OPTICALLY PROJECTED LENGTH SCALE FOR USE IN PHOTOGRAMMETRY

by

Patrick A. Thewlis

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
Optical Science & Engineering

Charlotte

2020

Approved by:

Dr. Angela D. Davies-Allen

Dr. Edward P. Morse

Dr. Christopher J. Evans

Dr. Stuart T. Smith

ABSTRACT

PATRICK A. THEWLIS. Optically projected length scale for use in photogrammetry. (Under the direction of DR. ANGELA D. DAVIES-ALLEN)

Photogrammetry is a measurement technique where 3D coordinate data and feature information can be extracted from 2D photographs or images. Length scale objects are calibrated artifacts placed into a photogrammetry scene which have precisely defined dimensions and are used to scale coordinate data. Scale artifacts can be cumbersome to use due to their length - often multiple meters. In this research, a novel technique has been explored whereby a low cost and portable module projects a structured light pattern into a photogrammetry scene to provide scale. A holographic diffraction grating is used in the module to create and project an 11×11 square grid of laser spots into the field, forming a structured light pattern. This grid is then duplicated using a pellicle beamsplitter and fold mirror such that two patterns are projected into the field.

Two novel algorithms were constructed to realize the optically projected length scale with the module. The first calibrates the pointing directions of beams in each structured light pattern from the module and finds the separation distance between the two projection origins. Photogrammetry is combined with a transformation fitting algorithm to solve for the projection source's position and pose within a space. Once these parameters are known, the dataset is transformed such that the projection origin lies at a global coordinate system origin, allowing classification of beam pointing directions in spherical coordinates. This calibration process requires a length scale artifact which is readily detected in a photogrammetry scene and whose length is precisely defined. A photogrammetry scale artifact was designed and fabricated for use in the calibration phase. Its length was characterized to 1677.66 ± 0.01 mm on a coordinate measuring machine.

The second algorithm uses the calibration information to trace a projected pattern in a scene back to its projection origin for a single photogrammetry measurement. By projecting two calibrated patterns into a photogrammetry scene and solving the distance between the two unscaled pattern origins, this unscaled distance can then be compared to the known calibrated separation length to obtain the scale factor by which all photogrammetry coordinates can be scaled.

A prototype dual pattern projection module was designed, fabricated, and tested. Using a novel method, the spherical coordinate pointing directions for the 121 beam pattern were calibrated and the projection origin separation distance in the module estimated as 55.16 ± 0.05 mm (coverage factor k=1). Following calibration, the module's performance was experimentally validated in two measurement trials of 20 measurements each to fractional uncertainty of parts per thousand. Monte Carlo simulations estimated the module's measurement uncertainty in its length scale to 3.4 parts per thousand, which agreed with the experimental results. Monte Carlo simulations were also used to explore design parameters which limit module performance. Additional simulation data shows the viability of redesigned modules with fractional uncertainty of parts per hundred thousand or beyond, supporting the conclusion that optical pattern modules could offer a portable alternative to traditional scale artifacts.

DEDICATION

To my grandmother, my parents, and Jessica.

ACKNOWLEDGEMENTS

A number of individuals have provided guidance, assistance, and advice on this research. First and foremost, I would like to thank my advisor Dr. Angela Davies-Allen for not only kindly introducing me to this project years ago, but also for the steady guidance she has provided over its duration. She has done an admirable job of keeping the research on track and guiding the work toward its conclusion.

I am also grateful to my committee members, who gave their time and insight to assist with this project. Dr. Edward Morse provided valuable input on the development of the algorithm. Dr. Christopher Evans' coursework and guidance has taught me much about precision metrology. Dr. Stuart Smith has offered advice and help on a number of optics related projects over the last half decade.

I would also like to acknowledge the UNC Charlotte Center for Precision Metrology, whose affiliate members provided valuable feedback on the development of this technology. There were a number of other CPM and engineering faculty, staff, and students, whose contributions to this project were deeply appreciated. In particular... Dr. Jimmie Miller provided helpful suggestions on the development of the fabricated scale artifact, Geoff Heacock fabricated the target blocks used on the artifact, and Yue Peng measured the artifact on the CMM. Thank you as well to Jennifer Chastain of CPM. In addition, I'd like to acknowledge Scott Williams for providing assistance over the years, as well as Dr. Tsing-Hua Her, from whose lab I learned much about hands-on optical science during the last years of my undergraduate and first few years of graduate education. Lastly, I graciously appreciate the help that fellow graduate students offered on the development of this project.

Funding for this research was provided by the UNC Charlotte Center for Precision Metrology. Additional support was provided by the UNC Charlotte Graduate Assistantship Support Plan (GASP) and various teaching assistantships.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

1D One Dimensional

2D Two Dimensional

3D Three Dimensional

ADC Analog to Digital Converter

CCD Charged Coupled Device

CMM Coordinate Measuring Machine

CNC Computer Numerical Control

CRP Close Range Photogrammetry

CTE Coefficient of Thermal Expansion

DOE Diffractive Optical Element

DOF Degrees of Freedom

FEA Finite Element Analysis

FFT Fast Fourier Transform

FWHM Full Width Half Maximum

GUM Guide to Uncertainty in Measurement

HeNe Helium Neon

HTM Homogeneous Transform Matrix

ID Inner Diameter

JPL Jet Propulsion Laboratory

OD Outer Diameter

RAD Ringed Automatically Detected

RMS Root Mean Square

ROV Remotely Operated Vehicle

SD Standard Deviation

SfM Structure From Motion

SIFT Scale Invariant Feature Transform

SLR Single Lens Reflex

SNR Signal-to-Noise Ratio

TEA Thin Element Approximation

TEC Thermoelectric Cooler

UAV Unmanned Aerial Vehicle

UTS Ultimate Tensile Strength

CHAPTER 1: INTRODUCTION

1.1 Project Motivation & Goals

Photogrammetry is a measurement technique where 3D coordinate data and feature information can be extracted from 2D photographs or images. In photogrammetry, a scene is imaged from different perspectives, and common control points, or targets, are identified in each image. A triangulation calculation, known as a bundle adjustment, allows for the 3D coordinates of control points to be extracted from 2D images. Photogrammetry allows for rapid, non-contact metrology to be performed on complex objects which can be difficult to measure by other means. These objects can range in size from small artifacts measured during an archaeological expedition, to large objects like parabolic antennas or aerospace wing and fuselage components during assembly. Photogrammetry has been used to measure bridge deflection during vehicular travel [1], and blade deflection of power generation scale wind turbines [2]. Recently NASA has incorporated photogrammetric targets on rocket fuel tanks to monitor separation trajectory during space vehicle launches [3].

Photogrammetry requires a calibrated length scale artifact to be present in the scene of the measurement to provide scale to obtained coordinates or point cloud data. These artifacts are typically on the order of the size of the object to be measured. Commercially available artifacts for use in measuring small objects up to larger manufacturing sized objects can range in price from hundreds of dollars into the thousands, and can provide fractional uncertainty on the order of parts in 10^4 to 10^6 [4, 5].

In addition to cost, when measuring larger objects, such as industrial sized machinery or parts, these scale artifacts can be large and cumbersome to work with in the

Figure 1.1: Photogrammetric measurement of a Siemens AG industrial scale power plant cast water valve housing. Black coded targets are visible on the surface, and a calibrated yellow scale object in the scene is used to provide scale to solved coordinates. *Image Credit: Linearis3D [6]*

field, as can be seen in Figure 1.1. As shown, photogrammetry is being performed on a steam turbine valve housing, some of which have a measurement volume of up to $5 \times 5 \times 3$ meters [6]. In many instances, a smaller scale reference could be beneficial in a working environment where portability is key, even if the fractional uncertainty of the scale reference is greater than that of traditional artifacts. There are other situations, such as underwater computer vision-based measurements or robotic planetary rovers where scale is desired, but a scale object cannot physically be placed into the scene.

This dissertation explores the feasibility of a small, low cost optically projected length scale. In this method, two structured light patterns are projected into the scene of a photogrammetry measurement, and an algorithm uses the unscaled photogrammetry delivered coordinates of each pattern to solve for the module's two pattern projection origins. The unscaled distance between projection origins is calculated and compared to a known calibrated separation length. This ratio establishes a scale factor by which all photogrammetry delivered coordinates can be scaled.

Similar work in this field was accomplished by Zheng in 2014, who used a diffractive grating and laser to project a square grid pattern into the field, utilizing the pattern as a degree of freedom sensor [7], though the preliminary algorithm was unable to obtain true 6 degree of freedom sensing. This problem was partially remedied by including a virtual image of the camera pattern along with standard photogrammetry pattern images into the software, allowing the photogrammetry software to solve the location and pose of the module during the course of its bundle adjustment. One of the problems Zheng experienced was in the calibration of diffractive optical patterns, as uncertainty in the pointing direction of beams adds to uncertainty in the solved position and pose. Zheng used a rotary table to calibrate on axis beam angles, but was unable to directly calibrate off axis beams, and relied instead upon diffraction models to provide calibration corrections.

While other methods like diffractometry were created to characterize ruled gratings, there is a lack of research in the area of calibrating a full pattern created by a holographic grating. Holographic gratings are being fabricated with increasingly complex topography and periodicity, enabling creation of complex beam patterns with arbitrary beam shapes, angles, etc. The angular properties of generated patterns can be difficult to assess, however, especially when they are not the last element in an optical system.

In this research, a simulation environment is developed which can model square grid beam patterns of arbitrary angle and number projected from arbitrary module geometries. That generated data is used to assist in the creation and fabrication of two novel algorithms. The first uses pattern calibration data to fully solve the 6 degrees of freedom of a pattern's projection origin. In creating a module that projects two grids, the distance between projection origins can function as a source of scale in photogrammetry measurements. At the time of writing this dissertation, no published works have been found in the area of an optically projected length scale. The second novel algorithm allows for the full calibration of the beam pointing directions of an arbitrary pattern generated by a diffractive optical element, and solves for the calibrated separation distance between two projected pattern origins.

Simulation data will be used to drive development and design of a low cost, portable prototype projection module. After fabrication, the module will be calibrated with the novel method, and then its performance will be assessed via a series of measurement experiments. Finally, Monte Carlo simulations will be used to examine the impact of module design parameters, such as full angle and beam number, on the uncertainty associated with the optically projected length scale. Using this information, an improved module design offering reduced fractional uncertainty will be proposed. Some of the practical limitations of this system will be discussed and suggestions given for future work and areas of improvement.

1.2 Dissertation Overview

This dissertation is divided into four main sections, as outlined in Figure 1.2, with each representing a different developmental phase of the research. An overview of this dissertation is as follows...

Chapter 2 discusses the theory of operation behind the optically projected length scale module. Two methods are created to generate simulated projection pattern data. Using the simulated data, an algorithm is developed which drives the operation of the optical length scale module system. The algorithm and simulation are used to evaluate module designs. A prototype design is evaluated, selected, and then fabricated.

In Chapter 3, the background and theory of diffractive optical elements is discussed, and a novel method is created to calibrate the beam patterns created by fanout diffractive elements. A photogrammetry specific length scale artifact required for module calibration is designed, fabricated, and characterized. Three module calibrations are performed and a Monte Carlo simulation is used to evaluate the uncertainty in the novel calibration method.

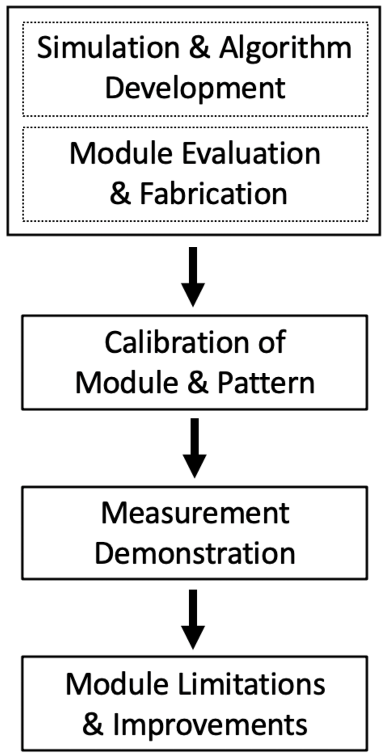


Figure 1.2: Dissertation layout.

In Chapter 4, calibration data of the module is utilized to drive the algorithm. A measurement demonstration is performed with the optically projected length scale module. The calibration process of cameras used for photogrammetry is discussed, and a calibration is performed on each camera. The experimental setup and procedure is described, and two separate measurement trials are completed. Possible biases in the measurement results are examined, and a Monte Carlo simulation is used to evaluate the uncertainty of measurements made by the module.

In Chapter 5, four key module design and measurement parameters are identified which have a large impact upon the performance and measurement uncertainty of the module. Each of these are evaluated by simulation to see where improvements can be made to the current module design. Other system and process improvements are discussed which could decrease the data post-processing burden on the user. An improved module is designed, and its performance and uncertainty estimated by Monte Carlo simulation. Optomechanical error sources within the module are examined and their impact on the module's uncertainty in measurement is evaluated. Finally, a drift test is performed on the module to examine how the module is impacted by temperature changes in the lab environment.

Chapter 6 summarizes the major conclusions from this research, and provides a final evaluation of module performance. The optically projected length scale module concept is compared to traditional length scale artifacts and its potential uses are evaluated.

In Chapter 7, advice and potential criticisms for future research into optically projected length scale modules is offered, including processing constraints and applications of projection modules. A few novel applications are highlighted where an optically projected scale module may prove useful.

CHAPTER 2: MODULE DESIGN & THEORY OF OPERATION

In this chapter, a brief overview of photogrammetry is given, some of the factors which influence its performance are discussed, and its capabilities in various applications are described. The theory which drives the operation of the optically projected length scale module is then discussed. The simulation environments used to generate point data of various module configurations on a variety of surfaces are described. The generated point data is used to assist in development of an algorithm which solves for projection location given a set of calibrated grid pointing directions for a beam pattern and a set of pattern point coordinates delivered from photogrammetry. After describing the algorithm, two module designs are evaluated. A final prototype module design is selected after weighing the benefits of each. The design and construction of the prototype module is then described.

2.1 Photogrammetry Overview

Photogrammetry is described by the American Society for Photogrammetry and Remote Sensing as "the art, science, and technology of obtaining reliable information about physical objects and the environment through processes of recording, measuring and interpreting photographic images and patterns of recorded radiant electromagnetic energy and other phenomena" [8]. In this research, photogrammetry is specifically referred to as the process of extracting three dimensional (3D) information from two dimensional (2D) data sources.

The roots of photogrammetry can in some ways be traced back to Leonardo da Vinci's work in the late 1400s on linear perspective. Da Vinci's work, and that of other painters at the time, drove the conversation of the mapping of 3D points, lines, angles, and other surfaces onto a 2D image. French architect and mathematician Desargues created the projective geometry foundation which underlies photogrammetry [9]. In 1759, Johan Heinrich Lambert mathematically described the geometrical properties of a perspective image and used space resection to mathematically describe the point in space from which the image was made [10, 11]. As defined by Moffit and Mikhail, space resection is the name given to the process in which the spatial position and orientation of a photograph is determined based upon the photogrammetric measurements of the images of ground control points appearing in the photograph [12]. From French inventor Daguerre came the first photograph, or the 'daguerrotype', and with photography came photogrammetry.

The direct link between photogrammetry and projective geometry was further expanded by Sturm and Hauck in 1883 [13], but it was German scientist Finsterwalder's expansion upon Sturm's work over a number of decades which provided the first analytical solution and vector description of multi-image photogrammetry [14, 15]. Analytical solutions were not immediately feasible owing to their significant computational burden; early photogrammetry was mainly performed on analog plotters. However, the proliferation of computers beginning in the 1950s enabled the rapid calculation of purely analytical solutions. The capabilities of photogrammetry were further expanded upon by the advent of digital camera technology in the 1990s. Two types of photogrammetry are common - aerial photogrammetry, and close range photogrammetry (CRP). Aerial photogrammetry refers to photogrammetry conducted at altitude from aircraft, unmanned aerial vehicles (UAV), etc., at distances greater than 300 meters. In comparison, close range photogrammetry is typically defined as photogrammetry taking place on a measurement scale of less than 300 meters. Though traditionally completed from the ground, recent improvements to UAVs and drones has allowed for easier photogrammetric measurement of larger objects, such as parabolic antennas or large structures.

In its earliest iterations, aerial photogrammetry was used almost exclusively for mapping purposes, first from balloons and kites, then later in aircraft. This was greatly expanded upon during the first and second world wars, and pushed the development of specialized photogrammetry equipment. With the advent of sophisticated drones and UAVs, in combination with advances to mobile processing, robotics, and sensor technology, centimeter-level resolution is currently possible with minimal investment [16]. Tens of square miles can be accurately mapped in less than an hour.

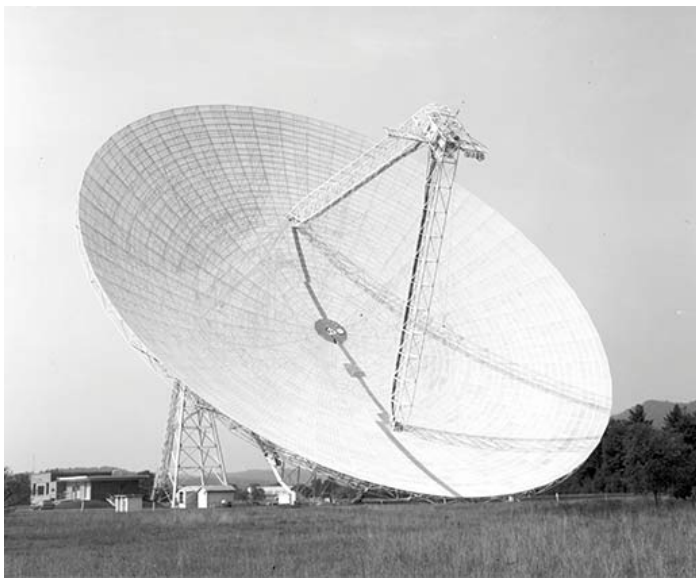


Figure 2.1: National Radio Astronomical Observatory 300 ft radio telescope, Green Bank, West Virginia. *Image Credit: NRAO [17]*

Following development of aerial photogrammetry, metrologists began to realize the role close range photogrammetry could play in non-contact measurement of a variety of surfaces and parts. Significant to the development of terrestrial photogrammetry was the drive to measure the form and tolerances of various antennas, including parabolic antennas. To emphasize this point, Ruze's antenna tolerance work tells us that a $\lambda/20$ RMS error in the form of a parabolic antenna leads to a loss in efficiency of approximately 33% [18]. While high accuracy measurement can be attained with other methods (theodolite, laser tracker, etc.), photogrammetry could perform non-contact metrology on large objects in a much shorter time span. Photogrammetry performed in 1962 on the National Radio Astronomy Observatory's 85 ft and 300 ft radio telescopes, the latter shown in Figure 2.1, was able to verify the surface form error to approximately 1.5 mm RMS, and 1 cm RMS, respectively [19].

2.1.1 Targets & Target Detection

Early photogrammetry relied upon analog user selection of features in different images to use as control points, or targets. Modern feature detection and computational methods have enabled a new version of photogrammetry known as structure from motion (SfM), whereby a 2D image sequence or video can allow for 3D coordinates to be found based upon feature detection. This can allow for rapid development of point cloud data and texture mapping to generated 3D models, but typically is less accurate than target based photogrammetry. Results can vary depending on ambient lighting, as well as surface feature density and quality [20].

As computer technology developed, and computer-vision based metrology methods became more sophisticated, algorithms were written to perform automated feature or target detection. Consistent selection of a single spot allowed for defined measurement. Because of their radially symmetric form, circular targets perform well in a photogrammetry environment. The target's center is invariant to rotation and invariant to scale over a wide range. Target physical dimensions should be selected to give at least 5 pixels in diameter in the image [21]. When too few pixels make up the target, the rough contour can reduce the sub-pixel precision of the centroid location [22, 23].

The basic circular target typically consists of a single black or white spot superimposed upon a white or black background, respectively. Circular targets are easily and reliably centroided by center-of-mass computations, and the high contrast boundary between the target and its background facilitates edge detection in thresholding algorithms [24]. As image lighting degrades, however, so does the contrast at the target and background boundary.

Using retroreflective material as a target surface was proposed in 1984 by Brown, who sought to increase target contrast in typical working conditions [25]. Retroreflective film is typically constructed from small microspheres ($\varnothing < 100\ \mu$m) embedded into an adhesive layer. While the surface is fragile and the material expensive, targets of any shape can be created. When used in conjunction with a flash system, camera exposure times can be reduced such that the background lighting of the image is essentially removed. While black and white targets can perform similarly to retroreflective targets under ideal conditions and lighting [26], a strobe in combination with retroreflective targets provides a black image with brightly illuminated target faces. The resulting high SNR allows for reliable and precise centroiding in any variety of working or lighting conditions.

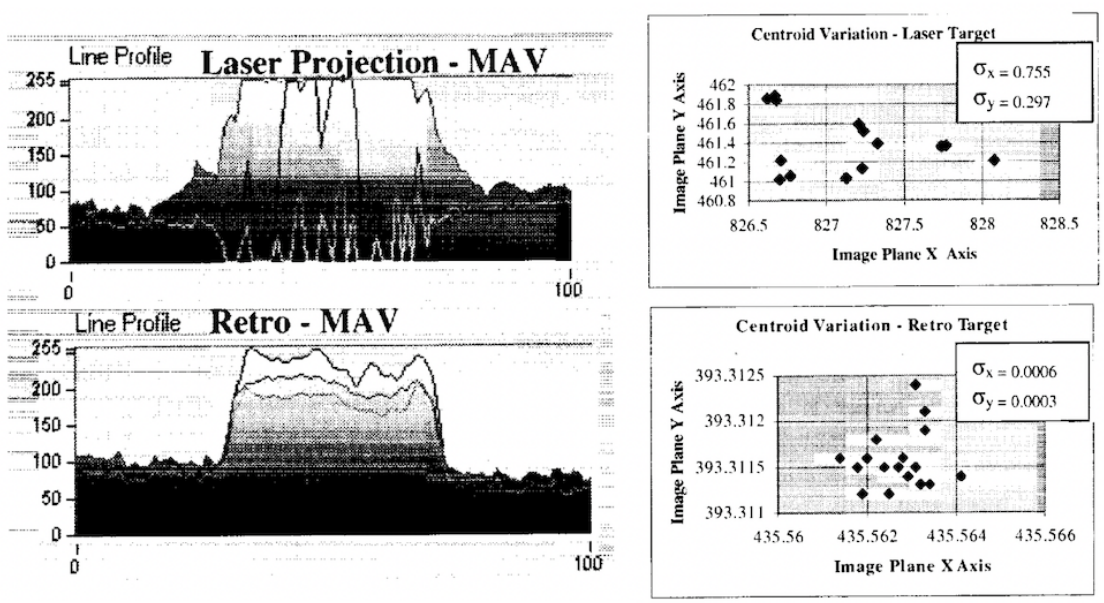


Figure 2.2: Variations in intensity arising from speckle can induce significant variation in the centroid of a laser projected spot compared to a standard retroreflective target. *Data & Images Courtesy: Jones & Pappa, NASA Langley Research Center [27].*

Laser targets have been studied, and offer a variety of advantages and disadvantages compared to traditional or retroreflective target surfaces. While the intensity yields a high SNR, and selective wavelength filtering can assist in centroiding and processing, the speckle arising from the coherent nature of the light yields variations in the spot intensity. The speckle pattern varies with viewing location, and results in a centroid which appears to change location with change in perspective. As shown by Jones and Pappa (2002), the stable cross sectional intensity of a retroreflective target yields orders of magnitude less centroid variation [27]. This variation is discussed in more detail in Section 5.1.1 along with ways in which centroid stability can be improved.

Coded targets are targets which have a unique identifier built in. This can be in the form of broken rings or circles, bar codes, patterns, structured dots, or by other means, as shown in Figure 2.3. Coded targets were developed in the late 1980s and early 1990s alongside of commercial photogrammetry systems to facilitate automatic detection and referencing during processing [28–30]. Ringed automatically detected (RAD) targets used in this research are similar to the targets shown in the upper right of Figure 2.3, and function as control points to facilitate automated processing of non-pattern targets.

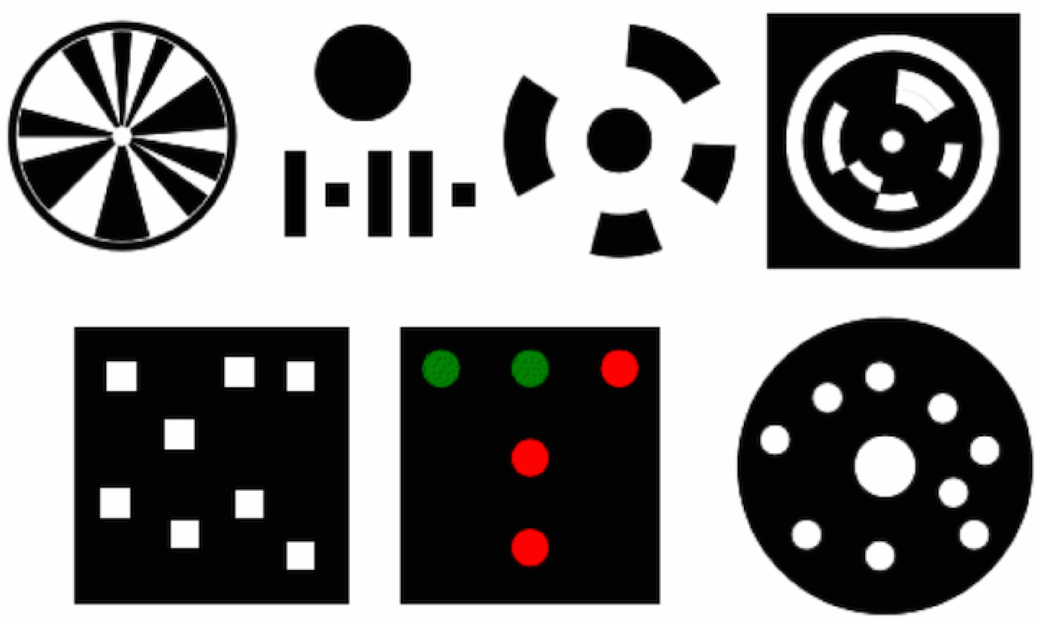


Figure 2.3: Coded targets. *Image Credit: Close-Range Photogrammetry and 3D Imaging - Luhmann et al. [21]*

2.1.2 System Performance

There are a variety of factors which influence the overall performance and precision of photogrammetric measurements. Those factors include the geometry of the photogrammetric system, the number of images taken, the resolution of the sensor or recording format, and the size of the object to be measured. Additionally, the camera, its calibration, and the length scale utilized can play a significant role in the outcome.

The geometry of the camera network plays a large role in the error associated with photogrammetric point measurement. Because photogrammetry is based upon triangulation, excessively shallow or steep convergence angles can result in increased error, as shown in Figure 2.4. The 3D placement of camera stations and their orientations is referred to as network design. Fraser's seminal work on network design discusses the establishment of a datum, the configuration (number of points, number of stations, camera position geometry, etc.), weighting, and densification of images. These parameters can be optimized to reduce error [31]. In general, a convergence angle of 70° to 110° provides an adequate result. As discussed in Chapter 4, this research utilizes four camera locations, or stations, with an average convergence angle of 70°.

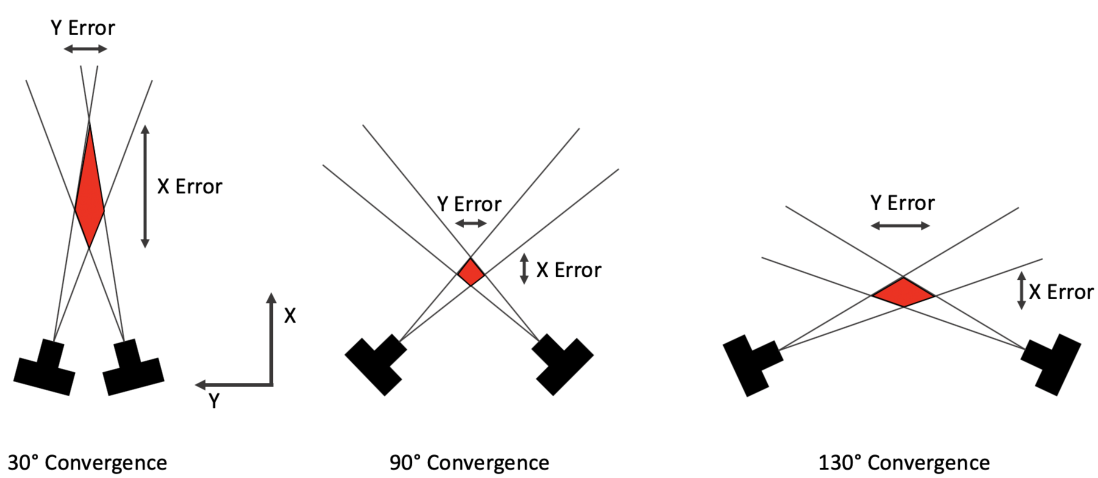


Figure 2.4: Impact of convergence angle on triangulation error.

The number of images has an impact on the overall uncertainty associated with a solved point location. Error typically scales as the square root of the number of images per camera station [32]. More images from more camera stations will reduce error. In a highly automated system, adding additional images is a simple way to decrease the error in a photogrammetric measurement.

Photogrammetry accuracy also varies depending on the size of the object to be measured. PhotoModeler® specifies that a high quality digital camera with automated targets can expect accuracy on the order of approximately 3 parts in 10^5 per meter. On a 15 m object, this is error on the order of 0.5 mm. On a 30 m object, the error doubles to 1 mm.

Additionally, the quality and resolution of the camera sensor has a direct impact upon the precision of photogrammetry measurements. Sub-pixel resolution is possible in commercial software packages. A higher sensor resolution in a high quality camera system generally leads to better results.

For accurate results, it is critical that the cameras be accurately calibrated such that the intrinsic and extrinsic properties of the camera are known, and can be used to correct flaws in the images. The theory and process of camera calibration is described in detail in Section 4.1. In the earliest days of aerial photogrammetry, metrologists utilized whatever equipment was available. It became apparent, however, that specialized camera equipment could dramatically improve the results of a photogrammetry measurement.

While any camera can be used for photogrammetry, literature differentiates between *metric* and *non-metric* cameras. Metric cameras are those which have defined and stable intrinsic parameters, which include focal length, principal point, and skew coefficient. Intrinsic parameters map from 3D camera coordinate space into 2D image coordinates. In a typical consumer grade off-the-shelf camera, the focal length is variable, lenses may be changed, etc., resulting in a change to the intrinsic parameters.

Figure 2.5: Geodetic Systems Inc. INCA3 Metric Camera. *Image Credit: GSI [33]*

A metric camera is a simple and robust camera specifically made for measurement. The lens often cannot be changed, is low distortion, and has a fixed focal length. While aerial photogrammetry typically uses wide angle lenses to facilitate data collection, a long focal length lens typically offers reduced distortion and is optimal for close range photogrammetry [34]. The optical axis should be perpendicular to the image sensor. The shutter is often a leaf shutter rather than focal plane shutter, which can reduce distortion and vibration during imaging [35]. Early film metric cameras utilized Reseau plates to impart a fiducial upon the image and also served as a 'vacuum plate' to ensure the film was flat, thereby reducing distortion. Even in the case of modern digital cameras, sensor flatness can serve as an error contributor. Metric cameras are often calibrated to account for and reduce this error [36]. While modern methods of calibration have improved the results with non-metric cameras [37], the intrinsic parameter stability offered by a metric camera will yield the best results.

2.1.3 Applications & Capabilities

The non-contact nature of photogrammetry makes it an ideal measurement tool for complicated structures or situations where traditional metrology may not be possible or practical. In this section, a few unique applications are described to help the reader understand where and how photogrammetry may be used, and the measurement precision of photogrammetry systems is discussed.

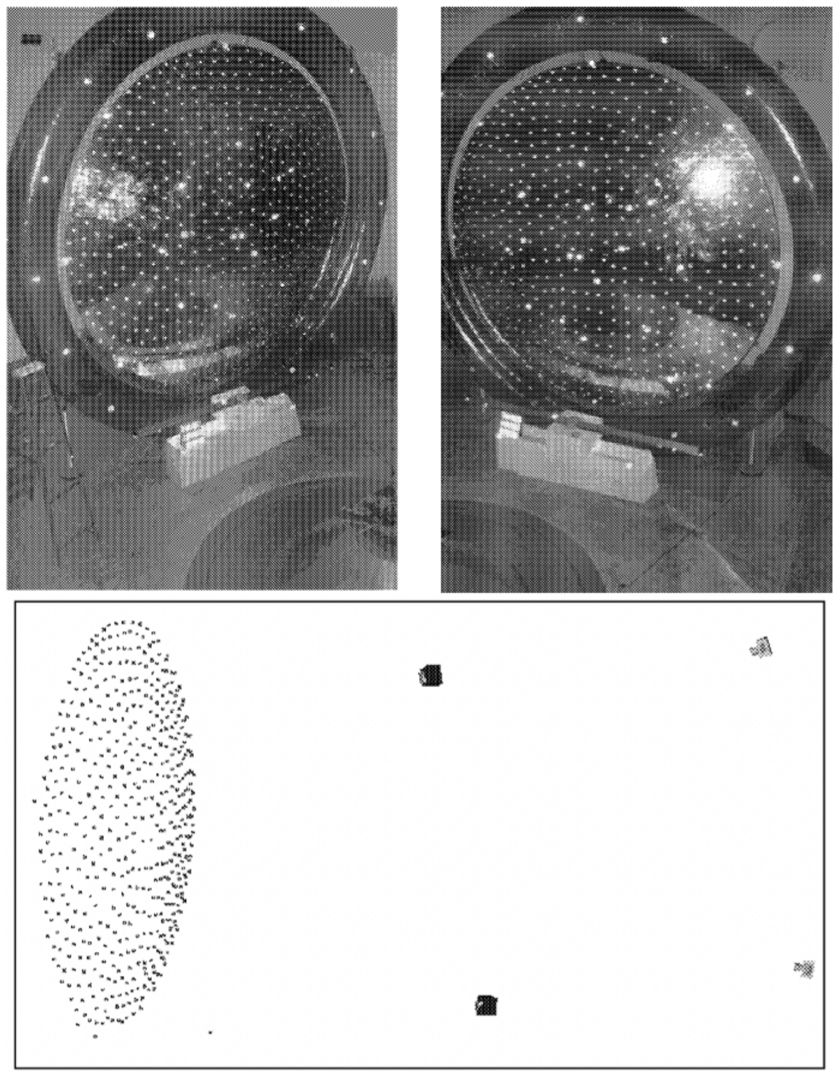


Figure 2.6: NASA measurement of 5 m inflatable space antenna with consumer grade 2.1 megapixel consumer camera with four total images. Form validated to $\approx \pm$ 0.5 mm. *Image Credit: NASA [38]*

As described in Section 2.1, only of the earliest uses of close range photogrammetry was for form measurement of parabolic antennas, which can range in size from a few meters to over 100 meters. Fraser utilized 680 retroreflective targets placed upon a 2.9 m deployable space antenna in a 10 cm grid, with a mean standard target error of 8 μm [39]. Similar measurements were performed on a composite antenna mold to verify form, with a mean standard error of 12 μm. Even a simplistic measurement performed with a 2.1 megapixel camera and just four images allowed NASA researchers to validate form of a 5 m inflatable soft space antenna with 500 targets to approximately 0.020 inches in plane and 0.050 inches out of plane [38].

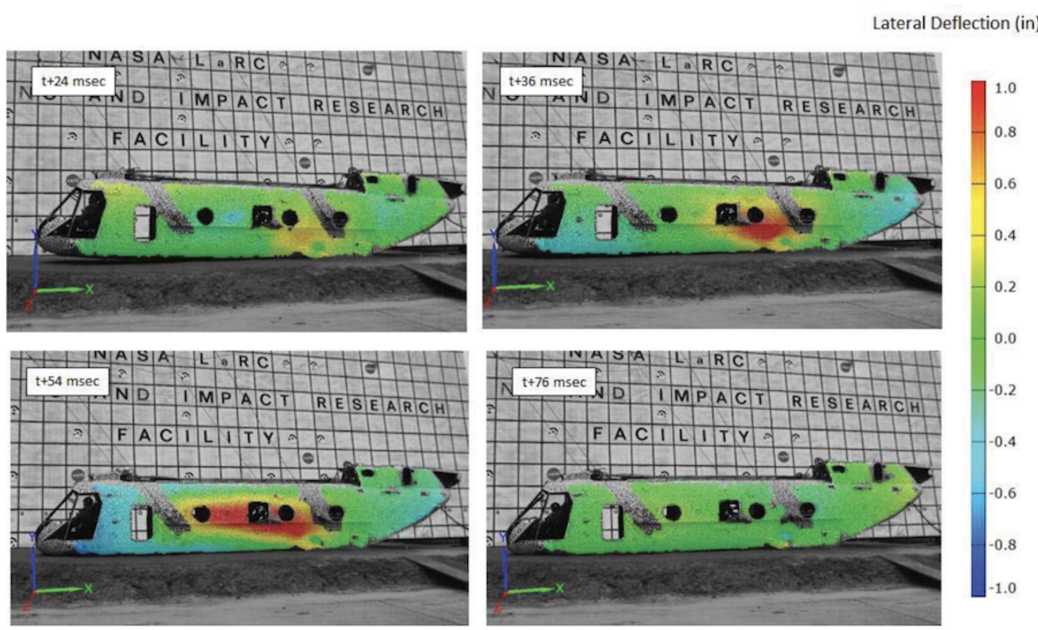


Figure 2.7: Structural deflection of airframe fuselage during crash impact testing. *Image Credit: NASA [40]*

Another field which has greatly benefited from photogrammetric measurement is that of structural dynamics. While finite element analysis (FEA) simulation offers full-field results over an entire structure, conventional measurement techniques like accelerometers and strain gauges can only provide measurements at a few discrete locations over a narrow range [41]. Non-contact methods like photogrammetry therefore offer distributed sensing capabilities to validate FEA without adding mass or affecting the dynamic motion of the structure via added sensors. Fraser was able to characterize the interior dimensions of a Bushmaster troop carrier armored truck before and after explosive ordinance detonation under the vehicle [42].

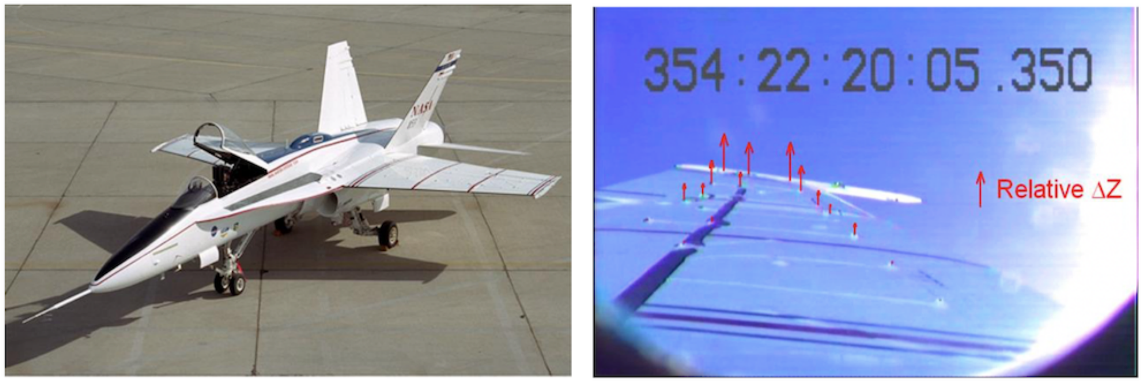


Figure 2.8: Single camera photogrammetry measurement of wing of F/A-18 test aircraft during flight. NASA Dryden Flight Research Center. *Image Credit: Burner et al. [43]*

Littell used photogrammetry in conjunction with high speed cameras to measure structural deflection during crash tests of aircraft fuselages [40]. An additional unique application of photogrammetry has been in measuring the in-motion deflection of rotor blades on wind turbines, helicopters, etc. Ozbek et al. measured wing tip deflection of 80 m diameter 2.5 megawatt wind turbines to an accuracy of ± 25 mm at a distance of 220 m [2]. Barrows et al. measured the rotating rotor blade deformation of a UH-60A helicopter during motion in a wind tunnel to simulate loading [44]. Burner et al. used single-camera photogrammetry to measure vertical wing deflection of an F/A-18 aircraft during flight at NASA Dryden Flight Research Center [43].

Other novel uses include civil engineering applications, such as railroad or roadway bridge deflection during loading [45]. Accuracy is comparable to traditional surveying, while reducing workloads by up to 50% [1].

The development of coded targets and sophisticated metric cameras, along with careful network optimization, allowed for commercial photogrammetry systems to approach uncertainty of one part in one million in the early 1990s [46]. PhotoModeler® offers performance on the order 1 part in 30,000 with traditional DSLR cameras. A survey of typical purpose-built commercial photogrammetry systems shows system performance on the order of less than 15 μm per meter, over a range of tens to hundreds of meters. As a non-contact measurement technique, photogrammetry has matured into a versatile system allowing for precision measurement in a variety of situations.

2.2 Theory of Module Operation

Traditional scale artifacts are often constructed from carbon fiber reinforced polymer, aluminum, or other speciality materials such as Invar. These artifacts have two more more high contrast targets, such as RAD coded targets or a retroreflective marking, with the latter allowing for a higher signal-to-noise ratio (SNR) for improved detection in typical working conditions when used in conjunction with a flash illumination system. The distance between targets is calibrated and the artifact is placed within the scene, shown in Figure 2.9, to provide a sense of scale to the measurement. Low fractional uncertainty, defined as $\delta L/L$, or the error in the calibrated distance divided by the distance between targets, is desirable in order to decrease uncertainty in the scaled coordinates.

The optically projected length scale concept, also shown in Figure 2.9, differs from traditional artifacts in that its performance comes from using a large number of sampled optical pattern points to precisely solve each pattern projection origin. This drives down fractional uncertainty in the length scale, δL, rather than relying on a large separation L between the two targets on the scale object, allowing for a smaller length scale to be used while still achieving satisfactory performance.

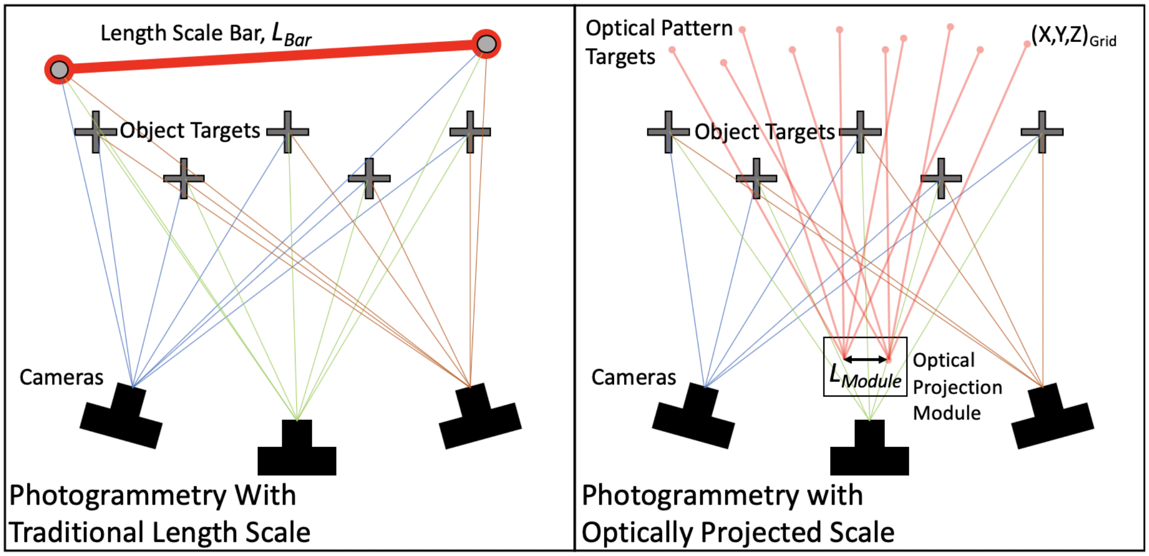


Figure 2.9: A comparison of scale methods for photogrammetry. A calibrated length scale bar is placed within the scene in traditional photogrammetry. Using an optical method, the length scale can be derived from a projected grid pattern.

The method requires two or more structured light patterns to be projected into the photogrammetry scene. In this case, a diode laser source and a diffractive optical element are used to generate a square $n \times n$ dot matrix grid, where n is defined as the number of beam spots per side of a square grid pattern. The prototype, detailed in Section 2.6, uses a single diode source and diffractive element used in conjunction with a beamsplitter and fold mirrors to duplicate the pattern. Multiple gratings could be utilized with a single source, as per the revised module discussed in Section 5.2, or multiple sources each with their own separate gratings with a larger separation.

During the bundle adjustment process, unscaled coordinates are delivered not only for the coded targets in the scene, but also for the optical grid points, $(X, Y, Z)_{Grid}$. Using pattern coordinates and well defined pointing directions for each beam in the pattern, an algorithm, described in Section 2.4, solves for the unscaled projection origin and pose $(X, Y, Z, A, B, C)_{Origin}$ of the two projection sources. As each projected grid can be traced back to its origin in unscaled photogrammetry space, the unscaled separation length, or length scale L, can be calculated between projection origins.

By comparing the unscaled photogrammetry derived length between the projection origins to the known calibrated length, a scale factor is established by which all other photogrammetry delivered coordinates can be scaled.

2.3 Simulation Development

Development of a proper simulation environment was a critical first step of this research. The goal was to pursue simulation-driven project development, where ideas were first modeled, and those models were used as justification for a process, design, or purchasing decision. Algorithm development required a way by which known test data could be generated. Having created this test data for a diffracted pattern grid, the data could be used to construct and evaluate a projection origin solving algorithm. Ultimately, two sources were relied upon for pattern data generation - MATLAB® and FRED®.

In a prior UNC Charlotte Center for Precision Metrology funded project, a student utilized a diffractive optical element as a pattern generator for a photogrammetry-based degree of freedom sensor for robotic arm applications [7]. This student's code was provided as a starting point in this research. While the algorithm and method has significantly diverged from the prior work, the optical pattern generation script is still in use today. This pattern generation code (*scatter.m & geometry.m*) can be found in Appendix A.7.

The code simulates a square grid with pattern dimensions $n \times n$ and full angle θ, projected some distance into the field as shown in Figure 2.10, and outputs the Cartesian coordinate position of each beam spot in an array. The code has been modified from its original form such that generated coordinate data is now in a right-handed coordinate system, with the pattern projected in the X direction about the YZ plane. The Cartesian coordinates are also converted into spherical coordinates for use by the algorithm.

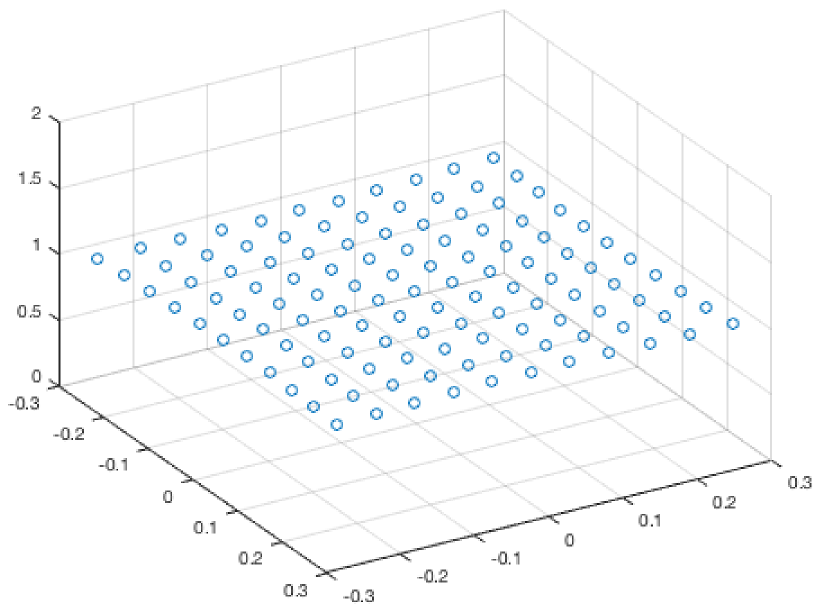


Figure 2.10: MATLAB® script written for far-field simulation of a diffracted optical pattern.

There are a handful of flaws in the MATLAB® pattern generation script. It is limited to projection from a (0,0,0) origin, is restricted to projection about a flat plane, and cannot allow for pose changes in the projected pattern. The origin and pose problems are resolved by using homogeneous transforms to perform translations and rotations as desired. The generated pattern can be shifted away from a (0,0,0) projection origin to simulate a second projection source, and its pose altered by a homogeneous transform.

In order to simulate a projected pattern on surfaces of complex topography, a license to FRED® was obtained. Photon Engineering provided a no-cost academic license of FRED® for the duration of the research [47]. FRED® is an optical ray tracing software that can simulate propagation of light from coherent or incoherent sources thorough an optomechanical system and provides various analytical capabilities.

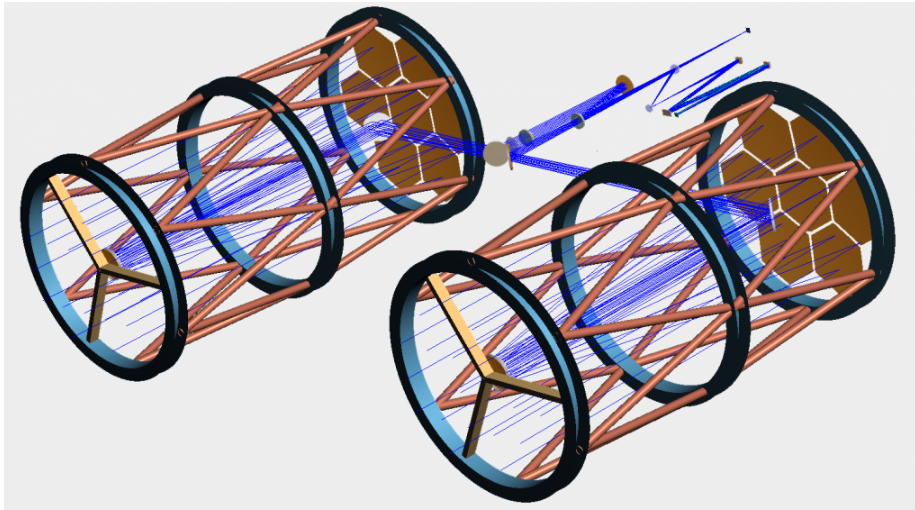


Figure 2.11: FRED® is an optical ray tracing software that allows for analysis of complex optical systems and sources. *Image Credit: Photon Engineering* [47]

In addition to being able to model and propagate light through complex systems, such as the binocular telescope shown in Figure 2.11, FRED® also allows for a variety of physical optics applications, including holography and diffractive optics. FRED® is used to simulate an $n \times n$ grating with a specific pattern full angle and propagate the light onto detector surfaces of various shapes. The incident X,Y,Z spot coordinate data is then exported back to MATLAB® for testing purposes in the algorithm.

Having acquired the ability to generate projected pattern data on a variety of surfaces and in a variety of orientations, development shifted to building the algorithm by which the system operates. Correct pattern data with known projection origins and module pose allows for evaluation of the algorithm's performance.

2.4 6 Degrees of Freedom Sensing Algorithm

A MATLAB® based algorithm was constructed to solve for a pattern's projection location, given inputs of a calibrated set of pointing directions for each beam, and an unscaled set of photogrammetry delivered 3D coordinates for a grid to be solved. This code can be found in Appendix A.1. In an effort to deliver 6 DOF sensing of robotic arms, Zheng et al. use a minimization algorithm to solve for the projection location of a single source module, given inputs of expected beam angles and a data set to be solved [48]. That algorithm only uses Cartesian translations in X,Y,Z, however, and cannot adjust for arbitrary pose changes in the beam pattern. Zheng's method is improved upon by constructing an algorithm which uses homogeneous transformations to view the photogrammetry delivered grid data set from different reference frames. The pattern is evaluated against calibration until a matched projection location and pose is found.

A homogeneous transform matrix (HTM) is a linear transformation commonly used in robotics applications where several bodies are present, each moving in a unique reference frame [49]. Here, the homogeneous transformation matrix T takes the form:

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.1}$$

where

$t_{x,y,z}$ is a 3×1 matrix describing the translation, and

r_{ij} is a 3×3 matrix describing the pose rotations to be applied.

A unique r_{ij} rotation matrix exists for rotation about each of the X, Y, and Z axes, as shown below in Equations 2.2-2.4:

$$Rot_X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.2}$$

$$Rot_Y = \begin{bmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.3}$$

$$Rot_Z = \begin{bmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.4}$$

where

Rot_X, Rot_Y, Rot_Z are the desired rotations about the X, Y, Z axes in radians.

An active (alibi) transformation alters the coordinates directly, about some fixed coordinate frame. In comparison, by multiplying coordinates by the the inverse T^{-1} homogeneous transformation, or the transpose, a passive (alias) transformation is performed, which alters not the coordinates themselves, but only the coordinate frame from which they are viewed [50]. In this manner, the data set is viewed from different perspectives. The observed pointing direction of each beam, relative to the central beam, is compared to that of its calibration until a best fit match is obtained.

A visualization of the algorithm is shown in Figure 2.12. In the figure, a pattern is shown projected on a surface. The laser pattern, shown in red, has a calibrated set of known pointing directions for each beam relative to the central beam. From the guessed projection origin, the spherical pointing directions of the pattern can be measured, as shown by the black lines. If incorrectly guessed, there will be a difference, $\delta\phi$ and $\delta\theta$, between the spherical coordinate pointing directions for a beam, and the calibrated pattern. In the event where the correct projection location is guessed, as shown on the right side of the figure, then no difference will exist, and the summation of all residuals of the pointing directions will be zero.

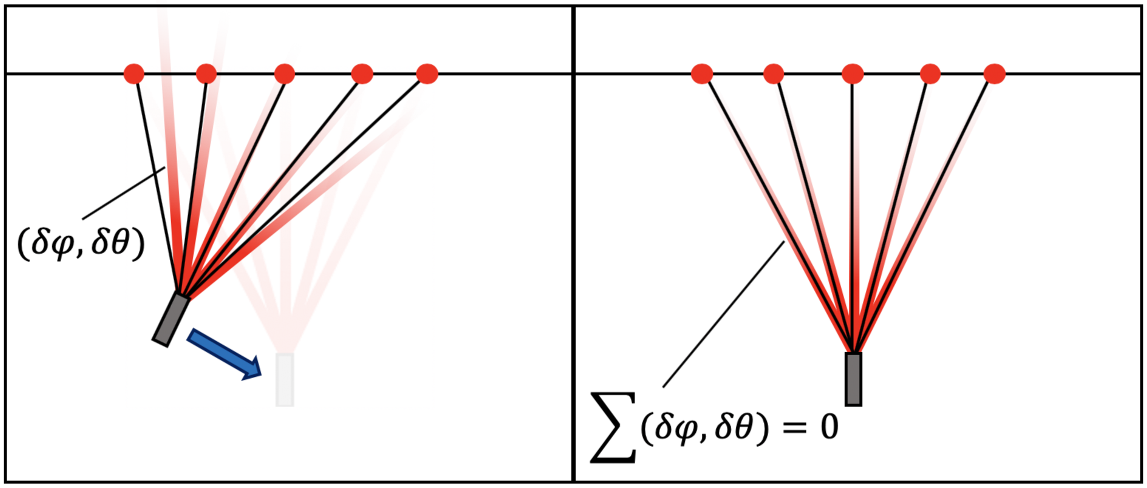


Figure 2.12: Algorithm visualization.

The unscaled 3D coordinates of a grid spot data set are obtained via photogrammetric measurement of a projected pattern. The coordinate set is sorted by beam number, and input into the MATLAB® algorithm. An initial guess of location and pose for the module, $(X, Y, Z, \alpha, \beta, \gamma)$ is provided, and the algorithm simultaneously evaluates the data set against the calibrated pointing directions for all beams as viewed from a given location and pose in space using the merit function shown below in Equation 2.5:

$$\sum_{i=1}^{N}(\phi_{Cal,i} - \phi_{Data,i})^2 + \sum_{i=1}^{N}(\theta_{Cal,i} - \theta_{Data,i})^2, \tag{2.5}$$

where

N is numeric ID of a specific beam number in the grid,

Cal represents the grid calibration set in radians, and

$Data$ represents the data set imported from the photogrammetry software.

A Nelder-Mead minimization delivers the final location of the pattern's projection origin [51]. Next, an additional minimization is performed with a second HTM to correctly 'clock' the grid and determine the pose. With the projection location and pose of both projected grids solved in unscaled coordinates, the distance L_{Photo} between them is calculated. The ratio of the unscaled length L_{Photo} to calibrated length L_{Cal} thus serves as the scale factor by which all photogrammetry coordinates can be scaled.

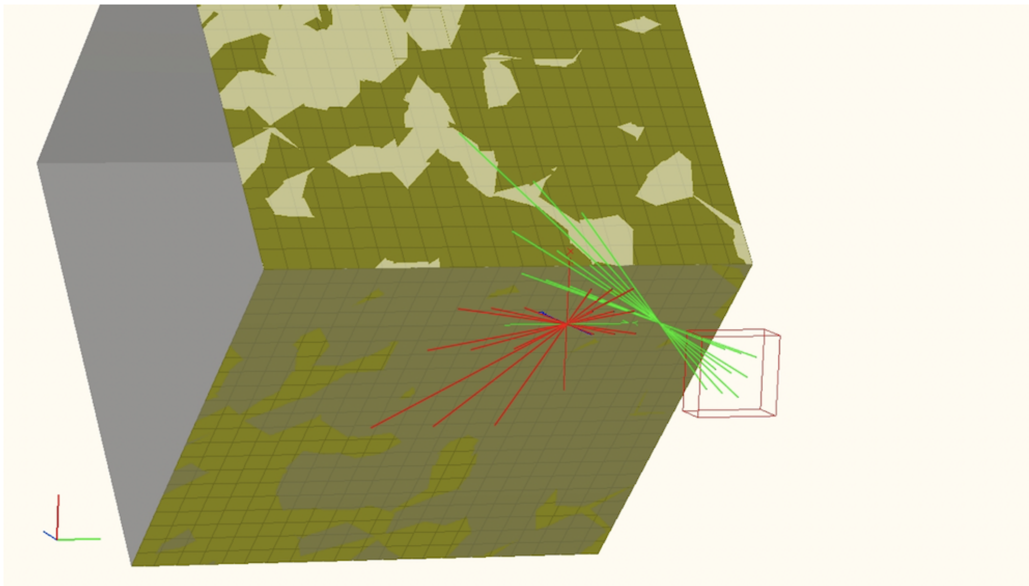


Figure 2.13: FRED optical ray tracing software. Source 1 is rolled $-15°$ about the Y axis. Source 2 is rolled $+15°$ about the X,Y, and Z axes. Patterns are projected upon the corner of the cube.

It deserves mention that each calibration is unique, and that one grid projection is not tied to the other. That is to say, multiple projection grids may be utilized with multiple sources, and there need not be pattern symmetry. This allows for significant design freedom in tailoring the number of sources, grid design, pointing directions etc. to any specific application, and is discussed in more detail in Chapter 5. All that is required for execution of this algorithm is a calibrated set of pointing directions for a specific projection source, the known separation distance between any number of projection sources, and the data sets to be evaluated.

Because the algorithm is driven by a set of calibrated pointing instructions for each beam, in theory, complex surface topography of an object or projection surface does not prevent the algorithm from functioning. So long as the position of each beam spot can be found in space, the spherical coordinates of the spot and the projection origin can be found can be found. The algorithm has been tested on a variety of simulated surfaces (spheres, cubes, parabolic surfaces) with a variety of projection source orientations, such as the example shown in Figure 2.13. The algorithm functioned correctly in each situation.

2.5 Module Design Evaluation

During this project's initial research proposal, the original idea behind the projection module was that vector representations of the length scale L could be realized from photogrammetry delivered coordinates, illustrated as the 'Proposed Method' in Figure 2.14. In this implementation, photogrammetry would deliver the spot coordinates of parallel beam pairs in the scene and vector projections from a beam spot to its parallel beam partner could yield a representation of the module length scale L, or the separation between projection origins. By evaluating a large number of representations of L, the uncertainty in the measured length scale could be decreased.

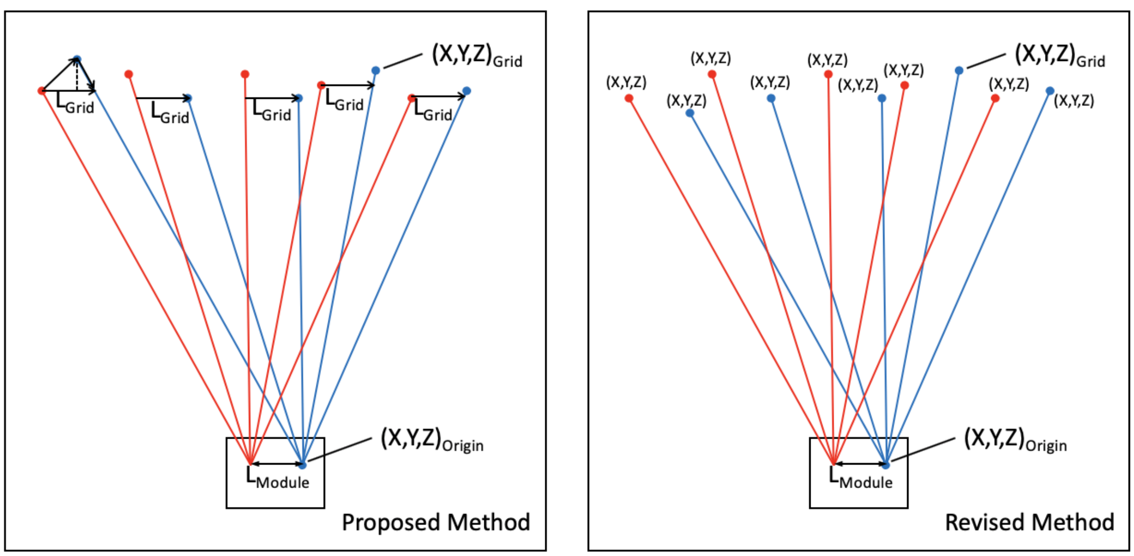


Figure 2.14: The original proposed optical length scale method used parallel beams to create many representations of the length scale, L, in the scene. The revised method uses many points to precisely triangulation the projection origin of each source, with the separation distance between origins serving as the scale.

After further work, a number of flaws were discovered in this method. Parallel alignment of each source is complicated in practice. A prototype module was fabricated using a steering mirror and beamsplitter to duplicate the pattern. While null fringes were attained on the surface of the beamsplitter and pellicle using a Zygo Verifire Fizeau interferometer, the module itself barely fit within the 4 inch operating aperture of the interferometer, as shown in Figure 2.15.

Additionally, the physical footprint of the module's optical board barely fit within the available space on the interferometer's multi-axis stage, making alignment difficult. Should a larger separation be desired, it would no doubt require an interferometer with a larger operating aperture. That said, even following alignment, there is still no guarantee that each beam in the grid will be parallel. Flaws during fabrication of a diffractive optical element, or in the optical components of the module, such as a beamsplitter or steering mirror, makes the likelihood of each beam pair being truly parallel rather unlikely. The further the module's projection length into the field, the more error is likely being introduced into each representation of L.

Figure 2.15: Null fringes on a prototype module following alignment on a Zygo Verifire™ Fizeau interferometer.

Following acknowledgement of these shortcomings, an alternative method was devised which fundamentally changed the method by which the module was designed to operate. Rather than rely upon beam pairs to generate a statistically large number of representations of the length scale, performance would instead come from the statistically large number of beam spots used to establish a single projection intersection point. By combining knowledge of the pointing directions for each beam along with unscaled coordinate data delivered via photogrammetry, the defined structure of the pattern allows the beam projection origin to be located. The operating algorithm for this method was discussed in Section 2.4.

The are a handful of notable side effects of the redesigned module operating principle. Because calibrations are unique, and the patterns need not be parallel, the sources need not be unique. By allowing the use of different sources, not only is the engineer afforded more freedom in module design, but this facilitates use of different wavelength diodes, such as the current red 658 nm diode along with a green 532 nm diode. By using wavelength filters to temporarily view one pattern at a time, the post-processing burden could be significantly reduced. In addition, adding a third pattern source to the scene in a different location would allow the module would grow from a single optically projected length scale represented in the scene to three length scales. Extra length scales in a scene provides a source of redundancy to spread out the error and allows cross-checking of calibrated length scale values.

With these concepts in mind, two different projection module designs were generated within FRED® which utilized a single diffractive optical element. The first module, shown in Figure 2.16, uses two mirrors to maintain pattern symmetry. A laser source with integrated diffractive optical element is incident on a beam splitter positioned between the steering mirrors. Because of the symmetry in the component design, each projected grid has an equal optical path length to the far field and pattern symmetry is preserved which can aid in post-processing of point data.

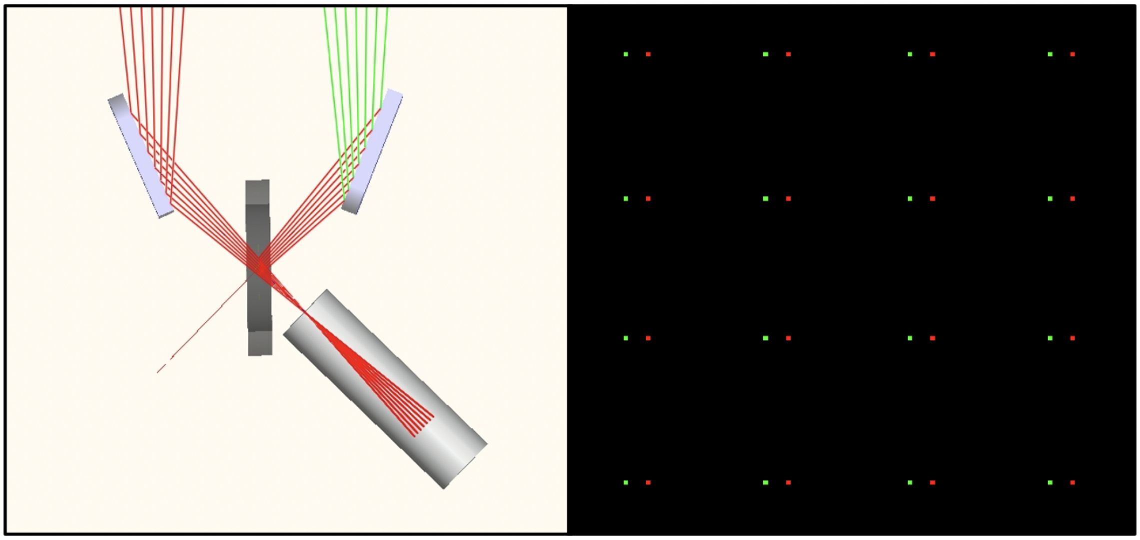


Figure 2.16: Double mirror module design with simulated beam pattern.

The additional components add complexity and cost to the module, however, and due to physical space constraints when mounting all of the components, the angle of incidence of each diffracted pattern is relatively shallow on the steering mirrors. As a result, the separation distance between mirrors must be kept to a minimum lest the aperture be overfilled. This unfortunately reduces the total separation length between projection origins, shortening the optically projected length scale and negatively impacting system performance.

The second module, shown in Figure 2.17, realizes two projected patterns by use of a beamsplitter and a single fold mirror to duplicate the pattern. The reduction of the second steering mirror saves cost and allows the source and diffractive element to be placed closer to the beamsplitter.

By placing the source closer to the beamsplitter, the separation distance between pattern projection origins can be increased versus the two mirror design. Unfortunately, it also means that the optical path length between the projected grids is different. This leads to one grid having expanded more than the other for a given distance from the module, which disturbs pattern symmetry and can lead to beam spot overlap. Additionally, by having one pattern expand at a faster rate than the

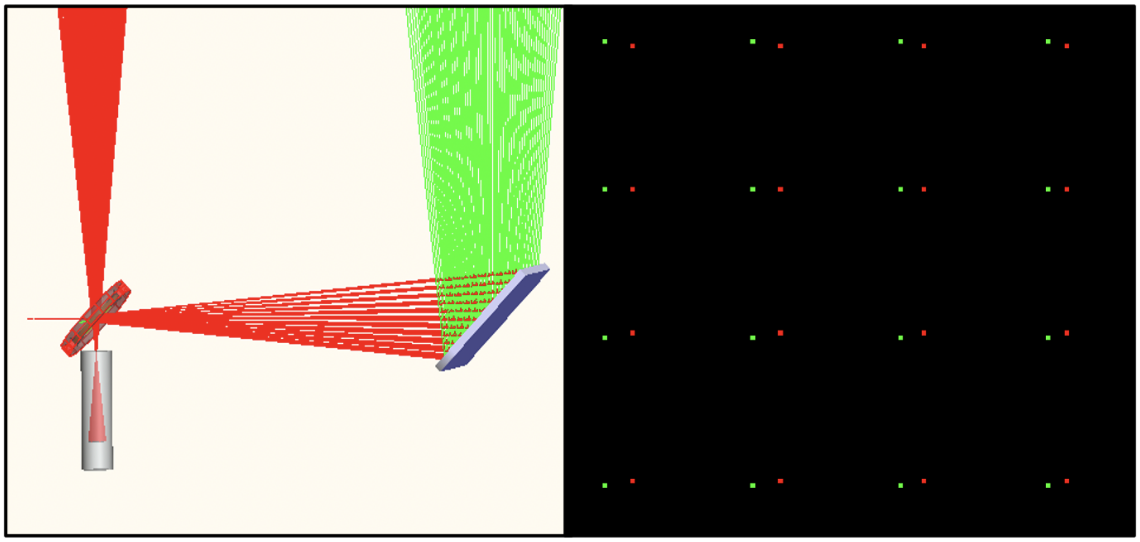


Figure 2.17: Single mirror module design with simulated beam pattern.

other, it means that the crossover length between the beam patterns is longer than that of a symmetrical module, as seen in the simulation data in Figure 2.18. This can result in an area in front of the module where beam patterns are crossing, making unique identification of each beam difficult. In experience, the overlap has not been a problem at a realistic working distance of 1 to 2 meters.

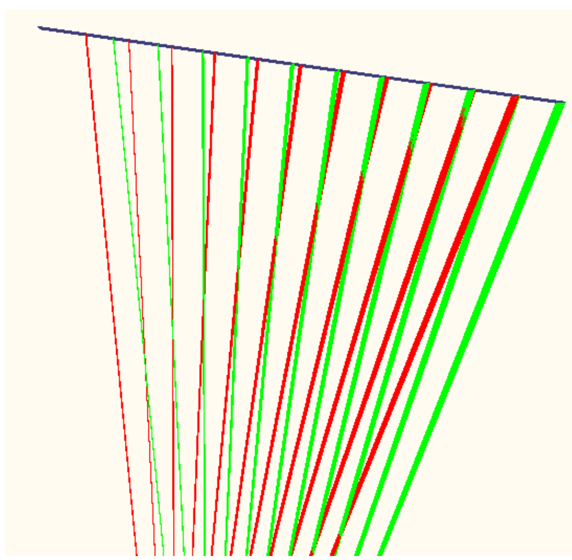


Figure 2.18: Single mirror module beam crossover.

Following evaluation of both system types, the single mirror system was selected to reduce not just cost and complexity, but also eliminate the drift associated with a second kinematically mounted mirror. As this is merely a prototype, pattern asymmetry is an acceptable flaw for what is essentially a proof-of-concept for the optical projected length scale method.

2.6 Projection Module Prototype Fabrication

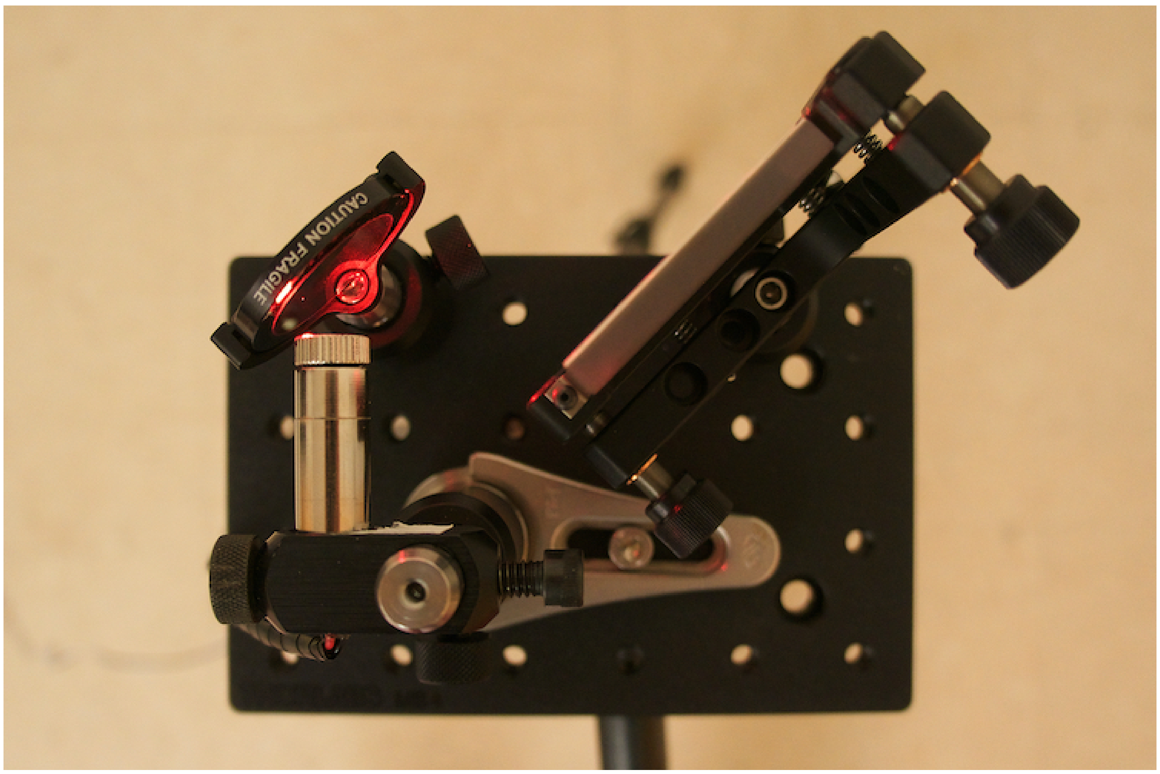


Figure 2.19: Prototype Optical Pattern Projection Module

The prototype module, shown in Figure 2.19, utilizes a single source and diffractive element in combination with a beamsplitter and fold mirror to generate the second grid pattern. The diffractive optical element was available from a prior project. As previously discussed, the initial projection location solving algorithm was based upon the idea of pattern symmetry and representations of the length scale in parallel beam pairs. A pellicle beamsplitter was chosen to minimize optical path length differences between the projected patterns. However, the final algorithm has functionally deviated significantly from its inception, now allowing for unique pattern calibrations and additional design freedom. The pellicle beamsplitter is no longer required, and in principle, something like a cube beamsplitter would be adequate for a production model. The prototype module uses a pellicle style beamsplitter solely because it was already purchased and available for use.

A 658 nm 40 mW diode laser is mounted atop an optical post to a 4" x 6" optical board. A Laser Components DE-R258 diffractive optical element, generating an 11 x 11 dot matrix grid with a nominal full angle of 29.3° x 29.3°, is fixtured in the end of the diode lens stack. The generated grid is incident on a Thor Labs BP145B1 45:55 1" pellicle beamsplitter in a fixed mount. One generated grid is transmitted through the beamsplitter, while the reflected grid is directed to a kinematic mounted 2" protected silver $\lambda/8$ square fold mirror, before being reflected out into the scene.

Following fabrication, a basic alignment was performed to ensure that beam spot overlap in the left and right grids at typical working distances was minimized, as shown in Figure 2.20. Following fabrication, the next step was to characterize the pointing directions of each pattern, and solve for the Euclidean length between the projection origins of the patterns. The module's calibration is discussed in Chapter 3.

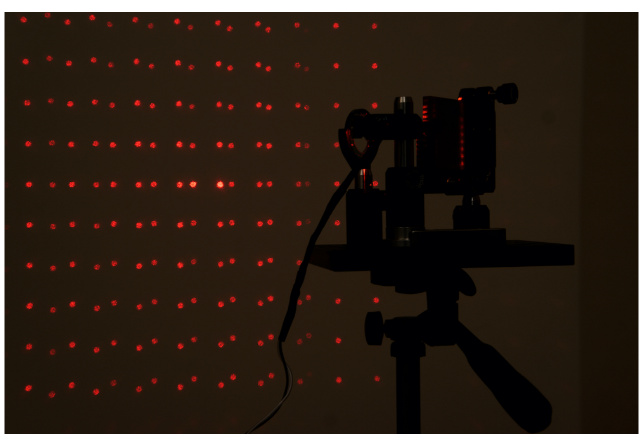


Figure 2.20: Projected Optical Pattern

CHAPTER 3: MODULE CALIBRATION

In this chapter, a review is given on the design and use of diffractive optical elements. A calibration method is proposed which allows for characterization of both the beam pattern pointing directions and the separation between projection origins. To accomplish this, photogrammetry is combined with a novel algorithm to calibrate optically projected beam patterns. This calibration process requires a calibrated length scale artifact purpose built for photogrammetry. The artifact is designed, fabricated, calibrated, and its performance evaluated. Three separate calibrations are performed, and uncertainty in the calibration is evaluated through a Monte Carlo simulation.

3.1 Background

Diffractive Optical Elements (DOE) serve a critical role in an optically projected length scale system by generating the optical pattern necessary for operation. A DOE is an optical component which modifies the amplitude and/or phase of an incident wave front. Wave front modification allow for a DOE to serve in a variety of roles, from Fresnel lenses to laser pattern generation for biomedical applications. While the sole use of a DOE in this application is as a fanout grating to generate the structured pattern, general background is provided on diffractive elements to explain why a novel method is required for characterization of the optically projected pattern.

While Leonardo da Vinci's writings hinted at the phenomenon of diffraction, discovering credit and the creator of the term *diffraction* belongs to Italian mathematician Francesco Maria Grimaldi, who rejected the corpuscular theory of light in his posthumous book, *De Lumine*. His writings describe experiments with pinholes and a scratched metal surface [52], the latter which could be considered the earliest experimental grating [53, 54]. Mathematician James Gregory would describe using a feather as a grating only a few years later in 1673 in his correspondence to colleague John Collins [55].

Dutch physicist Christiaan Huygens was a central figure in establishing the wave theory of light. In his 1690 *Treatise of Light*, he argued that each point on an optical wave front could be thought of as a source of wavelets, and the forward superposition of these wavelets established the new wave front [56]. French physicist Augustin-Jean Fresnel would expand upon this in 1818 in his *Memoir on the Diffraction of Light*, to form what is now known as the Huygens-Fresnel Principle [57]. Fresnel argued that diffraction could be explained by a combination of Huygens' wave theory along with Fresnel's own ideas about interference. A disturbance in light in these conditions at a given point could be thought of as arising from the superposition of secondary waves that originate from a surface between the light source and the disturbance. Kirchoff expanded upon this with his diffraction formula which provides a rigorous mathematical basis for the approximations made in the Huygens-Fresnel principle, and allows wave propagation to be mathematically modeled [58].

The Fraunhofer diffraction equation is a simplified approximation of Kirchoff's model used to calculate diffraction in the far field [60], and can be applied to single slit diffraction, double slit diffraction, infinite slits in the form of a grating, and so on. The far field is defined as any position greater than distance D:

$$D > \frac{A^2}{\lambda}, \tag{3.1}$$

where

A is is the width of the aperture through which the light passes, and

λ is the wavelength of the light.

A classic example of diffraction is Thomas Young's 1801 Double Slit experiment. Young reflected light off a steering mirror, passed the light through a pinhole, and

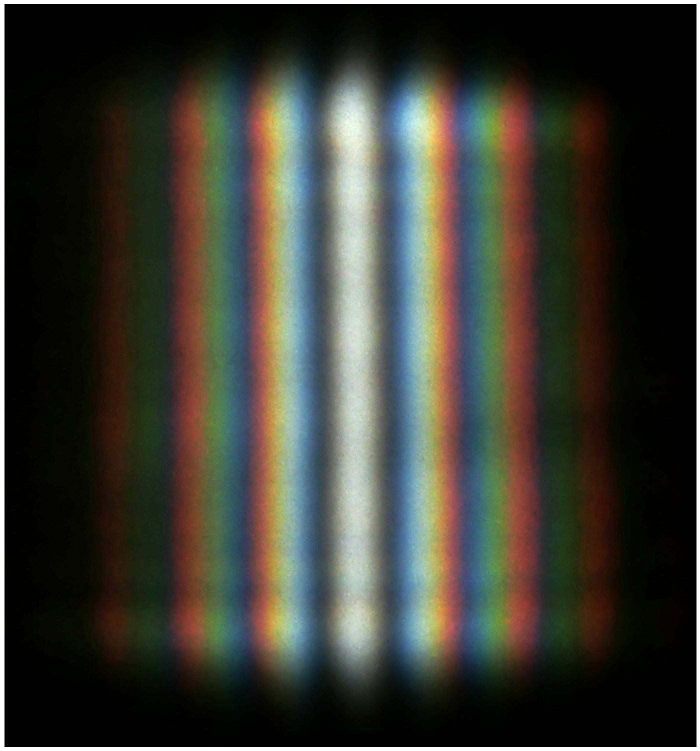


Figure 3.1: White light double slit diffraction pattern. *Image Credit: Aleksandr Berdnikov. Distributed under CC BY-SA 4.0. [59]*

split the resulting beam with a small piece of card stock [61]. He observed a series of black and white alternating intensities, or diffraction orders. Upon closer inspection, he could see color separation, as shown in Figure 3.1.

Diffraction, shown in Figure 3.2, can be readily observed to the naked eye through a single slit or pinhole when the aperture size begins to approach the wavelength of light propagating through. An Airy disc pattern is an example of diffraction. Via simple trigonometry β can be calculated, which is the phase difference between the edge and center rays as they travel to a given position in the far field. The quantity β is $\lambda/2$ out of phase when slit width b is approximately λ. Single slit diffraction can be modeled by the Fraunhofer diffraction equation, where the electric field in the far field, $\bar{E}$, can be approximated [60]:

$$\bar{E} = \frac{\sin\beta}{\beta}, \tag{3.2}$$

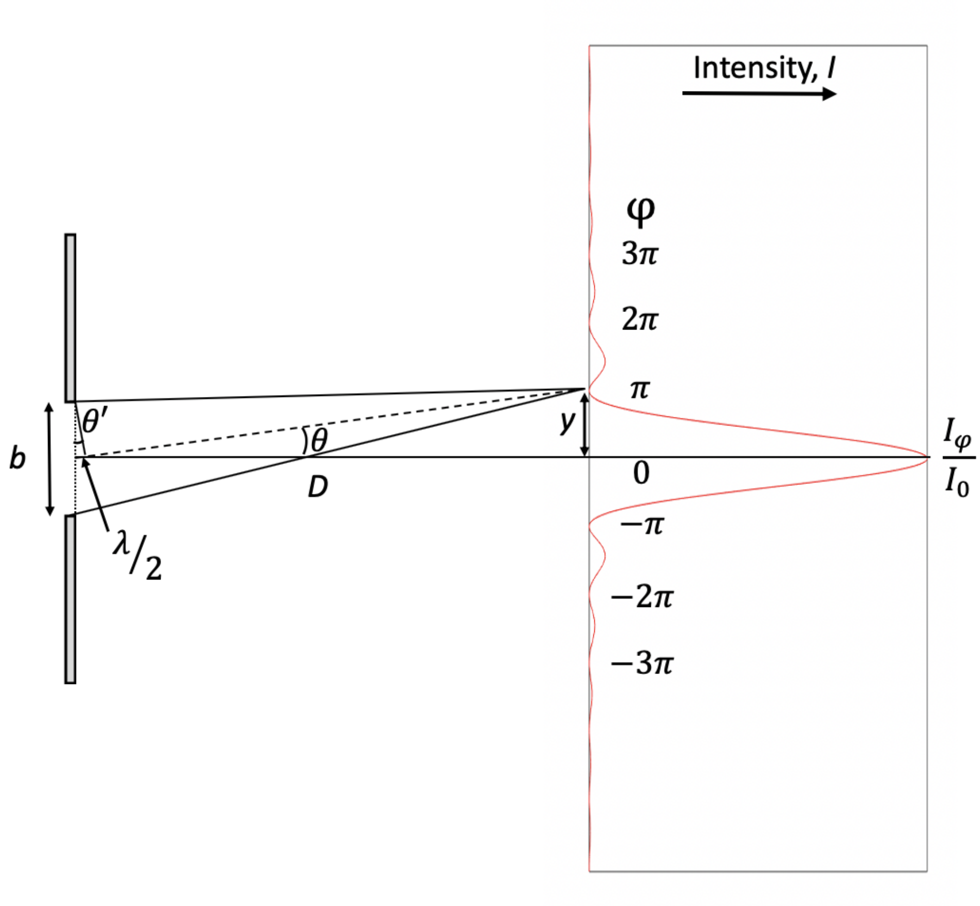


Figure 3.2: Single slit diffraction pattern.

$$\beta = \frac{b}{2} k \sin\theta, \tag{3.3}$$

where

b is the width of the slit, and

k is the wave number, $\frac{2\pi}{\lambda}$.

The square of the electric field $\bar{E}$ yields irradiance, which can be expressed as a function of the diffraction angle, θ:

$$I_\theta = I_0 \left(\frac{\sin\beta}{\beta}\right)^2. \tag{3.4}$$

The maximum irradiance occurs at a value of $\theta = 0$, where optical path length is equal and thus there is no phase difference. This is known as the central beam, or zeroth order. Irradiance minima occur when $\sin\beta$ takes a positive or negative integer value m of π. Substituting $m\pi$ for β, the angle θ_m of the m^{th} order of a minima is calculated:

$$m\lambda = b\sin\theta_m. \tag{3.5}$$

Suppose a second slit is added next to the first, with the same width, b, and a center to center separation distance, d. Each slit would produce the same irradiance in the far field some given distance D, separated by d. However, for the case of pattern overlap, the phase would not necessarily match. Just as β was previously defined as the phase difference between the outer and middle rays of a given slit, the same is done for the slit separation, d, which represents the phase difference from one slit to another, δ:

$$\delta = \frac{d}{2}k\sin\theta. \tag{3.6}$$

Expanding the Fraunhofer diffraction equation to accommodate two slits, the intensity, I_θ, is calculated [60]:

$$I_\theta = 4I_0\left(\frac{\sin^2\beta}{\beta^2}\right)\cos^2\delta. \tag{3.7}$$

When a wave front passes through a slit at normal incidence (i.e.: θ is 0), then $\beta = \delta = 0$, and $I_\theta = 4I_0$. If the slits are extremely narrow, then $\frac{\sin\beta}{\beta}$ approaches 1, resulting in what is essentially two line sources, with interference observed. Conversely, reducing the separation δ to 0 essentially yields a single large slit where diffraction dominates. In a manner of speaking, $\left(\frac{\sin^2\beta}{\beta^2}\right)$ may be thought of as a diffraction term, while $\cos^2\delta$ describes the inference between single slit irradiance patterns. By

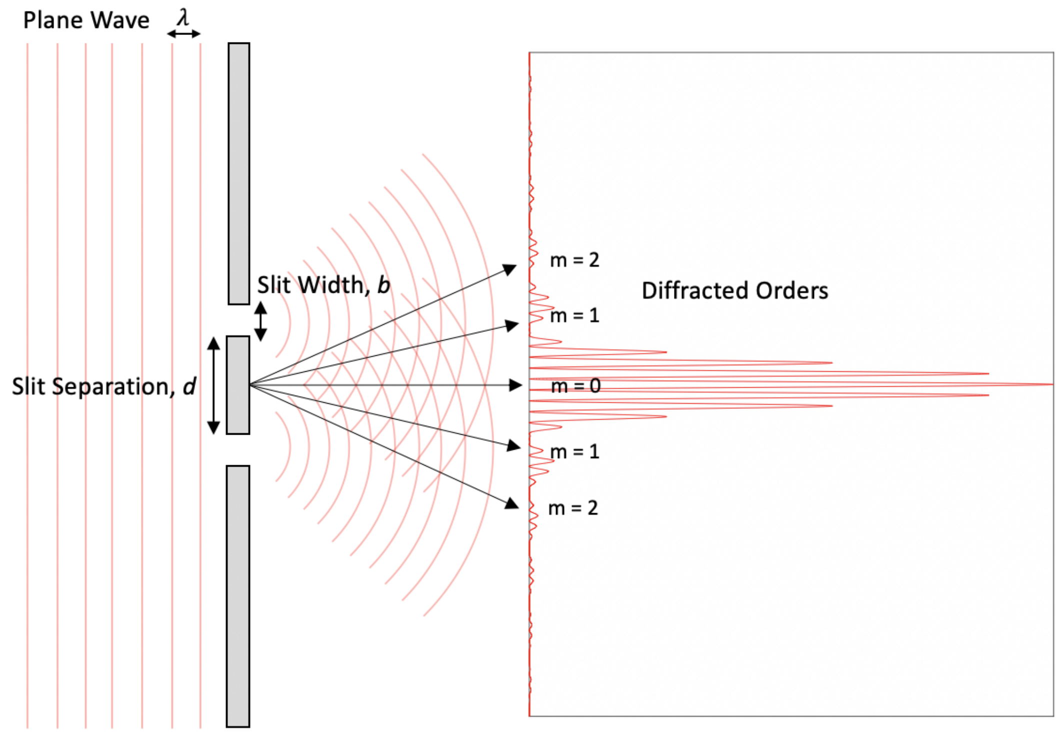


Figure 3.3: Double slit diffraction pattern.

allowing b to take a reasonably narrow value, diffraction patterns from both slits are observed, with interference in the produced irradiance patterns. This pattern can be seen in Figure 3.3, where the main irradiance envelope resembles that of a single slit, but additional interference fringes are visible within individual orders. Minima are observed in the far field where $\beta = \pm\pi, \pm 2\pi, \pm 3\pi....$ Where $\delta = \pm\pi/2, \pm 3\pi/2, \pm 5\pi/2...$, a phase difference of $\pi/2$ is observed, yielding destructive interference.

Expanding from two slits to thousands of slits yields what is essentially the classic diffraction grating, which can be envisioned as an infinitely long array of single slit diffraction sources. In such a scenario, the angular location of irradiance maxima and minima would not change compared to a double slit, but the order intensity appears uniform due to the large number of diffracting sources. With grating groove width on the order of the wavelength, periodicity controls the angle of the diffracted orders. The angle of diffracted orders from a grating, θ_m, can be calculated:

$$\Lambda sin\theta_m = m\lambda, \tag{3.8}$$

where

Λ is the grating period, commonly given as lines per unit length,

θ_m is the angle between the diffracted ray and the grating normal for a given order,

λ is the wavelength, and

m is the order number.

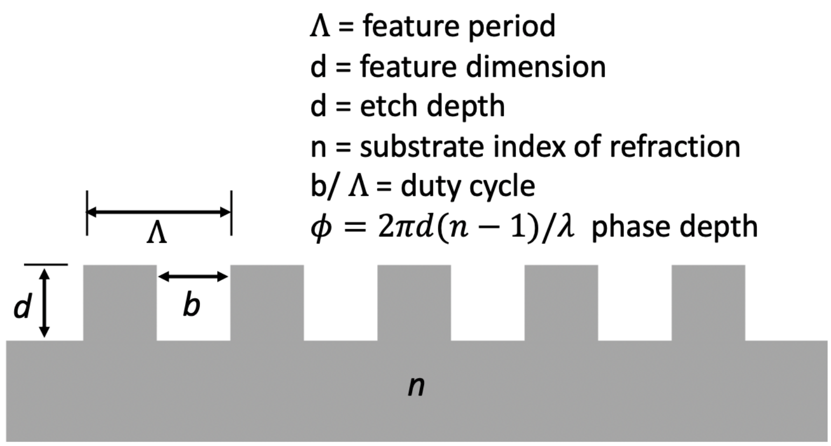


Figure 3.4: Binary phase grating. [62]

The grating equation is an estimation, and assumes the slit width to be infinitely narrow. A binary phase grating, illustrated in Figure 3.4, represents one of the most simplistic grating surfaces, and generally abides by Equation 3.8. Two dimensional gratings may have a triangular (blazed grating) surface profile created by a ruling device or diamond turning machine, a binary rectangular profile (Lamellar grating) created via lithography, or an etched sinusoidal grating created by two beam interference [63].

When light passes through a grating, it will be split into multiple orders. The period, Λ, sets the angle between diffracted orders. A larger period results in a narrower angle between orders. All light does not necessarily pass to a specific order, however. The percentage of light in a given pattern order is controlled by the phase depth, ϕ, which is dependant upon the etch depth, d.

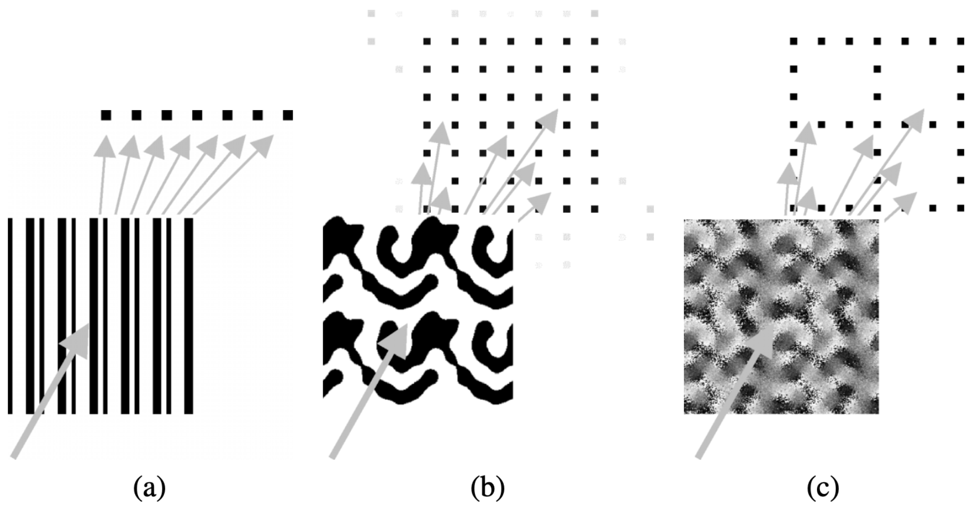


Figure 3.5: Fanout grating complexity can yield increasingly sophisticated diffraction patterns. Progression from the blazed grating in (a) to a surface with multi-dimensional periodic variation (b) yields a square dot grid. By adding control over etch depth and phase depth, a complex three dimensional surface can control the percentage of power to a given projected feature. *Image Credit: O'Shea et al. [62]*

When phase depth, ϕ, can be controlled, the diffraction gratings can be thought of not as surfaces with repetitive lined structures, but rather ones with a complex three dimensional topography. A single blazed grating can create a series of diffracted orders. Passing a monochromatic beam through two blazed gratings, with one grating rotated 90° about the optical axis, can create a square grid pattern of beam spots. Computer aided control of the feature period, depth, and dimensions can yield a grating that can create a variety of complex optical pattern designs, as illustrated in Figure 3.5.

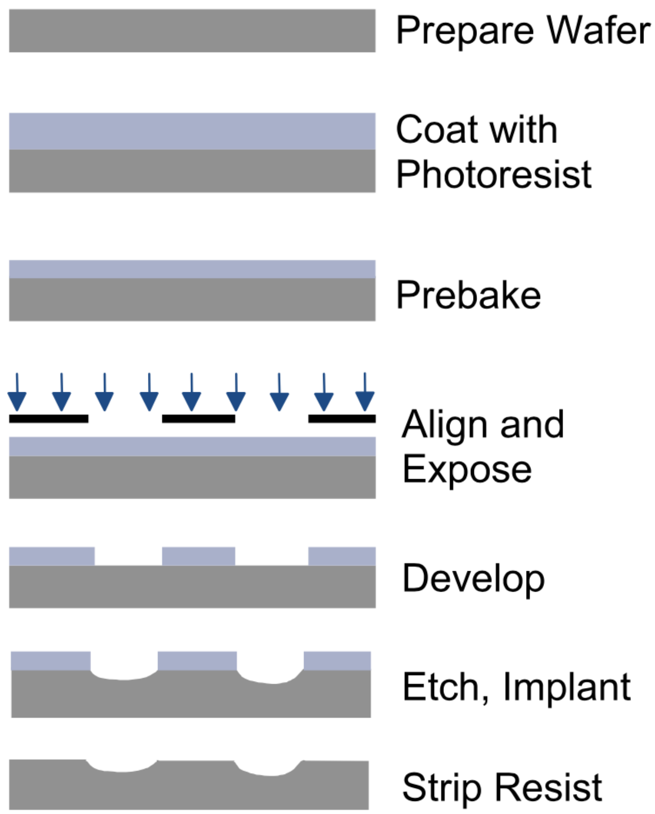


Figure 3.6: Optical lithography process. *Image Credit: Mack [64].*

Most gratings fabricated today are of two types - ruled, and holographic. Evans (1989) provides a thorough and insightful review of the historical development of mechanically ruled gratings [65]. While appearing simple in design, manufacturing of gratings nevertheless requires the utmost attention be given to precision of the process. For example, specific types of gratings require sub-nanometer periodic errors in groove spacing. As Evans points out:

> Mechanical ruling has long seemed the epitome of precision engineering, combining almost the simplest possible set of "wanted" machine motions with the specifications on "unwanted motions" that demand specific attention to the minutiae of machine design and performance(p. 87).

Demonstration of diffraction need not be complex. One of the earliest diffraction devices was created by David Rittenhouse, who had clock makers turn fine threads on small pieces of wire, then proceeded to lay parallel hairs across them [66]. Yet with demonstration of the diffraction concept soon came the desire for precision ruled gratings for use in the scientific community. Fraunhofer himself began fabrication of gratings from fine pieces of wire, but soon realized the limitations of the method, and set out creating grooves into gold leaf with a fine tipped diamond tool and ruling machine, though details pertaining to its construction are vague [65].

Early ruling machines created by John Barton and Joseph Saxton between 1820 and 1850 demonstrated reduced groove spacing owing to their increasingly sophisticated construction. Saxton created a series of engines, with his last being power driven to reduce human error in the ruled gratings. Further refinements in the latter half of the 1800s under Rutherford, Rogers, and Rowland came in the form of sophisticated lead screw fabrication and error correction, kinematic stabilization, automated ruling in temperature controlled environments, and so on. Henry J. Grayson continued Rowland's work in the early 1900s, and produced a quality engine that was used into the 1940s. With the advent of computers came improved machine control, improved grating quality, and larger grating dimensions.

Today, diamond tooled ruling machines are responsible for the fabrication of most ruled gratings, though demand for such gratings far exceeds the time-intensive manufacturing capabilities of mechanical ruling engines. Duplication techniques in resins and other substrates has allowed for copies to be created since the 1950s, which offer optically indistinguishable performance from the parent gratings, and in some cases, better. The creation of the laser in the 1960s, however, enabled a new and faster grating fabrication method - specifically, the creation of holographic gratings.

Indeed, the majority of diffractive elements manufactured today are produced by holographic optical lithography. Direct laser writing can also be utilized for generation of diffractive optical elements [67–69], including large diameter ($\approx$ 100 mm) diffractive optics [70]. Lawrence Livermore National Laboratory's Advanced Optical Components and Technologies program uses techniques such as ion-beam-etching and wet-etching to fabricate even larger meter-scale diffractive elements for use in high-energy laser applications and other areas [71].

In optical lithography, a light sensitive polymer, called a photoresist, is applied to the surface of a substrate, then exposed to patterned light. Following development of the photoresist, the base substrate can be etched. The remaining photoresist is stripped from the substrate, leaving the desired optical pattern in its surface [64]. Diffractive surfaces created by optical lithography can be subjected to further lithography in order to create multi-level surface features. The optical lithography processes is illustrated in Figure 3.6.

New efforts are being made in the field of precision glass molding, which presents a cost effective method for bulk fabrication of diffractive elements with complex sub-micrometer surface features. Prater et al. (2016) demonstrates molded glass DOEs with minimal degradation in pattern efficiency and surface feature quality [72].

Gratings can be designed by a process known as direct inversion. A Fourier transform can be used to map an electric field $\bar{E}(x, y, 0)$ in the near field some given length, L, to the far field, $\bar{E}(x, y, L)$, where L is of a sufficiently large distance. Conversely, if the output pattern intensity is known beforehand, it can be mapped to a grid of square discrete spots with phase values as a free variable. The inverse fast Fourier transform (FFT) is taken, amplitude set to unity, and phase restricted a small number of discrete values. Taking the FFT of this pattern to simulates reconstruction of the grid pattern. If correct, the inverse FFT of this generated phase pattern describes the correct DOE surface in the near field to generate the desired optical pattern in the far field [62].

The above method was greatly expanded upon by Gerchberg and Saxton's creation of one of an iterative Fourier transform algorithm (IFTA) [73]. IFTA methods are commonplace today [74–76], and have evolved to allow for completely arbitrary selection of angle for diffracted orders in binary gratings [77].

In a Gerchberg-Saxton IFTA process, illustrated in Figure 3.7, the desired far field pattern amplitude is input into an algorithm along with phase as a free variable. The inverse Fourier transform delivers the near field amplitude and phase. The phase is retained and the amplitude is set to unity, as the substrate is typically glass and it is desirable for all light to pass through for high efficiency. The Fourier transform is taken of the unity amplitude and retained phase to reconstruct the far field amplitude and pattern. Only this time, the reconstructed target amplitude is discarded. A second iteration occurs, with another inverse Fourier transform of the original desired pattern amplitude and newly solved far field phase. Once more, the near field phase is retained, amplitude set to unity, and the Fourier transform taken to reconstruct pattern amplitude and phase in the far field. After enough iterations, the algorithm converges, and a final inverse Fourier transform of the far field phase pattern results in the exact near field phase pattern that must be constructed on the surface of a diffractive optical element.

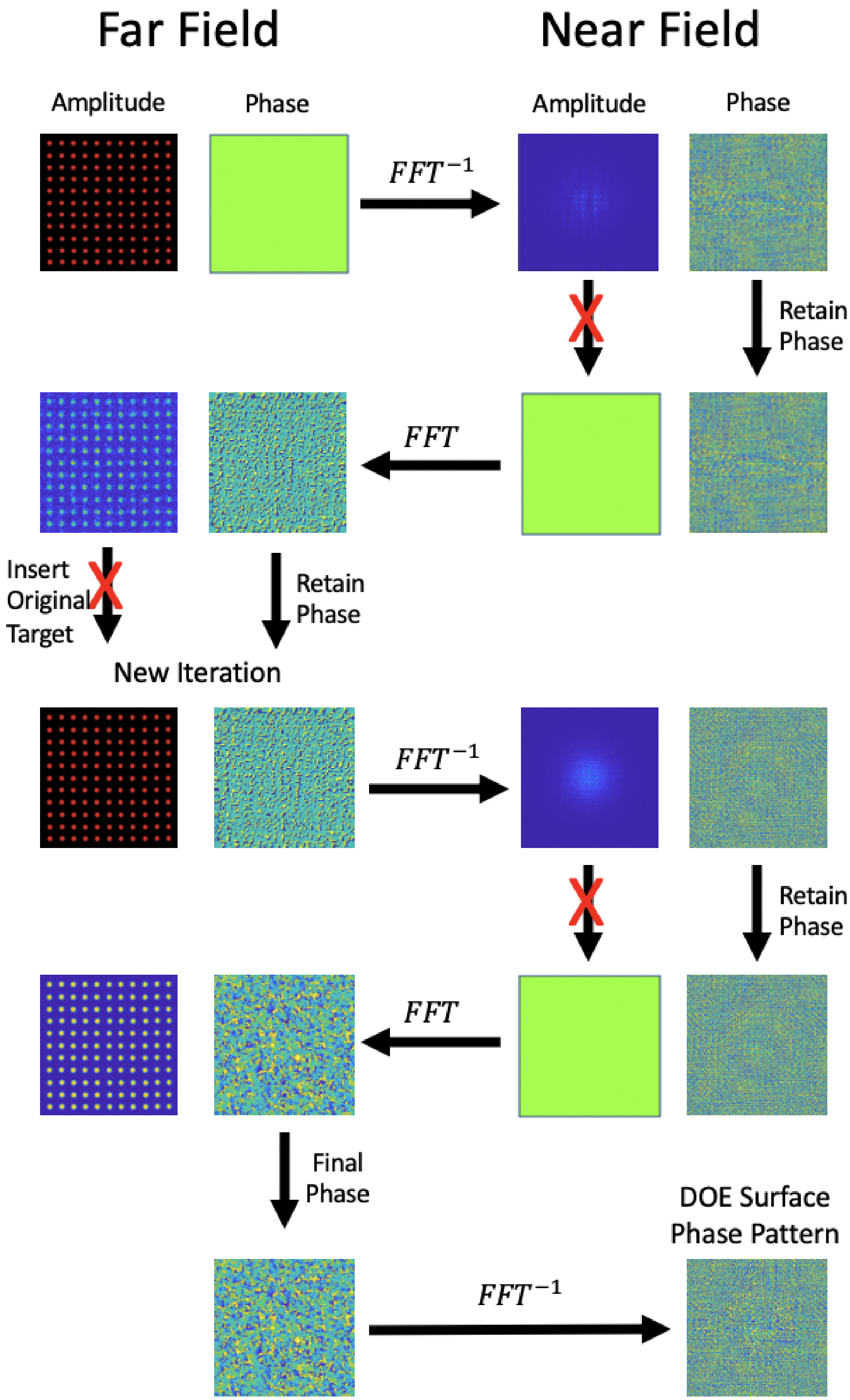


Figure 3.7: Gerchberg-Saxton iterative Fourier transform algorithm.

Many DOE algorithms and equations utilize the Thin Element Approximation (TEA), which neglects diffraction in the substrate [78,79]. This approximation is efficient when phase shifts are lower than 2π with a grating feature size much larger than the wavelength. As feature size decreases to sub-wavelength, or the substrate thickness increases as is necessary to shift phase by more than 2π, the TEA approximation begins to fail. In addition, the Fresnel and Fraunhofer diffraction equations, which readily lend themselves to Fourier transform applications, are based upon paraxial approximations which do not hold true at large transmission angles. Far field patterns produced by wide full angle diffractive elements can suffer from distortion.

While novel adaptations of the Gerchberg-Saxton algorithm can correct this distortion [80], angular calibration of a pattern which may pass through other optical elements can be difficult. While characterization of the period of a blazed or Lamellar ruled grating is possible via diffractometry and other methods [81–83], these methods do not as easily apply to holographic gratings.

In the case of the DOE within the module, while the grating equation allows for theoretical calculation of diffracted orders for 1D ruled gratings and some simple 2D gratings with adequate periodicity, it may not necessarily apply to a complex holographic grating. Additionally, while the manufacturer has specified the expected angular separation between beam orders of the module's fanout DOE, numerical errors will exist as the full angle of the pattern is relatively wide. Furthermore, one grid pattern generated by the DOE passes through a pellicle beamsplitter while the other is incident on a fold mirror. Both present additional opportunity for further distortion to occur in the diffracted pattern.

A selection of data from the beam pattern calibration completed in Section 3.6.2 is presented to illustrate the necessity of uniquely characterizing the pointing direction of each beam produced by the grating. Pointing directions of the 6^{th} row and 6^{th} column of the 11 x 11 grid pattern are examined. The 6^{th} row should nominally have all spots aligned with the Y axis, and the 6^{th} column should have all spots aligned with the Z axis, with elevation (θ) and azimuthal (ϕ) components of 0 radians, respectively.

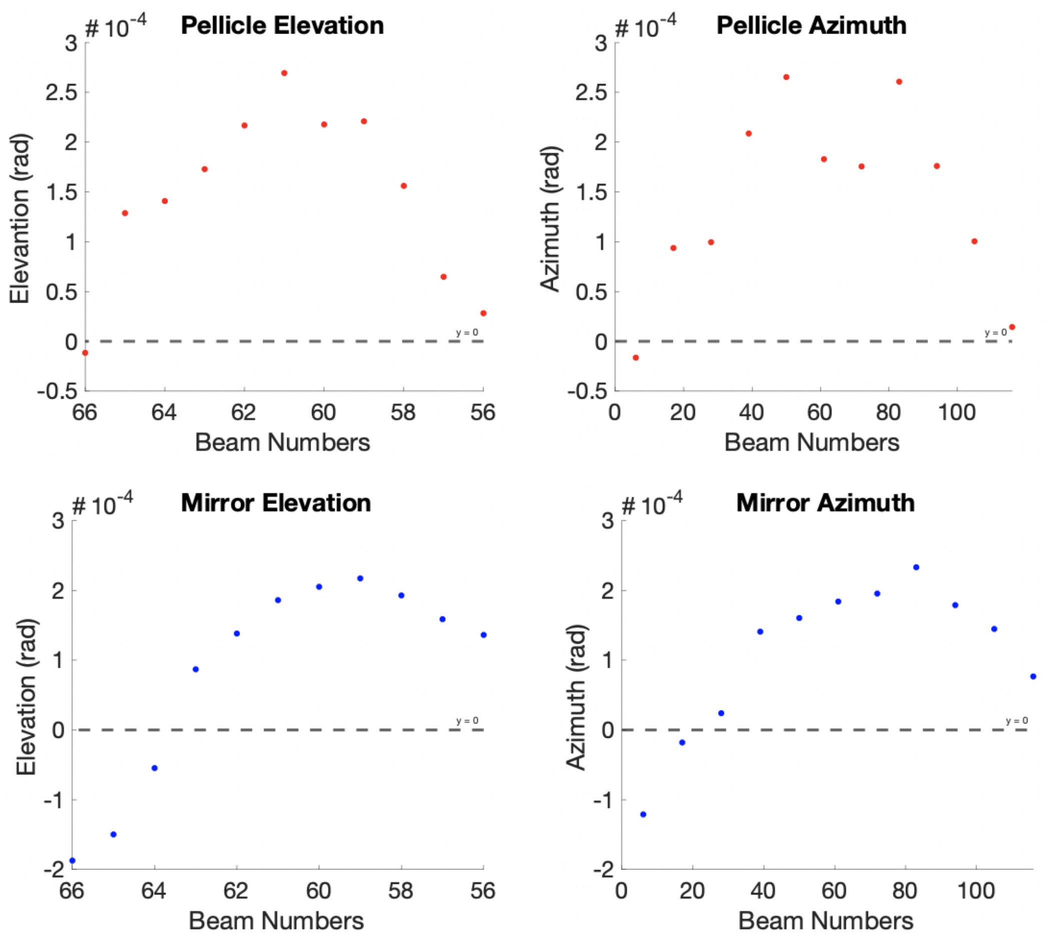


Figure 3.8: Spherical coordinate pointing directions for Y & Z on-axis beams.

In the case of Y axis beam numbers 56 to 66 of the pellicle and mirror grids, a range of 300-400 μrad of deviation can be observed at a projection distance of just one meter. Similarly, the spots on the vertical Z axis (every 11^{th} beam number from 6 to 117) show an overall deviation range of approximately 300 μrad at one meter. The level of uncertainty in the position of projected laser spots from the current photogrammetry system is on the order of 100 μm. The deviations from the manufacturer specified angles are significant and highlights the need for pointing direction calibration of the pattern on a beam by beam basis.

3.2 Calibration Method

As noted in Section 3.1, grating equations allow for theoretical estimates to be calculated for the angle between diffractive pattern orders. This approximation does not hold true for wider angle patterns due to the thin element approximation. Even with a relatively modest 29° pattern utilized in the module, deviations from calculated diffraction order angles are observed. Additionally, theoretical angle computations do not account for variations in the grating surface due to the manufacturing process variation or grating surface damage. In a complex optical system where the grating is not the final element through which the beam pattern may pass, optical abberations or surface deformations in optical elements, such as fold mirrors or beamsplitters, may further distort the pattern and alter the angular pointing directions.

An alternative approach to calibration is by use of a precision rotary stage. Zheng (2014) used a rotary table in conjunction with a CCD camera to image projected spots in a scene along each axis of an optically projected grid [7]. There are various problems associated with this design, however. This method assumes the projection origin of the module to be directly centered upon the rotational axis of the table, which is difficult in practice. Wavelength instability, rotary table uncertainty, and CCD imaging repeatability also must be considered. In addition to these issues, the rotary table method is inadequate for a module where multiple gratings are present and designed to be separated by a given distance. Once fabricated, the laser projection source and diffractive optical elements cannot be altered, lest the calibration become invalid. Other groups have positioned the DOE such that the pattern is shining down upon the surface of a 2D stage mounted to an optical table, and used the stage to translate an upward facing power meter from beam to beam [72]. This method may suffice for a shallow pattern angle, but becomes impractical and error prone if attempted with a wide angle pattern.

An additional problem with the rotary table method is that the calibration does not find the angular coordinates of all beams, but rather only for those spots on axis. Solving for distortion coefficients can allow for correction of off axis beam angles. Furthermore, and importantly, none of these methods solves for the separation distance between the projection origins. As such, a novel non-contact calibration method has been devised which utilizes photogrammetry and a processing algorithm to solve for the angular description of the beam pattern, while simultaneously solving for the projection origin.

While a beam pattern from a DOE can be observed in the far field, directly measuring the intersection origin of that pattern in the near field is not possible. The calibration method needs to be somewhat indirect. Photogrammetry can deliver the three dimensional coordinate data for projected laser spots, which under careful consideration is all the information necessary for calibration.

As the beam pattern propagates through the scene, a screen is placed at different distances from the module in order to sample the projected grid as it expands, as shown in Figure 3.9. In essence, a 'slice' of the pattern is acquired at a specific depth. Photogrammetry is used to obtain three dimensional coordinate data for each beam spot, $(X, Y, Z)_{Grid}$, relative to a coordinate system origin established elsewhere in the scene. These coordinates must be scaled by a well characterized photogrammetric length scale artifact. The scaled coordinate data of all ten slices is then utilized to plot what is similar to a best fit line for each beam, with the constraint that all beams must intersect at a single point, the module projection origin $(X, Y, Z)_{Origin}$.

This process obtains not only the module's origin, but also the pose of the projection sources relative to the pointing direction of their central zeroth order beam and the pattern rotation about that beam. When each pattern's origin has been solved, the Euclidean distance between origins can be computed, yielding the separation distance, L. Because the data points have been scaled with a real calibrated artifact, the module

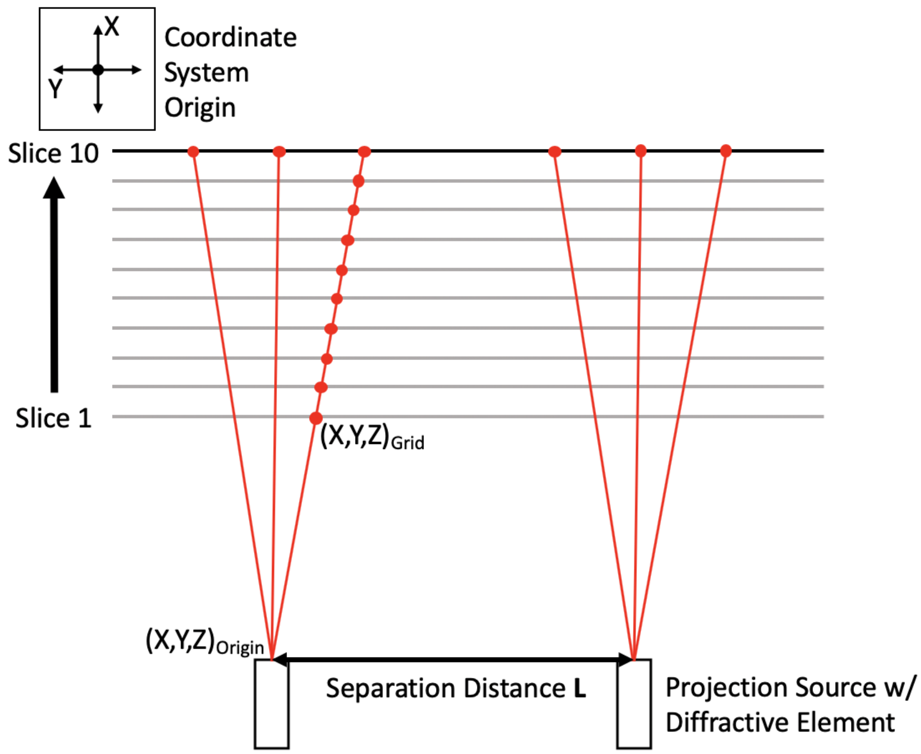


Figure 3.9: The beam pattern is sampled at ten different distances, or slices, away from the projection module. Photogrammetry gives coordinate data $(X, Y, Z)_{Grid}$ for all beam spots, which is then used to create a best fit line, with the constraint that all beams must intersect at a single origin point, $(X, Y, Z)_{Origin}$.

projection origin separation will be correctly scaled. The algorithm utilized to perform this calibration is detailed in Section 3.3. A photogrammetric length scale artifact was fabricated to provide scale to the calibrated values. The artifact is discussed in Section 3.4. As 121 points per slice are sampled over a total of 10 projection slices in the scene, simulations suggest that uncertainty in the solved projection origins is driven down due to the large number of points, despite larger amounts of noise in the laser spots. This is discussed more thoroughly in Section 3.6.3.

3.3 Calibration Algorithm

The algorithm used to calibrate the module is similar to the length scale solving algorithm described in Section 2.4. Complete code for the module calibration algorithm can be found in Appendix A.4. While both algorithms use homogeneous transform matrices to perform passive coordinate transformations, the calibration algorithm differs in a few key ways, which will be described below.

3.3.1 Algorithm Theory

In a photogrammetric measurement, the user must establish a global coordinate system by which all coordinates are assigned. After the global origin is selected, other targets or points in the scene are used to establish a dominant axis. Selection of a third point allows for the establishment of a plane, and two axes are constructed orthogonal to the dominant axis.

Figure 3.10: Three RAD targets, outlined with black boxes, allow the user to create an origin and orientation for the coordinate system. The established global coordinate system does not coincide with that of the module, whose local coordinate system is established by its projection origin, and grid orientation about the zeroth order central beam.

Illustrated in Figure 3.10 is a typical photogrammetry measurement scene. Three RAD targets are boxed by black tape. The origin is established at the point in the bottom left of the figure. The target in the upper left serves to establish the dominant Z axis, whilst the target in the upper right hand corner establishes the Y axis. A right handed coordinate system is established at the origin in the bottom left, with +Z in the direction of the upper left target, and -Y in the direction of the upper right target.

During the calibration process, not only must the separation between modules be defined, as shown in Figure 3.9, but also the spherical coordinates which describe the pointing direction of each beam. Illustrated in Figure 3.11 is a sample beam pattern. Once calibrated and aligned, the central beam should have spherical coordinate values of 0 radians in both the azimuth,ϕ, and elevation, θ.

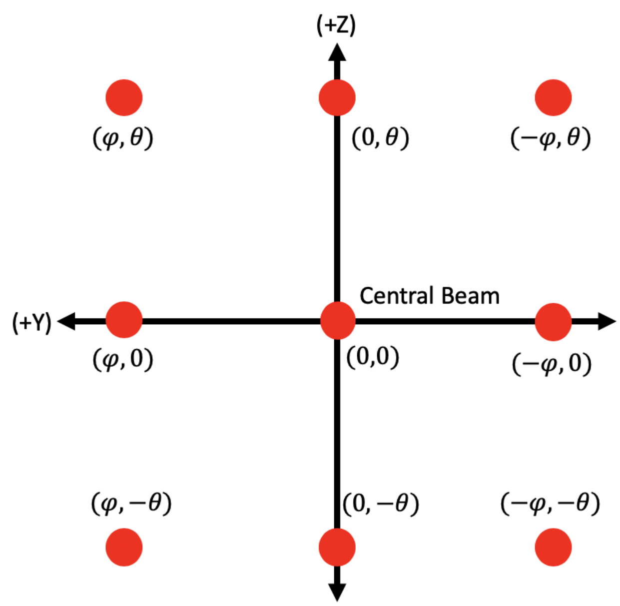


Figure 3.11: Sample beam pattern. Beam spots are characterized by their spherical coordinate pointing directions ϕ and θ relative to the central beam.

A problem arises during pattern characterization as the module's projection origins are located away from the global coordinate system origin. The beam pattern must be characterized relative to the pattern's projection origin, but MATLAB® computes spherical coordinates based upon the azimuthal and elevation of a point relative to the global coordinate system origin. With the module in its own local coordinate system, a mismatch exists. Even if the beam projection origin did coincide with the global coordinate system origin, there is no guarantee that the pose of the module aligns with the global coordinate system axes. While characterization of pose has no impact on correctly determining the projection location, knowing the pose helps assign a closer starting guess during processing of data with the length scale solving algorithm. Furthermore, pose characterization allows the module itself to be utilized as a true 6 DOF sensor, if desired.

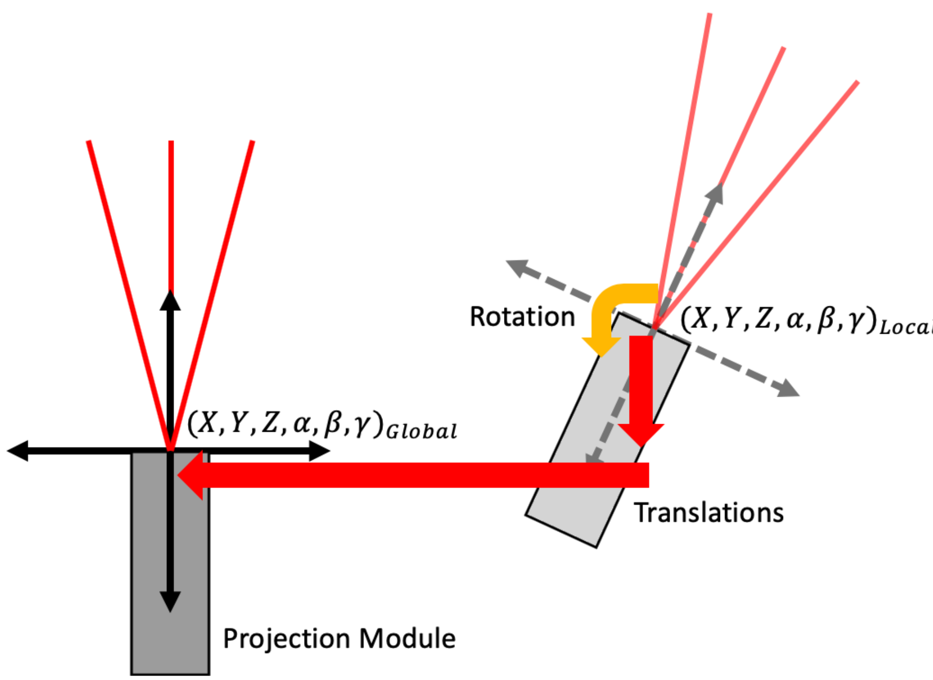


Figure 3.12: The module's projection origin is offset from the global coordinate system chosen during processing in PhotoModeler®. By solving for the translation and rotation of the module away from the global origin, a transformation can be performed to map from local to global coordinates, and correctly characterize the beam pattern.

This presents a unique problem as the module must be correctly described in the global coordinate system. A mapping is required to move from the module's local coordinate system to that of the global coordinate system. In a lab setting, the module's local coordinate system will always be translated and rotated away from that of the global coordinate system by some amount $X, Y, Z, \alpha, \beta, \gamma$, as has been illustrated in Figure 3.12. By solving for the translation and pose of the module away from the global coordinate system, a homogeneous transformation can be performed to re-center the data points at the global coordinate system origin and describe them in spherical coordinates.

The first step in the process is to utilize MATLAB® to simulate a data set that is translated and rotated relative to a global coordinate system origin. Once a transformed dataset has been created, it can be used to test an algorithm that will solve for those translations and rotations. Upon solving, those values can be used to 'correct' the data and realign module's local coordinate system with the global coordinate system. To create the test dataset, an inverse homogeneous transformation is applied to a simulated grid pattern. The transform shown in Equation 3.9 is an alias, or passive, homogeneous transformation, which does not alter the data points positions, but only the reference frame and coordinate description by which they are viewed.

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} r_{ij} & r_{ij} & r_{ij} & t_x \\ r_{ij} & r_{ij} & r_{ij} & t_y \\ r_{ij} & r_{ij} & r_{ij} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} X_{Grid} \\ Y_{Grid} \\ Z_{Grid} \\ 1 \end{bmatrix}, \tag{3.9}$$

where

$t_{x,y,z}$ describes the translation,

r_{ij} is a 3 x 3 matrix describes the pose rotations to be applied,

$(X, Y, Z)_{Grid}$ are the original beam spot coordinates, and

(X, Y, Z) are the modified beam spot coordinates as viewed in the new reference coordinate system.

A test data set was created in MATLAB®. The data set is an 11 x 11 beam pattern with a 29 degree full angle, projected from a (0,0,0) origin over a range of 10 meters, with a single data slice every meter. The transformation matrix was then instructed to translate the data set by 5 meters in the X, 2 meters in the Y, and rotate by 0.2 radians along each of the X, Y, and Z axes. The pre-transformation data set is illustrated in red in Figure 3.13. The post-transformation data set, illustrated in blue, shows the results of the coordinate transformation.

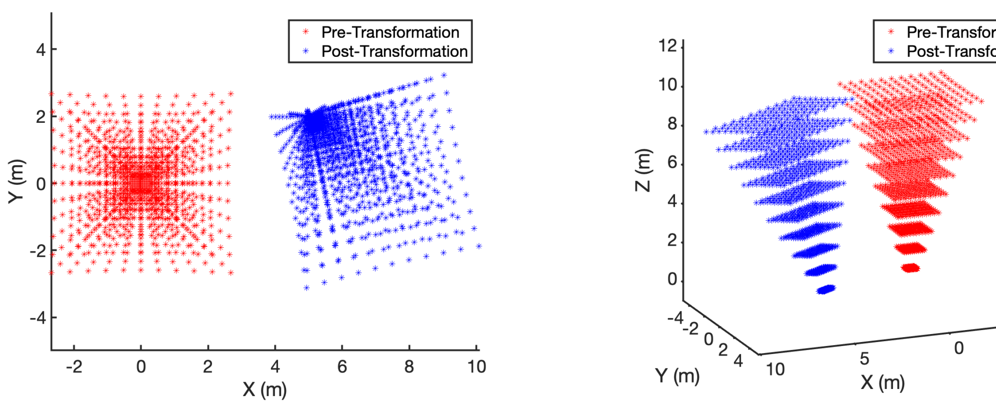


Figure 3.13: A test data set, shown in red, is translated by 5 meters in the X, 2 meters in the Y, and rotated 0.2 radians about the X, Y, and Z axes. The post-transformation data set is shown in blue.

Having simulated a projection source and dataset that is located away from the global coordinate system origin, an algorithm was then constructed which solves for the location of the projection origin and determines the correct translations and rotations necessary to map the local coordinate system into the global coordinate system. This process will be completed in two steps. In the first step, the projection location is solved by using homogeneous transforms to view the data from different reference frames. In the second step, described in Section 3.3.3, the pose of the module is determined. The general flow chart for first step of the algorithm is illustrated in

Figure 3.14.

3.3.2 Solving Translations

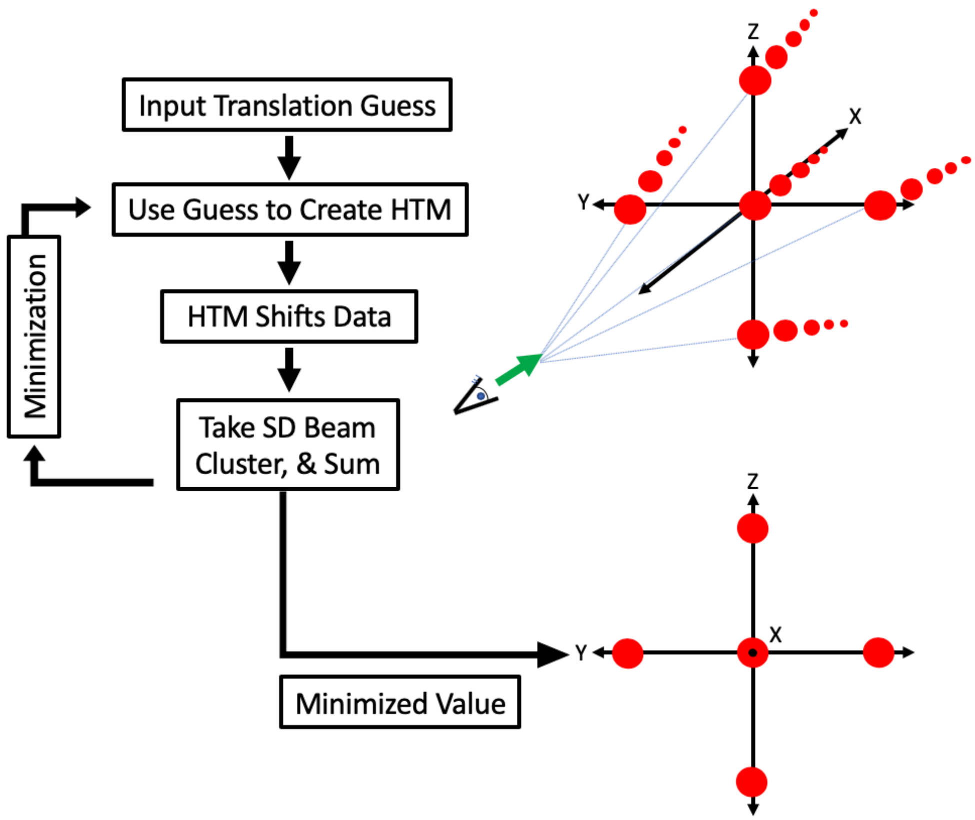


Figure 3.14: A flow chart for the translation solving step of the calibration algorithm. The data set is passively transformed and viewed from different reference frames until the spherical coordinate components for a cluster of spots corresponding to a single beam reaches a minimal value. All beams are evaluated simultaneously.

An initial guess of the module's projection origin (X, Y, Z) relative to the global coordinate system origin is input into the algorithm. In the lab setting, approximating module location to the nearest meter was sufficient for algorithm convergence. The input guess is used to create the first homogeneous transformation, as per Equation 3.9. The passive transformation, shown in Equation 3.10, strictly translates the reference frame from which the dataset is viewed.

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & t_x \\ 0 & 0 & 0 & t_y \\ 0 & 0 & 0 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} X_{Grid} \\ Y_{Grid} \\ Z_{Grid} \\ 1 \end{bmatrix} \tag{3.10}$$

After the transformation, the Cartesian coordinates are converted into spherical coordinates by Equations 3.11 - 3.13. When calculating the azimuthal component ϕ shown in Equation 3.11, it is important that the two-argument arctangent (MATLAB® function $atan2$) be used to avoid ambiguous returned values. While the Euclidean distance, r, is also delivered during the spherical coordinate conversion, this value is not used in the algorithm as only the pointing directions are required.

$$\phi = \arctan \frac{Y}{X} \tag{3.11}$$

$$\theta = \arctan \frac{\sqrt{X^2 + Y^2}}{Z} \tag{3.12}$$

$$r = \sqrt{X^2 + Y^2 + Z^2} \tag{3.13}$$

Following conversion to spherical coordinates, evaluation of the beam pattern is done in the follow manner. Recall that each beam in the grid pattern is sampled at ten different distances. These 10 points for each beam form a beam cluster. As viewed from the reference frame of the initial guess in the algorithm, the standard deviation is taken of the azimuthal components ϕ and the elevation components θ of the entire cluster. The standard deviations of each cluster are then summed over the entire set of total beams in the pattern. This value, shown in Equation 3.14 is then evaluated via a Nelder-Mead minimization routine [51].

$$Value = \sum_{1}^{N^2} \sqrt{\frac{1}{j-1}\sum_{i=1}^{j}(\phi_i - \bar{\phi})^2} + \sum_{1}^{N^2} \sqrt{\frac{1}{j-1}\sum_{i=1}^{j}(\theta_i - \bar{\theta})^2}, \tag{3.14}$$

where

N is the number of beams per side in the grid,

ϕ, θ are the azimuthal and elevation components of a beam spot, and

$\bar{\phi}, \bar{\theta}$ are the mean values of the azimuthal and elevation components of a set of 10 spots for a given single beam.

The mean values for the azimuthal and elevation components $\bar{\phi}, \bar{\theta}$ are defined as:

$$\bar{\phi}, \bar{\theta} = \frac{1}{j}\sum_{i=1}^{j} \phi_i, \theta_i, \tag{3.15}$$

where

j is the number of spots sampled per line.

Also shown in Figure 3.14 are two sample beam patterns. Shown in the upper beam pattern are five beam clusters where beam points do not overlap from the perspective of the viewer, and thus have a higher standard deviation for each of their spherical coordinate components. Viewing such a pattern would suggest that the beam intersection point lies in a different location (along the X axis), as illustrated by the eye in the scene. When the algorithm has converged upon the true projection origin for all beams in the pattern, as shown in the lower portion of Figure 3.14, each of the 10 spots in a beam cluster should nominally overlap with one another, and every beam cluster will exhibit this overlap simultaneously. At this point, the standard deviation for spherical components in all beam clusters should be reduced to a zero value.

By making iterative passive transformations around the scene, the grid pattern can be evaluated from different reference frames. When the algorithm has minimized the value and found the true reference frame where the pattern beam cluster standard deviations have been minimized, the correct translation of the grid away from the global coordinate system has been solved. This translation value is then used to perform a final transformation to move the entire data set such that the projection location coincides with the global coordinate system origin.

3.3.3 Solving Rotations

Once the translation has been solved, the rotations, or pose, of the module's projection sources can be solved. An outline for this process is illustrated in Figure 3.15. While this section of the algorithm is similar to the portion which solves for translation, a key difference is in the manner by which pose is evaluated. The algorithm begins with an initial guess of the module's rotation. As the module's grid is approximately level with the global horizontal axis in the lab, and pointing direction generally coincides with the $\hat{X}$ direction, a (0,0,0) guess for (α, β, γ) module pose typically leads to convergence.

The guessed pose is used to create the first homogeneous transform matrix by which the grid pattern will be rotated. Again, a passive transformation is used, shown in Equation 3.16, which alters not the points but only the reference frame from which they are viewed. As shown in the transformation matrix, the t_x, t_y, t_z values are now zero, as the reference frame is only being rotated.

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} X_{Grid} \\ Y_{Grid} \\ Z_{Grid} \\ 1 \end{bmatrix} \tag{3.16}$$

Once the rotation has been performed, a coordinate transformation changes the coordinates from Cartesian to spherical. The center beam, if correctly aligned along the X axis, should nominally have a value of 0 for both azimuthal and elevation components ϕ, θ. Similar claims can be made for points on axis. For points on the

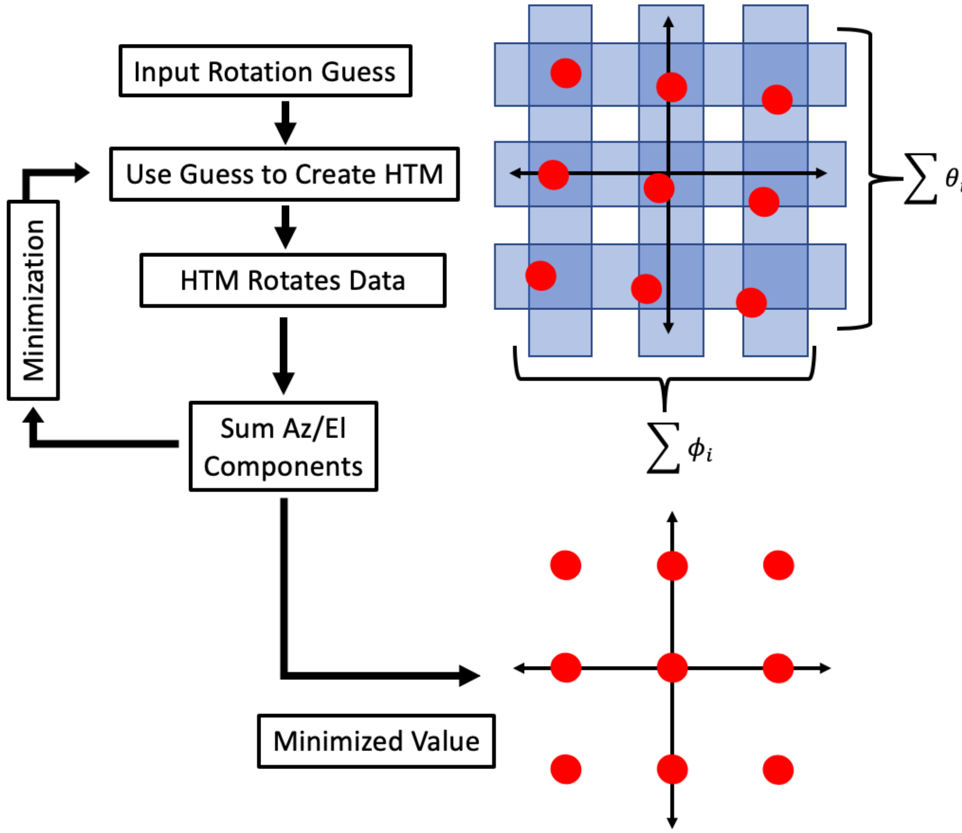


Figure 3.15: A flow chart for the rotation solving step of the calibration algorithm. Following processing, the pose of an unoriented projected grid, shown in top right, can be determined, and then corrected.

vertical Y axis, azimuthal values should be 0 radians. For points on the horizontal Z axis, a 0 radian value for elevation is expected. A similar argument can be extended to all points in the grid. The sum of all elevation components of the grid is minimized when the middle horizontal row of the grid is aligned with the horizontal Y axis. When the middle vertical row is aligned with the vertical Z axis, azimuthal contributions should be minimized. An example of a misaligned grid can be found in Figure 3.15. In each iteration of the algorithm, the azimuthal and elevation components of each i^{th} beam are summed over each slice, and then all slices j are summed, as shown in Equation 3.17. Once minimized the pose of the module relative to the global coordinate system has been determined.

$$Value = \sum_{j=1}^{Slices} \sum_{i=1}^{N^2} \phi_i + \sum_{j=1}^{Slices} \sum_{i=1}^{N^2} \theta_i \tag{3.17}$$

3.3.4 Calibration Algorithm Output

Following calibration, the module's two projection source positions and pose have been solved relative to the global coordinate system origin. With their locations characterized, the Euclidean distance between them is computed which serves as the calibrated length scale distance, L, for the unit. In addition to the distance, a 2 x 121 array is output which contains spherical coordinate pointing directions ϕ, θ for each beam in the pattern. With the beam pattern intersection having been correctly determined, the 10 spots for each beam should overlap, with all points having the same spherical coordinates. The pointing directions are averaged for the 10 points of each beam and this average serves as the calibrated pointing direction. This array is exported as a *.mat file for use in the main length scale algorithm. The locations, pose, and separation distance of the module's projection origins are also recorded for reference.

3.4 Calibration Length Scale Artifact

It is critically important that the separation distance between projection origins be well characterized. If the value is incorrect, the optically projected length scale will impart a bias onto coordinate data during the scaling of coordinates. As such, a high quality length scale artifact is required to scale point data during the initial module calibration. While a handful of existing measurement artifacts were available for use, most were tailored toward Coordinate Measuring Machine (CMM) verification rather than photogrammetry. Therefore, the decision was made to fabricate a photogrammetry scale artifact for use in calibration of the module. This artifact must have high contrast target surfaces which are not only readily detected during photogrammetric measurement, but also must be designed in a manner such that the separation between targets can be characterized.

3.4.1 Artifact Design & Construction

Photogrammetry programs rely on high contrast targets to reduce centroiding uncertainty. While retroreflective targets remain state of the art for positive identification, black on white or vice versa targets have long been used for photogrammetry, and are adequate for this use. After testing a combination of surfaces and colors, matte black paint on a sandblasted aluminum surface yielded a target that was reliably detected and centroided. The artifact performance is discussed more thoroughly in Section 3.4.3.

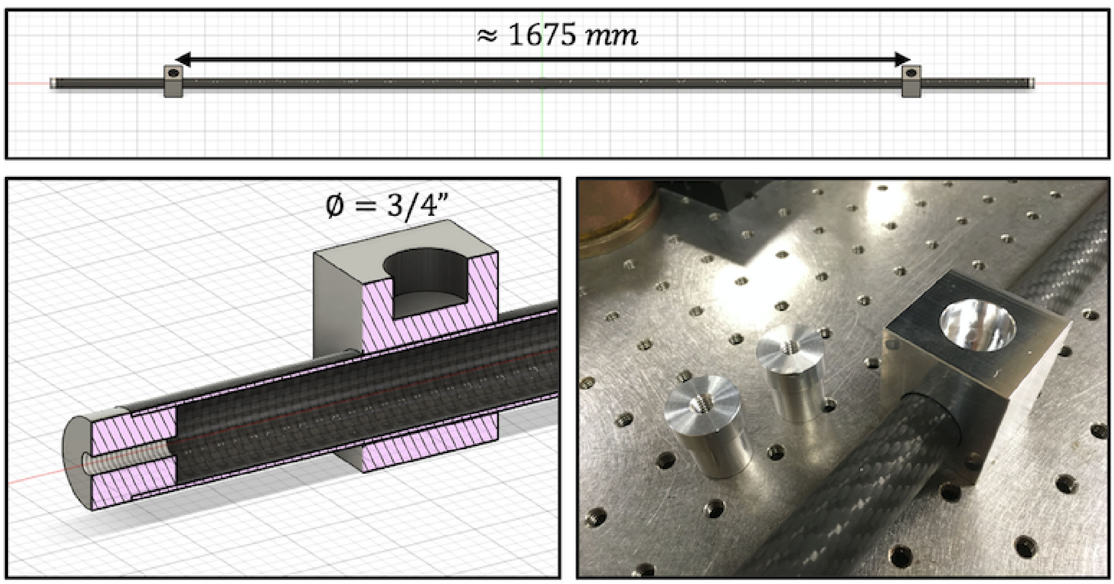


Figure 3.16: A length scale artifact was modeled in Autodesk Fusion 360®. Two aluminum blocks with a milled blind hole serve as targets. A bored hole allows the targets to slip over a cylindrical carbon fiber tube, and the targets are epoxied to the tube at a separation length of approximately 1675 mm on center.

Autodesk Fusion 360® was utilized to create a basic CAD model for the length scale artifact, shown in Figure 3.16. The artifact was constructed from a 6 ft x 0.832" OD x 3/4" ID carbon fiber checked tube. Two 6061 aluminum blocks are milled to size, and bored to a 3/4" ID, allowing them to slip over the tube. In each block, a ⌀3/4" x 0.4" blind hole has been milled, such that the inner diameter can be profiled on a CMM, and the hole-to-hole length measured. Aluminum end caps were fabricated on a lathe, and threaded to 1/4"-20 to allow for attachment of a threaded ball should use in a kinematic mount be necessary.

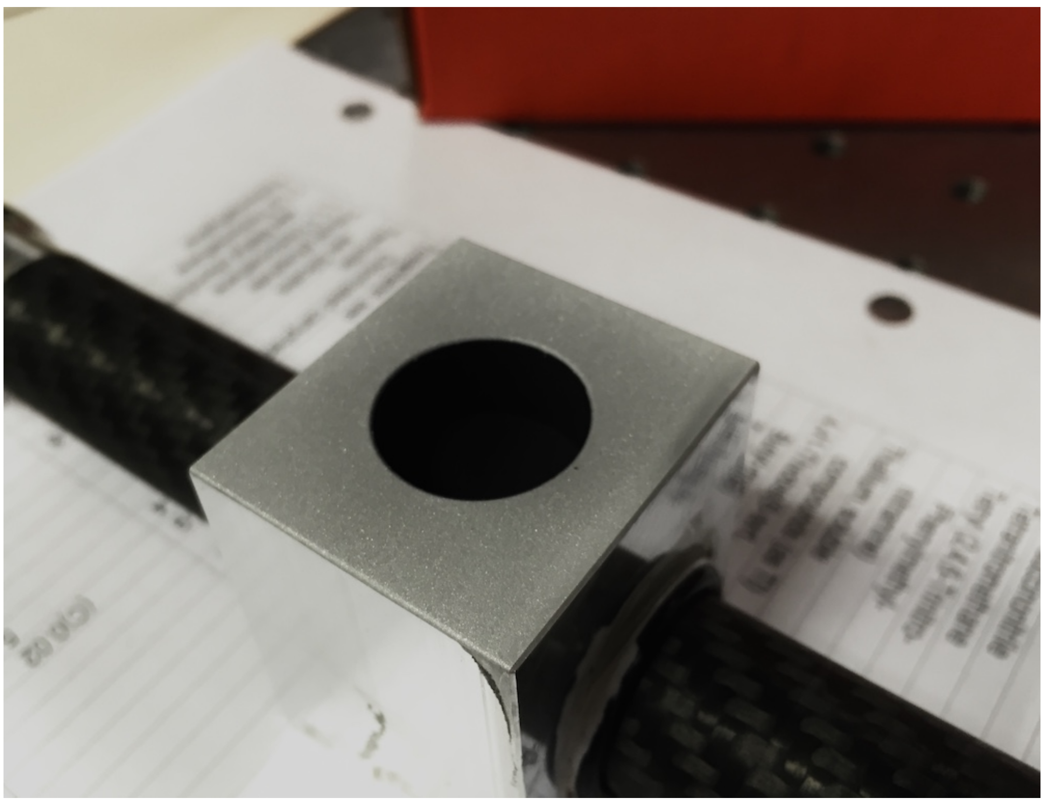


Figure 3.17: Following application of matte black paint, the target blind hole was blocked with a spherical plug, and the outer surface sandblasted to remove excess paint. A diffuse surface is created for optimal contrast against the black target.

After design, the STEP file was passed to College of Engineering machine shop staff for tool path generation and fabrication of the aluminum target blocks on a Computer Numerically Controlled (CNC) mill. Following fabrication, the aluminum blocks were inserted over the tube, placed onto an optical table to maintain a reasonable degree of coplanarity, and permanently secured with a two part catalyzed epoxy. The end caps were then inserted and secured with the epoxy. Following the calibration measurement of the artifact, discussed in Section 3.4.2, the target surface of the artifact was painted with matte black paint. Following the curing of the paint, the target blind hole was carefully blocked with a close fit sphere. Machining of the aluminum blocks resulted in a reflective surface finish, which is undesirable from a photogrammetry standpoint. The aluminum surface was sandblasted to remove excess paint, and create a diffuse surface finish, as shown in Figure 3.17. This results in a high contrast photogrammetry target that can be reliably centroided.

3.4.2 Artifact Calibration

After the preliminary assembly of the artifact, but prior to painting and sandblasting the surface, the artifact was calibrated on a Leitz PMM-F 30-20-16 coordinate measuring machine by graduate student Yue Peng, who handled the QUINDOS programming and machine operation. While the artifact was designed for an approximate length of 1675 mm center to center between the blind holes on each of the blocks, this length is arbitrary, and shifted during final assembly. As such, the length must be well characterized before use as a photogrammetry scale object.

The Leitz PMM-F 30-20-16, shown in Figure 3.18, resides in Siemens Large Manufacturing Lab at the the UNC Charlotte Energy Production and Infrastructure Center, and made available to UNC Charlotte staff and students via a contribution from Hexagon Metrology. The PMM-F allows for precise measurement over a large volume of 3 m x 2 m x 1.6 m. It is housed in a climate controlled laboratory facility, and has been utilized for measurement of everything from turbine blades to large industrial gears.

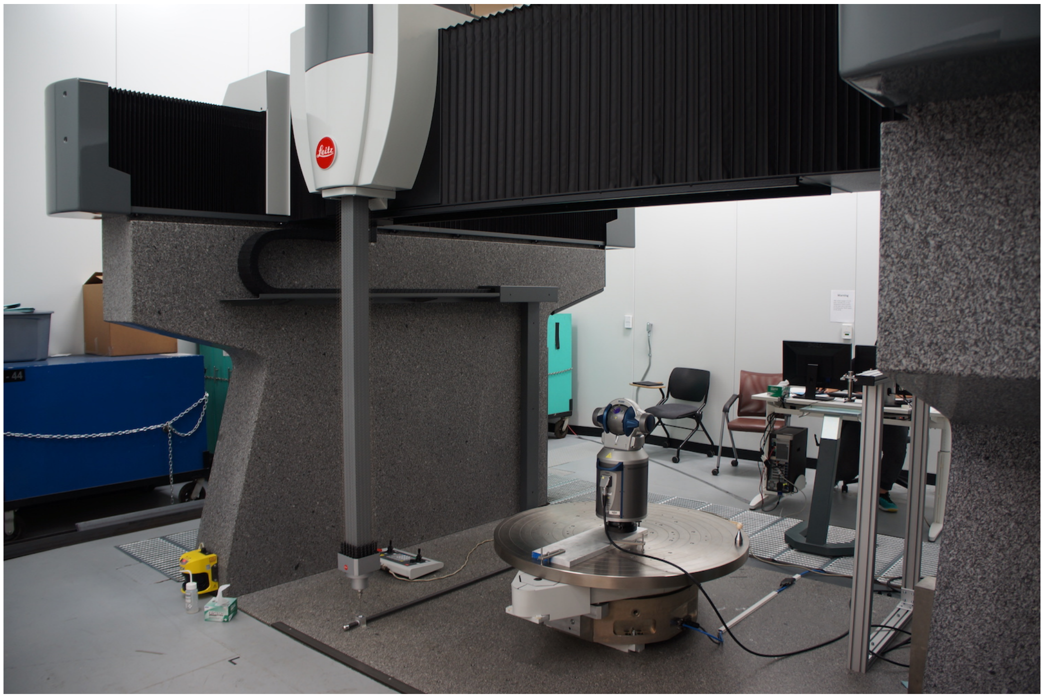


Figure 3.18: Leitz PMM-F 30-20-16 large volume coordinate measuring machine.

The artifact was transported to the Large Manufacturing Lab, cleaned, placed upon the granite surface, and allowed to come to thermal equilibrium over a time period of approximately 2 hours. The artifact was then secured to the granite surface with small tacks of hot glue at the interface of the aluminum block and granite, as shown in Figure 3.19, after which another 2 hours were given to come to equilibrium again. Because of the small size and shape of the artifact target block, a temperature probe could not be easily attached during the measurement test. A probe was instead secured to a block of aluminum placed next to the artifact.

The blind hole target design was chosen to facilitate center to center distance measurement on a coordinate measuring machine. To measure the artifact, the probe traced each face of the cube in order to validate the geometry. A circular trace of the inner circumference was completed at three difference depths to establish a cylinder. The X, Y, Z location was then computed along the cylinder center at depths of $D_1 =$ 3.00 mm, $D_2 = 4.60$ mm, and $D_3 = 6.20$ mm below the surface plane of the target

Figure 3.19: The artifact was oriented along the X axis of the Leitz PMM-F 30-20-16 CMM. Hot glue secures the artifact to the granite surface, and a long probe ensures adequate measurement range to probe the full depth of the target hole.

face of the cube, as shown in Figure 3.20. D_{11} corresponds to a measurement from left target D_1 to right target D_1, D_{22} from D_2 to D_2, and so on for D_{33}. An 80 mm long probe with 5 mm diameter Silicon Nitride spherical probe tip was mounted to a 20 mm center cube. A right hand coordinate system orients the X axis of the CMM along the length of the carbon fiber tube of the artifact.

Following probe path planning in QUINDOS, the artifact was measured 10 times. The temperature over the duration of the measurement was approximately 19.6° C. Initial probe calibration at the time of measurement showed <0.8 μm uncertainty in

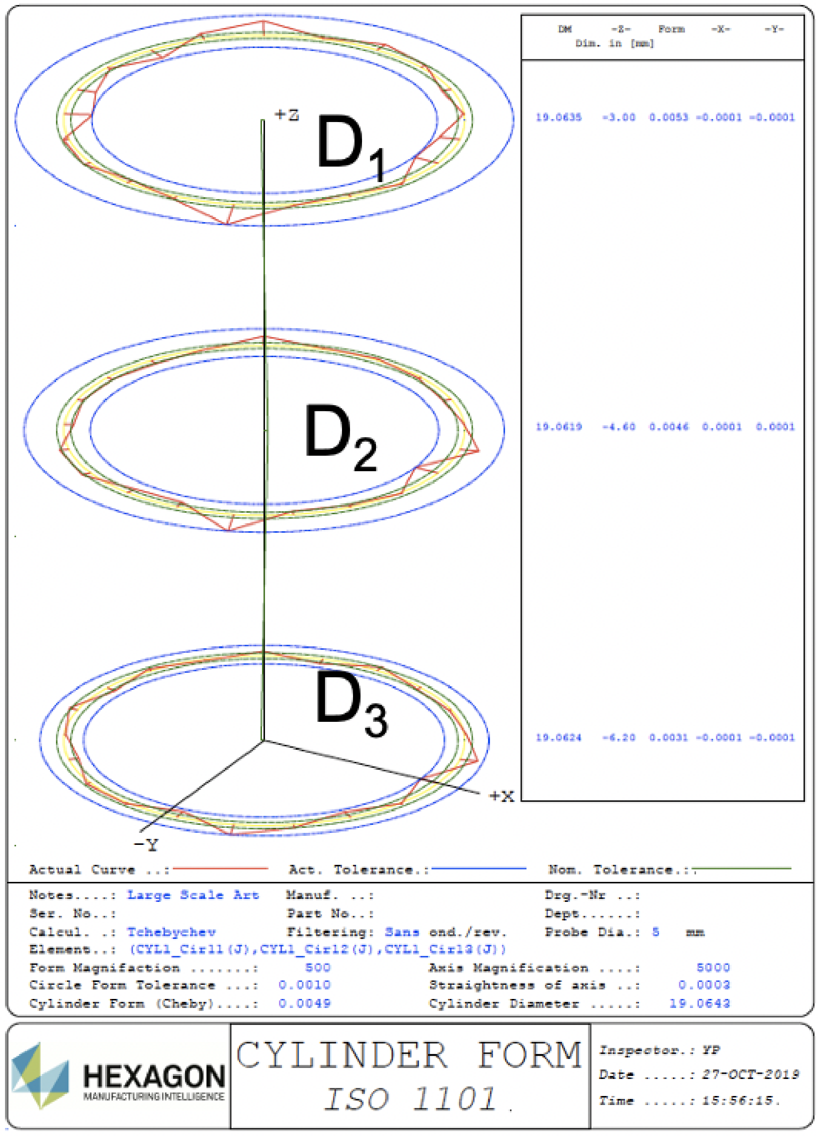


Figure 3.20: The blind holes in the artifact were probed at $D_1 = 3.00$ mm, $D_2 = 4.60$ mm, and $D_3 = 6.20$ mm below the surface plane of the target face of the cube. D_{11} corresponds to the measured length between D_1 of the first target block, and D_1 of the second target block.

probe position, though this value rose to approximately 3 μm during the measurement trial. The most recent calibration (2016) of the Leitz PMM-F 30-20-16 specifies the ISO 10360-2:2009 maximum permissible error $E_{L,MPE}$ of 2.3 μm + (L (mm) / 400 mm) measurement uncertainty along the X axis over the entire range 2 meter range [84, 85]. While the 2016 calibration indicated significantly less error over the

Table 3.1: CMM reported distance and standard deviation of 10 trial measurement set for calibration scale artifact.

Meas.	**10 Trial Mean (mm)**	**10 Trial SD (mm)**
D_{11}	1677.6639	0.0002
D_{22}	1677.6620	0.0002
D_{33}	1677.6609	0.0002

entire range, and despite the artifact being predominantly measured along the X axis, the maximum error is nevertheless assumed over the length of the scale, or approximately 6.5 μm for the 1677 mm artifact. This value is rounded to 10 μm for application of uncertainty to the length of the artifact. Because the CTE for the carbon fiber rod is on the order of $\pm 1 \mu$m/K, thermal effects on the artifact are ignored during use in the lab setting.

The largest distance shown in Table 3.1 is D_{11}, which represents the center to center distance closest to the surface of the cube. Tool deflection during the milling process of a hole can result in a tapered hole profile with sides that are not orthogonal to the base. Additionally, a few micrometers of form error exists such that the center of the cylinder is not orthogonal to the surface of the cube. The reported measurement data from the CMM indicates that this is true in both target blocks; the measured hole diameter of each block is largest at the depth $D_1 = 3.00$ mm below the target face, with a difference of approximately 3 μm in measured diameter between D_1 and D_3, and the distance D_{11} is the largest separation value between targets. As such, and because the visual separation distance of D_{11} will be closest to the distance perceived in an image by a camera, D_{11} is chosen to represent the total target to target length of the calibration scale artifact.

The mean over 10 trials for D_{11} was 1677.6639 mm, with a reported sub-micron standard deviation of 0.2 μm. While a very conservative uncertainty in length of 10 μm was selected for the length of the artifact, measurement repeatability indicates the uncertainty is far lower than that value. As a final value for the calibrated scale artifact, a total length of $L_{Cal} = 1677.66 \pm 0.01$ mm at 19.6° C is reported. The fractional uncertainty of the artifact is approximately 6 parts in 10^6.

3.4.3 Artifact Performance

Using retroreflective targets, a high quality camera system & network, and a well calibrated scale artifact, modern close range photogrammetry systems can deliver scene coordinates with point uncertainty on the order less than 10 μm. Some of these packages, such as the Geodetic Systems V-STARS®, include options for the user to account for the thickness contribution of a retroreflective target film to the point location and overall bundle adjustment. Small details like this matter when high accuracy results are desired. If a photogrammetry system can reach performance levels such that target film thickness must be accounted for, then how would the scale artifact, with three dimensional target surfaces, perform in a typical photogrammetry measurement?

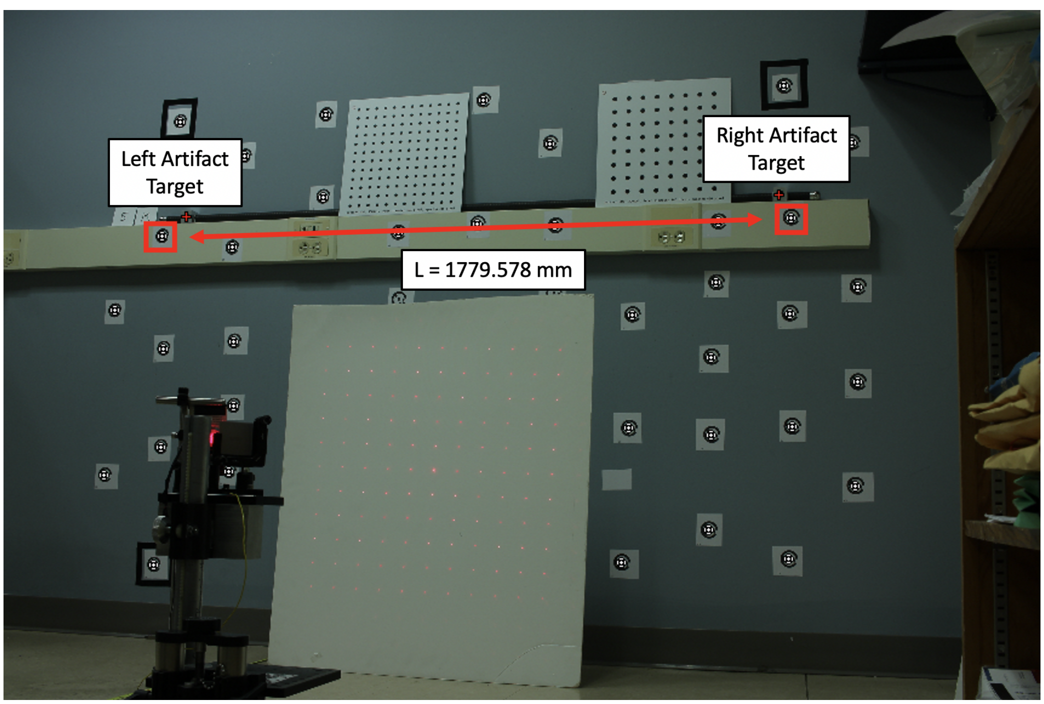


Figure 3.21: To gauge calibration artifact stability, photogrammetry coordinates are scaled with two RAD coded targets which are boxed in red. The calibration artifact target positions, noted with the red '+' symbol, are recorded over 20 measurements.

To allow the calibrated length scale artifact to be measured via CMM, a blind hole target design was utilized. The interior of the target hole was painted matte black in hopes that the low reflectivity of the target would mask the the hole topography and depth when imaged by a camera, appearing similar to a typical ink jet printed target. Prior to using this artifact to calibrate the module, the fabricated artifact's target stability was evaluated in a photogrammetry scene to study the performance of a photogrammetry target with depth, as used on the artifact, versus traditional ink-jet printed targets.

To characterize the stability of the blind-hole artifact target style, the reported coordinates for the RAD targets were examined versus calibration artifact targets, denoted 'Left Artifact Target' and 'Right Artifact Target' in Figure 3.21 and marked with red '+' symbols, over the course of 20 different photogrammetry measurements. In these measurements, only RAD targets and calibration artifact targets were marked while ignoring the optical pattern present in the scene. Two coded RAD targets, #6 and #10, shown in the figure boxed in red under under the left and right artifact targets respectively, are used to scale the delivered coordinates after the photogrammetry bundle adjustment. A length of approximately 1799.58 mm is assigned between the two RAD targets, a value measured via photogrammetry using the calibration scale artifact. The exact numerical value is arbitrary, as an order of magnitude measurement of the relative stability of the artifact's blind-hole type targets is adequate.

20 photogrammetry based measurements of the calibration artifact were conducted, and the results plotted in Figure 3.22. The artifact's reported mean length between the left and right targets was found to be 1677.66 mm, with a calculated standard error of the mean of less than 3 μm. The standard deviation of the 20 measurement set was 0.01 mm. There does not appear to be a bias or trend in the data.

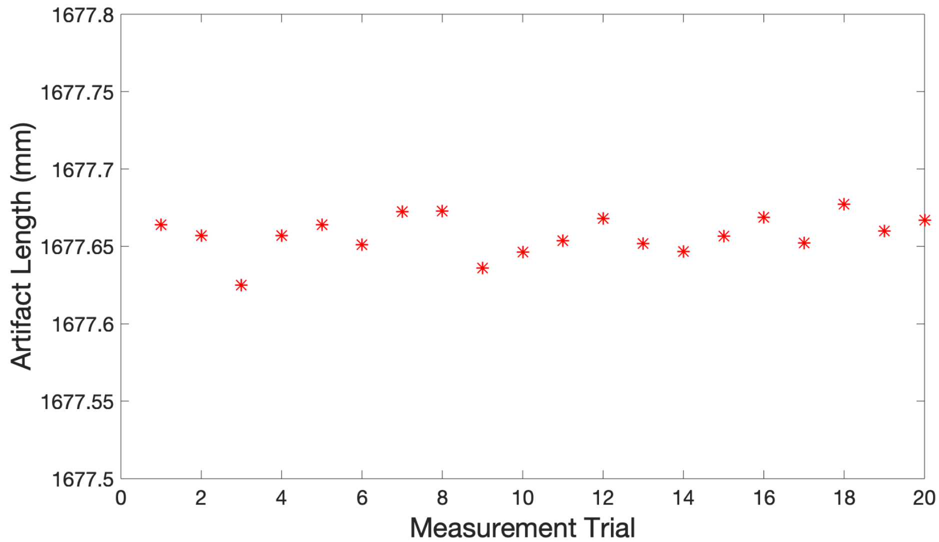


Figure 3.22: Mean Euclidean distance between the left and right targets on the calibration scale artifact.

Table 3.2: Reported standard deviation in X,Y,Z coordinates and P_X, P_Y, P_Z coordinate uncertainty over 20 measurements. Avg. RAD is the average over 46 coded targets visible in the scene. Artifact L and R are the values for the left and right side artifact targets, respectively.

Units (mm)	σ_X	σ_Y	σ_Z	σ_{P_X}	σ_{P_Y}	σ_{P_Z}
Avg. RAD	0.02207	0.01224	0.01739	0.00216	0.00120	0.00118
Artifact (L)	0.01156	0.00927	0.01550	0.00152	0.00092	0.00091
Artifact (R)	0.01792	0.01153	0.02806	0.00271	0.00224	0.00135

Shown in Table 3.2 is the standard deviation for X,Y,Z coordinates and point precision P_X, P_Y, P_Z, or the uncertainty in coordinates, as reported by PhotoModeler® after the bundle adjustment. While values should be rounded to the nearest 10 μm, it can be observed that the standard deviation of both calibration artifact targets is less than the average for the coded RAD targets in the X and Y dimensions. Only the Z dimension of the right side of the artifact performs worse on average than the RAD targets, which is surprising as the stated uncertainty P_Z for the right artifact target is lower than P_X or P_Y. Speculation is offered that the nature of the target being furthest away from two of the camera stations, thereby having a reduced convergence angle ($\approx 63°$) between camera stations, could have led to poorer performance compared to the left target (≈ 69°). The reported location of the right side target of the artifact for all 20 measurements is plotted in Figure 3.23, with a pattern that appears relatively random. Following the measurement trials, it can be concluded that the three dimensional target face appears planar in the images, and does not display any reported depth variation in the X that would indicate poor performance compared to traditional ink-jet printed targets. With its stability as a scale artifact confirmed, the artifact was confidently used in the pattern calibration process described in the next section.

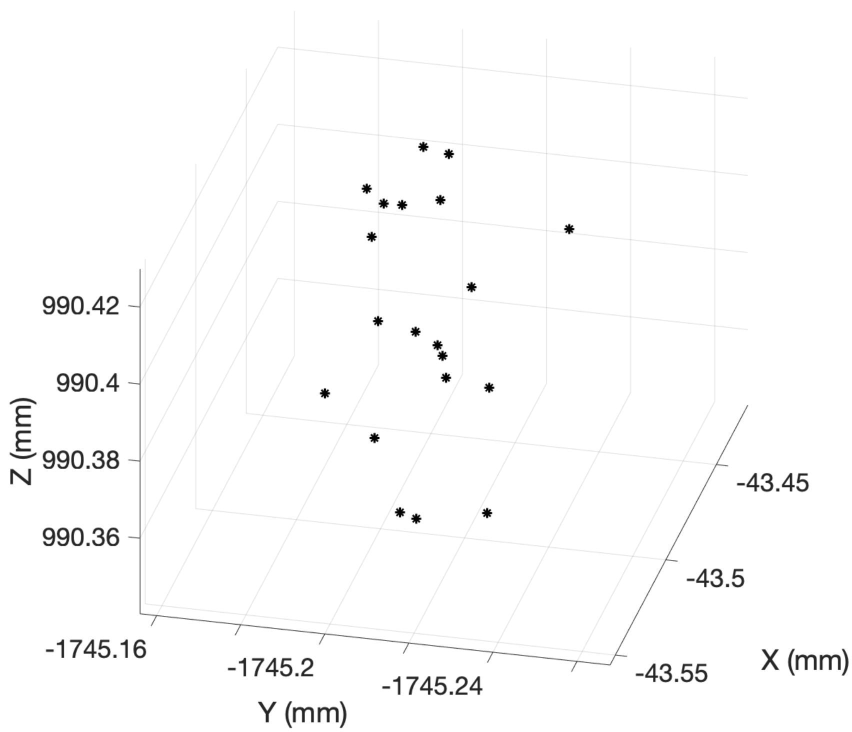


Figure 3.23: Reported right side target position of calibration scale artifact for 20 measurements.

3.5 Pattern Calibration Process

Three calibrations were performed on the module over a period of four months. The calibration process is time intensive; while each calibration measurement can be performed in approximately 20 minutes, the data post-processing is on the order of 15 hours per calibration. The general process of a single typical calibration is described here, and the specific results of each of the three calibrations are discussed in detail in Section 3.6.

A preliminary first calibration of the 121 point optically projected patterns was conducted with three additional targets added to the scene. A single coded RAD target served as the origin, and the two other targets on the calibration scale artifact served as a source of scale. All three were used to establish the coordinate system. A triangular orientation was established for the camera positions in the field, approximately 120° apart on a circle to optimize the camera network.

In the final two calibrations, an additional 51 RAD targets were added to the scene to improve the photogrammetric measurement stability. The camera network was changed to a four camera design in order to increase the convergence angle, and improve performance. The improved four camera network was utilized for the module's test measurements. A complete description of the camera network, computer-based image acquisition process, measurement description, and other photogrammetry setup details can be found in Chapter 4.

The reader may recall that for each calibration, the beam pattern is sampled at ten different distances, or 'slices', from the module. The movable screen is first placed approximately 0.75 m from the module. Each subsequent slice is taken at an additional 6.5 cm away from the module, resulting in a final module to screen distance of approximately 1.35 m. Sampling the 11 x 11 pattern over those 10 slices, a total of 1,210 laser projected data points are acquired per camera station. With four image stations and two grids requiring calibration, the data burden quite large at 9,680 marked points per calibration attempt. Each projected grid is calibrated separately, as shown in Figure 3.24, with the other grid pattern being temporarily blocked with a non-contact barrier suspended by a movable optical post.

Figure 3.24: Pellicle grid calibration. During the calibration process, each projected grid from the module is sampled individually to simplify data processing. A movable screen placed at 10 different positions allows for a line to be best fit to each beam from resulting coordinate data.

Each of the 10 slices is processed in PhotoModeler®, and the optical pattern coordinates scaled by the calibration length scale artifact. The output coordinate data is then sorted by beam number, placed into a multi dimensional array, and loaded into the MATLAB® module pattern calibration script, whose operation is detailed in Section 3.3. Complete code can be found in Appendix A.4.

Shown in Figure 3.25 is a collection of raw grid data from 10 different movable screen positions. While the curvature of the screen is visible, the ultimate position of the projected spot in space is not critical. As the algorithm proceeds through its evaluation of the data, it will converge upon the location of the module away from the global coordinate system origin. Upon solving for this translation in position, the data set is then moved to the global origin and aligned, as shown in Figure 3.26. In the figure, the blue data set represents raw uncalibrated data slices whose convergence point is located away from the global origin. The red data set shows the calibrated

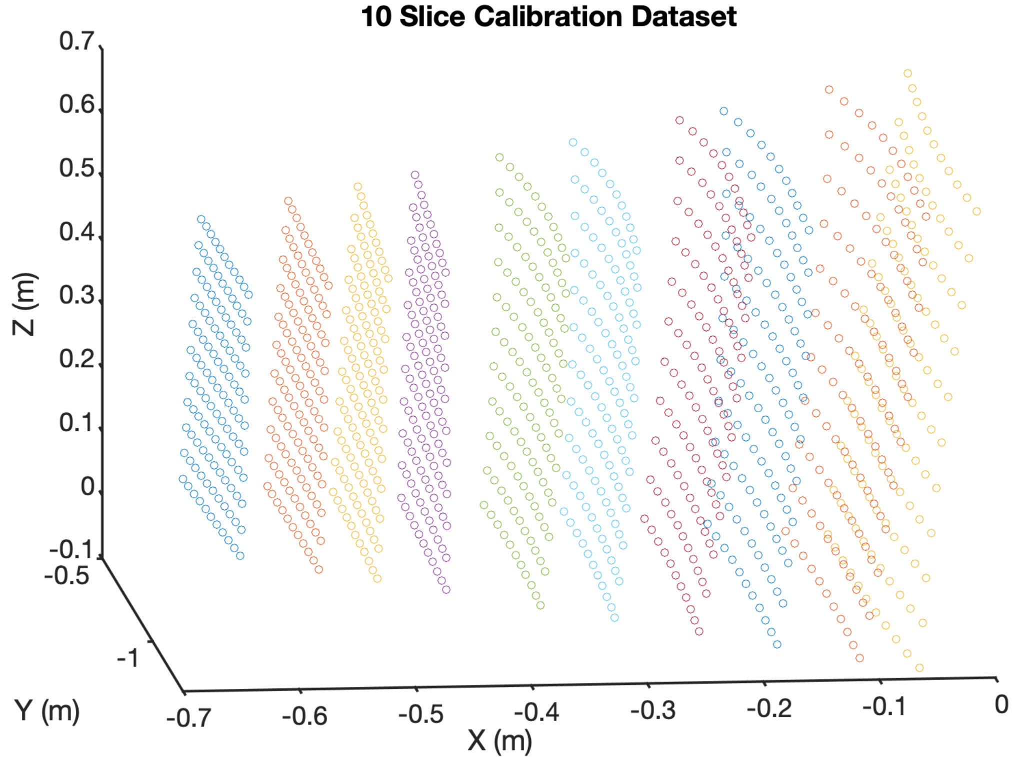


Figure 3.25: 10 slices of raw coordinate data obtained during grid calibration.

pattern, whose location and pose has been determined, allowing for transformation of the data set such that the intersection of all beams, or beam projection origin, is now located at (0,0,0) in the global coordinate system, and the relative pose and projection of the center beam has been aligned with the +X direction.

The spherical coordinates for all 121 beams in 10 slices are determined. Each beam cluster is then averaged, and a mean value pointing direction is determined. A final 2 x 121 array containing the average pointing directions for each beam is output, along with the module's location and pose relative to the global coordinate system. Once the calibration has been performed for each grid, the Euclidean distance between projection origins can be solved, and the real module separation length is determined.

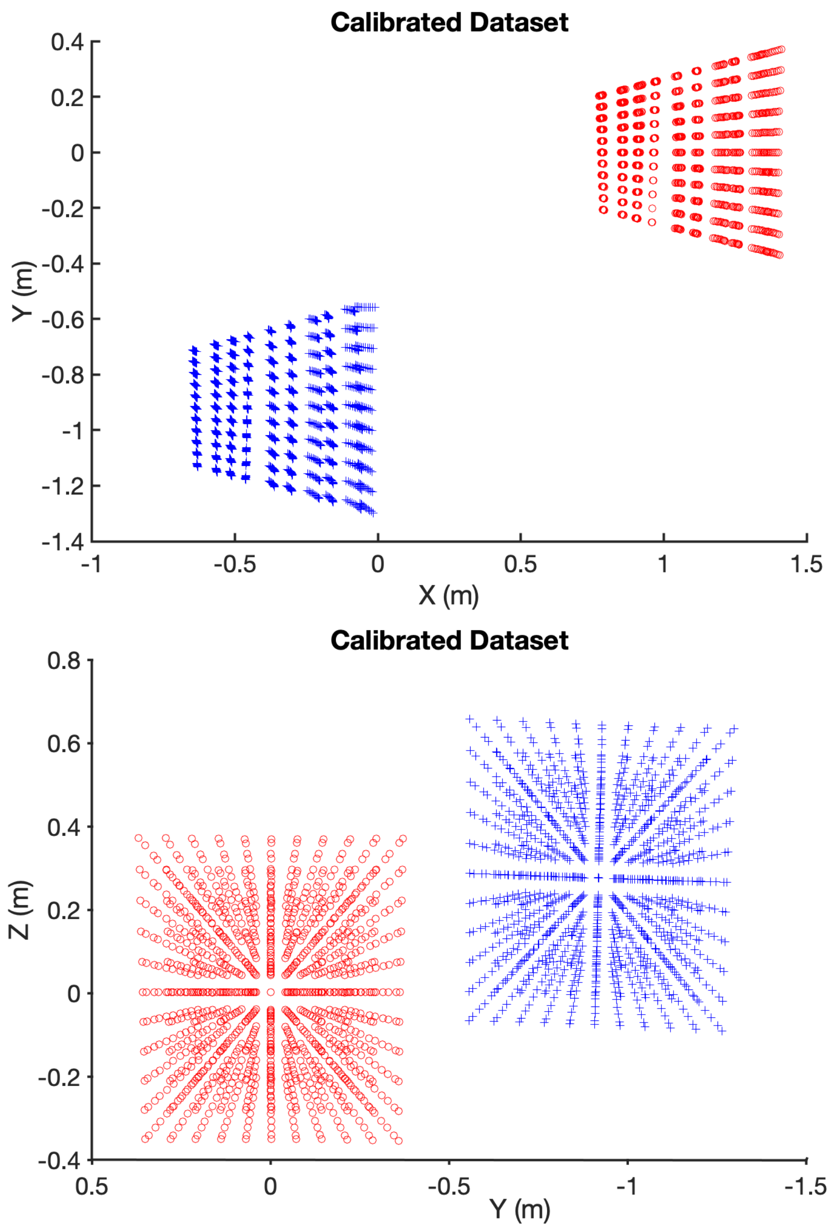


Figure 3.26: Blue data plots show the raw uncalibrated data set, translated and rotated away from the global coordinate system origin (0,0,0). The calibrated data set, shown in red, has been solved for location and pose, then relocated to the global origin such that spherical coordinate pointing directions can be assigned to all beams in the pattern.

3.6 Pattern Calibration Results & Uncertainty

Three calibrations were completed over a period of four months. After the first calibration, changes were made to improve the performance of the photogrammetry system, as detailed in Section 3.5. The location and pose of the projection origins of the module were solved during each calibration relative to the global coordinate system established during processing. Those module origins are expressed as Projection Origin 1 & 2, or PO_1 and PO_2 in data tables.

3.6.1 Calibration 1

The preliminary calibration, or Calibration 1, was performed prior to improvements to the photogrammetry system. Notably, the calibration took place over a duration of 10 hours. Due to concerns regarding the detection of points in an image, the question arose whether missed data points in the middle of the calibration process would jeopardize the entire calibration during post-processing. Accordingly, each calibration slice was imaged one at a time. Following imaging, the slice was immediately fully processed in PhotoModeler®prior to moving on to the next data slice. Shown in Table 3.3 are the results of the first calibration.

Table 3.3: Solved location and pose relative to global coordinate system for Calibration 1.

	X (mm)	**Y (mm)**	**Z (mm)**	$\boldsymbol{\alpha}$ (rad)	$\boldsymbol{\beta}$ (rad)	γ (rad)
PO_1	-1612.30	-852.96	341.11	-0.01614	-0.04411	0.01627
PO_2	-1650.81	-891.48	341.02	-0.02384	-0.03655	0.03008

Calibrated Separation Distance	54.47 mm

Following the preliminary calibration, a test measurement was performed as per the process outlined in Section 4.5. After processing and scaling of delivered point coordinates, the measurement artifact reported a length of approximately 1700 mm. This is slightly under an inch longer than the artifact's nominal calibrated length, and in the realm of error on the order of low parts in 10^2. This exceeds the expected error of the module by an order of magnitude.

It was assumed that errors existed somewhere in the calibration process or photogrammetry system. In the case of the latter, additional targets were added, the camera network was supplemented with an additional camera, all cameras were positioned for better convergence angles, and an improved camera calibration was completed on each of the cameras.

Potential error sources were examined in the processed calibration data. Incorrect or biased beam pointing directions would impart a bias upon the optically projected scale length and result in scene coordinates being incorrectly scaled. Or, if during the calibration process the module's origins were incorrectly solved, then the module would also suffer from a systematic bias in its length scale. Errors in the calibration are probed in the following manner...

The calibrated pointing directions ϕ, θ for each beam are used to construct a line of arbitrary length from the projection origin out into the field. The 10 points which were utilized during the calibration process to determine the pointing direction of each beam are then plotted along that arbitrary length line. The point-to-line distance d of each of the ten points relative to the line is then calculated:

$$d = \frac{|(\mathbf{c} - \mathbf{a}) \times (\mathbf{c} - \mathbf{b})|}{|\mathbf{b} - \mathbf{a}|},$$

where

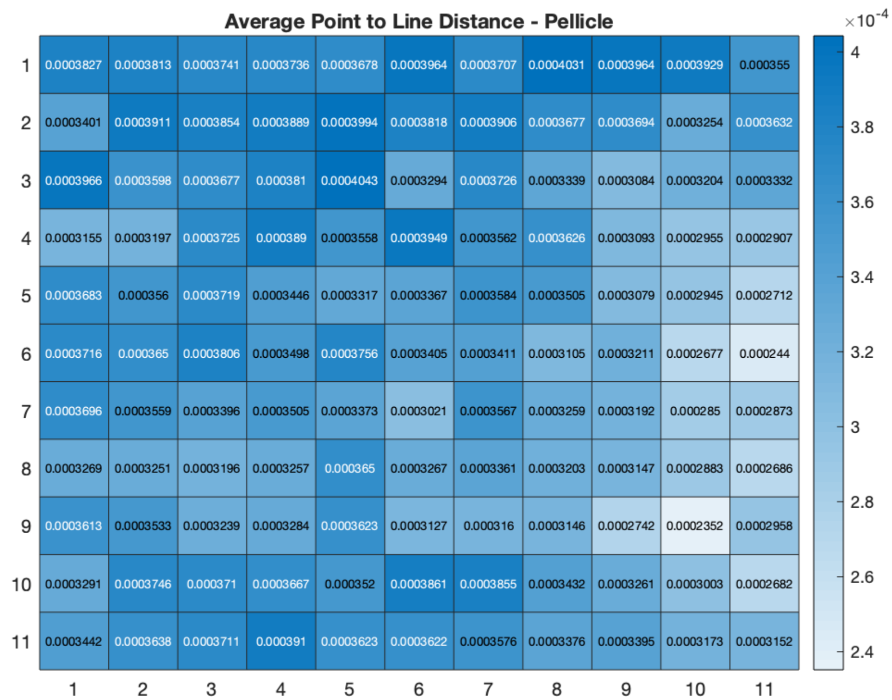
d is the distance from the point to the line,

a is the origin,

b an arbitrary point away from the origin describing the pointing direction, and

c is the coordinates of a beam point utilized in the calibration of a beam.

Following this process, the mean distance is computed for all ten points off the line as a measure of the quality of fit of the points to the supposed beam pointing direction. This mean distance value is determined for every beam in the grid and then plotted in an error map, shown in Figure 3.27. As is the standard order, projected beams are numbered from 1 to 121, ordered right to left, then top to bottom. An error map is generated for each projection pattern.



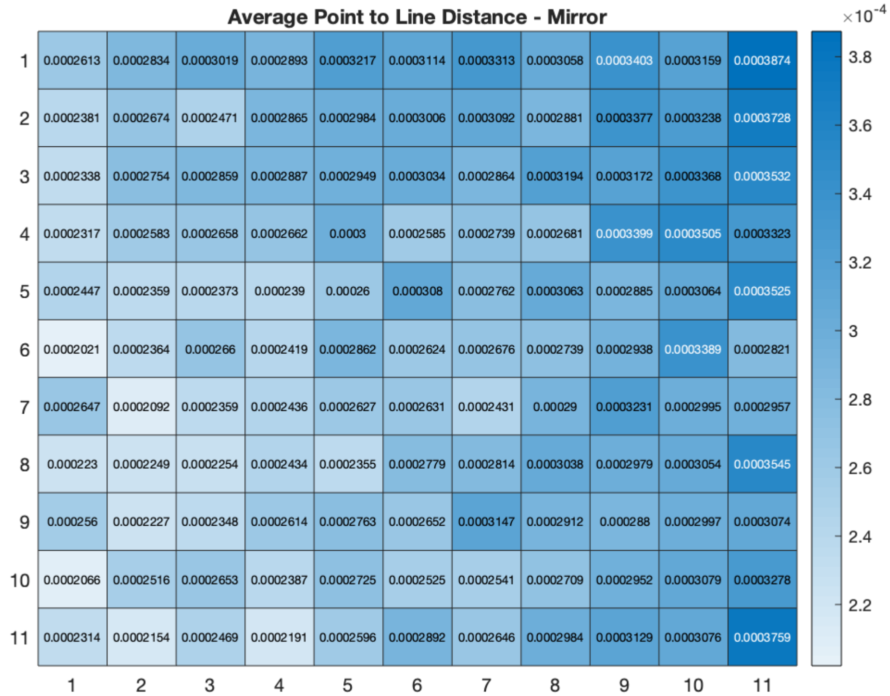
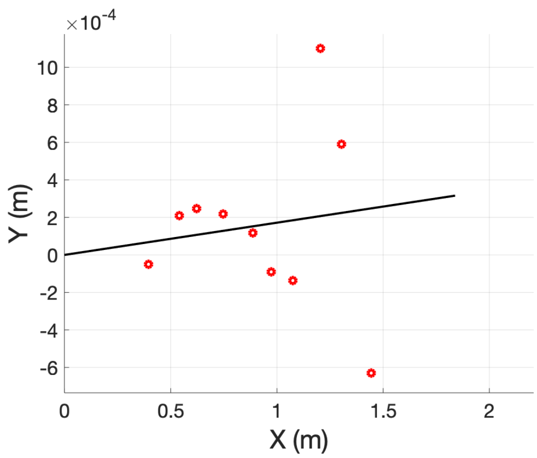


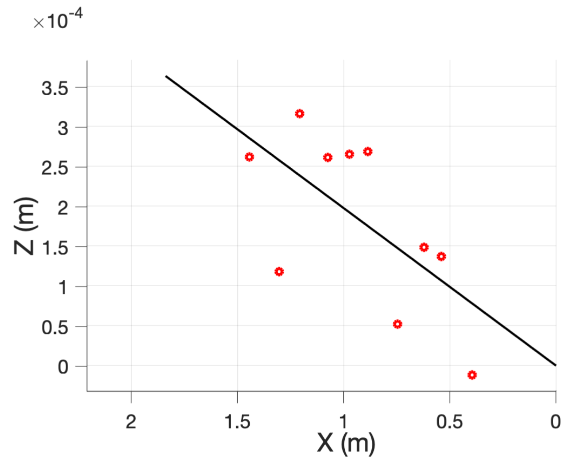
Figure 3.27: Calibration 1 average point-to-line distance for beam clusters from the line projected along pointing direction of each beam.

Assessing all points in the grid, the collective point-to-line mean for each beam cluster for the pellicle side, or Projection Origin 1, is 0.28 mm with a standard deviation of 0.04 mm. For the mirror side, or Projection Origin 2, the collective average across all beams is 0.34 mm with a standard deviation of 0.04 mm. In the pellicle error plot, increased average distance values having a leftward bias are observed, while the opposite is true for the mirror side. In addition to the error map, a plot was generated for each beam of the 10 points plotted alongside the arbitrary pointing direction line.

In addition to the error plots for the collective grid, attention should be given to the scatter of calibration points for each individual beam in the grid. Shown in Figure 3.28 are plots of the central beam (61 of 121) of the pellicle side grid. An arbitrary length length extends from $(0, 0, 0)$ about the pointing direction of the beam, which in this case should nominally be directly along $\hat{X}$. Examination of the plots reveals



(a)



(b)

Figure 3.28: Center beam (#61) pointing line with 10 data points. Calibration 1, pellicle side pattern.

significant deviation away from the line. The pattern repeats on the mirror side, suggesting possible drift in the coordinate system, module, or kinematic mounts, rather than instability in the diode source. It is of note that this initial calibration was performed over the course of an entire day, from approximately noon until late into the evening. The room is not rigorously climate controlled and it is suspected that thermal effects were responsible for the bulk of the error, although the nearly 1.5mm swing in Y position from slices 8 to 10 is questionable, and was duplicated across other beams in the grid.

3.6.2 Calibrations 2 & 3

Following analysis of the initial calibration, changes were made to the photogrammetry system as previously described in this section, including an additional camera, redesigned network, improved camera calibrations, added targets etc. In addition, lab lighting conditions and camera exposure times were standardized to values which resulted in very consistent automated marking and centroiding during post-processing of data. This allowed the full calibration measurement to be rapidly performed with post-processing performed after all ten slices were imaged, rather than collecting a single slice and processing before moving on. As a result, the calibration measurement duration was reduced from approximately 10 hours for both grids to 15 minutes. Additionally, while only a single image is needed per camera at a given screen depth, three image sets per station per slice were taken for redundancy, in case an issue should arise with a given image during post-processing.

Two calibrations were performed, Calibration 2 and Calibration 3, under these conditions. Shown in Table 3.4 are the computed results for the second calibration. Note that different X,Y,Z values are expected as the module was moved during this time relative to the established global coordinate system origin. In addition, the module's platform was changed from a camera tripod to a large optical board with a gear-driven elevator mount as shown in Figure 4.12 in Section 4.4. The separation distance between PO_1 and PO_2 was found to be approximately 55.17 mm, around 0.7 mm longer than the value determined during Calibration 1. This $\approx 1.3\%$ difference is significant with respect to correctly scaling objects in the field and roughly corresponds to the percentage of error in the test measurement following the first calibration.

Table 3.4: Solved location and pose relative to global coordinate system for Calibration 2.

	X (mm)	**Y (mm)**	**Z (mm)**	$\boldsymbol{\alpha}$ (rad)	$\boldsymbol{\beta}$ (rad)	$\boldsymbol{\gamma}$ (rad)
PO_1	-1337.34	-728.30	274.98	0.02998	-0.00099	0.02633
PO_2	-1376.11	-767.56	274.66	0.02228	0.00660	0.04088

Calibrated Separation Distance	55.17 mm

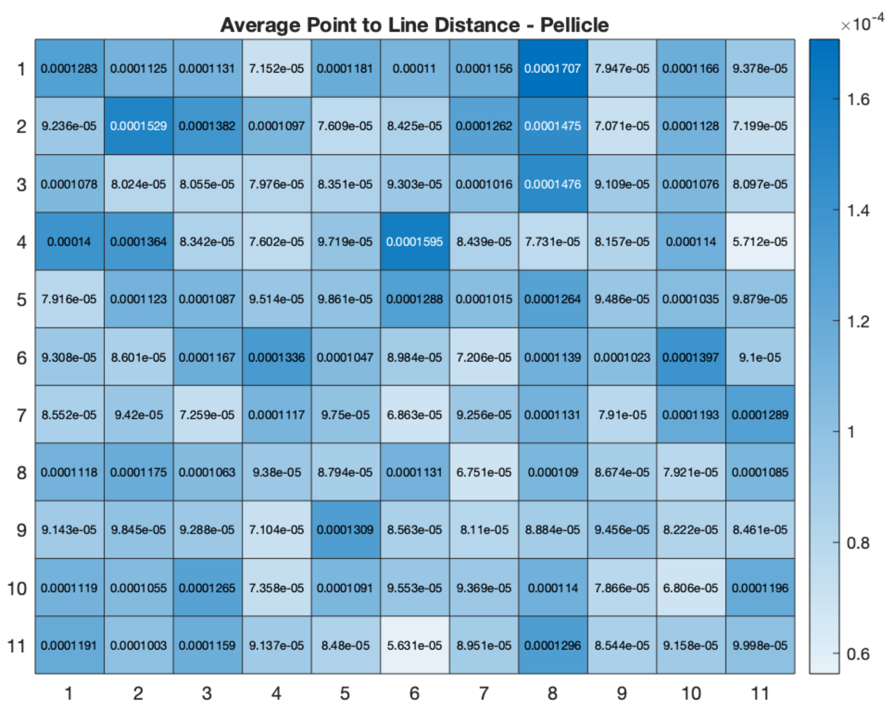
A third calibration was performed approximately one month after the second. The camera network remained the same, though additional RAD targets were added to the scene again to increase target density, with a final total target count of 52. The results for this calibration can be see in Table 3.5. Note again that the X,Y,Z position and pose varies from the previous calibration as the module was again moved. The calibration separation distance for Calibration 3 differs from Calibration 2 by approximately 10 μm.

Table 3.5: Solved location and pose relative to global coordinate system for Calibration 3.

	X (mm)	**Y (mm)**	**Z (mm)**	$\boldsymbol{\alpha}$ (rad)	$\boldsymbol{\beta}$ (rad)	γ (rad)
PO_1	-1421.73	-920.82	275.58	0.02943	-0.00057	0.00259
PO_2	-1461.41	-959.15	275.53	0.02194	0.00740	0.01711

Calibrated Separation Distance	55.16 mm

A heat map was again created showing the average point-to-line distance for both grid calibrations for Calibration 3, shown in Figure 3.29. Assessing all beams in the grid, the average over the pellicle side pattern, or PO_1, was 0.10 mm with a standard deviation of 0.02 mm. For the mirror side, or PO_2, the average across all beams was 0.10 mm, with a standard deviation of 0.02 mm. Compared to the Calibration 1, the mean deviation of calibration points off the pointing direction line over all beams is $1/3^{rd}$ the value of Calibration 1, with half the standard deviation. There no longer appears to be a bias in either pattern error map.



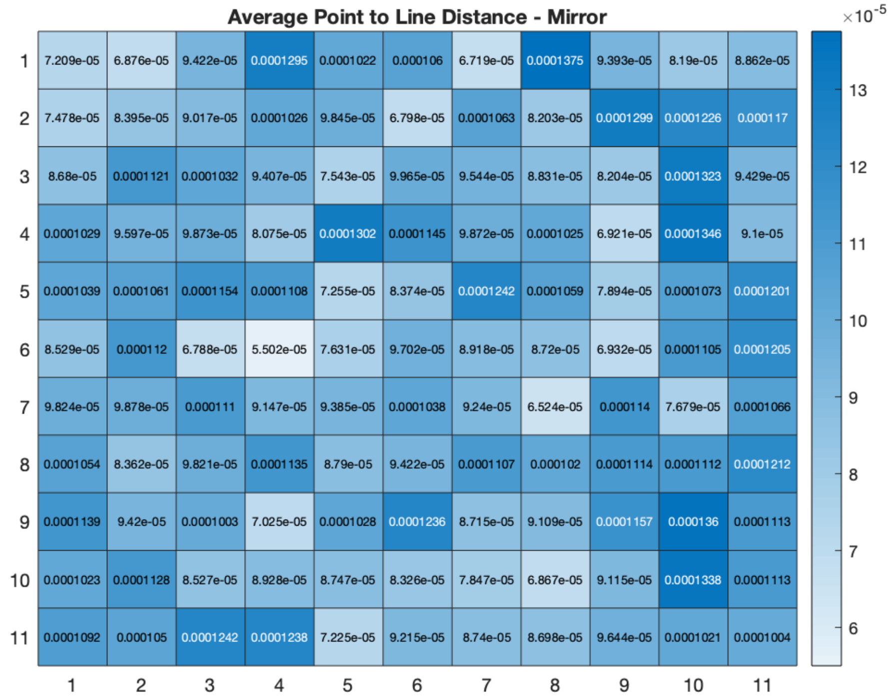
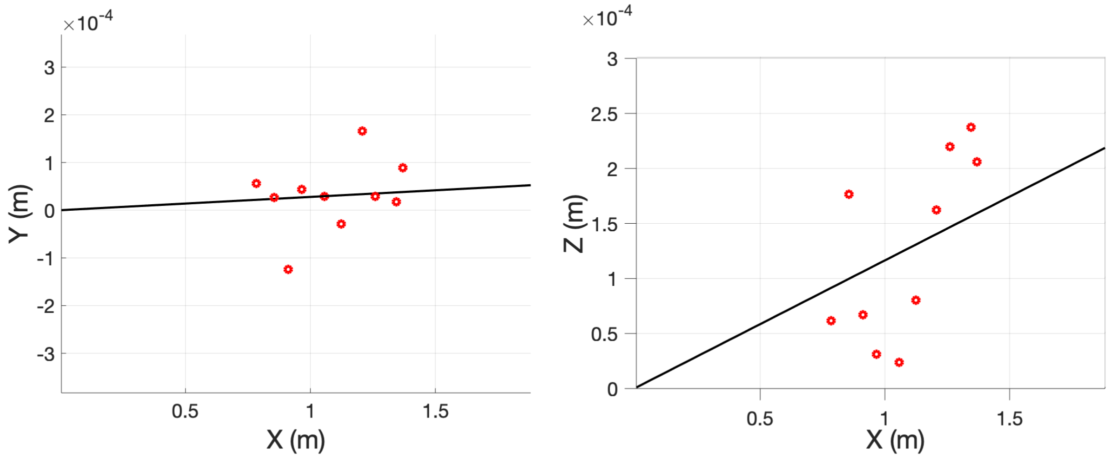


Figure 3.29: Calibration 3 average point-to-line distance for beam clusters from the line projected along pointing direction of each beam. Mean value for both patterns of 0.10 mm with SD of 0.02 mm.

Examining Calibration 3's center beam for the pellicle side (PO_1) grid, as shown in Figure 3.30, a significantly tighter distribution is observed in the XY plane. The heat map affirms this observation; Calibration 1 had an average point-to-line value of approximately 340 μm. In Calibration 3, that value is reduced to approximately 90 μm.

3.6.3 Pattern Calibration Uncertainty

During the pattern calibration process, the pointing directions of each beam in the pattern were solved, as well as for the module's location and pose in the global coordinate system. The uncertainty in those parameters is difficult to directly measure. While each calibration takes less than 15 minutes to measure, post-processing of the data can take on the order of 12 to 15 hours per calibration. The time burden of a traditional Type A uncertainty analysis would be staggering, while the Type B



(a) (b)

Figure 3.30: Center beam (#61) pointing line with 10 data points. Calibration 3, pellicle side pattern.

error propagation analysis presents other difficulties [86]. As a number of module parameters such as beam number and grid full angle are not only non-linear in their contribution to the overall uncertainty, the algorithmic nature of the solving process makes their error contribution difficult to directly assess.

A Monte Carlo simulation provides an alternative method by which uncertainty can be estimated for a complicated model. In addition to the standard GUM, the Joint Committee for Guide in Metrology (JCGM) Supplement 1 to the GUM (JCGM/101) provides guidance on propagation of distributions through models using Monte Carlo simulations to evaluate uncertainty [87]. During the calibration process, the only value measured is the position of laser spots in the field, and this is done via photogrammetry. Uncertainty in other parameters does exist; changes in the source wavelength, for example, results in angular changes in the diffracted beam orders in the grid. During a measurement, however, the only quantity that can be assessed is beam position. Because of this, the assumption is that due to the short duration of the measurement, on the order of 7-8 minutes per grid, the drift due to changes in the environment and module are minor.

The dominant source of uncertainty in the calibrated location and pose of the projection origin is thus assumed to be from the impact of speckle on determination of spot location. While the speckle influenced intensity profile of each beam spot is not necessarily Gaussian, and while the perceived location itself varies based upon viewing location, the scatter in perceived location is assumed to be approximately normally distributed over the entire 121 point pattern.

While the uncertainties in other module components, such as the source, are examined in Chapter 5, a comment is made here that the diode source exhibits an angular fluctuation at steady state operation on the order of 10 μm at 1 m projection distance. This is approximately an order of magnitude below the reported uncertainty of beam spot locations as determined in PhotoModeler®, and is thus neglected in the Monte Carlo simulation used to determine uncertainty in the solved projection location and pose. The other module parameter, beam number, is of course fixed.

An initial Monte Carlo simulation was performed to examine the impact of noise in beam spots upon the solved projection coordinate location and pose of a projection source in the calibration process. The code for this simulation can be found in Appendix A.5. A single module was created slightly offset from the origin, and 10 calibration slices generated at a screen distance of 1 to 10 meters, with a slice every meter. The projection source generated an 11 x 11 grid with a pattern full angle of $\theta = 29°$. A normal distribution of random noise was multiplied by the given spot coordinate uncertainty for each iteration. To reduce the computational burden, 500 total iterations were performed at each spot coordinate uncertainty value, over a spot noise range of 10 μm to 150 μm.

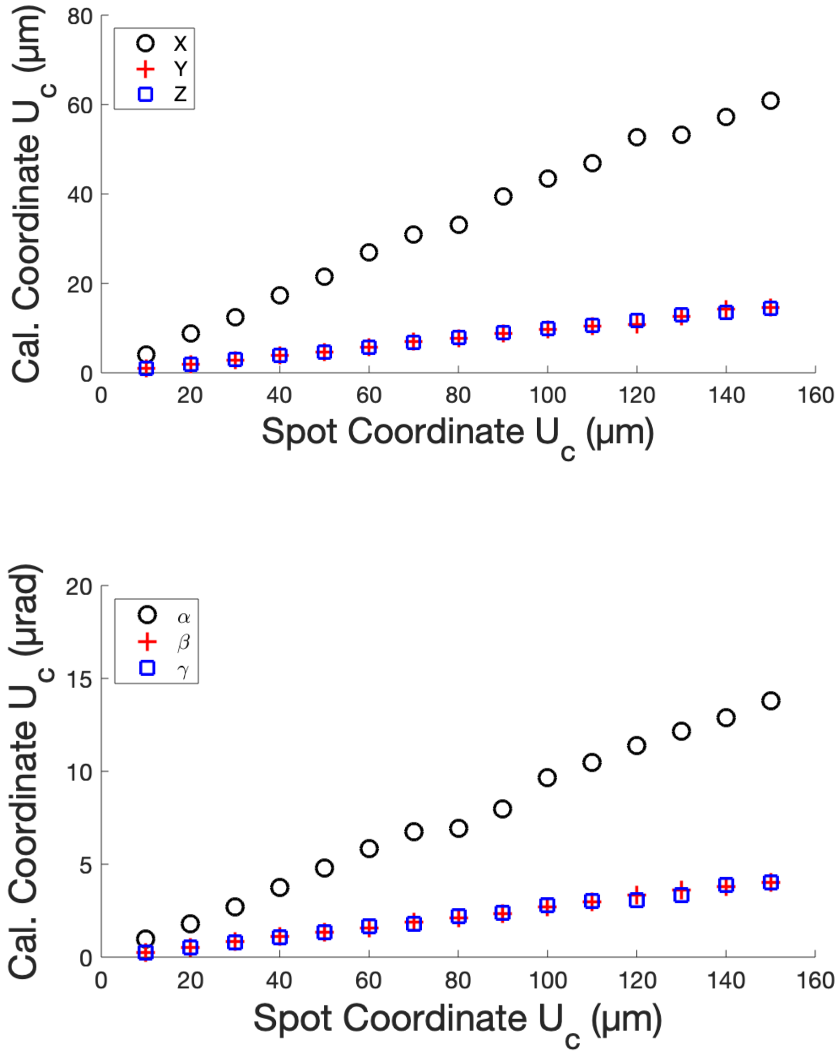


Figure 3.31: Monte Carlo simulation of uncertainty in calibration projection origin X,Y,Z coordinates and α, β, γ pose. 500 iterations per point, n=11, $\theta = 29°$.

Shown in Figure 3.31 is spot coordinate uncertainty versus the resulting uncertainty in the solved calibration location coordinates and pose. Even with approximately 80 μm of spot noise on each dimension, the solved projection calibration's location shows a standard deviation in the X of approximately 33 μm, and 8 μm in the Y and Z coordinates. Similarly, only approximately 7 μrad variation in α, and 2 μrad in β and γ are observed.

Leveraging a fit of multiple points is effective in reducing the overall uncertainty associated with the solved projection origin and pose. Individual coordinate uncertainty is significantly lower than the uncertainty applied to spot coordinates. The reader may recall that the merit function fits not just a single beam in the grid, but all beams simultaneously. This further serves to reduce the impact of a single bad point on an individual beam, as well as the entire collection. It is noted that calibration coordinate uncertainty in the Y and Z is approximately the same, which is expected. Larger uncertainty in the X coordinate position is also expected, as the relatively narrow ($\theta \approx 29°$) full angle of the grid allows variation in X with small impact on Y and Z. With an increased grid full angle, it is expected that the X calibration coordinate uncertainty would decrease and approach levels seen in the Y and Z coordinates.

Indeed, this is observed when the same test is performed with a 60° full angle grid. Again, an 11 x 11 grid is projected every meter over a distance of 1 to 10 meters, and 500 iterations are computed per given spot coordinate uncertainty value. Shown in Figure 3.32 are the results of the Monte Carlo simulation for the parameters described above. At 80 μm of spot coordinate uncertainty, the simulation shows approximately 15 μm calibration coordinate uncertainty in the X, and 7 in the Y and Z. The Y and Z values show little change in the 60° grid as compared to the 29° grid, but the X uncertainty decreased by more than half, from 33 μm to 15 μm. A similar reduction in pose uncertainty is seen, where the roll about the X, α, decreases by more than half from 7 μrad to approximately 3 μrad, but shows little change in β and γ. Slightly higher α values in roll about the X can likely be attributed to pattern symmetry, as slight misalignment has less overall impact on the merit function compared to roll about β and γ.

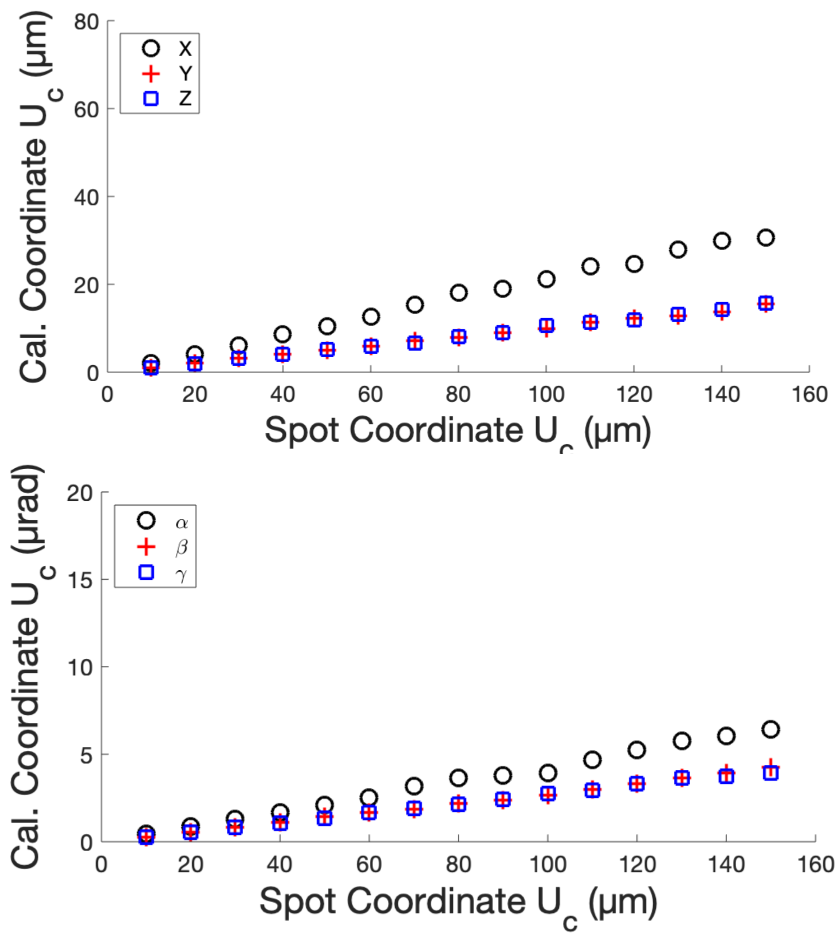


Figure 3.32: Monte Carlo simulation of uncertainty in calibration projection origin X,Y,Z coordinates and α, β, γ pose. 500 iterations per point, n=11, $\theta = 60°$.

A final simulation is run utilizing the current module parameters. An alternate MATLAB® script, found in Appendix A.6, was written to perform a Monte Carlo simulation with specific X,Y,Z spot noise. This simulation used an 11 x 11 grid with with full angle of 29°, with beam spot noise of 110 μm, 75 μm, 75μm in the X,Y,Z respectively. The 10 slices of beam coordinates are modeled at equally spaced intervals between projection distances of 0.75 m and 1.5 m from the module.

A 2000 iteration simulation was completed under these conditions, with reported uncertainty in the solved calibrated projection origin of 52 μm, 12 μm, 12 μm in the X,Y,Z respectively, and 35 μrad, 13 μrad, and 13 μrad in the α, β, γ. Compared to the original 80 μm spot noise simulation, the slight increases in X,Y,Z,α, β, γ can

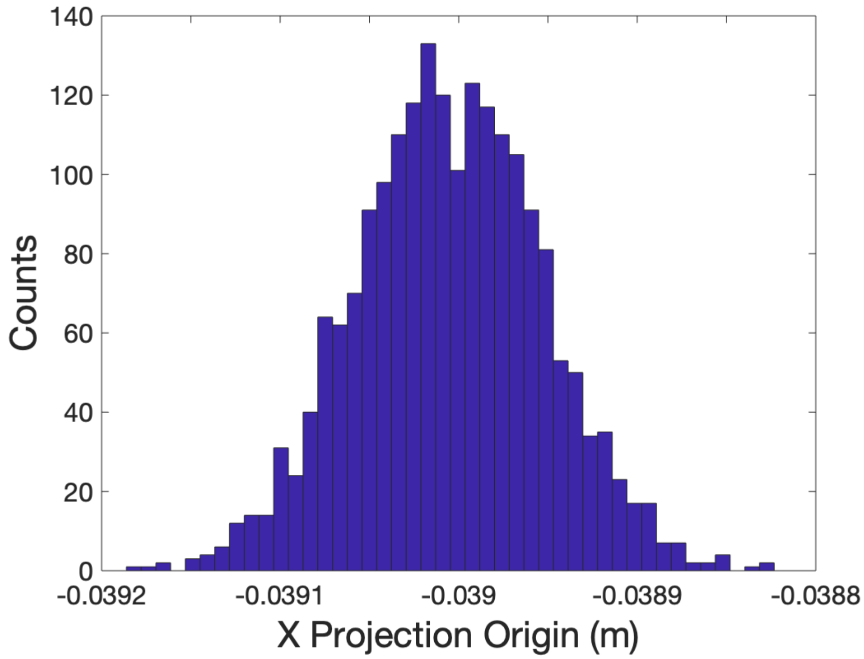


Figure 3.33: Histogram of calibration projection origin's X coordinate. 2000 iteration Monte Carlo simulation. SD $\approx 52\mu$m. N = 11, $\theta = 29$. Spot noise 110 μm X, 75 μm Y, 75 μm Z.

be attributed to the reduced projection depth over which the real calibration was sampled. A Gaussian distribution adds noise to the data points, and the distribution remains Gaussian following propagation through the algorithm, as shown in Figure 3.33.

A final Monte Carlo simulation is created to apply simulated calibration uncertainty to the real solved projection origin locations for Calibration 3, with the resulting distribution representing the final uncertainty in the module's calibrated length scale. The data for Calibration 3, given in Table 3.5, is used for the X,Y,Z values of PO_1 and PO_2.

The calibration uncertainty of 52 μm, 12 μm, 12 μm in the X,Y,Z is multiplied by a Gaussian distribution, and add the randomized noise to projection origins PO_1 and PO_2. The distance between PO_1 and PO_2 is then computed for each iteration. 1,000,000 total iterations were performed. The simulation reports a mean distance of approximately 0.05516 m, with a standard deviation of the distribution of approximately 50 μm.

Following the simulations, a final expanded uncertainty of $L_{Module} = 55.16 \pm 0.05$ mm ($k = 1$) is reported for the module's calibrated length scale as determined in Calibration 3, with a fractional uncertainty of approximately 9 parts in 10^4. This value is used for the measurements and evaluation discussed in Chapter 4.

CHAPTER 4: MEASUREMENT & EVALUATION

Prior to taking a photogrammetric measurement, a camera system must be calibrated in order to determine camera parameters such as distortion coefficients, principal point, focal length, etc. With a well characterized camera system, lens induced distortion in the image due can be corrected, making images suitable for use in photogrammetry measurements. In this chapter, the history and theory behind camera calibration for photogrammetry is described along with the calibrations performed on each camera. The photogrammetry software package, PhotoModeler®, experimental setup and procedure, and measurement trials are described. The measurement is analyzed and uncertainty evaluated by Monte Carlo simulation.

4.1 Camera Calibration

Camera calibration, or re-sectioning, describes the process by which the internal parameters of a lens and camera system can be estimated. In photogrammetry (and computer vision in general), it is critical to have a well described system, such that flaws in the image, such as distortion, can be corrected prior to utilization for measurement purposes. Distortion is a third order optical aberration which manifests as a change in magnification across the field of view of the image. Radial distortion occurs due to increased bending of light toward the edges of a lens. Tangential distortion occurs due to misalignment between the optical axes of the image plane and the lens. Dramatic examples of barrel and pincushion distortion are shown in Figure 4.1.

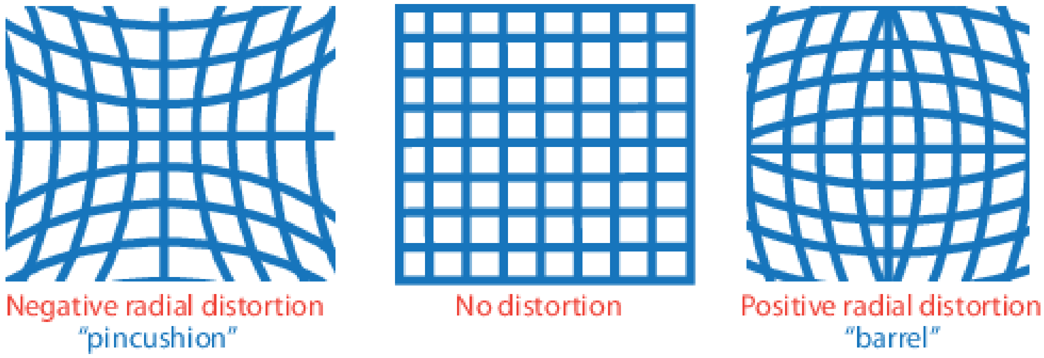


Figure 4.1: Illustration of negative and positive radial distortion in an image. *Image Credit: MathWorks* [88]

While photogrammetry can be completed without a calibration of the camera system, it will result in significant error. Lens selection in photogrammetry varies depending on the application; distant objects require the use of a telephoto lens, while photogrammetry completed in a lab setting can be accomplished with a simple wide angle lens. Yet, wide angle lenses often exhibit barrel distortion so correction is required.

Clarke and Fryer detail the history of the progression of camera calibration, from its earlier days and focus on correction of aerial photogrammetry, to modern computational methods [89]. Conrady's work in 1919 for the Royal Astronomical Society represents one of the first efforts to identify and correct for decentering error [90]. This work was expanded upon by Brown's introduction of the Brown-Conrady model in 1966, partly driven by a need to perform higher quality metrology on difficult to measure large objects, such as parabolic reflectors and antennas. Brown utilized a creative method - initially a series of oil-damped plumb-lines - to assist in describing and correcting radial and tangential (decentering) distortion [91]. The result of this work culminated in what essentially forms the baseline of the modern model to correct distortion [92, 93]. Computer-based computational methods dramatically increased system modeling capabilities. The introduction of the digital camera also meant that photogrammetrists were no longer bound to Reseau plates and comparators.

The combination of digital cameras and computer driven computational methods has further improved modeling capabilities and calibration processes for camera systems. Tsai published a detailed and well-modeled two step technique in 1987 to calibrate camera and lens systems for 3D machine vision metrology purposes. The first step computes the camera's external position relative to an object coordinate system, along with the camera focal length. The second step solves for image parameters [94]. This was finally expanded upon by Heikkila and Silven to a four-step method which solves for additional distortion coefficients and corrected distorted image coordinates [95]. Zhang's pinhole camera model in 1999 is a relatively elegant technique whereby a planar pattern viewed from multiple perspectives can be utilized to model and correct distortion [96]. This techniques serves as the basis for camera calibration techniques utilized by MATLAB®'s camera calibration toolbox, created by Jean-Yves Bouguet, as well as that of commercial systems like PhotoModeler® [88,97]. A simple description of Bouguet's photogrammetric calibration theory is provided in Section 4.1.1.

"Self-calibration" is another calibration technique advanced in aerial photogrammetry during the 1970s, where calibration is accomplished during measurement. This was expanded to the digital camera medium by Luong, Faugeras, and Maybank in 1992 [98,99]. The technique utilizes a single camera in a rigid environment to provide two constraints, one from the camera's internal parameters and another from image data. A collection of images taken with the same camera at various positions within a rigid scene allows for determination of internal and external parameters. This capability has been developed over nearly the last thirty years and is available within PhotoModeler®, but as this length scale system uses multiple cameras in to facilitate easier data collection in the lab environment, self-calibration is unfortunately not appropriate for this application. Self-calibration is advantageous in that cameras are essentially re-calibrated under measurement conditions every time.

4.1.1 Calibration Theory

A given camera system can be simplistically modeled as a pinhole camera, as illustrated in Figure 4.2, and its parameters represented in matrix form. Light rays from an object pass through a small aperture and an inverted image is projected onto an image plane.

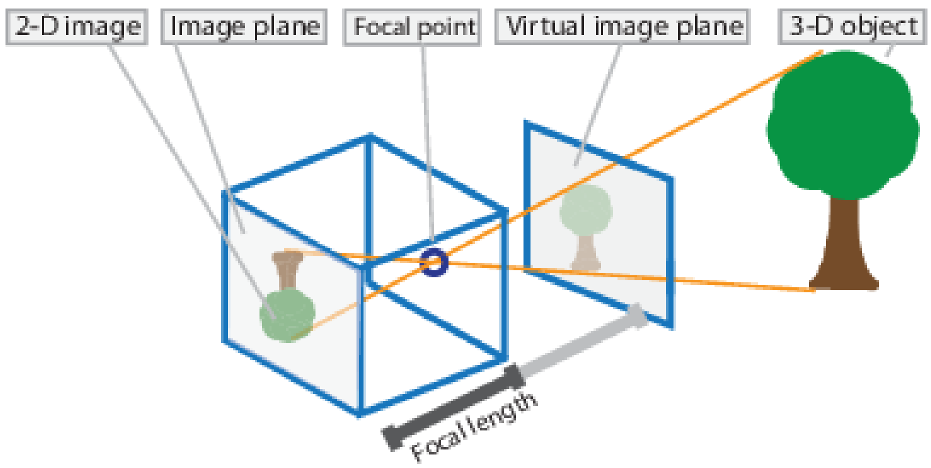


Figure 4.2: Model of a pinhole camera system. A combination of intrinsic and extrinsic parameters transform 3D world coordinates into 2D image space. *Image Credit: MathWorks* [88]

The pinhole camera model is expressed in the following matrix form:

$$w \begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix} P, \tag{4.1}$$

where

w is a scale factor,

x, y are image points,

X, Y, Z are world points from object space, and

P represents the camera matrix.

Parameters of the pinhole camera are presented by a 4×3 camera matrix, P, which maps the 3D world coordinates X, Y, Z from object space onto the image plane. The camera matrix P can be shown by the following matrix:

$$P = \begin{bmatrix} R \\ t \end{bmatrix} K, \tag{4.2}$$

where

R is the extrinsic rotations mapping world points to image space,

t is the extrinsic translations mapping world points to image space, and

K is the intrinsic parameters which map camera coordinates into pixel coordinates.

Extrinsic parameters represent a coordinate transformation from 3D world space into 3D camera coordinate space. Intrinsic parameters then map from 3D camera coordinate space into 2D image coordinates. Intrinsic parameters are defined by a 3×3 matrix:

$$K = \begin{bmatrix} f_x & 0 & 0 \\ s & f_y & 0 \\ c_x & c_y & 1 \end{bmatrix}, \tag{4.3}$$

where

f_x, f_y is the focal length in pixels,

c_x, c_y is the optical center, or principal point, in pixels, and

s is a pixel skew coefficient.

The skew coefficient, s, takes a non-zero value if the image axes are not orthogonal, where $s = f_x tan\ (\alpha)$. The parameter α represents the angular skew of the pixel. The focal length in pixels, $f_{x,y}$, equals $F/p_{x,y}$ where F is the focal length of the lens in millimeters and p is pixel size in world units.

The above described pinhole camera model is limited in that it does not include corrections for a lens and thus has no way to describe corrections to distorted image points. Image point distortion in the x, y plane can described by two sets of equations. Radial distortion in the x and y are solved:

$$x_{distorted} = x(1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6), \tag{4.4}$$

$$y_{distorted} = y(1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6), \tag{4.5}$$

where

k_1, k_2, k_3 are radial distortion coefficients of a lens.

For a moderate angles, two distortion coefficients are usually adequate. Wide angle lenses may require a third distortion coefficient. Tangential distortion, which occurs when the image plane is not aligned with the optical axis of the lens, also results in distortion to x, y image points. Tangential distortion is described:

$$x_{distorted} = x + (2 * p_1 * x * y + p_2 * (r^2 + 2 * x^2), \tag{4.6}$$

$$y_{distorted} = y + (p_1 * (r^2 + 2 * y^2) + 2 * p_2 * x * y), \tag{4.7}$$

where

p_1, p_2, p_3 are tangential distortion coefficients of a lens.

Again, two coefficients are typically adequate. Wide angle lenses may require a third coefficient to correctly describe the tangential distortion. Camera calibrations in this system use two coefficients.

4.1.2 Calibration Procedure & Data

PhotoModeler® features a built-in calibration tool to assist in the process of camera calibration. Similar to the planar method suggested by Tsai (1987), PhotoModeler® provides a 36" square 12 by 12 dot matrix grid template, shown in Figure 4.3. It contains four coded targets to assist in orientation and automatic detection and referencing of dot targets [94]. The pattern was printed out using a large format high resolution poster printer, and affixed it to a rigid foam board with adhesive. The foam board was placed on the ground and weighted in the corners to ensure it did not move during the calibration process.

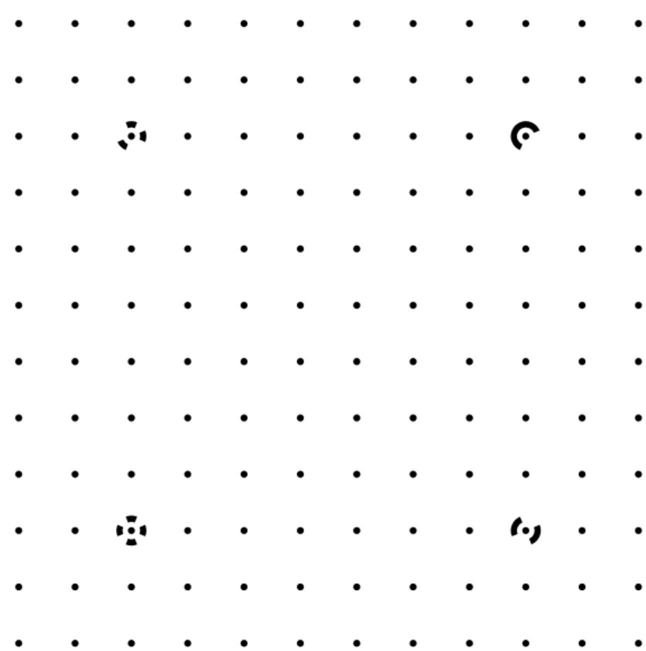


Figure 4.3: 36" x 36" dot matrix calibration grid, utilized for automated calibration of camera systems within PhotoModeler®. Four uniquely identified coded targets orient the board to the algorithm, and allows for automatic dot referencing.

The PhotoModeler® user manual provides suggestions to improve the overall quality of the calibration [100]. Some of those suggestions are described here. The process is generally as follows:

- Select a camera and appropriate focal length lens.
- Decide on the size of the calibration grid. Print the grid and affix to a rigid surface.
- Take between 6 to 12 photographs of the grid, ensuring that the camera is rolled about its optical axis for some of the images.
- Load the photos into the PhotoModeler® calibration wizard.
- Execute the calibration and review the quality of the calibration. If adequate, save the calibration to the PhotoModeler® camera library. It can be recalled for future projects.

This project utilizes four consumer grade Canon T3i Digital SLR (Single Lens Reflex) cameras, each with a Canon EF-S 24mm $f/2.8$ lens. The wide angle lenses were selected to provide a wide view for in-lab usage in order to minimize the number of photographs necessary for processing. Cameras are labeled A, B, C & D and a unique calibration was performed for each camera/lens combination. Ideally, a camera is calibrated at a focus which corresponds to the distance at which it will be used for photogrammetry. While these photogrammetry measurements were not strictly performed at that focal depth, efforts were made to match the calibration distance with the hyper-focal distance for each lens in order to maximize the sharpness of photogrammetry targets in the scene for the $f/11$ lens aperture. The estimated distance from the camera imaging plane to the calibration target was approximately 7 feet. Each of the focus rings on the four cameras were then temporarily affixed with a few drops of low-temperature hot glue.

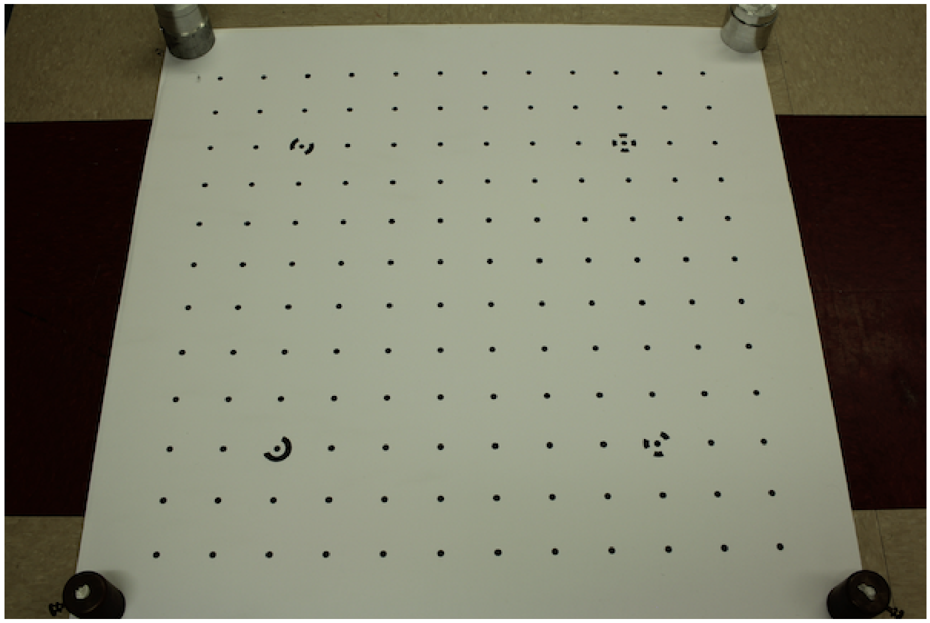


Figure 4.4: Sample PhotoModeler® calibration board photograph taken from calibration set. The calibration pattern is affixed to rigid foam board and weights on each of the four corners ensures the board remains relatively stationary during the calibration process.

The cameras were mounted to a tripod and 12 pictures were taken with each camera utilizing a delayed shutter function to minimize hand shake. The calibration board was photographed from each of its four sides. At each position, three separate pictures were taken, one at each of $-90°, 0°, 90°$ roll orientations about the optical axis. It is important that the calibration board cover as much of the image as possible so that distortion can be well sampled across the field of view. This does not mean that the calibration board must be 'zoomed in' to maximize board coverage in a single photo, but rather that at one of the three images taken at each position, one image should be left justified, one center justified, and one image right justified in the viewfinder of the camera, so to speak. A coverage factor percentage is provided in the calibration data output.

Table 4.1: Individual camera calibration constants and quality metrics as reported by the PhotoModeler® built in camera calibration tool.

Camera	**A**	**B**	**C**	**D**
Focal Length (mm)	25.1736	25.2316	25.2202	25.1666
Principal Point				
X (mm)	11.5471	11.4131	11.3574	11.3515
Y (mm)	7.4305	7.5091	7.6213	7.6149
Format Size				
Width (mm)	22.6728	22.6702	22.6732	22.6668
Height (mm)	15.1130	15.1130	15.1130	15.1130
Lens Distortion				
K1	2.017E-04	1.989E-04	1.994E-04	1.988E-04
K2	-2.996E-07	-2.953E-07	-2.808E-07	-3.018E-07
P1	5.315E-07	-5.804E-06	8.227E-06	9.783E-06
P2	-6.889E-06	-2.328E-05	-1.454E-05	-2.418E-05
Calibration Quality				
Residual RMS (px)	0.0389	0.0436	0.0470	0.0599
Max Residual (px)	0.1880	0.2485	0.2746	0.3756
Photo Coverage	93%	91%	91%	94%

Table 4.1 shows the final calibration data for the camera set, including principal point, format size, lens distortion, and various calibration quality metrics. Note that a third radial and tangential distortion coefficient were not necessary.

4.2 PhotoModeler®

In order to obtain 3D coordinates of the measurement artifact and projected grid points, a photogrammetry software package is required. Various commercial packages are available, such as PhotoModeler Technologies PhotoModeler®, Geodetic Systems V-STARS®, and AgiSoft Metashape®. Of these, a pre-existing license for PhotoModeler Technologies (formerly Eos Systems) Ver. 6 was available for lab use and was thus selected for this project [33, 101, 102].

4.2.1 Software Overview

PhotoModeler® is a commercial software package, which utilizes a proprietary bundle adjustment algorithm to deliver 3D target coordinates from a series of 2D photographs. While the software has the capabilities to perform other functions, such as textured reconstruction of models in 3D, only coordinate capture is used in this research. As a commercial software package, the user interface is relatively intuitive. An outline of the PhotoModeler® workflow is shown in Figure 4.5.

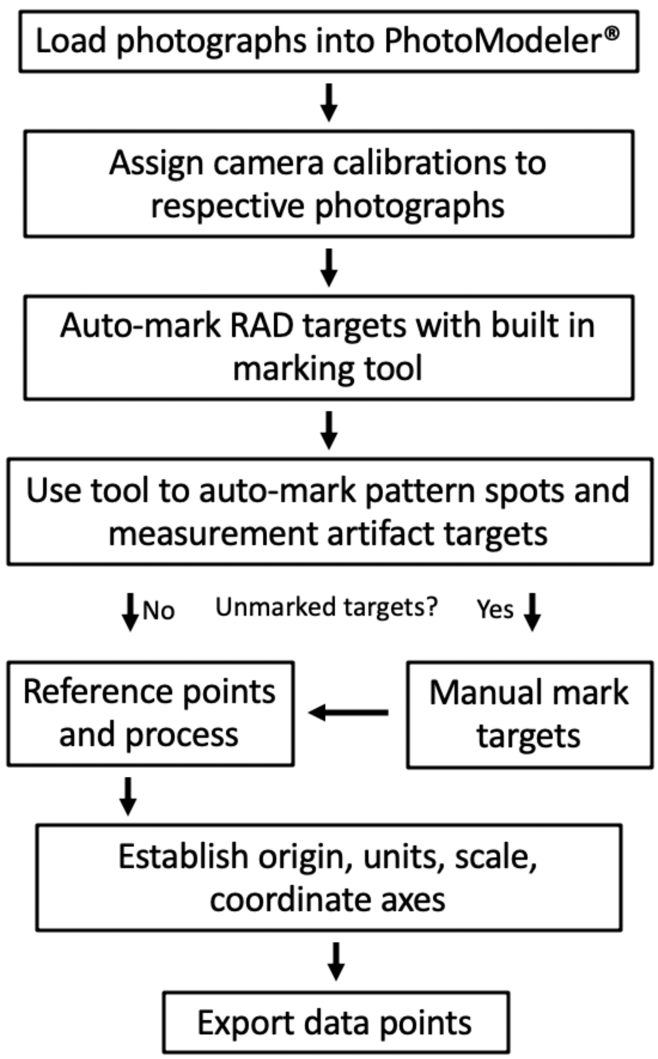


Figure 4.5: PhotoModeler® process flowchart

4.2.2 Data Processing & Point Marking

While the software is relatively straightforward to use, a few topics deserve further comment. PhotoModeler® has a handful of tools built in to assist with point marking and point referencing. Given the quantity of targets necessary for a high quality photogrammetry measurement, not to mention the additional marking an optically projected grid, these tools are absolutely necessary to reduce the processing burden for the user. Without them, a process that can take a handful of minutes would rapidly spiral into hours or days. Additional discussion on the processing burden can be found in Chapter 5.

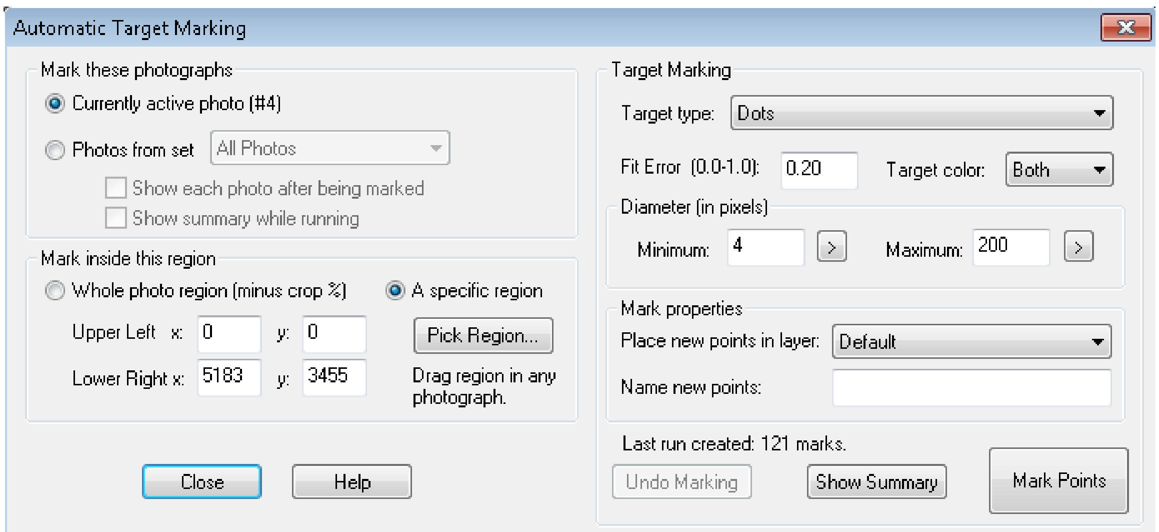


Figure 4.6: PhotoModeler® built in target marking tool. This tool allows for automatic marking of all RAD targets in images. It also allows for group selection of black or white spherical targets.

Shown in Figure 4.6 is a tool to automatically mark different style targets. Target type allows for RAD coded targets, or generic circular dots. The target color can be set to either black, white, or both. The options also allow for searching of the entire set of photos or a single photo. Within that selection, one can search the whole photo or a single specified rectangular region as drawn by a mouse.

After loading the photo set and selecting the correct camera calibration profile for each photograph, the automatic target marking tool is used to auto mark all RAD targets within the entire set of photos. Periodically, one of these targets is missed, and it must be manually marked with the sub-pixel target marking tool. Because the tool looks for a high contrast circular target, its effectively centroids the target to sub-pixel accuracy.

RAD targets and the black targets on the measurement artifact usually auto mark well, and are immediately detected with high precision. Optical pattern targets, however, can present challenges with for automatic detection and marking. The PhotoModeler® automatic marking tool will mark optical pattern spots; the target type must be selected as 'Dots', and the target color must be set to 'White'. While the projection source is a 658 nm diode laser, a saturated spot appears relatively white relative to its background to the detection algorithm, and is typically reliably detected. For each image where the optical pattern is marked, there are typically 1-3 projection spots that are not detected, and these must be manually marked. The small size of the active layer of a typical edge-emitting laser diode leads to large angular divergence along one axis [103]. This characteristic, along with flaws in the grating and optical system can result in an irradiance pattern which is elliptical, but non-symmetric in its intensity. Speckle presents additional challenges. While the impact of speckle on target recognition is more thoroughly discussed in Chapter 5.1.1, it can be mentioned here is that it simply disrupts the circular appearance of each spot, negatively impacting the centroiding ability of the software. Occasionally these spots cannot be centroided with the sub-pixel marking tool and a manual spot must be selected. These missed spots occur more frequently with camera stations that image the rigid foam board at non-orthogonal angles.

Figure 4.7: Two spots from the optically projected pattern are shown. Quality of the left spot is high, and it is easily centroided. The right spot presents an uneven intensity profile with insufficient high contrast pixels to allow for automatic detection and centroiding.

Figure 4.7 shows two optically projected spots as an example. The left spot presents an ideal intensity profile with an adequate number of pixels to allow for high precision centroiding. The spot on the right was not automatically detected nor could it be marked with the sub-pixel marking tool. The occurrence of poor quality projected spots is infrequent over the 121 points in the pattern, so their overall error contribution also remains low.

Referencing is a critical step in processing data points. Referencing is the process of telling the software that a marked point in different images represents the same point in space. This allows for the bundle adjustment to identify it as a unique point. The software automatically references RAD targets. Because the numerically coded ring around each coded target serves as a unique identifier, the software is able to automatically reference those points from image to image. Optically projected pattern targets do not have unique identifiers, however, and thus must be referenced. Figure 4.8 shows a sample image containing marked and referenced targets, consisting of an array of coded RAD targets on the wall as well as projected pattern spots.

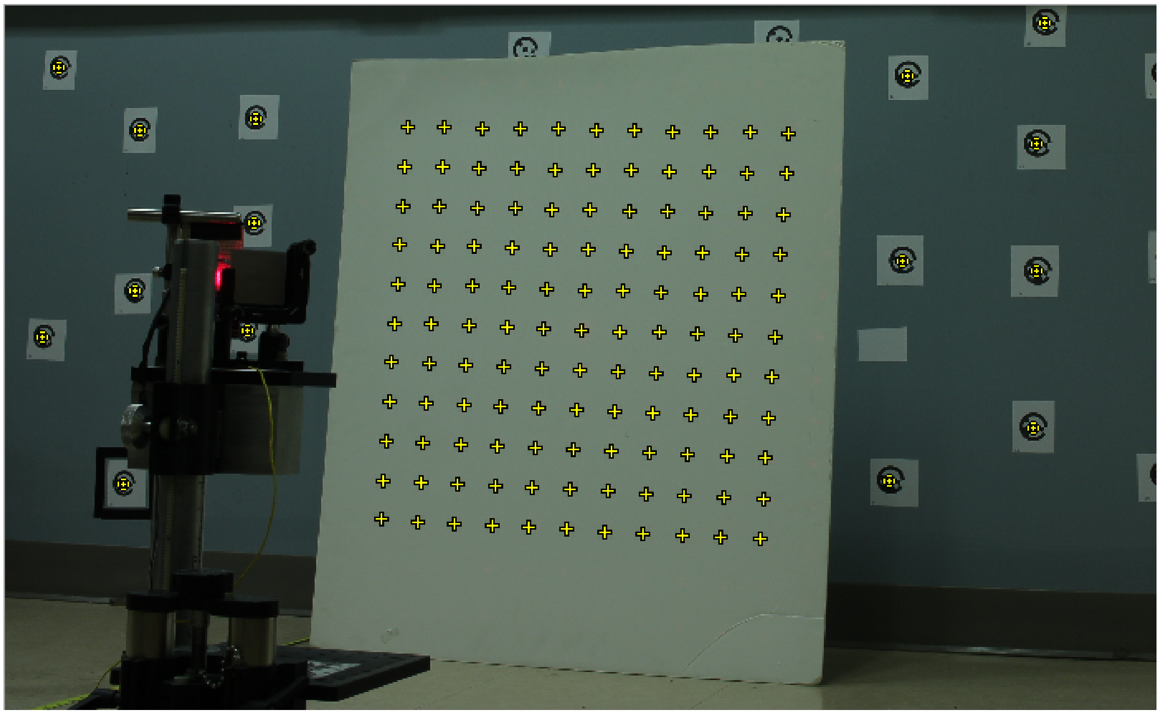


Figure 4.8: A mixture of marked and referenced RAD and optically projected pattern targets.

Coded targets are overlaid with a cross hair, signifying their recognition and auto-referenced status as RAD targets. Projected pattern spots are indicated by a '+' symbol. Manual referencing each image would be very time consuming, especially as each measurement has 121 projected pattern spots, two measurement artifact spots, and a series of RAD coded targets. To reduce the processing burden, PhotoModeler® provides an automatic referencing tool for non-RAD targets, shown in Figure 4.9. When enough points have been referenced, which is typically accomplished by detection of well distributed RAD targets, a basic bundle adjustment can be completed to provide a geometric baseline. At this point, the remainder of the non-coded points can be automatically referenced by the tool. The distance between projected pattern spots is centimeter scale, so a search distance of 1-2 mm is adequate for single-pass automated referencing.

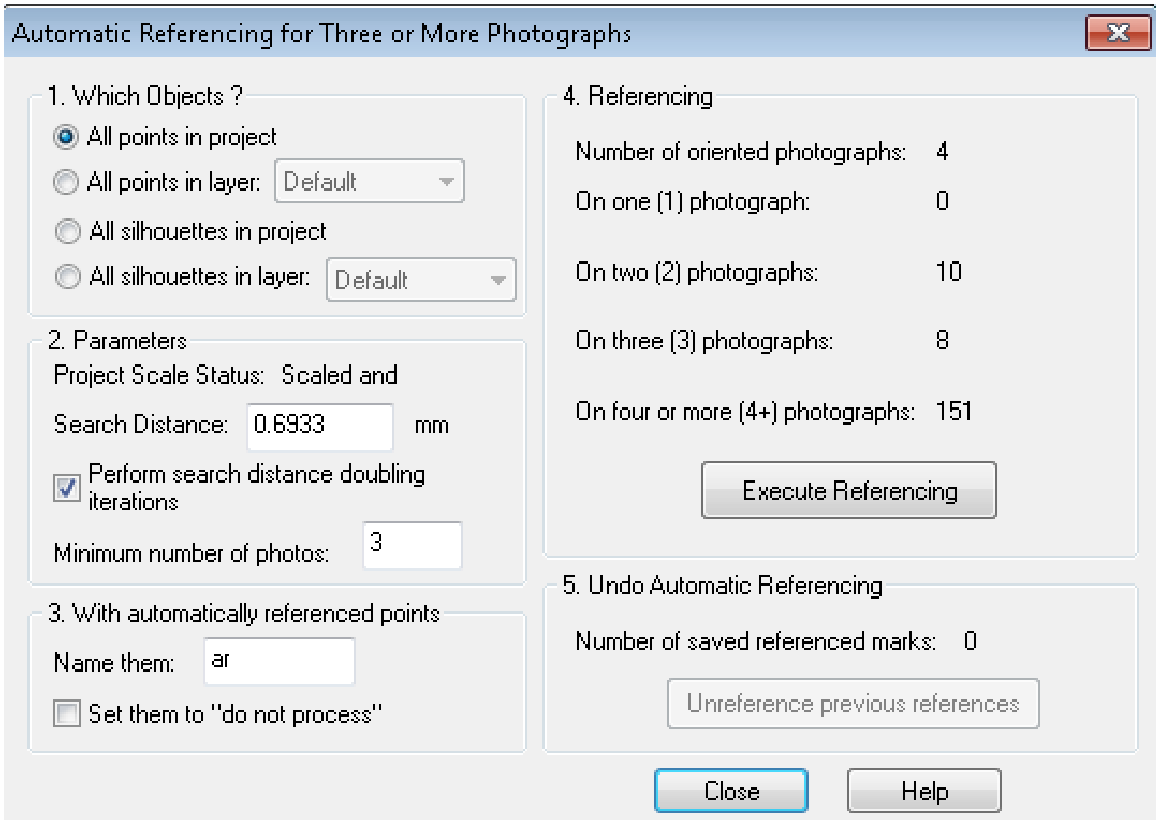


Figure 4.9: Once a basic bundle adjustment has been performed using coded RAD targets, the PhotoModeler® automatic referencing tool utilizes an iterative distance search to find paired points, and automatically reference them.

Naming and ordering for data output is another critical issue. Due to the coded numerical nature of RAD targets, PhotoModeler® assigns the numerical coded value of the target as its name or ID. No such classification exists for projected laser spots. When 3D coordinate data files are output from PhotoModeler® in *.txt format, the ordering given to coordinates of projected spots is randomized due to a multi-threaded CPU process used during data processing and the bundle adjustment. The algorithm must match a given projected target with its expected calibrated pointing direction. Positive identification of each projected spot in the pattern is required.

To resolve this problem, code was written to sort the data. Knowing the pattern dimensions and thus how many dots are present in the pattern, points are sorted by highest Z value first. Because the pattern is not overly rotated with respect to the established coordinate system and because the spacing between points is on the order of centimeters, a given horizontal row can be relatively grouped within a certain range of Z values. Once sorted into 11 horizontal rows, the algorithm then sorts by the Y value. Pattern points are sorted from top to bottom, then right to left. The upper-right most projected spot is number 1, and the bottom-left most projected spot is number 121. The sorting algorithm is found in Appendix A.7.

The data set must also be translated, scaled, and rotated by setting the origin, specifying the units, scaling the coordinates with a point to point measurement within the scene, and determining the coordinate system orientation. Section 4.3 provides details on coordinate system orientation.

A final comment is made on the discussion of inclusion of the optically projected points within the bundle adjustment. PhotoModeler® allows the user to exclude specific points from a bundle adjustment. This can prove useful where the 3D coordinate data of a specific point is desired, but the user does not want the noise associated with that point to impact the overall bundle adjustment. In this setup, 52 RAD targets, 2 targets on the measurement artifact, and 121 projected targets are marked per image. The optically projected points make up more than two thirds of the marked targets. Processing was tested with and without optical points included in the bundle adjustment, and a lower overall residual RMS for the bundle adjustment was found when taking all points into account as opposed to neglecting the optically projected spots and relying on coded targets alone. Processing with over one hundred additional targets in the scene, albeit ones with more centroiding noise, nevertheless seemed to have a positive impact on the overall stability of the bundle adjustment.

4.3 Experiment Design & Setup

To prepare for experimental validation of the measurement technique, the lab was set up as follows: 52 coded black ink on white background RAD targets were printed on adhesive backed paper, and affixed to the wall in a well distributed pattern. The

RAD coded targets were generated with PhotoModeler®'s built in target generation tool. Using the target-distance estimation tool provided and assuming an average 3 m camera to target distance, PhotoModeler® recommended an inner target diameter of 9.53 mm, with a coded ring diameter of 45.27 mm. Approximately 35% of the target diameter value was used as white space for a border. The targets are shown in Figure 4.10. Three of the coded targets, outlined in black tape, serve to establish of a right-hand coordinate system, which is indicated in the top left of the figure. The bottom left outlined target is selected as the origin. The origin and the outlined target in the top left hand corner define the dominant axis. The upper right hand outlined target establishes the (-Y) axis orthogonal to the dominant axis. The calibrated measurement test artifact is visible in the figure, and has a target center to center length of 1677.66 mm $\pm$ 10 μm.

Figure 4.10: Typical photogrammetry measurement scene. Coded black on white RAD targets allow for automatic identification and referencing by the software. Three black outlined coded targets establish a right handed coordinate system. The carbon fiber measurement test artifact is visible.

The module, described in Section 2.6, is placed on a position-locked elevated mount and fastened to a rigid optical board which is shimmed to sit stable on the floor. A K-type thermocouple is attached to the module's optical board to monitor temperature. The module projects its pattern onto a vertically supported rigid foam board. Four Canon T3i consumer grade DSLR cameras with Canon EF-S 24mm $f/2.8$ lenses are used to take pictures of the scene. The cameras use an 18 megapixel sensor to produce images with a resolution of 5184 x 3456 pixels [104]. Section 4.1.2 details the calibration of the cameras. All cameras are attached to tripods and are placed in a 'four-corners' formation, as shown in Figure 4.11. Lab space constraints restricted camera separation to no more than a 70° convergence angle, which is adequate for this measurement. Cameras are powered by external power supplies.

Figure 4.11: The camera network was constructed from four Canon T3i cameras with EF-S 24mm $f/2.8$ lenses, shown boxed in red. The average convergence angle between two cameras and a given target was 70°.

Each camera is linked to a central hub with a wired remote actuation cable which synchronizes the shutter actuation of all cameras in the system to within approximately 10 ms. Each camera is also linked via USB cable to a laptop computer. An open-source camera control software package, digiCamControl Ver. 2.1.1.0, allows for remote control of shutter actuation, camera setting changes (ISO, shutter speed, aperture etc.), and automatic transfer of recorded images for data processing [105]. An $f/11$ aperture was used to provide adequate depth of field while maintaining image sharpness, with a shutter duration of $1/4^{th}$ second.

4.4 Measurement Procedure

The measurement procedure is as follows:

- The projection module's diode laser is powered on and allowed to stabilize for 30 minutes. The dual grid pattern is projected onto a foam board screen in the scene.

- The mirror side grid pattern from the module is blocked (Figure 4.12). The computer program digiCamControl remotely triggers the linked camera system and captures images of the pellicle side optical pattern projection in the scene, which are then transferred to the PC via USB. The pellicle side grid pattern from the module is then blocked and the image capture process repeated for the mirror grid. The projected grid data from the two sides of the module are acquired separately to simplify data processing. This process is repeated for additional measurement trials.

- Image sets are processed in PhotoModeler®. An automated tool is used to detect and mark coded RAD targets. The measurement artifact is manually marked and the projected grid spots are automatically marked, with missed spots marked manually. A global coordinate system is established using the three outlined coded targets discussed above, then is processed, with the bundle adjustment delivering the $(X, Y, Z)_{Grid}$ coordinates of each optically projected spot.

- Grid spot coordinates are sorted by position (top right to bottom left) then imported into MATLAB®. A guessed value of the grid projection origin is input into the algorithm as a starting point. The algorithm is run and the location and pose $(X, Y, Z, \alpha, \beta, \gamma)_{Origin}$ of the projection origin is delivered.

- This process is repeated for the second grid projection location. The Euclidean distance between the two projection locations, L_{Photo}, is solved. L_{Photo} is the unscaled length between projection origins as determined via photogrammetry. $L_{Calibration}$ is the real calibrated length between the projection origins as determined during the calibration process outlined in Chapter 3. The ratio of $(L_{Calibration}/L_{Photo})$ serves as the scale factor for all photogrammetry delivered coordinates.

- Photogrammetry coordinates for the measurement artifact are scaled and the artifact length solved.

4.5 Artifact Measurement & Results

In order to evaluate the performance of the module, a user would ideally image the optically projected pattern, solve for the coordinates of the projected points, and then utilize the algorithm to measure the projection origins. The solved length scale L_{Photo} could then be compared against the calibrated value L_{Cal}, yielding a scale factor for all targets in the scene. Ideally, all of this functionality would be built into the photogrammetry software itself.

Unfortunately, PhotoModeler® was not built with this new method in mind. The software requires the user to select at least two points in the scene for which the dimension is known, and scales the coordinates accordingly. This is a required step

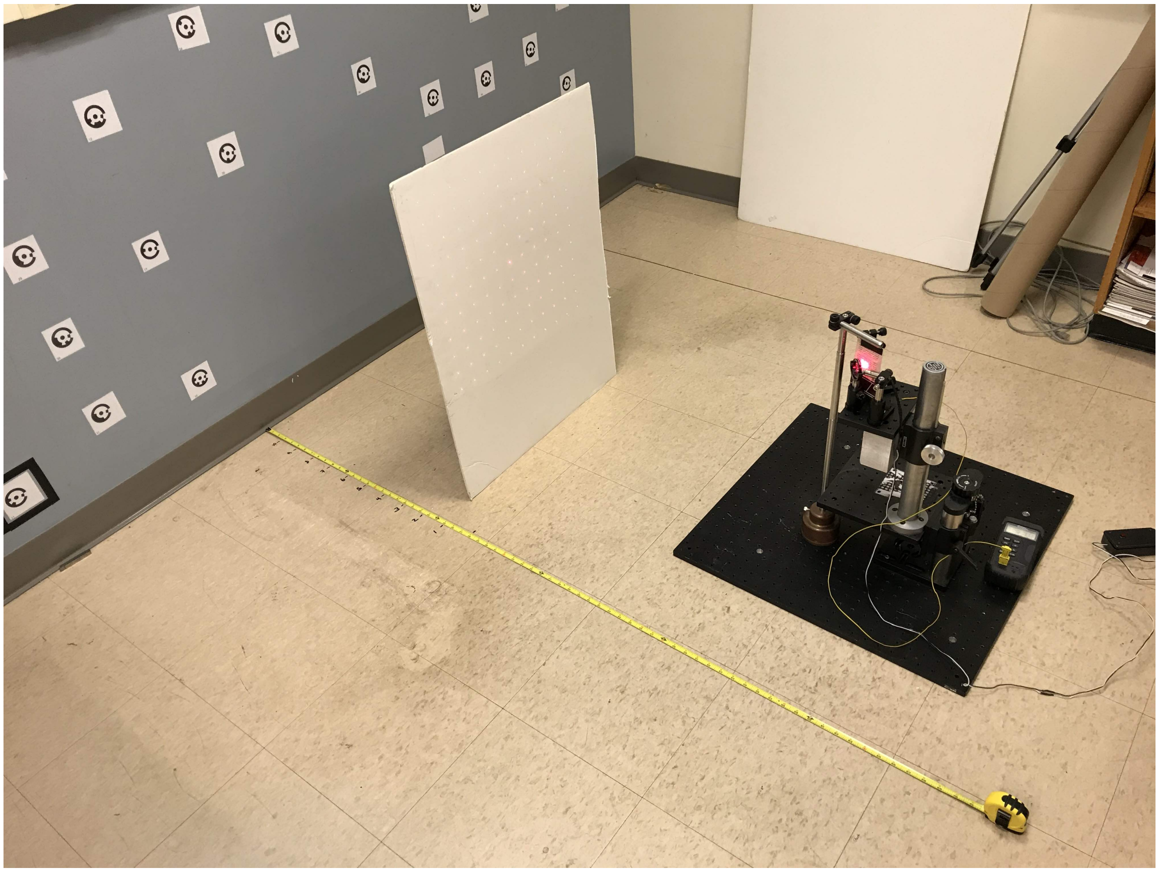


Figure 4.12: Temporarily blocking an optically projected grid allows for each side to be imaged and processed individually. This dramatically reduces the processing burden and simplifies point data acquisition.

and cannot be skipped. For the purpose of testing the module, the measurement could be scaled by any of the points available in the scene. Solving the optical patterns would then yield a scaled L_{Photo}, which could be compared to the calibrated value.

Instead of scaling with an arbitrary set of points in the scene, however, a choice was made to utilize the artifact which was created for the camera calibration. Its design and calibration is described in detail in Chapter 3. There are a number of advantages to using this artifact which deserve mention and explanation. Purpose built for photogrammetry, it is reliably detected by PhotoModeler® with detection performance on par or better than the coded RAD targets shown in the scene. The artifact has been calibrated to a known length of 1677.66 mm $\pm$ 10 μm. Importantly, if any such error existed in the CMM measurement of the calibration artifact, that same error will have influenced the module's calibrated length scale value. Therefore, by "reading back" the test artifact with the module, the measurement results are not influenced by potential calibration scale error.

By scaling the pattern coordinates with the calibrated length scale, the coordinates of the projected patterns should be scaled to correct real world coordinates. This means that after processing through the algorithm, the separation distance between projection origins will also be scaled to real coordinates. Thus, the length scale α, which is a ratio of $L_{Photo}/L_{Calibration}$ should nominally have a value of 1, as the photogrammetrically solved separation should now match that of the calibration. This is fundamentally no different than scaling with arbitrary points, solving the length scale, then scaling a known artifact in the scene and measuring its length. Adopting the former method merely skips a step during processing.

Two measurement experiments were completed, with the projection screen distance being changed between each experiment. Each experiment consisted of 20 individual measurement trials conducted as per the procedure described in Section 4.4. For each measurement, a grid was blocked, and a single projection grid imaged 20 times over a 7.5 minute period. The first pattern would then be blocked, and the second grid imaged 20 times over an additional 7.5 minute period. A full measurement experiment lasted approximately 15 minutes. After the first measurement, the screen was moved away from the module, and the experiment completed again.

Following each measurement experiment, PhotoModeler® was used to solve for the pattern coordinates for each of the 20 measured trials, and the algorithm used to find the projection origins of both patterns. The solved origins of the 20 trials per side were averaged, with the Euclidean distance between the mean positions of the origins giving the final calculated length scale L_{Photo}. This process was then repeated for the second measurement experiment.

The first measurement experiment was performed with the movable screen positioned approximately 990 mm from the projection module at a stabilized module temperature of $T = 22.4°$ C. The separation distance over 20 averaged measurement trials per grid was 54.99 mm, yielding a scale factor of $\alpha = 99.69\%$. The mean and standard deviation for each projection origin are shown in Table 4.2.

Table 4.2: Measurement 1 projection origin mean and standard deviation. Screen depth 990 mm.

	X (mm)	**Y (mm)**	**Z (mm)**	$\boldsymbol{\alpha}$ (rad)	$\boldsymbol{\beta}$ (rad)	$\boldsymbol{\gamma}$ (rad)
$\boldsymbol{PO_1}$						
Mean	-1421.17	-920.96	275.57	0.02946	-0.00044	0.00257
SD(σ)	0.02	0.11	0.10	0.00001	0.00009	0.00010
$\boldsymbol{PO_2}$						
Mean	-1461.35	-959.07	275.60	0.02228	0.00760	0.01694
SD(σ)	0.03	0.07	0.07	0.00001	0.00006	0.00006

Separation Distance	54.99 mm
Scale Factor α	99.69%

The second measurement experiment was performed with the movable screen positioned approximately 1145 mm from the projection module at a stabilized module temperature of $T = 22.4°$ C. The separation distance over 20 averaged measurement trials per grid was 55.57 mm, yielding a scale factor of $\alpha = 100.74\%$. The mean and standard deviation for each projection origin are shown in Table 4.3.

Table 4.3: Measurement 2 projection origin mean and standard deviation. Screen depth 1145 mm.

	X (mm)	**Y (mm)**	**Z (mm)**	$\boldsymbol{\alpha}$ (rad)	$\boldsymbol{\beta}$ (rad)	$\boldsymbol{\gamma}$ (rad)
$\boldsymbol{PO_1}$						
Mean	-1421.68	-920.79	275.81	0.02949	-0.00023	0.00243
SD(σ)	0.02	0.15	0.11	0.00002	0.00009	0.00013
$\boldsymbol{PO_2}$						
Mean	-1461.45	-959.28	275.32	0.02228	0.00738	0.01712
SD(σ)	0.06	0.09	0.16	0.00002	0.00013	0.00007

Separation Distance	55.57 mm
Scale Factor α	100.74%

4.6 Module Performance: Simulation & Uncertainty in Measurement Value

Assessing the uncertainty in a measurement performed by the optical projection system is complicated. Diode temperature changes and instability can lead to changes in pointing directions of the pattern beams. Temperature fluctuations in the local module environment could result in dimensional changes in the module and thus a change in the calibrated separation value between grid projection origins. These fluctuations manifest as changes in the projected pattern spot locations, which ultimately is the only good quality metric that can be directly measured.

Owing to the short duration of a measurement, which can be completed in mere minutes, temperature changes in a relatively stable lab environment will be minimal. As such, it is reasonable to assume that the dominant source of uncertainty will be from the scatter in spot location the arises due to objective speckle. Over the course of the measurements, an average scatter was found in the locations of projected grid spots of approximately 110 μm, 73 μm and 73 μm in the X, Y, and Z components.

A Monte Carlo simulation was used to evaluate the uncertainty in the solved projection origins. Normally distributed noise was added to simulated pattern spots, and the algorithm repeatedly solved for projection locations. The simulation was constructed with parameters and dimensions that matched the fabricated prototype. A projection separation length was assumed of approximately 0.05516 m, with a 45° rearward offset between grid projection locations. The full angle of each 11 x 11 grid was 29°. 5000 iterations were performed, resulting in a standard deviation of approximately 186 μm for the separation length, or a fractional uncertainty of 3.4 parts in 10^3. The standard deviation for a single grid's solved location was approximately 30 μm, 110 μm and 110 μm in the X, Y, and Z. Assuming spot location uncertainty as the dominant factor, a simulated expanded uncertainty of $U_C = 0.4$ mm (k=2) is reported for the solved length between projection locations.

4.7 Analysis & Conclusions

In the two measurement trials conducted, separation distances were estimated of approximately 54.99 mm and 55.57 mm. Given the calibrated length scale separation of 55.16 mm and an uncertainty in the solved length scale of 0.4 mm (k=2), both of these values generally fall within 2σ of the expected value, with the second measurement just outside of the bounds by approximately 10 μm. Nevertheless, the first measurement experiment shows a bias toward under-reporting the correct length of the module. The second experiment appears biased toward over-reporting.

The two measurement experiments were conducted sequentially within a 30 minute time period. The reported temperature on the optical board for both experiments was 22.4° C. The only factor that really changed between experiments was the position of the movable screen upon which the pattern was projected. The first experiment was conducted with the screen positioned 990 mm from the module and the second conducted with the screen 1145 mm from the module.

A possible biasing factor is the laser diode source, which is not collimated. The laser comes to a focus approximately 2.5 m into the field. From 990 mm to 1145 mm, the laser pattern spots are contracting, as shown in Figure 4.13.

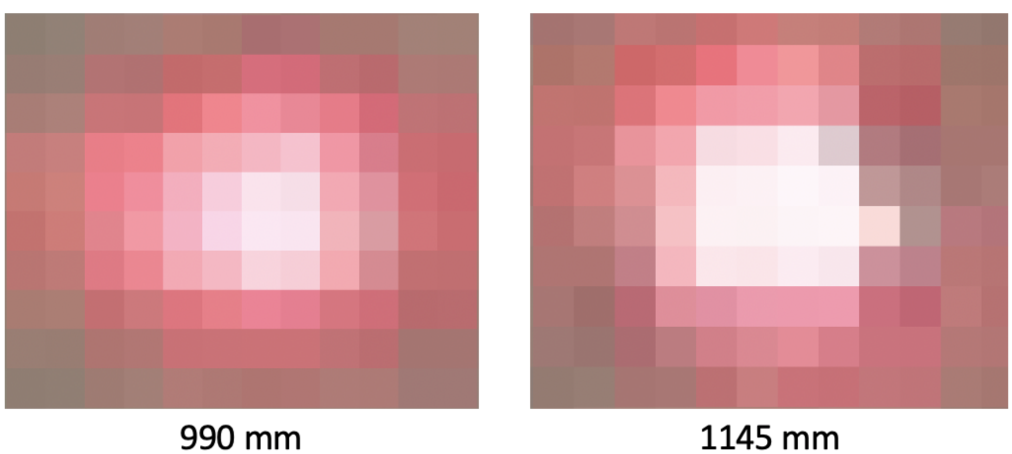


Figure 4.13: Due to an uncollimated source, pattern intensity changes as a function of screen distance from the module.

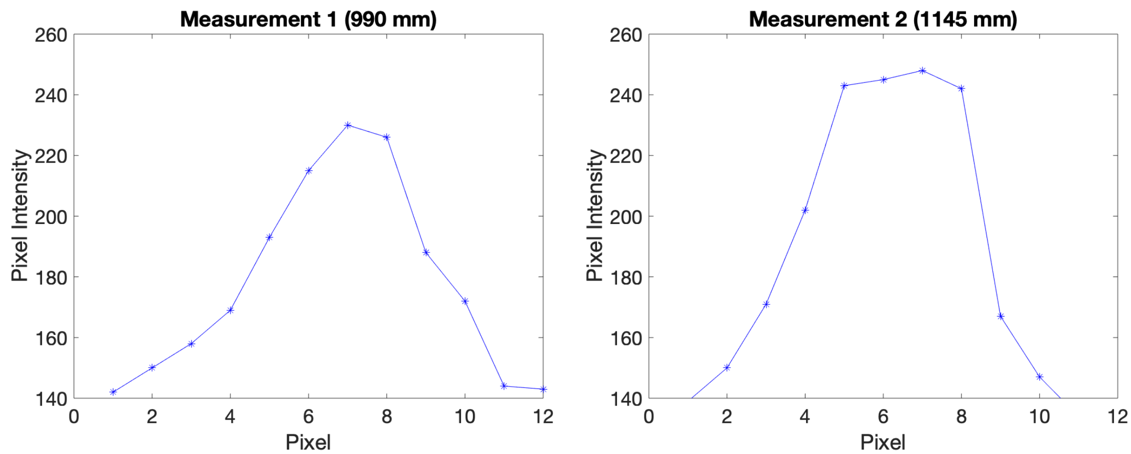


Figure 4.14: Changes in the intensity profile across the 6^{th} row of a 12×12 pixel image of the pellicle side central beam. Screen depths of 990 mm and 1145 mm for measurements 1 and 2, respectively.

This resulting apparent change in the intensity is duplicated across the entire dataset for every trial in both experiments. To explore this more, a 12×12 pixel crop centered about the central beam was examined for the projected pellicle pattern from the first trial of each of the two measurement experiments. The image was converted from RGB to gray scale and the sixth row about each image plotted as a function of intensity per pixel, as shown in Figure 4.14. As the pattern beams come to a focus, the intensity (power per unit area) of the beam increases. As many beams in the pattern exhibit an asymmetrical intensity profile, the centroid of the spot can laterally shift as the intensity varies as a function of screen depth. This can be observed in Figure 4.14.

While PhotoModeler® does not disclose its centroiding method, based upon user experience of marking thousands of pattern spots, there is a certain threshold intensity below which spots will not be marked. If the intensity threshold for centroiding by the software were to occur at value of 220, for example, there would be a shift of almost a pixel in the centroided spot location. Put into perspective, a 1 pixel shift at this working distance is equivalent to approximately 0.5 mm. If this phenomenon occurs over the entire pattern, error could manifest in the algorithm's solved projection origin. This change in centroid could appear to make the pattern shrink or expand, which could shift the projection location forward or backward by a small amount. While this is occurring in both patterns, because the optical path lengths of the grids are not the same, and because the grid pattern beams are not parallel, this could induce cosine error. Given the relatively small length scale employed by the module, these shifts could result in a notable change in the solved length of L_{Photo}.

There is not yet sufficient evidence to explicitly state whether this phenomenon is responsible for the bias. Beam stability is examined in Section 5.5, but no conclusive evidence was found to support the claim that small temperature changes on the module were influencing the pattern behavior more so than typical scatter in the pattern due to speckle. Regardless, a collimated source and pattern should yield more stable measurement results. It is recommended that this phenomenon's potential bias be studied more in future work.

There are a number of factors which directly impact the uncertainty of the projection module measurements. Four are considered: noise associated with a beam spot centroid, separation distance between grid projection sources, number of spots sampled, and the full angle of the projected pattern. The change in uncertainty resulting from changes to module parameters is explored in Section 5.1. Building upon that knowledge, an improved module is proposed in Section 5.2. Additionally, process improvements are proposed in Section 5.3 which could help drive down uncertainty and dramatically decrease post-processing time.

CHAPTER 5: LIMITATIONS, ERROR, IMPROVED MODULE DESIGN

The calibration and measurement chapters of this dissertation explored the uncertainty in the module's calibration, evaluated the module in a series of measurements, and modeled the system to evaluate the uncertainty of those measurements. This chapter explores the impact of key module design parameters, such as beam number and pattern full angle, on the module's overall performance as an optically projected length scale. Some practical limitations of these parameters are discussed and suggestions are made where further reductions in uncertainty can be achieved. Error contributions of the module's components and design are examined and a simple, more robust module design is proposed which offers an estimated order of magnitude reduction in fractional uncertainty over the current design while maintaining handheld portability and ease of manufacturing.

5.1 Limitations of Module Performance

There will always be uncertainty in measurement and photogrammetry is no exception. During the course of a photogrammetric measurement, the end user ideally will have access to a length scale artifact that has low fractional uncertainty given its length and is readily detected during image processing. During the design phase of this research, the prototype module's design and components were selected in part based upon preexisting availability. For example, the diode source and diffraction grating were available from a prior student's work, as well as a set of digital SLR cameras, optical breadboard components, and so on. Module construction decisions were made in part due to the nature of the project; ultimately, the goal was to prove the feasibility of the concept and the underlying science, rather than to engineer and sell a commercial product. The selected design was not optimal with respect to overall performance but it was adequate for research and development of the optically projected length scale concept.

Following completion of the module's calibration and having tested the module during the measurement phase, it was necessary to examine some of the factors which limited the module's performance. By characterizing the influence of various module parameters on the uncertainty of the measurement, parameters which have a dominant role in the uncertainty can be prioritized, allowing targeted improvements to the module which yield the greatest reduction in uncertainty for the least amount of effort.

There are only a handful of parameters that can be changed during construction of the module. These parameters include the noise associated with the laser spots, the number of projected spots in the grid, the full angle of the pattern, and the physical separation distance and geometry between projection source origins. There are infinitely many module variations possible, and a given configuration may optimize performance for a user under specific measurement conditions. This section analyzes the impact of these design parameters on the overall uncertainty and explores some of the pros and cons associated with those choices. By understanding the basic trends, a designer can select module parameters which best fit their needs and simulate the module design to understand how that module will perform in the field. Furthermore, the existing photogrammetry system is discussed, including practical limitations of data processing on module design, and where improvements might be found.

5.1.1 Spot Noise

When performing photogrammetry measurements in conjunction with an optically projected length scale, it is difficult to assess the impact of various design parameters and outside effects on the quality of the measurement. For example, direct characterization of thermal effects and drift of the module is quite difficult. One can wish to determine the extent of the projection origin variation over the course of a given measurement, for example, but ultimately direct measurement of that point is not possible. Perhaps the user desires to know the impact of environment vibration on the module. Regardless of the source, all of these effects impact the projected pattern in the field, and in many ways that pattern is the only reliable source of data. The ability to interpret that data depends largely on the spot noise present.

Following the proliferation of coherent light sources in the 1960s, scientists immediately became aware of a phenomenon known as 'speckle'. Noise and variability in projected laser spot location arises largely from speckle, a name given to the pattern formed by the constructive and destructive interference of coherent wave fronts arising from the incidence of coherent light on a surface with roughness much greater than that of the source wavelength [60, 106, 107]. Illustrated in Figure 5.1, a typically Gaussian intensity profile is replaced by a speckle pattern, which manifests as a grainy mixture of high and low intensity. While the speckle pattern carries information about the surface and can be exploited by a variety of techniques for positive outcomes, in this application it negatively impacts determination of the true center, or centroid, of the projected spot.

Clarke (1994) compares various photogrammetry target types, including traditional high contrast black and white targets, retroreflettctive targets, and also laser targets [108]. In the latter case, the Gaussian profile of a typical laser offers the potential of a high contrast target that dramatically increases the SNR of the target, making it readily visible even above typical background illumination. In addition, the relatively narrow spectral bandwidth allows the removal of extraneous wavelengths by filtering. Speckle in the projected spots, however, is a major limitation which results in uncertainty in the centroid of the projected spot.

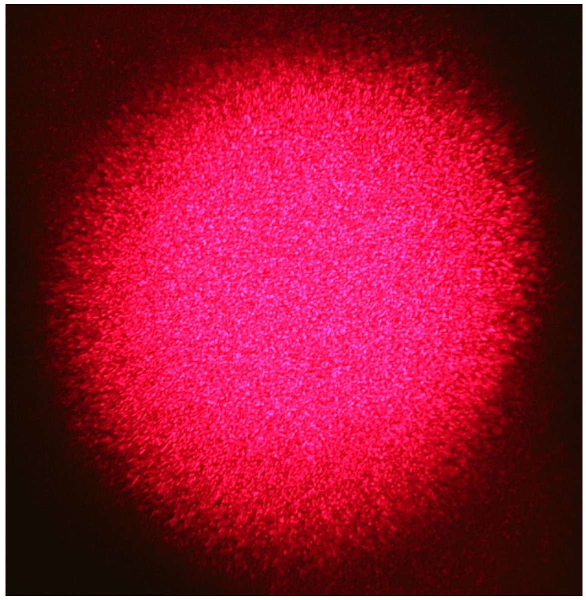
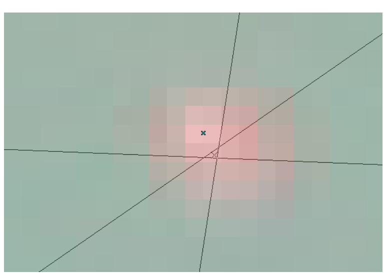
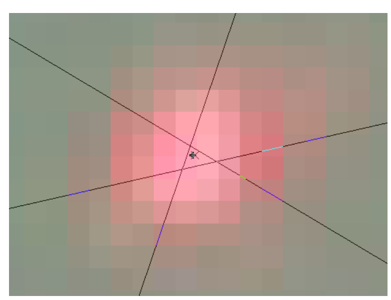


Figure 5.1: Subjective speckle pattern from a diode laser beam spot incident on a drywall surface.

Because of the coherent nature of a laser source, constructive and destructive interference of wavelets at the plane of the detector causes a speckle pattern, making centroiding difficult. As Clarke and Katsimbris (1994) note, this granular speckle pattern varies with viewing location and angle, resulting in variability of the target location [109]. This phenomenon is illustrated in Figure 5.2. In 5.2a, the location agreement is poor, with the automatically marked high intensity 'center' of the beam



(a) Centroiding Mismatch



(b) Centroiding Match

Figure 5.2: Speckle leads to variation in the perceived centroid of a laser projected spot. In (a), speckle induced intensity variation results in disagreement between the marked center and the expected center. In (b), the marked spot corresponds well with the the solved centroid.

spot not corresponding with the perceived center of the projected beam target. In comparison, 5.2b exhibits a more traditional Gaussian profile, and the automatically marked center in the final image agrees well with the other three images used for triangulation. Clarke and Katsimbris show that while a large aperture can be utilized as a method to decrease speckle size by averaging speckle over the area of a pixel, this can be undesirable for photogrammetry due to the reduced depth of field available for the measurement. Even with speckle reduced in the intensity profile, degradation in the location performance still occurs.

Dorsch et al. (1994) found that uncertainty in the centroid stemmed mainly from three sources: observational aperture, speckle contrast, and triangulation angle [110]. Decreasing contrast via reducing source coherence, increasing aperture size, and maximizing the triangulation angle to 90 degrees allows for the best reduction in centroid uncertainty. In the case of the latter, good network design means that a user is already striving for maximum triangulation angle. Extended sources with wider spectral bandwidth will also reduce the impact of speckle on centroiding.

Other methods can be used to reduce the spatial coherence of the source. Ellis (1979) vibrates an optical fiber with a piezoelectric transducer to change optical path length, thereby time averaging speckle to zero [111]. Asakura (1970) and Estes et al. (1971) describe a method by which spinning ground glass is used to reduce the spatial coherence of laser light [112, 113]. Davenport (1992) uses the spinning ground glass in conjunction with a multimode optical fiber bundle to reduce spatial coherence for laser-based biological imaging [114]. Stangner et al. (2017) provide a step-by-step description of that method in conjunction with focusing optics to re-couple the light into a multimode optical fiber [115]. This method was used by Zheng in his doctoral work to reduce uncertainty in projected laser targets by approximately 56% [7].

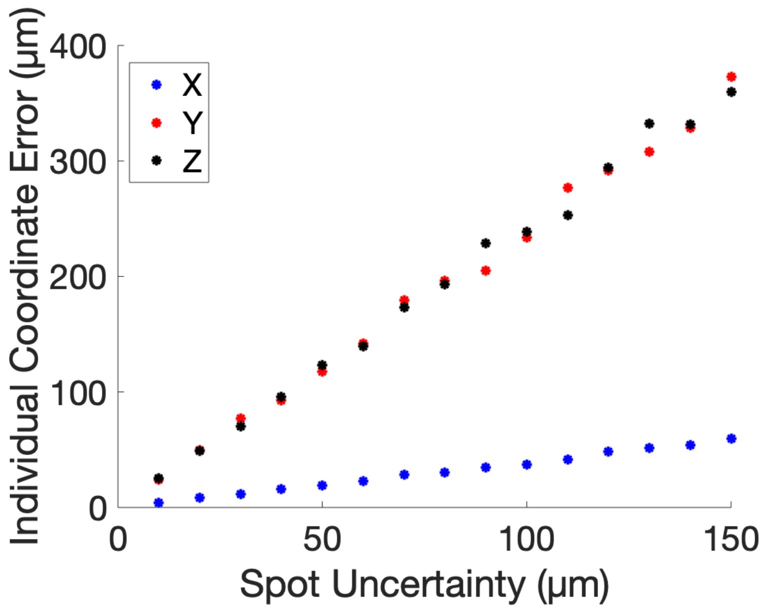


Figure 5.3: Spot location noise was uniformly applied to the X,Y and Z coordinates of a simulated projected grid. Standard deviation values for the solved projection origin coordinates are determined by a 500 iteration Monte Carlo simulation.

To gauge the impact of spot noise on the overall solved projection origin uncertainty, a Monte Carlo simulation was completed using existing module geometry and parameters ($\theta = 29°$, $N = 11$) to simulate the impact of spot noise levels on the solved projection origin X,Y,Z coordinates after processing through the algorithm. The plot shown in Figure 5.3 illustrates the linear impact of reducing spot uncertainty on the overall uncertainty of solved projection location coordinates. Spot noise was multiplied by random values from a Gaussian distribution, and applied to each coordinate dimension across all points in a simulated 11 x 11 grid. The grid's projection location was solved using the algorithm, with 500 iterations completed at each spot noise level.

It can be seen in Figure 5.3 that halving the spot uncertainty results in half the uncertainty in the individual coordinates. Especially in situations where the length scale might be short, reducing spatial coherence is a worthwhile endeavor to drive the fractional uncertainty down. Is such an approach feasible? There are a variety of methods that can be used to reduce spatial coherence, but all would serve to increase the size of the module. If portability is a concern, utilizing a small laser diode minimizes size and cost. In this case, typical projected spots show centroiding uncertainty on the order of 110/70/70 μm in the X,Y,Z coordinates of the spot. That said, a compromise in the form of an FC fiber optic connector on the module used in conjunction with a partially coherent external source could serve as a comfortable middle ground between portability and performance.

5.1.2 Pattern Full Angle

To understand the impact of the pattern full angle on the uncertainty of the solved projection origin, a 500 iteration Monte Carlo simulation was performed for pattern full angles ranging from 5° to 90° at 5° intervals. Figure 5.4 illustrates the inverse scaling of the projected grid pattern full angle on the uncertainty in the solved projection origin coordinates. This simulation was based on current module parameters of 11 beams per side, and 110/70/70 μm X,Y,Z average spot noise in the delivered photogrammetry coordinates. While increasing the full angle to 90° minimizes triangulation error, the pattern may be somewhat unusable in a larger scale industrial setting due to its rapid expansion in the field. Conversely, a very narrow field may allow for pattern projection to a distant surface, but solved origin uncertainty will suffer, and thermal lensing could become an issue over long projection lengths. By increasing pattern full angle to 60°, a 3x to 4x reduction in solved coordinate uncertainty is observed in comparison to the current 29° module pattern, while preserving usability of the module in the field. A theoretical improved module with a 60° pattern full angle is proposed, simulated, and assessed in Section 5.2.

A module's pattern full angle also implies a desirable module geometry in order to take full advantage of the projection origin error ellipse. Where a 90° module with near identical X,Y,Z uncertainty may benefit from projection sources being positioned in side-by-side horizontal layout, a narrower pattern angle with larger Y,Z uncertainty will benefit from one projection source being staggered rearward.

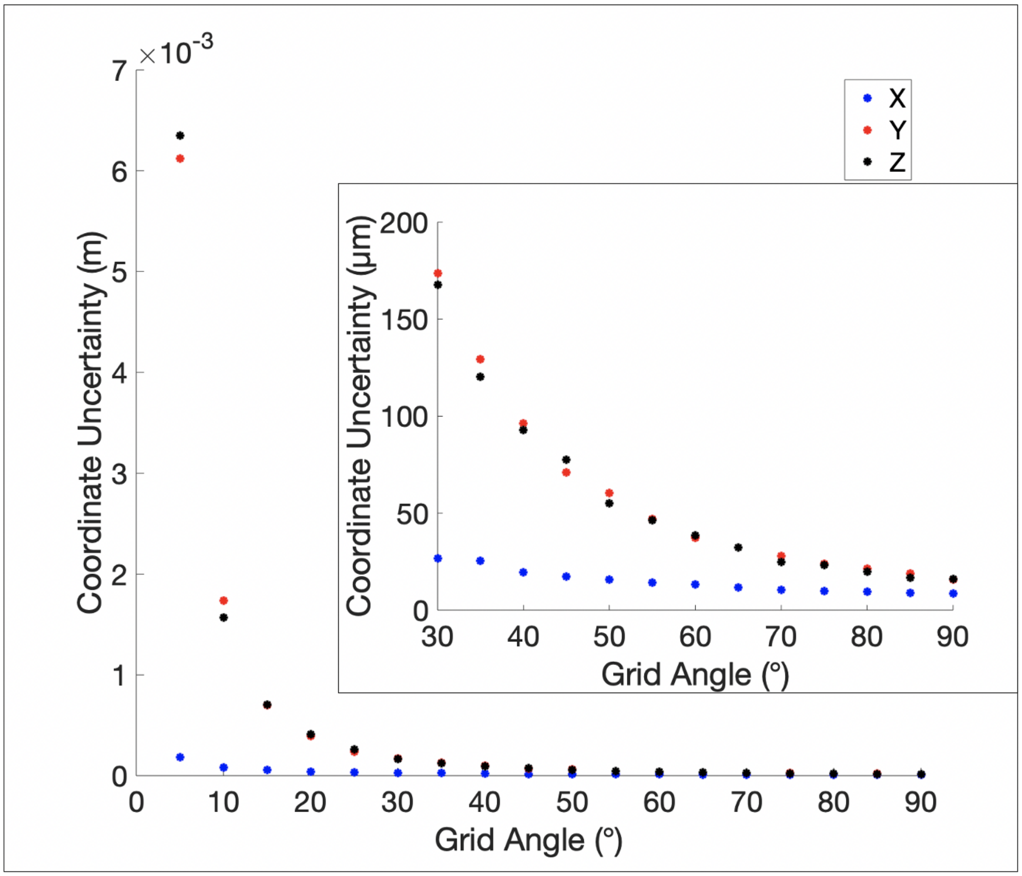


Figure 5.4: Monte Carlo simulation to explore impact of pattern full angle on solved projection origin coordinate uncertainty. N = 11 beams per side. 500 iterations per grid angle.

5.1.3 Beam Number

Similar to adding more coded targets to a photogrammetry scene, it is expected that increasing the density of optically projected targets will similarly drive down coordinate uncertainty. A Monte Carlo simulation was performed under current module parameters of a 29° pattern full angle and average 110/70/70 μm spot noise in the X,Y,Z respectively. 500 iterations were performed at each beam number of N, with N ranging from 5 to 35 beams per side at odd number values. Figure 5.5 shows that the coordinate uncertainty scales inversely with increasing beam number. The higher level of coordinate uncertainty in the Y and Z are a result of the relatively narrow 29° degree pattern full angle. Should a wider angle be selected, a reduction in Y and

Z uncertainty is expected until a minimum is reached with a 90° full angle pattern.

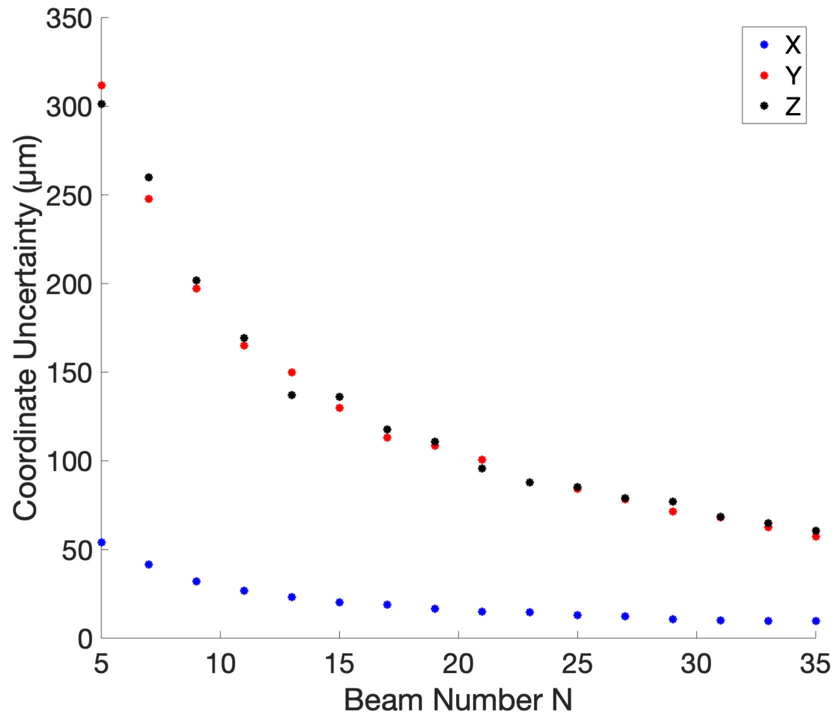


Figure 5.5: Monte Carlo simulation to explore impact of pattern beam number N (per side) on solved projection origin coordinate uncertainty. 500 iterations per pattern beam number value with full angle $\theta = 29°$.

At a casual glance, one might think that increasing the beam density per side as high as is available could be beneficial in reducing the overall coordinate uncertainty. That would be true, but it does not account for the extra processing time required. In the prototype module, an 11 beam per side diffractive optical element was used as it had been purchased for a prior experiment and was immediately available. The goal was not to produce the best performing optically projected length scale module, but to create a working prototype and algorithm which allowed for measurement and comparison against simulated models.

The current 11 x 11 grid produces 121 optical targets which must be marked and processed. To move to a 21-sided grid results in a $1/3^{rd}$ reduction in Y and Z uncertainty, but produces a 441 point pattern, or roughly a 360% increase in targets requiring marking and processing. A significant bottleneck in the measurement process is the ability to process data. While a measurement takes seconds, the post processing for a single 11 x 11 grid measurement requires approximately 15-20 minutes. As the density of optical targets increases, so does the processing burden on the user and software. This burden is further discussed in Section 5.3.

5.1.4 Projection Origin Separation

The fractional uncertainty of a traditional length scale artifact is defined as the ratio of the uncertainty of the length divided by the length of the artifact, or $\delta L/L$. Commercially available Invar artifacts approach 3 meters in length and offer uncertainty on the order of parts per million [5] with good thermal stability. Ideally, the photogrammetrist will use an artifact with a length that is on the order of magnitude of the part to be measured. Because the artifact is used to scale all of the coordinates, long length with low fractional uncertainty is ideal. In this work, a large number of points are leveraged to statistically drive down the uncertainty associated with each projection origin rather than relying on a long artifact to reduce fractional uncertainty. Regardless, it is nevertheless beneficial to create as large of a separation between projection origins as is possible in order to further reduce fractional uncertainty. Using a single diffractive optical element and duplicating the grid pattern with a fold mirror places a practical limitation on the magnitude of the separation between projection origins. Using a wide angle pattern in conjunction with a fold mirror could result in the pattern over-filling the aperture of the mirror at large separation distances. By using two diffractive elements rather than one, the separation distance can be increased while maintaining portability.

5.2 Theoretical Improved Module

Over the course of this chapter, the roles that module parameters such as beam number and pattern full angle play in the uncertainty of the solved projection origin locations have been discussed. By reducing this uncertainty, the overall fractional uncertainty of the optically projected length scale can be decreased. In this section, changes to module parameters and geometry are incorporated into a revised module design, and its performance is simulated.

5.2.1 Design Considerations

The current module was fabricated with off-the-shelf components to keep costs low. Its handheld design offers portability for lab use while providing a platform which serves to experimentally prove the concept of an optically projected length scale. The module utilizes a small optical breadboard with post mounts, a nitrocellulose beamsplitter, and a kinematically mounted square mirror to duplicate the 29° full angle 11 x 11 beam pattern. The second pattern origin is offset from the first by approximately 39 mm horizontally and 39 mm to the rear. The resulting separation between projection origins is approximately 55 mm.

A revised theoretical design, illustrated in Figure 5.6, is proposed which uses a wider grid angle and increased projection separation. Portability is maintained as the module could potentially fit into an enclosure no larger than the size of a TV remote control. The pellicle beamsplitter and kinematically mounted mirror originally used to duplicate the pattern are discarded in favor of a design utilizing two diffractive optical elements. A fiber-coupled laser source connects by FC connector to the module and is incident on a 50:50 cube beamsplitter. One beam passes through the beamsplitter and is incident upon a diffractive optical element to generate the first beam spot pattern. The reflected beam is incident on a right angle prism mirror, before being reflected and passing through its own diffractive element to generate the second pattern. The separation between grid projection points is increased by nearly 4 times compared to the current prototype module while maintaining a similar footprint. By removing the kinematic mount and eliminating the post-holder design, the module is more robust and less prone to drift.

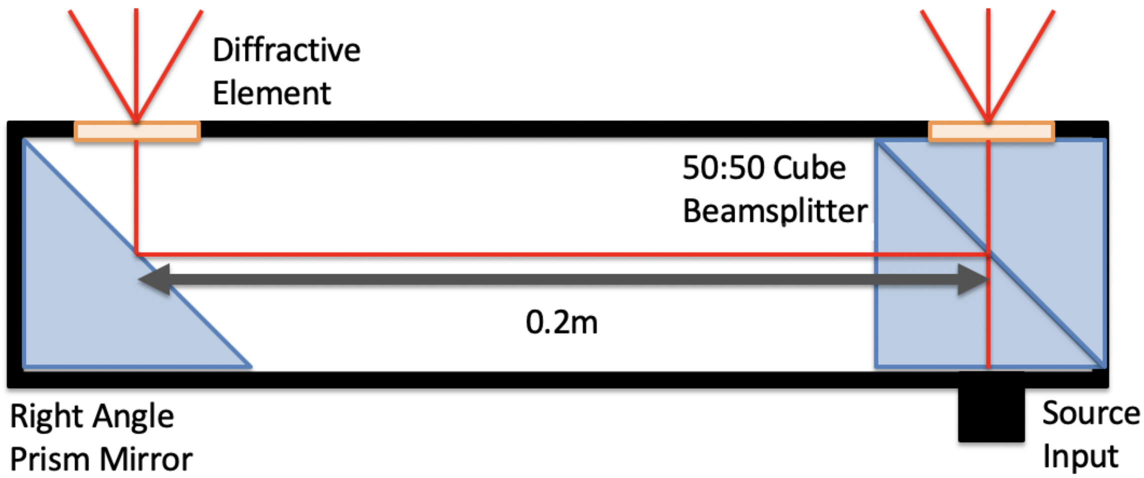


Figure 5.6: To reduce fractional uncertainty of the optically projected length scale, the module was redesigned. The kinematic mount was removed, with a right angle prism mirror and cube beamsplitter directing the beam to two diffractive elements for pattern creation.

5.2.2 Simulated Performance

A 5000 iteration Monte Carlo simulation was performed to assess the performance of the current module against the theoretical improved modules, with the simulated results compared against the existing module in Table 5.1. Full angle variations of 60° and 90° were tested, each with 0.2 m projection origin separation. Beam number was held constant at 11 beams per side. While the 90° full angle design performed the best, an order of magnitude improvement in fractional uncertainty to 3 parts in 10^4 for the 60° full angle revision is observed along with a reduction in coordinate error to less than $1/3^{rd}$ of the current module's estimated projection origin coordinate error.

Table 5.1: A comparison of the current prototype module simulated performance relative to revised module designs which utilize an increased projection origin separation length and wider pattern angle. Portability is maintained and cost remains low for all versions.

	Current Design	60° Revision	90° Revision
Grid Full Angle	30°	60°	90°
Beam Number N (per side)	11	11	11
Separation Length	55.154 mm	200 mm	200 mm
Length Scale Error	186μm	69μm	30μm
Fractional Uncertainty	3.4 in 10^3	3.4 in 10^4	1.5 in 10^4
Coordinate Error (X/Y/Z)	30/180/180μm	17/49/49μm	14/22/21μm

The proposed module revisions are not intended to maximize performance, but rather to illustrate order of magnitude gains that can be made with just a few adjustments to module parameters while preserving the portability and low cost. Additional gains could be realized if the beam number was increased, and a partially coherent source integrated into the module.

5.3 Photogrammetry System & Data Processing Improvements

It deserves mention that while the image acquisition portion of the measurement process takes mere seconds, post-processing of the images can take hours. While sub-optimal, a four camera network with wide angle lenses and a single image per station was chosen to allow for the minimal amount of photographs necessary for processing. As noted by Fraser, hyper-redundancy of photographs, or the dramatic increase of photographs taken per station, allows for a simple way to reduce the mean standard error of coordinates, $\bar{\sigma}_C$ [31], and can be estimated by Equation 5.1 [32]:

$$\bar{\sigma_C} = \frac{\sigma}{\sqrt{k}} q(d/f), \tag{5.1}$$

where

σ is the image coordinate standard deviation,

k represents the number of images divided by the number of camera stations,

q is an empirically determined scalar,

d represents the object distance, and

f is the focal length of the camera.

While there are practical limits to increasing the number of photographs per station, should processing of the optical pattern become automated through purpose-built software, additional photos per station would impose only a small burden on the user. In addition, not only would an automatic solution dramatically decrease post-processing time, it would also allow an increased number of points in the grid. These changes would further decrease the uncertainty associated with the solved projection location origins, thus reducing the fractional uncertainty of the optically projected length scale.

In addition to the revised modules discussed in Section 5.6, simulations were conducted for a module which used not only the 90° full angle and 0.2 m separation, but also utilized 41 beams per side. The resulting simulation yielded a fractional uncertainty of 5 parts in 10^5, with coordinate uncertainty of 3/7/6 μm in the X,Y,Z. Such a dramatic increase in number of optically projected points of the module would require sophisticated automatic identification and processing of the data. But, performance approaching that of a traditional retroreflective Invar bar is theoretically achievable if the separation length is simultaneously increased. There are infinitely many possible combinations of full angle, beam separation, points per side, and so on. Ultimately it is up to the system designer to prioritize performance versus portability versus processing burden, and make an informed choice.

A final mention is made here of retroreflective targets in photogrammetric measurements. Retroreflective film is constructed from glass microspheres affixed to a tape surface with an adhesive, yielding a highly reflective surface that actively returns light to the its source. Since their proposed use in 1980 by Brown, retroreflective photogrammetry targets have become industry standard in most commercial systems [116]. When used in conjunction with a flash, retroreflective targets offer a method by which target intensity can be controlled while minimizing or eliminating background illumination, thereby improving the SNR of a given target. Retroreflective targets are not used in the system described in this research, though the high intensity nature of a laser targets similarly improves the SNR [108]. It is noted, however, that passive black on white targets can offer performance similar to to that retroreflective targets under correct lighting conditions [26]. Using passive targets, Seitz (1988) demonstrates measurement precision to $1/100^{th}$ of a pixel [117].

5.4 Optomechanical Design Analysis

Following module construction, potential hardware error sources were identified within the apparatus which could be analyzed for their contribution to uncertainty. These contributors were:

- Diode Laser Source
- Kinematic Mirror Mount
- Pellicle Beamsplitter

In the following sections, a basic optomechanical error analysis is performed on each of these and, if possible, their error contribution to the measurement is characterized.

5.4.1 Diode Source

It is important to note that laser wavelength stability can have a profound impact upon the error associated with this apparatus. The module utilizes a holographic diffraction element, and while wavelength variations should be closely examined in conjunction with the IFTA responsible for the grating design, the periodicity of the structures are such that wavelength variation in the system can be crudely modeled by the same grating equation as a traditional blazed or binary diffractive element. Angular error in the system can be modeled with the diffraction grating equation:

$$dsin\theta_m = m\lambda, \tag{5.2}$$

where

d is the grating spacing (estimated 1.26 x 10^{-5} m),

θ_m is the angle between the diffracted ray and the grating normal for a given order,

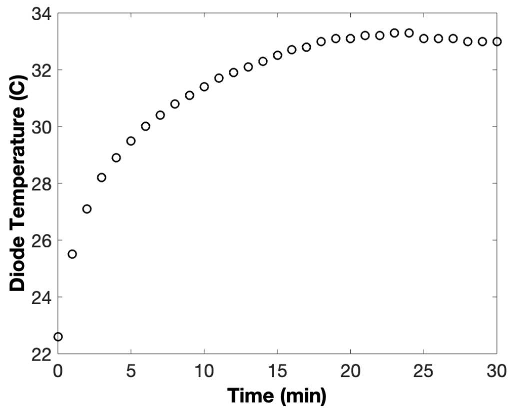
λ is the wavelength, and

m is the order number.

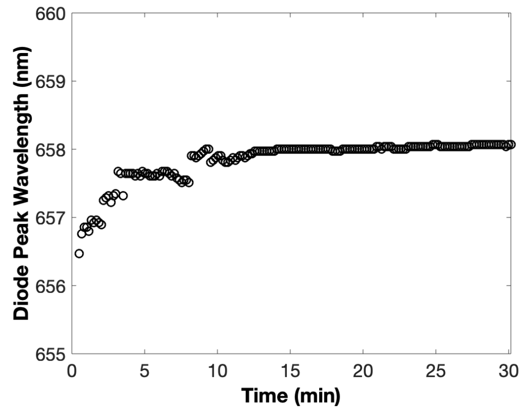
Solving for the variation in θ_m, the error in the diffracted rays for the m^{th} order is expressed:

$$error_m = \arcsin\left(\frac{m(\lambda + \Delta\lambda)}{d}\right) - \arcsin\left(\frac{m\lambda}{d}\right). \tag{5.3}$$

Generally, band gap energy varies inversely with wavelength, as $E_{BG} = hc/\lambda$. As temperature increases, the band gap energy decreases between the valence and conduction bands of a semiconductor [118]. While cavity length is also impacted by temperature, the greatest change in wavelength comes from change to the band gap energy. Temperature increase in the diode results in red shifting of the laser wavelength. The impact of temperature on the wavelength of a typical AlGaInP



(a) Laser Diode Temperature



(b) Laser Diode Peak Wavelength

Figure 5.7: Laser Diode Temperature and Wavelength

diode is on the order of 0.2 nm per degree Celsius [119].

In this application, a thermoelectric cooled (TEC) diode is avoided for the sake of cost, power consumption, and size. While a TEC controlled diode would be advantageous from a stability perspective, the idea of this system is to have a small, portable optical device that is cost competitive against a traditional bar style photogrammetry length scale. In a higher cost system, a TEC could provide a more stable source. Another alternative is a Helium Neon (HeNe) laser, which displays excellent wavelength stability on the order of parts in 10^6 to 10^7 [120].

To calculate the error in the diffracted rays using Equation 5.3, the diode was characterized for both temperature and wavelength, shown in Figure 5.7. A thermocouple was affixed with thermally conductive metallic tape directly to the outside of the diode's aluminum housing, as close to the internal location of the diode as was possible. The temperature was sampled at 1 sample per minute for 30 minutes. Note that in Figure 5.7a, stable state occurs around the 15-20 minute mark at approximately 33° C. The drop in temperature at this point was likely due to HVAC cycling in the lab. From startup at ambient (T = 21 °C), the diode exhibited an approximate 12° C increase in diode temperature.

The laser diode's peak wavelength shown in Figure 5.7b was measured with an Avantes AvaSpec-3648 spectrometer, sampling every 4 ms, for 30 minutes. At steady state operation beginning around the 15-20 minute mark, an average wavelength of 658.03 nm is observed, with a range from the 15-30 minute mark of 0.10 nm, and a standard deviation of 0.03 nm.

At steady state operation, a worst-case approximate 0.1 nm temperature variation is assumed. Plugging this fluctuation into Equation 5.3 solves for a diffracted ray error of $\pm$ 8.0 μrad at steady state operation. At a projection distance of 1.5 m, this would result in approximately 12 μm of variation in the spot location along the Y and Z axes. This is nearly an order of magnitude smaller than the spot centroid variation due to speckle. Disregarding the pattern variations that occur during laser instability during startup, it is therefore assumed safe to neglect the error due to angular variation in diffracted pattern spots during steady state operation.

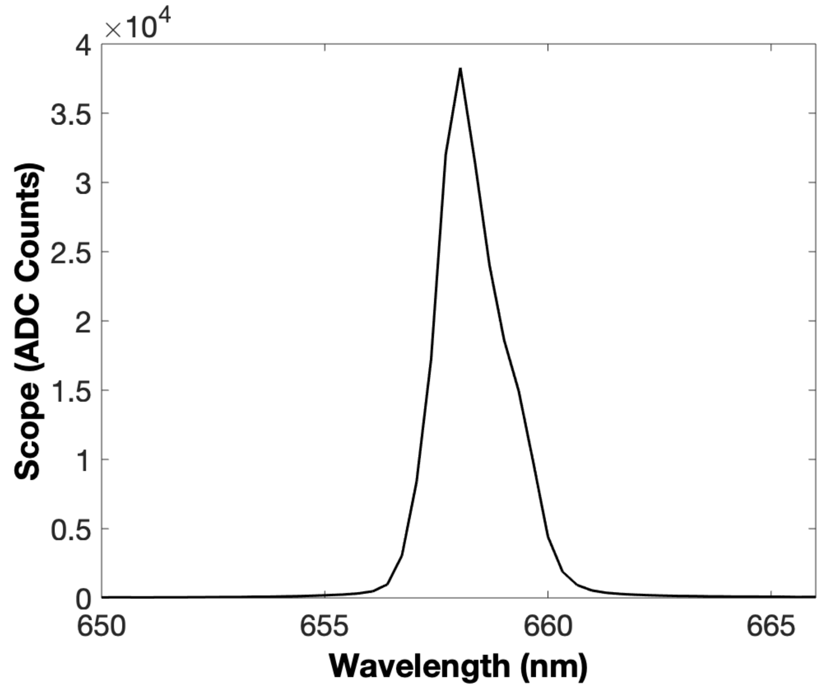
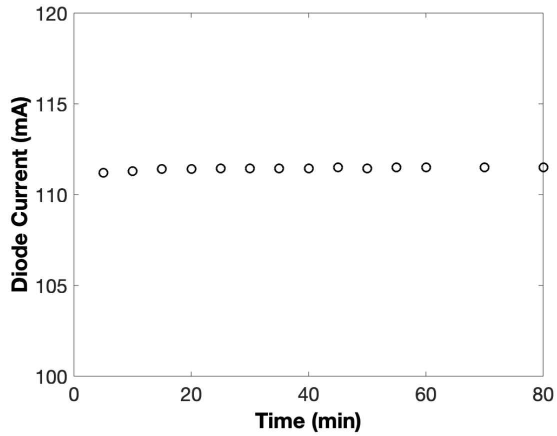
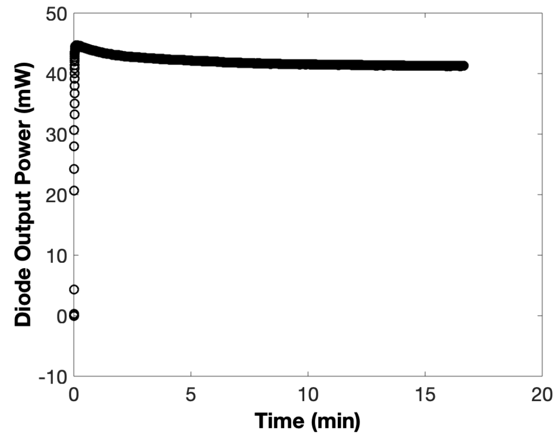


Figure 5.8: 658 nm laser diode spectrum. FWHM approximately 2 nm.



(a) Laser Diode Current



(b) Laser Diode Output Power

Figure 5.9: Laser Diode Output Power and Current

The spectral line width was measured, shown in Figure 5.8 and is approximately 2 nm full width half maximum (FWHM). Spectral broadening results in an elongation, or smearing, of the diffracted spot. This does not significantly impact the centroid of steady state targets, and thus is ignored.

In addition to characterizing the wavelength and temperature of the diode, it is imperative to examine the laser's output power and current in order to calculate the diode's efficiency. From this data, the power is dissipated into the module by the diode can be found, and the source's thermal impact on the system explored. The output power was measured with a Thor Labs PM100USB power meter and 10,000 samples were taken at a rate of 0.1 seconds per sample for approximately 17 minutes. The diode current was later monitored, with 14 samples taken over an 80 minute duration.

Figure 5.9a illustrates current fluctuation on startup. The output power also displays similar variation during the first few minutes of operation, as observed in Figure 5.9b. Current stabilizes to around 111.5 mA, with an RMS output power of 42.0 mW. This yields a power to current ratio of approximately 0.4 mW/mA, which is on par for typical laser diode efficiency.

With a diode efficiency of approximately 40%, 0.35 W is dissipated by the diode into its housing, apparatus, and environment at the diode operating voltage of 5 V. As the prototype is not enclosed and in open air on an optical breadboard, this may not be a problem. The diode is mounted on a 3 inch optical post. It is worst-case assumed that all power dissipated by the diode travels down the post, and into the to the 1/2 inch optical post holder and optical breadboard.

Thermal resistance is described:

$$R_t = \frac{L}{kA}, \tag{5.4}$$

where

L is length,

k is thermal conductivity, and

A is cross sectional area.

Inserting the values for the stainless steel post:

$$R_t = \frac{0.762m}{14.7(W/m * K) \times 0.127m^2} = 0.4082K/W. \tag{5.5}$$

Solving for the temperature change at the bottom of the post:

$$\Delta T = R_t \times Q, \tag{5.6}$$

where

Q is the watts dissipated.

The estimated change in temperature at the bottom of the post is therefore:

$$\Delta T = 0.4082K/W \times 0.35W = 0.1429K. \tag{5.7}$$

The estimated one to two tenths kelvin temperature change expected at the bottom of the post holding the diode, in an open air setting, can be assumed to be negligible. Any minor increase would be dissipated by the aluminum post holder and breadboard with minimal impact on breadboard dimensions, or the mirror and pellicle mounts.

5.4.2 Kinematic Mount

While not necessarily the best choice for an end-user product, off-the-shelf optical mounts and posts serve as an important prototyping tool for any optical engineer during the research and design phase. These post mounts are examined to get an estimate of deflection, along with the mirror mount utilized in the prototype to get an idea of mirror drift with thermal variation.

Thor Labs 1/2 inch post mounts are used to hold the laser diode assembly, the pellicle beamsplitter, and the 2 inch square mirror with kinematic mirror mount. Each post with mounted component has its own unique resonant frequency. Flex in the mount may also be an issue if the load is sufficiently high, but as the largest optical component still weighs less than a pound, the major error likely originates from the mirror mount. As shown in Newport Labs data in Figure 5.10, the deflection at such low loads amounts to fractions of a milliradian, or approximately 5 μm of deflection at 3 inches [121]. This contribution is negligible in light of the other larger error sources.

The mirror mount used in the module is a Thor Labs 2 inch square aluminum mirror mount. Thor Labs does not provide drift estimates or data for aluminum mounts, but Newport Labs does provide data for a basic steel mount versus their Suprema ZeroDrift Thermally Compensated Mirror Mounts [122].

For a steel mount, Newport Labs calls out approximately a maximum of 35 μrad pitch drift with a 12° C temperature change. The yaw variation over the same temperature drift is significantly less - a maximum of 4 μrad. Using this as a baseline, and making the admittedly generous assumption that drift is due to the material only,

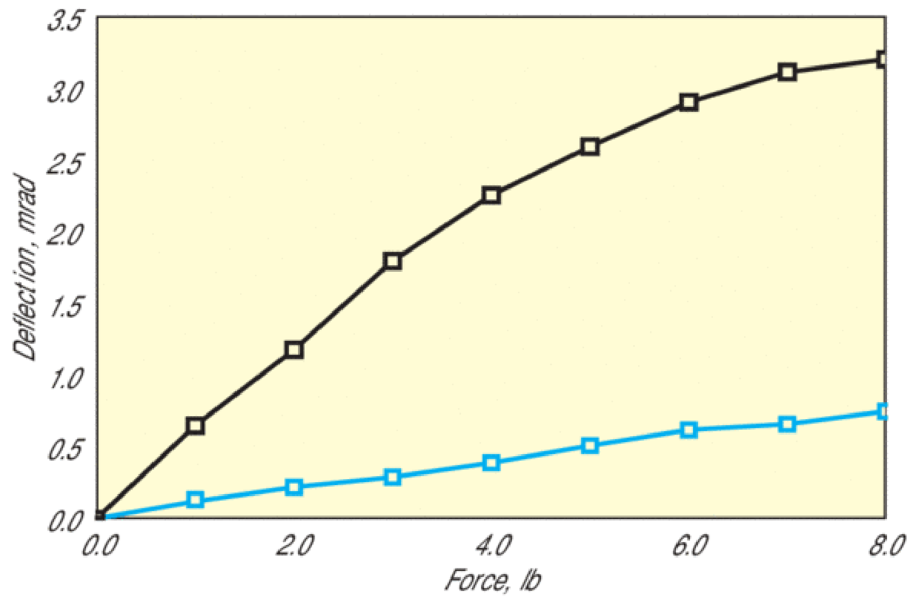


Figure 5.10: 1/2 inch post (black) vs. 1 inch pedestal (blue) flex under load. *Image Credit: Newport Labs*

rather than design or construction differences between brands, twice the expansion is assumed for aluminum versus steel. This yields values of 70 μrad and 8 μrad for pitch/yaw drift with the aluminum mount in question over the same 12° C range.

Note that the yaw drift is approximately $1/10^{th}$ the drift of the pitch. Because of the staggered orientation of the module projection origins, pitch drift of the mirror has a less substantial impact upon the calibrated length compared to yaw in the overall calibrated length scale. Applying the tangent of 8 μrad at 2 inches, a lengthening of the calibrated length scale on the order of parts in 10^5 over the 12° C temperature variation can be estimated. At steady state operation in a lab environment, mirror mount drift is thus assumed to be negligible.

5.4.3 Pellicle Beamsplitter

The module utilizes an off-the-shelf Thor Labs B145B1 45:55 beamsplitter which provides approximately 50% reflectivity at 658 nm for a 45° angle of incidence. The pellicle beamsplitter is constructed from a cylindrical anodized aluminum ring, over which a 2 μm thick nitrocellulose film has been adhered under tension. In essence, the pellicle beamsplitter resembles a small drum, and can be modeled as such for the sake of vibrational analysis [123]. A drum's fundamental mode (F_{01}) is given by the following equation:

$$F_{01} = 0.766\left(\frac{\sqrt{T/\sigma}}{D}\right), \tag{5.8}$$

where

F is the fundamental frequency of membrane,

T is tension of the membrane,

σ is area mass density of the membrane, and

D is diameter of the membrane surface.

The diameter of the pellicle is 1 inch, or 0.0254 m. Unfortunately, many of the variables listed above in Equation 5.8 were not published by Thor Labs for this particular membrane material, and were unavailable upon request. The density of nitrocellulose was given as 770 kg/m^3. Area density is determined as:

$$\sigma = \rho \times l, \tag{5.9}$$

where

ρ is material density, and

l is layer thickness.

With layer thickness known to be 2 μm, and density given, area density is calculated σ = 0.00154 kg/m^3. As noted above, tension data was not provided for the pellicle film. While this can be measured via probing the surface directly, this risks damaging a multi-hundred dollar optical element, and was not recommended. Instead, one can work backward from Ultimate Tensile Strength (UTS), which is published as 9000-16000 psi at 24° C and 50% humidity for this material. Tension is estimated with a UTS value of 5000 psi, which is below the point of plastic deformation in the material.

As pounds per square inch is a measure of force per unit area, one can solve backward for the force applied over the least cross sectional area of the material, which is the thickness. Let the width of the pellicle cross section area equal 1 inch. The nitrocellulose layer has a thickness of 7.87 $\times 10^{-5}$ inches. Solving for the cross sectional area:

$$\text{Area} = 1 \text{ inch} \times 7.87 \times 10^{-5} \text{ inches} = 7.87 \times 10^{-5} \text{ inch}^2.$$

Applying 5000 psi over this area, the force is approximately F = 0.4 lbs, or 1.78 N. This force needs to be applied over a dimension of the pellicle in order to set the tension of the pellicle. One can choose either the pellicle diameter, or its thickness. Considering that the strength in this film is due to forces pulling parallel to its surface, this direction is selected. Note that if the diameter increases, there is a decrease in tension (given the same force), and a subsequent decrease in the fundamental frequency.

With all variables known, the fundamental frequency of the pellicle is solved below in Equation 5.10:

$$F_{01} = 0.766\left(\frac{\sqrt{\frac{1.78N}{0.0254m}/0.00154kg/m^2}}{0.0254m}\right), \tag{5.10}$$

$$F_{01} = 6.433kHz.$$

While the fundamental frequency is outside of the worst frequency range of a typical industrial acoustic spectrum, the pellicle could still experience acoustically forced displacement in the right environment. It deserves mention that F_{01} is on the kilohertz level. Given that the cameras used for collection of photographs have an exposure time on the order of tenths of a second, it is reasonable to assume that whatever deviations occur in the pellicle surface would be averaged many times over during the course of a single image. The target smearing that could result in the spot profile would likely form elliptical spots due to the radial symmetry of the fundamental mode, and thus the centroided center the spot should not in theory change. Higher order modes are ignored for the sake of this analysis.

Lastly, it is noted that the efficiency of the pellicle beamsplitter is specified as 99.5% at 658 nm. This results in an estimated absorbed power of 0.0002 W by the pellicle membrane. While the Coefficient of Thermal Expansion (CTE) of nitrocellulose is somewhat high ($\alpha = 100$), the comparatively thin 2 μm material thickness would likely rapidly dissipate any absorbed energy back into the environment, rather than making its way to the aluminum ring. The thermal conductivity of the nitrocellulose film could not be found, and as such, heat transfer to the ring, expansion of the membrane, along with the change in tension and fundamental mode could not be computed for this work.

5.5 Module Stability & Beam Drift Test

During the measurement trials, a slight bias in the solved optical length scale was observed. This was discussed in depth in Section 4.5. Following the measurement trials, a handful of questions remained about the impact of environmental factors on the module's performance and stability. Environmental stability of the lab was examined to determine whether thermal variations could be causing significant dimensional changes in the module, the projected pattern, or shifting in the global coordinate system origin during normal operation.

To accomplish this, a 24 hour drift test was conducted. The temporary pattern beam block was removed from the module so that both projected patterns would be visible in the photogrammetry scene, as shown in Figure 5.11. During the drift test, three sets of beams, labeled A, B and C, were monitored. Pair A was located on the Y axis of the projected grid, and pair C was located on the Z axis. Pair B represented the central beams. As these propagate directly through the diffraction grating, they can be used to assess changes in the module only, while ignoring fluctuations in the source which could alter the projection angle of higher order beam pairs. For each beam pair, left and right beams were labeled, denoted B_L and B_R for the central beam pair. Pairs A and C were similarly categorized. A type K thermocouple was affixed to the corner of the optical breadboard to monitor the temperature, shown in Figure 5.11.

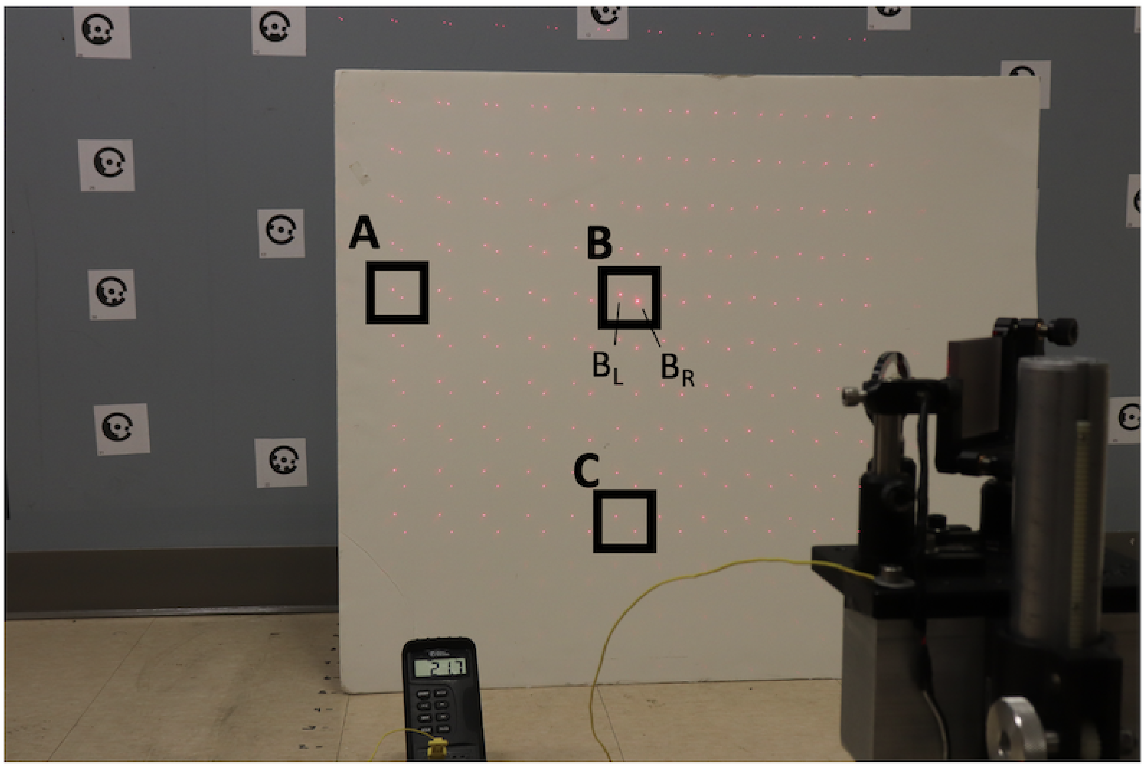


Figure 5.11: Beam stability test. Three beam pairs, A, B, C, are monitored hourly over a 24 hour duration. Photogrammetry is performed to find beam spot locations. Relative drift between beam pairs yields information about module behavior.

The measurement was automated using a script in the digiCamControl software, which took 3 sets of photographs per hour for a 24 hour duration. The lab was sealed with no researchers present during the duration of the measurement, and a one hour dwell was incorporated into the software to allow the module to reach thermal equilibrium prior to beginning image capture. Photogrammetry was performed on each image set, and coordinates were scaled by the calibrated length scale artifact present in the scene.

Table 5.2 shows the standard deviation in the solved X,Y,Z coordinate positions for each of the six observed beam spots over the 24 hour measurement duration. In general, the reported uncertainty of laser projected spots as reported by PhotoModeler® was on the order of 110 μm, 70 μm, and 70μm in the X,Y,Z, respectively. The standard deviation σ_X for C_L and C_R was slightly higher than the rest of the group, but those spots were also in an area of the scene where there were no other RAD targets or laser projected targets present to provide additional stability during the bundle adjustment. From this initial data, there did not appear to be any significant drift in the points that would be outside of the norm for laser projected targets in a photogrammetric environment.

Table 5.2: Standard deviation of X,Y,Z coordinate position for tested spots over a 24 hour measurement duration.

	σ_X (mm)	σ_Y (mm)	σ_Z (mm)
A_L	0.06	0.06	0.09
A_R	0.09	0.08	0.07
B_L	0.07	0.06	0.03
B_R	0.07	0.04	0.06
C_L	0.13	0.05	0.06
C_R	0.11	0.05	0.07

An additional method to assess drift in the module itself is to look at distance between the positions of B_L and B_R, which are the zeroth order beams in the pattern, and thus not susceptible to changes in the source wavelength impacting their projection direction. By looking at the relative distance between B_L and B_R, potential changes in the movement of the global coordinate system origin, which could occur due to thermal variations in the lab, are ignored. Figure 5.12 shows the Euclidean distance between the points plotted in blue as a function of time. Additionally, the temperature of the module has been plotted in red along the second vertical axis on the right side of the figure. At a glance, there does not appear to be a significant

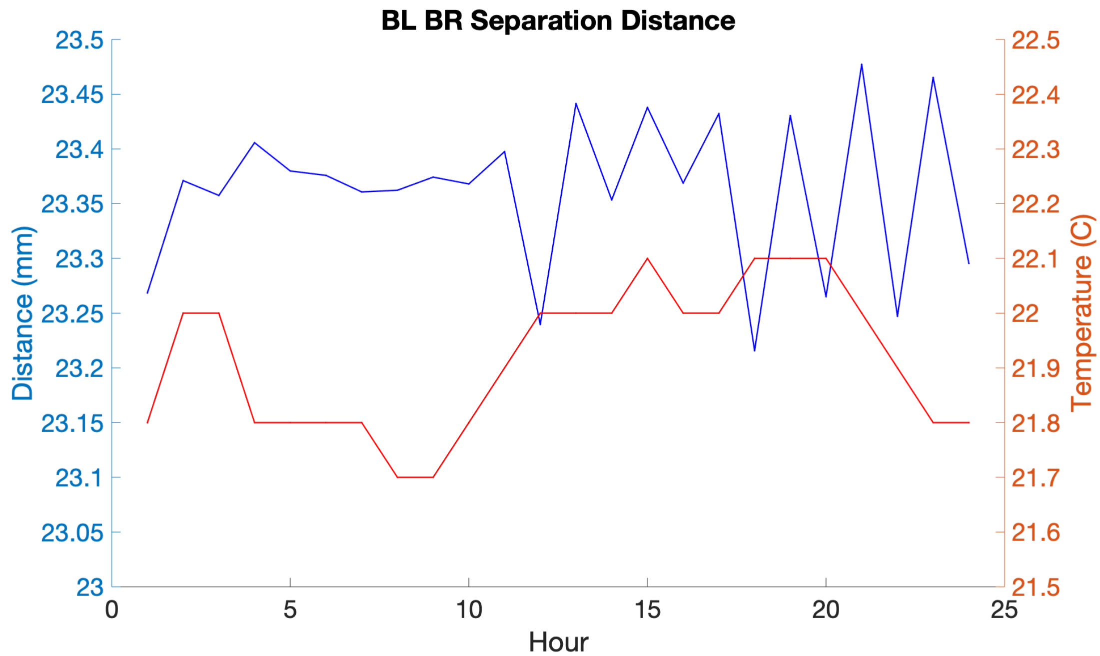


Figure 5.12: Euclidean distance between central order spots B_L and B_R as a function of time.

correlation between changes in the local temperature and the separation distance between B_L and B_R. The standard deviation (1σ) of the calculated distance between B_L and B_R is approximately 97 μm. A loose correlation might be observed between temperature and distance in A_L and A_R, as shown in Figure 5.13. That is not observed in C_L and C_R in Figure 5.14, however, and there is simply not enough evidence to support the idea that the minor change in temperature is responsible for pattern position and angle changes, rather than scatter in the points due to speckle.

Finally, the position of a given spot can be sequentially plotted to see if the position is changing with time in a structured manner. The positions of B_L and B_R are plotted in Figure 5.15, and appear random, with location changes that are not correlated with any particular temperature change direction. While not pictured, the paired plots of A_L, A_R and C_L, C_R show similar random paths over the duration of the measurement.

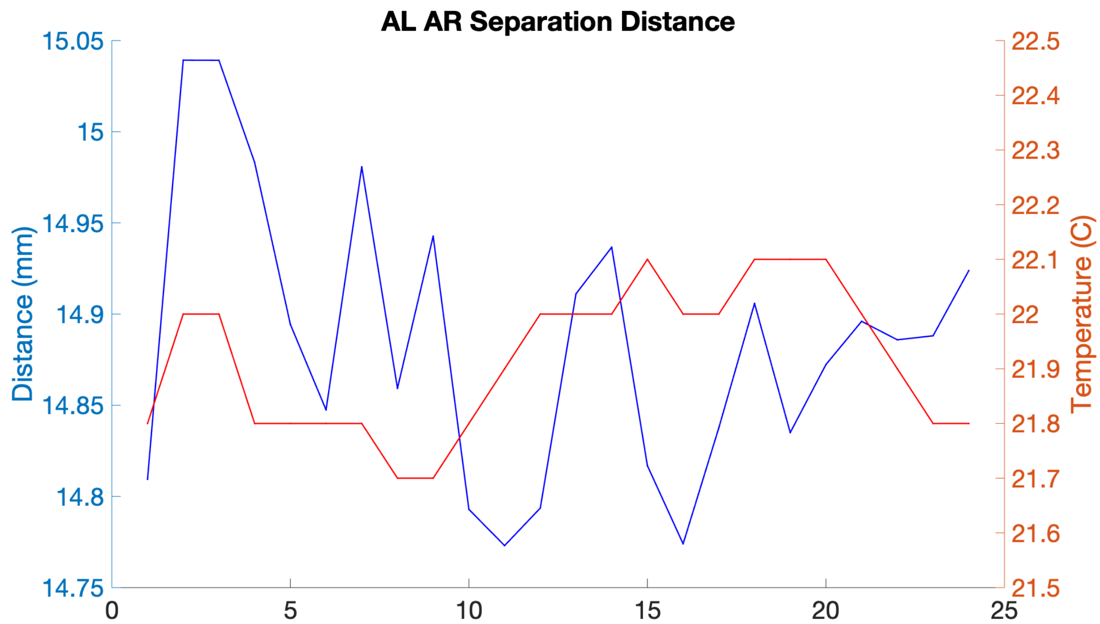


Figure 5.13: Euclidean distance between central order spots A_L and A_R as a function of time.

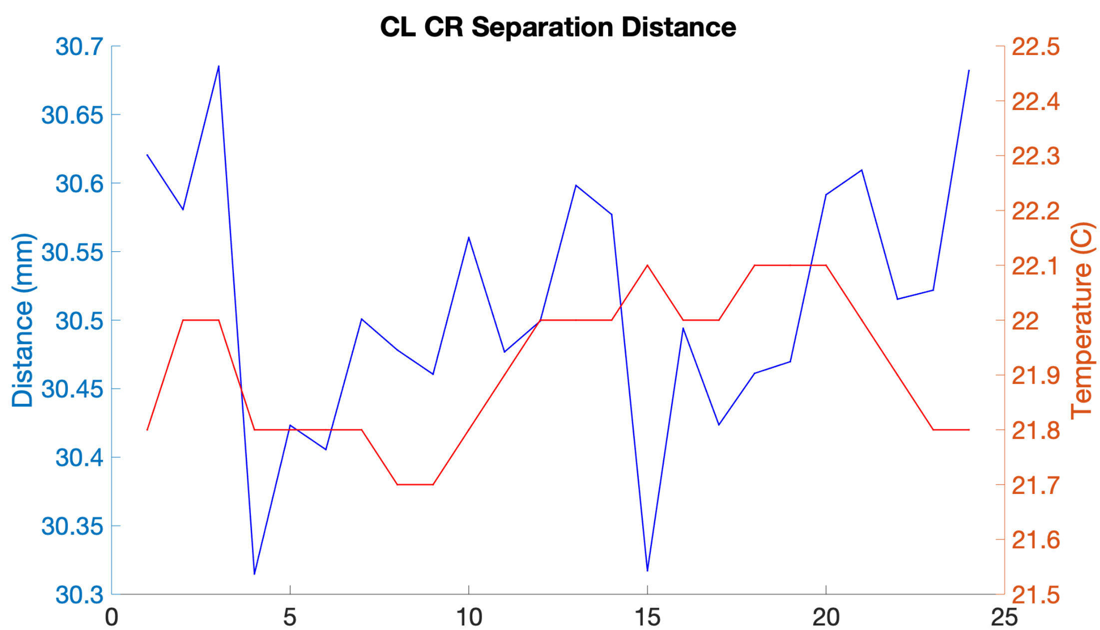


Figure 5.14: Euclidean distance between central order spots C_L and C_R as a function of time.

Ultimately, drift in the module and its projected beam spots is minimal over a 24 hour duration compared to the scatter in the spots that naturally arises from speckle. There isn't enough evidence over the $\approx \pm 0.1°$ C temperature change to see meaningful changes in the pattern or locations with temperature.

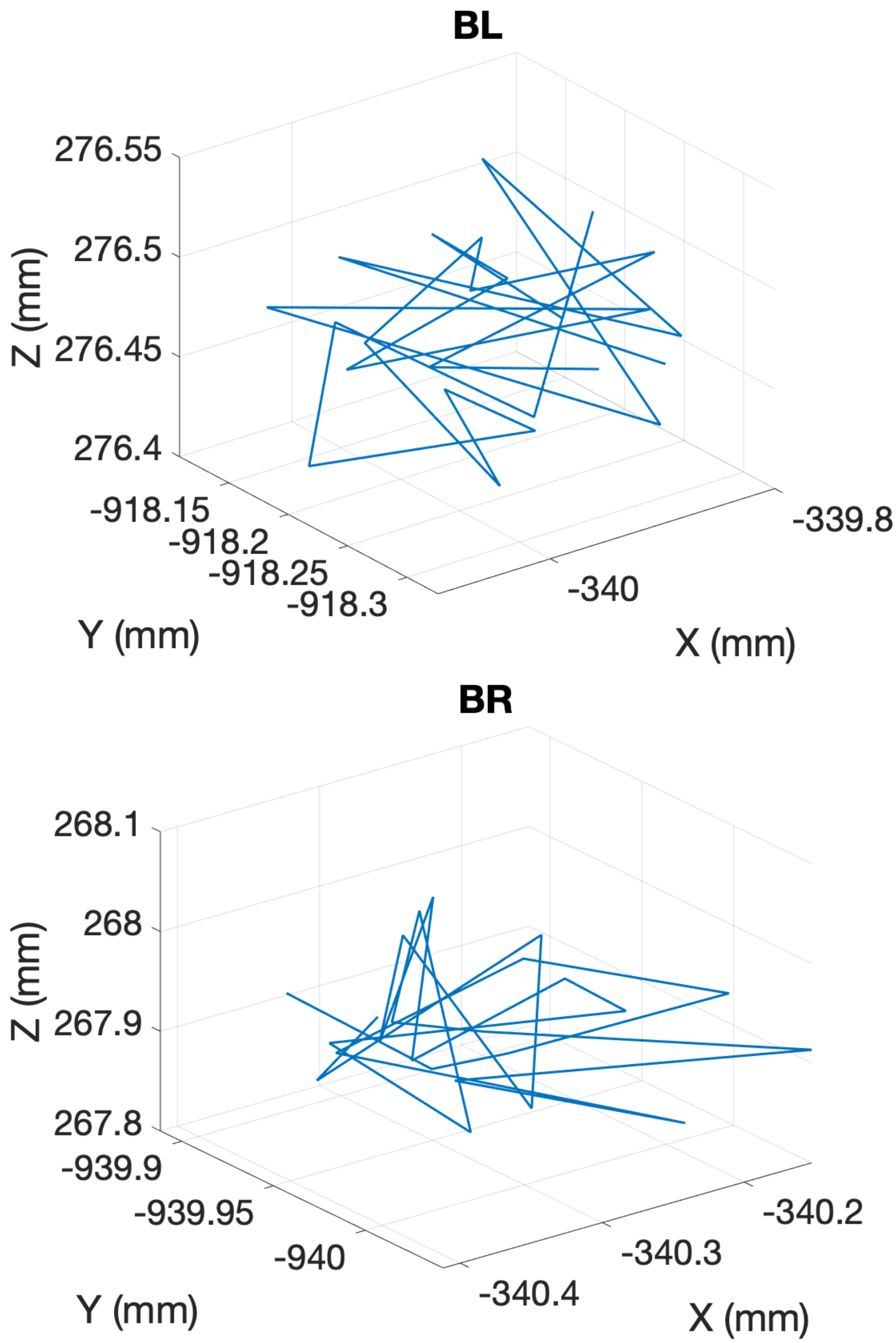


Figure 5.15: Reported position of B_L and B_R plotted sequentially over the duration of 24 hours.

CHAPTER 6: CONCLUSIONS

Over the course of this research, a low cost and portable optically projected length scale for use in photogrammetry has been successfully demonstrated. The project was divided into four distinct phases: algorithm/simulation development and prototype fabrication, prototype pattern calibration, measurement evaluation, and an exploration of factors limiting module performance and uncertainty.

6.1 Simulation, Algorithm, and Prototype

A simulation environment was established which generates an optically projected square grid pattern in the field, with known parameters such as pattern full angle, spot noise, and beam number. The simulation software allows arbitrary placement of each beam source such that any real module can be modeled. This simulation capability was created within MATLAB® and later expanded to FRED® to generate patterns on more complex surfaces. With knowledge of the pattern's pointing directions in spherical coordinates, an algorithm was created which uses passive homogeneous transform matrices to view an optical pattern from different perspectives. Using the pattern's calibration angles as a reference, the projected pattern is evaluated from different locations and poses until a match is found. This algorithm was tested to sub-nanometer resolution in a noise-free simulation environment.

Following development of the algorithm, module performance could be simulated for any desired module parameters or geometry. Two module prototype designs were created in FRED®, their properties were evaluated, and their expected performance simulated. A design was selected which utilized a single DOE and fold mirror to duplicate the pattern. The prototype module was then fabricated.

6.2 Pattern Calibration

A critical part of this research was the novel method devised to characterize the beam pattern created by the DOE. The DOE serves an important role in the module, but calibration of the beam pointing directions is difficult, especially after propagation through an optical system which imparts its own errors onto the projected pattern. Holographic gratings are not well modeled with the diffraction grating equation, and manufacturer specifications can differ from grating performance, especially at wide pattern angles. Preliminary calibration data demonstrated that even a well calibrated pattern's on-axis beams can show angular deviation on the order of hundreds of microradians.

To remedy this, a novel calibration method was created which combines photogrammetry with an algorithm to determine the module location and pose. The projected patterns were imaged at 10 different depths in the field and pattern spot coordinates determined via photogrammetry. With 10 spots for each beam forming a beam cluster, viewing perspective was shifted using a homogeneous transform matrix driven algorithm, and the spherical coordinate pointing directions for each cluster were evaluated until overlap was found for all ten points. An optimization was performed by evaluating all points in the pattern simultaneously until the algorithm solved for the translation and pose of the module away from some global coordinate system origin. Once these parameters were known, a transformation moved the coordinate set back to the global origin, clocked the pattern, and then characterized the spherical coordinate pointing directions for each beam.

To perform this calibration, a photogrammetry scale artifact was required. An artifact was fabricated from a carbon rod and aluminum target blocks, then calibrated on a CMM to 1677.66 ± 0.01 mm, yielding a fractional uncertainty of 6 parts in 10^6. Artifact stability was tested, and it performed on par or better than an average RAD target in the scene.

A calibration was performed for both patterns on the module, and a separation distance between projection origins of approximately 55.16 mm was found. A Monte Carlo simulation showed that pattern parameters such as spot noise, pattern angle, and beam count impacted the uncertainty associated with the module's calibrated length. With a pattern full angle of approximately 29°, 121 total beams, and coordinate spot noise of 110μm, 75μm, 75μm in the X,Y,Z, respectively, Monte Carlo simulation estimated calibration uncertainty of the length scale of $\pm$ 50 μm, or fractional uncertainty in the calibration on the order of 9 parts in 10^4. It was demonstrated that by merely increasing the angle to 60° and holding all other parameters constant, uncertainty in the calibrated separation length was driven down to $\approx$ 16 μm, or 3 parts in 10^4. By adding more beam points, increasing the separation length, or reducing uncertainty associated with spot location, the uncertainty in the calibrated length decreases.

6.3 Measurement Evaluation

To experimentally validate the concept of the optically projected length scale, two measurement experiments were conducted. A four camera photogrammetry network was established in the lab and convergence angles of the cameras and the targets were maximized to an average of 70°. The cameras were calibrated to correct for distortion and a computer controlled image acquisition system was established using digiCamControl. Two measurement experiments were performed, with the pattern projected at 990 mm and 1145 mm from the module.

For each experiment, 20 image sets were taken of the left and right grid patterns. The projection origin of the visible pattern was calculated for each image set and each side's 20 image set derived locations were averaged. The Euclidean distance was taken between the mean projection origin location for each side.

Each grid was imaged separately to avoid the data processing and sorting burden that arises if one images both patterns together in a one-shot measurement.

PhotoModeler® was not designed to accommodate structured light pattern derived length scales, and sorting pattern data is not trivial given the asymmetric pattern. Processing each side separately does add to the uncertainty of the measurement as it incorporates photogrammetry noise associated with the global coordinate system origin in two set of images. In the first experiment at a projection distance of 990 mm, the module length scale, L_{Photo}, was measured as 54.99 mm. In the second experiment at a distance of 1145 mm, L_{Photo} was measured as 55.57 mm. The pattern coordinates were scaled with the calibrated artifact, and each measurement should return a nominal value of approximately 55.16 mm.

A Monte Carlo simulation was used to assess the uncertainty in the prototype module's measured length scale. Inputting module parameters and reported noise on the beam spots as reported from PhotoModeler®, the expanded uncertainty (coverage factor k=2) for the solved length scale was 0.4 mm. Both of the experimental test values fell within approximately 2σ of the calibrated value, experimentally validating the optically projected length scale module concept.

Intensity of the beam spots was examined to determine if a bias in measurement had arisen due to screen position. As the pattern source was not collimated, changing the screen depth changed the intensity of each spot. In changing the projection screen depth from 990 mm to 1145 mm, data suggests that the central beam's centroid may have shifted enough to bias the measurement. While PhotoModeler® does not disclose their centroiding method, this bias could be significant and should be examined in the future. A collimated beam is recommended for consistency.

6.4 Performance Limitations

Finally, factors were examined which limit the module's performance and contribute to uncertainty in the measurement. Speckle creates uncertainty in the projected spot's perceived locations. Evaluation by Monte Carlo simulation demonstrated that the spot uncertainty scales linearly with X,Y,Z coordinate error of the solved projection location. By reducing spot uncertainty by 50%, the coordinate error drops by 50%.

This research examined the impact of the pattern's full angle on the ability of the algorithm to correctly triangulate the position of a pattern projection origin. A Monte Carlo simulation showed inverse scaling, and while the pattern can quickly become too wide for ease-of-use at large angles, by increasing the module's full angle from 29° to 60°, a 3 to 4× reduction in projection origin coordinate uncertainty is estimated.

An additional Monte Carlo simulation showed that beam number also scales inversely with solved projection pattern coordinate uncertainty. Increasing the beam number dramatically increases the processing burden on the user due to current software limitations. As will be discussed in Future Work (Chapter 7), processing limitations are, in the opinion of the author, what holds this concept back, and something that and must be addressed for it to offer a competitive advantage to traditional scale artifacts.

To further demonstrate the viability of a portable length scale projection module, the prototype was theoretically redesigned to incorporate an increased pattern full angle of 60°. By using two diffractive elements, the projection origin separation distance can be increased while maintaining nearly the same footprint as the prototype. Construction with a prism mirror and beamsplitter cube makes the module more robust. A Monte Carlo simulation demonstrated a decrease in the measured length scale error by approximately $2/3^{rds}$ relative to the prototype module, from 186 μm to 69 μm. This is a potential reduction in fractional uncertainty of the length scale by a full order of magnitude, now down to 3.4 parts in 10^4.

6.5 Final Remarks

This research has demonstrated the viability of the optically projected length scale concept. A low cost, portable prototype length scale was designed and fabricated, with a fractional uncertainty evaluated by simulation to 3.4 parts in 10^3. Experimental results showed agreement with simulation data. A set of algorithms, tools, and methods were developed which allow for evaluation of module performance, calibration of diffractive optical elements with any arbitrary beam pattern, and the ability to solve those characterized patterns for their projection location. Evaluation by Monte Carlo simulation has shown which module parameters limit performance, and where design and process improvements can be made.

It is reasonable to evaluate the system's performance against a traditional length scale. Parts in 10^3 may be adequate for some measurement situations, but the question remains whether or not an optical length scale could replace traditional RAD coded or retroreflective length scale artifacts, which offer fractional uncertainty on the order of parts in 10^4 or better [4] [5]. While the prototype module's simulated fractional uncertainty was on the order of parts in 10^3, with experimental results demonstrating agreement, simulation data for a redesigned module estimates fractional uncertainty on the order of low parts in 10^4. By increasing the beam number, quantity of photos, and pattern full angle, a functioning module with a fractional uncertainty of parts in 10^5 is possible. The system is currently being held back by post-processing constraints, as most commercial software built for photogrammetry was not designed with an optical pattern projection length scale module in mind. This topic is expanded upon further in the next chapter.

CHAPTER 7: FUTURE WORK

In this section, process improvements which increase the viability of an optically projected length scale are discussed. The practical limitations of module parameters are explored and suggestions are made for design changes in the module to facilitate reduced uncertainty. Finally, unique research areas are presented where a non-contact optically projected scale might be applied.

7.1 Process Improvements

Post-processing of optical pattern spots represents the most significant bottleneck in the entire measurement process. While acquisition of pattern images can be completed in seconds, processing through PhotoModeler® and MATLAB® can take upwards of 15 to 20 minutes per measurement. A full calibration can be imaged in less than 20 minutes, but will take approximately 12-15 hours to fully process. Quite simply put, software like PhotoModeler was not built with an optically projected measurement tool in mind, and it shows.

Sections 5.1.3 and 5.3 discussed pattern beam number and hyper-redundant photogrammetry. The data suggests a very obvious link between an increased number of beam points and a decrease in the module's uncertainty. Uncertainty scales inversely with beam number. Taking four times as many photos can yield a 50% decrease in the uncertainty of a measurement. Things like this are necessary to make the optically projected module competitive, but they cannot be done with current commercial software like PhotoModeler®. This could be a wonderful area for computer vision or computer science researchers to explore. By using pattern recognition algorithms or machine learning to intelligently look for a specific optical patterns, the marking process could be entirely automated in the same way RAD coded targets automated detection and processing in traditional photogrammetry. Given the freedom and flexibility available in DOE design, this is an area where fiducials could be incorporated into a optical pattern to aid in processing. The bottom line is that software changes need to occur for this idea to be fully realized.

A consequence of using a single DOE and fold mirror is the difference in optical path length between the two patterns, resulting in additional expansion of the mirror side pattern out in the field. Asymmetrical patterns resulted in small overlap of some beam spots in the field, which necessitated separate sets of images for each pattern. An attempt was made to process the patterns simultaneously, but an approximately five-fold increase in processing time for a single measurement discouraged further efforts. By performing the photogrammetry twice, once for each pattern, more uncertainty is introduced into the results, which stems from variation in the global coordinate system origin. Automated pattern processing, along with smart module design (symmetrical patterns), would not only reduce the post-processing burden but allow for both patterns to be imaged in the same set of photographs, thereby eliminating the extra error associated with the noise in the origin.

7.2 Module Improvements

A number of compromises were made during the fabrication of the module. In the early stages of the project, it was unclear as to how the optically projected length scale in the scene would be realized. After rejecting the idea of parallel beams, the algorithm and method changed completely, but the module did not. The module was constructed largely from off-the-shelf parts that were already available in the lab, with the exception of the pellicle beamsplitter that was purchased later. The diode source and 11×11 grid were remnants of a previous CPM project, and passable for a proof-of-concept prototype.

While the module has functioned well enough for testing the new algorithms, it is sub-optimal in a handful of ways. Incorporating a kinematic mirror mount and pellicle allowed for a handful of adjustments which ultimately are not necessary. The pellicle is fragile and potentially susceptible to vibration, while the mirror mount is unnecessary and another potential source of drift. These components add cost to the module and are not adequately robust for a typical working environment. The theoretical module proposed in Section 5.2 addresses a number of these concerns by replacing the pellicle ($225) with a corner cube beamsplitter ($160) and replacing the kinematic mounted mirror ($255) with a simple right angle prism mirror ($70).

Section 5.1 discussed a handful of factors which directly contribute to the measurement performance of the module, and have a pronounced impact on the uncertainty of the measurement. Of these, some brief suggestions are offered next in regards to spot noise, pattern full angle, beam number, and the separation distance between projection origins.

Simulation data shows that scatter associated with the projected beam spots has a linear impact upon the uncertainty of the module. Part of the goal of this research was to make an inexpensive and portable length scale projection module. The diode source used may have in fact been the least expensive component of the setup. Obviously this could be replaced with a more sophisticated source, such as a stabilized gas laser, TEC stabilized diode, a fiber source, and so on. In the case of a fiber source, Zheng's previous work in this area showed that the uncertainty associated with the beam spot could be roughly halved if the source were used in conjunction with a spinning ground glass and optics to refocus the light into a multimode optical fiber [7]. All of this adds cost and complexity to the setup, however.

In the case of the pattern full angle, the single 29° full angle DOE was used because it was immediately available. Simulation data makes it clear that uncertainty in the module's solved position scales inversely with the grid full angle. Fortunately the grating available on hand was a decent performer, but realistically it would be ideal to push the grid full angle to something along the lines of 60°. The resulting uncertainty in the solved origin drops to $1/3^{rd}$ the value while still remaining relatively usable in the field, with a pattern that doesn't expand too quickly. With only having slightly extended the separation length and altering the pattern angle in the theoretical module, 69 μm length scale uncertainty was estimated, down from 186 μm of the current prototype, and all while maintaining a near identical footprint. That module would be simple to fabricate and a good baseline for continued research in this area.

If one wishes to drive down the fractional uncertainty of the length scale, an obvious place to make gains is in the separation length between projection origins. Having use of a single DOE was actually quite restrictive. It's ideal to use as large of a pattern full angle as is manageable in the field, but to replicate that pattern from a single DOE means that either your separation length is going to suffer, or you're going to have some enormous mirrors, both of which are counterproductive. The proposed theoretical module removes this limitation just by allowing a second diffractive optical element. Without having to worry about a source expanding, a single beam could be duplicated as per the provided example. Perhaps an even more inexpensive alternative would be to create a simple rigid module housing that would hold two diode sources, each with its own DOE. This removes the cost associated with the mirror and beamsplitter, which should easily allow for an additional diffractive element to be purchased. At this point, the enclosure is the limit when it comes to separation distance between projection origins. Something like this would be significantly more simplistic than what was fabricated, and likely a better performer. Using two diodes were of different wavelengths also allows for easy filtering during post-processing to separate patterns, reducing the time and labor involved.

7.3 Other Comments & Novel Applications

During the course of this research, there was not an opportunity to test the module on any surface outside of the lab environment. The projection screen was fairly ideal, with its slightly textured surface allowing for relatively good spot visibility in each of the cameras. Limited testing was completed on a drywall surface, for which the module also performed adequately, though detection by PhotoModeler® was negatively impacted. As speckle is tied to surface feature roughness, it would be valuable to test the module on a variety of surfaces, both specular and diffuse. Scatter in the spots due to speckle was modeled in Section 5.1.1, but one of the larger issues is software spot detection. PhotoModeler® is limited in target selection, as it was not built with a structured light pattern in mind. The user is asked to specify either a white or black target, as is common in photogrammetry. It was fortunate that the optical pattern was detected using the white color option, but this is a place where purpose-built software could be advantageous.

It would also be interesting to combine this method with the visual camera model introduced by Zheng et al., whereby an artificial image of the camera pattern as observed from the projection origin is included in the photos used for the bundle adjustment [124]. This trick lets the software solve for the pose and location of the projection origin during the bundle adjustment, allowing the user to skip post-processing by algorithm in MATLAB®. One of the problems Zheng et al. experienced was that their pattern calibration was still based upon a rotary table calibration of on-axis spots, with distortion corrected theoretical estimates for other beam points. Having created a novel method by which diffractive patterns can be calibrated, this would lend itself well to the method by Zheng et al., and allow for less uncertainty in the measurement owing to the better angular description of the beam pattern.

It is also worth mentioning situations in which photogrammetry or other forms of computer vision based measurement can be applied, but in which scale of measurement is difficult to achieve. A potential area of application for a structured light length scale could be in remotely operated vehicles, or ROVs. In many of these applications, planetary rovers or underwater submersibles require terrain feature recognition and scaling for navigation and hazard avoidance.

Photogrammetry has seen use in undersea environments to monitor the three dimensional characteristics of marine habitats [125], such as coral reef growth and decay [126]. Providing a source of scale underwater can be difficult. Some divers have deployed calibration artifacts, while others trailed a measuring tape behind them as they swam while colleagues photographed features [127]. Due to motion during diving, it would be difficult to apply something like an optically projected scale.

Laser scalers are commonly used in the biological sciences [128], and are common in underwater ROV applications to provide scale at a distance. Two parallel lasers are projected into the scene and imaged. Because the beams are parallel and the separation is known, scale can be provided at a distance, albeit with varying amounts of error. Other ROV efforts experimented with a structured light pattern of 25 parallel laser lines to for a one-shot reconstruction of 3D features in underwater environments [129]. While monocular systems are popular in submersible ROVs due to design constraints in high pressure environment, stereoscopic systems are seeing increased use in specific underwater applications. Given an ROV designed with a multi-camera network, the optically projected length scale system could provide a unique way scale to underwater photogrammetry or structure from motion (SfM) based measurements.

Perhaps the most interesting application area could be in the field of planetary rovers. NASA's Jet Propulsion Laboratory (JPL) experimented with using structured laser patterns for scale and feature detection in the 1990s for use on planetary rovers, though processing limitations were imposed by the relatively slow Intel 8085 computer, which was utilized due to its radiation-hardened robustness [130,131]. This work was expanded upon by Liebe et al. at JPL, where a single diffraction grating was used to generate a 400 beam square grid. Triangulation at calibrated distances between a single camera and the projection source yields the spot location in the scene [132, 133]. Many rover systems rely on a newer type of photogrammetry called structure from motion, which is based upon Lowe's Structure Invariant Feature Transform (SIFT) algorithm, which maps image features with vectors [134]. By combining this technique with a stereoscopic camera system, measurements made by feature detection in a scene become possible [135]. Such systems work well at close range, but suffer in low ambient lighting, which is where a system based upon structured light patterns offers an advantage for low-light navigation. A laser pattern could quickly be turned on and off, conserving power in a space application, while generating an array of targets that not only could provide scale to a scene in both low and high light situations, but also theoretically serve as laser targets for which terrain topography information could be extracted.

REFERENCES

[1] R. Jiang, D. V. Jauregui, and K. R. White, "Close-range photogrammetry applications in bridge measurement: Literature review," Measurement **41**, 823–834 (2008). doi:10.1016/j.measurement.2007.12.005.

[2] M. Ozbek, D. J. Rixen, O. Erne, and G. Sanow, "Feasibility of monitoring large wind turbines using photogrammetry," Energy **35**, 4802–4811 (2010). doi:10.1016/j.energy.2010.09.008.

[3] K. Henry, "NASA Space Launch System Solid Rocket Boosters 'on Target' for First Flight," (2017). [accessed on 2020-7-6].

[4] Cultural Heritage Imaging, "Calibrated Photogrammetric Scale Bars," `http://culturalheritageimaging.org/What_We_Offer/Gear/Scale_Bars/index.html`. [accessed 2020-05-03].

[5] Brunson Corporation, "803-MCP +/- 0.003 mm Metric Unit Kit With Retroreflective Targets," `https://www.brunson.us/metric-invar-cal-kit-w-4-photo803-mcp.html`. [accessed 2020-05-03].

[6] Linearis3D, "3D Scanning Photogrammetrie|Linearis3D - Valve Housing," `http://www.linearis3d.com/valve.html`. [accessed 2020-06-28].

[7] B. Zheng, "Positioning sensor by combining optical projection and photogrammetry," Ph.D. thesis, University of North Carolina - Charlotte (2014).

[8] J. C. McGlone, E. M. Mikhail, J. Bethel, and R. Mullen, *Manual of Photogrammetry* (American Society for Photogrammetry and Remote Sensing, 2004), 5^{th} ed.

[9] F. J. Doyle, "The historical development of analytical photogrammetry," Photogrammetric Engineering **30**, 259–265 (1964).

[10] Center for Photogrammetric Training, "History of Photogrammetry," `https://ibis.geog.ubc.ca/courses/geob373/lectures/Handouts/History_of_Photogrammetry.pdf`. [accessed 2020-06-07].

[11] J. Lambert, *Freie Perspektiv* (1759).

[12] F. H. Moffit and E. M. Mikhail, *Photogrammetry* (Harper & Row, 1980), 3^{rd} ed.

[13] J. Hauck, *Konslruklion einer Dritten Perspektive einer Objekles aus Zwei Gegebenen Perspektiven* (1883).

[14] S. Finsterwalder, "Die geometrischen grundlagen der photogrammetrie," Jahresbericht der Deutschen Mathematisch Vereinigung **6** (1899).

[15] S. Finsterwalder, "Die hauptaufgabe der photogrammetrie," Sitzungsbericht der Mathematisch-Physikalischen Klasse der K. Bayerisch Akademie der Wissenschaften (1932).

[16] I. Colomina and P. Molina, "Unmanned aerial systems for photogrammetry and remote sensing: A review," ISPRS Journal of Photogrammetry and Remote Sensing **92**, 79 – 97 (2014). doi:10.1016/j.isprsjprs.2014.02.013.

[17] National Radio Astronomical Observatory, "NRAO 300-foot Telescope," `https://public.nrao.edu/telescopes/300-foot-telescope/`. [accessed on 2020-7-8].

[18] J. Ruze, "Antenna tolerance theory - a review," Proceedings of the IEEE **54**, 633–640 (1966). doi:10.1109/proc.1966.4784.

[19] J. W. Findlay, "Operating experience at the national radio astronomy observatory*," Annals of the New York Academy of Sciences **116**, 25–40 (1964). doi:10.1111/j.1749-6632.1964.tb33937.x.

[20] A. Mendikute, J. Yague-Fabra, M. Zatarain, A. Bertelsen, and I. Leizea, "Self-calibrated in-process photogrammetry for large raw part measurement and alignment before machining," Sensors **17**, 2066 (2017). doi:10.3390/s17092066.

[21] T. Luhmann, S. Robson, S. Kyle, and J. Boehm, *Close-range photogrammetry and 3D imaging* (Walter de Gruyter GmbH Co. KG, Berlin), 2^{nd} ed.

[22] A. Tommaselli and A. Berveglieri, "Measuring photogrammetric control targets in low contrast images," Boletim de Ciencias Geodesicas **24**, 171–185 (2018). doi:10.1590/S1982-21702018000200012.

[23] J. Trinder, J. Jansa, and Y. Huang, "An assessment of the precision and accuracy of methods of digital target location," ISPRS Journal of Photogrammetry and Remote Sensing **50**, 12–20 (1995). doi:10.1016/0924-2716(95)98211-H.

[24] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Transactions on Systems, Man, and Cybernetics **9**, 62–66 (1979). doi:10.1109/TSMC.1979.4310076.

[25] D. C. Brown, *Tools of the trade - close range photogrammetry and surveying - state of the art* (The American Society of Photogrammetry, 1984).

[26] G. Burgess, M. Shortis, and P. Scott, "Photographic assessment of retroreflective film properties," ISPRS Journal of Photogrammetry and Remote Sensing **66**, 743 – 750 (2011). doi:10.1016/j.isprsjprs.2011.07.002.

[27] T. Jones and R. Pappa, "Dot projection photogrammetric technique for shape measurements of aerospace test articles," in "40^{th} AIAA Applied Aerodynamics Conference," (American Institute of Aeronautics and Astronautics, 2002). doi:10.2514/6.2002-532.

[28] C. Schneider and K. Sinnreich, "Optical 3-d measurement systems for quality control in industry," International Archives of Photogrammetry and Remote Sensing **29**, 56–59 (1992).

[29] F. Heuvel, R. Kroon, and R. Le Poole, "Digital close-range photogrammetry using artificial targets," International Archives of Photogrammetry and Remote Sensing **29** (1992).

[30] C. Fraser, "Innovations in automation for vision metrology systems," Photogrammetric Record **15**, 901–911 (1997). doi:10.1111/0031-868x.00099.

[31] C. S. Fraser, "Network design considerations for non-topographic photogrammetry," Photogrammetric Engineering Remote Sensing **50**, 1115–1126 (1984).

[32] C. S. Fraser, A. Woods, and D. Brizzi, "Hyper redundancy for accuracy enhancement in automated close range photogrammetry," Photogrammetric Record **20**, 205–217 (2005). doi:10.1111/j.1477-9730.2005.00327.x.

[33] Geodetic Systems, "Picture Perfect Measurements - V-Stars System," `https://www.geodetic.com/v-stars/`. [accessed 2020-05-14].

[34] O. Kolbl, "Metric or non-metric cameras," Photogrammetric Engineering and Remote Sensing **42**, 103–113 (1976).

[35] S. Ray, *Scientific Photography and Applied Imaging* (Focal Press, 1999).

[36] C. S. Fraser, "Digital camera self-calibration," ISPRS Journal of Photogrammetry and Remote Sensing **52**, 149 – 159 (1997). doi:10.1016/S0924-2716(97)00005-1.

[37] E. Oniga and M. Diac, "Metric and non-metric cameras calibration for the improvement of real-time monitoring process results," Environmental Engineering and Management Journal **12**, 719–726 (2013). doi:10.30638/eemj.2013.088.

[38] R. S. Pappa, L. R. Geirsch, and J. M. Quagliaroli, "Photogrammetry of a 5m inflatable space antenna with consumer digital cameras," NASA Technical Report TM-2000-210627 (2000).

[39] C. S. Fraser, "Microwave antenna measurement," Photogrammetric Engineering and Remote Sensing **52**, 1627 – 1635 (1986).

[40] J. D. Littell, "Experimental photogrammetric techniques used on five full-scale aircraft crash tests," NASA Technical Report TM-2016-219168 (2016).

[41] J. Baqersad, P. Poozesh, C. Niezrecki, and P. Avitabile, "Photogrammetry and optical methods in structural dynamics – a review," Mechanical Systems and Signal Processing **86**, 17–34 (2017). doi:10.1016/j.ymssp.2016.02.011.

[42] C. Fraser, "Automated vision metrology:a mature technology for industrial inspection and engineering surveys," Australian Surveyor **46**, 5–11 (2001). doi:10.1080/00050355.2001.10558822.

[43] A. Burner, W. Lokos, and D. Barrows, "In-flight aeroelastic measurement technique development," in "Proceedings of SPIE," , vol. 5191 (2003), vol. 5191, pp. 186–199.

[44] D. Barrows, L. Olson, A. Abrego, and A. Burner, "Blade displacement measurements of the full-scale uh-60a airloads rotor," in "29th AIAA Applied Aerodynamics Conference," (AIAA, 2011). doi:10.2514/6.2011-3655.

[45] D. Ribeiro, R. Calcada, J. Ferreira, and T. Martins, "Non-contact measurement of the dynamic displacement of railway bridges using an advanced video-based system," Engineering Structures **75**, 164–180 (2014). doi:10.1016/j.engstruct.2014.04.051.

[46] C. Fraser, "Photogrammetric measurement to one part in a million," Photogrammetric Engineering and Remote Sensing **58**, 305–310 (1992).

[47] Photon Engineering, "FRED Software | Photon Engineering," (2020). [accessed 2020-06-28].

[48] B. Zheng, Y. Dong, A. Davies, B. Mullany, and E. Morse, "Using optical projection in close-range photogrammetry for 6dof sensor positioning," Photogrammetric Engineering and Remote Sensing **79**, 79–86 (2013). doi:10.14358/pers.79.1.79.

[49] J. K. Davidson and K. H. Hunt, *Robots and screw theory : applications of kinematics and statics to robotics* (Oxford University Press, 2004).

[50] J. Selig, "Active versus passive transformations in robotics," IEEE Robotics Automation Magazine **13**, 79–84 (2006). doi:10.1109/mra.2006.1598057.

[51] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," The Computer Journal **7**, 308–313 (1965). doi:10.1093/comjnl/7.4.308.

[52] E. Weise, *The encyclopedia of spectroscopy - Ed: G.L. Clark* (Reinhold Book, 1960).

[53] F. M. Grimaldi, *Physico mathesis de lumine, coloribus, et iride, aliisque annexis libri duo* (1665).

[54] R. Cecchini and G. Pelosi, "Diffraction: the first recorded observation," IEEE Antennas and Propagation Magazine **32**, 27–30 (1990). doi:10.1109/74.80496.

[55] S. J. Rigaud, *Correspondence of Scientific Men of the Seventeenth Century - Letter from James Gregory to John Collins, dated 13 May 1673*, vol. 2 (Oxford University Press, 1841).

[56] C. Huygens, *Treatise on Light* (Macmillian, 1690). Translator: Silvanus P. Thompson.

[57] H. Crew, *The wave theory of light; memoirs of Huygens, Young and Fresnel* (Cincinnati American Book Company, 1900).

[58] E. Born, Max Wolf, *Principles of Optics* (Pergamon Press Inc., 1980), 6th ed.

[59] C. Commons, "Creative Commons Attribution-ShareAlike International 4.0 License," `https://creativecommons.org/licenses/by-sa/4.0/` (June 2020).

[60] E. Hecht, *Optics* (Addison-Wesley, 2001), 4^{th} ed.

[61] T. Young, "The bakerian lecture: On the theory of light and colours," Royal Society (1802). doi:10.1098/rstl.1802.0004.

[62] D. C. O'Shea, T. J. Suleski, A. D. Kathman, and D. W. Prather, *Diffractive Optics: Design, Fabrication, and Test* (SPIE Press, 2003).

[63] Y. Soskind, *SPIE Field Guide to Diffractive Optics* (SPIE Press, 2011). doi:10.1117/3.895041.

[64] C. A. Mack, *Field Guide to Optical Lithography* (SPIE Press, Bellingham, Wash, 2006). doi:10.1117/3.665802.

[65] C. Evans, *Precision engineering : an evolutionary view* (Cranfield Press, Bedford, UK, 1989).

[66] D. Rittenhouse, *Transactions of the American philosophical society*, vol. 2 (1786).

[67] M. T. Gale and K. Knop, "The Fabrication Of Fine Lens Arrays By Laser Beam Writing," in "Industrial Applications of Laser Technology," , vol. 0398, W. F. Fagan, ed., International Society for Optics and Photonics (SPIE, 1983), vol. 0398, pp. 347 – 353. doi:10.1117/12.935397.

[68] P. Ehbets, H. P. Herzig, D. Prongué, and M. T. Gale, "High-efficiency continuous surface-relief gratings for two-dimensional array generation," Opt. Lett. **17**, 908–910 (1992). doi:10.1364/OL.17.000908.

[69] A. G. Poleshchuk, E. G. Churin, V. P. Koronkevich, V. P. Korolkov, A. A. Kharissov, V. V. Cherkashin, V. P. Kiryanov, A. V. Kiryanov, S. A. Kokarev, and A. G. Verhoglyad, "Polar coordinate laser pattern generator for fabrication of diffractive optical elements with arbitrary structure," Appl. Opt. **38**, 1295–1301 (1999). doi:10.1364/AO.38.001295.

[70] J. P. Bowen, R. L. Michaels, and C. G. Blough, "Generation of large-diameter diffractive elements with laser pattern generation," Appl. Opt. **36**, 8970–8975 (1997). doi:10.1364/AO.36.008970.

[71] Lawrence Livermore National Labs, "High-Efficiency, Ion-Beam-Etched Transmission Gratings - LLNL-WEB-411639," `https://lasers.llnl.gov/content/assets/docs/high-efficiency.pdf`. [accessed 2020-07-30].

[72] K. Prater, J. Dukwen, T. Scharf, H. P. Herzig, S. Ploger, and A. Hermerschmidt, "Micro-structuring of glassy carbon for precision glass molding of binary diffractive optical elements," Opt. Mater. Express **6**, 3407–3416 (2016). doi:10.1364/OME.6.003407.

[73] R. Gerchberg and W. Saxton, "A practical algorithm for the determination of phase from image and diffraction plane pictures," Optik **35**, 237–246 (1972).

[74] F. Wyrowski, "Diffractive optical elements: iterative calculation of quantized, blazed phase structures," J. Opt. Soc. Am. A **7**, 961–969 (1990). doi:10.1364/JOSAA.7.000961.

[75] D. C. Kim, A. Hermerschmidt, P. Dyachenko, and T. Scharf, "Adjoint method and inverse design for diffractive beam splitters," in "Components and Packaging for Laser Systems VI," , vol. 11261, A. L. Glebov and P. O. Leisher, eds., International Society for Optics and Photonics (SPIE, 2020), vol. 11261, pp. 182 – 188. doi:10.1117/12.2543367.

[76] Y. Miao, Y. Zhao, H. Ma, M. Jiang, J. Lin, and P. Jin, "Design of diffractive optical element projector for a pseudorandom dot array by an improved encoding method," Appl. Opt. **58**, G169–G176 (2019). doi:10.1364/AO.58.00G169.

[77] A. Hermerschmidt, S. Kruger, and G. Wernicke, "Binary diffractive beam splitters with arbitrary diffraction angles," Optics letters **32**, 448–450 (2007). doi:10.1364/ol.32.000448.

[78] G. N. Nguyen, "Modeling, design and fabrication of diffractive optical elements based on nanostructures operating beyond the scalar paraxial domain," Ph.D. thesis, Universite de Bretagne Occidentale (2014).

[79] J. W. Goodman, *Introduction to Fourier optics* (McGraw Hill, 1996), 2nd ed.

[80] H. Pang, S. Yin, Q. Deng, Q. Qiu, and C. Du, "A novel method for the design of diffractive optical elements based on the rayleighâsommerfeld integral," Optics and Lasers in Engineering **70**, 38–44 (2015). doi:10.1016/j.optlaseng.2015.02.007.

[81] H. Ichikawa and T. Toda, "Improving accuracy in period measurement of gratings with multiple diffraction orders," Appl. Opt. **53**, 5290–5293 (2014). doi:10.1364/AO.53.005290.

[82] D. A. Brasil, J. A. P. Alves, and J. R. Pekelsky, "An imaging grating diffractometer for traceable calibration of grating pitch in the range 20 m to 350 nm," Journal of Physics: Conference Series **648**, 012–013 (2015). doi:10.1088/1742-6596/648/1/012013.

[83] W. Y. Song, K. Y. Jung, O. Beom-hoan, and B. C. Park, "Precision laser diffractometry for grating period measurements," Journal of Korean Physical Society **45** (2004).

[84] D. Flack, "Good Practice Guide 42 - CMM Validation," `https://www.npl.co.uk/gpgs/cmm-verification` (2011). [accessed 2020-06-19].

[85] ISO 13360-2:2009(en), "Geometrical product specifications (GPS) - Acceptance and reverification tests for coordinate measuring machines (CMM) - Part 2: CMMs used for measuring linear dimensions," (2009).

[86] JCGM 100:2008, "Evaluation of measurement data â Guide to the expression of uncertainty in measurement," `https://www.bipm.org/utils/common/documents/jcgm/JCGM_100_2008_E.pdf` (2008). [accessed 2020-01-01].

[87] JCGM 101:2008, "Evaluation of measurement data â Supplement 1 to the âGuide to the expression of uncertainty in measurementâ â Propagation of distributions using a Monte Carlo method ," `https://www.bipm.org/utils/common/documents/jcgm/JCGM_101_2008_E.pdf` (2008). [accessed 2020-01-01].

[88] Mathworks, "What is Camera Calibration?" `https://www.mathworks.com/help/vision/ug/camera-calibration.html`. [accessed 2020-05-14].

[89] T. A. Clarke and J. G. Fryer, "The development of camera calibration methods and models," Photogrammetric Record **16**, 51–66 (1998). doi:10.1111/0031-868x.00113.

[90] A. Conrady, "Decentered lens-systems," Monthly Notices of the Royal Astronomical Society **79**, 384–390 (1919).

[91] D. C. Brown, "Decentering distortion of lenses," Photogrammetric Engineering **32**, 444–462 (1966).

[92] D. C. Brown, "Close-range camera calibration," Photogrammetric Engineering **37**, 855–866 (1971).

[93] D. C. Brown and J. Fryer, "Lens distortion for close-range photogrammetry," Photogrammetric Engineering and Remote Sensing **52**, 51–58 (1986).

[94] R. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," IEEE Journal on Robotics and Automation **3**, 323–344 (1987). doi:10.1109/jra.1987.1087109.

[95] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in "Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition," (IEEE, 1997), pp. 1106–1112. doi:10.1109/cvpr.1997.609468.

[96] Z. Zhang, "A flexible new technique for camera calibration," IEEE Transactions on Pattern Analysis and Machine Intelligence **22**, 1330–1334 (2000). doi:10.1109/34.888718.

[97] J.-Y. Bouguet, "Camera Calibration Toolbox for Matlab," `http://www.vision.caltech.edu/bouguetj/calib_doc/`. [accessed 2020-05-15].

[98] S. Maybank and O. Faugeras, "A theory of self-calibration of a moving camera," International Journal of Computer Vision **8**, 123–151 (1992). doi:10.1007/bf00127171.

[99] O. D. Faugeras, Q.-T. Luong, and S. J. Maybank, "Camera self-calibration: Theory and experiments," in "Proceedings of the Second European Conference on Computer Vision," (Springer-Verlag, Berlin, Heidelberg, 1992), ECCV, pp. 321–334.

[100] PhotoModeler Technologies, "PhotoModeler® 6 User Manual," [Electronic Manual; 2009].

[101] PhotoModeler Technologies, "Photogrammetry, 3D Measurements From Photos, PhotoModeler," `https://www.photomodeler.com/`. [accessed 24-April-2020].

[102] AgiSoft LLC, "AgiSoft Metashape," `https://www.agisoft.com`. [accessed 2020-05-14].

[103] M. Saleh, Bahaa E. A. Teich, *Fundamentals of Photonics* (Wiley, New York, 2007).

[104] Canon U.S.A. Inc, "EOS Rebel T3i Body | Canon USA," `{https://www.usa.canon.com/internet/portal/us/home/support/details/cameras/support-dslr/eos-rebel-t3i-body}`. [accessed 2020-05-17].

[105] digiCamControl, "Easy, free to use solution for complex camera control | digiCamControl," `http://www.digicamcontrol.com/`. [accessed 2020-04-24].

[106] J. C. Dainty, *Laser Speckle and Related Phenomena*, Topics in applied physics (Springer-Verlag, 1975).

[107] J. W. Goodman, "Some fundamental properties of speckle," Journal of the Optical Society of America **66** (1976). doi:10.1364/josa.66.001145.

[108] T. A. Clarke, "Analysis of the properties of targets used in digital close-range photogrammetric measurement," Proceedings of SPIE **2350**, 251–262 (1994).

[109] T. Clarke and A. Katsimbris, "The use of diode laser collimators for targeting 3-d objects," International Archives of Photogrammetry and Remote Sensing **30**, 47–54 (1994).

[110] R. G. Dorsch, G. Hausler, and J. M. Herrmann, "Laser triangulation: fundamental uncertainty in distance measurement," Applied Optics **33**, 1306–1314 (1994). doi:10.1364/ao.33.001306.

[111] G. W. Ellis, "A fiber-optic phase-randomizer for microscope illumination by laser," Journal of Cell Biology **83** (1979).

[112] T. Asakura, "Spatial coherence of laser light passed through rotating ground glass," Opto-Electronics **2**, 115–123 (1970). doi:10.1007/bf01415089.

[113] L. E. Estes, L. M. Narducci, and R. A. Tuft, "Scattering of light from a rotating ground glass," J. Opt. Soc. Am. **61**, 1301–1306 (1971). doi:10.1364/JOSA.61.001301.

[114] C. Davenport and A. Gmitro, "Angioscopic fluorescence imaging system," in "International Society for Optical Engineering," , vol. 1649 (1992), vol. 1649, pp. 192–202.

[115] T. Stangner, Z. Hanqing, D. Tobias, W. Krister, and M. Andersson, "Step-by-step guide to reduce spatial coherence of laser light using a rotating ground glass diffuser," Applied Optics **56**, 5427–5435 (2017). doi:10.1364/ao.56.005427.

[116] D. C. Brown, *Application of close-range photogrammetry to measurements of structures in orbit* (Geodetic Services Incorporated, 1980).

[117] P. Seitz, "Optical superresolution using solid-state cameras and digita; signal processing," Optical Engineering **27**, 277535 (1988). doi:10.1117/12.7976719.

[118] A. Fox, *Optical Properties of Solids*, Oxford master series in condensed matter physics (Oxford University Press, 2001).

[119] R. Paschotta, "Rp-photonics encyclopedia - laser diode," `https://www.rp-photonics.com/laser_diodes.html`. [accessed on 2019-04-22].

[120] K. D. Mielenz, K. F. Nefflen, W. R. C. Rowley, D. C. Wilson, and E. Engelhard, "Reproducibility of helium–neon laser wavelengths at 633 nm," Appl. Opt. **7**, 289–293 (1968). doi:10.1364/AO.7.000289.

[121] Newport Labs, "Stability of Optical Posts vs. Pedestals," `https://www.newport.com/n/stability-of-optical-posts-vs-pedestals`. [accessed on 2019-4-19].

[122] Newport Labs, "Suprema ZeroDrift Thermally Compensated Mirror Mounts," `https://www.newport.com/f/zero-drift-mirror-mounts`. [accessed on 2019-4-19].

[123] R. Nave, "Circular membrane modes," `http://hyperphysics.phy-astr.gsu.edu/hbase/Music/cirmem.html`. [accessed 2019-3-07].

[124] B. Zheng, Y. Dong, B. Mullany, E. Morse, and A. Davies, "Positioning sensor by combining photogrammetry, optical projection and a virtual camera model," Measurement Science and Technology **24**, 8 (2013). doi:10.1088/0957-0233/24/10/105106.

[125] G. Marre, F. Holon, S. Luque, P. Boissery, and J. Deter, "Monitoring marine habitats with photogrammetry: A cost-effective, accurate, precise and high-resolution reconstruction method," Frontiers in Marine Science (2019). doi:10.3389/fmars.2019.00276.

[126] R. Ferrari, W. F. Figueira, M. S. Pratchett, T. Boube, A. Adam, T. Kobelkowsky-Vidrio, S. S. Doo, T. B. Atwood, and M. Byrne, "3d photogrammetry quantifies growth and external erosion of individual coral colonies and skeletons," Scientific reports **7**, 1–9 (2017). doi:10.1038/s41598-017-16408-z.

[127] P. Piazza, V. Cummings, A. Guzzi, I. Hawes, A. Lohrer, S. Marini, P. Marriott, F. Menna, E. Nocerino, A. Peirano, S. Kim, and S. Schiaparelli, "Underwater photogrammetry in antarctica: long-term observations in benthic ecosystems and legacy data rescue," Polar Biology **42**, 1061–1079 (2019). doi:10.1007/s00300-019-02480-w.

[128] N. L. Barrickman, A. L. Schreier, and K. E. Glander, "Testing parallel laser image scaling for remotely measuring body dimensions on mantled howling monkeys," American Journal of Primatology **77**, 823–832 (2015). doi:10.1002/ajp.22416.

[129] M. Massot-Campos and G. Oliver-Codina, "Underwater laser-based structured light system for one-shot 3d reconstruction," in "IEEE SENSORS 2014 Proceedings," (IEEE, 2014), December, pp. 1138–1141. doi:10.1109/icsens.2014.6985208.

[130] L. Matthies, E. Gat, R. Harrison, B. Wilcox, R. Volpe, and T. Litwin, "Mars microrover navigation: performance evaluation and enhancement," in "Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots," , vol. 1 (IEEE, 1995), vol. 1, pp. 433–440. doi:10.1109/iros.1995.525832.

[131] L. Matthies, T. Balch, and B. Wilcox, "Fast optical hazard detection for planetary rovers using multiple spot laser triangulation," in "Proceedings of International Conference on Robotics and Automation," , vol. 1 (IEEE, 1997), vol. 1, pp. 859–866. doi:10.1109/robot.1997.620142.

[132] C. Liebe, C. Padgett, and J. Chang, "Three dimensional imaging utilizing structured light," in "2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)," , vol. 4 (IEEE, 2004), vol. 4, pp. 2647–2655. doi:10.1109/aero.2004.1368060.

[133] C. Liebe, C. Padgett, J. Chapsky, D. Wilson, K. Brown, S. Jerebets, H. Goldberg, and J. Schroeder, "Spacecraft hazard avoidance utilizing structured light," in "2006 IEEE Aerospace Conference," (IEEE, 2006), p. 10 pp. doi:10.1109/aero.2006.1655898.

[134] D. G. Lowe, "Object recognition from local scale-invariant features," in "Proceedings of the International Conference on Computer Vision," , vol. 2 of *ICCV 1999* (IEEE Computer Society, USA, 1999), vol. 2 of *ICCV 1999*, p. 1150.

[135] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," The International Journal of Robotics Research **21**, 735–758 (2002). doi:10.1177/027836402761412467.

APPENDIX A: MATLAB® CODE

A.1 Length Scale Solver

A.1.1 LSolver.m

```
% L Solver

% This program utilizes calibration data to process
   photogrammetry delivered
% coordinates (pre-sorted), and solves each grid for its
   position, then
% finds the unscaled Euclidean distance between the
   projection points, "LPhoto".
%
% L is then compared to the calibrated L, and we have the
   length of the
% scale object.  Scale = LCal / LPhoto

clear;
clc;

% Set the beam number, n
n=11; % n is the (odd) numbner of beams along an axis.  11
    x 11 Grid

% Load photogrammetry data
load ('dMWP.mat'); % loads photogrammetry datasets for
   mirror and pellicle
load ('dPWP.mat'); % 121x3 matrix (X,Y,Z) format

% Input starting guess location for algorithm
startPellicle=[-1420 -920 275 0 0 0]; % Initial guess - [x
    y z a b g]
startMirror=[-1460 -960 275 0 0 0]; % 6 DoF, translations
   and pose

% Pass variables to minimization algorithm and solve
[Lmean]=solver_left(n,dPWP,startPellicle);
[Rmean]=solver_right(n,dMWP,startMirror);

% Clean up and process the data...
combined = [Lmean;Rmean]; %fit into one matrix
combineddist = combined(1:2,1:3);
```

```
% Use pdist to find Euclidean distance.  (rows =
   observations, columns =
% variables... ie: row 1 -> X1 Y1 Z1, row 2 -> X2 Y2 Z2
LSolved = pdist(combineddist);

%Output the solved LPhoto length scale in meters
LSolved_m = LSolved/1000
```

A.1.2 solver_left.m

```
function [meanvalueleft]=solver_left(n,dataset,start)

% Load the grid calibration data
load('calibrationsphWP.mat');

% Pass variables to location solving algorithm.  Minimize
   the merit
% function.  Output is the location of the left grid
OPTIONS = optimset('Display','iter','TolFun',1e-10,'TolX'
   ,1e-10,'MaxIter',6000,'MaxFunEvals',6000);
[meanvalueleft]= fminsearch(@(x)Algorithm_LSolver(n,
   dataset,pelliclecalspherical,x),start,OPTIONS);

end
```

A.1.3 solver_right.m

```
function [meanvalueright]=solver_right(n,dataset,start)

% Load the grid calibration data
load('calibrationsphWP.mat');

% Pass variables to location solving algorithm.  Minimize
   the merit
% function.  Output is the location of the left grid
OPTIONS = optimset('Display','iter','TolFun',1e-10,'TolX'
   ,1e-10,'MaxIter',6000,'MaxFunEvals',6000);
[meanvalueright]= fminsearch(@(x)Algorithm_LSolver(n,
   dataset,mirrorcalspherical,x),start,OPTIONS);
end
```

A.1.4 BulkLSolver.m

```
% Bulk L Solver - Patrick Thewlis - 5/24/2020
% Loads up an entire directory of measurement files, and
   solves them all
% for locations

clear;
clc;
n=11; % n is the (odd) numbner of beams along an axis.  11
    x 11 Grid

for i = 1:20
% Load up calibration and photogrammetry data
% Pulls dMWP(#).mat files from active folder
filenameM = ['dMWP' sprintf('%1.f',i) '.mat'];
filenameP = ['dPWP' sprintf('%1.f',i) '.mat'] ;

load (filenameM); % loads photogrammetry datasets for
   mirror and pellicle
load (filenameP); % 121x3 matrix (X,Y,Z) format

% Input guesses for algorithm
startPellicle=[-1420 -920 275 0 0 0]; % Initial guess - [x
    y z a b g]
startMirror=[-1460 -960 275 0 0 0]; % 6 DoF, translations
   and pose

%Pass data to the minimization functions
[Lmean(i,:)]=solver_left(n,dPWP,startPellicle);
[Rmean(i,:)]=solver_right(n,dMWP,startMirror);

% Clean up and process the data...
combined = [Lmean(i,:);Rmean(i,:)]; %fit into one matrix
combineddist = combined(1:2,1:3);

%use pdist to find Euclidean distance...  (rows =
   observations, columns =
%variables... ie: row 1 -> X1 Y1 Z1, row 2 -> X2 Y2 Z2
LSolved(i,:) = pdist(combineddist);

%Output the solved LPhoto length scale in meters
LSolved_m(i,:) = LSolved(i,:)/1000

end
```

A.2 Monte Carlo Length Scale Solver Simulation

A.2.1 MonteCarloLSim.m

```
% Monte Carlo L Solver - Patrick Thewlis - 2/16/2020

% Monte Carlo LSolver simulation.  Creates a data set,
   adds noise to the X,Y,Z coordinates, and
% repeatedly finds L.  Sampling a large number of Ls, we
   can explore
% uncertainty in the system.

tic
clear;
clc;

%Input desired noise, iterations, initial guess point
sx = 0.000110; % spot noise X
sy = 0.000073; % spot noise Y
sz = 0.000073; % spot noise Z

%Input starting guesses
separationdistance = 3.0; %manually input the separation
   distance b/t projection points for calculating fracUc
guessleft=[0.0001 0.0001 0.0001 0 0 0]; %initial guess - [
   x y z a b g] -   6 DOF, translation and orientation
guessright=[0.0001 -3.01 0.0001 0 0 0]; %ORIG (-0.039
   -0.039 0 0 0 0)
iter=10; % iteration count 1 for external loop
n=11; % n=input('input n (# of horizontal beams):') %
   horizontal dot number
dL=1; %left side projection distance d=input('input
   distance from module to projection surface:')
dR=1; % (ORIG 1.039) right side projection distance to
   wall/surface
theta0A=60/(n-1); % theta0=input('input initial interbeam
   angle (fullangle / n-1 beams):') BEAM A
theta0B=60/(n-1); % theta0=input('input initial interbeam
   angle (fullangle / n-1 beams):') BEAM B
oriL =[0 0 0]; % oriL=input('input the origin (recommended
    [0 0 0]):') %Format [X Y Z]
htmX = -(dL-dR);     % X offset = input('input the offset
   between modules') (SIGNS ARE FLIPPED)
htmY = 3.00; % (ORIG 0.039 Y offset = input('input the
   offset between modules') (SIGNS ARE FLIPPED)
htmZ = 0;     % Z offset = input('input the offset between
```

```
    modules') (SIGNS ARE FLIPPED)
htmA = 0;     % HTM X axis rotation (SIGNS ARE FLIPPED)
htmB = 0;     % HTM Y axis rotation (SIGNS ARE FLIPPED)
htmC = 0;     % HTM Z axis rotation (SIGNS ARE FLIPPED)

[aL,mL,uvcalL,sphcalL] =scatterleft(n,dL,theta0A,oriL);
[aR,mR,uvcalR,sphcalR] = scatterright(n,dR,theta0B,oriL);

% Prep left side calibration data
%==================================================
calbaseL(:,2) = mL(:,1); % x->y
calbaseL(:,3) = mL(:,2); % y->z
calbaseL(:,1) = mL(:,3); % z->x

for i = 1:n*n
    [azL(i,:) elL(i,:) rL(i,:)] = cart2sph(calbaseL(i,1),
       calbaseL(i,2), calbaseL(i,3));
end

sphdataL(1,:) = azL;
sphdataL(2,:) = elL;

% Prep right side calibration data
%==================================================

calbaseR(:,2) = mR(:,1); % x->y
calbaseR(:,3) = mR(:,2); % y->z
calbaseR(:,1) = mR(:,3); % z->x

for i = 1:n*n
    [azR(i,:) elR(i,:) rR(i,:)] = cart2sph(calbaseR(i,1),
       calbaseR(i,2), calbaseR(i,3));
end

sphdataR(1,:) = azR;
sphdataR(2,:) = elR;

% Pre left and right grid data
%==================================================

%left
leftdata(:,2) = mL(:,1); % x->y
leftdata(:,3) = mL(:,2); % y->z
leftdata(:,1) = mL(:,3); % z->x
```

```
%right
%mR(:,1) = mR(:,1) + offset;

rightdataprep(:,2) = mR(:,1); % x->y
rightdataprep(:,3) = mR(:,2); % y->z
rightdataprep(:,1) = mR(:,3); % z->x

% Create HGT based upon the initial guesses
hgt = makehgtform('translate',[htmX htmY htmZ],'xrotate',
   htmA,'yrotate',htmB,'zrotate',htmC); % 6 DOF

rightdataprep=rightdataprep';
rightdataprep(4,:)=1;

rightdataoffset=inv(hgt)*rightdataprep;
rightdata = rightdataoffset(1:3,:);
rightdata = rightdata';

save('sphcalL.mat','sphdataL');
save('sphcalR.mat','sphdataR');
save('Left.mat','leftdata');
save('Right.mat','rightdata');

% Begin External Monte Carlo Loop

% Preallocate LSim array to speed up computation
LSim = zeros(iter,1);

for q = 1:iter

% Send data to have noise added, then algorithm
   minimization
[Lmean]=noiseleftminimization(n,sx,sy,sz,leftdata,
   guessleft);
[Rmean]=noiserightminimization(n,sx,sy,sz,rightdata,
   guessright);

% Store the solved projection origin locations after each
   iteration
xl(q,:)=Lmean;
xr(q,:)=Rmean;

%fit into one matrix
combined = [Lmean;Rmean];
```

```
%use pdist to find Euclidean distance...  (rows =
   observations, columns =
%variables... ie: row 1 -> X1 Y1 Z1, row 2 -> X2 Y2 Z2

LSim(q,1) = pdist(combined);

disp(q)
end

% Create a histogram of the solved location data
figure
hist(LSim,sqrt(iter))

% Save simulation data
save('LSim.mat','LSim','xr','xl')

% Print standard deviation for all MC simulated length
   scales
std(LSim)

% Log data to text files
dlmwrite('xleft.txt',xl, '-append', 'delimiter','\t','
   precision',20);
dlmwrite('xright.txt',xr, '-append', 'delimiter','\t','
   precision',20);

toc

% Compute fractional uncertainty in length scale
fracUc = std(LSim)/separationdistance
```

A.3 Monte Carlo Module Parameter Simulator

A.3.1 MonteCarloBeamNum.m

```
% Monte Carlo Beam Number Solver - Patrick Thewlis -
   7/31/2020

% Monte Carlo style simulation.  Creates a data set, and
   adds noise, and
% repeatedly finds L.  Sampling a large number of Ls, we
   can explore
% uncertainty in the system.

% This is the main function for Monte Carlo 2D simulation.
tic
clear;
clc;
for n=5:2:35
%Input desired noise, iterations, initial guess point
sx = 0.0000110; % spot noise X
sy = 0.000073; % spot noise Y
sz = 0.000073; % spot noise Z
variable = n;
%Input starting guesses
guessleft=[0 0 0 0 0 0]; %initial guess - [x y z a b g] -
     6 DOF, translation and orientation
guessright=[-0.039 -0.039 0 0 0 0];
iter=500; % iteration count 1 for external loop
%n=11; % n=input('input n (# of horizontal beams):') %
   horizontal dot number
dL=1; %left side projectioj distance d=input('input
   distance from module to projection surface:')
dR=1.039; % right side projection distance to wall/surface
theta0A=30/(n-1); % theta0=input('input initial interbeam
   angle (fullangle / n-1 beams):') BEAM A
theta0B=30/(n-1); % theta0=input('input initial interbeam
   angle (fullangle / n-1 beams):') BEAM B
oriL=[0 0 0]; % oriL=input('input the origin (recommended
   [0 0 0]):') %Format [X Y Z]
htmX = -(dL-dR);     % X offset = input('input the offset
   between modules') (SIGNS ARE FLIPPED)
htmY = 0.039; % Y offset = input('input the offset between
    modules') (SIGNS ARE FLIPPED)
htmZ = 0;      % Z offset = input('input the offset between
    modules') (SIGNS ARE FLIPPED)
htmA = 0;      % HTM X axis rotation (SIGNS ARE FLIPPED)
```

```
htmB = 0;      % HTM Y axis rotation (SIGNS ARE FLIPPED)
htmC = 0;      % HTM Z axis rotation (SIGNS ARE FLIPPED)

[aL,mL,uvcalL,sphcalL] = scatterleft(n,dL,theta0A,oriL);
[aR,mR,uvcalR,sphcalR] = scatterright(n,dR,theta0B,oriL);

% Prep left side calibration data
%==================================================

calbaseL(:,2) = mL(:,1); % x->y
calbaseL(:,3) = mL(:,2); % y->z
calbaseL(:,1) = mL(:,3); % z->x

for i = 1:n*n
    [azL(i,:) elL(i,:) rL(i,:)] = cart2sph(calbaseL(i,1),
       calbaseL(i,2), calbaseL(i,3));
end

sphdataL(1,:) = azL;
sphdataL(2,:) = elL;

% Prep right side calibration data
%==================================================

calbaseR(:,2) = mR(:,1); % x->y
calbaseR(:,3) = mR(:,2); % y->z
calbaseR(:,1) = mR(:,3); % z->x

for i = 1:n*n
    [azR(i,:) elR(i,:) rR(i,:)] = cart2sph(calbaseR(i,1),
       calbaseR(i,2), calbaseR(i,3));
end

sphdataR(1,:) = azR;
sphdataR(2,:) = elR;

% Pre left and right grid data
%==================================================

%left
leftdata(:,2) = mL(:,1); % x->y
leftdata(:,3) = mL(:,2); % y->z
leftdata(:,1) = mL(:,3); % z->x

%right
```

```
rightdataprep(:,2) = mR(:,1); % x->y
rightdataprep(:,3) = mR(:,2); % y->z
rightdataprep(:,1) = mR(:,3); % z->x

hgt = makehgtform('translate',[htmX htmY htmZ],'xrotate',
   htmA,'yrotate',htmB,'zrotate',htmC); % 6 DOF

rightdataprep=rightdataprep';
rightdataprep(4,:)=1;

rightdataoffset=inv(hgt)*rightdataprep;
rightdata = rightdataoffset(1:3,:);
rightdata = rightdata';

save('sphcalL.mat','sphdataL');
save('sphcalR.mat','sphdataR');
save('Left.mat','leftdata');
save('Right.mat','rightdata');

% Begin External Monte Carlo Loop
% Preallocate LSim array to speed up computation
LSim = zeros(iter,1);

for q = 1:iter

%Do the minimization
[Lmean]=noiseleftminimization(n,sx,sy,sz,leftdata,
   guessleft);
[Rmean]=noiserightminimization(n,sx,sy,sz,rightdata,
   guessright);

xl(q,:)=Lmean;
xr(q,:)=Rmean;

%fit into one matrix
combined = [Lmean;Rmean];

%use pdist to find euclidean distance...  (rows =
   observations, columns =
%variables... ie: row 1 -> X1 Y1 Z1, row 2 -> X2 Y2 Z2

LSim(q,1) = pdist(combined);

disp(q)
```

```
end

% Save and process data

stdleft = std(xl);
stdwriteleft = [variable stdleft];
dlmwrite('stdleft.txt',stdwriteleft, '-append', 'delimiter
   ','\t','precision',20);


stdright = std(xr);
stdwriteright = [variable stdright];
dlmwrite('stdright.txt',stdwriteright, '-append', '
   delimiter','\t','precision',20);


meanvalueleft=mean(xl);
meanvaluewriteleft = [variable meanvalueleft];
dlmwrite('meanvalueleft.txt',meanvaluewriteleft, '-append'
   , 'delimiter','\t','precision',20);


rangeleft=max(xl)-min(xl);
rangewriteleft = [variable rangeleft];
dlmwrite('rangeleft.txt',rangewriteleft, '-append', '
   delimiter','\t','precision',20);


meanvalueright=mean(xr);
meanvaluewriteright = [variable meanvalueright];
dlmwrite('meanvalueright.txt',meanvaluewriteright, '-
   append', 'delimiter','\t','precision',20);


rangeright=max(xr)-min(xr);
rangewriteright = [variable rangeright];
dlmwrite('rangeright.txt',rangewriteright, '-append', '
   delimiter','\t','precision',20);


LSimstd = std(LSim);
stdwriteLSim = [variable LSimstd];
dlmwrite('stdLSim.txt',stdwriteLSim, '-append', 'delimiter
   ','\t','precision',20);
```

```
LSimmean = mean(LSim);
meanwriteLSim = [variable LSimmean];
dlmwrite('meanLSim.txt',meanwriteLSim, '-append', '
   delimiter','\t','precision',20);


% ==========
clear %Resets all variables before next loop

end
```

A.3.2 MonteCarloSpotNoise.m

```
% Monte Carlo Spot Noise Solver - Patrick Thewlis -
   7/31/2020

% Monte Carlo style simulation.  Creates a data set, and
   adds noise, and
% repeatedly finds L.  Sampling a large number of Ls, we
   can explore
% uncertainty in the system.

% This is the main function for Monte Carlo 2D simulation.
tic
clear;
clc;
for s = 0.000010:0.000010:0.000300
%Input desired noise, iterations, initial guess point
% sx = 0.0000110; % spot noise X
% sy = 0.000065; % spot noise Y
% sz = 0.000065; % spot noise Z
sx = s;
sy = s;
sz = s;
variable = s;
%Input starting guesses
guessleft=[0 0 0 0 0 0]; %initial guess - [x y z a b g] -
     6 DOF, translation and orientation
guessright=[-0.039 -0.039 0 0 0 0];
iter=100; % iteration count 1 for external loop
n=11; % n=input('input n (# of horizontal beams):') %
   horizontal dot number
dL=1; %left side projectioj distance d=input('input
   distance from module to projection surface:')
dR=1.039; % right side projection distance to wall/surface
theta0A=30/(n-1); % theta0=input('input initial interbeam
   angle (fullangle / n-1 beams):') BEAM A
theta0B=30/(n-1); % theta0=input('input initial interbeam
   angle (fullangle / n-1 beams):') BEAM B
oriL=[0 0 0]; % oriL=input('input the origin (recommended
   [0 0 0]):') %Format [X Y Z]
htmX = -(dL-dR);     % X offset = input('input the offset
   between modules') (SIGNS ARE FLIPPED)
htmY = 0.039; % Y offset = input('input the offset between
    modules') (SIGNS ARE FLIPPED)
htmZ = 0;     % Z offset = input('input the offset between
```

```
    modules') (SIGNS ARE FLIPPED)
htmA = 0;      % HTM X axis rotation (SIGNS ARE FLIPPED)
htmB = 0;      % HTM Y axis rotation (SIGNS ARE FLIPPED)
htmC = 0;      % HTM Z axis rotation (SIGNS ARE FLIPPED)

[aL,mL,uvcalL,sphcalL] = scatterleft(n,dL,theta0A,oriL);
[aR,mR,uvcalR,sphcalR] = scatterright(n,dR,theta0B,oriL);

% Prep left side calibration data
%==================================================

calbaseL(:,2) = mL(:,1); % x->y
calbaseL(:,3) = mL(:,2); % y->z
calbaseL(:,1) = mL(:,3); % z->x


for i = 1:n*n
    [azL(i,:) elL(i,:) rL(i,:)] = cart2sph(calbaseL(i,1),
       calbaseL(i,2), calbaseL(i,3));
end

sphdataL(1,:) = azL;
sphdataL(2,:) = elL;

% Prep right side calibration data
%==================================================

calbaseR(:,2) = mR(:,1); % x->y
calbaseR(:,3) = mR(:,2); % y->z
calbaseR(:,1) = mR(:,3); % z->x

for i = 1:n*n
    [azR(i,:) elR(i,:) rR(i,:)] = cart2sph(calbaseR(i,1),
       calbaseR(i,2), calbaseR(i,3));
end

sphdataR(1,:) = azR;
sphdataR(2,:) = elR;

% Pre left and right grid data
%==================================================

%left
leftdata(:,2) = mL(:,1); % x->y
leftdata(:,3) = mL(:,2); % y->z
```

```
leftdata(:,1) = mL(:,3); % z->x

%right

rightdataprep(:,2) = mR(:,1); % x->y
rightdataprep(:,3) = mR(:,2); % y->z
rightdataprep(:,1) = mR(:,3); % z->x

hgt = makehgtform('translate',[htmX htmY htmZ],'xrotate',
   htmA,'yrotate',htmB,'zrotate',htmC); % 6 DOF

rightdataprep=rightdataprep';
rightdataprep(4,:)=1;

rightdataoffset=inv(hgt)*rightdataprep;
rightdata = rightdataoffset(1:3,:);
rightdata = rightdata';

save('sphcalL.mat','sphdataL');
save('sphcalR.mat','sphdataR');
save('Left.mat','leftdata');
save('Right.mat','rightdata');

% Begin External Monte Carlo Loop
% Preallocate LSim array to speed up computation
LSim = zeros(iter,1);

for q = 1:iter

%Do the minimization
[Lmean]=noiseleftminimization(n,sx,sy,sz,leftdata,
   guessleft);
[Rmean]=noiserightminimization(n,sx,sy,sz,rightdata,
   guessright);

xl(q,:)=Lmean;
xr(q,:)=Rmean;

%fit into one matrix
combined = [Lmean;Rmean];

%use pdist to find euclidean distance...  (rows =
   observations, columns =
%variables... ie: row 1 -> X1 Y1 Z1, row 2 -> X2 Y2 Z2
```

```
LSim(q,1) = pdist(combined);

disp(q)
end

%save and process data

stdleft = std(xl);
stdwriteleft = [variable stdleft];
dlmwrite('stdleft.txt',stdwriteleft, '-append', 'delimiter
   ','\t','precision',20);


stdright = std(xr);
stdwriteright = [variable stdright];
dlmwrite('stdright.txt',stdwriteright, '-append', '
   delimiter','\t','precision',20);


meanvalueleft=mean(xl);
meanvaluewriteleft = [variable meanvalueleft];
dlmwrite('meanvalueleft.txt',meanvaluewriteleft, '-append'
   , 'delimiter','\t','precision',20);


rangeleft=max(xl)-min(xl);
rangewriteleft = [variable rangeleft];
dlmwrite('rangeleft.txt',rangewriteleft, '-append', '
   delimiter','\t','precision',20);


meanvalueright=mean(xr);
meanvaluewriteright = [variable meanvalueright];
dlmwrite('meanvalueright.txt',meanvaluewriteright, '-
   append', 'delimiter','\t','precision',20);


rangeright=max(xr)-min(xr);
rangewriteright = [variable rangeright];
dlmwrite('rangeright.txt',rangewriteright, '-append', '
   delimiter','\t','precision',20);


LSimstd = std(LSim);
stdwriteLSim = [variable LSimstd];
```

```
dlmwrite('stdLSim.txt',stdwriteLSim, '-append', 'delimiter
   ','\t','precision',20);


LSimmean = mean(LSim);
meanwriteLSim = [variable LSimmean];
dlmwrite('meanLSim.txt',meanwriteLSim, '-append', '
   delimiter','\t','precision',20);


clear %Resets all variables before next loop

end
```

A.3.3 MonteCarloTheta.m

```
% Monte Carlo Theta Solver - Patrick Thewlis - 7/31/2020

% Monte Carlo style simulation.  Creates a data set, and
   adds noise, and
% repeatedly finds L.  Sampling a large number of Ls, we
   can explore
% uncertainty in the system.

% This is the main function for Monte Carlo 2D simulation.
tic
clear;
clc;
for theta=5:5:90
%Input desired noise, iterations, initial guess point
sx = 0.0000110; % spot noise X
sy = 0.000073; % spot noise Y
sz = 0.000073; % spot noise Z
variable = theta;
%Input starting guesses
guessleft=[0 0 0 0 0 0]; %initial guess - [x y z a b g] -
     6 DOF, translation and orientation
guessright=[-0.039 -0.039 0 0 0 0];
iter=500; % iteration count 1 for external loop
n=11; % n=input('input n (# of horizontal beams):') %
   horizontal dot number
dL=1; %left side projectioj distance d=input('input
   distance from module to projection surface:')
dR=1.039; % right side projection distance to wall/surface
theta0A=theta/(n-1); % theta0=input('input initial
   interbeam angle (fullangle / n-1 beams):') BEAM A
theta0B=theta/(n-1); % theta0=input('input initial
   interbeam angle (fullangle / n-1 beams):') BEAM B
oriL=[0 0 0]; % oriL=input('input the origin (recommended
   [0 0 0]):') %Format [X Y Z]
htmX = -(dL-dR);      % X offset = input('input the offset
   between modules') (SIGNS ARE FLIPPED)
htmY = 0.039; % Y offset = input('input the offset between
    modules') (SIGNS ARE FLIPPED)
htmZ = 0;      % Z offset = input('input the offset between
    modules') (SIGNS ARE FLIPPED)
htmA = 0;      % HTM X axis rotation (SIGNS ARE FLIPPED)
htmB = 0;      % HTM Y axis rotation (SIGNS ARE FLIPPED)
htmC = 0;      % HTM Z axis rotation (SIGNS ARE FLIPPED)
```

```
[aL,mL,uvcalL,sphcalL] = scatterleft(n,dL,theta0A,oriL);
[aR,mR,uvcalR,sphcalR] = scatterright(n,dR,theta0B,oriL);

% Prep left side calibration data
%==================================================

calbaseL(:,2) = mL(:,1); % x->y
calbaseL(:,3) = mL(:,2); % y->z
calbaseL(:,1) = mL(:,3); % z->x

for i = 1:n*n
    [azL(i,:) elL(i,:) rL(i,:)] = cart2sph(calbaseL(i,1),
       calbaseL(i,2), calbaseL(i,3));
end

sphdataL(1,:) = azL;
sphdataL(2,:) = elL;

% Prep right side calibration data
%==================================================


calbaseR(:,2) = mR(:,1); % x->y
calbaseR(:,3) = mR(:,2); % y->z
calbaseR(:,1) = mR(:,3); % z->x

for i = 1:n*n
    [azR(i,:) elR(i,:) rR(i,:)] = cart2sph(calbaseR(i,1),
       calbaseR(i,2), calbaseR(i,3));
end

sphdataR(1,:) = azR;
sphdataR(2,:) = elR;
%sphcal = sphdata(1:2,:);

% Pre left and right grid data
%==================================================

%left
leftdata(:,2) = mL(:,1); % x->y
leftdata(:,3) = mL(:,2); % y->z
leftdata(:,1) = mL(:,3); % z->x

%right
```

```
%mR(:,1) = mR(:,1) + offset;
rightdataprep = zeros(n*n,3);
rightdataprep(:,2) = mR(:,1); % x->y
rightdataprep(:,3) = mR(:,2); % y->z
rightdataprep(:,1) = mR(:,3); % z->x

hgt = makehgtform('translate',[htmX htmY htmZ],'xrotate',
   htmA,'yrotate',htmB,'zrotate',htmC); % 6 DOF

rightdataprep=rightdataprep';
rightdataprep(4,:)=1;

rightdataoffset=inv(hgt)*rightdataprep;
rightdata = rightdataoffset(1:3,:);
rightdata = rightdata';

save('sphcalL.mat','sphdataL');
save('sphcalR.mat','sphdataR');
save('Left.mat','leftdata');
save('Right.mat','rightdata');

% guessL = guessleft.*randn(1,6);
% guessR = guessright.*randn(1,6);

% Begin External Monte Carlo Loop
% Preallocate LSim array to speed up computation
LSim = zeros(iter,1);

for q = 1:iter

%Do the minimization
[Lmean]=noiseleftminimization(n,sx,sy,sz,leftdata,
   guessleft);
[Rmean]=noiserightminimization(n,sx,sy,sz,rightdata,
   guessright);

xl(q,:)=Lmean;
xr(q,:)=Rmean;

%fit into one matrix
combined = [Lmean;Rmean];

%use pdist to find euclidean distance...  (rows =
   observations, columns =
%variables... ie: row 1 -> X1 Y1 Z1, row 2 -> X2 Y2 Z2
```

```
LSim(q,1) = pdist(combined);

disp(q)
end

%save and process data
stdleft = std(xl);
stdwriteleft = [variable stdleft];
dlmwrite('stdleft.txt',stdwriteleft, '-append', 'delimiter
   ','\t','precision',20);

stdright = std(xr);
stdwriteright = [variable stdright];
dlmwrite('stdright.txt',stdwriteright, '-append', '
   delimiter','\t','precision',20);

meanvalueleft=mean(xl);
meanvaluewriteleft = [variable meanvalueleft];
dlmwrite('meanvalueleft.txt',meanvaluewriteleft, '-append'
   , 'delimiter','\t','precision',20);

rangeleft=max(xl)-min(xl);
rangewriteleft = [variable rangeleft];
dlmwrite('rangeleft.txt',rangewriteleft, '-append', '
   delimiter','\t','precision',20);

meanvalueright=mean(xr);
meanvaluewriteright = [variable meanvalueright];
dlmwrite('meanvalueright.txt',meanvaluewriteright, '-
   append', 'delimiter','\t','precision',20);

rangeright=max(xr)-min(xr);
rangewriteright = [variable rangeright];
dlmwrite('rangeright.txt',rangewriteright, '-append', '
   delimiter','\t','precision',20);

LSimstd = std(LSim);
stdwriteLSim = [variable LSimstd];
dlmwrite('stdLSim.txt',stdwriteLSim, '-append', 'delimiter
   ','\t','precision',20);

LSimmean = mean(LSim);
meanwriteLSim = [variable LSimmean];
dlmwrite('meanLSim.txt',meanwriteLSim, '-append', '
```

```
   delimiter','\t','precision',20);

% figure
% hist(LSim,sqrt(iter))
% % save('LSim.mat','LSim','xr','xl')
% std(LSim)
% %
% % dlmwrite('xleft.txt',xl, '-append', 'delimiter','\t','
   precision',20);
% % dlmwrite('xright.txt',xr, '-append', 'delimiter','\t
   ','precision',20);
%
% toc
%
% (mean(LSim) - std(LSim))/mean(LSim)
end
```

A.4 Module Calibrator

A.4.1 ModuleCalibrator.m

```
% Module Calibrator - Patrick Thewlis - 7/31/2020

clear;
clc;
close all

% Transforms & Guesses...
starttrans = [-1.46 -0.96 0.28]; % Guess for X/Y/Z
   translation 3 DoF
startrot = [0 0 0]; % Guess for B/C rotation

% OPPOSITE Translations and rotations [X Y Z A B C], where
% X Y Z are the shifts in those axis, and A B C are the
   rotations about X Y Z
% REMINDER - X = projection depth, Y = horiz, Z = vertical
   .. RH system!
% A B C rotations are rotations about X Y Z in radians.
% REMEMBER!  These are opposite...  Translation of [0 1 1
   0 0 0] for
% example will move the projection point to [0 -1 -1 0 0
   0].  SHIFT OF
% PERSPECTIVE!

n=11; % n=input('input n (# of horizontal beams):') %
   horizontal dot number
% theta0=29/(n-1); % theta0=input('input initial interbeam
    angle (fullangle / n-1 beams):')
oriL=[0 0 0]; % oriL=input('input the origin (recommended
   [0 0 0]):') %Format [X Y Z]

% Load the 121x3x10 sorted array of calibration data (10
   slices)
load Mirror.mat
Total = Mirror/1000; %convert data in mm to meters

%Take transpose to correct formatting
for i = 1:10
    raw(:,:,i) = Total(:,:,i)';
end

% Next, plot the 3D view of all ten slices to check...
figure
```

```
hold on
title('10 Slice Calibration Dataset')
for i=1:10
    scatter3(raw(1,:,i),raw(2,:,i),raw(3,:,i))
end
xlabel('X (m)','FontSize',20)
ylabel('Y (m)','FontSize',20)
zlabel('Z (m)','FontSize',20)
ax = gca;
ax.FontSize = 20;
ax.TickLength = [0.01, 0.005]; % Make tick marks longer.
ax.LineWidth = 2; % Make tick marks thicker.

% Duplicate the dataset to hold for later
raw_duplicate = raw;

data_formatted=raw_duplicate;

%Create a fourth row of all 1s, as required for the below
   HGT application
data_formatted(4,:,:)=1;

data_processed = data_formatted;

%Next, utilize the optimization function to find the
   projection location!
%Recall that a projection position is guessed... this is
   input into the
%HTM, and then differences are taken.

OPTIONS = optimset('Display','iter','TolFun',1e-60,'TolX'
   ,1e-60,'MaxIter',3000,'MaxFunEvals',3000);
[trans]= fminsearch(@(x)CalDataMinimization(n,
   data_processed,x),starttrans,OPTIONS);

trans; %output the translation value

% NEXT - take that location, and use HGT to shift the
   coordinate system
% back to the original projection location... Once THAT is
    done, you can
% look at raw computing the rotations in X/Y/Z like you
   did before!

% 3 DoF Final HGT
```

```
hgtmid = makehgtform('translate',[trans(1,1) trans(1,2)
   trans(1,3)]);

% Apply the HGT to the initial data array, slice by slice
   ...  We're getting
% going to center the projection point on the global
   coordinate system's
% origin for calibration.  (Angular relationship between
   beams is what we
% want!

for i = 1:10
    datatrans(:,:,i)=inv(hgtmid)*data_processed(:,:,i);
end

% Rotate the matrix to correct for 6 DoF
%
% OPTIONS = optimoptions(@fminunc,'Display','iter','TolFun
   ',1e-30,'TolX',1e-30,'MaxIter',3000,'Algorithm','quasi-
   newton', 'MaxFunctionEvaluations',3000);
% [rot]= fminunc(@(x)newrotation(n,datatrans,x),startrot,
   OPTIONS);

OPTIONS = optimset('Display','iter','TolFun',1e-60,'TolX'
   ,1e-60,'MaxIter',1000,'MaxFunEvals',1000);
[rot]= fminsearch(@(x)newrotation_v3(n,datatrans,x),
   startrot,OPTIONS);

rot % output the rotation value
trans % displays the translation value

hgtfinal = makehgtform('xrotate',rot(1),'yrotate',rot(2),'
   zrotate',rot(3));

% Apply HGT..  LOCAL projection data set coordinate system
    is now translated and rotated to
% align with the global coordinate system!

for i = 1:10
    caldata(:,:,i)=inv(hgtfinal)*datatrans(:,:,i);
end

% Eliminate 1's row of HGT
caldata=caldata(1:3,:,:);
caldatamean = mean(caldata,3);
```

```
% Convert the final translated & rotated data to a
   spherical coordinate system
for j = 1:10
    for i = 1:n*n
        [azcal(:,i,j) elcal(:,i,j) rcal(:,i,j)] = cart2sph
           (caldata(1,i,j), caldata(2,i,j), caldata(3,i,j)
           );
    end
end

sphcaldata(1,:,:) = azcal;
sphcaldata(2,:,:) = elcal;
sphcaldata(3,:,:) = rcal;
sphcaldatamean = mean(sphcaldata,3);

% Plot original data vs. rotated / translate data (shifted
    to global
% coordinate system) as a quality test to gauge the
   effectiveness of the
% algorithm.

caldatamean = mean(caldata,3);
raw_duplicatemean = mean(raw_duplicate,3);

% Plot calibrated dataset to verify
figure
title('Calibrated Dataset')
hold on
for i=1:10
    scatter3(raw_duplicate(1,:,i),raw_duplicate(2,:,i),
       raw_duplicate(3,:,i), 'blue', '+')
end
for i=1:10
    scatter3(caldata(1,:,i),caldata(2,:,i),caldata(3,:,i),
        'red', 'o')
end
xlabel('X (m)','FontSize',20)
ylabel('Y (m)','FontSize',20)
zlabel('Z (m)','FontSize',20)
ax = gca;
ax.FontSize = 20;
ax.TickLength = [0.01, 0.005]; % Make tick marks longer.
ax.LineWidth = 2; % Make tick marks thicker.
```

```
% Save the output calibration data
pelliclecalspherical = sphcaldatamean;
save('pelliclecalspherical.mat','pelliclecalspherical');
save('pellicledata.mat','trans','rot');

mirrorcalspherical = sphcaldatamean;
save('mirrorcalspherical.mat','mirrorcalspherical');
save('mirrordata.mat','trans','rot');
```

A.4.2 newrotation_v3.m

```
function [value] = newrotation_v3(n,a,guess)
format long
value = 0;

% Input the rotation guesses
A = guess(1,1);
B = guess(1,2);
C = guess(1,3);

% Use the guess to create the HGT
hgt = makehgtform('xrotate', guess(1), 'yrotate',guess(2),
    'zrotate',guess(3));

% Apply the shift to every slice
for i = 1:10
    cartdata(:,:,i)=inv(hgt)*a(:,:,i);
end

% Pull out the spherical coordinates of the translated
    dataset
for j = 1:10
    for i = 1:n*n
        [az(:,i,j) el(:,i,j) r(:,i,j)] = cart2sph(cartdata
            (1,i,j), cartdata(2,i,j), cartdata(3,i,j));
    end
end

% Arrange spherical translational data into single data
    array
sphdata(1,:,:) = az;
sphdata(2,:,:) = el;
sphdata(3,:,:) = r;

% Next, we'll judge the beams as follows...


% sphdata has format..3 rows.. (az,el,r), n*n columns =
    beam numbers, and depth)
% sphdata is 3 x 121 x 10, az el r rows

sphaz = sphdata(1,:,:);
sphel = sphdata(2,:,:);
```

```
for i = 1:10
    sphazreshaped(:,:,i) = reshape(sphaz(:,:,i),[11,11]);
    sphelreshaped(:,:,i) = reshape(sphel(:,:,i),[11,11]);
end

sphazrot = rot90(sphazreshaped,3);
sphelrot = rot90(sphelreshaped,3);

azvalue = sum(sum(sum(abs(sphazrot(:,:,:)))));
elvalue = sum(sum(sum(abs(sphelrot(:,:,:)))));

value = azvalue + elvalue;

guess; % output a guess to keep track during the
   minimization process


end
```

A.5 Monte Carlo Calibration Simulator - Spot Noise Range

A.5.1 MCCalibratorSRange.m

```
% Calibration Monte Carlo Simulation - Range of Spot Noise
    S
% Patrick Thewlis - 7/31/2020
% Program generates 10 'slices' of data points, and runs
   calibration....
% (n x n x distance) array, where distance is ten
   projection distances

clear;
clc;
close all

% Transforms & Guesses...
starttrans = [-0.4 -0.4 0]; % Guess for X/Y/Z translation
   3 DoF
startrot = [0 0 0]; % Guess for B/C rotation

% Translate & Rotate Sample Data Values
transrot = [-0.039 -0.039 0 0 0 0];

% OPPOSITE Translations and rotations [X Y Z A B C], where
% X Y Z are the shifts in those axis, and A B C are the
   rotations about X Y Z

% REMINDER - X = Module Calibrator 2-26-2020projection
   depth, Y = horiz, Z = vertical.. RH system!

% A B C rotations are rotations about X Y Z in radians.
% REMEMBER!  These are opposite...  Translation of [0 1 1
   0 0 0] for

% example will move the projection point to [0 -1 -1 0 0
   0].  SHIFT OF
% PERSPECTIVE!

iterations = 200;
n=11; % n=input('input n (# of horizontal beams):') %
   horizontal dot number
theta0=29/(n-1); % theta0=input('input initial interbeam
   angle (fullangle / n-1 beams):')
oriL=[0 0 0]; % oriL=input('input the origin (recommended
   [0 0 0]):') %Format [X Y Z]
```

```
calibration_range = linspace(0.75,1.5,10);
% % Generate the sample calibration data array
for d=1:1:10 %d=input('input distance from module to
   projection surface:') %distance from projection wall
    depth = calibration_range(1,d);
    [aL(:,:,d),grid_raw(:,:,d)] = scatterleft(n,depth,
       theta0,oriL);
end

%Take transpose to correct formatting
for i = 1:10
    grid_raw_transp(:,:,i) = grid_raw(:,:,i)';
end


% Change coordinates from =weird left handed system to
   something that will
% play nicely with a spherical coordinate system.  Z->(+X)
   . Y->(+Z), X->(-Y)
% In Spherical... X = boresight direction (projection...
   Z is vertical,
% and LEFT Y is positive, right side Y is negative...  IE:
    RH system!
% Pointer = +X, middle = +Y, Thumb = +Z.

% This section fixes Ben's odd reverse left hand
   coordinate system
grid_RHC = grid_raw_transp;
grid_RHC(1,:,:) = grid_raw_transp(3,:,:);
grid_RHC(2,:,:) = grid_raw_transp(1,:,:);
grid_RHC(3,:,:) = grid_raw_transp(2,:,:);
grid_RHC(2,:,:) = grid_RHC(2,:,:).*(-1); % And flip the
   sign on the Y's to make it a RH system...


hgt = makehgtform('translate',[transrot(1) transrot(2)
   transrot(3)],'xrotate',transrot(4),'yrotate',transrot
   (5),'zrotate',transrot(6));
grid_htm_prep=grid_RHC;

%Create a fourth row of all 1s, as required for the below
   HGT application
grid_htm_prep(4,:,:)=1;

grid_transformed = grid_htm_prep;
```

```
% %Apply the HGT to the array, slice by slice
for i = 1:10
    grid_transformed(:,:,i)=(hgt)*grid_htm_prep(:,:,i);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% % Next, plot the 3D view of all ten slices...
% figure
% hold on
% for i=1:10
%     scatter3(grid_RHC(1,:,i),grid_RHC(2,:,i),grid_RHC
   (3,:,i),'r*')
%     scatter3(grid_transformed(1,:,i),grid_transformed
   (2,:,i),grid_transformed(3,:,i),'b*')
% end
% axis equal

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Noise adding for loops
%for s = 0.000050:0.000010:0.000070
for s = 0.000075
    for loop_iteration = 1:iterations

        [Lmean]=loop_external_SRange(n,s,grid_transformed,
           starttrans,startrot);
        xl(loop_iteration,:)=Lmean;
        loop_iteration
        s
    end

variable = s;

stdleft = std(xl);
stdwriteleft = [variable stdleft];
dlmwrite('stdleft.txt',stdwriteleft, '-append', 'delimiter
   ','\t','precision',20);

meanvalueleft=mean(xl);
meanvaluewriteleft = [variable meanvalueleft];
dlmwrite('meanvalueleft.txt',meanvaluewriteleft, '-append'
   , 'delimiter','\t','precision',20);
```

```
rangeleft=max(xl)-min(xl);
rangewriteleft = [variable rangeleft];
dlmwrite('rangeleft.txt',rangewriteleft, '-append', '
   delimiter','\t','precision',20);

end
```

A.5.2 loop_external_SRange.m

```
function [meanvalueleft]=loop_external_SRange(n,s,
   grid_transformed,starttrans,startrot)

%
% % Add noise to the data... - 121 x 3 (X Y Z array for
   all points)
% grid_noise=zeros(n*n,3); %pre-allocate array to speed
   calculation up
% spotx=randn(n*n,1); %generate a random array
% spoty=randn(n*n,1); %generate a random array
% spotz=randn(n*n,1); %generate a random array
%
%
% for i=1:n*n
%         grid_noise(i,1)=grid(i,1)+(sx*spotx(i));
%         grid_noise(i,2)=grid(i,2)+(sy*spoty(i));
%         grid_noise(i,3)=grid(i,3)+(sz*spotz(i));
% end

% start(1:3) = start(1:3) + (0.0001.*randn(1,3)); %
   slightly randomizes the starting guess to ensure we
   approach it from all directions

grid_noise=zeros(3,n*n);
spot=randn(3,n*n,10);

for m=1:10
    for i=1:3
        for j=1:n*n
        grid_noise(i,j,m)=grid_transformed(i,j,m)+s*spot(i
           ,j,m);
        end
    end
end

grid_noise(4,:,:)=1;


OPTIONS = optimset('Display','off','TolFun',1e-8,'TolX',1e
   -8,'MaxIter',2000,'MaxFunEvals',2000);
[trans]= fminsearch(@(x)CalDataMinimization(n,grid_noise,x
   ),starttrans,OPTIONS);
```

```
trans; %output the translation value

% 3 DoF Final HGT
hgtmid = makehgtform('translate',[trans(1,1) trans(1,2)
   trans(1,3)]);

% Apply the HGT to the data array
for i = 1:10
    datatrans(:,:,i)=inv(hgtmid)*grid_noise(:,:,i);
end

% Rotate the matrix to correct for 6 DoF
OPTIONS = optimset('Display','off','TolFun',1e-15,'TolX',1
   e-15,'MaxIter',2000,'MaxFunEvals',2000);
[rot]= fminsearch(@(x)newrotation_v3(n,datatrans,x),
   startrot,OPTIONS);

rot; % output the rotation value
trans; % displays the translation value

meanvalueleft(1,1:3)=trans;
meanvalueleft(1,4:6)=rot;

end
```

A.5.3 CalDataMinimization.m

```
function [value] = CalDataMinimization(n,a,guess)
format long

value = 0;

X = guess(1,1);
Y = guess(1,2);
Z = guess(1,3);
% A = guess(1,4);
% B = guess(1,5);
% C = guess(1,6);
% B = guess(1,4);
% C = guess(1,5);

% hgt = makehgtform('translate',[X Y Z],'xrotate',A,'
   yrotate',B,'zrotate',C); % 6 DOF
% hgt = makehgtform('translate',[X Y Z],'yrotate',B,'
   zrotate',C); % 5 DOF

hgt = makehgtform('translate',[X Y Z]); % 3 DOF

% Apply the HTM to all slices of the cartesian data array
for i=1:10
    mm(:,:,i)=inv(hgt)*a(:,:,i);
end

% Remove the 4th row of 1's (used for HTM)
for i=1:10
        m(:,:,i) = mm(1:3,:,i);
end

% Convert the data to a spherical coordinate system
for j = 1:10
    for i = 1:n*n
        [az(:,i,j) el(:,i,j) r(:,i,j)] = cart2sph(m(1,i,j)
            , m(2,i,j), m(3,i,j));
    end
end

sphdata(1,:,:) = az;
sphdata(2,:,:) = el;
sphdata(3,:,:) = r;
```

```
% Preallocate arrays for faster calculations
workingarrayAz = zeros(1,10,n^2);
workingarrayEl = zeros(1,10,n^2);

% Pull out all angular components and put them into arrays
for beam = 1:(n*n)
    for slice = 1:10
        workingarrayAz(1,slice,beam) = sphdata(1,beam,
           slice);
        workingarrayEl(1,slice,beam) = sphdata(2,beam,
           slice);
    end
end

stdAz = std(workingarrayAz);
stdEl = std(workingarrayEl);

Aztotal = sum(stdAz);
Eltotal = sum(stdEl);

% absAz = abs(workingarrayAz);
% absEl = abs(workingarrayEl);
%
% Aztotal = sum(sum(absAz));
% Eltotal = sum(sum(absEl));

value = (Aztotal + Eltotal);

guess;
save('arrays.mat','workingarrayAz','workingarrayEl');

end
```

A.6 Monte Carlo Calibration Simulator - X/Y/Z Spot Noise

A.6.1 MCCalibratorXYZS.m

```
% Calibration Monte Carlo Simulation - Specific XYZ Noise
   S
% Patrick Thewlis - 7/31/2020
% Program generates 10 'slices' of data points, and runs
   calibration....
% (n x n x distance) array, where distance is ten
   projection distances

clear;
clc;
close all

% Transforms & Guesses...
starttrans = [-0.04 -0.04 0]; % Guess for X/Y/Z
   translation 3 DoF
startrot = [0 0 0]; % Guess for B/C rotation

% Translate & Rotate Sample Data Values
transrot = [-0.039 -0.039 0 0 0 0];

% OPPOSITE Translations and rotations [X Y Z A B C], where
% X Y Z are the shifts in those axis, and A B C are the
   rotations about X Y Z
% REMINDER - X = Module Calibrator 2-26-2020projection
   depth, Y = horiz, Z = vertical.. RH system!
% A B C rotations are rotations about X Y Z in radians.
% REMEMBER!  These are opposite...  Translation of [0 1 1
   0 0 0] for
% example will move the projection point to [0 -1 -1 0 0
   0].  SHIFT OF
% PERSPECTIVE!

sx = 0.000110; % Spot noise in the X,Y,Z
sy = 0.000075;
sz = 0.000075;
s = [sx sy sz];

iterations = 20;
n=11; % n=input('input n (# of horizontal beams):') %
   horizontal dot number
theta0=29/(n-1); % theta0=input('input initial interbeam
   angle (fullangle / n-1 beams):')
```

```
oriL=[0 0 0]; % oriL=input('input the origin (recommended
   [0 0 0]):') %Format [X Y Z]
calibration_range = linspace(0.75,1.5,10);
% % Generate the sample calibration data array
for d=1:1:10 %d=input('input distance from module to
   projection surface:') %distance from projection wall
    depth = calibration_range(1,d);
    [aL(:,:,d),grid_raw(:,:,d)] = scatterleft(n,depth,
       theta0,oriL);
end

%Take transpose to correct formatting
for i = 1:10
    grid_raw_transp(:,:,i) = grid_raw(:,:,i)';
end

% Change coordinates from =weird left handed system to
   something that will
% play nicely with a spherical coordinate system.  Z->(+X)
   . Y->(+Z), X->(-Y)
% In Spherical... X = boresight direction (projection...
   Z is vertical,
% and LEFT Y is positive, right side Y is negative...  IE:
    RH system!
% Pointer = +X, middle = +Y, Thumb = +Z.

% This section fixes Ben's odd reverse left hand
   coordinate system
grid_RHC = grid_raw_transp;
grid_RHC(1,:,:) = grid_raw_transp(3,:,:);
grid_RHC(2,:,:) = grid_raw_transp(1,:,:);
grid_RHC(3,:,:) = grid_raw_transp(2,:,:);
grid_RHC(2,:,:) = grid_RHC(2,:,:).*(-1); % And flip the
   sign on the Y's to make it a RH system...


hgt = makehgtform('translate',[transrot(1) transrot(2)
   transrot(3)],'xrotate',transrot(4),'yrotate',transrot
   (5),'zrotate',transrot(6));
grid_htm_prep=grid_RHC;

%Create a fourth row of all 1s, as required for the below
   HGT application
grid_htm_prep(4,:,:)=1;
```

```
grid_transformed = grid_htm_prep;

% %Apply the HGT to the array, slice by slice
for i = 1:10
    grid_transformed(:,:,i)=(hgt)*grid_htm_prep(:,:,i);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% % Next, plot the 3D view of all ten slices...
% figure
% hold on
% for i=1:10
%     scatter3(grid_RHC(1,:,i),grid_RHC(2,:,i),grid_RHC
   (3,:,i),'r*')
%     scatter3(grid_transformed(1,:,i),grid_transformed
   (2,:,i),grid_transformed(3,:,i),'b*')
% end
% axis equal

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for loop_iteration = 1:iterations

        [Lmean]=loop_external_XYZS(n,s,grid_transformed,
           starttrans,startrot);
        xl(loop_iteration,:)=Lmean;
        loop_iteration
    end

variable = s;

stdleft = std(xl);
stdwriteleft = [variable stdleft];
dlmwrite('stdleft.txt',stdwriteleft, '-append', 'delimiter
   ','\t','precision',20);

meanvalueleft=mean(xl);
meanvaluewriteleft = [variable meanvalueleft];
dlmwrite('meanvalueleft.txt',meanvaluewriteleft, '-append'
   , 'delimiter','\t','precision',20);

rangeleft=max(xl)-min(xl);
rangewriteleft = [variable rangeleft];
dlmwrite('rangeleft.txt',rangewriteleft, '-append', '
   delimiter','\t','precision',20);
```

A.6.2 loop_external_XYZS.m

```
function [meanvalueleft]=loop_external_XYZS(n,s,
   grid_transformed,starttrans,startrot)

% % Add noise to the data... - 121 x 3 (X Y Z array for
   all points)
grid_noise=zeros(3,n*n);
spot=randn(3,n*n,10);

for m=1:10
        for j=1:n*n
        grid_noise(1,j,m)=grid_transformed(1,j,m)+s(1,1)*
           spot(1,j,m); % X noise
        grid_noise(2,j,m)=grid_transformed(2,j,m)+s(1,2)*
           spot(2,j,m); % Y noise
        grid_noise(3,j,m)=grid_transformed(3,j,m)+s(1,3)*
           spot(3,j,m); % Z noise
        end
end

grid_noise(4,:,:)=1;

OPTIONS = optimset('Display','off','TolFun',1e-8,'TolX',1e
   -8,'MaxIter',2000,'MaxFunEvals',2000);
[trans]= fminsearch(@(x)CalDataMinimization(n,grid_noise,x
   ),starttrans,OPTIONS);

trans; %output the translation value

% 3 DoF Final HGT
hgtmid = makehgtform('translate',[trans(1,1) trans(1,2)
   trans(1,3)]);

% Apply the HGT to the data array
for i = 1:10
    datatrans(:,:,i)=inv(hgtmid)*grid_noise(:,:,i);
end

% Rotate the matrix to correct for 6 DoF
OPTIONS = optimset('Display','off','TolFun',1e-15,'TolX',1
   e-15,'MaxIter',2000,'MaxFunEvals',2000);
[rot]= fminsearch(@(x)newrotation_v3(n,datatrans,x),
   startrot,OPTIONS);
```

```
rot; % output the rotation value
trans; % displays the translation value

meanvalueleft(1,1:3)=trans;
meanvalueleft(1,4:6)=rot;

end
```

A.7 Supporting Functions

A.7.1 Algorithm_LSolver.m

```
function [value] = Algorithm_LSolver(n,dataset,sphcal,
   guess)
format long
value = 0; % clear out variable

X = guess(1,1); % Format guessed data
Y = guess(1,2);
Z = guess(1,3);
A = guess(1,4);
B = guess(1,5);
C = guess(1,6);

% Create the HGT from the input guessed location
hgt = makehgtform('translate',[X Y Z],'xrotate',A,'yrotate
   ',B,'zrotate',C); % 6 DOF

% Reformat the matrix to prepare for HGT
dataset=dataset';
dataset(4,:)=1;

% Take the inverse HGT to perform a passive transformation
dataset_trans=inv(hgt)*dataset;

% Remove the 4th row of 1's (used for HTM)
data(:,:) = dataset_trans(1:3,:);

% Convert the data to a spherical coordinate system
for i = 1:n*n
    [az(:,i) el(:,i) r(:,i)] = cart2sph(data(1,i), data(2,
       i), data(3,i));
end

% Reformat the data
data_transformed(1,:) = az;
data_transformed(2,:) = el;
sphcal = sphcal(1:2,:);

% Take the difference between the calibration and the new
   data spherical
% coordinate locations
residual = sphcal - data_transformed;
```

```
% Square the data
res_square = residual.^2;

% Sum the difference over the entire set of beams.. This
   value will be
% minimized.
value = sum(sum(res_square));
```

A.7.2 scatterleft.m / scatteright.m

Scatter is a MATLAB® script written to generate a fanout beam pattern for simulation purposes. The program scatter.m was written by Benrui Zheng as part of his 2014 Ph.D. dissertation [7]. It has been slightly altered from its original form in order to put the generated pattern into a right-hand coordinate system, and then converted to spherical coordinates. Left and right versions are used such that the generated simulation pattern can be uniquely saved to a different file. A single version is presented here, though both are functionally the same.

```
function [a,m,uv,sphcal] = scatterleft(n,d,theta0,ori2)

%   This function was written by Benrui Zheng, and used in
    completion of
%   his 2014 PhD dissertation, titled "Positioning sensor
   by combining
%   optical projection and photogrammetry".

%   The function has been altered from its original form
   to convert the
%   solved Cartesian coordinates to a right hand
   coordinate system, and
%   then into spherical coordinates


%   n represents the length of the scattered dot matrix
   you want to
%   plot,d represents the distence between the scattered
   dot matrix and
%   light source,theta0 represents the included angle,
   enter them in order
%   please.for example,myscatter( 7,5,2). a and m are the
   solution matrix.
%   format double

global x;
global y;
global ori;
global theta;
format long
ori=ori2;
theta=theta0/180*pi;
a=cell(n);
a{(n+1)/2,(n+1)/2}=ori+[0,0,d];

for i=1:(n-1)/2
```

```
        a{(n+1)/2-i,(n+1)/2}(1,2)=d*tan(theta*i)+a{(n+1)/2,(n
           +1)/2}(1,2);
        a{(n+1)/2-i,(n+1)/2}(1,1)=a{(n+1)/2,(n+1)/2}(1,1);
        a{(n+1)/2-i,(n+1)/2}(1,3)=a{(n+1)/2,(n+1)/2}(1,3);
        a{(n+1)/2,(n+1)/2+i}(1,1)=d*tan(theta*i)+a{(n+1)/2,(n
           +1)/2}(1,1);
        a{(n+1)/2,(n+1)/2+i}(1,2)=a{(n+1)/2,(n+1)/2}(1,2);
        a{(n+1)/2,(n+1)/2+i}(1,3)=a{(n+1)/2,(n+1)/2}(1,3);
    end

    for j=1:(n-1)/2
        for i=j:(n-1)/2
             for t=1:10
            x=a{(n+1)/2-j,(n+1)/2+i-1};
            y=a{(n+1)/2-j+1,(n+1)/2+i};
            options=optimset('Display','off','TolFun',1e-20);
            a{(n+1)/2-j,(n+1)/2+i}=fsolve(@geometry,[(a{(n+1)
               /2-j,(n+1)/2+i-1}(1,1)+a{(n+1)/2-j+1,(n+1)/2+i
               }(1,1))/2+t+d,(a{(n+1)/2-j,(n+1)/2+i-1}(1,2)+a
               {(n+1)/2-j+1,(n+1)/2+i}(1,2))/2+t+d,d],options)
               ;
            if a{(n+1)/2-j,(n+1)/2+i}(1,1)-(a{(n+1)/2-j,(n+1)
               /2+i-1}(1,1)+a{(n+1)/2-j+1,(n+1)/2+i}(1,1))/2>0
                && a{(n+1)/2-j,(n+1)/2+i}(1,2)-(a{(n+1)/2-j,(n
               +1)/2+i-1}(1,2)+a{(n+1)/2-j+1,(n+1)/2+i}(1,2))
               /2>0
               a{(n+1)/2-j,(n+1)/2+i}=a{(n+1)/2-j,(n+1)/2+i};
                  break;
            end
            end
        end

        for i=j:(n-1)/2
            for t=1:10
            x=a{(n+1)/2-i+1,(n+1)/2+j};
            y=a{(n+1)/2-i,(n+1)/2+j-1};
            options=optimset('Display','off','TolFun',1e-20);
            a{(n+1)/2-i,(n+1)/2+j}=fsolve(@geometry,[(a{(n+1)
               /2-i+1,(n+1)/2+j}(1,1)+a{(n+1)/2-i,(n+1)/2+j
               -1}(1,1))/2+t+d,(a{(n+1)/2-i+1,(n+1)/2+j}(1,2)+
               a{(n+1)/2-i,(n+1)/2+j-1}(1,2))/2+t+d,d],options
               );
           if a{(n+1)/2-i,(n+1)/2+j}(1,1)-(a{(n+1)/2-i+1,(n+1)
              /2+j}(1,1)+a{(n+1)/2-i,(n+1)/2+j-1}(1,1))/2>0 &&
              a{(n+1)/2-i,(n+1)/2+j}(1,2)-(a{(n+1)/2-i+1,(n+1)
```

```
          /2+j}(1,1)+a{(n+1)/2-i,(n+1)/2+j-1}(1,1))/2>0
           a{(n+1)/2-i,(n+1)/2+j}=a{(n+1)/2-i,(n+1)/2+j};
              break;
        end
        end
     end
end
for i=1:(n-1)/2
    for j=0:(n-1)/2
        a{(n+1)/2+i,(n+1)/2+j}(1,1)=a{(n+1)/2-i,(n+1)/2+j
           }(1,1);
         a{(n+1)/2+i,(n+1)/2+j}(1,2)=2*a{(n+1)/2,(n+1)
            /2}(1,2)-1*a{(n+1)/2-i,(n+1)/2+j}(1,2);
         a{(n+1)/2+i,(n+1)/2+j}(1,3)=a{(n+1)/2-i,(n+1)/2+j
            }(1,3);
    end
end

for i=1:(n+1)/2
    for j=1:(n-1)/2
        a{i,(n-1)/2-j+1}(1,1)=2*a{(n+1)/2,(n+1)/2}(1,1)-1*
           a{i,(n+1)/2+j}(1,1);
         a{i,(n-1)/2-j+1}(1,2)=a{i,(n+1)/2+j}(1,2);
         a{i,(n-1)/2-j+1}(1,3)=a{i,(n+1)/2+j}(1,3);
    end
end
for i=(n+1)/2+1:n
    for j=1:(n-1)/2
        a{i,(n-1)/2-j+1}(1,1)=2*a{(n+1)/2,(n+1)/2}(1,1)-1*
           a{i,(n+1)/2+j}(1,1);
         a{i,(n-1)/2-j+1}(1,2)=a{i,(n+1)/2+j}(1,2);
         a{i,(n-1)/2-j+1}(1,3)=a{i,(n+1)/2+j}(1,3);
    end
end



m=zeros(n,3);
uv=zeros(n,3);

tt=cell2mat(a); %converts data cell array to a matrix

for i=1:n
    for j=1:n
        m(n*(i-1)+j,1:3)=tt(i,3*j-2:3*j);
```

```
    end
end

xaxis=m(:,1);
yaxis=m(:,2);
zaxis=m(:,3);

% scatter(xaxis,yaxis)
% figure
% scatter3(xaxis,yaxis,zaxis)

%writes the data matrix m to a file!
dlmwrite('raw.txt',m, 'delimiter','\t','precision',20);
for i = 1:n*n
    uv(i,:) = m(i,:)/norm(m(i,:));
end
[az,el,r] = cart2sph(m(:,1),m(:,2),-m(:,3));  % RH
   coordinate System, then Spherical
sphcal = [az el r];
end
```

A.7.3 geometry.m

Geometry.m is a MATLAB® sub-function script written assist scatter.m in the generation of simulation beam data. This sub-function was originally titled 'zuobiao.m', which loosely translates to 'coordinate' in the Chinese language. The script was written by Benrui Zheng as part of his 2014 Ph.D. dissertation [7].

```
function F = geometry(z)

% format double

%   This function was written by Benrui Zheng, and used in
    completion of
%   his 2014 PhD dissertation, titled "Positioning sensor
   by combining
%   optical projection and photogrammetry".

%   Coordinate geometry computation sub-function of
   scatter.m

global x;
global y;
global ori;
global theta;
p1=x-ori;
p2=y-ori;
p3=z-ori;
F(1)=1000000*(cos(theta)-dot(p3,p2)/(norm(p3)*norm(p2)));
F(2)=1000000*(cos(theta)-dot(p3,p1)/(norm(p3)*norm(p1)));
F(3)=p3(1,3)-p2(1,3);
end
```

A.7.4 noiseleftminimization.m

```
function [meanvalueleft]=noiseleftminimization(n,sx,sy,sz,
   patterndata,start)

% Load the spherical coordinate calibration data for left
   grid
load('sphcalL.mat');

% Add noise to the data... - 121 x 3 (X Y Z array for all
   points)
noisydataleft=zeros(n*n,3); %pre-allocate array to speed
   calculation up

spotx=randn(n*n,1); %generate a random array for spot
   noise X
spoty=randn(n*n,1); %generate a random array for spot
   noise Y
spotz=randn(n*n,1); %generate a random array for spot
   noise Z

for i=1:n*n
        noisydataleft(i,1)=patterndata(i,1)+(sx*spotx(i));
        noisydataleft(i,2)=patterndata(i,2)+(sy*spoty(i));
        noisydataleft(i,3)=patterndata(i,3)+(sz*spotz(i));
end


%slightly randomize the starting guess to ensure we
   approach it from all directions
start(1:3) = start(1:3) + (0.0001.*randn(1,3));

% Use LSolver algorithm to apply passive HTM and guess
   position... Function is
% minimized at location convergence.
OPTIONS = optimset('Display','off','TolFun',1e-60,'TolX',1
   e-60,'MaxIter',5000,'MaxFunEvals',5000);
[meanvalueleft]= fminsearch(@(x)Algorithm_LSolver(n,
   noisydataleft,sphdataL,x),start,OPTIONS);

end
```

A.7.5 noiserightminimization.m

```
function [meanvalueright]=noiserightminimization(n,sx,sy,
   sz,patterndata,start)

% Load the spherical coordinate calibration data for right
    grid
load('sphcalR.mat');

% Add noise to the data... - 121 x 3 (X Y Z array for all
   points)
noisydataright=zeros(n*n,3); %pre-allocate array to speed
   calculation up

spotx=randn(n*n,1); %generate a random array for spot
   noise X
spoty=randn(n*n,1); %generate a random array for spot
   noise Y
spotz=randn(n*n,1); %generate a random array for spot
   noise Z

for i=1:n*n
        noisydataright(i,1)=patterndata(i,1)+(sx*spotx(i))
           ;
        noisydataright(i,2)=patterndata(i,2)+(sy*spoty(i))
           ;
        noisydataright(i,3)=patterndata(i,3)+(sz*spotz(i))
           ;
end

%slightly randomizes the starting guess to ensure we
   approach it from all directions
start(1:3) = start(1:3) + (0.0001.*randn(1,3));

% Use LSolver algorithm to apply passive HTM and guess
   position... Function is
% minimized at location convergence.
OPTIONS = optimset('Display','off','TolFun',1e-60,'TolX',1
   e-60,'MaxIter',5000,'MaxFunEvals',5000);
[meanvalueright]= fminsearch(@(x)Algorithm_LSolver(n,
   noisydataright,sphdataR,x),start,OPTIONS);

end
```

A.8 Other Scripts

A.8.1 Mirror Data Sorting "DataSortingM.m"

```
% Data Sorting Program
% Output from Photomodeler has no order!  We need to sort
   the grid top
% right to bottom left for the algorithm to have things
   worked out.

%Load the mirrorGRID*.txt file w/ MATLAB Import Data
   Function for a given
%data slice.  That data array needs to be named mirrorGRID
   .

n = 11; % Defines number of points on each side of the
   grid

sorteddata = sortrows(mirrorGRID,3,'descend'); %perform
   initial top to bottom data sorting

%Next, sort each set of of the horizontal grid rows so
   that they are
%ordered right to left.. (This is 'ascending' ordering...)
for k = 1:n
    sorteddata((((k*n)-n)+1):(k*n),:) = sortrows(sorteddata
       ((((k*n)-n)+1):(k*n),:),2,'ascend');
end

dM = sorteddata;
save('dM.mat','dM');

clear
clc

%Code to put the whole rest of the series into a matrix
   once everything is
%processed...

% Mirror(:,:,1) = M1GRID;
% Mirror(:,:,2) = M2GRID;
% Mirror(:,:,3) = M3GRID;
% Mirror(:,:,4) = M4GRID;
% Mirror(:,:,5) = M5GRID;
% Mirror(:,:,6) = M6GRID;
% Mirror(:,:,7) = M7GRID;
```

```
% Mirror(:,:,8) = M8GRID;
% Mirror(:,:,9) = M9GRID;
% Mirror(:,:,10) = M10GRID;
% save('Mirror.mat','Mirror');
```

A.8.2 Pellicle Data Sorting "DataSortingP.m"

```
% Data Sorting Program
% Output from Photomodeler has no order!  We need to sort
   the grid top
% right to bottom left for the algorithm to have things
   worked out.

n = 11; % Defines number of points on each side of the
   grid

% Load dataset (pellicleGRID.txt from PhotoModeler) via
   Import Data function.

sorteddata = sortrows(pellicleGRID,3,'descend'); %Perform
   initial top to bottom data sorting

%Next, sort each set of of the horizontal grid rows so
   that they are
%ordered right to left. (This is 'ascending' ordering...)

for k = 1:n
    sorteddata((((k*n)-n)+1):(k*n),:) = sortrows(sorteddata
       ((((k*n)-n)+1):(k*n),:),2,'ascend');
end

dP = sorteddata;
save('dP.mat','dP');
clear
clc

%Put the rest of the series into a matrix once everything
   is processed...

% Pellicle(:,:,1) = P1GRID;
% Pellicle(:,:,2) = P2GRID;
% Pellicle(:,:,3) = P3GRID;
% Pellicle(:,:,4) = P4GRID;
% Pellicle(:,:,5) = P5GRID;
% Pellicle(:,:,6) = P6GRID;
% Pellicle(:,:,7) = P7GRID;
% Pellicle(:,:,8) = P8GRID;
% Pellicle(:,:,9) = P9GRID;
% Pellicle(:,:,10) = P10GRID;
% save('Pellicle.mat','Pellicle');
```

A.8.3 Calibration Error Mapping "errorplotter.m"

```
%Calibration Error Plotter

clear
clc

load('PellicleTotal.mat') %Load calibration dataset (10
   slices x 121 beams)

n = 11; % 11 beams per side
gridn = n^2;

for beamID = 1:gridn

% Your two points
origin = [0,0,0];
meandirection = caldatamean(:,beamID); %Pull mean
   direction for first beam
meandirection = meandirection*2; %2x scales vector length
   so that we can be sure we fit all points
meandirection = meandirection';

% Vertical concatenation
pts = [origin; meandirection];

figure('DefaultAxesFontSize',18)
hold on
grid on
xlabel('X (m)', 'FontSize', 24);
ylabel('Y (m)', 'FontSize', 24);
zlabel('Z (m)', 'FontSize', 24);
ylim([-5E-4 5E-4])
plot3(pts(:,1), pts(:,2), pts(:,3),'-k','Linewidth',2) %
   Plot the mean point line

% Plot the existing points from the old calibration data
   set...

testsetdata = caldata(:,beamID,:);

for i = 1:10
    testset = testsetdata(:,:,i);
    testset = testset';
    finaltestset(i,:) = testset;
```

```
end

scatter3(finaltestset(:,1), finaltestset(:,2),
   finaltestset(:,3),'or','Linewidth',3)

for j = 1:10
    d(j,:,beamID) = point_to_line(finaltestset(j,:),origin
       ,meandirection);
end
    dmean = mean(d);
    hold off
end

ddata(1:gridn) = dmean(1,1,1:gridn); %Restructure data
   from array to matrix
matrix = vec2mat(ddata,11);

figure('DefaultAxesFontSize',18)

h = heatmap(matrix);

h.Title = 'Average Point to Line Distance - Pellicle';
```

A.8.4 point_to_line.m

```
function d = point_to_line(pt, v1, v2)
% pt should be nx3
% v1 and v2 are vertices on the line (each 1x3)
% d is a nx1 vector with the orthogonal distances
v1 = repmat(v1,size(pt,1),1);
v2 = repmat(v2,size(pt,1),1);
a = v1 - v2;
b = pt - v2;
d = sqrt(sum(cross(a,b,2).^2,2)) ./ sqrt(sum(a.^2,2));
```

A.8.5 Beam Stability Test "BeamStability.m"

```
% Beam Stability Test - 6/30/2020

% Assesses stability in select projected beam spots

clc
clear all
tic

% import all text files
for i = 1:24
    filenameM = ['BS' sprintf('%1.f',i) '.txt'];
    data(:,:,i) = readmatrix(filenameM);
end

% Extract beam and length scale spot targets only
data_targets = data(50:57,:,:);

% Sort by Z value
data_targets_sorted = zeros(8,7,24); %preallocate zeros
for i = 1:24
    data_targets_sorted(:,:,i) = sortrows(data_targets
       (:,:,i),4);
end

% Take SD of whole target array
data_targets_sorted_sd = std(data_targets_sorted,0,3);

% Assign the points to groups.  A = left, B = Center, C =
   lower, LS =
% Length Scale
% L = left, R = right.. ie: BL = Center beam, leftmost.

AL = data_targets_sorted(6,:,:);
AL = squeeze(AL);
AL = AL';
AL = AL(:,2:7);

AR = data_targets_sorted(4,:,:);
AR = squeeze(AR);
AR = AR';
AR = AR(:,2:7);

BL = data_targets_sorted(5,:,:);
```

```
BL = squeeze(BL);
BL = BL';
BL = BL(:,2:7);

BR = data_targets_sorted(3,:,:);
BR = squeeze(BR);
BR = BR';
BR = BR(:,2:7);

CL = data_targets_sorted(2,:,:);
CL = squeeze(CL);
CL = CL';
CL = CL(:,2:7);

CR = data_targets_sorted(1,:,:);
CR = squeeze(CR);
CR = CR';
CR = CR(:,2:7);

LSL = data_targets_sorted(8,:,:);
LSL = squeeze(LSL);
LSL = LSL';
LSL = LSL(:,2:7);

LSR = data_targets_sorted(7,:,:);
LSR = squeeze(LSR);
LSR = LSR';
LSR = LSR(:,2:7);

%%%%

figure
plot3(BL(:,1),BL(:,2),BL(:,3), 'Linewidth',1.5)
title('BL')
ax = gca;
ax.FontSize = 22;
ylabel('Y (mm)')
xlabel('X (mm)')
zlabel('Z (mm)')
grid on

figure
plot3(BR(:,1),BR(:,2),BR(:,3), 'Linewidth',1.5)
title('BR')
ax = gca;
```

```
ax.FontSize = 22;
ylabel('Y (mm)')
xlabel('X (mm)')
zlabel('Z (mm)')
grid on

toc

% Let's take a look at the distance between the center
   beams.. is it
% random?  Or following some trend?

% Do BL and BR, Center Beams
for i = 1:24
    %Pair up the correct data points
    pair(1,:) = BL(i,1:3);
    pair(2,:) = BR(i,1:3);
    %Take the distance between them
    dist(i,:) = pdist(pair);
end

% Plot this
temp = readmatrix('BeamStabilityTemperature.txt');
trial = linspace(1,24,24);
figure
hold on
yyaxis left
plot(trial,dist,'.-b','Linewidth',1.2,'Markersize',2);
ylabel('Distance (mm)');
ylim([23.0 23.5])
ax = gca;
ax.FontSize = 22;
yyaxis right
ylabel('Temperature (C)');
plot(temp(:,1),temp(:,2),'.-r','Linewidth',1.2,'Markersize
   ',2);
ylim([21.5 22.5])
title('BL BR Separation Distance')
xlabel('Hour');
hold off

% Next, AL and AR.. Left side spots
for i = 1:24
    %Pair up the correct data points
    pair(1,:) = AL(i,1:3);
```

```
    pair(2,:) = AR(i,1:3);
    %Take the distance between them
    dist(i,:) = pdist(pair);
end

% Plot this
% Plot this
temp = readmatrix('BeamStabilityTemperature.txt');
trial = linspace(1,24,24);
figure
hold on
yyaxis left
plot(trial,dist,'.-b','Linewidth',1.2,'Markersize',2);
ylabel('Distance (mm)');

ax = gca;
ax.FontSize = 22;
yyaxis right
ylabel('Temperature (C)');
plot(temp(:,1),temp(:,2),'.-r','Linewidth',1.2,'Markersize
   ',2);
ylim([21.5 22.5])
title('AL AR Separation Distance')

% Last, CL and CR.. Left side spots
for i = 1:24
    %Pair up the correct data points
    pair(1,:) = CL(i,1:3);
    pair(2,:) = CR(i,1:3);

    %Take the distance between them
    dist(i,:) = pdist(pair);
end

% Plot this
temp = readmatrix('BeamStabilityTemperature.txt');
trial = linspace(1,24,24);
figure
hold on
yyaxis left
plot(trial,dist,'.-b','Linewidth',1.2,'Markersize',2);
ylabel('Distance (mm)');

ax = gca;
ax.FontSize = 22;
```

```
yyaxis right
ylabel('Temperature (C)');
plot(temp(:,1),temp(:,2),'.-r','Linewidth',1.2,'Markersize
   ',2);
ylim([21.5 22.5])
title('CL CR Separation Distance')
```