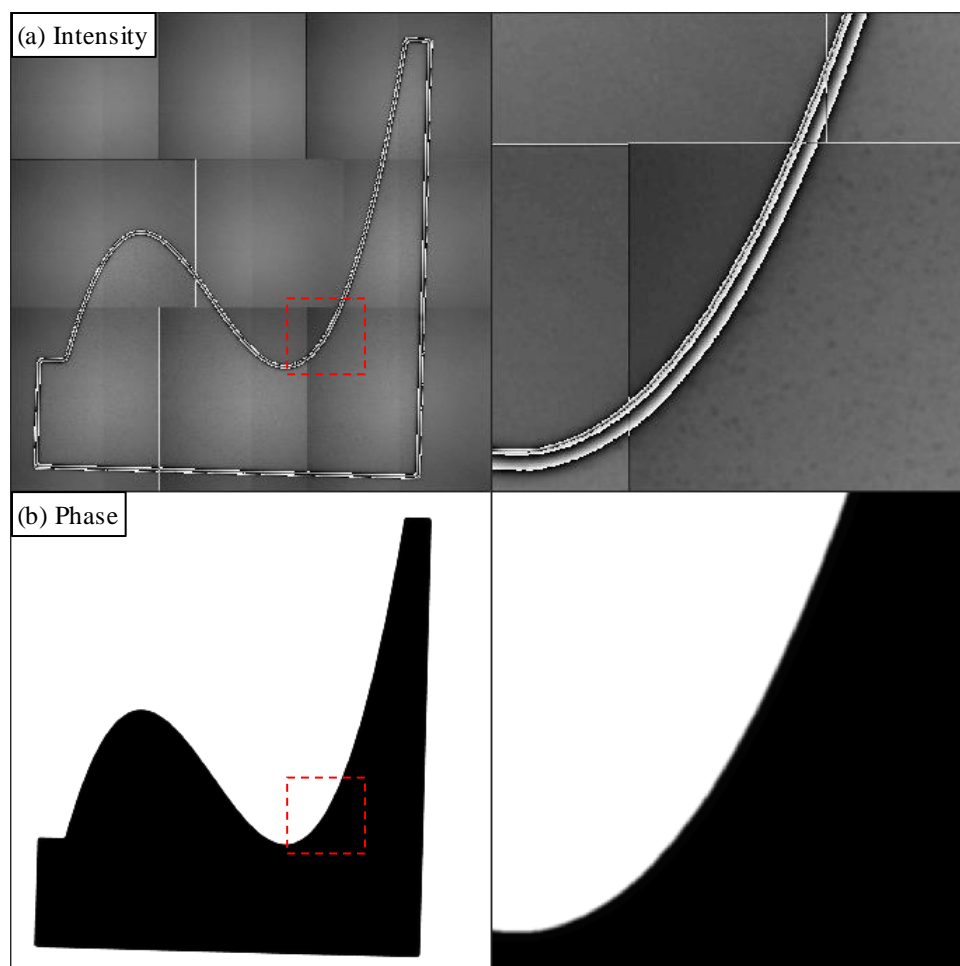


Figure 5.22. Intensity and phase images using the 'XY Adjust' stitching algorithm.

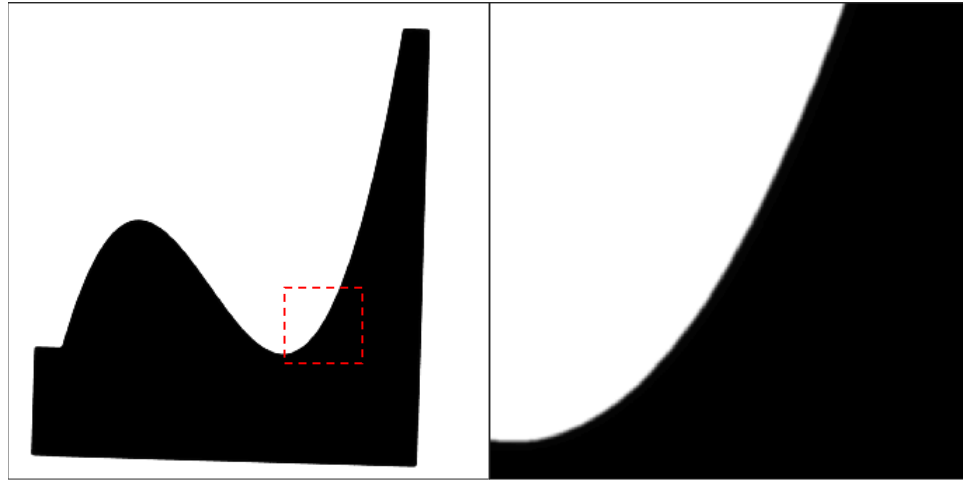


Figure 5.23. Phase images using the 'Adaptive Adjust stitching algorithm.

5.3.2.2 Chain Link

The last feature that will be evaluated is the chain link formed by four arcs of equal radius with two fiducials. The second smallest chain link feature constructed by 75 μm radius arcs was measured using the 20 \times objective by the CSI. For CT measurement, seven image slices were averaged together for measurement. After best fitting circles to the four arcs, the measurements reported are the fitted circle's radius, the standard deviation of the residuals of the arc data points from the fitted circle, and the distance between the centroids of the fitted circles. From the CSI data, the measured depth of etch is 8.71 μm .

Radii measurements and the arc names are shown in Figure 5.24. From the CSI data, each arc was within 0.3 μm of the 75 μm nominal with a 74.99 μm average radius. The CSI top and bottom arcs are measuring larger than nominal, while the cross and circle arcs are smaller. X-ray CT arc measurements were between 0.6 μm to 1.16 μm larger than nominal and a 75.78 μm average radius. Although the cross and circle arcs from the CT data were close, the top and bottom arcs were not. From the plots of the borders shown in

Figure 5.25, the CSI data appears smooth and the CT data has some form errors. Since the spatial sampling is calculated from a scale bar on the CT images, the scaling may be off by about 1% based on the average radius to the nominal. Scaled radius values are 75.32 μm for the top arc, 74.83 μm for the cross arc, 74.77 μm for the circle arc, and 74.88 μm for the bottom radius.

The next set of spatial measurements are the centroid-to-centroid distances. Maximum difference from the nominal distances was found to be 0.08 μm with the CSI and 1.98 μm with the CT; both occurring in the distance measurement between the two arcs containing fiducials. With the radii measurements, CT measurements were all larger than nominal. Scaling the CT measurements down by 1% reduced the 1.98 μm difference to 0.52 μm . The measured and scaled distance measurements can be found in Table 5.3.

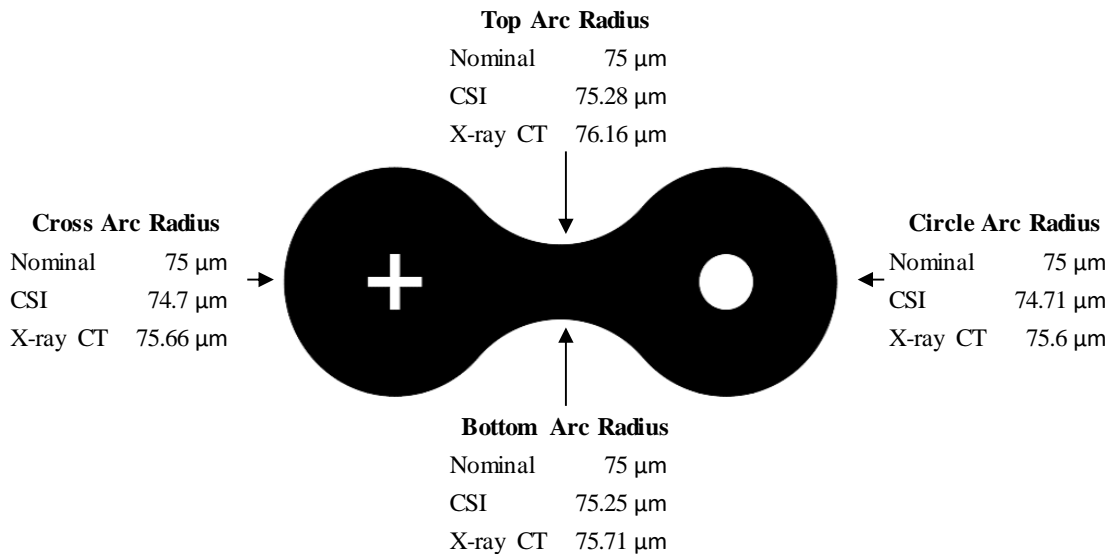


Figure 5.24. Radial measurements of the chain-link arcs.

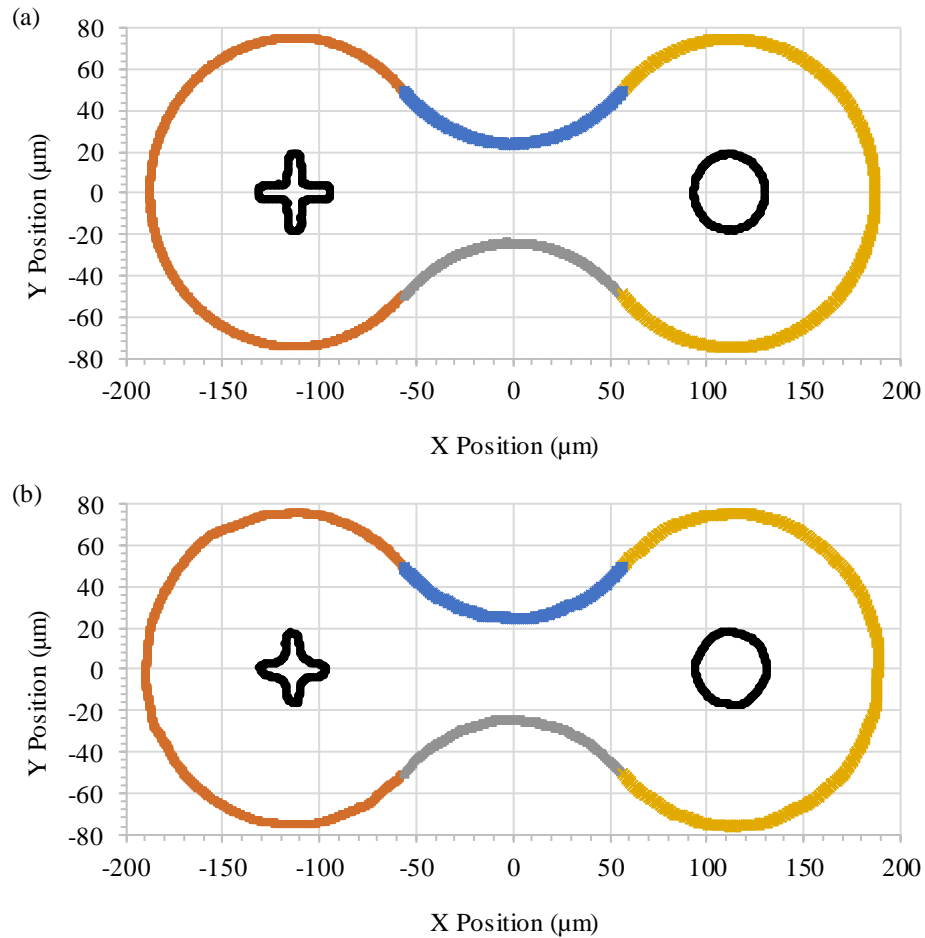


Figure 5.25. Chain link plots of the (a) CSI and (b) X-ray CT data.

Standard deviations are nearly identical to the asymmetric cubic values measured by the CSI and X-ray CT data with best fitted curves. Total standard deviations are $0.12 \mu\text{m}$ for the CSI and $0.34 \mu\text{m}$ for the CT. The standard deviations for individual arcs are given in Table 5.4 and the residuals are plotted in Figure 5.26. As seen during the asymmetric cubic measurements, the CT residuals have larger spikes than the CSI data. Unlike the radius and distance measurements where scaling the spatial sampling has a significant impact, scaling the residuals yields a negligible change in the standard deviation.

Table 5.3. Centroid-to-centroid distance measurements of the chain link.

Arcs	Nominal	CSI	CT	Scaled CT	
Cross-Circle	225	224.92	226.98	224.48	μm
Cross-Top	150	149.96	151.85	150.18	μm
Cross-Bottom	150	149.94	151.4	149.73	μm
Circle-Top	150	150.07	151.73	150.06	μm
Circle-Bottom	150	150	151.31	149.65	μm

Table 5.4. Standard deviations of the chain link's arcs.

Arcs	CSI	CT	
Cross	0.12	0.4	μm
Circle	0.13	0.33	μm
Top	0.12	0.23	μm
Bottom	0.11	0.24	μm
Combined	0.12	0.34	μm

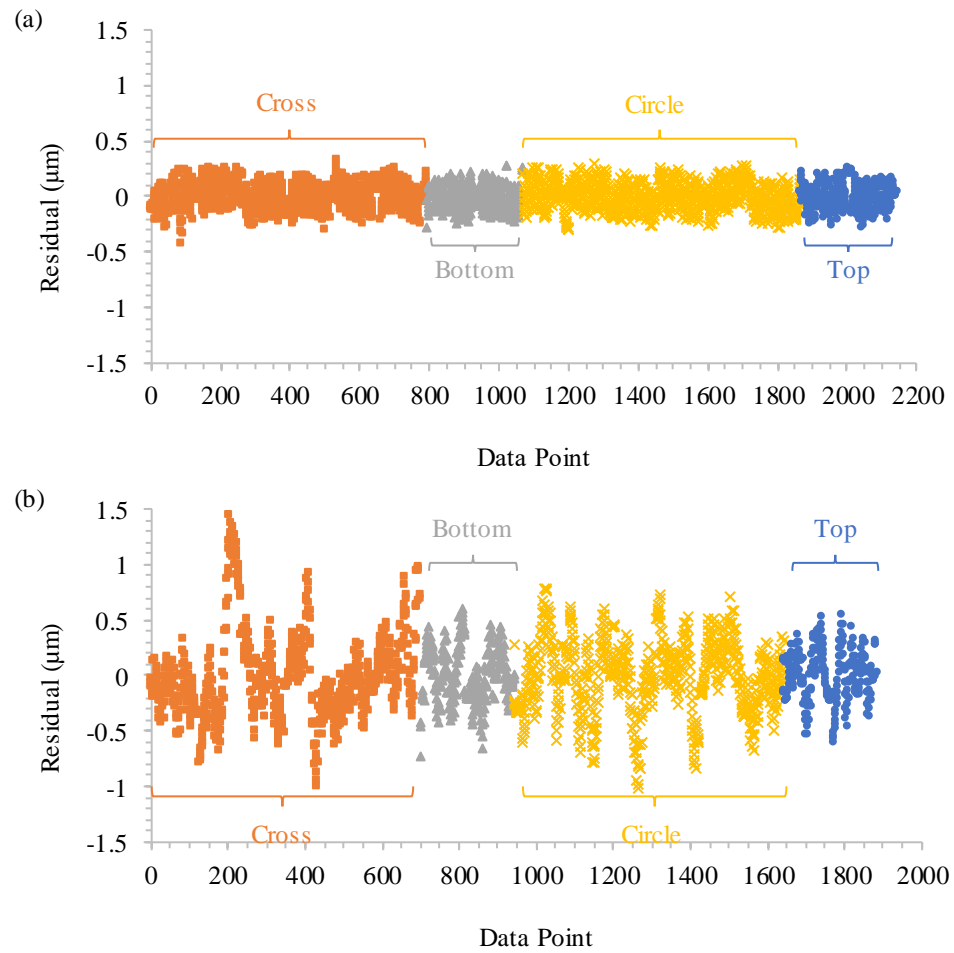


Figure 5.26. Residuals of the (a) CSI and (b) X-ray CT data from the fitted arcs.

CHAPTER 6: CONCLUSIONS

6.1 X-ray Facility

Developing an X-ray facility is presented in the first part of the dissertation. Conceived originally for X-ray interferometry studies, the facility was designed for adaptability as research needs grow. The first task of the project was to design and construct an enclosure satisfying safety regulations, provide space for internal components, and fit on a vibration isolation table. The enclosure was built from multiple steel frames that interlock together and lined with lead sheets. For visibility into the enclosure, lead infused acrylic windows were integrated into the sides and front doors. Interlock switches disable the X-ray system from exposing users to radiation when doors are opened while the system is emitting radiation.

Inside the enclosure, the system is separated into source, specimen, and detector subsection. The source subsection is the X-ray generator affixed to a stepper motor controlled rotary table to adjust the pitch of the X-ray beam. Making up the specimen subsection are multiple linear and a single rotary stages where the silicon interferometer or other targets for CT or reflectivity studies are placed. The final subsection has two linear stages that align the X-ray detectors to the X-ray beam. Outside the enclosure are a host computer and other controllers to operate and monitor the subsystems. To reduce the noise produced by the water chiller that cools the X-ray generator, an acoustic chamber was built around the chiller decreased the noise level by 8 dB. Finally, a crystal aligner to aid with manufacturing new interferometers is being developed.

Performance of the facility was evaluated using thermal sensors, radiation detectors, and by performing diffraction and reflectivity experiments. Temperature measurements of the lab exhibited an oscillation caused by the air conditioning system's control system which were filtered by the X-ray enclosure acting as a low pass filter. Thermal measurements inside the enclosure showed an elevated temperature, about 0.5 °C, relative to the lab. The primary source of the temperature increase within the enclosure is when the X-ray generator is on and the shutter is open. Over a period of 80 min, the area near the X-ray generator increased 1.6 °C, while the areas near the specimen and camera increased 0.8 °C.

Radiation measurements consists of a commercial ionization meter, Geiger-Müller detectors, and X-ray cameras. The ion chamber detector with its 3.5 s time constant was deemed too slow for use as a detector inside the chamber. A Geiger-Müller circuit with a maximum frequency just over 30 kHz was built and optimized for an organic tube. After the organic tube deteriorated from X-ray exposure, it was replaced with an inorganic tube. In practice, the maximum count rate was under 23,000 counts·s⁻¹ of the sensor. Imaging the beam started with a commercial X-ray camera where image correction techniques were tested to reduce image noise. After the commercial camera failed, a custom X-ray camera was built from off-the-shelf items. Camera readout noise and gain factor were measured as 11.3 e⁻ and 0.25 e⁻·ADU⁻¹ respectively. One improvement made to the camera design was replacing the leaded glass that protected the CCD sensor with a turning mirror, increasing light to the sensor by 12.5%. Another improvement was depositing an aluminum mirror coating to the external face of the scintillator, nearly doubling the light seen by the sensor.

Finally, tests were performed to evaluate the X-ray source for divergence and its ability to be used in diffraction and reflectivity studies. The divergence of the beam was measured by changing the camera distance and measuring the FWHM beam intensity. Divergence of the beam was found to be between 0.04° and 0.05° . Using a bulk piece of silicon saved when the interferometer was built, the X-ray facility was capable of diffracting off (111) and (220) planes. Reflectivity was also successfully accomplished using a polished tungsten carbide gage block.

6.2 CT Reference Object

Developing the reference object started as a side project while developing the facility. Feedback provided by interested individuals however moved it from a simple 2D object with linear features to a bonded 3D object with linear and curved features. Since it started as a 2D etched wafer, early research focused on characterizing the silicon etching tool. After the suggestion to move towards a 3D object, focus switched to developing a bonding process that works with the resources available. The manufacturing process is broken into an etching phase, a bonding phase, and final assembly.

Building the reference object uses prime grade silicon wafers with a thermal oxide grown to protect the silicon surface. After cleaning the wafers in a 'Piranha' solution of sulfuric acid and hydrogen peroxide, a $1.3\text{ }\mu\text{m}$ thick coating of photoresist is spin coated onto the wafer and soft-baked. Using a mask aligner with a UV source, the photoresist is exposed through a photomask to transfer the patterns. Post-exposure baking reduced the standing waves and the wafers are developed in a developer solution to wash away UV expose photoresist. Wafers are dry etched using a plasma source using deep reactive-ion

etching to form near-vertical walls up to 20 μm etch depth, and then diced into $17.1 \times 17.1 \mu\text{m}^2$ dies.

Direct bonding starts with selecting four dies and stripping them of the thermal oxide with hydrofluoric acid, rinsing with deionized water, and ‘Piranha’ cleaning. In a plasma etching system, the four dies are exposed to an oxygen plasma on both sides to clean the surfaces and alter the surfaces to improve low temperature direct bonding. After rinsing the dies in deionized water, they are dried with nitrogen, stacked, and clamped together. The clamped parts are annealed in an oven to 400 °C and then unclamped for a second anneal to 1100 °C. Direct bonded dies are then diced into sixteen reference objects with dimensions $4.2 \times 4.2 \times 2 \text{ mm}^3$.

In the final assembly process, a protective shell of epoxy and glass is constructed around the reference object. One reference object is stripped again of its oxide surface and then cleaned with two glass end plates. Using a custom-built clamp, the reference object is sandwiched between the glass end plates and epoxy is injected into the gap between the end plates and surrounding the reference object. After two hours in a UV oven, the epoxy is cured, and the reference object is complete. Dimensions of the fully assembled reference object measure $6 \times 6 \times 4 \text{ mm}^3$.

Reference objects were scanned by a Nikon industrial X-ray CT system. The CT scans showed that for an adhesive bonded reference object, the adhesive was visible in the interfaces between layers and that gaps between layers were present. For the external surfaces, the surface determination procedure generated a smoother silicon surface for the silicon-air interface than for the silicon-glass interface formed by the protective shell.

Features were extracted from the scans and compared to features scanned by a coherence scanning interferometer using the same MATLAB® algorithm. Linear dimensional measurements of the asymmetric cubic feature were within 1 μm from the nominal using the CSI and up to 3.1 μm using the CT. Two fitting algorithms were used, one that calculated the polynomial coefficients and the other using the nominal coefficients. Calculated coefficients of a best fit polynomial for the curve in the CSI data were closer to the nominal coefficients than the data from the CT. Standard deviation of the residuals from the nominal curve were 0.14 μm and 0.39 μm for the CSI and CT respectively. From the best fit curves, the values dropped to 0.12 μm and 0.32 μm respectively. When the 50 \times object was used for the CSI measurements, four stitching algorithms were tested. ‘Adaptive Adjust’ yielded dimensions within 0.2 μm to the three nominal dimensions and had the lowest standard deviation of the cubic curve’s residuals at 0.065 μm using the best fit line. Stitch misalignment in the ‘Cartesian’ and ‘Overlay’ images were spotted in the intensity images. This misalignment is the cause of their standard deviation of the cubic curve’s residuals to exceed 0.2 μm . The second feature measured was the chain link composed from four arcs. Similar results were seen between the CSI and CT measurements. CT measurements of the arc radiuses were all larger than nominal, up to 1.16 μm , while the CSI were within 0.3 μm of either side of nominal. Likely culprits to the CT dimensional measurements being larger are an inaccurate spatial sampling size, the large voxel sizes in the CT scan, and blur from the CT system. Standard deviations were also similar, at 0.12 μm for the CSI and 0.34 μm for the CT.

Measurements made by the CT and CSI are not indicators of the system’s performance, but have been provided to demonstrate the measurement algorithm and show

ways in which the reference object can be measured. The primary take-away from the reference object project is that a manufacturing process and measurement algorithm have been successfully demonstrated.

CHAPTER 7: FUTURE WORK

7.1 X-ray Facility

Improving the facility has the potential to open up more research opportunities. Areas of the facility that can be improved are the motion stage configurations in the subsystems, the X-ray camera, and programming. Other future work involves the crystal aligner, an inclinometer (see Appendix D), and a monochromator.

In the facility's current configuration, the pitch of the X-ray generator is adjusted when using the interferometer to reach the Bragg angle. In turn, this requires the Z-axis stages for the interferometer and detectors also move to maintain alignment. The issue with this setup, specifically with respect to the camera, is that the angle of the camera is not adjusted to maintain perpendicularity to the beam and the distance from source to detector changes. Introducing a goniometer as part of the detector subsystem would improve the facility for interferometry and reflectivity studies by allowing the user to couple the specimen and camera rotational axes.

Future improvements to the X-ray camera are changing the lens and digital camera. While zoomable CCTV lens is convenient for alignment and viewing both the transmitted and reflected beams, it is not suitable for high resolution imaging. In place of the CCTV lens, the camera system should employ microscope objectives with set magnifications that are matched to scintillators. Drawbacks to Atik 428EX digital camera are the sensor read time, data transfer rate to the computer, and no adjustable gain. Currently, the Atik camera takes over a second to read and transfer the data to the computer because it has a single readout channel and uses USB 2.0 for the communication. Switching to a camera with multiple

readout channels and USB 3.0 would improve the sensor read and transfer speed. Having an adjustable gain would reduce the capture time during alignment and setup testing, when image quality is not as important.

Programming in LabView is a messy process with data lines crisscrossing across the screen to reach functions. Communication with the motion controllers should be handled by one program and the camera handled by a separate program, similar to how the Bede microcontroller program runs independently. This would allow the user to manually make on the fly changes without needing to restart the main program. Moving to a producer-consumer design with the stages and camera controls decoupled from the main program would increase functionality, reduce the program writing time, and improve troubleshooting.

For the crystal aligner, adding a self-centering vise would improve stability while the cylinder is rotating and improve workpiece holding. With the crystal aligner finished, using the facility to manufacture an inclinometer can commence.

An inclinometer for the rotary table holding the X-ray generated was started to assist with aligning the X-ray interferometer. Finishing the inclinometer involves developing a calibration process, applying the calibration, and adding a relative measurement function.

Manufacturing a monochromator for filtering out unwanted wavelengths is also on the list of future work. During the X-ray interferometer manufacturing, a second piece of silicon was also aligned and cut to produce one flat edge. The spare block is almost the ideal candidate to turn into a double-crystal monochromator. Although these suggested improvements would help towards the goal of increasing performance and research

opportunities, many challenges still lie ahead. Problems have plagued the facility throughout the project, but the results have shown the facility allows for a unique amount of user control not offered by commercial systems.

7.2 CT Reference Object

Areas where reference object improvements need to be made are the smoothing the etch wall, bonding process, and measurement process.

Creating a smoother etched sidewall can be accomplished a few ways. The most common method is to increase the number of etch-passivation cycles and reducing the etching rate. A continuous passivation method combines the passivating and etching gases [165]. And one other method is a three-step cycling process that adds a passivating layer, removes the passivating layer at the bottom of the trench with the etchant, and then etches the silicon while also adding passivation [166].

Cleaning and maintaining the cleanliness is the most difficult part of the project. Moving to an ISO class 1 or 2 clean room would be helpful in reducing the environmental contamination. Use of tweezers on the surfaces should be minimized by designing need fixtures than can transfer, align, and place the parts in contact. Oxygen plasma was found to help improve the bonding success, however the way it was done is by placing the parts on a carrier wafer which could add contamination and requires entering the ISO class 6 bays to use the tool. As of April 2019, a PIE Scientific Tergeo Plus tabletop plasma cleaner has been installed in the ISO class 5 area. The silicon dies can be loaded into the plasma cleaner on quartz boats that contact the edges instead of the flat surfaces that will be bonded. Although the use of the standard clean 1 and 2 solutions did not show any

noticeable benefit when tested, it would be worth taking some time to reevaluate the wet chemical cleaning process.

Finally, the process is designed so that parts can be measured before bonding. While CSI is useful for measuring the step height of a feature, the spatial resolution is diffraction-limited. Critical-dimension atomic force microscopes with quantified uncertainties could be used as the reference instrument for spatial dimensioning [167–171]. If access to a metrology AFM can be obtained, a path towards traceability would be to measure one pattern on four separate dies that when combined form a primary reference object. This traceable reference object would be scanned by a high-resolution X-ray CT system with sub-micrometer voxel sizes to then qualify secondary reference objects. This process would satisfy both the traceability issue and that the test uncertainty ratio of the reference instrument is ten times better than the ones being tested. However, this path towards traceability should not be started until the failure chance of bonding is minimized, as the metrology AFM will take a large amount of time to perform measurements.

REFERENCES

- [1] Sansare K, Khanna V, Karjodkar F. Early victims of X-rays: a tribute and current perception. *Dentomaxillofac Radiol* 2011;40:123–5. doi:10.1259/dmfr/73488299.
- [2] Tzvetkov G, Späth A, Fink RH. Soft X-ray induced damage in PVA-based membranes in water environment monitored by X-ray absorption spectroscopy. *Radiat Phys Chem* 2014;103:84–8. doi:10.1016/J.RADPHYSICHEM.2014.03.038.
- [3] Wang J, Morin C, Li L, Scholl A, Doran A. Radiation damage in soft X-ray microscopy. *J Electron Spectros Relat Phenomena* 2009;170:25–36. doi:10.1016/J.ELSPEM.2008.01.002.
- [4] Coffey T, Urquhart S., Ade H. Characterization of the effects of soft X-ray irradiation on polymers. *J Electron Spectros Relat Phenomena* 2002;122:65–78. doi:10.1016/S0368-2048(01)00342-5.
- [5] Laurinas VC, Kasymov SS, Yurov VM, Eremin EN, Vedyashkin M V. Effect of ultraviolet and x-ray radiation on optical properties of epoxy polymers dyed with organic phosphors. *IOP Conf Ser Mater Sci Eng* 2017;168:012023. doi:10.1088/1757-899X/168/1/012023.
- [6] Mori K, Nishioka Y, Takeda A, Takebayashi N, Takenaka R, Sakakura S, et al. Total ionizing dose effects on the SOI pixel sensor for X-ray astronomical use. *Nucl Instruments Methods Phys Res Sect A Accel Spectrometers, Detect Assoc Equip* 2018. doi:10.1016/J.NIMA.2018.07.057.
- [7] Srour JR, Marshall CJ, Marshall PW. Review of displacement damage effects in silicon devices. *IEEE Trans Nucl Sci* 2003;50:653–70. doi:10.1109/TNS.2003.813197.
- [8] Monmeyran C, Patel NS, Bautz MW, Grant CE, Prigozhin GY, Agarwal A, et al.

Annealing bounds to prevent further Charge Transfer Inefficiency increase of the Chandra X-ray CCDs. Nucl Instruments Methods Phys Res Sect B Beam Interact with Mater Atoms 2016;389–390:23–7. doi:10.1016/J.NIMB.2016.11.020.

- [9] Röntgen W. Ueber eine neue Art von Strahlen. Vorläufige Mitteilung. Aus Den Sitzungsberichten Der Würzburg Phys 1895:137–47.
- [10] Bragg WH, Bragg WL. The Reflection of X-rays by Crystals. Proc R Soc A Math Phys Eng Sci 1913;88:428–38. doi:10.1098/rspa.1913.0040.
- [11] Coolidge WD. Vacuum Tube. US Patent 1203495, 1913.
- [12] Duane, William; Hunt FL. On X-Ray Wave-Lengths. Phys Rev 1915;6:166–72. doi:10.1103/PhysRev.6.166.
- [13] Coolidge WD. X-ray apparatus. US Patent 1215116, 1917.
- [14] Hessenbruch A. A brief history of x-rays. Endeavour 2002;26:137–41. doi:10.1016/S0160-9327(02)01465-5.
- [15] WATSON JD, CRICK FHC. Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. Nature 1953;171:737–8. doi:10.1038/171737a0.
- [16] Oldendorf WH. Isolated Flying Spot Detection of Radiodensity Dis-Continuities-Displaying the Internal Structural Pattern of a Complex Object. Ire Trans Biomed Electron 1961;8:68–72. doi:10.1109/TBMEL.1961.4322854.
- [17] Oldendorf WH. Radiant energy apparatus for investigating selected areas of the interior of objects obscured by dense material 1963.
- [18] Bonse U, Hart M. An X-ray Interferometer. Appl Phys Lett 1965;6:155–6. doi:10.1063/1.1754212.
- [19] Bonse U, Hart M. An X-ray Interferometer with Long Separated Interfering Beam

Paths. Appl Phys Lett 1965;7:99–100. doi:10.1063/1.1754330.

- [20] Bonse U, Hart M. An X-ray interferometer with Bragg case beam splitting and beam recombination. Zeitschrift Für Phys 1966;194:1–17. doi:10.1007/BF01326517.
- [21] Bonse U, Hart M. Combined Laue-Bragg-case X-ray interferometers. Acta Crystallogr Sect A 1968;24:240–5. doi:10.1107/S0567739468000331.
- [22] Becker P, Bonse U. The skew-symmetric two-crystal X-ray interferometer. J Appl Crystallogr 1974;7:593–8. doi:10.1107/S0021889874010491.
- [23] Yoneyama A, Takeda T, Tsuchiya Y, Wu J, Thet-Thet-Lwin, Koizumi A, et al. A phase-contrast X-ray imaging system—with a 60×30 mm field of view—based on a skew-symmetric two-crystal X-ray interferometer. Nucl Instruments Methods Phys Res Sect A Accel Spectrometers, Detect Assoc Equip 2004;523:217–22. doi:10.1016/J.NIMA.2003.12.008.
- [24] Momose A, Takeda T, Yoneyama A, Koyama I, Itai Y. Wide-area phase-contrast X-ray imaging using large X-ray interferometers. Nucl Instruments Methods Phys Res Sect A Accel Spectrometers, Detect Assoc Equip 2001;467–468:917–20. doi:10.1016/S0168-9002(01)00523-X.
- [25] Momose A, IUCr. Phase-contrast X-ray imaging based on interferometry. J Synchrotron Radiat 2002;9:136–42. doi:10.1107/S0909049502003771.
- [26] Hirano K, Momose A. Development of an X-Ray Interferometer for High-Resolution Phase-Contrast X-Ray Imaging. Jpn J Appl Phys 1999;38:L1556–8. doi:10.1143/JJAP.38.L1556.
- [27] Momose A, Koyama I, Hirano K. Improvement of spatial resolution in phase-contrast x-ray computed tomography. In: Bonse U, editor. vol. 4503, International Society for Optics and Photonics; 2002, p. 71. doi:10.1117/12.452871.

- [28] Yoneyama A, Yamada S, Takeda T. Fine Biomedical Imaging Using X-Ray Phase-Sensitive Technique. In: Gargiulo GD, McEwan A, editors. *Adv. Biomed. Eng.*, Rijeka: IntechOpen; 2011. doi:10.5772/20456.
- [29] Momose A, Takeda T, Itai Y, Hirano K. Phase-contrast X-ray Computed Tomography for Observing Biological Soft Tissues. *Nat Med* 1996;2:473–5. doi:10.1038/nm0496-473.
- [30] Momose A. Phase-sensitive Imaging and Phase Tomography Using X-ray Interferometers. *Opt Express* 2003;11:2303–14.
- [31] Momose A. Demonstration of Phase-contrast X-ray Computed Tomography using an X-ray Interferometer. *Nucl Instruments Methods Phys Res Sect A Accel Spectrometers, Detect Assoc Equip* 1995;352:622–8. doi:10.1016/0168-9002(95)90017-9.
- [32] Wu J, Takeda T, Thet Lwin T, Momose A, Sunaguchi N, Fukami T, et al. Imaging renal structures by X-ray phase-contrast microtomography. *Kidney Int* 2009;75:945–51. doi:10.1038/KI.2009.42.
- [33] David C, Nöhammer B, Solak HH, Ziegler E. Differential x-ray phase contrast imaging using a shearing interferometer. *Appl Phys Lett* 2002;81:3287–9. doi:http://dx.doi.org/10.1063/1.1516611.
- [34] Pauwels B, Bruyndonckx P, Liu X, Tapfer A, Velroyen A, Yaroshenko A, et al. First small-animal in-vivo phase-contrast micro-CT scanner. In: Stock SR, editor. vol. 8506, *International Society for Optics and Photonics*; 2012, p. 85060J. doi:10.1117/12.929567.
- [35] Han H, Sharma Y, Zan G, Wu Z, Wang S, Wu Y, et al. Preliminary research on body composition measurement using X-ray phase contrast imaging. *Phys Medica* 2018;52:1–8. doi:10.1016/J.EJMP.2018.06.009.

- [36] Han H, Hu R, Wali F, Wu Z, Gao K, Wang S, et al. Phase-contrast imaging for body composition measurement. *Phys Medica* 2017;43:25–33. doi:10.1016/J.EJMP.2017.10.006.
- [37] Bech M, Bunk O, David C, Ruth R, Rifkin J, Loewen R, et al. Hard X-ray phase-contrast imaging with the Compact Light Source based on inverse Compton X-rays. *J Synchrotron Radiat* 2009;16:43–7. doi:10.1107/S090904950803464X.
- [38] Weitkamp T, Diaz A, David C, Pfeiffer F, Stampanoni M, Cloetens P, et al. X-ray phase imaging with a grating interferometer 2005;13. doi:10.1364/OPEX.13.006296.
- [39] Hart M. An Ångström Ruler. *J Phys D Appl Phys* 1968;1:1405.
- [40] Deslattes RD. Optical and X-Ray Interferometry of a Silicon Lattice Spacing. *Appl Phys Lett* 1969;15:386–8. doi:10.1063/1.1652870.
- [41] Deslattes RD, Henins A. X-Ray to Visible Wavelength Ratios. *Phys Rev Lett* 1973;31:972–5. doi:10.1103/PhysRevLett.31.972.
- [42] Basile G, Becker P, Bergamin A, Cavagnero G, Franks A, Jackson K, et al. COXI - A Combined Optical and X-ray Interferometer for high precision dimensional metrology. *Int Prog Precis Eng* 1993:248–59. doi:10.1016/B978-0-7506-9484-1.50030-0.
- [43] Basile G, Becker P, Bergamin A, Cavagnero G, Franks A, Jackson K, et al. Combined optical and X-ray interferometry for high-precision dimensional metrology. *Proc R Soc London Ser A Math Phys Eng Sci* 2000;456:701–29. doi:10.1098/rspa.2000.0536.
- [44] Yacoot A, Koenders L. From nanometre to millimetre: a feasibility study of the combination of scanning probe microscopy and combined optical and x-ray interferometry. *Meas Sci Technol* 2003;14:N59–63. doi:10.1088/0957-

0233/14/9/402.

- [45] Çelik M, Hamid R, Kuertgens U, Yacoot A. Picometre displacement measurements using a differential Fabry–Perot optical interferometer and an x-ray interferometer. *Meas Sci Technol* 2012;23:085901. doi:10.1088/0957-0233/23/8/085901.
- [46] Miller JA. From STM to nanomemory: a transfer of technology feasibility study 1994.
- [47] Miller JA, Hocken R, Smith ST, Harb S. X-ray calibrated tunneling system utilizing a dimensionally stable nanometer positioner. *Precis Eng* 1996;18:95–102. doi:10.1016/0141-6359(96)00037-2.
- [48] Lin W, Dacheng L, Mang C, Koenders L, Kuertgens U, Becker P. Angstrom ruler for high-accuracy pitch measurement. *Appl Opt* 2000;39:4535. doi:10.1364/AO.39.004535.
- [49] Lin W, Kuertgens U, Becker P, Koenders L, Dacheng L, Mang C. Calibration of standards for precision pitch measurement in the nanometre region by combined scanning tunnelling microscopy and x-ray interferometry. *Nanotechnology* 1999;10:412–7. doi:10.1088/0957-4484/10/4/309.
- [50] Yacoot A, Kuertgens U, Koenders L, Weimann T. A combined scanning tunnelling microscope and x-ray interferometer. *Meas Sci Technol* 2001;12:1660–5. doi:10.1088/0957-0233/12/10/306.
- [51] Park J, Lee M-Y, Lee D-Y. A nano-metrology system with a two-dimensional combined optical and X-ray interferometer and an atomic force microscope. *Microsyst Technol* 2009;15:1879–84. doi:10.1007/s00542-009-0916-0.
- [52] Chetwynd DG, Siddons DP, Bowen DK. X-ray interferometer calibration of microdisplacement transducers. *J Phys E* 1983;16:871–4. doi:10.1088/0022-3735/16/9/013.

- [53] Bowen DK, Chetwynd DG, Davies ST. Calibration Of Surface Roughness Transducers At Angstrom Levels Using X-Ray Interferometry. In: Marshall GF, editor. vol. 0563, International Society for Optics and Photonics; 1985, p. 412. doi:10.1117/12.949695.
- [54] Bowen DK, Chetwynd DG, Schwarzenberger DR. Sub-nanometre displacements calibration using X-ray interferometry. *Meas Sci Technol* 1990;1:107. doi:10.1088/0957-0233/1/2/002.
- [55] Becker P, Seyfried P. Calibration Of Optical And Mechanical Sensors By X-Ray Interferometry. In: Langenbeck P, editor. vol. 1015, International Society for Optics and Photonics; 1989, p. 124. doi:10.1117/12.949457.
- [56] Bowen DKK, Chetwynd DGG, Schwarzenberger DRR, Smith STT. Sub-nanometre transducer characterization by X-ray interferometry. *Precis Eng* 1990;12:165–71. doi:10.1016/0141-6359(90)90089-H.
- [57] Park JW, Eom C Il, Byun SH. Calibration of nanotransducer by a monolithic x-ray interferometer. In: Decker JE, Brown N, editors. vol. 5190, International Society for Optics and Photonics; 2003, p. 178. doi:10.1117/12.509725.
- [58] Yacoot A, Downs MJ. The use of x-ray interferometry to investigate the linearity of the NPL Differential Plane Mirror Optical Interferometer. *Meas Sci Technol* 2000;11:1126–30. doi:10.1088/0957-0233/11/8/305.
- [59] Yacoot A, Cross N. Measurement of picometre non-linearity in an optical grating encoder using x-ray interferometry. *Meas Sci Technol* 2003;14:148–52. doi:10.1088/0957-0233/14/1/321.
- [60] Chetwynd DG, Harb SM, Krylova NO, Smith ST. The feasibility of extended range monolithic X-ray interferometric calibrators. *Nanotechnology* 1993;4:183. doi:10.1088/0957-4484/4/4/002.

- [61] Chetwynd DG, Krylova NO, Smith ST. Metrological x-ray interferometry in the micrometre region. *Nanotechnology* 1996;7:1–12. doi:10.1088/0957-4484/7/1/001.
- [62] Becker P, Stümpel J. X-ray Optics in Metrological Applications. *Metrologia* 1990;27:127–32. doi:10.1088/0026-1394/27/3/003.
- [63] Windisch D, Becker P. Angular measurements with X-ray interferometry. *J Appl Crystallogr* 1992;25:377–83. doi:10.1107/S0021889891014231.
- [64] Kuetsgens U, Becker P. X-ray angle interferometry: a practical set-up for calibration in the microrad range with nanorad resolution. *Meas Sci Technol* 1998;9:1072–5. doi:10.1088/0957-0233/9/7/011.
- [65] Chetwynd DG, Schwarzenberger DR, Bowen DK. Two-dimensional X-ray interferometry. *Nanotechnology* 1990;1:19–26. doi:10.1088/0957-4484/1/1/004.
- [66] Chetwynd DG, Krylova NO, Bryanston-Cross PJ, Wang Z. Applications of x-ray interferometer-generated Moiré patterns. *Nanotechnology* 1998;9:125–32. doi:10.1088/0957-4484/9/2/015.
- [67] Carmignato S, Dewulf W, Leach R. *Industrial X-ray computed tomography*. Springer; 2018.
- [68] De Chiffre L, Carmignato S, Kruth J-P, Schmitt R, Weckenmann A. Industrial applications of computed tomography. *CIRP Ann* 2014;63:655–77. doi:10.1016/J.CIRP.2014.05.011.
- [69] Kruth JPP, Bartscher M, Carmignato S, Schmitt R, Chiffre L De, Weckenmann A, et al. Computed tomography for dimensional metrology. *CIRP Ann* 2011;60:821–42. doi:10.1016/J.CIRP.2011.05.006.
- [70] du Plessis A, le Roux SG. Standardized X-ray tomography testing of additively manufactured parts: A round robin test. *Addit Manuf* 2018;24:125–36.

doi:10.1016/J.ADDMA.2018.09.014.

- [71] Zanini F, Sbettega E, Carmignato S. X-ray computed tomography for metal additive manufacturing: challenges and solutions for accuracy enhancement. *Procedia CIRP* 2018;75:114–8. doi:10.1016/J.PROCIR.2018.04.050.
- [72] Thompson A, Maskery I, Leach RK. X-ray computed tomography for additive manufacturing: a review. *Meas Sci Technol* 2016;27:072001. doi:10.1088/0957-0233/27/7/072001.
- [73] Thompson A, Senin N, Maskery I, Leach R. Effects of magnification and sampling resolution in X-ray computed tomography for the measurement of additively manufactured metal surfaces. *Precis Eng* 2018;53:54–64. doi:10.1016/J.PRECISIONENG.2018.02.014.
- [74] Nikishkov Y, Airolidi L, Makeev A. Measurement of voids in composites by X-ray Computed Tomography. *Compos Sci Technol* 2013;89:89–97. doi:10.1016/J.COMPSCITECH.2013.09.019.
- [75] Carmignato S, Dreossi D, Mancini L, Marinello F, Tromba G, Savio E. Testing of x-ray microtomography systems using a traceable geometrical standard. *Meas Sci Technol* 2009;20:084021. doi:10.1088/0957-0233/20/8/084021.
- [76] Villarraga-Gómez H, Lee C, Smith ST. Dimensional metrology with X-ray CT: A comparison with CMM measurements on internal features and compliant structures. *Precis Eng* 2018;51:291–307. doi:10.1016/J.PRECISIONENG.2017.08.021.
- [77] Hiller J, Reindl LM. A computer simulation platform for the estimation of measurement uncertainties in dimensional X-ray computed tomography. *Measurement* 2012;45:2166–82. doi:http://dx.doi.org/10.1016/j.measurement.2012.05.030.
- [78] Weckenmann A, Krämer P. Assessment of measurement uncertainty caused in the

preparation of measurements using computed tomography. Proc. XIX IMEKO World Congr. Fundam. Appl. Metrol. Lisbon, Port., 2009, p. 6–11.

- [79] Müller P. Coordinate Metrology by Traceable Computed Tomography. Kgs. Lyngby: 2013.
- [80] Schmitt R, Niggemann C. Uncertainty in measurement for x-ray-computed tomography using calibrated work pieces. Meas Sci Technol 2010;21:054008. doi:10.1088/0957-0233/21/5/054008.
- [81] Carmignato S, Savio E, De Chiffre L. CT techniques for reconstructing 3D geometrical models of complex parts: an approach for traceability establishment and uncertainty evaluation. IMEKO Int. Symp. Mediterr. Conf. Meas., 2004, p. 387–90.
- [82] Carmignato S, Savio E. Traceable volume measurements using coordinate measuring systems. CIRP Ann 2011;60:519–22. doi:10.1016/J.CIRP.2011.03.061.
- [83] Villarraga-Gómez H, **Thousand J**, Morse E, Smith S. CT Measurements and their estimated uncertainty: The significance of temperature and bias determination. 15th Int. Conf. Metrol. Prop. Eng. Surfaces, Charlotte, NC: American Society for Precision Engineering; 2015.
- [84] Stolfi A, De Chiffre L. CT crown for on-machine scale calibration in Computed Tomography. Proc. euspen's 16th Int. Conf. Exhib., 2016.
- [85] Staude A, Goebbels J, Ehrig K, Bartscher M, Koch M, Neuschaefer-Rube U. A new test piece for geometry and defect measurements with micro-CT. Proc. Eur. Conf. non-destructive Test. ECNDT, 2010.
- [86] Kraemer A, Lanza G. Assessment of the Measurement Procedure for Dimensional Metrology with X-ray Computed Tomography. Procedia CIRP 2016;43:362–7. doi:10.1016/J.PROCIR.2016.02.018.

- [87] Müller P. Use of reference objects for correction of measuring errors in X-ray computed tomography 2010.
- [88] Müller P, Hiller J, Cantatore A, Tosello G, De Chiffre L. New reference object for metrological performance testing of industrial CT systems. 12th euspen Int. Conf., 2012.
- [89] Stolfi A, De Chiffre L. 3D artefact for concurrent scale calibration in Computed Tomography. CIRP Ann 2016;65:499–502. doi:10.1016/J.CIRP.2016.04.069.
- [90] Léonard F, Brown SB, Withers PJ, Mummery PM, McCarthy MB. A new method of performance verification for x-ray computed tomography measurements. Meas Sci Technol 2014;25:065401. doi:10.1088/0957-0233/25/6/065401.
- [91] Neukamm M, Bartscher M, Hilpert U, Ehrig K, Goebbels J, Staude A. Untersuchung der Oberflächenextraktion industrieller CT am Beispiel sphärischer Formelemente 2009.
- [92] Bartscher M, Hilpert U, Goebbels J, Weidemann G. Enhancement and Proof of Accuracy of Industrial Computed Tomography (CT) Measurements. CIRP Ann 2007;56:495–8. doi:10.1016/J.CIRP.2007.05.118.
- [93] Kasperl S, Hiller J, Krumm M. Computed tomography metrology in industrial research and development. Mater Test 2009;51:405–11.
- [94] Borges de Oliveira F, Bartscher M, Neuschaefer-Rube U, Tutsch R, Hiller J. Multi-material Acceptance Testing for CT-Based Coordinate Measurement Systems, Springer, Cham; 2019, p. 131–54. doi:10.1007/978-3-030-18177-2_14.
- [95] Bartscher M, Illemann J, Neuschaefer-Rube U. ISO test survey on material influence in dimensional computed tomography. Case Stud Nondestruct Test Eval 2016. doi:10.1016/j.csndt.2016.04.001.

- [96] Marinello F, Savio E, Carmignato S, De Chiffre L. Calibration artefact for the microscale with high aspect ratio: The fiber gauge. *CIRP Ann* 2008;57:497–500. doi:10.1016/J.CIRP.2008.03.086.
- [97] Kiekens K, Welkenhuyzen F, Tan Y, Bleys P, Voet A, Dewulf W, et al. A test object for calibration and accuracy assessment in X-ray CT metrology. 10th Int. Symp. Meas. Qual. Control 2010, ISMQC 2010, 2010, p. 528–31.
- [98] Kiekens K, Welkenhuyzen F, Tan Y, Bleys P, Voet A, Kruth J-P, et al. A test object with parallel grooves for calibration and accuracy assessment of industrial computed tomography (CT) metrology. *Meas Sci Technol* 2011;22:115502. doi:10.1088/0957-0233/22/11/115502.
- [99] Angel J, De Chiffre L, Kruth JP, Tan Y, Dewulf W. Performance evaluation of CT measurements made on step gauges using statistical methodologies. *CIRP J Manuf Sci Technol* 2015;11:68–72. doi:10.1016/J.CIRPJ.2015.08.002.
- [100] Stolfi A, De Chiffre L. Interlaboratory comparison of a physical and a virtual assembly measured by CT. *Precis Eng* 2018;51:263–70. doi:10.1016/J.PRECISIONENG.2017.08.018.
- [101] Xue L, Suzuki H, Ohtake Y, Fujimoto H, Abe M, Sato O, et al. A method for improving measurement accuracy of cylinders in dimensional CT metrology. *Comput Des* 2015;69:25–34. doi:10.1016/J.CAD.2015.07.015.
- [102] Hiller J, Hornberger P. Measurement accuracy in X-ray computed tomography metrology: Toward a systematic analysis of interference effects in tomographic imaging. *Precis Eng* 2016;45:18–32. doi:10.1016/J.PRECISIONENG.2015.12.003.
- [103] Heinzl C, Kastner J, Georgi B, Lettenbauer H. Comparison of surface detection methods to evaluate cone beam computed tomography data for three dimensional metrology. *Int. Symp. Digit. Ind. Radiol. Comput. Tomogr.*, 2007.

- [104] Heinzl C, Klingsberger R, Kastner J, Gröller ME. Robust Surface Detection for Variance Comparison and Dimensional Measurement. EuroVis, 2006, p. 75–82.
- [105] Carmignato S. Accuracy of industrial computed tomography measurements: Experimental results from an international comparison. CIRP Ann 2012;61:491–4. doi:10.1016/J.CIRP.2012.03.021.
- [106] QRM GmbH. QRM GmbH - Quality Assurance in Radiology and Medicine n.d. <http://www.qrm.de/content/products/microct.htm> (accessed September 17, 2018).
- [107] X-RAY Wor X GmbH. JIMA resolution tests n.d. <https://www.x-ray-worx.com/x-ray-worx/index.php/mikrofokus-roentgenroehren-produktuebersicht/jima-aufloesungstests> (accessed September 24, 2018).
- [108] Japan Inspection Instruments Manufacturers' Association. Manufacturers Association for Non-Destructive Testing (NDT) : Application Usage n.d. <https://www.jima.jp/english/assen-e.html> (accessed August 20, 2019).
- [109] NIST. X-Ray Mass Attenuation Coefficients - Molybdenum n.d. <https://physics.nist.gov/PhysRefData/XrayMassCoef/ElemTab/z42.html> (accessed August 27, 2019).
- [110] Bjeoumikhov A, Bjeoumikhova S. Capillary Optics for X-Rays. In: Erko A, Idir M, Krist T, Michette AG, editors. Mod. Dev. X-Ray Neutron Opt., Berlin, Heidelberg: Springer Berlin Heidelberg; 2008, p. 287–306. doi:10.1007/978-3-540-74561-7_18.
- [111] NIST. X-Ray Mass Attenuation Coefficients - Lead n.d. <https://physics.nist.gov/PhysRefData/XrayMassCoef/ElemTab/z82.html> (accessed August 27, 2019).
- [112] NIST. X-Ray Mass Attenuation Coefficients - Silicon n.d. <https://physics.nist.gov/PhysRefData/XrayMassCoef/ElemTab/z14.html> (accessed August 27, 2019).

- [113] Knoll GF. Radiation detection and measurement. New York: John Wiley; 2010.
- [114] Warren BE. X-ray Diffraction. Dover Publications; 1990.
- [115] Davis TJ, Stevenson AW. Direct Measure of the Phase Shift of an X-ray Beam. *J Opt Soc Am A* 1996;13:1193–1198. doi:10.1364/JOSAA.13.001193.
- [116] **Thousand JD**, Smith ST. DRIE process performance for manufacture of an artifact for x-ray phase-contrast and micro-CT characterization. 28th Annu. Meet. Am. Soc. Precis. Eng., St. Paul, MN: American Society for Precision Engineering; 2013, p. 389–92.
- [117] Hecht E. Optics. 4th ed. San Francisco: Addison-Wesley; 2002.
- [118] MarShield. Clear Lead Acrylic | Leaded Acrylic Sheet | MarShield Custom Radiation n.d. <https://marshield.com/lead-acrylic-2/> (accessed September 22, 2018).
- [119] Hull. AW. A New Method of X-Ray Crystal Analysis. *Phys Rev* 1917;10:661–96. doi:10.1103/PhysRev.10.661.
- [120] Barney JA, **Thousand JD**, Smith ST. Design of a mechanical actuator for phase control of an x-ray interferometer. 29th Annu. Meet. Am. Soc. Precis. Eng., Boston, MA: American Society for Precision Engineering; 2014, p. 302–6.
- [121] Fluke Biomedical. 451B Ion Chamber Survey Meter with Beta Slide n.d. <https://www.flukebiomedical.com/products/radiation-measurement/radiation-safety/451b-ion-chamber-survey-meter-beta-slide> (accessed February 23, 2019).
- [122] Digilent Inc. Debouncing With the NE555 Timer IC [Reference.Digilentinc] n.d. <https://reference.digilentinc.com/learn/microprocessor/tutorials/debouncing-with-ne555-timer/start> (accessed September 22, 2018).
- [123] **Thousand JD**, Smith ST. Development of an x-ray interferometric and micro-CT

facility. 29th Annu. Meet. Am. Soc. Precis. Eng., Boston, MA: American Society for Precision Engineering; 2014, p. 375–9.

- [124] Atik Cameras. Atik Series 4 User Manual 2012.
- [125] Photometrics White Papers: Read Noise for Scientific CCD Cameras n.d. <https://www.photometrics.com/resources/whitepapers/read-noise> (accessed March 27, 2019).
- [126] Touš J, Horváth M, Pína L, Blažek K, Sopko B. High-resolution application of YAG:Ce and LuAG:Ce imaging detectors with a CCD X-ray camera. Nucl Instruments Methods Phys Res Sect A Accel Spectrometers, Detect Assoc Equip 2008;591:264–7. doi:<http://dx.doi.org/10.1016/j.nima.2008.03.070>.
- [127] Plöchl A, Kräuter G. Wafer direct bonding: tailoring adhesion between brittle materials. Mater Sci Eng R Reports 1999;25:1–88. doi:[http://dx.doi.org/10.1016/S0927-796X\(98\)00017-5](http://dx.doi.org/10.1016/S0927-796X(98)00017-5).
- [128] Kern W. The Evolution of Silicon Wafer Cleaning Technology. J Electrochem Soc 1990;137:1887. doi:10.1149/1.2086825.
- [129] Gale GW, Cui H, Reinhardt KA. Chapter 4 - Aqueous Cleaning and Surface Conditioning Processes. In: Reinhardt KA, Kern WBT-H of SWCT (Third E, editors., William Andrew Publishing; 2018, p. 185–252. doi:<https://doi.org/10.1016/B978-0-323-51084-4.00004-6>.
- [130] Madou MJ. Fundamentals of microfabrication : the science of miniaturization 2002.
- [131] Robbins H, Schwartz B. Chemical Etching of Silicon, I. The System HF, HNO₃, and H₂O. J Electrochem Soc 1959;106:505–8. doi:10.1149/1.2427397.
- [132] Gosálvez MA, Zubel I, Viinikka E. Wet Etching of Silicon. Handb Silicon Based MEMS Mater Technol 2015:470–502. doi:10.1016/B978-0-323-29965-7.00022-1.

- [133] Laermer F, Schilp A. Method of anisotropically etching silicon 1996.
- [134] Dixit P, Miao J. Effect of SF₆ flow rate on the etched surface profile and bottom grass formation in deep reactive ion etching process. *J Phys Conf Ser* 2006;34:577.
- [135] Tan Y, Zhou R, Zhang H, Lu G, Li Z. Modeling and simulation of the lag effect in a deep reactive ion etching process. *J Micromechanics Microengineering* 2006;16:2570–5. doi:10.1088/0960-1317/16/12/008.
- [136] Hooda MK, Wadhwa M, Verma S, Nayak MM, George PJ, Paul AK. A systematic study of DRIE process for high aspect ratio microstructuring. *Vacuum* 2010;84:1142–8. doi:10.1016/j.vacuum.2010.01.052.
- [137] Oh J, Yuan H-C, Branz HM. An 18.2%-efficient black-silicon solar cell achieved through control of carrier recombination in nanostructures. *Nat Nanotechnol* 2012;7:743–8. doi:10.1038/nnano.2012.166.
- [138] Liu X, Coxon PR, Peters M, Hoex B, Cole JM, Fray DJ. Black silicon: fabrication methods, properties and solar energy applications. *Energy Environ Sci* 2014;7:3223–63. doi:10.1039/C4EE01152J.
- [139] Marvell Nanofabrication Laboratory. Equipment Manual Contents n.d. https://nanolab.berkeley.edu/public/manuals/equipment_manual.shtml (accessed January 2, 2019).
- [140] Kuo-Shen Chen, Ayon AA, Zhang X, Spearing SM. Effect of process parameters on the surface morphology and mechanical performance of silicon structures after deep reactive ion etching (DRIE). *Microelectromechanical Syst J* 2002;11:264–75. doi:10.1109/JMEMS.2002.1007405.
- [141] Schmidt MA. Wafer-to-wafer bonding for microstructure formation. *Proc IEEE* 1998;86:1575–85. doi:10.1109/5.704262.

- [142] Dresbach C, Krombholz A, Ebert M, Bagdahn J. Mechanical properties of glass frit bonded micro packages. *Microsyst Technol* 2006;12:473–80. doi:10.1007/s00542-005-0031-9.
- [143] Wiemer M, Jia C, Toepper M, Hauck K. Wafer Bonding with BCB and SU-8 for MEMS Packaging. 2006 1st Electron. Syst. Technol. Conf., IEEE; 2006, p. 1401–5. doi:10.1109/ESTC.2006.280194.
- [144] Wolffenbuttel RF. Low-temperature intermediate Au-Si wafer bonding; eutectic or silicide bond. *Sensors Actuators A Phys* 1997;62:680–6. doi:10.1016/S0924-4247(97)01550-1.
- [145] Knechtel R. Glass Frit Wafer Bonding. *Handb. Wafer Bond.*, Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA; 2012, p. 1–17. doi:10.1002/9783527644223.ch1.
- [146] Niklaus F, Lu J-Q. Polymer Adhesive Wafer Bonding. *Handb. Wafer Bond.*, Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA; 2012, p. 33–61. doi:10.1002/9783527644223.ch3.
- [147] Koyanagi M, Motoyoshi M. Eutectic Au-In Bonding. *Handb. Wafer Bond.*, Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA; 2012, p. 139–59. doi:10.1002/9783527644223.ch8.
- [148] H. Ko W, T. Suminto J, J. Yeh G. Bonding techniques for microsensors. vol. 20. 1985.
- [149] Reiche M, Gösele U. Direct Wafer Bonding. *Handb. Wafer Bond.*, Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA; 2012, p. 81–100. doi:10.1002/9783527644223.ch5.
- [150] University Wafer. Silicon Wafer FAQ n.d. <http://universitywafer.com/phone/silicon-wafer-faq.html> (accessed December 14,

2018).

- [151] Miki N, Zhang X, Khanna R, Ayón AA, Ward D, Spearing SM. Multi-stack silicon-direct wafer bonding for 3D MEMS manufacturing. *Sensors Actuators A Phys* 2003;103:194–201. doi:10.1016/S0924-4247(02)00332-1.
- [152] Rotondaro ALP, Muscat AJ. Chapter 6 - Gas-Phase Wafer Cleaning Technology. In: Reinhardt KA, Kern WBT-H of SWCT (Third E, editors., William Andrew Publishing; 2018, p. 305–77. doi:<https://doi.org/10.1016/B978-0-323-51084-4.00006-X>.
- [153] Hess DW, Reinhardt KA. Chapter 7 - Plasma Stripping, Cleaning, and Surface Conditioning. In: Reinhardt KA, Kern WBT-H of SWCT (Third E, editors., William Andrew Publishing; 2018, p. 379–455. doi:<https://doi.org/10.1016/B978-0-323-51084-4.00007-1>.
- [154] Reidy RF, Lauerhaas JM. Chapter 8 - Cryogenic Aerosols and Supercritical Fluid Cleaning and Surface Conditioning*. In: Reinhardt KA, Kern WBT-H of SWCT (Third E, editors., William Andrew Publishing; 2018, p. 457–502. doi:<https://doi.org/10.1016/B978-0-323-51084-4.00008-3>.
- [155] Wiemer M, Wuensch D, Braeuer J, Gessner T. Plasma-Activated Bonding. *Handb. Wafer Bond.*, Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA; 2012, p. 101–18. doi:10.1002/9783527644223.ch6.
- [156] Masteika V, Kowal J, Braithwaite NSJ, Rogers T. A Review of Hydrophilic Silicon Wafer Bonding. *ECS J Solid State Sci Technol* 2014;3:Q42–54. doi:10.1149/2.007403jss.
- [157] Gösele U, Tong Q-Y. SEMICONDUCTOR WAFER BONDING. *Annu Rev Mater Sci* 1998;28:215–41. doi:10.1146/annurev.matsci.28.1.215.
- [158] Tong Q -Y., Schmidt E, Gösele U, Reiche M. Hydrophobic silicon wafer bonding.

Appl Phys Lett 1994;64:625–7. doi:<http://dx.doi.org/10.1063/1.111070>.

- [159] **Thousand JD**, Smith ST. Development of calibration artefacts for x-ray radiographic and micro-CT metrology, Proc. 16th Int. Conf. Eur. Soc. Precis. Eng. Nanotechnol., Nottingham, UK: European Society for Precision Engineering and Nanotechnology; 2016.
- [160] **Thousand JD**, Smith ST. A direct silicon bonded reference object for performance assessment of computed tomography systems. *Precis Eng* 2019;58:16–24.
- [161] Zhao J, Hu X, Zou J, Zhao G, Lv H, Xu L, et al. Method for correction of rotation errors in Micro-CT System. *Nucl Instruments Methods Phys Res Sect A Accel Spectrometers, Detect Assoc Equip* 2016;816:149–59. doi:10.1016/J.NIMA.2016.01.051.
- [162] Flay N, Sun W, Brown S, Leach R, Blumensath T. Investigation of the Focal Spot Drift in Industrial Cone-beam X-ray Computed Tomography. 2015.
- [163] Hiller J, Maisl M, Reindl LM. Physical characterization and performance evaluation of an x-ray micro-computed tomography system for dimensional metrology applications. *Meas Sci Technol* 2012;23:085404.
- [164] Weiss D, Lonardoni R, Deffner A, Kuhn C. Geometric image distortion in flat-panel X-ray detectors and its influence on the accuracy of CT-based dimensional measurements. 2012.
- [165] Solehmainen K, Aalto T, Dekker J, Kapulainen M, Harjanne M, Kukli K, et al. Dry-etched silicon-on-insulator waveguides with low propagation and fiber-coupling losses. *J Light Technol* 2005;23:3875–80. doi:10.1109/JLT.2005.857750.
- [166] Gao F, Ylinen S, Kainlahti M, Kapulainen M. Smooth silicon sidewall etching for waveguide structures using a modified Bosch process. *J Micro/Nanolithography, MEMS, MOEMS* 2014;13:013010. doi:10.1117/1.JMM.13.1.013010.

- [167] Xie H, Hussain D, Yang F, Sun L. Atomic force microscopy deep trench and sidewall imaging with an optical fiber probe. *Rev Sci Instrum* 2014;85:123704. doi:10.1063/1.4903466.
- [168] Dai G, Pohlenz F, Danzebrink H-U, Xu M, Hasche K, Wilkening G. Metrological large range scanning probe microscope. *Rev Sci Instrum* 2004;75:962–9. doi:10.1063/1.1651638.
- [169] Dixon R, Fu J, Orji N, Guthrie W, Allen R, Cresswell M. CD-AFM reference metrology at NIST and SEMATECH. In: Silver RM, editor. vol. 5752, International Society for Optics and Photonics; 2005, p. 324. doi:10.1117/12.601972.
- [170] Dixon RG, Allen RA, Guthrie WF, Cresswell MW. Traceable calibration of critical-dimension atomic force microscope linewidth measurements with nanometer uncertainty. *J Vac Sci Technol B Microelectron Nanom Struct* 2005;23:3028. doi:10.1116/1.2130347.
- [171] Dixon R, G. Orji N, Potzick J, Fu J, Allen R, Cresswell MW, et al. Photomask Applications of Traceable Atomic Force Microscope Dimensional Metrology at NIST. *Proc. SPIE - Int. Soc. Opt. Eng.*, vol. 6730, 2007. doi:10.1117/12.746755.
- [172] Schwartz B, Robbins H. Chemical Etching of Silicon: IV . Etching Technology. *J Electrochem Soc* 1976;123:1903–9. doi:10.1149/1.2132721.

APPENDIX A: SUPPLEMENTARY FIGURES

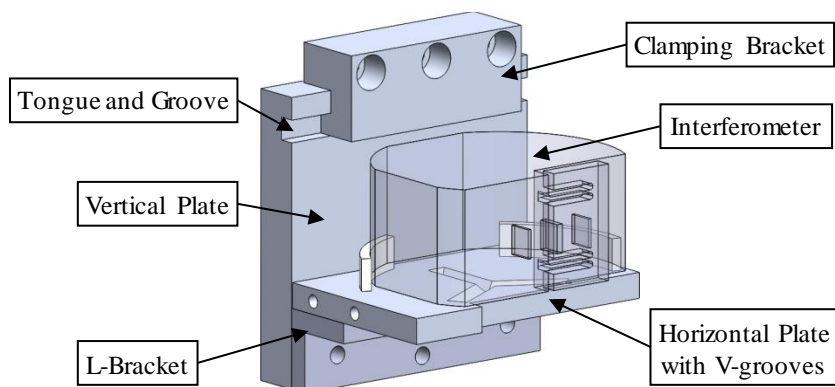


Figure A.1. First attempt at a kinematic mount for the interferometer. The flaw in a single-clamp design was as the clamp was engaged, the vertical plate rotated.

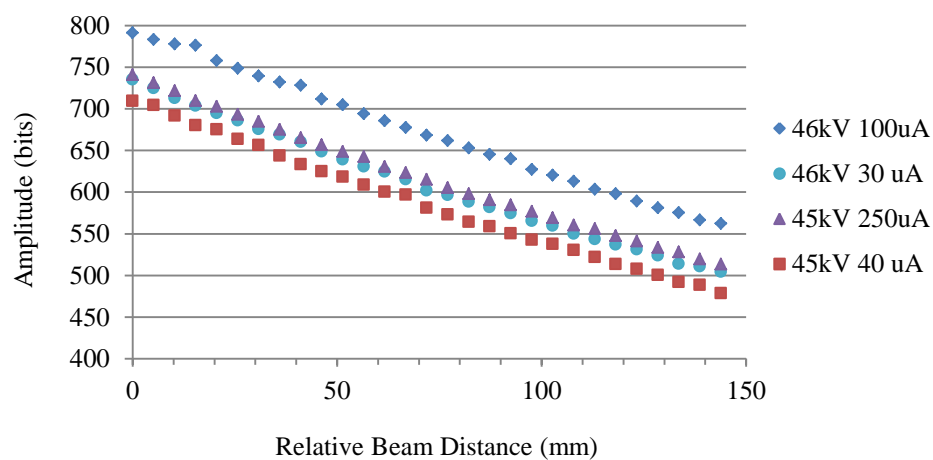


Figure A.2. Comparison of the beam amplitude at different distances from the X-ray source.

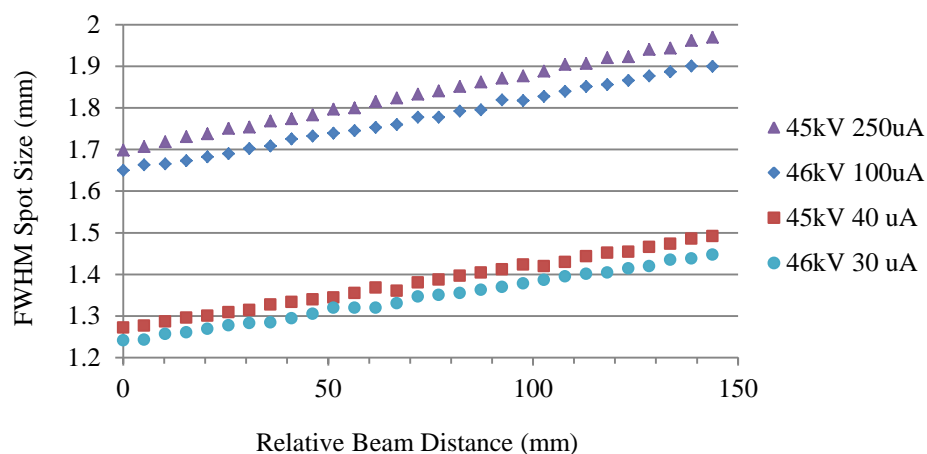


Figure A.3. Comparison of the FWHM spot sizes at different distances from the X-ray source showing differences in voltages and current do not significantly affect the cone angle.

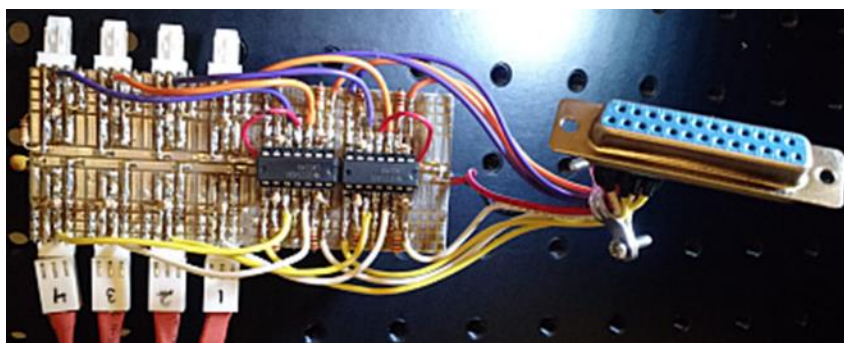


Figure A.4. Photograph of the temperature sensor circuit. The circuit was constructed to accommodate up to eight LM35 sensors one time.

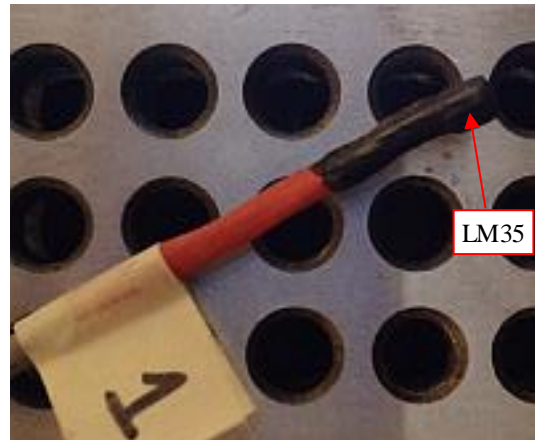


Figure A.5. The LM35 sensor was coated with a brushed-on water resistant polymer after heat shrinking the electrical contacts.

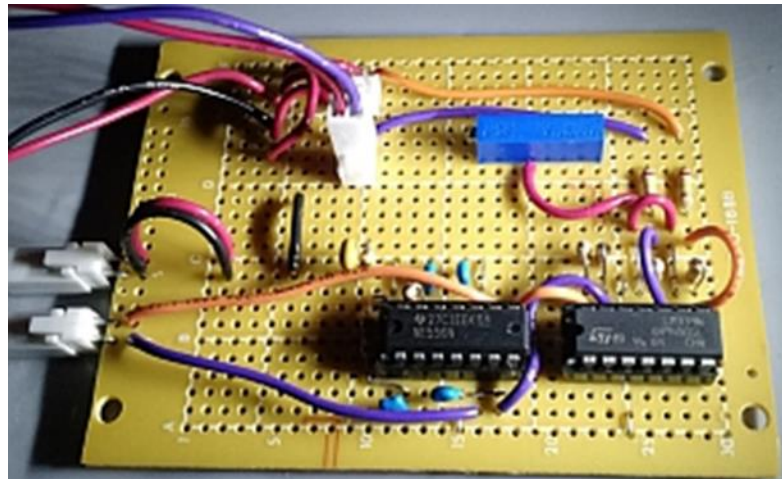


Figure A.6. Photograph of the GM Tube sensor circuit.

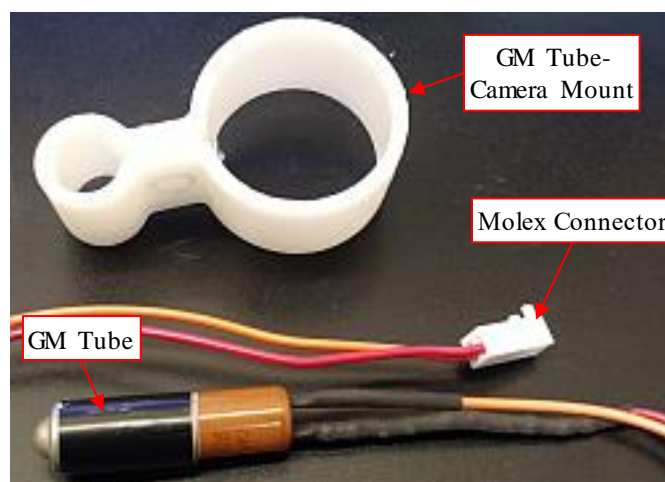


Figure A.7. Geiger-Müller tube with a 3D printed mount for attachment to the in-line camera configuration.

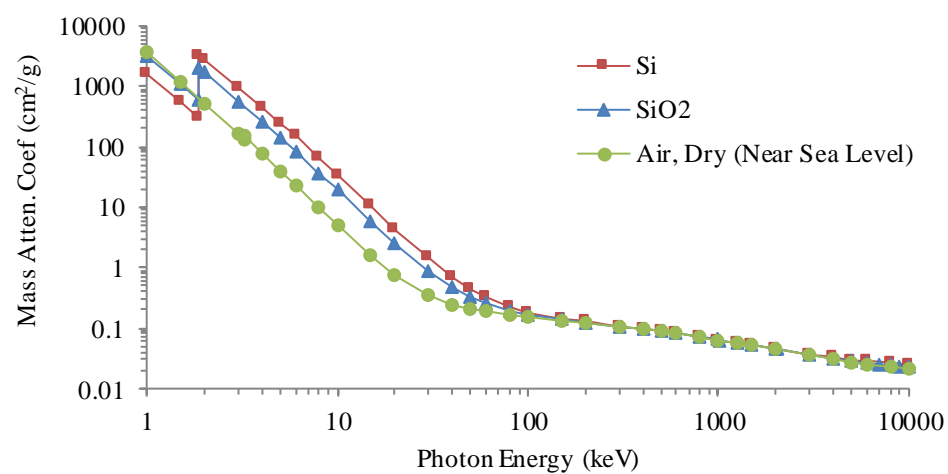


Figure A.8. Plot of the mass attenuation coefficient for silicon, silica, and air.

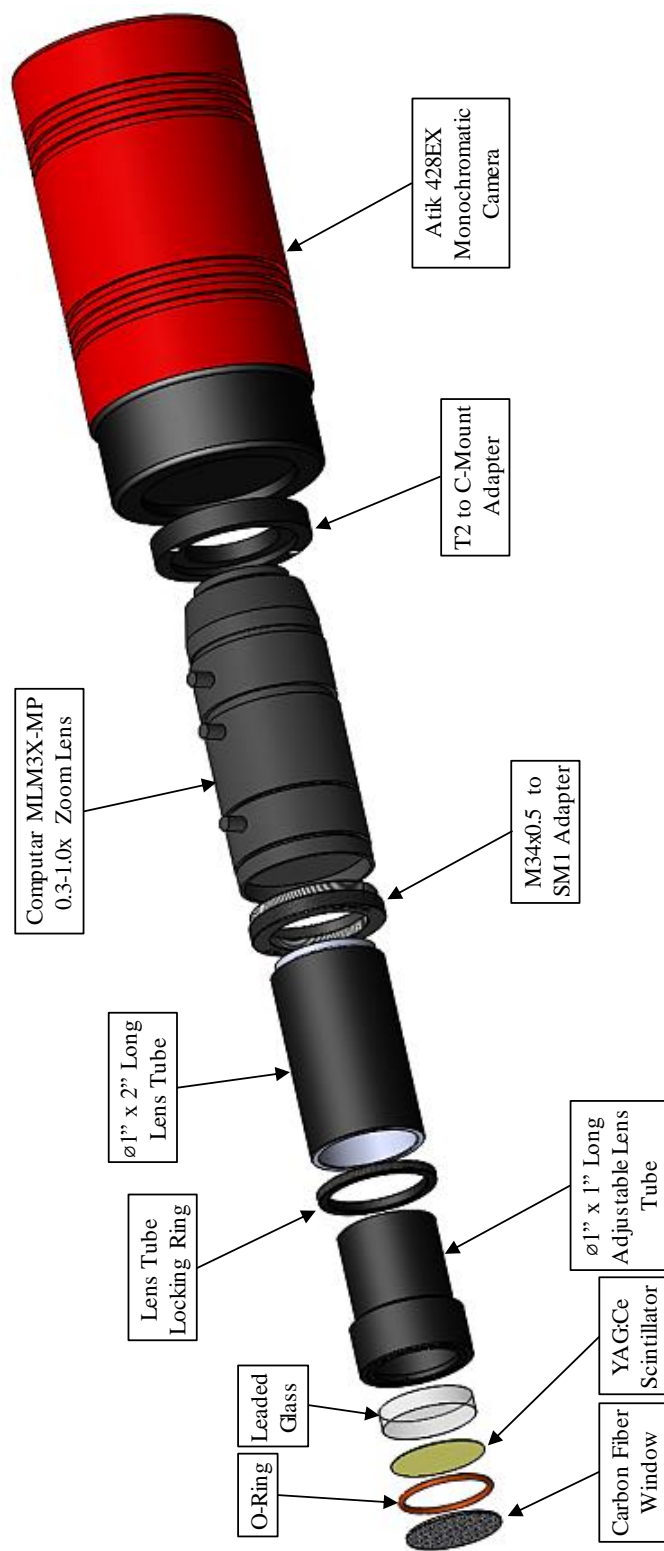


Figure A.9. Exploded solid model showing the components for the original in-line configuration.

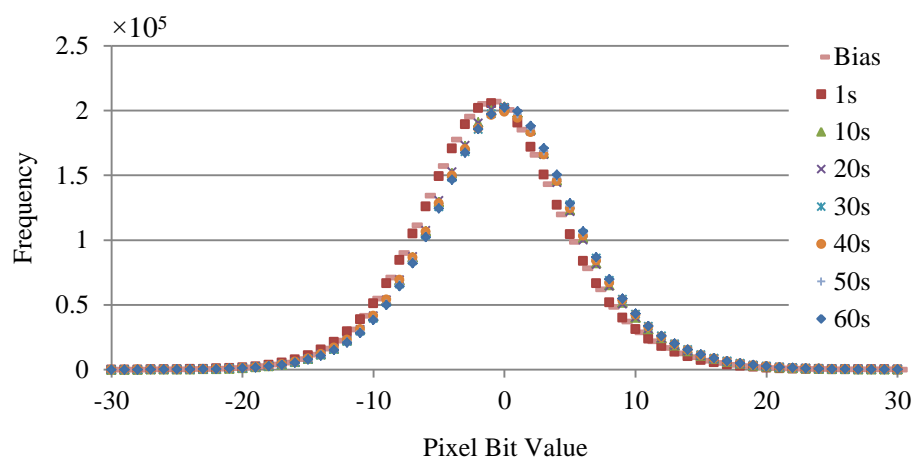


Figure A.10. Histogram of the dark frames after removing tilt using the same tilt profile.

APPENDIX B: INCLINOMETER FOR X-RAY ROTARY TABLE

The facility design requires rotating the X-ray generator when setup for interferometry. To monitor the change in angle will be an inclinometer. The inclinometer utilizes an electrolytic dual-axis tilt sensor manufactured by the Fredericks Company (model 0729-1752-99). Each each axis has an operating range of $\pm 60^\circ$, $\pm 25^\circ$ linear range, $\pm 0.1^\circ$ repeatability, and 0.003° resolution with a 100 ms time constant. The tilt sensor communicates via RS-232 to an Adafruit Feather M0 microcontroller. The microcontroller has a 48 MHz ARM Cortex M0 processor with 3.3 V logic. Wireless data transmission is made possible by an HC-05 Bluetooth module. Alternatively, the Adafruit FeatherWing 128x32 OLED display board displays the readings and provides buttons for user control. Powering the device is a 3.7 V, 500 mAh lithium ion battery that is recharged through the Feather's USB connection. The top case is printed plastic mounted to a machined aluminum base. Three inset spheres with a neodymium magnet were added for kinematic mounting. A mating kinematic mount base was machined from aluminum for attaching to the X-ray generator. Pictures of the inclinometer and kinematic base are shown in Figure B.1.

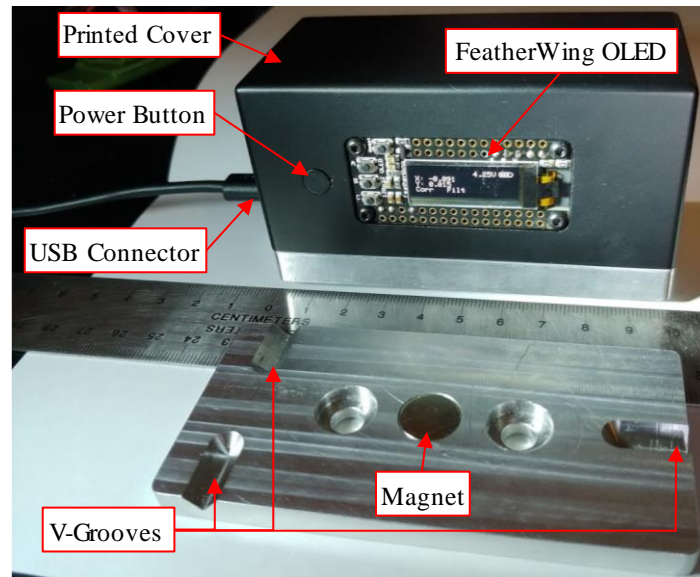


Figure B.1. Custom inclinometer with kinematic mount.

APPENDIX C: REFERENCE OBJECT POLYNOMIAL SIMULATION

C.1 Overview

After designing the various features of the reference object and writing an algorithm to extract feature properties from real data, a simulation for the multi degree polynomials was written using MATLAB® script. By simulating the data, the impact from pixel resolution, tilt, and noise on the uncertainty of the curve fitting parameters can be assessed in a controlled manner. To run the simulation, initial values for the start and stop position along the x-axis, x-axis stepping increment, rotation transformation, pixel resolution, polynomial coefficients, a maximum random noise value, and a Gaussian filter cutoff are entered by the user. For the simulation to provide meaning information, simulated data must be manipulated to mimic characteristics of the instrument and the metrology algorithm.

C.2 Algorithm

An array of the X values is created from the start, stop, and step initial values. The step value must be set smaller than the resolution, at most 0.25% the pixel resolution. Y values were calculated by feeding the X values into a cubic polynomial. To simulate using a camera with square pixels to capture data, some rotation can be applied to the line followed by transforming the data from high density floating point into lower density integer values by dividing by the resolution and rounding. Having higher data density is important as the rotation and pixelating of the data can cause missing points as demonstrated in Figure C.1. Before fitting the polynomial, the exact rotational value is removed.

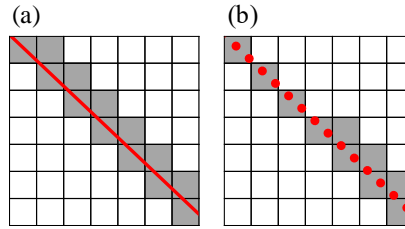


Figure C.1. (a) High density compared to (b) low density data and their effect on simulating pixelated data from a camera.

After pixelating, the next process is to imitate the edge detection on binary data by removing points that fall inside of a feature. The shaded region in Figure C.2(a) is an example of a section from a feature with a curved edge. When converting the line into pixel data, Figure B.2(b) is generated. However, when using edge detection, the result is similar to Figure C.2 (c) where only points near the transition are recorded. To replicate Figure C.2(c) using data similar to Figure C.2(b), an algorithm to detected and remove inner points was developed.

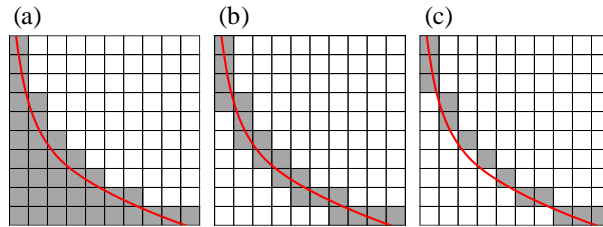


Figure C.2. (a) The shaded area is the feature with the red line indicating the feature's edge. (b) Theoretical line converted into pixelated data. (c) Pixelated data from edge detection missing points on the inside edge of the feature's edge.

As the simulated data is created from left to right, the algorithm steps through each point (x_n, y_n) and checks whether the previous point (x_{n-1}, y_{n-1}) and next point (x_{n+1}, y_{n+1}) are adjacent corners by calculating their distance apart. With the data in pixel form, when the distance is equal to the diagonal of a unit square, $\sqrt{2}$, the two points declared adjacent corners. If the points are adjacent corners, the slope m , y-intercepts b_1 and b_2 , and x direction \vec{x} are calculated from Equations (C.1) through Equation (C.4).

Using these calculated values, the logic table in Table C.1 is used to determine if the current point (x_n, y_n) is an inner or outer point.

$$m = \frac{y_{n+1} - y_{n-1}}{x_{n+1} - x_{n-1}} \quad (\text{C.1})$$

$$b_1 = y_{n-1} - mx_{n-1} \quad (\text{C.2})$$

$$b_2 = y_n - mx_n \quad (\text{C.3})$$

$$\vec{x} = x_{n+1} - x_{n-1} \quad (\text{C.4})$$

Table C.1. Logic to determine whether the point falls inside or outside the feature.

Logic	$\vec{x} \geq 0$	$\vec{x} < 0$
$b_1 > b_2$	Inside	Outside
$b_1 < b_2$	Outside	Inside

With the inner points removed, data is converted back into micrometers by multiplying by the resolution. Using MATLAB®'s polyfit function, coefficients of a best fit polynomial are calculated. The final process is calculating the root-mean-square (RMS) of the distances for each data point relatively to the best fit curve. The measurement is satisfied by the condition that the slope from the data point (x_2, y_2) to a point on the curve (x_1, y_1) is perpendicular to the tangent line y'_1 at that curve's point. This can be solved by substituting Equations (C.5) and (C.6) into Equation (C.7) and solving for x_1 . While it is possible multiple solutions exist, the real number closest to x_2 without an imaginary component is the solution. This solution is then substituted into Equation (C.5) to calculate

y_1 . Substituting the points into Equation (C.8) yields the distance. Once each points distance is calculated, the results are used to find the mean distance, deviation, and RMS.

$$y_1 = dx_1^3 + cx_1^2 + bx_1^1 + ax_1^0 \quad (C.5)$$

$$y_1' = 3dx_1^2 + 2cx_1^1 + bx_1^0 \quad (C.6)$$

$$y_1'(y_2 - y_1) + x_1 - x_2 = 0 \quad (C.7)$$

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (C.8)$$

As an added feature, there is an option to add noise to the line before converting into pixelated data. An array with more elements than the line's array and composed of random numbers of range ± 1 is passed through a Gaussian filter to generate a smooth and continuous noise profile. The noise array larger is larger to avoid edge distortion by the filter. Instead of superimposing the noise profile onto the line by summing the y-values directly, the superimposition is added normal to the tangent for each point along the theoretical. Therefore, a point of zero noise is always an intersection point between the original line and the noisy line.

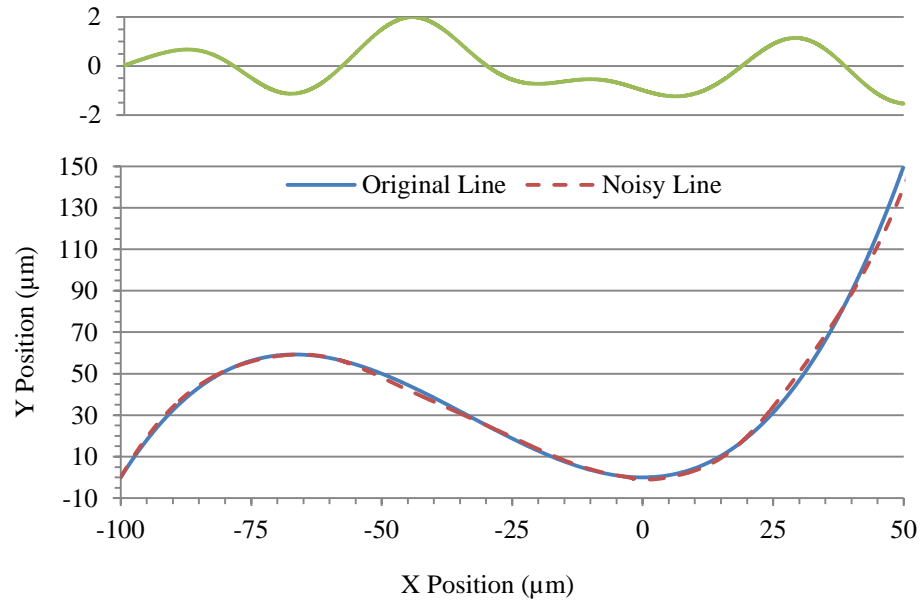


Figure C.3. Superimposing the noise profile (top) normal to the tangent of the original line.

C.3 Comparison to Actual Data

To determine how accurate the simulation mimics real data, the standard deviation of the tangent normal residuals was calculated for the largest asymmetric cubic curve and the second largest quadratic and symmetric cubic curves. Noise and rotation were not added to the simulated line. The spatial sampling of the simulation was set to $0.408 \mu\text{m}\cdot\text{pixel}^{-1}$, equaling the CSI with $20\times$ objective. Standard deviations are tabulated in Table C.2. Using the nominal coefficients yields a higher standard deviation than a best fit polynomial curve and from the simulation. However, with a maximum difference between the best fit and simulation values, the simulation is close to providing an accurate representation of the extracted line. Graphs of the residuals for each of the three shapes are provided in Figure C.4, Figure C.5, and Figure C.6. Similar patterns in the data can be seen in the residuals for each curve.

Table C.2. Comparison of 1σ standard deviation values between CSI data and simulation.

Feature	Nominal (nm)	Best Fit (nm)	Simulation (nm)
AC1	141	117	112
QC2	160	123	112
SC2	126	126	115

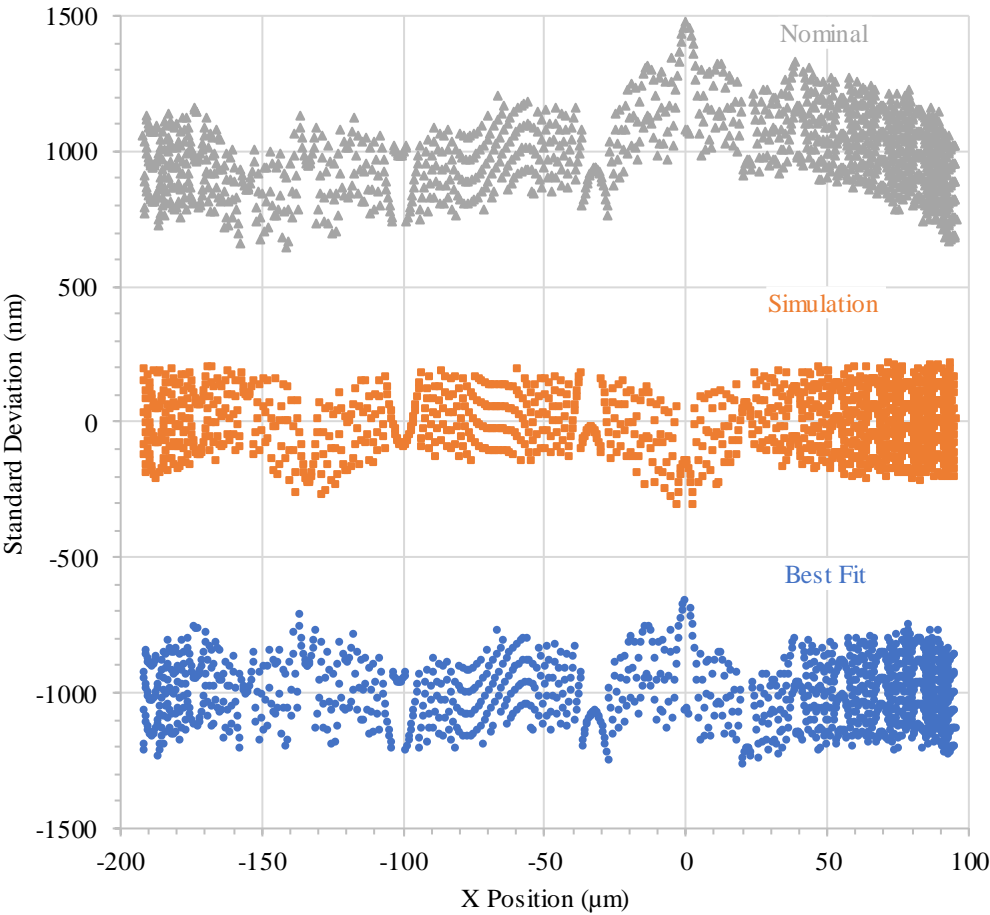


Figure C.4. Plot of the asymmetric cubic curve’s tangent normal distances.

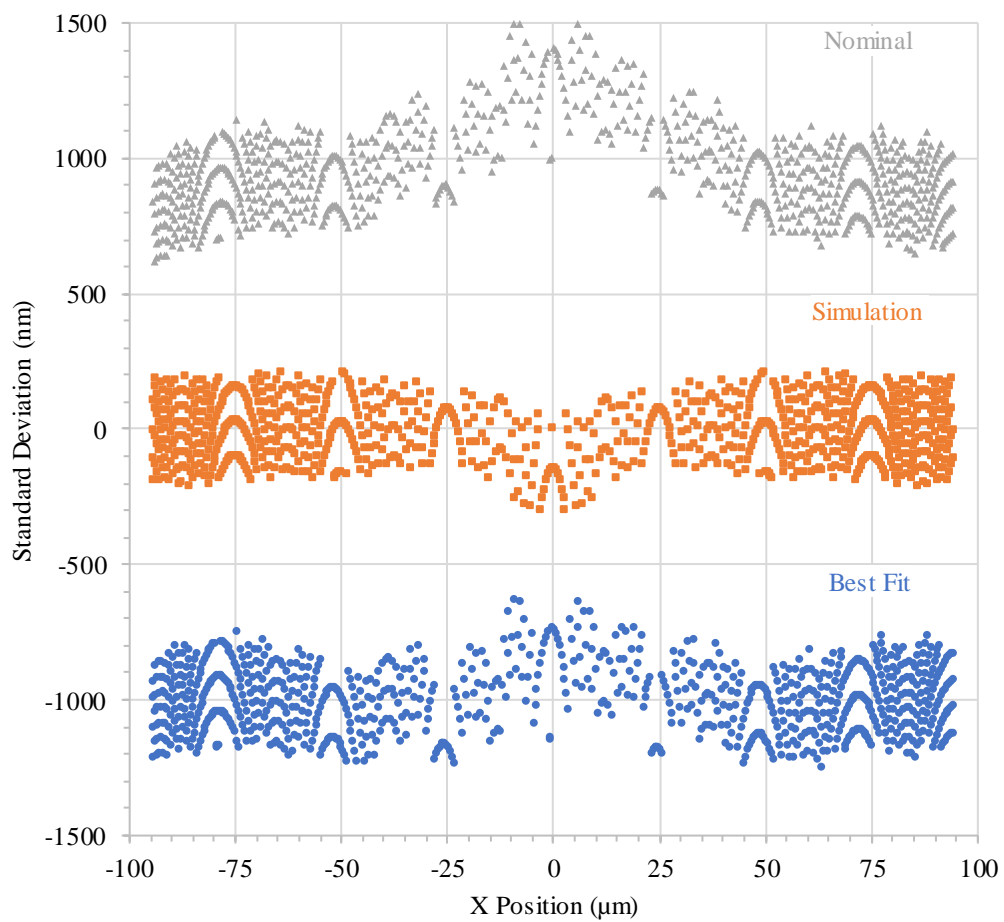


Figure C.5. Plot of the quadratic curve's tangent normal distances.

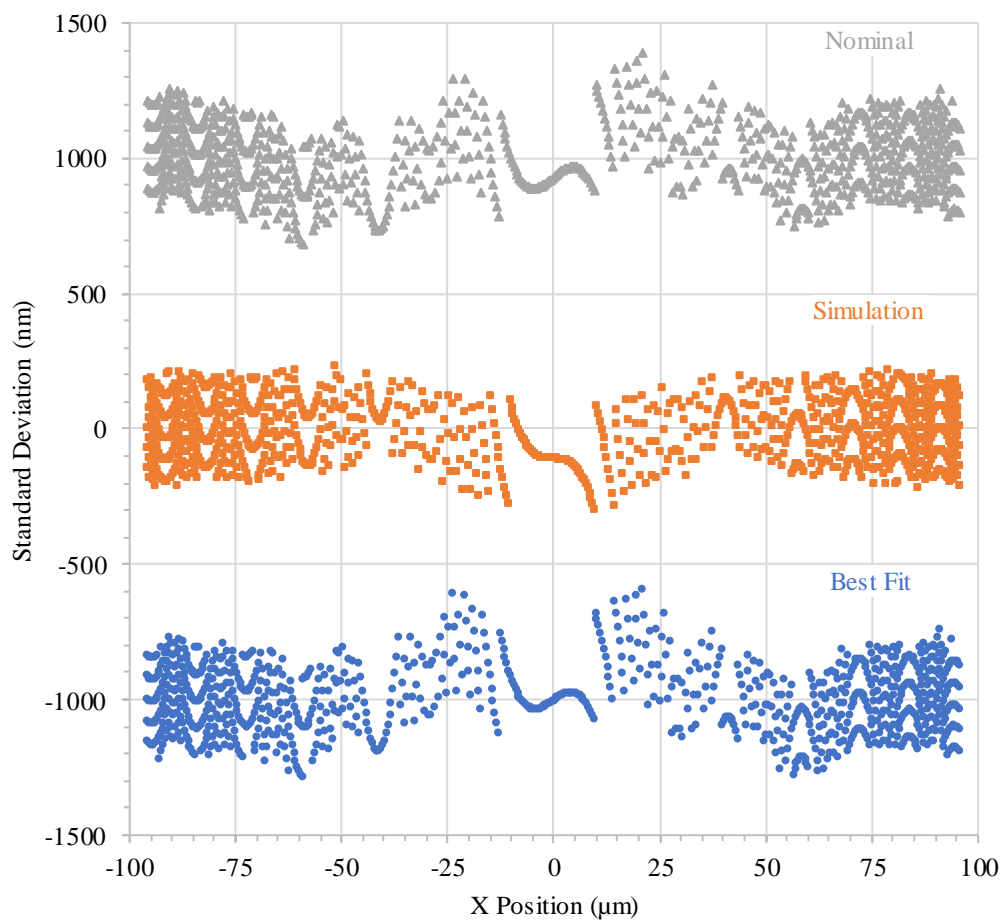


Figure C.6. Plot of the symmetric cubic curve's tangent normal distances.

APPENDIX D: SECONDARY INTERFEROMETER

D.1 Equipment

A Bridgeport Harig 612 grinding machine (Model: N6T34NC18A, see Figure D.) with a Suburban Tool magnetic chuck (Model: MCB-612-FP) was used with two different Norton 6 inch diameter diamond wheels to cut the spare silicon into secondary interferometers. Rough cutting of the silicon to size used a 100 grit, 0.055 inch thick grinding wheel (Specification: M4D100-N75M99-1/8, UPC: 69014192099). For cutting the interferometer blades, a 220 grit, 0.25 inch thick wheel (Specification: SD220-R100B99-1/4, UPC: 69014192764) was employed. To utilize the magnetic chuck of the surface grinder, the silicon parts had to be mounted to 1-2-3 block.



Figure D.1. Bridgeport Harig 612 surface grinder used to cut the silicon pieces into secondary interferometers.

D.2 Mounting

Mounting the silicon piece onto the grinder required a piece of float glass and a 1-2-3 block. Each part was first cleaned with isopropyl alcohol and heated on the hotplate just past the hotplate's 'Low' setting. When the temperature of the 1-2-3 block was sufficiently high, optical wax was melted and spread onto the top surface followed by

placing the float glass onto the waxed steel surface. To ensure the interface was completely coated with wax and that the glass was mounted flat to the block, pressure was applied as the glass was moved around. A second 1-2-3 block was then cleaned and clamped to the first block to serve as an alignment datum for the silicon part. Wax was melted onto the glass surface the cleaned piece of silicon was set onto the waxed glass. After moving the silicon around to once again ensure complete coverage of wax and flat mounting, the flattest side of the silicon piece was pushed against the second 1-2-3 block. Components are shown in Figure D.2 the clamped 1-2-3 blocks are shown in Figure D.3.



Figure D.2. Components shown are (a) 1-2-3 blocks, (b) optical wax, (c) and the hot plate necessary for mounting and securing silicon piece to be machined.

As the parts cooled and wax solidified, creep of the parts relative positions was a threat. To minimize a creeping effect, tilt was added to the jig by placing washers under the bottom 1-2-3 block opposite to the alignment block. Originally the parts were passively cooled by moving the parts off the hotplate and onto a large piece of aluminum to act as a heatsink. Total time to passively cool the setup to room temperature was in excess of 30 min, therefore active cooling with a 120 mm fan powered by the computer power supply was used. With the fan, cooling was reduced to around 15 min.

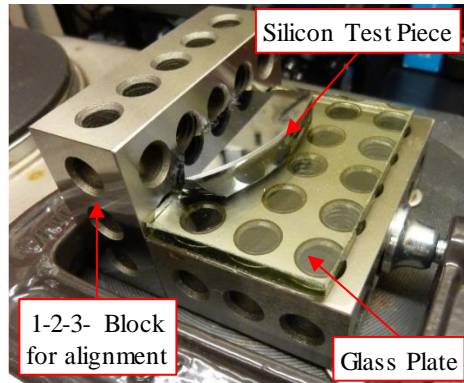


Figure D.3. Using a 1-2-3 as the sample holder and another block as a datum plane for squaring the silicon piece to the side.

D.3 Machining

Before machining was attempted on the actual silicon that would become the secondary interferometers, a spare block of silicon was tested on until a comfortable feed rate was found. Approximately the feed rate calculated was about $2 \text{ mm} \cdot \text{min}^{-1}$ with the spindle running at 3425 RPM. A piece of paper was used as a feeler gage to position the grinding wheel just above the glass surface. Moving the parts out of the way of the grinding wheel, the grinding wheel was further lowered an additional 0.005 inches such that the wheel would cut at most a few thousands of an inch into the glass. Coolant for cooling and debris removal was turned on and monitored throughout the cut to make sure it was in continuous contact with the wheel. To contain the coolant overspray, plastic shielding was constructed around the magnetic chuck as shown in Figure D.4. After checking that the grinding wheel was cutting into the glass, the parts were moved back into position and the cutting of silicon began.

Cuts were made to determine the feasibility of 1 mm blades. The first attempt was unsuccessful after the piece broke part-way through the cut. Pieces were measured with a

digital caliper and it was found that the grinding wheel measured 0.06 inches thick while the cut width was 0.063 inches, about 0.008 inches larger than the manufacturer's specification. Cutting 1 mm thick blades was reevaluated and instead of building the interferometer as a 1-2-1 mm configuration that matched the primary interferometer, the secondary interferometer blades would be cut to 1.5 mm thick. With the 0.063 inch diamond wheel still mounted on the grinder, the first few cuts were made into the main silicon block. Two cuts spaced 44 mm apart were made before the block was rotated 90 degrees and cut into two equal parts. With this configuration, up to four interferometers are capable of being made.

The 0.25 inch thick grinding wheel was installed and a test piece of silicon with dimensions near that of the main block was selected and mounted. As the faces of the interferometer's blades require a high degree of parallelism and equally spacing, a facing cut was made first with into the side of the part around 0.06 inches and 10 mm deep. With this facing cut successfully completed, the dial indicator was zeroed, and the stage was translated 1.5 mm plus 0.25 inches to account for the grinding wheel thickness. This was done two more times to create the three interferometer blades. Halfway through the second cut that would have produced the center blade, the center piece broke off as shown in Figure D.5. Again the blade width was measured with the calipers and discovered to be closer to 0.263 inches. Width of cut was also verified by also measuring the distance between the silicon blades and with gage blocks. The drawings were updated to account for the true width of cut and the depth was reduced from 10 mm to 7.5 mm to improve the strength of the interferometer blades during cutting and reduce the chance of failure.

The main silicon block was remounted into position on the glass with an additional piece of sacrificial silicon added for support. Using the paper feeler gauge method, the grinding wheel was positioned and the first facing cut was made 0.031 inches into the part. Setting the face cut as zero, a second facing cut was made 1.932 inches away. The stage was moved back to its original position and translated 0.322 inches (i.e. 1.5 mm + 0.263 inches). After each cut, the indicator was zeroed and moved another 0.322 inches until all the interferometer blades were machined.

The final cut made splits the interferometer into two equal interferometers. Prior to changing the grinding wheel back to the 0.063 inch thick blade, the mount was removed from the magnetic surface plate. The wheel was aligned to the block and set to cut into the glass before translating it 0.897 inches and make the last cut. Pictures showing the progression of cuts are shown in Figure D.6. Drawings used when manufacturing can be found in subsection 0, with additional images of the interferometer build in subsection D.7.



Figure D.4. Surface grinder with clear plastic shielding to contain the flood coolant.

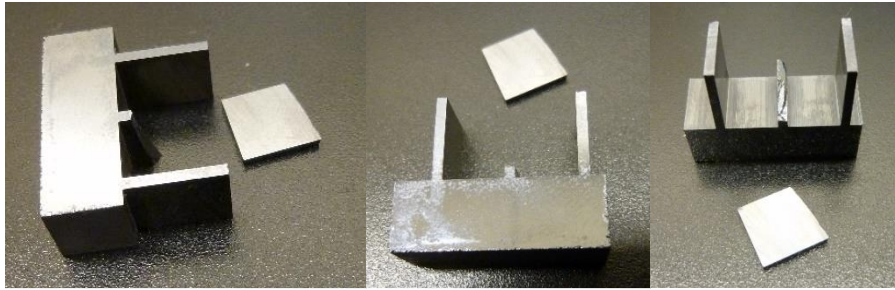


Figure D.5. Silicon test piece cut into the approximate size of the final part. Dimensions of the blades and feed rate are critical to avoid breaking the silicon blades.

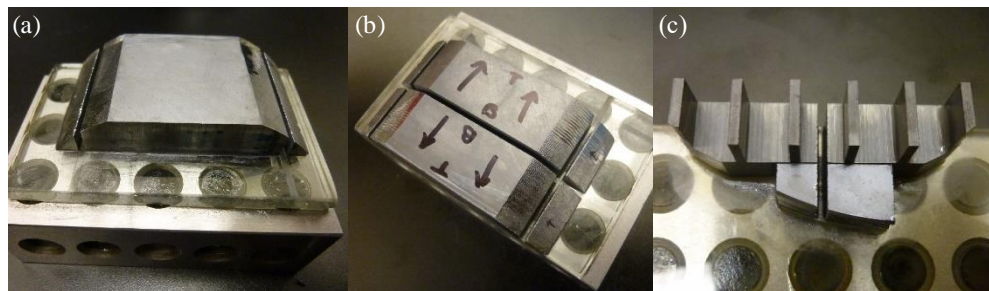


Figure D.6. Progression of cuts through the actual silicon piece to form the interferometers. Two interferometers were produced one block and the second block was preserved for future use. A scrap part was mounted behind the interferometers for support during machining.

D.4 Unmounting and Cleaning

Unmounting and cleaning the silicon starts by heating the hotplate to the wax melting point and removing the parts. While the parts were still hot, they were submerged into a bath of isopropyl alcohol and left for least 3 min. Using a soft, clean rag and isopropyl alcohol, the wax coated surfaces were carefully wiped until clean. To clean the interferometer blades, isopropyl alcohol-soaked cotton swabs were delicately swiped across the blade faces. A new swab was introduced after a few swipes until the swab no longer darkened from fine silicon particles.

D.5 Silicon Etching

Wet etching of silicon was practiced on the silicon pieces made during the test cuts. Using the ternary plot from Schwartz and Robbins [172] to estimate the etch rate based on the etchant composition, solution ratios similar to the primary interferometer build were tried. This etching solution consist of 200 mL nitric acid (HNO_3 , 70%) and 50 mL of hydrofluoric acid (HF, 49%). As HF etches glass, the etchant solution was contained in a polypropylene beaker and the silicon parts were placed in a plastic basket. On a flat test piece of silicon, five strips of electrical tape were attached and placed in the etching solution for 5 min. Following a deionized (DI) water rinse, one strip of tape would be removed and the process repeated until no strips remained. When it was time to etch the secondary interferometer, the etchant solution was altered to 200 mL nitric and 25 mL HF. This change was made to reduce the etching rate with a desired effect being a smoother etched surface finish. Etching time in the solution was 10 min with an estimated etch depth between 50 μm and 80 μm . Finishing off the secondary interferometer build was a DI water rinse and nitrogen blow dry.

D.6 Part Drawings

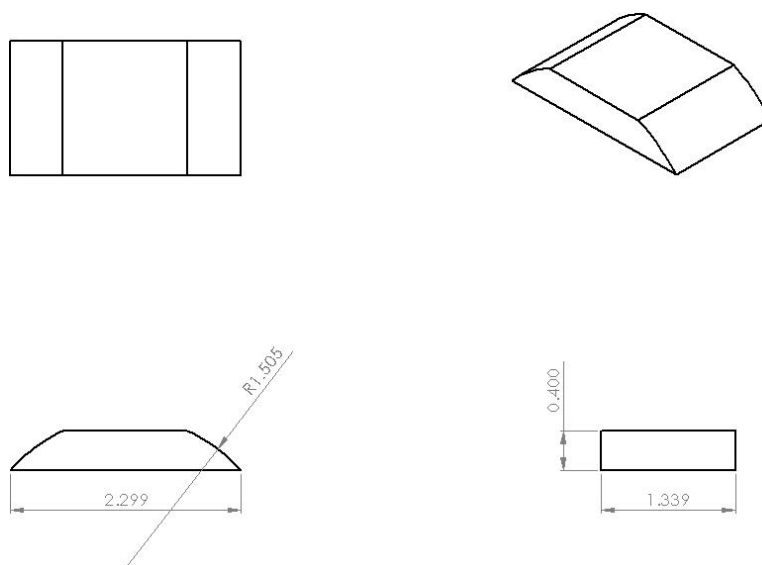


Figure D.7. Dimensions in inches of the silicon block that would be used to create the extra interferometers.

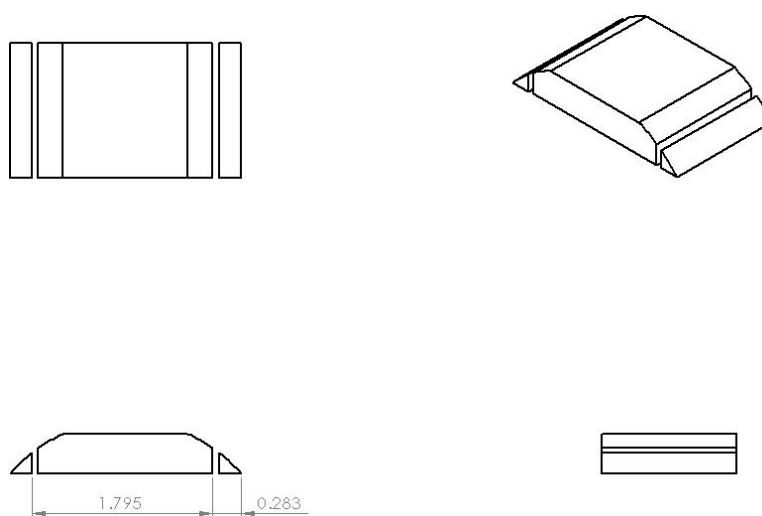


Figure D.8. Locations of the first cuts using a 0.063 inch diamond wheel.

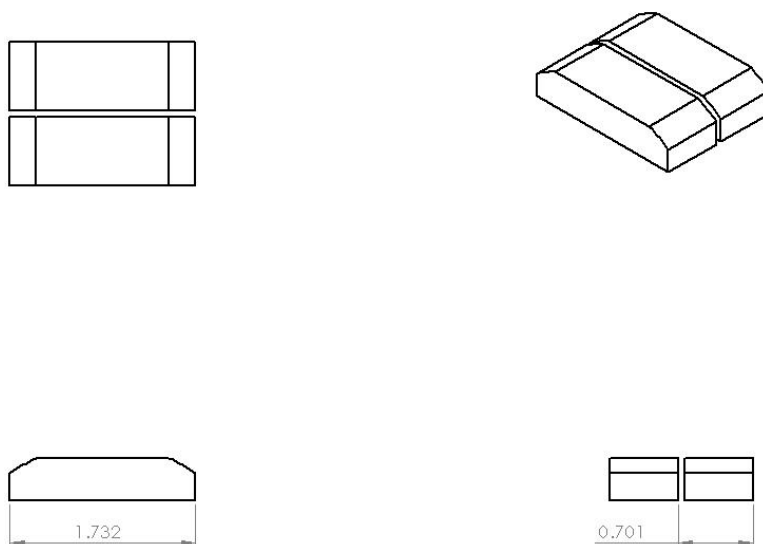
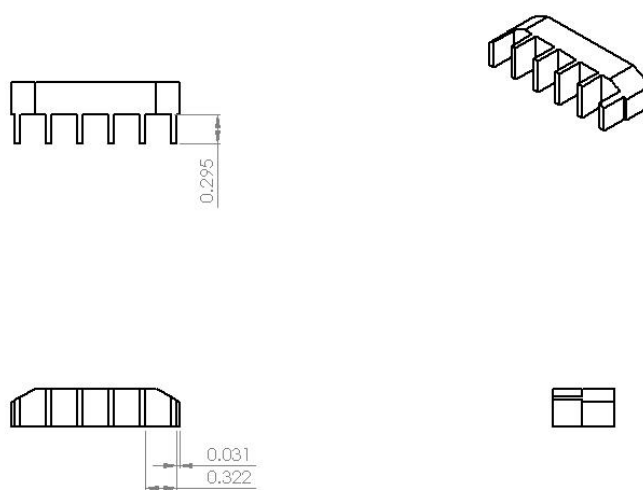


Figure D.9. The second cut splitting the piece into two. The width of each part is about 16.2 mm and the length is 44 mm.



Cut sides first. Move 1.932 inches to cut opposite side.

Figure D.10. Facing the sides and cutting the interferometer blades into the block. Blades are 1.5 mm thick and the diamond wheel used is 0.263 inches thick.

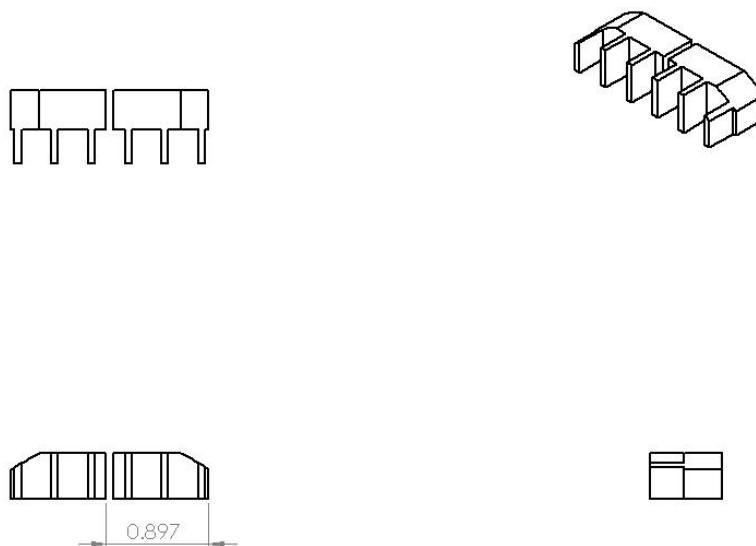


Figure D.11. Cutting the piece once again in half using the 0.063 inch diamond wheel, creating two secondary interferometers.

D.7 Additional Images

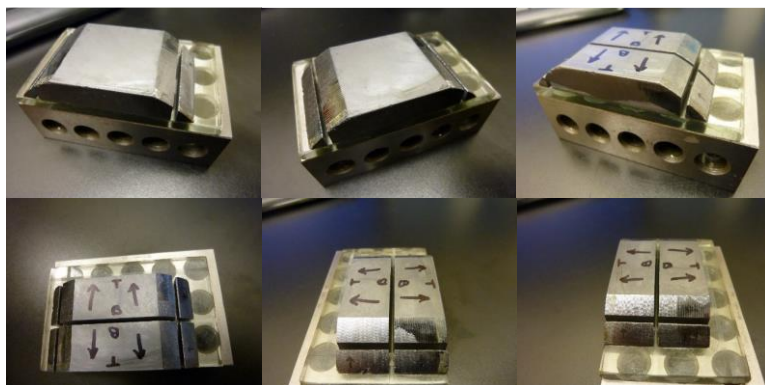


Figure D.12. Additional images showing the single part being split into the four potential interferometers.



Figure D.13. Additional images following the final cuts that produce the blades before the final cut that separates the part into two interferometers.



Figure D.14. Additional images of the part after cutting in half to yield two interferometers.

APPENDIX E: MATLAB® CODES

E.1 Main Measurement Scripts

E.1.1 CSI and Confocal: 'Artefact_Surface_Metrology_v5.m'

```

%% Clean Slate
close all;
clearvars -except PathLast;
addpath functions shapes

%% Adjustable Variables
scale_img = 1; %0.1638/0.4713; % Scale image size
step_Padding = 15; % Padding the edges before measuring step height.
perc_Thres = 0.5; % Percentage between polish and etched surfaces used
for leveling mask cutoff
perc_Length = 0.1; % Percentage of edge length to remove
lvl_Padding = 10; % Padding to add during last leveling process
User_Level = false; % Give user control of first leveling stage.
LEXT_Correction = true; % Subtract the LEXT form error?
Path_Default = '..\..\Data\Artifact Metrology';

%% Open File
% File dialog box.
if (exist('PathLast','var') == 0)
    [FileName,PathName] =
    uigetfile(fullfile(pwd,Path_Default,'*.csv;*.lext;*.dat;*.txt'),'Select
    the data file');
else
    [FileName,PathName] =
    uigetfile(fullfile(PathLast,'*.csv;*.lext;*.dat;*.txt'),'Select the data
    file');
end
PathLast = PathName; % Save last path so that the next run opens in
the same location.

if isequal(FileName,0) || isequal(PathName,0)
    % Display if user selects cancel.
    disp('User pressed cancel');
else
    % Save the image data, resolution, and axis sizes.
    disp('-----')
    disp(['Opening File: ' FileName]);
    [img_Data,res_XY,Instr,img_Intensity] = openfile(PathName,
    FileName);
end

%% LEXT Confocal Correction
if (exist('img_Data','var') && strcmp(Instr,'Confocal'))
    % Load and subtract correction map based on no. of pixels and the
    % resolution of the pixels.
    if (LEXT_Correction == true)

```

```

        if exist('Confocal_Correction.mat','file') == 2
            load('Confocal_Correction.mat')
            correction_name =
strcat(num2str(size(img_Data,1)), 'px_', num2str(res_XY), 'um');

            % Load the proper correction map based in pixel resolution
            % and spatial resolution.
            img_Correction = Confocal_Correction(correction_name);

            % Remove correction map from raw data.
            img_Data = img_Data - img_Correction;

            display('Confocal correction map found and applied...');

            % Clear correction data to free memory.
            clear correction_name Confocal_Correction img_Correction
        else
            display('No confocal correction map on file. Continuing
without correction.')
        end
    end
end

%% Main Code
if (exist('img_Data','var'))
    % Scale Data
    img_Data = imresize(img_Data, scale_img);
    res_XY = res_XY/scale_img

    % Copy data into a new array for leveling
    img_Dataraw = img_Data;

    % Automated Leveling
    % Stage 1 level
    disp('Stage 1 leveling...')
    [img_Data,vector0,center0] = autolevel(img_Data,perc_Thres,0);

    % Stage 2 leveling
    disp('Stage 2 padded leveling...')
    [img_Data,vector1,center1] =
autolevel(img_Data,perc_Thres,lvl_Padding);

    % Final Binary image used for edge/corners
    [peak_low,peak_high,vector2,center2] = histpeaks(img_Data);
    level_Thres = peak_high - perc_Thres*(peak_high - peak_low)/2;
    img_Binary = img_Data*0;
    img_Binary(img_Data>level_Thres) = 1;
    clear peak_low peak_high level_Thres perc_Thres

    cenvec0 = [center0' vector0'];
    cenvec1 = [center1' vector1'];
    cenvec2 = [center2' vector2'];

```

```

img_Angle = img_Dataraw - img_Data;
img_Angle = img_Angle - max(max(img_Angle));
%-----
----

% Object Counting and Borders
disp('Counting objects...');
img_Binary = imcomplement(img_Binary); % Complement
[img_Labeled,num_Objects] = bwlabel(img_Binary,8); % Label each
region

% Remove outliers
if num_Objects > 1
    for i = 1:num_Objects
        if (size(find(img_Labeled==i)) < 1000)
            img_Labeled(img_Labeled==i) = 0;
        end
    end

    [img_Labeled,num_Objects] = bwlabel(img_Labeled,8);
end

%-----
----

% Prompt user for shape
shapes = {'Asym. Cubic','Barbell','Chain Link','Rectangle','Sym.
Cubic','Sym. Quadratic','Triangle'};
[s,v] = listdlg('PromptString','Select the
feature:','SelectionMode','single','ListString',shapes);
if v == 1
    shape = strjoin(shapes(s));
    disp(['User Selected: ' shape])
else
    disp('User Cancelled')
    return % Ends the script here.
end

%-----
----

% Call shape sensitive scripts
switch shape
case 'Asym. Cubic'
    run asymmetric_rules.m
    run shapeorient.m
    run asymmetric_meas3.m
case 'Barbell'
    run barbell_rules.m
    run shapeorient.m
    run barbell_meas.m

```

```

case 'Chain'
    run chain_rules2.m
    run chain_meas3.m
case 'Rectangle'
    run rectangle_rules.m
    run shapeorient.m
    run rectangle_meas.m
case 'Sym. Quadratic'
    run secondorder_rules.m
    run shapeorient.m
    run secondorder_meas3.m
case 'Sym. Cubic'
    run thirdorder_rules.m
    run shapeorient.m
    run thirdorder_meas3.m
case 'Triangle'
    run triangle_rules.m
    %run triangle_meas.m
end

%% Calculate the step height of the feature

run stepmeas.m;

end

```

E.1.2 CT Images: 'Artefact_Image_Metrology.m'

```

%% Open File
% File dialog box.
if (exist('PathName','var') == 0)
    clear all
    [FileName,PathName] = uigetfile({'*.tif;*.tiff;*.png'},'Select an
image file','MultiSelect','on');
else
    clearvars -except PathName
    [FileName,PathName] =
uigetfile(fullfile(PathName,'*.tif;*.tiff;*.png'),'Select the image
file','MultiSelect','on');
end

perc_Length = 0.1; % How much of the total edge length to remove
addpath('shapes','functions')

if isequal(FileName,0)
    disp('User selected Cancel');
    clear file path
else
    disp(['User selected ', fullfile(PathName,FileName)]);

    if iscell(FileName)
        num_imgs = length(FileName);

        ImageFile = strcat(PathName, FileName{1});
        img_temp = imread(ImageFile);

        img_Avg = zeros(size(img_temp,1),size(img_temp,2));

        for i = 1:1:num_imgs
            ImageFile = strcat(PathName, FileName{i});
            img_temp = imread(ImageFile);
            img_Avg(:, :) = img_Avg(:, :) + double(img_temp(:, :, 1));
        end

        img_A = img_Avg/num_imgs;
    else
        img_A = imread(fullfile(PathName,FileName));
        img_A = img_A(:, :, 1);
    end

    figure;imshow(uint8(img_A))

% Get the resolution. Requires user to input scale bar
res_line = img_A(size(img_A,1)-11,:);
res_line(res_line<254)=0;
res_line(res_line==255)=1;
res_loc = find(res_line == 1);
res_loc = res_loc(1:2);

```



```

    res_XY = inputdlg('Value on Scale Bar (mm): ','Scale Bar
Resolution',1,{ '0.55' });

    res_XY = str2double(res_XY{1})*1000; % Converts to number and to
micrometers

    res_XY = (res_XY)/(res_loc(2)-res_loc(1));
    clear res_line res_loc
    close all

    % Find peaks for thresholding
    img_A = double(img_A);
    counts = hist(img_A(:),256);
    [pks,locs] =
findpeaks(counts,'MinPeakDistance',50,'MinPeakHeight',2500);
    if (size(pks) == 1)
        mean_A = round(locs/2);
    else
        mean_A = mean(locs);
    end
    %figure; plot(counts);

    % Use threshold to create binary mask
    img_B = img_A;
    img_B(img_A<=mean_A) = 0;
    img_B(img_A>mean_A) = 1;
    img_B = boolean(img_B);
    %figure;imshow(B);

    img_Binary = imcomplement(img_B); % Complement
    [img_Labeled,num_Objects] = bwlabel(img_Binary,8); % Label each
region

%% % Prompt user for shape
    shapes = {'Rectangle','Second Order','Third
Order','Asymmetric','Triangle','Barbell','Chain'};
    [s,v] = listdlg('PromptString','Select the
feature:','SelectionMode','single','ListString',shapes);
    if v == 1
        shape = strjoin(shapes(s));
        disp(['User Selected: ' shape])
    else
        disp('User Cancelled')
        return % Ends the script here.
    end

    % Call shape sensitive scripts
    switch shape
        case 'Rectangle'
            run rectangle_rules.m
            run shapeorient.m
            run rectangle_meas.m
        case 'Second Order'
            run secondorder_rules.m
            run shapeorient.m
            run secondorder_meas2.m

```

```

        case 'Third Order'
            run thirdorder_rules.m
            run shapeorient.m
            run thirdorder_meas2.m
        case 'Asymmetric'
            run asymmetric_rules.m
            run shapeorient.m
            run asymmetric_meas3.m
        case 'Triangle'
            run triangle_rules.m
            %run triangle_meas.m
        case 'Barbell'
            run barbell_rules.m
            run shapeorient.m
            run barbell_meas.m
        case 'Chain'
            run chain_rules2.m
            run chain_meas3.m
    end

    % Calculate the step height of the feature
    % step_Padding = round(min(shape_size)*.05);
    % run stepmeas.m;
end

```

E.1.3 Create Average Image: 'LoadMultipleImages.m'

```
[Filename, Pathname, Filterindex] = uigetfile('*.tif;*.tiff;*.png',
'Pick a TIFF or PNG image file.','MultiSelect','on');

% Loads multiple images and creates an average image.

if isequal(Filename,0) || isequal(Pathname,0)
    % Cancelled
    disp('User pressed cancel')
else
    % Load the selected image for extracting some data
    % disp(['User selected ', fullfile(Pathname, Filename)])
    %
    % ImageFile = strcat(Pathname, Filename);
    % img = imread(ImageFile);
    %
    % figure;
    % imshow(img);
    num_imgs = length(Filename);

    if num_imgs > 1
        ImageFile = strcat(Pathname, Filename{1});
        img_temp = imread(ImageFile);

        img_Avg = zeros(size(img_temp,1),size(img_temp,2));

        for i = 1:1:num_imgs
            ImageFile = strcat(Pathname, Filename{i});
            img_temp = imread(ImageFile);
            img_Avg(:, :) = img_Avg(:, :) + double(img_temp(:, :, 1));
        end

        for i = 1:1:num_imgs

        end

        figure; imagesc(img_Avg);
        axis tight equal
    end
end

end
```

E.2 Simulation Scripts

E.2.1 'Run_APS.m'

```

clear all;
clc;
%% Polyline Initial Values
% Second
% x_start1 = -200; x_stop1 = 200; k1 = [0,1/200,0,0]; highestpower =
2;
% x_start2 = -100; x_stop2 = 100; k2 = [0,1/50,0,0]; highestpower =
2;
% x_start3 = -50; x_stop3 = 50; k3 = [0,1/12.5,0,0]; highestpower =
2;
% x_start4 = -100; x_stop4 = 100; k4 = [0,1/100,0,0]; highestpower =
2;
% x_start5 = -50; x_stop5 = 50; k5 = [0,1/25,0,0]; highestpower =
2;
% x_start6 = -25; x_stop6 = 25; k6 = [0,1/6.25,0,0]; highestpower =
2;

% Third
% x_start1 = -200; x_stop1 = 200; k1 = [1/40000,0,0,0]; highestpower =
3;
% x_start2 = -100; x_stop2 = 100; k2 = [1/5000,0,0,0]; highestpower =
3;
% x_start3 = -50; x_stop3 = 50; k3 = [1/625,0,0,0]; highestpower =
3;
% x_start4 = -100; x_stop4 = 100; k4 = [1/10000,0,0,0]; highestpower =
3;
% x_start5 = -50; x_stop5 = 50; k5 = [1/1250,0,0,0]; highestpower =
3;
% x_start6 = -25; x_stop6 = 25; k6 = [1/156.25,0,0,0]; highestpower =
3;

% Asymetric
% x_start1 = -200; x_stop1 = 100; k1 = [1/10000,1/50,0,0];
highestpower = 3;
% x_start2 = -100; x_stop2 = 50; k2 = [1/1250,1/12.5,0,0];
highestpower = 3;
% x_start3 = -50; x_stop3 = 25; k3 = [1/156.25,1/3.125,0,0];
highestpower = 3;
% x_start4 = -100; x_stop4 = 50; k4 = [1/2500,1/25,0,0];
highestpower = 3;
% x_start5 = -50; x_stop5 = 25; k5 = [1/312.5,1/6.25,0,0];
highestpower = 3;
% x_start6 = -25; x_stop6 = 12.5; k6 = [1/39.0625,1/1.5625,0,0];
highestpower = 3;

%% Initial values
theta = 0:1:45; % angle to rotate in degree
res = 0.407; % resolution in um/pixel

```

```

dx = 0.005*res;           % increment to step in um

noise_max = 0;           % range of noise in um (mask accuracy is 0.5 um)
lambdac = 50;            % cutoff in um

DelInnerPts = 1; % Delete points considered inside the figure?
showplots = 0; % Show plots on finish?

m = length(theta);
for i = 1:length(theta)
    [kbf1(i,:),rms_dist1(i),mean_dist1(i),std_dist1(i),num_pts1(i)] =
    ArtefactPolySim(x_start1,x_stop1,dx,k1,highestpower,theta(i),res,noise_
    max,lambdac,DelInnerPts,showplots);
    [kbf2(i,:),rms_dist2(i),mean_dist2(i),std_dist2(i),num_pts2(i)] =
    ArtefactPolySim(x_start2,x_stop2,dx,k2,highestpower,theta(i),res,noise_
    max,lambdac,DelInnerPts,showplots);
    [kbf3(i,:),rms_dist3(i),mean_dist3(i),std_dist3(i),num_pts3(i)] =
    ArtefactPolySim(x_start3,x_stop3,dx,k3,highestpower,theta(i),res,noise_
    max,lambdac,DelInnerPts,showplots);
    [kbf4(i,:),rms_dist4(i),mean_dist4(i),std_dist4(i),num_pts4(i)] =
    ArtefactPolySim(x_start4,x_stop4,dx,k4,highestpower,theta(i),res,noise_
    max,lambdac,DelInnerPts,showplots);
    [kbf5(i,:),rms_dist5(i),mean_dist5(i),std_dist5(i),num_pts5(i)] =
    ArtefactPolySim(x_start5,x_stop5,dx,k5,highestpower,theta(i),res,noise_
    max,lambdac,DelInnerPts,showplots);
    [kbf6(i,:),rms_dist6(i),mean_dist6(i),std_dist6(i),num_pts6(i)] =
    ArtefactPolySim(x_start6,x_stop6,dx,k6,highestpower,theta(i),res,noise_
    max,lambdac,DelInnerPts,showplots);
    m=i
end

% theta = theta';
% rms_dist1 = rms_dist1';
% mean_dist1 = mean_dist1';
% std_dist1 = std_dist1';

output1 = [theta',kbf1,rms_dist1',mean_dist1',std_dist1',num_pts1'];
%output2 = [theta',kbf2,rms_dist2',mean_dist2',std_dist2',num_pts2'];
%output3 = [theta',kbf3,rms_dist3',mean_dist3',std_dist3',num_pts3'];
%output4 = [theta',kbf4,rms_dist4',mean_dist4',std_dist4',num_pts4'];
%output5 = [theta',kbf5,rms_dist5',mean_dist5',std_dist5',num_pts5'];
%output6 = [theta',kbf6,rms_dist6',mean_dist6',std_dist6',num_pts6'];

%figure(1);plot(theta,rms_dist,'.b')
%figure(2);plot(theta,std_dist,'.r')

```

E.2.2 Circle Simulation: 'ArtefactCircleSim.m'

```
% Circle 2 (Multi-run random XY)
clear all; close all; clc;
r0 = 200; % Radius of the circle
res = 0.1; % Resolution of the device

for k = 1:1000
    psize = ceil(r0/res);

    xc0(k) = (rand()-0.5)*res;
    yc0(k) = (rand()-0.5)*res;

    % Create the circle within the grid
    [xx,yy] = meshgrid(-(psize+10):(psize+10));
    C = sqrt((xx+xc0(k)).^2+(yy+yc0(k)).^2)<=psize;
    %figure;imshow(C)

    % Find the circle's border
    mtx_Border = bwboundaries(C,8);
    xp = mtx_Border{1}(:,2);
    yp = mtx_Border{1}(:,1);
    % figure;plot(xp,yp,'*b');
    % figure;plot(xp*res,yp*res,'*r');

    % Count number of points in the border
    xlen(k) = length(xp);

    % Fit a circle to the border
    [xc1(k),yc1(k),r1(k),~] = circfit(xp,yp);

    r1(k)=r1(k)*res;

    % Statistics
    dist = sqrt((xp-xc1(k)).^2+(yp-yc1(k)).^2)*res;
    dist_dev_r0 = dist - r0;
    dist_dev_r1 = dist - r1(k);
    dist_rms_r0(k) = rms(dist_dev_r0);
    dist_rms_r1(k) = rms(dist_dev_r1);

    close all; % Close any figures that were opened
    1000-k % Countdown for remaining calculations
end

figure; plot(res,r1); title('BF Radius')
figure; plot(res,xlen),title('Pixel Count')
figure; plot(res,r1./xlen),title('R1:Pixel Ratio')
figure; plot(res,dist_rms_r0); title('RMS r0 Distance')
figure; plot(res,dist_rms_r1); title('RMS r1 Distance')
figure; plot(res,dist_rms_r0./dist_rms_r1),title('RMS Ratio')
res=res'; r1=r1'; xlen=xlen'; dist_rms_r1 = dist_rms_r1';
xc0=xc0'; yc0=yc0'; xc1=xc1'; yc1=yc1';

%% Circle 2 (Multi-run resolution)
% clear all; close all; clc;
% r0 = 200; % Radius of the circle
```

```

% res = (0.05:0.001:1.0); Resolution of the device
%
% for k = 1:length(res)
%     psize = ceil(r0/res(k));
%
%     xc0 = 0;
%     yc0 = 0;
%
%     % Create the circle within the grid
%     [xx yy] = meshgrid(-(psize+1):(psize+1));
%     C = sqrt((xx).^2+(yy).^2)<=psize;
%     %figure;imshow(C)
%
%     % Find the circle's border
%     mtx_Border = bwboundaries(C,8);
%     xp = mtx_Border{1}(:,2);
%     yp = mtx_Border{1}(:,1);
%     % figure;plot(xp,yp,'*b');
%     % figure;plot(xp*res,yp*res,'*r');
%
%     % Count number of points in the border
%     xlen(k) = length(xp);
%
%     % Fit a circle to the border
%     [xc1,yc1,r1(k),~] = circfit(xp,yp);
%
%     r1(k)=r1(k)*res(k);
%
%     % Statistics
%     dist = sqrt((xp-xc1).^2+(yp-yc1).^2)*res(k);
%     dist_dev_r0 = dist - r0;
%     dist_dev_r1 = dist - r1(k);
%     dist_rms_r0(k) = rms(dist_dev_r0);
%     dist_rms_r1(k) = rms(dist_dev_r1);
%
%     close all;
%     length(res)-k
% end
%
% figure; plot(res,r1); title('BF Radius')
% figure; plot(res,xlen),title('Pixel Count')
% figure; plot(res,r1./xlen),title('R1:Pixel Ratio')
% figure; plot(res,dist_rms_r0); title('RMS r0 Distance')
% figure; plot(res,dist_rms_r1); title('RMS r1 Distance')
% figure; plot(res,dist_rms_r0./dist_rms_r1),title('RMS Ratio')
% res=res'; r1=r1'; xlen=xlen'; dist_rms_r1 = dist_rms_r1';

%% Circle 2 (Single run)
% clear all; %clc;
% r0 = 120;
% res = 0.5;
%
% psize = ceil(r0/res);
%
% xc0 = 0;

```

```

% yc0 = 0;
%
% % Create the circle within the grid
% [xx,yy] = meshgrid(-(psize+1):(psize+1));
% C = sqrt((xx).^2+(yy).^2)<=psize;
% figure;imshow(C)
%
% % Find the circle's border
% mtx_Border = bwboundaries(C,8);
% xp = mtx_Border{1}(:,2);
% yp = mtx_Border{1}(:,1);
% % figure;plot(xp,yp,'*b');
% % figure;plot(xp*res,yp*res,'*r');
%
% % Fit a circle to the border
% [xc1,yc1,r1,~] = circfit(xp,yp);
%
% r1=r1*res
%
% % Statistics
% dist = sqrt((xp-xc1).^2+(yp-yc1).^2)*res;
% dist_dev_r0 = dist - r0;
% dist_dev_r1 = dist - r1;
% dist_rms_r0 = rms(dist_dev_r0)
% dist_rms_r1 = rms(dist_dev_r1)

%close all;

%% Circle 1 (Generate using angle)
% clear all; %clc;
%
% r = 200;
% ts = r*(1E-6);
% res = 0.1;
%
% dist_dev_rms = zeros(length(ts),1);
% xy_count = dist_dev_rms;
%
% t = 0:ts:2*pi;
%
% x = r*cos(t);
% y = r*sin(t);
%
% for k = 1:1
%     % Print k in increments of 10
%     if mod(k,10) == 0
%         k
%     end
%
%     % Add random position within a pixel
%     x_rand = 0*(rand()-0.5)*res;
%     y_rand = 0*(rand()-0.5)*res;
%
%     x = x+x_rand;
%     y = y+y_rand;
%
%     % Convert to pixels

```



```

%      xp = round(x/res);
%      yp = round(y/res);
%
%      % Remove duplicates
%      j = 1;
%      while (j < length(xp))
%          % Append only if previous is not equal to current values
%          if (isequal([xp(j) yp(j)], [xp(j+1) yp(j+1)]) == 1)
%              xp(j+1) = [];
%              yp(j+1) = [];
%          else
%              j = j+1;
%          end
%      end
%
%
%      xy_count(k) = length(xp);
%      [xc,yc,R,~] = circfit(xp,yp);
%      R = R*res;
%
%      dist = sqrt((xp-xc).^2+(yp-yc).^2)*res;
%      dist_dev_r = dist - r;
%      dist_dev_R = dist - R;
%      dist_rms_r(k) = rms(dist_dev_r);
%      dist_rms_R(k) = rms(dist_dev_R);
%      %dist_dev_rms2 = rms(dist_dev(1:round(length(dist_dev)/4)))
%
%      %figure; plot(x,y,'b',xp,yp,'*r');
%      %figure; plot(dist_dev);
%  end
%  figure;plot(xy_count);title('Count');
%  figure;plot(dist_rms_r);title('True R');
%  figure;plot(dist_rms_R);title('Best Fit R');

```

E.3 Functions

E.3.1 'autolevel.m'

```
function [data_Out,vector,center] = autolevel(data_In, perc_Thres,
num_Padding)
    % Find threshold locations
    [peak_low,peak_high,vector,center] = histpeaks(data_In);
    thres = peak_high - (peak_high - peak_low)*perc_Thres;

    % Create stage 1 binary mask, values above threshold are 1
    img_Binary = data_In*0;
    img_Binary(data_In>thres) = 1;
    img_Binary2 = img_Binary;
    % Create a padded region around the etched regions to remove
    potential noise regions
    if (num_Padding > 0)
        disp('-Padding regions...')
        img_Binary = padarray(img_Binary,[num_Padding num_Padding],1);
        [row,col] = find(img_Binary==0);

        % Steps through all locations with 0 and makes the surrounding
        % locations 0 up to the 'padding' value.
        for i = 1:length(row)
            img_Binary(row(i)-num_Padding:row(i)+num_Padding,col(i)-
num_Padding:col(i)+num_Padding)=0;
        end

        % Crop the padded image back to the original size.
        img_Binary = img_Binary(num_Padding+1:end-
num_Padding,num_Padding+1:end-num_Padding);
        figure; imagesc(img_Binary+img_Binary2); axis tight equal
    end

    data_Out = planeFit(data_In,img_Binary);
end
```

E.3.2 'ArtefactPolySim.m'

```

function [kbf,rms_dist,mean_dist,std_dist,num_pts] =
ArtefactPolySim(x_start,x_stop,dx,k,highestpower,theta,res,noise_max,la
mbdac,DelInnerPts,showplots)
    %% Theoretical line
    x = (x_start:dx:x_stop);
    y = k(4) + k(3)*x + k(2)*(x.^2) + k(1)*(x.^3); % Create initial y
data

    %% Noise profile with rotation
    if (noise_max ~= 0)
        noise = rand(1,floor(length(x)+4*lambdac/dx)); % create noise
array
        noise = noise*2-1; % Scale to range(-1,+1)
        noise_filtered = GaussFilter(dx,noise,lambdac); % Create
profile from noise

        tnum = floor((length(noise)-length(x))/2); % Truncate amount
        noise = noise(tnum:end-tnum-1); % Truncate array to size

        tnum = floor((length(noise_filtered)-length(x))/2); % Truncate
amount
        noise_filtered = noise_filtered(tnum:end-tnum-1); % Truncate
array to size

        noise_filtered = noise_filtered/max(abs(noise_filtered)); %
Normalize to 1
        noise_filtered = noise_filtered*max(abs(noise))*noise_max; %
Rescale to no more than noise_max

        % Add noise/rotation
        dydx = k(4) + 2*k(3)*x + 3*k(2)*(x.^2); % Take derivative for
slope
        dydx = -1./dydx; % Change to normal
        slope = atan(dydx); % Find angle

        % Convert slope to dx and dy constants
        dx = cos(slope);
        dy = sin(slope);

        % When slope goes negative, make dx and dy also negative to
remove
        % discontinuity (stepping)
        for i = 1:length(dx)
            if (slope(i) < 0)
                dx(i) = -dx(i);
                dy(i) = -dy(i);
            end
        end

        % Add noise to data
        xn = x + noise_filtered.*dx;
        yn = y + noise_filtered.*dy;

```

```

else
    % No
    xn = x;
    yn = y;
end

% Rotate Data
theta = theta*pi/180; cost = cos(theta); sint = sin(theta);

xr = xn*cost - yn*sint;
yr = yn*cost + xn*sint;

%% Pixelate Data
% Create and populate pixel array with unique pairs
xrp = round(xr(1)/res);
yrp = round(yr(1)/res);

%inc = 1;
for i = 2:length(xr)
    xp = round(xr(i)/res); % x pixel value
    yp = round(yr(i)/res); % y pixel value

    % Append only if previous is not equal to current values
    if (isequal([xp yp],[xrp(end) yrp(end)]) == 0)
        %inc = inc + 1;
        xrp = [xrp xp];
        yrp = [yrp yp];
    end
end

% Remove inner points
if DelInnerPts == true
    [xrp, yrp] = DeleteInnerPts(xrp,yrp);
end

num_pts = length(xrp);

%% Best fitting
% Convert pixels to um
xrf = xrp*res;
yrf = yrp*res;

% Calculate rotation removal constants
cost = cos(-1*theta);
sint = sin(-1*theta);

% Remove rotation from the data
xf = xrf*cost - yrf*sint;
yf = yrf*cost + xrf*sint;

% Calculate the fit parameters for the theoretical and the noisy
data

```

```

%tpoly = polyfit(x, y, highestorder);    % theoretical data
npoly = polyfit(xf, yf, highestpower);    % noisy data

% Expand kbf to 4 elements if needed (output order is [x^3 x^2 x^1
x^0])
kbf = npoly;
while length(kbf) < 4
    kbf = [0 kbf];
end

% Calculate best fit line data points
ybf = kbf(4) + kbf(3)*xf + kbf(2)*(xf.^2) + kbf(1)*(xf.^3);

%
distances = NormDist(xf,yf,kbf);
%
rms_dist = rms(distances);
%
mean_dist = mean(distances);
%
std_dist = std(distances);

% Get the final normal vector deviation
[norm_dist0, norm_x0, norm_y0] = NormDist(xf,yf,kbf);

% Change NaN to 0;
norm_dist0(isnan(norm_dist0)) = 0;
norm_x0(isnan(norm_x0)) = 0;
norm_y0(isnan(norm_y0)) = 0;

% Statistics
rms_dist = rms(NormDist(xf,yf,kbf));
mean_dist = mean(norm_dist0);
std_dist = std(norm_dist0);

%
disp(['Coeff of Best Fit: ',num2str(npoly)])
%
disp(['RMS of Best Fit: ',num2str(rms_dist)])

if showplots == true
    if (noise_max ~= 0)
        figure(1);
        hold on;
        plot(x,noise,'b');
        plot(x,noise_filtered,'r');
        hold off;
    end

    figure(2);
    hold on;
    plot(x,y,'b');
    plot(xn,yn,'r');
    hold off;

    figure(3)
    hold on
    plot(xf,yf,'.b')

```

```
plot(xf,ybf,'r')
hold off

figure(4)
hold on
plot(xf,distances,'.b')
hold off

figure(5)
hist(distances,10)
end
end
```

E.3.3 'circfit.m'

```

function [xc,yc,R,a] = circfit(x,y)
%CIRCFIT  Fits a circle in x,y plane
%
% [XC, YC, R, A] = CIRCFIT(X,Y)
% Result is center point (yc,xc) and radius R.  A is an optional
% output describing the circle's equation:
%
%    $x^2+y^2+a(1)*x+a(2)*y+a(3)=0$ 
%
% by Bucher izhak 25/oct/1991

n=length(x);  xx=x.*x;  yy=y.*y;  xy=x.*y;
A=[sum(x) sum(y) n;sum(xy) sum(yy) sum(y);sum(xx) sum(xy) sum(x)];
B=[-sum(xx+yy) ; -sum(xx.*y+yy.*y) ; -sum(xx.*x+xy.*y)];
a=A\B;
xc = -.5*a(1);
yc = -.5*a(2);
R  = sqrt((a(1)^2+a(2)^2)/4-a(3));
end

```

E.3.4 'DeleteInnerPts.m'

```

%% Delete Inner Points Function
function [xt,yt] = DeleteInnerPts(xt,yt)
    i = 2;

    while i < length(xt)
        dx = xt(i-1)-xt(i+1);
        dy = yt(i-1)-yt(i+1);

        d = sqrt(dx^2 + dy^2);

        % Check if the points are adjacent
        if (round(d*10000)/10000 == 1.4142)
            % If true, calculate slope, y-intercepts, and vector
            m = (yt(i+1)-yt(i-1))/(xt(i+1)-xt(i-1));
            b1 = yt(i-1) - m*xt(i-1);
            b2 = yt(i) - m*xt(i);
            x_vector = xt(i+1) - xt(i-1);

            % Determine if 'inner' point by y-intercept and x vector
            direction
            if ((b1 > b2) && (x_vector >= 0)) % Current point is inner
            point
                xt(i) = []; yt(i) = [];
            elseif ((b1 < b2) && (x_vector < 0)) % Current point is
            inner point
                xt(i) = []; yt(i) = [];
            else
                % If false, increment i to check next point
                i = i+1;
            end

            else
                % If false, increment i to check next point
                i = i + 1;
            end
        end
    end
end

```


E.3.5 'distmeas.m'

```
function [dist] = distmeas(line1,line2)

% Switches the columns if column 1 STD is smaller than column 2. This
% should prevent a vertical line which might cause an error in the
% calculation.
if std(line1(:,1)) < std(line1(:,2))
    line1 = fliplr(line1);
    line2 = fliplr(line2);
end

%DISTMEAS Summary of this function goes here
% Detailed explanation goes here
P1 = polyfit(line1(:,1),line1(:,2),1);
P2 = polyfit(line2(:,1),line2(:,2),1);

% Calculate the normal line
x_int1 = mean(line1(:,1));
y_int1 = P1(1)*x_int1 + P1(2);

% If slope is less than 1e-3 (effectively calling this zero), use
x_int1 as x_int2
if abs(P1(1)) < 1e-3
    x_int2 = x_int1;
else
    slope = -1/P1(1);
    intercept = P1(2) - slope*x_int1;

    x_int2 = (P2(2)-intercept)/(slope-P2(1));
end

y_int2 = P2(1)*(x_int2)+P2(2);

dist = sqrt((x_int2-x_int1)^2+(y_int2-y_int1)^2);
end
```

E.3.6 'GaussFilter.m'

```
function output = GaussFilter(dx,y,lambdac)
    x = (-lambdac:dx:lambdac);

    alpha = 0.4697;

    S = (1/(alpha*lambdac))*exp(-pi*(x/(alpha*lambdac)).^2); % generate
    Gaussian filter
    S = S/sum(S);
    output = conv(y,S);
end
```

E.3.7 'histpeaks.m'

```
function [peak_low,peak_high,vector,center] = histpeaks(Data)
    % Determine the bin range
    bin_min = floor(min(Data(:)));
    bin_max = ceil(max(Data(:)));

    %
    [vector,center] = hist(Data(:),bin_min:0.1:bin_max);
    % figure; barh(bin_min:0.1:bin_max,vector);
    % ylabel('Height (um)'); xlabel('Number of Pixels');

    % Find the two peaks separated by at least 2 um.
    loc = 0;
    i = 0;

    % When using 'XY Adjust', 'Cartesian', and 'Overlay' CSI stitched
    data,
    % comment out next set of logic and use this one.
    while (length(loc) ~= 2)
        i = i+1;
        [~,loc] =
findpeaks(vector,'MinPeakHeight',1000,'MinPeakDistance',i);
    end

    % When using 'Adaptive Adjust' CSI stitched data, comment out
    previous
    % logic and use this one.
    % while (length(loc) ~= 2)
    %     i = i+1;
    %     [~,loc] = findpeaks([0 vector
    0],'MinPeakHeight',1000,'MinPeakDistance',i);
    %     end

    loc = loc - 1;

    peak_low = center(loc(1));
    peak_high = center(loc(2));
end
```

E.3.8 'maxdist.m'

```

function [ind,dist,ang] = maxdist(mtx)
%MAXDIST This function will find the points with the largest distance
%between them and return the indices, the distance, and the
%angle.

    x = mtx(:,1);
    y = mtx(:,2);

    len = length(x);
    ind = [0,0];
    dist = 0;

    for m = 1:1:len-1
        for n = m+1:1:len
            tempdist = sqrt((x(n)-x(m))^2+(y(n)-y(m))^2);

            if tempdist > dist
                dist = tempdist;
                ind(1) = m;
                ind(2) = n;
                ang = atan((y(n)-y(m))/(x(n)-x(m)));
            end
        end
    end
end
end
end

```

E.3.9 'normdist.m'

```

function [dev_out, x_out, y_out] = normdist(x,y,k)
    % Create arrays for distances from tangent
    dev_out = zeros(1,length(x));
    x_out = dev_out; y_out = dev_out;

    % Create coefficient array and find the roots of the polynomial set
    to zero.
    % Find  $f(x_1) = y_1'(y_1 - y_2) + x_1 - x_2 = 0$ 
    %  $y_1 = a + b*(x_1) + c*(x_1)^2 + d*(x_1)^3$ 
    %  $y_1'$  is the derivative of  $y_1$ 
    %  $a = k(4) \dots d = k(1)$ 
    coeff = [0 0 0 0 0 0];

    for i = 1:length(x)
        x2 = x(i);
        y2 = y(i);

        coeff(1) = 3*k(1)^2; %  $x^5$ 
        coeff(2) = 5*k(2)*k(1); %  $x^4$ 
        coeff(3) = 4*k(3)*k(1) + 2*k(2)^2; %  $x^3$ 
        coeff(4) = 3*k(4)*k(1) + 3*k(3)*k(2) - 3*k(1)*y2; %  $x^2$ 
        coeff(5) = 2*k(4)*k(2) + k(3)^2 + 1 - 2*k(2)*y2; %  $x^1$ 
        coeff(6) = k(4)*k(3) - k(3)*y2 - x2; %  $x^0$ 

        % The root with the lowest imaginary number is the x position
        on the theoretical line
        c_roots = roots(coeff); % calculate the roots

        % if length(c_roots) > length(imag(c_roots))
        % % Delete elements with imaginary numbers
        c_roots(imag(c_roots)~=0)=[];
        % end

        % Use the root closest to x2
        [~,ind] = min(abs(x2-real(c_roots)));
        x1 = c_roots(ind);

        % Calculate  $y_1(x_1)$ 
        y1 = k(4) + k(3)*x1 + k(2)*(x1^2) + k(1)*(x1^3);

        % Find sign, into shape is negative
        y0 = k(4) + k(3)*x2 + k(2)*(x2^2) + k(1)*(x2^3);

        dev_sign = (y1-y0)/abs(y1-y0);

        dev_out(i) = dev_sign*sqrt((x2-x1)^2+(y2-y1)^2); % calculate
        distance between (x2,y2) and (x1,y1)
        x_out(i) = x1;
        y_out(i) = y1;
    end
end

```

E.3.10 'openfile.m'

```

function [img_Height,res_XY,Instr,img_Intensity] =
openfile(PathName,FileName)
% img_Height is the height map data in micrometers
% res_XY is the instruments spatial resolution in micrometers
% img_Intensity is the intensity map from the Confocal

fileOpen = [PathName FileName];

if strfind(FileName,'.lxt') > 0
    %% Opens Olympus Confocal (.lxt) files
    % WARNING: THE XY RESOLUTION IS NOT EXACT. I could not find
    corrections for the values saved within the file.
    Instr = 'Confocal';

    % Opens up the info and images of the .LXT files, parses through
    the
    % into for the resolution values, and creates the axes in
    micrometers.

    % Reads the file structure
    Info = imfinfo(fileOpen);

    % Finds the Z resolution within the Info structure
    Info_Zstart =
strfind(Info(1).ImageDescription,'<HeightDataPerPixelZ>')+length('<Heig
htDataPerPixelZ>');
    Info_Zend =
strfind(Info(1).ImageDescription,'</HeightDataPerPixelZ>')-1;

    % Saves each img to a matrix.
    img_Intensity = double(imread(fileOpen, 1, 'Info', Info));
    img_Height = double(imread(fileOpen, 3, 'Info', Info));

    % Resolution of the lateral and depth. Values have to be adjusted
    to
    % micrometers.
    res_XY = double(Info(3).XResolution * 1e-6);
    res_Z =
str2double(Info(1).ImageDescription(Info_Zstart:Info_Zend))*1e-6;

    % Pixel size of the img.
    size_Width = double(Info(3).Width);
    size_Height = double(Info(3).Height);

    img_Height = double(img_Height) * res_Z;

    disp('Olympus LEXT Confocal Data Loaded');

elseif strfind(FileName,'.dat') > 0
    %% Opens Zygo/Metro Pro (.dat) files
    Instr = 'SWLI';
    % Reads the file
    [~, InfoHeader,img_Intensity,img_Height] =
ReadMetroProFile(fileOpen);

```

```

img_Intensity = NaN; % Doesn't exist

% Creates resolution and converts image height
res_XY = InfoHeader.CameraRes*1E6; % Converts resolution to um
img_Height = img_Height.*1E6; % Converts img_Height to um

size_Width = double(InfoHeader.PhaseWidth);
size_Height = double(InfoHeader.PhaseHeight);

% Remove NaN border
% img_Height = img_Height(2:size_Height-1,2:size_Width-1);

% Pixel size of the img.
% size_Width = size_Width - 2;
% size_Height = size_Height - 2;
disp('Zygo/Metro Pro SWLI Data Loaded');

elseif strfind(FileName,'txt') > 0
    %% Opens AFM (.txt) files
    Instr = 'AFM';

    % Grabs header information and places in an array
    header_Info =
    textread(fileOpen,'%s',357,'delimiter','\n','headerlines',1);

    img_Intensity = NaN; % Doesn't exist

    str = char(header_Info(335));
    loc = strfind(str,' ');

    % Pixel size of the img.
    size_Width = str2num(str(loc(3)+1:size(str,2)));
    size_Height = size_Width;

    % Cuts the AFM data from the file
    img_Height = textread(fileOpen,'%f','headerlines',359);
    img_Height = reshape(img_Height,size_Height,size_Width);

    str = char(header_Info(337));
    loc = strfind(str,' ');
    scan_size = [0 0]';

    for i=1:2
        scan_size(i) = str2num(str(loc(i+1)+1:loc(i+2)));
    end
    clear loc str i

    res_XY = scan_size(1)/size_Width;
    disp('AFM Data Loaded');
elseif strfind(FileName,'csv') > 0
    Instr = 'Unknown';

    img_Intensity = NaN; % Doesn't exist

    res_XY = csvread(fileOpen,1,1,[1 1 1 1])/1000; % Loads resolution
    img_Height = csvread(fileOpen,6,0); % Loads height data

```

```

    % Pixel size of the img.
    % size_Height = csvread(fileOpen,2,1,[2 1 2 1]);
    % size_Width = csvread(fileOpen,3,1,[3 1 3 1]);

    disp('CSV Data Loaded');
end
end

% ZYGO WHITE LIGHT INTERFEROMETER

%+FUH-----
----
% ReadMetroProFile(filename);
% content: Read MetroPro .dat file in Matrix for further operation
% parameters: filename
% returns: Error, InfoHeader of MetroPro data, IntensityMap as Matrix,
%          PhaseMap as Matrix [unit, meter]
% -FUH-----
----
function
[lError,InfoHeader,IntensityMap,PhaseMap]=ReadMetroProFile(filename)
% declarations/defines
%implementation
warning off all
lError=-1;
% InfoHeader=0; % This causes an error in MATLAB 2018a
IntensityMap=0;
PhaseMap=0;
    fid = fopen(filename,'r', 'b');
    try
        message = ferror(fid);
    catch
        emessage=strcat('File doesn''t exist or there is an access
violation to file : ',filename);
        errordlg(emessage,'File Type Error','modal')
    %     message = ferror(fid,'clear');
        lError=-1;
        return;
    end
    InfoHeader.MagicNumber=fread(fid,1,'*int32');
    if InfoHeader.MagicNumber~-2011495569 && InfoHeader.MagicNumber~-2011495568 && InfoHeader.MagicNumber~-2011495567
        lError=fclose(fid);
        lError=-1;
        return;
    end

    InfoHeader.HeaderFormat=fread(fid,1,'*int16');
    if InfoHeader.HeaderFormat~=1 && InfoHeader.HeaderFormat~=2 &&
InfoHeader.HeaderFormat~=3
        lError=fclose(fid);
        lError=-1;
        return;
    end
end

```



```

InfoHeader.HeaderSize=fread(fid,1,'*int32');
if InfoHeader.HeaderSize~=834 && InfoHeader.HeaderSize~=4096
    lError=fclose(fid);
    lError=-1;
    return;
end
InfoHeader.SoftwareType=fread(fid,1,'*int16');
InfoHeader.SoftwareDate=fread(fid,30,'uchar=>char');
InfoHeader.MajorVersion=fread(fid,1,'*int16');
InfoHeader.MinorVersion=fread(fid,1,'*int16');
InfoHeader.BugVersion=fread(fid,1,'*int16');
InfoHeader.IntensOriginX=fread(fid,1,'*int16');
InfoHeader.IntensOriginY=fread(fid,1,'*int16');
InfoHeader.IntensWidth=fread(fid,1,'*int16');
InfoHeader.IntensHeight=fread(fid,1,'*int16');
InfoHeader.NBuckets=fread(fid,1,'*int16');
InfoHeader.IntensRange=fread(fid,1,'*uint16');
InfoHeader.NrOfBytesIntens=fread(fid,1,'*int32');
%Phase
InfoHeader.PhaseOriginX=fread(fid,1,'*int16');
InfoHeader.PhaseOriginY=fread(fid,1,'*int16');
InfoHeader.PhaseWidth=fread(fid,1,'*int16');
InfoHeader.PhaseHeight=fread(fid,1,'*int16');
InfoHeader.NrOfBytesPhase=fread(fid,1,'*int32');
%timestamp
InfoHeader.TimeStamp=fread(fid,1,'*int32');
InfoHeader.Comment=fread(fid,82,'uchar=>char');
InfoHeader.Source=fread(fid,1,'*int16');
InfoHeader.IntfScaleFactor=fread(fid,1,'*float32');
InfoHeader.WaveLengthIn=fread(fid,1,'*float32');
InfoHeader.NumericAperture=fread(fid,1,'*float32');
InfoHeader.ObliquityFactor=fread(fid,1,'*float32');
InfoHeader.Magnification=fread(fid,1,'*float32');
InfoHeader.CameraRes=fread(fid,1,'*float32');
InfoHeader.AcquireMode=fread(fid,1,'*int16');
InfoHeader.IntensAvgs=fread(fid,1,'*int16');
InfoHeader.PZTCal=fread(fid,1,'*int16');
InfoHeader.PZTGainTolerance=fread(fid,1,'*int16');
InfoHeader.PZTGain=fread(fid,1,'*int16');
InfoHeader.PartThickness=fread(fid,1,'*float32');
InfoHeader.AGC=fread(fid,1,'*int16');
InfoHeader.TargetRange=fread(fid,1,'*float32');
InfoHeader.Spare1=fread(fid,1,'*int16');
InfoHeader.MinMod=fread(fid,1,'*int32');
InfoHeader.MinModPts=fread(fid,1,'*int32');
InfoHeader.PhaseRes=fread(fid,1,'*int16');
InfoHeader.MinimumAreaSize=fread(fid,1,'*int32');
InfoHeader.DisconAction=fread(fid,1,'*int16');
InfoHeader.DisconFilter=fread(fid,1,'*float32');
InfoHeader.ConnectionOrder=fread(fid,1,'*int16');
InfoHeader.DataSign=fread(fid,1,'*int16');
InfoHeader.CameraWidth=fread(fid,1,'*int16');
InfoHeader.CameraHeight=fread(fid,1,'*int16');
InfoHeader.SystemType=fread(fid,1,'*int16');
InfoHeader.SystemBoard=fread(fid,1,'*int16');
% I guess Serial Number of Instrument not to be negative so *uint16
% instead of *int16

```

```

InfoHeader.SystemSerial=fread(fid,1,'*uint16');
InfoHeader.InstrumentId=fread(fid,1,'*int16');
InfoHeader.ObjectiveName=fread(fid,12,'uchar=>char');
InfoHeader.PartNum=fread(fid,40,'uchar=>char');
InfoHeader.CodeVType=fread(fid,1,'*int16');
InfoHeader.PhaseAvg=fread(fid,1,'*int16');
InfoHeader.SubtractSysErr=fread(fid,1,'*int16');
InfoHeader.Spare2=fread(fid,16,'uchar=>char');
InfoHeader.PartSerNum=fread(fid,40,'uchar=>char');
InfoHeader.RefractiveIndex=fread(fid,1,'*float32');
InfoHeader.RemoveTiltBias=fread(fid,1,'*int16');
InfoHeader.RemoveFringes=fread(fid,1,'*int16');
InfoHeader.MaxAreaSize=fread(fid,1,'*int32');
InfoHeader.SetupType=fread(fid,1,'*int16');
InfoHeader.Spare3=fread(fid,2,'uchar=>char');
InfoHeader.PreConnectFilter=fread(fid,1,'*float32');
InfoHeader.Wavelength2=fread(fid,1,'*float32');
InfoHeader.WavelengthFold=fread(fid,1,'*int16');
InfoHeader.Wavelength1=fread(fid,1,'*float32');
InfoHeader.Wavelength3=fread(fid,1,'*float32');
InfoHeader.Wavelength4=fread(fid,1,'*float32');
InfoHeader.WavelengthSelect=fread(fid,8,'uchar=>char');
InfoHeader.FdaRes=fread(fid,1,'*int16');
InfoHeader.ScanDescription=fread(fid,20,'uchar=>char');
InfoHeader.NFiducials=fread(fid,1,'*int16');
InfoHeader.Fiducial1X=fread(fid,1,'*float32');
InfoHeader.Fiducial1Y=fread(fid,1,'*float32');
InfoHeader.Fiducial2X=fread(fid,1,'*float32');
InfoHeader.Fiducial2Y=fread(fid,1,'*float32');
InfoHeader.Fiducial3X=fread(fid,1,'*float32');
InfoHeader.Fiducial3Y=fread(fid,1,'*float32');
InfoHeader.Fiducial4X=fread(fid,1,'*float32');
InfoHeader.Fiducial4Y=fread(fid,1,'*float32');
InfoHeader.Fiducial5X=fread(fid,1,'*float32');
InfoHeader.Fiducial5Y=fread(fid,1,'*float32');
InfoHeader.Fiducial6X=fread(fid,1,'*float32');
InfoHeader.Fiducial6Y=fread(fid,1,'*float32');
InfoHeader.Fiducial7X=fread(fid,1,'*float32');
InfoHeader.Fiducial7Y=fread(fid,1,'*float32');
InfoHeader.PixelWidth=fread(fid,1,'*float32');
InfoHeader.PixelHeight=fread(fid,1,'*float32');
InfoHeader.ExitPupilDiam=fread(fid,1,'*float32');
InfoHeader.LightLevelPct=fread(fid,1,'*float32');
InfoHeader.CoordsState=fread(fid,1,'*int32');
InfoHeader.XPos=fread(fid,1,'*float32');
InfoHeader.YPos=fread(fid,1,'*float32');
InfoHeader.ZPos=fread(fid,1,'*float32');
InfoHeader.XRot=fread(fid,1,'*float32');
InfoHeader.YRot=fread(fid,1,'*float32');
InfoHeader.ZRot=fread(fid,1,'*float32');
InfoHeader.CoherenceMode=fread(fid,1,'*int16');
InfoHeader.SurfaceFilter=fread(fid,1,'*int16');
InfoHeader.SysErrFile=fread(fid,28,'uchar=>char');
InfoHeader.ZoomDescr=fread(fid,8,'uchar=>char');
InfoHeader.Spare4=fread(fid,264,'uchar=>char');

```

% jumping to Intensity Map

```

fseek(fid,double(InfoHeader.HeaderSize),'bof');
if (InfoHeader.NBuckets>1)

IntensityMap=zeros(double(InfoHeader.NBuckets),double(InfoHeader.Intens
Height),double(InfoHeader.IntensWidth),'uint16');
    for n=1:InfoHeader.NBuckets
        IntensTemp=fread(fid,[double(InfoHeader.IntensWidth)
double(InfoHeader.IntensHeight)], '*uint16');
        IntensTemp=IntensTemp';
        IntensTemp(IntensTemp >= 65535)=NaN;
        IntensityMap(n, :, :)=IntensTemp;
    end
else
    IntensityMap=fread(fid,[double(InfoHeader.IntensWidth)
double(InfoHeader.IntensHeight)], '*uint16');
    IntensityMap=IntensityMap';
    IntensityMap(IntensityMap >= 65535)=NaN;
end
if isempty(IntensityMap)
    IntensityMap=0;
end
% jumping to Phase Map

fseek(fid,double(InfoHeader.HeaderSize)+double(InfoHeader.IntensWidth)*
double(InfoHeader.IntensHeight)*double(InfoHeader.NBuckets)*2,'bof');
%     fseek(fid,HeaderSize+NrofBytesIntens,'bof');

PhaseMap=zeros(double(InfoHeader.IntensHeight),double(InfoHeader.Intens
Width));
    PhaseMap(:)=NaN;
    PhaseMapTemp=double(fread(fid,[double(InfoHeader.PhaseWidth)
double(InfoHeader.PhaseHeight)], '*int32'));
    PhaseMapTemp=PhaseMapTemp';
    PhaseMapTemp(PhaseMapTemp >= 2147483640)=NaN;
    if (InfoHeader.IntensHeight==0 && InfoHeader.IntensWidth==0)
        PhaseMap=PhaseMapTemp;
    else
        PhaseMap(double(InfoHeader.PhaseOriginY)+1:double(InfoHeader.PhaseOrigi
nY)+double(InfoHeader.PhaseHeight),double(InfoHeader.PhaseOriginX)+1:do
uble(InfoHeader.PhaseOriginX)+double(InfoHeader.PhaseWidth))=PhaseMapTe
mp;
    end

    switch InfoHeader.PhaseRes
        case 0
            R=4096;
        case 1
            R=32768;
        case 2
            R=131072;
        otherwise
            R=32768;
    end
end

```

```
PhaseMap=PhaseMap.*double(InfoHeader.IntfScaleFactor).*double(InfoHeader.ObliquityFactor).*double(InfoHeader.WaveLengthIn)./R;

    if isempty(PhaseMap)
        PhaseMap=0;
    end

    lError = fclose(fid);
return;
end
```

E.3.11 'planefit.m'

```

function [Leveled_Data] = planefit(Raw_Data,varargin)
% Levels Raw_Data either using the entire data or partial data based on
% varargin.

    % If varargin exist, use that as the binary mask, otherwise make a
mask
    % that is filled with 1's to use all data.
    if (nargin == 2)
        Binary_Mask = varargin{1};
    else
        Binary_Mask = ones(size(Raw_Data));
    end

    % Change to double or an error occurs
    Binary_Mask = double(Binary_Mask);

    % Change zeros to NaN so they do not affect the leveling.
    Binary_Mask(Binary_Mask == 0) = NaN;

    % Cut the data from the image based on the mask outcome.
    refPlane = Binary_Mask.*Raw_Data;

    % size of reference plane
    gridSize = size(refPlane);

    % Make address matrices
    [x,y] = meshgrid(1:1:gridSize(2),1:1:gridSize(1));

    % Find non-NaN data
    i = find(~isnan(refPlane));

    % Vectors of addresses with good data
    X = x(i);
    Y = y(i);
    Z = refPlane(i);

    % Create the fit matrix
    fit = [X Y ones(size(X))];

    % Find coefficients using linsolve (fit*c=Z)
    c = fit\Z;

    % Array of ones
    Leveled_Plane = ones(size(Raw_Data));

    % Generate best fit plane for the tilted reference
    Leveled_Plane = x.*c(1)+y.*c(2)+Leveled_Plane.*c(3);

    % Remove leveled plane from raw data
    Leveled_Data = Raw_Data - Leveled_Plane;

end

```


E.3.12 'polyminddev.m'

```

function [x_out,y_out,k_out,std_out] =
polyminddev(x_in,y_in,k_in,k_deg,bfp)
% This function uses the Procrustes analysis to minimize the
deviation
% for either the desire polynomial or for a best fitted line.
% See: https://www.mathworks.com/help/stats/procrustes.html

% x_in, y_in are curve data points
% k_in is polynomial fit input (can be set to 0 if bfp is 1)
% k_deg is polynomial degree
% bfp is (0)to use k_in OR (1) to use a best fit poly

% Remove initial offset
k_fit = polyfit(x_in,y_in,k_deg);
p = real(roots(polyder(k_fit)));

x_temp = x_in - max(p);
y_temp = y_in - polyval(k_fit,max(p));

std_plot = [];
rms_plot = [];
std_theory = [0 0 0];
i = 0;

while(1)
    i = i+1;

    if bfp == 1
        % Calculate best fit poly coefficients
        k_temp = polyfit(x_temp, y_temp, k_deg);

        % Pad coefficients with leading 0's if degree smaller than
3
        if k_deg < 3
            k_temp = [zeros(1,3 - k_deg) k_temp];
        end
    else
        % Use nominal poly coefficients
        k_temp = k_in;
    end

    % Calculate vector normal distance
    [dist_theory, x_theory, y_theory] =
normdist(x_temp,y_temp,k_temp);

    std_plot(i) = std(dist_theory);
    rms_plot(i) = rms(dist_theory);

    % Fit data to theoretical points
    [~,Z,~] = procrustes([x_theory' y_theory'],[x_temp
y_temp'],'scaling',0);

    % Save transformed data to temporary
    x_temp = Z(:,1); y_temp = Z(:,2);

```

```

    % Exit loop check
    if i > 2
        % Shift the theoretical STDev
        std_theory = [std_theory(2) std_theory(3) std(dist_theory)];

        % Break from loop when difference between STDev is less than
0.0001
        if abs(std_theory(3) - std_theory(1)) < 0.0001
            break;
        end
    else
        std_theory(i+1) = std(dist_theory);
    end
end

% Final outputs
x_out = x_temp;
y_out = y_temp;

% Recalculate polynomials
if bfp == 1
    % Recalculate best fit poly coefficients
    k_out = polyfit(x_temp, y_temp, k_deg);

    % Pad coefficients with leading 0's if degree smaller than 3
    if k_deg < 3
        k_out = [zeros(1,3 - k_deg) k_out];
    end
else
    % Use nominal poly coefficients
    k_out = k_in;
end

std_out = std(normdist(x_out,y_out,k_out));
end

```


E.3.13 'ReadMetroProFile.m'

```

%+FUH-----
----
% ReadMetroProFile(filename);
% content: Read MetroPro .dat file in Matrix for further operation
% parameters: filename
% returns: Error, InfoHeader of MetroPro data, IntensityMap as Matrix,
%          PhaseMap as Matrix [unit, meter]
% Developed by Dr. Chris Evans
% -FUH-----
----
function [lError,
InfoHeader,IntensityMap,PhaseMap]=ReadMetroProFile(filename)
% declarations/defines
%implementation
warning off all
lError=-1;
%InfoHeader=0;
IntensityMap=0;
PhaseMap=0;
    fid = fopen(filename,'r', 'b');
    try
        message = ferror(fid);
    catch
        emessage=strcat('File doesn''t exist or there is an access
violation to file : ',filename);
        errordlg(emessage,'File Type Error','modal')
    %    message = ferror(fid,'clear');
        lError=-1;
        return;
    end
    InfoHeader.MagicNumber=fread(fid,1,'*int32');
    if InfoHeader.MagicNumber~-2011495569 && InfoHeader.MagicNumber~-2011495568 && InfoHeader.MagicNumber~-2011495567
        lError=fclose(fid);
        lError=-1;
        return;
    end

    InfoHeader.HeaderFormat=fread(fid,1,'*int16');
    if InfoHeader.HeaderFormat~=1 && InfoHeader.HeaderFormat~=2 &&
InfoHeader.HeaderFormat~=3
        lError=fclose(fid);
        lError=-1;
        return;
    end

    InfoHeader.HeaderSize=fread(fid,1,'*int32');
    if InfoHeader.HeaderSize~=834 && InfoHeader.HeaderSize~=4096
        lError=fclose(fid);
        lError=-1;
        return;
    end

    InfoHeader.SoftwareType=fread(fid,1,'*int16');
    InfoHeader.SoftwareDate=fread(fid,30,'uchar=>char');
    InfoHeader.MajorVersion=fread(fid,1,'*int16');
    InfoHeader.MinorVersion=fread(fid,1,'*int16');

```

```

InfoHeader.BugVersion=fread(fid,1,'*int16');
InfoHeader.IntensOriginX=fread(fid,1,'*int16');
InfoHeader.IntensOriginY=fread(fid,1,'*int16');
InfoHeader.IntensWidth=fread(fid,1,'*int16');
InfoHeader.IntensHeight=fread(fid,1,'*int16');
InfoHeader.NBuckets=fread(fid,1,'*int16');
InfoHeader.IntensRange=fread(fid,1,'*uint16');
InfoHeader.NrOfBytesIntens=fread(fid,1,'*int32');
%Phase
InfoHeader.PhaseOriginX=fread(fid,1,'*int16');
InfoHeader.PhaseOriginY=fread(fid,1,'*int16');
InfoHeader.PhaseWidth=fread(fid,1,'*int16');
InfoHeader.PhaseHeight=fread(fid,1,'*int16');
InfoHeader.NrOfBytesPhase=fread(fid,1,'*int32');
%timestamp
InfoHeader.TimeStamp=fread(fid,1,'*int32');
InfoHeader.Comment=fread(fid,82,'uchar=>char');
InfoHeader.Source=fread(fid,1,'*int16');
InfoHeader.IntfScaleFactor=fread(fid,1,'*float32');
InfoHeader.WaveLengthIn=fread(fid,1,'*float32');
InfoHeader.NumericAperture=fread(fid,1,'*float32');
InfoHeader.ObliquityFactor=fread(fid,1,'*float32');
InfoHeader.Magnification=fread(fid,1,'*float32');
InfoHeader.CameraRes=fread(fid,1,'*float32');
InfoHeader.AcquireMode=fread(fid,1,'*int16');
InfoHeader.IntensAvgs=fread(fid,1,'*int16');
InfoHeader.PZTCal=fread(fid,1,'*int16');
InfoHeader.PZTGainTolerance=fread(fid,1,'*int16');
InfoHeader.PZTGain=fread(fid,1,'*int16');
InfoHeader.PartThickness=fread(fid,1,'*float32');
InfoHeader.AGC=fread(fid,1,'*int16');
InfoHeader.TargetRange=fread(fid,1,'*float32');
InfoHeader.Spare1=fread(fid,1,'*int16');
InfoHeader.MinMod=fread(fid,1,'*int32');
InfoHeader.MinModPts=fread(fid,1,'*int32');
InfoHeader.PhaseRes=fread(fid,1,'*int16');
InfoHeader.MinimumAreaSize=fread(fid,1,'*int32');
InfoHeader.DisconAction=fread(fid,1,'*int16');
InfoHeader.DisconFilter=fread(fid,1,'*float32');
InfoHeader.ConnectionOrder=fread(fid,1,'*int16');
InfoHeader.DataSign=fread(fid,1,'*int16');
InfoHeader.CameraWidth=fread(fid,1,'*int16');
InfoHeader.CameraHeight=fread(fid,1,'*int16');
InfoHeader.SystemType=fread(fid,1,'*int16');
InfoHeader.SystemBoard=fread(fid,1,'*int16');
% I guess Serial Number of Instrument not to be negative so *uint16
% instead of *int16
InfoHeader.SystemSerial=fread(fid,1,'*uint16');
InfoHeader.InstrumentId=fread(fid,1,'*int16');
InfoHeader.ObjectiveName=fread(fid,12,'uchar=>char');
InfoHeader.PartNum=fread(fid,40,'uchar=>char');
InfoHeader.CodeVType=fread(fid,1,'*int16');
InfoHeader.PhaseAvgs=fread(fid,1,'*int16');
InfoHeader.SubtractSysErr=fread(fid,1,'*int16');
InfoHeader.Spare2=fread(fid,16,'uchar=>char');
InfoHeader.PartSerNum=fread(fid,40,'uchar=>char');
InfoHeader.RefractiveIndex=fread(fid,1,'*float32');

```

```

InfoHeader.RemoveTiltBias=fread(fid,1,'*int16');
InfoHeader.RemoveFringes=fread(fid,1,'*int16');
InfoHeader.MaxAreaSize=fread(fid,1,'*int32');
InfoHeader.SetupType=fread(fid,1,'*int16');
InfoHeader.Spare3=fread(fid,2,'uchar=>char');
InfoHeader.PreConnectFilter=fread(fid,1,'*float32');
InfoHeader.Wavelength2=fread(fid,1,'*float32');
InfoHeader.WavelengthFold=fread(fid,1,'*int16');
InfoHeader.Wavelength1=fread(fid,1,'*float32');
InfoHeader.Wavelength3=fread(fid,1,'*float32');
InfoHeader.Wavelength4=fread(fid,1,'*float32');
InfoHeader.WavelengthSelect=fread(fid,8,'uchar=>char');
InfoHeader.FdaRes=fread(fid,1,'*int16');
InfoHeader.ScanDescription=fread(fid,20,'uchar=>char');
InfoHeader.NFiducials=fread(fid,1,'*int16');
InfoHeader.Fiducial1X=fread(fid,1,'*float32');
InfoHeader.Fiducial1Y=fread(fid,1,'*float32');
InfoHeader.Fiducial2X=fread(fid,1,'*float32');
InfoHeader.Fiducial2Y=fread(fid,1,'*float32');
InfoHeader.Fiducial3X=fread(fid,1,'*float32');
InfoHeader.Fiducial3Y=fread(fid,1,'*float32');
InfoHeader.Fiducial4X=fread(fid,1,'*float32');
InfoHeader.Fiducial4Y=fread(fid,1,'*float32');
InfoHeader.Fiducial5X=fread(fid,1,'*float32');
InfoHeader.Fiducial5Y=fread(fid,1,'*float32');
InfoHeader.Fiducial6X=fread(fid,1,'*float32');
InfoHeader.Fiducial6Y=fread(fid,1,'*float32');
InfoHeader.Fiducial7X=fread(fid,1,'*float32');
InfoHeader.Fiducial7Y=fread(fid,1,'*float32');
InfoHeader.PixelWidth=fread(fid,1,'*float32');
InfoHeader.PixelHeight=fread(fid,1,'*float32');
InfoHeader.ExitPupilDiam=fread(fid,1,'*float32');
InfoHeader.LightLevelPct=fread(fid,1,'*float32');
InfoHeader.CoordsState=fread(fid,1,'*int32');
InfoHeader.XPos=fread(fid,1,'*float32');
InfoHeader.YPos=fread(fid,1,'*float32');
InfoHeader.ZPos=fread(fid,1,'*float32');
InfoHeader.XRot=fread(fid,1,'*float32');
InfoHeader.YRot=fread(fid,1,'*float32');
InfoHeader.ZRot=fread(fid,1,'*float32');
InfoHeader.CoherenceMode=fread(fid,1,'*int16');
InfoHeader.SurfaceFilter=fread(fid,1,'*int16');
InfoHeader.SysErrFile=fread(fid,28,'uchar=>char');
InfoHeader.ZoomDescr=fread(fid,8,'uchar=>char');
InfoHeader.Spare4=fread(fid,264,'uchar=>char');

% jumping to Intensity Map
fseek(fid,double(InfoHeader.HeaderSize),'bof');
if (InfoHeader.NBuckets>1)

IntensityMap=zeros(double(InfoHeader.NBuckets),double(InfoHeader.Intens
Height),double(InfoHeader.IntensWidth),'uint16');
    for n=1:InfoHeader.NBuckets
        IntensTemp=fread(fid,[double(InfoHeader.IntensWidth)
double(InfoHeader.IntensHeight)],'*uint16');
        IntensTemp=IntensTemp';
        IntensTemp(IntensTemp >= 65535)=NaN;
    end
end

```

```

        IntensityMap(n, :, :) = IntensTemp;
    end
    else
        IntensityMap = fread(fid, [double(InfoHeader.IntensWidth)
double(InfoHeader.IntensHeight)], '*uint16');
        IntensityMap = IntensityMap';
        IntensityMap(IntensityMap >= 65535) = NaN;
    end
    if isempty(IntensityMap)
        IntensityMap = 0;
    end
    % jumping to Phase Map

    fseek(fid, double(InfoHeader.HeaderSize) + double(InfoHeader.IntensWidth) *
double(InfoHeader.IntensHeight) * double(InfoHeader.NBuckets) * 2, 'bof');
    %     fseek(fid, HeaderSize + NrOfBytesIntens, 'bof');

    PhaseMap = zeros(double(InfoHeader.IntensHeight), double(InfoHeader.Intens
Width));
    PhaseMap(:) = NaN;
    PhaseMapTemp = double(fread(fid, [double(InfoHeader.PhaseWidth)
double(InfoHeader.PhaseHeight)], '*int32'));
    PhaseMapTemp = PhaseMapTemp';
    PhaseMapTemp(PhaseMapTemp >= 2147483640) = NaN;
    if (InfoHeader.IntensHeight == 0 && InfoHeader.IntensWidth == 0)
        PhaseMap = PhaseMapTemp;
    else

    PhaseMap(double(InfoHeader.PhaseOriginY) + 1 : double(InfoHeader.PhaseOrigi
nY) + double(InfoHeader.PhaseHeight), double(InfoHeader.PhaseOriginX) + 1 : do
uble(InfoHeader.PhaseOriginX) + double(InfoHeader.PhaseWidth)) = PhaseMapTe
mp;
    end

    switch InfoHeader.PhaseRes
    case 0
        R = 4096;
    case 1
        R = 32768;
    case 2
        R = 131072;
    otherwise
        R = 32768;
    end

    PhaseMap = PhaseMap .* double(InfoHeader.IntfScaleFactor) .* double(InfoHeade
r.ObliquityFactor) .* double(InfoHeader.WaveLengthIn) ./ R;

    if isempty(PhaseMap)
        PhaseMap = 0;
    end

    lError = fclose(fid);
    return;

```


E.3.14 'rotatedata.m'

```
function [xy_rot] = rotatedata(xy_orig,theta,type)
%   [x,y] array, amount to rotate, 'deg' or 'rad'

    if strcmp(type,'deg')
        theta = theta*pi/180;
    end

    x = xy_orig(:,1);
    y = xy_orig(:,2);

    x_rot = x*cos(theta) - y*sin(theta);
    y_rot = x*sin(theta) + y*cos(theta);

    xy_rot = [x_rot y_rot];
end
```

E.3.15 'sumsquare.m'

```

function [sumsq] = sumsquare(xy_array)
%SUMSQUARE Find the sum of the squared values in each quadrant and with
respect to
% each axis. Output row is quadrant, column 1 is x-axis, column 2 is y-
axis
% 2 | 1
% ----
% 3 | 4

sumsq = zeros(4,2);

ind_q1 = find(xy_array(:,1)>0 & xy_array(:,2)>0);
sumsq(1,1) = sum(xy_array(ind_q1,1).^2);
sumsq(1,2) = sum(xy_array(ind_q1,2).^2);

ind_q2 = find(xy_array(:,1)<0 & xy_array(:,2)>0);
sumsq(2,1) = sum(xy_array(ind_q2,1).^2);
sumsq(2,2) = sum(xy_array(ind_q2,2).^2);

ind_q3 = find(xy_array(:,1)<0 & xy_array(:,2)<0);
sumsq(3,1) = sum(xy_array(ind_q3,1).^2);
sumsq(3,2) = sum(xy_array(ind_q3,2).^2);

ind_q4 = find(xy_array(:,1)>0 & xy_array(:,2)<0);
sumsq(4,1) = sum(xy_array(ind_q4,1).^2);
sumsq(4,2) = sum(xy_array(ind_q4,2).^2);

end

```

E.4 Shape Scripts

E.4.1 'asymmetric_rules.m'

```

%% Find object's border
% Ask user to select the feature they want to measure or select
% object 1 if only 1 exist.
obj_selected = 0;
if (num_Objects > 1)
    figure; imagesc(img_Labeled); title(['Select feature to measure
from: ' strrep(FileName, '_', '\_')]);
    axis tight equal

    while obj_selected == 0
        obj_pos = round(ginput(1));
        obj_selected = img_Labeled(obj_pos(2),obj_pos(1));
    end
else
    obj_selected = 1;
end

% Show the selected feature
imagesc(img_Labeled==obj_selected); title('Feature Selected');
axis tight equal

% Supress all other objects that were not selected from the mask by
% changing their value to 0
for i = 1:size(img_Binary,1)
    for j = 1:size(img_Binary,2)
        if img_Labeled(i,j) ~= obj_selected
            img_Binary(i,j) = 0;
        end
    end
end

% Finds boundaries of binary image
disp('Finding borders...');

mtx_Borders = cell2mat(bwboundaries(img_Binary,8,'noholes'));

% Make a temporary array containing the xy positions of the feature.
temp_xy = [mtx_Borders(:,2) mtx_Borders(:,1)];

% Remove last row if duplicate to first
if temp_xy(1,:) == temp_xy(end,:)
    temp_xy = temp_xy(1:end-1,:);
end

% Unit conversion from pixels to micrometers
temp_xy(:,1) = temp_xy(:,1)*res_XY;
temp_xy(:,2) = temp_xy(:,2)*res_XY;

```



```

%% Points of max distance
% Find the two points with the maximum distance between them, then
place
% the left most data point at the origin.
disp('Finding points of max distance...');
[ind_corners,dist,ang] = maxdist(temp_xy);

% Flip array so that the origin is in element 1
if temp_xy(ind_corners(1),1) > temp_xy(ind_corners(2),1)
    ind_corners = fliplr(ind_corners);
end

temp_x = temp_xy(:,1) - temp_xy(ind_corners(1),1);
temp_y = temp_xy(:,2) - temp_xy(ind_corners(1),2);

% Add an element between the two max distance points. This will store
% the index value for the corner point in between.
ind_corners = [ind_corners(1) 0 ind_corners(2)];

%% Find 3rd corner point
% Rotate data to normal orientation
% Rotate the data so that the two points composing the maximum dist lie
on
% the x-axis. After rotating, shift x-axis to half the distal point
% distance.
disp('Correcting data orientation...');
rotated_xy = rotatedata([temp_x,temp_y],-ang,'rad');
rotated_xy(:,1) = rotated_xy(:,1) - (dist/2);

% Find the sum of the squared values in each quadrant and with respect
to
% each axis.
[sumsq] = sumsquare(rotated_xy);

% Flip about x-axis if required.
if sum(sumsq(1:2,1)) > sum(sumsq(3:4,1))
    rotated_xy(:,2) = -rotated_xy(:,2);
    [sumsq] = sumsquare(rotated_xy);
end

% Find the minimum y-value that makes up a third corner of the shape.
[~,ind_corners(2)] = min(rotated_xy(:,2));

%% Two edge distances
% Find the distance between the 3 corner points already found by
% first finding the index of the min(y). First point is origin
% (0,0), second is furthest point below the y-axis, and third is
% the furthest point from origin.
x1 = rotated_xy(ind_corners(1),1);
y1 = rotated_xy(ind_corners(1),2);
x2 = rotated_xy(ind_corners(2),1);

```

```

y2 = rotated_xy(ind_corners(2),2);
x3 = rotated_xy(ind_corners(3),1);
y3 = rotated_xy(ind_corners(3),2);

dist1 = sqrt((x2-x1)^2+(y2-y1)^2);
dist2 = sqrt((x3-x2)^2+(y3-y2)^2);

%% Shape version and corners
% Shape comes in 3 versions: large (1), medium (2), small (3)
shape_ratios = [350/400 200/400 125/400];

% Determine version by comparing ratios
shape_compare = shape_ratios - min([dist1 dist2])/max([dist1 dist2]);
[~,shape_version] = min(abs(shape_compare));

% Flip about y-axis if required.
if (dist1 > dist2)
    rotated_xy(:,1) = -rotated_xy(:,1);
    [sumsq] = sumsquare(rotated_xy);
end

% Use the version to grab what should be the coordinates of all
% corner points (before scaling)
if shape_version == 1
    corners = [0 0;1 0;1 1;325/350 1; 25/350 1/4; 0 1/4];
elseif shape_version == 2
    corners = [0 0;1 0;1 1;7/8 1; 1/8 1/4; 0 1/4];
elseif shape_version == 3
    corners = [0 0;1 0;1 1;3/4 1; 1/4 1/4; 0 1/4];
end

```

E.4.2 'asymmetric_meas3.m'

```
% Calculate offsets from corner points

offset = zeros(1,6);
for i = 1:5
    offset(i) = round((ind_corners(i+1)-ind_corners(i))*perc_Length/2);
end
offset(6) = round((length(leveled_xy)-ind_corners(5))*perc_Length/2);

% Create an array for each edge with the calculated offset from the
corner
% point
edge1 = leveled_xy((ind_corners(1)+offset(1):ind_corners(2)-
offset(1)),:);
edge2 = leveled_xy((ind_corners(2)+offset(2):ind_corners(3)-
offset(2)),:);
edge3 = leveled_xy((ind_corners(3)+offset(3):ind_corners(4)-
offset(3)),:);
edge4 = leveled_xy((ind_corners(4)+offset(4):ind_corners(5)-
offset(4)),:);
edge5 = leveled_xy((ind_corners(5)+offset(5):ind_corners(6)-
offset(5)),:);
edge6 = leveled_xy((ind_corners(6)+offset(6):length(leveled_xy)-
offset(6)),:);

%% Calculate and display linear dimensions
dist1_3 = distmeas(edge1,edge3); % Measure distance between edges 1 and
3
dist1_5 = distmeas(edge1,edge5); % Measure distance between edges 1 and
5
dist3_5 = distmeas(edge3,edge5); % Measure distance between edges 3 and
5
dist2_6 = distmeas(fliplr(edge2),fliplr(edge6)); % Measure distance
between edges 2 and 6

disp(['Edge 1 to 3:   ' num2str(dist1_3) ' um']);
disp(['Edge 1 to 5:   ' num2str(dist1_5) ' um']);
disp(['Edge 2 to 6:   ' num2str(dist2_6) ' um']);

%% Measure assymetric line (edge 4)
k = [1/10000    , 1/50      , 0 , 0;
     1/1250     , 1/12.5    , 0 , 0;
     1/156.25   , 1/3.125   , 0 , 0;
     1/2500     , 1/25      , 0 , 0;
     1/312.5    , 1/6.25    , 0 , 0;
     1/39.0625  , 1/1.5625  , 0 , 0];

% Check if this is the smaller versions (dist1_3 ~= 400 um, smaller
will be
% half the size)
if 400*0.7 > dist1_3
    shape_version = shape_version + 3;
```

```

end

k = k(shape_version,:);

% Convert edge from pixels to actual unit values.
x = edge4(:,1);
y = edge4(:,2);

% Nominal polynomial coefficient fit output
[x_nom,y_nom,k_nom,~] = polymindev(x,y,k,3,0);
[norm_dist_nom, norm_x_nom, norm_y_nom] = normdist(x_nom,y_nom,k_nom);

mean_nom = mean(norm_dist_nom);
rms_nom = rms(norm_dist_nom);
std_nom= std(norm_dist_nom);

norm_dist_nom = norm_dist_nom';
norm_x_nom = norm_x_nom';
norm_y_nom = norm_y_nom';

% Best fit polynomial coefficient fit output
[x_bfp,y_bfp,k_bfp,~] = polymindev(x,y,k,3,1);
[norm_dist_bfp, norm_x_bfp, norm_y_bfp] = normdist(x_bfp,y_bfp,k_bfp);

mean_bfp = mean(norm_dist_bfp);
rms_bfp = rms(norm_dist_bfp);
std_bfp = std(norm_dist_bfp);

norm_dist_bfp = norm_dist_bfp';
norm_x_bfp = norm_x_bfp';
norm_y_bfp = norm_y_bfp';

% Nominal from best fit polynomial coefficient
[norm_dist_nombfp, norm_x_nombfp, norm_y_nombfp] =
normdist(x_bfp,y_bfp,k_nom);

mean_nombfp = mean(norm_dist_nombfp);
rms_nombfp = rms(norm_dist_nombfp);
std_nombfp = std(norm_dist_nombfp);

norm_dist_nombfp = norm_dist_nombfp';
norm_x_nombfp = norm_x_nombfp';
norm_y_nombfp = norm_y_nombfp';

%% Display polynomial statistics
disp('Nominal Polynomial Coefficients');
disp(['Coeffs: ' num2str(k_nom)]);
disp(['Mean: ' num2str(mean_nom)]);
disp(['RMS: ' num2str(rms_nom)]);
disp(['STD: ' num2str(std_nom)]);
disp('');
disp('Best Fit Polynomial Coefficients');

```

```

disp(['Coeffs: ' num2str(k_bfp)]);
disp(['Mean: ' num2str(mean_bfp)]);
disp(['RMS: ' num2str(rms_bfp)]);
disp(['STD: ' num2str(std_bfp)]);
disp('');
disp('Nominal After Best Fit Polynomial Coefficients');
disp(['Mean: ' num2str(mean_nombfp)]);
disp(['RMS: ' num2str(rms_nombfp)]);
disp(['STD: ' num2str(std_nombfp)]);

%% Plot data

% Data with corners and edges
figure; plot(leveled_xy(:,1),leveled_xy(:,2),'.b'); hold;
plot(leveled_xy(ind_corners,1),leveled_xy(ind_corners,2),'*r');
plot(edge1(:,1),edge1(:,2),'.g'); % Show edge 1
plot(edge2(:,1),edge2(:,2),'.g'); % Show edge 2
plot(edge3(:,1),edge3(:,2),'.g'); % Show edge 3
plot(edge5(:,1),edge5(:,2),'.g'); % Show edge 5
plot(edge4(:,1),edge4(:,2),'.g'); % Show edge 5
plot(edge6(:,1),edge6(:,2),'.g'); % Show edge 6
hold off;

% Actual vs theoretical
figure;plot(x_nom,y_nom,'.b',norm_x_nom,norm_y_nom,'r'); title('Actual
vs Theoretical'); axis tight

% Fit curve residuals
figure;plot(norm_x_nom,norm_dist_nom,'.b'); title('Nominal (norm-x-
nom,norm-dist-nom)'); axis tight
figure;plot(norm_x_bfp,norm_dist_bfp,'.r'); title('Best Fit (norm-x-
bfp,norm-dist-bfp)'); axis tight
figure;plot(norm_x_nombfp,norm_dist_nombfp,'.g'); title('Nom from BF
(norm-x-nombfp,norm-dist-nombfp)'); axis tight

```

E.4.3 'barbell_rules.m'

```

%% Find object's border
% Ask user to select the feature they want to measure or select
% object 1 if only 1 exist.
obj_selected = 0;
if (num_Objects > 1)
    figure; imagesc(img_Labeled); title(['Select feature to measure
from: ' strrep(FileName, '_', '\_')]);
    axis tight equal

    while obj_selected == 0
        obj_pos = round(ginput(1));
        obj_selected = img_Labeled(obj_pos(2),obj_pos(1));
    end
else
    obj_selected = 1;
end

% Show the selected feature
imagesc(img_Labeled==obj_selected); title('Feature Selected');
axis tight equal

%-----

% Supress all other objects that were not selected from the mask by
% changing their value to 0
for i = 1:size(img_Binary,1)
    for j = 1:size(img_Binary,2)
        if img_Labeled(i,j) ~= obj_selected
            img_Binary(i,j) = 0;
        end
    end
end

% Finds boundaries of binary image
disp('Finding borders...');

mtx_Borders = cell2mat(bwboundaries(img_Binary,8,'noholes'));

% Make a temporary array containing the xy positions of the feature.
temp_xy = [mtx_Borders(:,2) mtx_Borders(:,1)];

% Remove last row if duplicate to first
if temp_xy(1,:) == temp_xy(end,:)
    temp_xy = temp_xy(1:end-1,:);
end

% Unit conversion from pixels to micrometers
temp_xy(:,1) = temp_xy(:,1)*res_XY;
temp_xy(:,2) = temp_xy(:,2)*res_XY;

%% Points of max distance
% Find the two points with the maximum distance between them, then
place

```

```

% the left most data point at the origin.
disp('Finding points of max distance...');
[ind_corners,dist,ang] = maxdist(temp_xy);

% Flip array so that the origin is in element 1
if temp_xy(ind_corners(1),1) > temp_xy(ind_corners(2),1)
    ind_corners = fliplr(ind_corners);
end

temp_x = temp_xy(:,1) - temp_xy(ind_corners(1),1);
temp_y = temp_xy(:,2) - temp_xy(ind_corners(1),2);

% Add an element between the two max distance points. This will store
% the index value for the corner point in between.
ind_corners = [ind_corners(1) 0 ind_corners(2)];

%% Find 3rd corner point
% Rotate data to normal orientation
% Rotate the data so that the two points composing the maximum dist lie
on
% the x-axis. After rotating, shift x-axis to half the distal point
% distance.
disp('Correcting data orientation...');
rotated_xy = rotatedata([temp_x,temp_y],-ang,'rad');
rotated_xy(:,1) = rotated_xy(:,1) - (dist/2);

% Find the sum of the squared values in each quadrant and with respect
to
% each axis.
[sumsq] = sumsquare(rotated_xy);

% Find the minimum y-value that makes up a third corner of the shape.
[~,ind_corners(2)] = min(rotated_xy(:,2));

%% Two edge distances
% Find the distance between the 3 corner points already found by
% first finding the index of the min(y). First point is origin
% (0,0), second is furthest point below the y-axis, and third is
% the furthest point from origin.
x1 = rotated_xy(ind_corners(1),1);
y1 = rotated_xy(ind_corners(1),2);
x2 = rotated_xy(ind_corners(2),1);
y2 = rotated_xy(ind_corners(2),2);
x3 = rotated_xy(ind_corners(3),1);
y3 = rotated_xy(ind_corners(3),2);

dist1 = sqrt((x2-x1)^2+(y2-y1)^2);
dist2 = sqrt((x3-x2)^2+(y3-y2)^2);

% Mirror about x-axis if required. No y-axis flip required.
if (dist1 > dist2)
    rotated_xy(:,1) = -rotated_xy(:,1);
    [sumsq] = sumsquare(rotated_xy);

```

end

```
%% Shape corners
% Use the version to grab what should be the coordinates of all
% corner points (before scaling)
corners = [0 0;1 0;1 1/3;22/40 1/3;22/40 2/3;1 2/3;1 1;0 1;0 2/3;9/20
2/3;9/20 1/3;0 1/3];
```


E.4.4 'barbell_meas.m'

```
% Calculate offsets from corner points
offset = (1:12);
for i = 1:11
    offset(i) = round((ind_corners(i+1)-ind_corners(i))*perc_Length/2);
end
offset(12) = round((length(levelled_xy)-ind_corners(12))*perc_Length/2);

% Create an array for each edge with the calculated offset from the
corner
% point
edge1 = levelled_xy((ind_corners(1)+offset(1):ind_corners(2)-
offset(1)),:);
edge2 = levelled_xy((ind_corners(2)+offset(2):ind_corners(3)-
offset(2)),:);
edge3 = levelled_xy((ind_corners(3)+offset(3):ind_corners(4)-
offset(3)),:);
edge4 = levelled_xy((ind_corners(4)+offset(4):ind_corners(5)-
offset(4)),:);

edge5 = levelled_xy((ind_corners(5)+offset(5):ind_corners(6)-
offset(5)),:);
edge6 = levelled_xy((ind_corners(6)+offset(6):ind_corners(7)-
offset(6)),:);
edge7 = levelled_xy((ind_corners(7)+offset(7):ind_corners(8)-
offset(7)),:);
edge8 = levelled_xy((ind_corners(8)+offset(8):ind_corners(9)-
offset(8)),:);

edge9 = levelled_xy((ind_corners(9)+offset(9):ind_corners(10)-
offset(9)),:);
edge10 = levelled_xy((ind_corners(10)+offset(10):ind_corners(11)-
offset(10)),:);
edge11 = levelled_xy((ind_corners(11)+offset(11):ind_corners(12)-
offset(11)),:);
edge12 = levelled_xy((ind_corners(12)+offset(12):length(levelled_xy)-
offset(12)),:);

% Measure distance between edges 1 and 3/11
dist1_311 = distmeas(edge1,[edge3;edge11]);

% Measure distance between edges 1 and 5/9
dist1_59 = distmeas(edge1,[edge5;edge9]);

% Measure distance between edges 1 and 7
dist1_7 = distmeas(edge1,edge7);

% Measure distance between edges 2/6 and 8/12
dist26_812 = distmeas(fliplr([edge2;edge6]),fliplr([edge8;edge12]));

% Measure distance between edges 4 and 10
dist4_10 = distmeas(fliplr(edge4),fliplr(edge10));

%% Display statistics
```

```

disp(['Edge 1 to 3/11:   ' num2str(dist1_311) ' um']);
disp(['Edge 1 to 5/9:   ' num2str(dist1_59) ' um']);
disp(['Edge 1 to 7:     ' num2str(dist1_7) ' um']);
disp(['Edge 2/6 to 8/12: ' num2str(dist26_812) ' um']);
disp(['Edge 4 to 10:    ' num2str(dist4_10) ' um']);

% Plot data with corners
figure; plot(leveled_xy(:,1),leveled_xy(:,2),'.b'); hold;
plot(leveled_xy(ind_corners,1),leveled_xy(ind_corners,2),'*r'); hold
off;

```

E.4.5 'chain_rules2.m'

```

%% Find object's border
% Ask user to select the feature they want to measure or select
% object 1 if only 1 exist.
obj_selected = 0;
if (num_Objects > 1)
    figure; imagesc(img_Labeled); title(['Select feature to measure
from: ' strrep(FileName, '_', '\_')]);
    axis tight equal

    while obj_selected == 0
        obj_pos = round(ginput(1));
        obj_selected = img_Labeled(obj_pos(2),obj_pos(1));
    end
else
    obj_selected = 1;
end

% Show the selected feature
figure; imagesc(img_Labeled==obj_selected); title('Feature Selected');
axis tight equal

%-----

% Supress all other objects that were not selected from the mask by
% changing their value to 0
for i = 1:size(img_Binary,1)
    for j = 1:size(img_Binary,2)
        if img_Labeled(i,j) ~= obj_selected
            img_Binary(i,j) = 0;
        end
    end
end

% Finds boundaries of binary image
disp('Finding borders...');

mtx_Borders = bwboundaries(img_Binary,8);

if length(mtx_Borders) ~= 3
    disp('Error: The number of border elements for chain is not equal
to 3.')
    return;
end

%% Start of orient code
% Separate the variables
outer = fliplr(mtx_Borders{1}(:,:))*res_XY;
inner1 = fliplr(mtx_Borders{2}(:,:))*res_XY;
inner2 = fliplr(mtx_Borders{3}(:,:))*res_XY;

temp_xy = [outer;inner1;inner2]; % Used for step height

% Flip columns to set them as (X,Y);

```

```

% outer = outer;
% inner1 = inner1;
% inner2 = inner2;

% Remove last row if duplicate to first
if outer(1,:) == outer(end,:)
    outer = outer(1:end-1,:);
end

if inner1(1,:) == inner1(end,:)
    inner1 = inner1(1:end-1,:);
end

if inner2(1,:) == inner2(end,:)
    inner2 = inner2(1:end-1,:);
end

inner1_mean = mean(inner1);
inner2_mean = mean(inner2);

% Calculate the mean radius.
inner1_r = sqrt((inner1(:,1)-inner1_mean(1)).^2+(inner1(:,2)-
inner1_mean(2)).^2);
inner2_r = sqrt((inner2(:,1)-inner2_mean(1)).^2+(inner2(:,2)-
inner2_mean(2)).^2);

% Determine which fiducial is the circle/cross sign. (circle > cross)
if min(inner1_r) > min(inner2_r)
    circle = inner1;

    cross = inner2;
    cross_cntr = inner2_mean;
else
    circle = inner2;

    cross = inner1;
    cross_cntr = inner1_mean;
end

clear inner1 inner2 inner1_mean inner2_mean

% Move cross fiducial to centroid (0,0)
outer(:,1) = outer(:,1) - cross_cntr(1);
outer(:,2) = outer(:,2) - cross_cntr(2);

circle(:,1) = circle(:,1) - cross_cntr(:,1);
circle(:,2) = circle(:,2) - cross_cntr(:,2);

cross(:,1) = cross(:,1) - cross_cntr(:,1);
cross(:,2) = cross(:,2) - cross_cntr(:,2);

% Correct for rotation, placing the circle's centroid on the x-axis
circle_cntr = mean(circle);
angle = -atan2(circle_cntr(2),circle_cntr(1))*180/pi;
outer = rotatedata(outer,angle,'deg');
circle = rotatedata(circle,angle,'deg');

```

```

cross = rotatedata(cross,angle,'deg');

%% Continue correcting from here by removing centroids, changing
shape_size to use both x and y mean values
% Centroids of fiducials
circle_cntr = mean(circle);
cross_cntr = mean(cross);

% Calculate the fiducial distance and cutoff for where circles merge
dist_fiducials = circle_cntr(1);
shape_cutoff = dist_fiducials/4;

% Center the figure using the fiducials
outer(:,1) = outer(:,1) - dist_fiducials/2;
circle(:,1) = circle(:,1) - + dist_fiducials/2;
cross(:,1) = cross(:,1) - + dist_fiducials/2;

% Create arrays for the four arcs
ind_arc = find(outer(:,1)>=shape_cutoff);
arc_circle = outer(ind_arc,:);

ind_arc = find(outer(:,1)<=-shape_cutoff);
arc_cross = outer(ind_arc,:);

ind_arc = find(outer(:,1)<shape_cutoff & outer(:,1)>-shape_cutoff);
arc_middle = outer(ind_arc,:);

ind_arc = find(outer(:,1)<shape_cutoff & outer(:,1)>-shape_cutoff &
outer(:,2)>0);
arc_top = outer(ind_arc,:);

ind_arc = find(outer(:,1)<shape_cutoff & outer(:,1)>-shape_cutoff &
outer(:,2)<0);
arc_bottom = outer(ind_arc,:);

% % Plot
% figure; hold on;
% plot(arc_circle(:,1),arc_circle(:,2),'r');
% plot(arc_cross(:,1),arc_cross(:,2),'b');
% plot(arc_top(:,1),arc_top(:,2),'y');
% plot(arc_bottom(:,1),arc_bottom(:,2),'g');
% plot(circle(:,1),circle(:,2),'k');
% plot(cross(:,1),cross(:,2),'k');

```

E.4.6 'chain_meas3.m'

```

%% Metrology section
% Best fit circles to arcs
[xc_cross,yc_cross,R_cross,~] = circfit(arc_cross(:,1),arc_cross(:,2));
[xc_circle,yc_circle,R_circle,~] =
circfit(arc_circle(:,1),arc_circle(:,2));
[xc_top,yc_top,R_top,~] = circfit(arc_top(:,1),arc_top(:,2));
[xc_bottom,yc_bottom,R_bottom,~] =
circfit(arc_bottom(:,1),arc_bottom(:,2));

% Calculate the distance of each point from their respective centroid
dist_cross = sqrt((arc_cross(:,1)-xc_cross).^2+(arc_cross(:,2)-
yc_cross).^2);
dev_cross = dist_cross - R_cross;

dist_circle = sqrt((arc_circle(:,1)-xc_circle).^2+(arc_circle(:,2)-
yc_circle).^2);
dev_circle = dist_circle - R_circle;

dist_top = sqrt((arc_top(:,1)-xc_top).^2+(arc_top(:,2)-yc_top).^2);
dev_top = dist_top - R_top;

dist_bottom = sqrt((arc_bottom(:,1)-xc_bottom).^2+(arc_bottom(:,2)-
yc_bottom).^2);
dev_bottom = dist_bottom - R_bottom;

dev_all = [dev_cross;dev_circle;dev_top;dev_bottom];

% Calculate distances between circles
dist_crosscircle = sqrt((xc_cross-xc_circle)^2+(yc_cross-yc_circle)^2);
dist_topbottom = sqrt((xc_top-xc_bottom)^2+(yc_top-yc_bottom)^2);
dist_crosstop = sqrt((xc_cross-xc_top)^2+(yc_cross-yc_top)^2);
dist_crossbottom = sqrt((xc_cross-xc_bottom)^2+(yc_cross-yc_bottom)^2);
dist_circletop = sqrt((xc_circle-xc_top)^2+(yc_circle-yc_top)^2);
dist_circlebottom = sqrt((xc_circle-xc_bottom)^2+(yc_circle-
yc_bottom)^2);

%% Display statistics

% Display centroids, radii, and rms
disp('Cross Arc');
disp(['Xc: ' num2str(xc_cross)]);
disp(['Yc: ' num2str(yc_cross)]);
disp(['R: ' num2str(R_cross)]);
disp(['RMS: ' num2str(rms(dist_cross))]);
disp(['StDev: ' num2str(std(dist_cross))]);
disp('-----');

disp('Circle Arc');
disp(['Xc: ' num2str(xc_circle)]);
disp(['Yc: ' num2str(yc_circle)]);
disp(['R: ' num2str(R_circle)]);

```

```

disp(['RMS: ' num2str(rms(dist_circle))]);
disp(['StDev: ' num2str(std(dist_circle))]);
disp('-----');

disp('Top Arc');
disp(['Xc: ' num2str(xc_top)]);
disp(['Yc: ' num2str(yc_top)]);
disp(['R: ' num2str(R_top)]);
disp(['RMS: ' num2str(rms(dist_top))]);
disp(['StDev: ' num2str(std(dist_top))]);
disp('-----');

disp('Bottom Arc');
disp(['Xc: ' num2str(xc_bottom)]);
disp(['Yc: ' num2str(yc_bottom)]);
disp(['R: ' num2str(R_bottom)]);
disp(['RMS: ' num2str(rms(dist_bottom))]);
disp(['StDev: ' num2str(std(dist_bottom))]);
disp('-----');

disp('RMS Distances');
disp(['STD Total: ' num2str(std(dev_all))]);
disp(['Cross-Circle Dist: ' num2str(dist_crosscircle)]);
disp(['Top-Bottom Dist: ' num2str(dist_topbottom)]);
disp(['Cross-Top Dist: ' num2str(dist_crosstop)]);
disp(['Cross-Bottom Dist: ' num2str(dist_crossbottom)]);
disp(['Circle-Top Dist: ' num2str(dist_circletop)]);
disp(['Circle-Bottom Dist: ' num2str(dist_circlebottom)]);

%% Plot
figure; plot(dist_cross, '.b'); title('Arc: Cross');
figure; plot(dist_circle, '.r'); title('Arc: Circle');
figure; plot(dist_top, '.c'); title('Arc: Top');
figure; plot(dist_bottom, '.g'); title('Arc: Bottom');
figure; plot(dev_all, '.b');

dev_pos = [1 length(dev_cross) 0 0 0];
dev_pos(3) = dev_pos(2) + length(dev_bottom);
dev_pos(4) = dev_pos(3) + length(dev_circle);
dev_pos(5) = dev_pos(4) + length(dev_top);

figure; hold on;
plot(dev_pos(1):dev_pos(2), dev_cross, '.b');
plot(dev_pos(2)+1:dev_pos(3), dev_bottom, '.g');
plot(dev_pos(3)+1:dev_pos(4), dev_circle, '.r');
plot(dev_pos(4)+1:dev_pos(5), dev_top, '.c');
hold off;
title('Deviations from Fit');

% Display figure with fit circles
t = linspace(0, 2*pi);

figure; hold on;

```

```

plot(arc_cross(:,1),arc_cross(:,2),'.b')
plot(R_cross*cos(t)+xc_cross,R_cross*sin(t)+yc_cross,'r')

plot(arc_circle(:,1),arc_circle(:,2),'.r')
plot(R_circle*cos(t)+xc_circle,R_circle*sin(t)+yc_circle,'r')

plot(arc_top(:,1),arc_top(:,2),'.c')
plot(R_top*cos(t)+xc_top,R_top*sin(t)+yc_top,'r')

plot(arc_bottom(:,1),arc_bottom(:,2),'.g')
plot(R_bottom*cos(t)+xc_bottom,R_bottom*sin(t)+yc_bottom,'r')

plot(cross(:,1),cross(:,2),'.k');
plot(circle(:,1),circle(:,2),'.k');
hold off; axis tight equal

```


E.4.7 'rectangular_rules.m'

```

%% Find object's border
% Ask user to select the feature they want to measure or select
% object 1 if only 1 exist.
obj_selected = 0;
if (num_Objects > 1)
    figure; imagesc(img_Labeled); title(['Select feature to measure
from: ' strrep(FileName, '_', '\_')]);
    axis tight equal

    while obj_selected == 0
        obj_pos = round(ginput(1));
        obj_selected = img_Labeled(obj_pos(2),obj_pos(1));
    end
else
    obj_selected = 1;
end

% Show the selected feature
imagesc(img_Labeled==obj_selected); title('Feature Selected');
axis tight equal

%-----

% Supress all other objects that were not selected from the mask by
% changing their value to 0
for i = 1:size(img_Binary,1)
    for j = 1:size(img_Binary,2)
        if img_Labeled(i,j) ~= obj_selected
            img_Binary(i,j) = 0;
        end
    end
end

% Finds boundaries of binary image
disp('Finding borders...');

mtx_Borders = cell2mat(bwboundaries(img_Binary,8,'noholes'));

% Make a temporary array containing the xy positions of the feature.
temp_xy = [mtx_Borders(:,2) mtx_Borders(:,1)];

% Remove last row if duplicate to first
if temp_xy(1,:) == temp_xy(end,:)
    temp_xy = temp_xy(1:end-1,:);
end

% Unit conversion from pixels to micrometers
temp_xy(:,1) = temp_xy(:,1)*res_XY;
temp_xy(:,2) = temp_xy(:,2)*res_XY;

%% Points of max distance
% Find the two points with the maximum distance between them, then
place
% the left most data point at the origin.
disp('Finding points of max distance...');

```

```

[ind_corners,dist,ang] = maxdist(temp_xy);

% Flip array so that the origin is in element 1
if temp_xy(ind_corners(1),1) > temp_xy(ind_corners(2),1)
    ind_corners = fliplr(ind_corners);
end

temp_x = temp_xy(:,1) - temp_xy(ind_corners(1),1);
temp_y = temp_xy(:,2) - temp_xy(ind_corners(1),2);

% Add an element between the two max distance points. This will store
% the index value for the corner point in between.
ind_corners = [ind_corners(1) 0 ind_corners(2)];

%% Find 3rd corner point
% Rotate data to normal orientation
% Rotate the data so that the two points composing the maximum dist lie
on
% the x-axis. After rotating, shift x-axis to half the distal point
% distance.
disp('Correcting data orientation...');
rotated_xy = rotatedata([temp_x,temp_y],-ang,'rad');
rotated_xy(:,1) = rotated_xy(:,1) - (dist/2);

% Find the sum of the squared values in each quadrant and with respect
to
% each axis.
[sumsq] = sumsquare(rotated_xy);

% Find the minimum y-value that makes up a third corner of the shape.
[~,ind_corners(2)] = min(rotated_xy(:,2));

%% Shape corners
% Orientation doesn't matter, so do nothing
corners = [0 0;1 0;1 1;0 1];

```

E.4.8 'rectangular_meas.m'

```
% Calculate offsets from corner points
offset = [1:12];
for i = 1:3
    offset(i) = round((ind_corners(i+1)-ind_corners(i))*perc_Length/2);
end
offset(4) = round((length(leveled_xy)-ind_corners(4))*perc_Length/2);

% Create an array for each edge with the calculated offset from the
corner
% point
edge1 = leveled_xy([ind_corners(1)+offset(1):ind_corners(2)-
offset(1)],:);
edge2 = leveled_xy([ind_corners(2)+offset(2):ind_corners(3)-
offset(2)],:);
edge3 = leveled_xy([ind_corners(3)+offset(3):ind_corners(4)-
offset(3)],:);
edge4 = leveled_xy([ind_corners(4)+offset(4):length(leveled_xy)-
offset(4)],:);

% Measure distance between edges 1 and 3
dist1_3 = distmeas(edge1,edge3);
dist2_4 = distmeas(fliplr(edge2),fliplr(edge4));

%% Display statistics
disp(['Edge 1 to 3: ' num2str(dist1_3) ' um']);
disp(['Edge 2 to 4: ' num2str(dist2_4) ' um']);

figure; plot(leveled_xy(:,1),leveled_xy(:,2),'.b'); hold;
plot(leveled_xy(ind_corners,1),leveled_xy(ind_corners,2),'*r'); hold
off;
```

E.4.9 'secondorder_rules.m'

```

%% Find object's border
% Ask user to select the feature they want to measure or select
% object 1 if only 1 exist.
obj_selected = 0;
if (num_Objects > 1)
    figure; imagesc(img_Labeled); title(['Select feature to measure
from: ' strrep(FileName, '_', '\_')]);
    axis tight equal

    while obj_selected == 0
        obj_pos = round(ginput(1));
        obj_selected = img_Labeled(obj_pos(2),obj_pos(1));
    end
else
    obj_selected = 1;
end

% Show the selected feature
imagesc(img_Labeled==obj_selected); title('Feature Selected');
axis tight equal

%-----

% Suppress all other objects that were not selected from the mask by
% changing their value to 0
for i = 1:size(img_Binary,1)
    for j = 1:size(img_Binary,2)
        if img_Labeled(i,j) ~= obj_selected
            img_Binary(i,j) = 0;
        end
    end
end

% Finds boundaries of binary image
disp('Finding borders...');

mtx_Borders = cell2mat(bwboundaries(img_Binary,8,'noholes'));

% Make a temporary array containing the xy positions of the feature.
temp_xy = [mtx_Borders(:,2) mtx_Borders(:,1)];

% Remove last row if duplicate to first
if temp_xy(1,:) == temp_xy(end,:)
    temp_xy = temp_xy(1:end-1,:);
end

% Unit conversion from pixels to micrometers
temp_xy(:,1) = temp_xy(:,1)*res_XY;
temp_xy(:,2) = temp_xy(:,2)*res_XY;

%% Points of max distance
% Find the two points with the maximum distance between them, then
place
% the left most data point at the origin.
disp('Finding points of max distance...');

```

```

[ind_corners,dist,ang] = maxdist(temp_xy);

% Flip array so that the origin is in element 1
if temp_xy(ind_corners(1),1) > temp_xy(ind_corners(2),1)
    ind_corners = fliplr(ind_corners);
end

temp_x = temp_xy(:,1) - temp_xy(ind_corners(1),1);
temp_y = temp_xy(:,2) - temp_xy(ind_corners(1),2);

% Add an element between the two max distance points. This will store
% the index value for the corner point in between.
ind_corners = [ind_corners(1) 0 ind_corners(2)];

%% Find 3rd corner point
% Rotate data to normal orientation
% Rotate the data so that the two points composing the maximum dist lie
on
% the x-axis. After rotating, shift x-axis to half the distal point
% distance.
disp('Correcting data orientation...');
rotated_xy = rotatedata([temp_x,temp_y],-ang,'rad');
rotated_xy(:,1) = rotated_xy(:,1) - (dist/2);

% Find the sum of the squared values in each quadrant and with respect
to
% each axis.
[sumsq] = sumsquare(rotated_xy);

% Flip about x-axis if required.
if sum(sumsq(1:2,1)) > sum(sumsq(3:4,1))
    rotated_xy(:,2) = -rotated_xy(:,2);
    [sumsq] = sumsquare(rotated_xy);
end

% Find the minimum y-value that makes up a third corner of the shape.
[~,ind_corners(2)] = min(rotated_xy(:,2));

%% Two edge distances
% Find the distance between the 3 corner points already found by
% first finding the index of the min(y). First point is origin
% (0,0), second is furthest point below the y-axis, and third is
% the furthest point from origin.
x1 = rotated_xy(ind_corners(1),1);
y1 = rotated_xy(ind_corners(1),2);
x2 = rotated_xy(ind_corners(2),1);
y2 = rotated_xy(ind_corners(2),2);
x3 = rotated_xy(ind_corners(3),1);
y3 = rotated_xy(ind_corners(3),2);

dist1 = sqrt((x2-x1)^2+(y2-y1)^2);
dist2 = sqrt((x3-x2)^2+(y3-y2)^2);

%% Shape version and corners
% Shape comes in 3 versions: large (1), medium (2), small (3)

```

```

shape_ratios = [300/450 250/300 150/300];

% Determine version by comparing ratios
shape_compare = shape_ratios - min([dist1 dist2])/max([dist1 dist2]);
[~,shape_version] = min(abs(shape_compare));

% Flip about y-axis if required.
if (shape_version == 1 && dist2 > dist1) || (shape_version ~= 1 &&
dist1 > dist2)
    rotated_xy(:,1) = -rotated_xy(:,1);
    [sumsq] = sumsquare(rotated_xy);
end

% Use the version to grab what should be the coordinates of all
% corner points (before scaling)
if shape_version == 1
    corners = [0 0;1 0;1 1;425/450 1;25/450 1;0 1];
elseif shape_version == 2
    corners = [0 0;1 0;1 1;225/250 1;25/250 1;0 1];
elseif shape_version == 3
    corners = [0 0;1 0;1 1;125/150 1;25/150 1;0 1];
end

```

E.4.10 'secondorder_meas3.m'

```

% Calculate offsets from corner points
offset = (1:6);
for i = 1:5
    offset(i) = round((ind_corners(i+1)-ind_corners(i))*perc_Length/2);
end
offset(6) = round((length(leveled_xy)-ind_corners(6))*perc_Length/2);

% Create an array for each edge with the calculated offset from the
corner
% point
edge1 = leveled_xy((ind_corners(1)+offset(1):ind_corners(2)-
offset(1)),:);
edge2 = leveled_xy((ind_corners(2)+offset(2):ind_corners(3)-
offset(2)),:);
edge3 = leveled_xy((ind_corners(3)+offset(3):ind_corners(4)-
offset(3)),:);
edge4 = leveled_xy((ind_corners(4)+offset(4):ind_corners(5)-
offset(4)),:);
edge5 = leveled_xy((ind_corners(5)+offset(5):ind_corners(6)-
offset(5)),:);
edge6 = leveled_xy((ind_corners(6)+offset(6):length(leveled_xy)-
offset(6)),:);

%% Calculate and display linear dimensions
dist1_35 = distmeas(edge1,[edge3;edge5]); % Measure distance between
edges 1 and 3/5
dist2_6 = distmeas(edge2,edge6); % Measure distance between edges 2 and
6

disp(['Edge 1 to 3/5:   ' num2str(dist1_35) ' um']);
disp(['Edge 2 to 6:   ' num2str(dist2_6) ' um']);

%% Measure 2rd order line (edge 4)
% Check if this is the smaller versions (dist1_3 ~= 400 um, smaller
will be
% half the size)
k = [0 , 1/200 , 0 , 0;
     0 , 1/50  , 0 , 0;
     0 , 1/12.5 , 0 , 0;
     0 , 1/100 , 0 , 0;
     0 , 1/25  , 0 , 0;
     0 , 1/6.25 , 0 , 0];

if 400*0.7 > dist1_35
    shape_version = shape_version + 3;
end

k = k(shape_version,:);

% Convert edge from pixels to actual unit values.
x = edge4(:,1);
y = edge4(:,2);

% Nominal polynomial coefficient fit output
[x_nom,y_nom,k_nom,~] = polymindev(x,y,k,2,0);

```

```

[norm_dist_nom, norm_x_nom, norm_y_nom] = normdist(x_nom,y_nom,k_nom);

mean_nom = mean(norm_dist_nom);
rms_nom = rms(norm_dist_nom);
std_nom= std(norm_dist_nom);

norm_dist_nom = norm_dist_nom';
norm_x_nom = norm_x_nom';
norm_y_nom = norm_y_nom';

% Best fit polynomial coefficient fit output
[x_bfp,y_bfp,k_bfp,~] = polymindev(x,y,k,2,1);
[norm_dist_bfp, norm_x_bfp, norm_y_bfp] = normdist(x_bfp,y_bfp,k_bfp);

mean_bfp = mean(norm_dist_bfp);
rms_bfp = rms(norm_dist_bfp);
std_bfp = std(norm_dist_bfp);

norm_dist_bfp = norm_dist_bfp';
norm_x_bfp = norm_x_bfp';
norm_y_bfp = norm_y_bfp';

% Nominal from best fit polynomial coefficient
[norm_dist_nombfp, norm_x_nombfp, norm_y_nombfp] =
normdist(x_bfp,y_bfp,k_nom);

mean_nombfp = mean(norm_dist_nombfp);
rms_nombfp = rms(norm_dist_nombfp);
std_nombfp = std(norm_dist_nombfp);

norm_dist_nombfp = norm_dist_nombfp';
norm_x_nombfp = norm_x_nombfp';
norm_y_nombfp = norm_y_nombfp';

%% Display polynomial statistics
disp('Nominal Polynomial Coefficients');
disp(['Coeffs: ' num2str(k_nom)]);
disp(['Mean: ' num2str(mean_nom)]);
disp(['RMS: ' num2str(rms_nom)]);
disp(['STD: ' num2str(std_nom)]);
disp('');
disp('Best Fit Polynomial Coefficients');
disp(['Coeffs: ' num2str(k_bfp)]);
disp(['Mean: ' num2str(mean_bfp)]);
disp(['RMS: ' num2str(rms_bfp)]);
disp(['STD: ' num2str(std_bfp)]);
disp('');
disp('Nominal After Best Fit Polynomial Coefficients');
disp(['Mean: ' num2str(mean_nombfp)]);
disp(['RMS: ' num2str(rms_nombfp)]);
disp(['STD: ' num2str(std_nombfp)]);

%% Plot Data

% Data with corners and edges
figure; plot(leveled_xy(:,1),leveled_xy(:,2),'.b'); hold;
plot(leveled_xy(ind_corners,1),leveled_xy(ind_corners,2),'*r');

```



```

plot(edge1(:,1),edge1(:,2),'.g'); % Show edge 1
plot(edge2(:,1),edge2(:,2),'.g'); % Show edge 2
plot(edge3(:,1),edge3(:,2),'.g'); % Show edge 2
plot(edge5(:,1),edge5(:,2),'.g'); % Show edge 5
plot(edge6(:,1),edge6(:,2),'.g'); % Show edge 6
hold off;

% Actual vs theoretical
figure;plot(x_nom,y_nom,'.b',norm_x_nom,norm_y_nom,'r'); title('Actual
vs Theoretical'); axis tight

% Fit curve residuals
figure;plot(norm_x_nom,norm_dist_nom,'.b'); title('Nominal (norm-x-
nom,norm-dist-nom)'); axis tight
figure;plot(norm_x_bfp,norm_dist_bfp,'.r'); title('Best Fit (norm-x-
bfp,norm-dist-bfp)'); axis tight
figure;plot(norm_x_nombfp,norm_dist_nombfp,'.g'); title('Nom from BF
(norm-x-nombfp,norm-dist-nombfp)'); axis tigh

```

E.4.11 'thirdorder_rules.m'

```

%% Find object's border
% Ask user to select the feature they want to measure or select
% object 1 if only 1 exist.
obj_selected = 0;
if (num_Objects > 1)
    figure; imagesc(img_Labeled); title(['Select feature to measure
from: ' strrep(FileName, '_', '\_')]);
    axis tight equal

    while obj_selected == 0
        obj_pos = round(ginput(1));
        obj_selected = img_Labeled(obj_pos(2),obj_pos(1));
    end
else
    obj_selected = 1;
end

% Show the selected feature
imagesc(img_Labeled==obj_selected); title('Feature Selected');
axis tight equal

%-----

% Supress all other objects that were not selected from the mask by
% changing their value to 0
for i = 1:size(img_Binary,1)
    for j = 1:size(img_Binary,2)
        if img_Labeled(i,j) ~= obj_selected
            img_Binary(i,j) = 0;
        end
    end
end

% Finds boundaries of binary image
disp('Finding borders...');

mtx_Borders = cell2mat(bwboundaries(img_Binary,8,'noholes'));

% Make a temporary array containing the xy positions of the feature.
temp_xy = [mtx_Borders(:,2) mtx_Borders(:,1)];

% Remove last row if duplicate to first
if temp_xy(1,:) == temp_xy(end,:)
    temp_xy = temp_xy(1:end-1,:);
end

% Unit conversion from pixels to micrometers
temp_xy(:,1) = temp_xy(:,1)*res_XY;
temp_xy(:,2) = temp_xy(:,2)*res_XY;

%% Points of max distance
% Find the two points with the maximum distance between them, then
place
% the left most data point at the origin.
disp('Finding points of max distance...');

```

```

[ind_corners,dist,ang] = maxdist(temp_xy);

% Flip array so that the origin is in element 1
if temp_xy(ind_corners(1),1) > temp_xy(ind_corners(2),1)
    ind_corners = fliplr(ind_corners);
end

temp_x = temp_xy(:,1) - temp_xy(ind_corners(1),1);
temp_y = temp_xy(:,2) - temp_xy(ind_corners(1),2);

% Add an element between the two max distance points. This will store
% the index value for the corner point in between.
ind_corners = [ind_corners(1) 0 ind_corners(2)];

%% Find 3rd corner point
% Rotate data to normal orientation
% Rotate the data so that the two points composing the maximum dist lie
on
% the x-axis. After rotating, shift x-axis to half the distal point
% distance.
disp('Correcting data orientation...');
rotated_xy = rotatedata([temp_x,temp_y],-ang,'rad');
rotated_xy(:,1) = rotated_xy(:,1) - (dist/2);

% Find the sum of the squared values in each quadrant and with respect
to
% each axis.
[sumsq] = sumsquare(rotated_xy);

% Flip about x-axis if required.
if sum(sumsq(1:2,1)) > sum(sumsq(3:4,1))
    rotated_xy(:,2) = -rotated_xy(:,2);
    [sumsq] = sumsquare(rotated_xy);
end

% Find the minimum y-value that makes up a third corner of the shape.
[~,ind_corners(2)] = min(rotated_xy(:,2));

%% Two edge distances
% Find the distance between the 3 corner points already found by
% first finding the index of the min(y). First point is origin
% (0,0), second is furthest point below the y-axis, and third is
% the furthest point from origin.
x1 = rotated_xy(ind_corners(1),1);
y1 = rotated_xy(ind_corners(1),2);
x2 = rotated_xy(ind_corners(2),1);
y2 = rotated_xy(ind_corners(2),2);
x3 = rotated_xy(ind_corners(3),1);
y3 = rotated_xy(ind_corners(3),2);

dist1 = sqrt((x2-x1)^2+(y2-y1)^2);
dist2 = sqrt((x3-x2)^2+(y3-y2)^2);

%% Shape version and corners
% Shape comes in 3 versions: large (1), medium (2), small (3)

```

```

shape_ratios = [400/450 250/400 150/400];

% Determine version by comparing ratios
shape_compare = shape_ratios - min([dist1 dist2])/max([dist1 dist2]);
[~,shape_version] = min(abs(shape_compare));

% Flip about y-axis if required.
if (shape_version == 1 && dist2 > dist1) || (shape_version ~= 1 &&
dist1 > dist2)
    rotated_xy(:,1) = -rotated_xy(:,1);
    [sumsq] = sumsquare(rotated_xy);
end

% Use the version to grab what should be the coordinates of all
% corner points (before scaling)
if shape_version == 1
    corners = [0 0;1 0;1 1;40/45 1];
elseif shape_version == 2
    corners = [0 0;1 0;1 1;20/25 1];
elseif shape_version == 3
    corners = [0 0;1 0;1 1;10/15 1];
end

```

E.4.12 'thirdorder_meas3.m'

```

% Calculate offsets from corner points
offset = (1:4);
for i = 1:3
    offset(i) = round((ind_corners(i+1)-ind_corners(i))*perc_Length/2);
end
offset(4) = round((length(leveled_xy)-ind_corners(4))*perc_Length/2);

% Create an array for each edge with the calculated offset from the
corner
% point
edge1 = leveled_xy((ind_corners(1)+offset(1):ind_corners(2)-
offset(1)),:);
edge2 = leveled_xy((ind_corners(2)+offset(2):ind_corners(3)-
offset(2)),:);
edge3 = leveled_xy((ind_corners(3)+offset(3):ind_corners(4)-
offset(3)),:);
edge4 = leveled_xy((ind_corners(4)+offset(4):length(leveled_xy)-
offset(4)),:);

%% Calculate and display linear dimensions
dist1_3 = distmeas(edge1,edge3); % Measure distance between edges 1 and
3

disp(['Edge 1 to 3:   ' num2str(dist1_3) ' um']);

%% Measure 3rd order line (edge 4)
k = [1/40000 , 0 , 0 , 0;
      1/5000  , 0 , 0 , 0;
      1/625   , 0 , 0 , 0;
      1/10000 , 0 , 0 , 0;
      1/1250  , 0 , 0 , 0;
      1/156.25 , 0 , 0 , 0];

% Check if this is the smaller versions (dist1_3 ~= 400 um, smaller
will be
% half the size)
if 400*0.7 > dist1_3
    shape_version = shape_version + 3;
end

k = k(shape_version,:);

% Convert edge from pixels to actual unit values.
x = edge4(:,1);
y = edge4(:,2);

% Nominal polynomial coefficient fit output
[x_nom,y_nom,k_nom,~] = polymindev(x,y,k,3,0);
[norm_dist_nom, norm_x_nom, norm_y_nom] = normdist(x_nom,y_nom,k_nom);

mean_nom = mean(norm_dist_nom);
rms_nom = rms(norm_dist_nom);
std_nom= std(norm_dist_nom);

norm_dist_nom = norm_dist_nom';

```

```

norm_x_nom = norm_x_nom';
norm_y_nom = norm_y_nom';

% Best fit polynomial coefficient fit output
[x_bfp,y_bfp,k_bfp,~] = polymindev(x,y,k,3,1);
[norm_dist_bfp, norm_x_bfp, norm_y_bfp] = normdist(x_bfp,y_bfp,k_bfp);

mean_bfp = mean(norm_dist_bfp);
rms_bfp = rms(norm_dist_bfp);
std_bfp = std(norm_dist_bfp);

norm_dist_bfp = norm_dist_bfp';
norm_x_bfp = norm_x_bfp';
norm_y_bfp = norm_y_bfp';

% Nominal from best fit polynomial coefficient
[norm_dist_nombfp, norm_x_nombfp, norm_y_nombfp] =
normdist(x_bfp,y_bfp,k_nom);

mean_nombfp = mean(norm_dist_nombfp);
rms_nombfp = rms(norm_dist_nombfp);
std_nombfp = std(norm_dist_nombfp);

norm_dist_nombfp = norm_dist_nombfp';
norm_x_nombfp = norm_x_nombfp';
norm_y_nombfp = norm_y_nombfp';

%% Display polynomial statistics
disp('Nominal Polynomial Coefficients');
disp(['Coeffs: ' num2str(k_nom)]);
disp(['Mean: ' num2str(mean_nom)]);
disp(['RMS: ' num2str(rms_nom)]);
disp(['STD: ' num2str(std_nom)]);
disp('');
disp('Best Fit Polynomial Coefficients');
disp(['Coeffs: ' num2str(k_bfp)]);
disp(['Mean: ' num2str(mean_bfp)]);
disp(['RMS: ' num2str(rms_bfp)]);
disp(['STD: ' num2str(std_bfp)]);
disp('');
disp('Nominal After Best Fit Polynomial Coefficients');
disp(['Mean: ' num2str(mean_nombfp)]);
disp(['RMS: ' num2str(rms_nombfp)]);
disp(['STD: ' num2str(std_nombfp)]);

%% Plot Data
% Data with corners and edges
figure; plot(veled_xy(:,1),veled_xy(:,2),'.b'); hold;
plot(veled_xy(ind_corners,1),veled_xy(ind_corners,2),'*r');
plot(edge1(:,1),edge1(:,2),'.g'); % Show edge 1
plot(edge3(:,1),edge3(:,2),'.g'); % Show edge 3
hold off;

% Actual vs theoretical
figure;plot(x_nom,y_nom,'.b',norm_x_nom,norm_y_nom,'r'); title('Actual
vs Theoretical'); axis tight

```

```
% Fit curve residuals
figure;plot(norm_x_nom,norm_dist_nom,'.b'); title('Nominal (norm-x-
nom,norm-dist-nom)'); axis tight
figure;plot(norm_x_bfp,norm_dist_bfp,'.r'); title('Best Fit (norm-x-
bfp,norm-dist-bfp)'); axis tight
figure;plot(norm_x_nombfp,norm_dist_nombfp,'.g'); title('Nom from BF
(norm-x-nombfp,norm-dist-nombfp)'); axis tight
```

E.4.13 'triangle_rules.m'

```

%% Find object's border
% Finds boundaries of binary image
disp('Finding borders...');

for i = 1:num_Objects
    mtx_Borders{i} =
cell2mat(bwboundaries(img_Labeled==i,8,'noholes'));
end

% Loop through all objects
for i = 1:num_Objects
    % Make a temporary array containing the xy positions of the
    feature.
    temp_xy = [mtx_Borders{i}(:,2) mtx_Borders{i}(:,1)];

    % Remove last row if duplicate to first
    if temp_xy(1,:) == temp_xy(end,:)
        temp_xy = temp_xy(1:end-1,:);
    end

    %% Find corners
    % Calculate centroid
    mean_1 = mean(temp_xy(:,1));
    mean_2 = mean(temp_xy(:,2));

    % Find furthest point from centroid to labeled as point 1
    dist = sqrt((temp_xy(:,1)-mean_1).^2 + (temp_xy(:,2)-mean_2).^2);
    [point_1,ind_1] = max(dist);

    % Rearrange the arrays so that the starting point is point furthest
    from
    % centroid
    temp_xyr = [temp_xy(ind_1:end,1)
temp_xy(ind_1:end,2);temp_xy(1:ind_1-1,1) temp_xy(1:ind_1-1,2)];

    mean_1r = mean(temp_xyr(:,1));
    mean_2r = mean(temp_xyr(:,2));

    distr = sqrt((temp_xyr(:,1)-mean_1r).^2 + (temp_xyr(:,2)-
mean_2r).^2);

    % Find the min between the start/end and middle point, then use the
    min to
    % the middle to find the next peak.
    ind_25 = round(length(distr)/4);
    ind_50 = round(length(distr)/2);
    ind_75 = round(3*length(distr)/4);

    % Location of points 2 and 3
    [point_1r,ind_1r] = max(distr);
    [point_2r,ind_2r] = max(distr(ind_25:ind_50));

```



```

[point_3r,ind_3r] = max(distr(ind_50:ind_75));

ind_2r = ind_2r + ind_25 - 1;
ind_3r = ind_3r + ind_50 - 1;


%      % Plot a bunch of figures.
%      figure;
%      plot(distr);
%      hold on;
%      plot(ind_1r,distr(ind_1r),'*r');
%      plot(ind_2r,distr(ind_2r),'*r');
%      plot(ind_3r,distr(ind_3r),'*r');
%      hold off;
%
%      figure;
%      plot(temp_xyr(:,1),temp_xyr(:,2));
%      hold on;
%      plot(temp_xyr(ind_1r,1),temp_xyr(ind_1r,2),'*r');
%      plot(temp_xyr(ind_2r,1),temp_xyr(ind_2r,2),'*r');
%      plot(temp_xyr(ind_3r,1),temp_xyr(ind_3r,2),'*r');
%      hold off;

end

```

E.4.14 'triangle_meas.m'

```

% Create edges
edge_Long1 = temp_xyr(ind_1r:ind_2r-1,:);
edge_Long2 = temp_xyr(ind_3r:end,:);
edge_Short = temp_xyr(ind_2r:ind_3r-1,:);

% Truncate edges
trunc_edge = floor(perc_Length*length(edge_Long1)/2);
edge_Long1 = edge_Long1(trunc_edge:end-trunc_edge,:);

trunc_edge = floor(perc_Length*length(edge_Long2)/2);
edge_Long2 = edge_Long2(trunc_edge:end-trunc_edge,:);

trunc_edge = floor(perc_Length*length(edge_Short)/2);
edge_Short = edge_Short(trunc_edge:end-trunc_edge,:);

% Try to fit a line to the short edge. If it throws a warning about the
x
% scale, switch the axes
flip_axes = 0;
s = warning('error','MATLAB:polyfit:RepeatedPointsOrRescale');
try
    coefs_Short = polyfit(edge_Short(:,1),edge_Short(:,2),1);
catch ME
    edge_Short = fliplr(edge_Short);
    edge_Long1 = fliplr(edge_Long1);
    edge_Long2 = fliplr(edge_Long2);

    coefs_Short = polyfit(edge_Short(:,1),edge_Short(:,2),1);
    flip_axes = 1;
end

coefs_Long1 = polyfit(edge_Long1(:,1),edge_Long1(:,2),1);
coefs_Long2 = polyfit(edge_Long2(:,1),edge_Long2(:,2),1);

% Find intersection of the lines
int_SL1x = fzero(@(x) polyval(coefs_Short-coefs_Long1,x),3);
int_SL1y = polyval(coefs_Short,int_SL1x);

int_SL2x = fzero(@(x) polyval(coefs_Short-coefs_Long2,x),3);
int_SL2y = polyval(coefs_Short,int_SL2x);

int_L1L2x = fzero(@(x) polyval(coefs_Long1-coefs_Long2,x),3);
int_L1L2y = polyval(coefs_Long1,int_L1L2x);

coefs_Theory = [-1/coefs_Short(1) int_L1L2y - (-
1*int_L1L2x/coefs_Short(1))];

int_Theoryx = fzero(@(x) polyval(coefs_Theory-coefs_Short,x),3);
int_Theoryy = polyval(coefs_Theory,int_Theoryx);

int_Meanx = (int_SL1x + int_SL2x)/2;
int_Meany = (int_SL1y + int_SL2y)/2;

% Plot figures
figure;

```

```

hold on;
plot(edge_Long1(:,1),edge_Long1(:,2),'r'); axis tight equal
plot(edge_Long2(:,1),edge_Long2(:,2),'b'); axis tight equal
plot(edge_Short(:,1),edge_Short(:,2),'g'); axis tight equal
plot(int_SL1x,int_SL1y,'*r',int_SL2x,int_SL2y,'*g',int_L1L2x,int_L1L2y,
'*b'); axis tight equal
title(['Triangle ' num2str(i)]);
hold off;

% Give distances between L1L2 intersect to SL1 and SL2, along with
maybe
% the angles
dist_SL1toL1L2 = sqrt((int_SL1x-int_L1L2x)^2 + (int_SL1y-
int_L1L2y)^2)*res_XY;
dist_SL2toL1L2 = sqrt((int_SL2x-int_L1L2x)^2 + (int_SL2y-
int_L1L2y)^2)*res_XY;
dist_SL1toSL2 = sqrt((int_SL1x-int_SL2x)^2 + (int_SL1y-
int_SL2y)^2)*res_XY;
dist_L1L2toMean = sqrt((int_L1L2x-int_Meanx)^2 + (int_L1L2y-
int_Meany)^2)*res_XY;
dist_L1L2toTheory = sqrt((int_L1L2x-int_Theoryx)^2 + (int_L1L2y-
int_Theoryy)^2)*res_XY;

% Display results
disp(' ')
disp(['---- Triangle ' num2str(i) ' ----'])
disp(['SL1 to L1L2: ' num2str(dist_SL1toL1L2)])
disp(['SL2 to L1L2: ' num2str(dist_SL2toL1L2)])
disp(['SL1 to SL2: ' num2str(dist_SL1toSL2)])
disp(['Mean to L1L2: ' num2str(dist_L1L2toMean)])
disp(['Theory to L1L2: ' num2str(dist_L1L2toTheory)])

```

E.5 Supporting Scripts

E.5.1 'shapeorient.m'

```
% Shift the shape back half the maximum distance.
rotated_xy(:,1) = rotated_xy(:,1) + (dist/2);

% Grab the index of the origin and the last point for leveling.
if rotated_xy(ind_corners(1),1) > rotated_xy(ind_corners(3),1)
    ind_corners = fliplr(ind_corners);
end

% Determine which way to take the vector between the two points used
% for leveling. Also, add a small buffer to avoid corner noise. (20%
% length)
vec1 = abs(ind_corners(2) - ind_corners(1));
vec2 = (ind_corners(2) + length(rotated_xy) - ind_corners(1));
vec_offset = round(min([vec1 vec2])*0.2);

if vec2 > vec1
    vec_lvl =
rotated_xy(min(ind_corners(1:2))+vec_offset:max(ind_corners(1:2))-
vec_offset,:);
else
    vec_lvl = rotated_xy([ind_corners(1)+vec_offset:end
1:ind_corners(2)-vec_offset],:);
end

%     figure; plot(rotated_xy(:,1),rotated_xy(:,2),'.');
%     hold; plot(vec_lvl(:,1),vec_lvl(:,2),'*g');
%     return;

% Calculate the best fit line of the level vector, then apply that
angle to
% the rotated data to produce leveled data.
P = polyfit(vec_lvl(:,1),vec_lvl(:,2),1);
lvl_ang = atan(-P(1));
leveled_xy = rotatedata(rotated_xy,lvl_ang,'rad');

clear P lvl_ang vec_lvl vec_offset vec1 vec2 dist dist1 dist2 x1 x2 x3
y1 y2 y3;

%-----

% Find remaining corners and reorganize array
% Check direction of array and change array to run counter-clockwise if
% it runs clockwise
disp('Checking data direction...')
ind_corners = ind_corners - ind_corners(1);
for i = 2:3
    if ind_corners(i) < 0
        ind_corners(i) = ind_corners(i) + length(rotated_xy);
    end
end
end
```

```

if ind_corners(2) > ind_corners(3)
    leveled_xy = flipud(leveled_xy(:, :));
end

% Shift data back so that the first point is (0,0)
disp('Finding remaining corners...');
leveled_xy(:,1) = leveled_xy(:,1) - min(leveled_xy(:,1));
leveled_xy(:,2) = leveled_xy(:,2) - min(leveled_xy(:,2));
temp_x = leveled_xy(:,1);
temp_y = leveled_xy(:,2);

% Determine shape's size
shape_size = [max(temp_x) max(temp_y)];

% Scale the corner values and find the closest data point to that value
corners(:,1) = corners(:,1)*shape_size(1);
corners(:,2) = corners(:,2)*shape_size(2);

for i = 1:length(corners(:,1))
    temp_dist = sqrt((temp_x-corners(i,1)).^2+(temp_y-
corners(i,2)).^2);
    [~,ind_corners(i)] = min(temp_dist);
end

% Reorganize array so that the first element is the origin and the data
% goes counter clockwise. Change corner indices and sort.
disp('Changing array to start at origin...');
leveled_xy = leveled_xy([ind_corners(1):end 1:ind_corners(1)-1],:);
ind_corners = ind_corners - ind_corners(1) + 1;

for i = 1:length(ind_corners)
    if (ind_corners(i) < 1)
        ind_corners(i) = ind_corners(i)+length(leveled_xy);
    end
end

% figure; plot(leveled_xy(:,1),leveled_xy(:,2),'b'); hold;
% plot(leveled_xy(ind_corners,1),leveled_xy(ind_corners,2),'*r'); hold
off;

```

E.5.2 'stepmeas.m'

```

%for m = 0:1:50
%step_Padding = m;

% Measures the step height of the feature
img_Step = img_Data*0;

img_Step(img_Labeled==obj_selected) = 1;
img_Test2 = img_Step;

% Create a padded region around the etched regions to remove potential
noise regions
if (step_Padding > 0)
    img_Step = padarray(img_Step,[step_Padding step_Padding],1);
    col = temp_xy(:,1)/res_XY+step_Padding;
    row = temp_xy(:,2)/res_XY+step_Padding;

    % Steps through all locations with 0 and makes the surrounding
    % locations 0 up to the 'padding' value.
    for i = 1:length(row)
        img_Step(row(i)-step_Padding:row(i)+step_Padding,col(i)-
step_Padding:col(i)+step_Padding)=0;
    end

    % Crop the padded image back to the original size.
    img_Step = img_Step(step_Padding+1:end-
step_Padding,step_Padding+1:end-step_Padding);
end

%figure; imagesc((img_Step+img_Test2));colorbar;
%axis tight equal

% Calculate the step height
img_Step = img_Data.*img_Step;
stepheight = abs(mean(img_Step(img_Step < 0)));
stepdev = std(img_Step(img_Step < 0));
disp(['Step Height: ' num2str(stepheight) ' um +/- ' num2str(stepdev) '
um'])
figure; hist(img_Step(img_Step ~= 0),1000)

%out(m+1,:) = [step_Padding stepheight stepdev];
%end

% Plots when using the padding loop
%figure;plot(out(:,1),out(:,2),'.r');
%figure;plot(out(:,1),out(:,3),'.r');

```