



Graduate Theses, Dissertations, and Problem Reports

2018

Calibration of Cameras and LiDAR for Applications in Autonomous Navigation.

Jacob J. Hikes

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Hikes, Jacob J., "Calibration of Cameras and LiDAR for Applications in Autonomous Navigation." (2018). *Graduate Theses, Dissertations, and Problem Reports*. 8200.
<https://researchrepository.wvu.edu/etd/8200>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

CALIBRATION OF CAMERAS AND LIDAR FOR APPLICATIONS IN AUTONOMOUS NAVIGATION

Jacob J. Hikes

**Thesis submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements for the degree of
Master of Science
in
Aerospace Engineering**

Jason Gross, Ph.D., Committee Chairperson

Yu Gu, Ph.D.

John A. Christian, Ph.D.

Department of Mechanical and Aerospace Engineering

**Morgantown, West Virginia
2018**

Keywords: Camera, LiDAR, Calibration

Copyright © 2018 Hikes

ABSTRACT

Calibration of Cameras and LiDAR for Applications in Autonomous Navigation

Jacob J. Hikes

Autonomous systems have the potential to increase safety and productivity in a wide array of applications. Practical application of autonomous systems requires an accurate navigation solution, which can only be achieved with sufficient mathematical knowledge of the underlying sensor systems. Formulation and estimation of these sensor models is known as calibration.

The following work overviews three classes of optical sensors used in autonomous navigation. Mathematical models are formulated for perspective cameras, omnidirectional cameras, and a system built of a camera and light detection and ranging sensor (LiDAR). Descriptions are provided for perspective camera calibration methods (namely calibration via checkerboard images, goniometric calibration, and calibration via star fields), omnidirectional camera calibration via checkerboard images, and calibration of a LiDAR-fisheye camera system via a planar fiducial array. These calibration methods were then verified experimentally.

Acknowledgments

Since there are far too many to single out, I would like to thank anyone and everyone who has made a positive impact on my life. My family and friends deserve a special thank you for boosting my confidence or humbling my ego when the need arose, for providing the necessary distractions when a reprieve was required, and most of all for always backing my play. I love you all and could not have achieved this without you. A special shout out must also be given to the wonderful educators that have blessed my life. Your guidance and time has done more to help me than words can express. Most of all I would like to thank my Lord and Saviour Jesus Christ for the aforementioned gifts and the many others that have not been discussed. I thank Him for providing the meaning and Salvation that has made my life worth while.

I would also like to acknowledge the funding that has made this work possible: NASA award # NNX13AJ25A, USDA NIFA project # 2017-67022-25926, and the West Virginia Space Grant Consortium Training Grant # NNX15AIO1H.

Contents

Abstract	ii
Acknowledgments	iii
List of Figures	vi
1 Introduction	1
2 Literature Review	3
3 Background	8
3.1 Perspective Cameras	8
3.1.1 Pinhole Camera	8
3.1.2 Brown's Model for Lens Distortion	10
3.2 Omnidirectional Cameras	20
3.2.1 Previous Work	20
3.2.2 The Taylor Model	21
3.3 Camera-LiDAR System	24
4 Technical Approach	25
4.1 Perspective Camera Calibration	25
4.1.1 Checkerboard Calibration	25
4.1.2 Goniometric Calibration	27

4.1.3	Star Field Calibration	32
4.2	Omnidirectional Camera Calibration	35
4.3	Camera-LiDAR Camera Calibration	41
5	Results	46
5.1	Perspective Cameras	46
5.2	Omnidirectional Cameras	58
5.3	Camera-LiDAR System	60
6	Conclusions and Future Work	66
A	Levenberg-Marquardt Routine for Perspective Camera Calibration	1
Appendix		4

List of Figures

3.1	Pinhole Camera	9
3.2	Distortion Effects (Replicated from [1])	11
3.3	2-D Visualization of a Perturbed Wavefront	12
3.4	Radial Symmetry	13
3.5	Decentering Tangential Distortion (Image reproduced from [2])	17
4.1	Sample of a checkerboard image used for calibration.	26
4.2	Depiction of simple collimating lens system (Image reproduced from [3]). .	28
4.3	Example of a compound image generated by an ensemble of collimated point source images. Note that the fiducials have been marked for visibility.	30
4.4	Example of the construction of a star quad using interior star angles (Image reproduced from [4]).	33
4.5	Example of the stellar drift across an image rotated by Earth rate over a one hour period [1].	35
4.6	Example checkerboard image for calibration of a camera with a 180° fisheye lens.	36
4.7	Example image of a calibration board containing reflective fiducials. Image was captured with a downward facing 180° fisheye lens.	42
4.8	Example arrangement of fiducial locations for easy matching between LiDAR points and image centroids.	43

5.1	Distortion map calculated from the first 30 images of the image ensemble [3].	47
5.2	Distortion map calculated from the second half of the image ensemble [3].	48
5.3	Distortion map calculated from the entire 60 images of image ensemble [3].	49
5.4	Distortion map calculated using the first grid [3].	51
5.5	Distortion map calculated using the second grid [3].	52
5.6	Net residuals between estimated image locations and observed image locations [3].	52
5.7	Magnitude of the (u, v) residuals between estimated image locations and observed image locations [3].	53
5.8	Magnified view of an observed point source showing the effects of lens distortion [3].	54
5.9	Magnitude of post-calibration directional residuals across the training images [3].	55
5.10	Magnitude of post-calibration directional residuals across the entire image ensemble [3].	56
5.11	Magnitude of post-calibration residuals across the entire image ensemble [3].	57
5.12	Solution to the <i>Taylor</i> projection function related to the distance from the image center and the angle of the incoming ray.	58
5.13	Estimates of the checkerboard pose in each image.	59
5.14	(a) Reprojection error before corner refinement. (b) Reprojection error after corner refinement.	60
5.15	Depiction of the experimental set up used for this manuscript.	61
5.16	Example ensemble of calibration board images	62
5.17	Comparison of hand measured transformation and the interior point solution. .	63

5.18 Comparison of the angular error (in degrees) between the transformed Li-DAR vectors and their corresponding image vectors using the (a) measured transformation and (b) interior point solution.	63
5.19 Comparison of the angular error (in degrees) between the transformed Li-DAR vectors and their corresponding image vectors using the (a) identity transformation and (b) interior point solution.	64
5.20 Euler angle rotation solutions, in degrees, found via the interior point optimization over 2000 random initializations.	64
5.21 Translation solution elements, in meters, found via the interior point optimization over 2000 random initializations.	65
5.22 Post-calibration angular errors at each fiducial over 2000 random initializations.	65

Acronyms

LiDAR	Light Detection and Ranging
NASA	National Aeronautics and Space Administration
WVU	West Virginia University
SLAM	Simultaneous Localization And Mapping
SIFT	Scale Invariant Feature Transform
OPNAV	Optical Navigation
PnP	Perspective- <i>n</i> -Point

Nomenclature

Symbol	Units	Description
x		Lowercase bold indicates a column vector
X		Uppercase bold indicates a matrix
I		The identity matrix

Chapter 1

Introduction

The interest in autonomous systems continues to grow as modern society progresses. Examples of current research can be found in autonomous vehicles [5], unmanned aerial vehicles [6], unmanned underwater vehicles [7], and many robotic applications such as robotic pollination [8]. To achieve true autonomy, a vehicle must be able to safely navigate within its environment. Among many techniques and sensors to achieve this task are optical systems.

The term optical systems is a broad stroking brush that covers many classes of sensors, but the three of common interest to the navigation community [9, 10, 11] and this manuscript are light detection and ranging (LiDAR) systems, perspective cameras, and omnidirectional cameras. Navigation using optical systems requires that a 3-D scene coordinate, or a line-of-sight to said coordinate, can be related to some observed data point. Thus, accurate knowledge of the geometric projections of these optical systems, gained through calibration of some parameterization of the optics, is critical for successful autonomous navigation. The following manuscript will proceed as follows: Chapter 2 provides a review of a selection of applications of optical systems to further motivate the need for calibration, Chapter 3 provides background and derivations for the state-of-the art in perspective and omnidirectional camera modeling and Chapter 4 provides an explanation of modern cali-

bration techniques. These techniques were experimentally tested, the results of which are reviewed in Chapter 5, and a few conclusions are drawn in Chapter 6.

Chapter 2

Literature Review

A review of the state-of-the-art in the background and camera modeling that would classically be found within the literature review can be found in Chapter 3. For the purpose of this manuscript, I have chosen to use the literature review to discuss branches of research on which this research touches in order to motivate the need for sensor calibration. The following section will overview a few applications; however, there is a wide variety of applications that I will leave to the interested reader, such as: feature matching [12], object detection and tracking [13], visual odometry [9], etc. Note that any observation or claim I make to motivate the need for this manuscript that is independent of the work being discussed in the paragraph will be italicized.

Let us begin the discussion with the most widely relatable application, aesthetics of images. In [14], Zorin and Barr discuss an approach for camera calibration based on visual perception, which is describes a person's ability to perceive planar patterns as reasonable depictions of 3-D scenes. The basic idea is to use desirable visual properties, defined by experimental psychologists [15], as mathematical models and restrictions for image corrections. The authors show that these transformations can be decomposed into perspective projections — projection of the scene onto a plane — and a planar projection, which is the transformation of the planar scene into an image. The authors show that that two most

perceptually important properties, lines appear as lines and spheres appear as circles, cannot be simultaneously satisfied; however, the image can be corrected reduce the deviation from the desired perceptual characteristics.

Another growing application is surgical robotics. In [16], Rosen and Ma discuss the progression of surgical robotics towards fully autonomous operation to reduce the trauma in regions surrounding the surgery sight. The authors' sought to develop an algorithm based on stereo computer vision and dynamic registration. They hope to reach a point where the surgeon is merely a decision maker that dictates the tasks at which the robot is set to complete. Computer vision was used to find the pose of surgical tools and collision detection.

It is here I would like to make the observation that deviation in the observed image from some expected image yields error in the collision detection, thus begging the need for effective modelling of the vision system. They tested their algorithm on a surgical robot test bed and compared the algorithm's performance in a tissue manipulation test against that of a human teleoperating the same robotic system. The test consisted of surgical operations performed in a pegboard environment. Using the computer vision based autonomy, the autonomous operation outperformed the human controlled test bed.

Vision systems are also used in a wide variety of navigation applications. One such application is the optical navigation of spacecraft (OPNAV). In [17], authors Christian and Robinson review a method for horizon-based OPNAV. Horizon-based OPNAV takes advantage of the geometry present using planetary bodies which lack an atmosphere as image targets. An ellipsoidal planetary body with no atmosphere, such as the moon, will appear as an ellipse in an image. Leveraging this relation, an apparent horizon will be related to the true planetary body by a conic. Factorization of the scene into Cholesky factorized space [18] transforms the scene into unit-sphere geometry, thereby allowing for a non-iterative analytical approximation of the camera's range to the target body. The relative range to a planetary body, in conjunction with the attitude retrieved from a star tracker, yields full pose of a spacecraft. *I propose that this motivates the need for accurate camera calibration.*

The potential range between the camera and target body means that small pixel deviations, which implies angular error in perspective geometry, can yield large range approximation errors. Thus OPNAV requires accurate knowledge of the camera system achieved, at the very least, through ground calibration but preferably regularly confirmed in-flight [1].

Visual systems are also useful for robotic navigation in unknown environments. In such a situation, it is desirable to simultaneously develop a map of the environment and localize within it. Simultaneous localization and mapping (SLAM) is well defined with range sensors, but is an active area of research using vision systems [19]. The authors in [19], chose to provide a review of the state-of-the-art within the field of visual SLAM. The authors overview the general step-by-step procedure of visual SLAM techniques; from feature extraction, to image matching, to loop closures. The authors note that accurate image matching relies on accurate estimation of feature locations, which requires a good camera model. They proceed to discuss the use of Kalman filters, structure from motion, and bio-inspired models for the solution to the SLAM problem. The idea of visual SLAM provides a good platform on which to transition from the use of perspective lenses to the implementation of an omnidirectional camera. As discussed in [19], the ability to observe features over a wide field-of-view is advantageous to robotic navigation.

Peter Corke [20] discusses the application of near spherical cameras (catadioptric and dioptric lenses) to image-based visual servoing. He discusses the elimination of the need to consciously keep features in the field-of-view, which is an issue present with the more narrow field-of-view perspective cameras. He continues to derive a spherical image-based visual servoing scheme based on the unified camera model put forth by Geyer and Daniilidis [21]. Corke uses the model to project image points onto a unit sphere. Then, he applies spherical SIFT [22] and structure from motion on the image sphere to recover scene structure. He shows similar results to the literature on the use of perspective cameras in these applications.

Omnidirectional cameras can also be applied to SLAM problem, as shown by the

Caruso, Engel and Cremers in [10]. The authors were able to formulate a direct SLAM approach, which utilizes direct image alignment rather than abstracting the image to set of key points, using omnidirectional cameras. They derived a method to perform depth estimation directly on the camera model based on the equations for the epipolar curves of omnidirectional camera geometry. The authors tested their technique against a rectified image with a perspective model. They showed that the application of omnidirectional cameras to the SLAM problem provides a more robust solution under the drastic rotational movement compared to the implementation of perspective cameras.

Recently, there has been interest in implementing both a camera and a LiDAR for a more dense SLAM solution than just a LiDAR is capable of yielding. One such work [11] motivates the need for for dense data to improve the accuracies of motion estimation, which are unreliable with sparse data sets. The authors proposed a direct visual SLAM system in conjunction with a LiDAR. They make the observation that the direct method is needed to handle the sparse data set from the LiDAR and a low resolution camera (the association of sparse point clouds with visual keypoints would yield few useful features, thereby reducing the visual odometry performance). The method proposed only implements visual SLAM when corresponding depth data is available and scraps visual features from the motion estimation process. Additionally, they perform sliding window pose estimation to minimize local drift and pose graph optimization for the global level. They evaluated their algorithm using KITTI datasets [23] and an independent experiment with an 8-ray scanning LiDAR. They showed that their technique was able to produce robust, accurate SLAM results using sparse depth measurements. *Accuracy of this technique is inherently dependent on knowledge of the rigid body transform between the LiDAR and imager.*

Pandey, *et. al.* [24] introduces a method for automatic estimation of the the rigid body transform by maximizing the mutual information between intensity and reflectance data from an image and LiDAR scan, respectively. Their method is a data driven technique that only requires sensors that provide some type of meaningful intensity data. They claim

that the correlation between laser reflectivity and image intensities is maximized at a specific rigid-body transform. The authors show that solving for this transform computes the *extrinsic calibration* between the two sensors.

Another algorithm for *extrinsic calibration* of a LiDAR-camera system was proposed in [25]. The authors of this work extended to any camera what had only been well documented for perspective cameras. The algorithm does not require the use of any specific calibration target, but rather a set of user defined point correspondences captured from a natural, static scene. They convert the depth data into what they annotate as bearing angle image. The data is first re-arranged into a matrix, and then define the bearing angle to be the angle between a directional scan (horizontal, vertical, and diagonal) and the segment connecting two data points. The well documented perspective- n -points (PnP) problem [26] is then solved and the solution is used in the motion estimation problem [27] to obtain the full rigid body transform.

As industrial and public use robotics become more prevalent, accurate navigation and obstacle avoidance is becoming a greater safety necessity. Optical sensors are widely applicable for a variety of navigation applications; however, to meet the accuracy requirements for safe and practical application, thorough knowledge of the geometry of a sensor system must be obtained. Proper calibration of optical systems provides the mathematical parameterization needed for accurate navigation.

Chapter 3

Background

3.1 Perspective Cameras

Perspective cameras have been researched and modelled by photogrammetrists since the early twentieth century [28]. Since that time, there have been a variety of camera models developed [29, 30, 31]. However, one such parameterization [2] has become the ubiquitous perspective camera model in the computer vision, photogrammetry and optical navigation communities. The following section overviews a full derivation of the *Brown Model*, starting from the original ideas put forth by Conrady [28] derived through the idea of wave fronts [32] and then finalized by Brown [2].

3.1.1 Pinhole Camera

Consider a camera constructed with a single thin lens separated from a focal plane by a distance f . If the lens were reduced to a point (or a pinhole), then only a single ray of light from some object point will pass through the lens. Additionally, the incoming light will not be affected by any refraction.

The image formed on the focal plane will be an exact replica of the object, but will be flipped about the center point of the image. In order to avoid the added computational

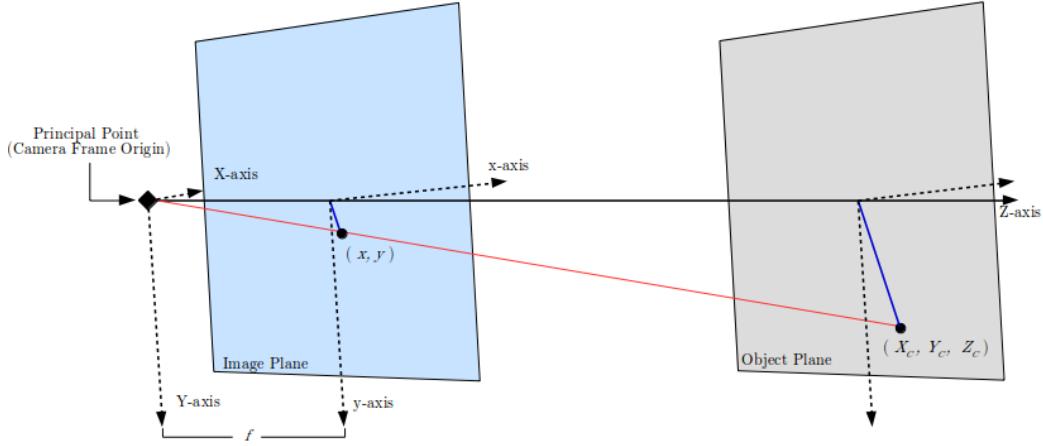


Figure 3.1: Pinhole Camera

considerations of a flipped image we will continue the following derivation using the image plane, which is the plane a distance f in front of the lens that is parallel to the focal plane. The image plane generates a dimensionally identical image to the focal plane but does not require any negation [1]. See Figure 3.1.

Clearly, the pinhole camera model is straightforward to establish through similar triangles. It is governed by Eq. 3.1.

$$\frac{x_i}{f} = \frac{X_{C_i}}{Z_{C_i}} \quad \frac{y_i}{f} = \frac{Y_{C_i}}{Z_{C_i}} \quad (3.1)$$

where (x_i, y_i) is the coordinate of the i^{th} observed point on the image plane and $(X_{C_i}, Y_{C_i}, Z_{C_i})$ is the coordinate of the i^{th} observed point in the 3-D space defined by the camera reference frame [1]. Note if the full 3-D location of the object point is not known, a line-of-sight to the point is sufficient information and can be used as a substitute for $(X_{C_i}, Y_{C_i}, Z_{C_i})$.

Assuming the use of a digital imager, image plane coordinates must now be converted to pixel coordinates to generate a physical image. The conversion from image plane to

focal plane array is performed via the *camera intrinsic matrix*, denoted by \mathbf{K} in Eq. 3.2 [1].

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} d_x & \alpha & u_p \\ 0 & d_y & v_p \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i/f \\ y_i/f \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} x_i/f \\ y_i/f \\ 1 \end{bmatrix} \quad (3.2)$$

The principal point, (u_p, v_p) , is the intersection of the focal plane array and the optical axis of the lens system. Thus, it is simply the origin of the image plane as expressed in the pixel coordinate system. The skew factor, α , accounts for non-orthogonality in the structure of the focal plane array. The scale of the individual pixels' height and width are parameterized in d_x and d_y , where $d_x = d_y$ if the pixels are square.

3.1.2 Brown's Model for Lens Distortion

Image distortion is defined as the deviation in the observed image from the ideal perspective projection as described by the pinhole camera model [33]. Such image distortion is primarily introduced in two ways: optical aberrations and manufacturing defects. The first leads to radial distortion and the latter leads to decentering distortion. Radial distortion is caused by non-uniform magnification of the lens system. The result is a radial perturbation of the image point from the ideal. If the image is magnified greater in its center than its periphery, we call this barrel distortion. Pincushion distortion, on the other hand, is the converse; magnification of the image increases with the distance from the principle point [1].

In a perfectly built camera, the ray orthogonal to the detector plane coincides with the optical axis of each lens of the lens system. Any misalignment of the system components, either angular or translational, will result in a decentering distortion. Depending on the form of the misalignment, decentering can produce radial distortion, tangential distortion (tip/tilt), or a combination of distortions to the overall distortion of the image [2]. See Figure 3.2.

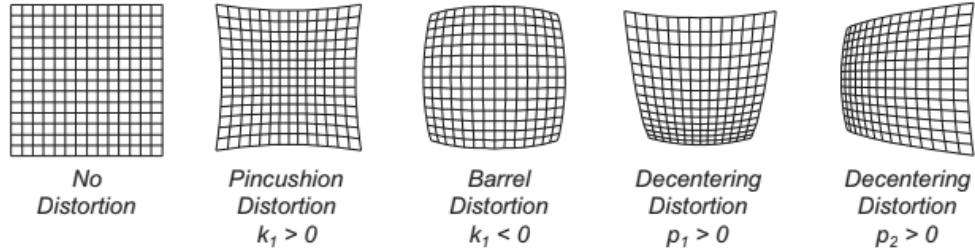


Figure 3.2: Distortion Effects (Replicated from [1])

The net distortion of an image is simply the sum of these two effects. Thus, we have a distortion model governed by:

$$\Delta_{net} = \Delta + \delta \quad (3.3)$$

where Δ and δ are the respective radial distortion and decentering distortion components.

Radial Distortion

We begin the derivation of radial distortion with the concept of wavefronts. A wavefront is defined as the locus of points propagating at the same relative phase. For our purposes of modelling a camera, a wavefront is the curve of a constant optical path measured from an object point, O . The optical path is determined by the product of the Euclidean distance the light must travel and the local refractive index. Wavefronts are generally orthogonal to the outgoing rays; thus, if we consider all rays emitted from the object, assuming a surrounding environment of constant refractive index, then wavefronts will form concentric spheres centered at O [32].

In an ideal scenario, with no optical aberrations, wavefronts corresponding to the image plane are again concentric spheres but centered at the image point. This is not the case in practice. We will consider the wave aberration to be the perturbation of a wavefront from the ideal reference sphere [32]. See Figure 3.4

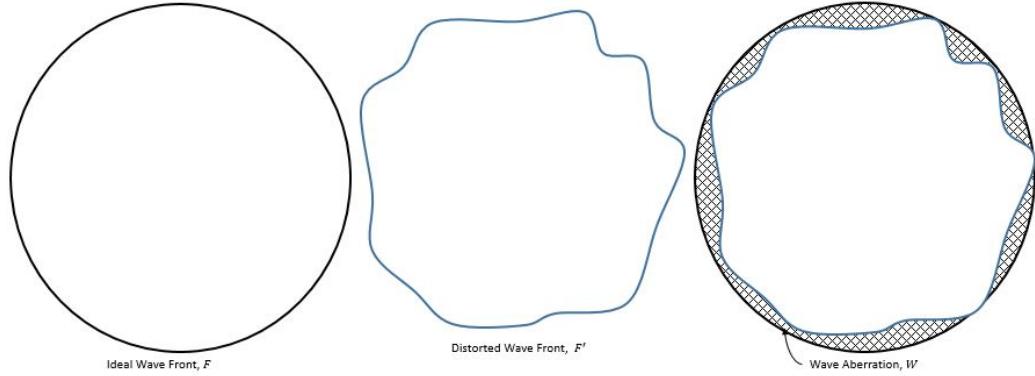


Figure 3.3: 2-D Visualization of a Perturbed Wavefront

Define F as the ideal wave front function and F' as the distorted wave front function. The resultant wave aberration function is given by $W = F' - F$.

If we fix the object point, the wave aberration function is clearly a function of just the image coordinates, (\hat{x}, \hat{y}) . In general, however, we are concerned with the aberration of a wavefront originating from any arbitrary point, (x, y) , on the object plane. Thus, the wave aberration is a function of four variables $W(x, y, \hat{x}, \hat{y})$ [32].

It is desirable to reduce the number of unknowns within the wave aberration. Assume rotational symmetry about the optical axis. Then, let \mathbf{r} be the vector from the origin of the object plane to (x, y) , $\hat{\mathbf{r}}$ be the vector from the principle point to (\hat{x}, \hat{y}) , and ϕ be the angle between them (Figure 3.4). Thus, the rotationally symmetric variable exchange is given by [32]:

$$\mathbf{r} \cdot \mathbf{r} = r^2 = x^2 + y^2$$

$$\hat{\mathbf{r}} \cdot \hat{\mathbf{r}} = \hat{r}^2 = \hat{x}^2 + \hat{y}^2$$

$$\mathbf{r} \cdot \hat{\mathbf{r}} = \|\mathbf{r}\| \|\hat{\mathbf{r}}\| \cos \phi = x\hat{x} + y\hat{y}$$

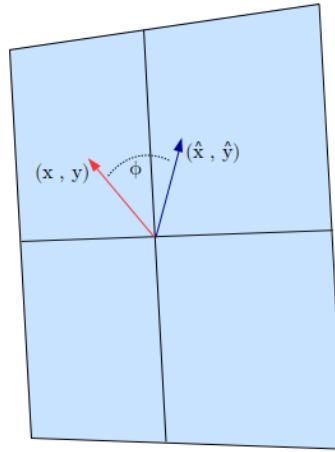


Figure 3.4: Radial Symmetry

Considering the new variables, write the wave aberration function as

$$W \equiv W(x, y, \hat{x}, \hat{y}) \equiv W(\mathbf{r} \cdot \mathbf{r}, \hat{\mathbf{r}} \cdot \hat{\mathbf{r}}, \mathbf{r} \cdot \hat{\mathbf{r}}) \equiv W(x^2 + y^2, \hat{x}^2 + \hat{y}^2, x\hat{x} + y\hat{y}) \quad (3.4)$$

Note that the wave aberration is now only dependent on the magnitudes of the object and image vectors, and the angle between them.

Now, let us define the following summations:

$$S = \sum_{i=1}^{\infty} c_i (\hat{x}^2 + \hat{y}^2)^{i+1} \quad (3.5a)$$

$$C = \sum_{i=1}^{\infty} d_i (\hat{x}^2 + \hat{y}^2)^i (\hat{x}\hat{x} + \hat{y}\hat{y}) \quad (3.5b)$$

$$A = \sum_{i=1}^{\infty} \ell_i (x^2 + y^2)^{i-1} (\hat{x}\hat{x} + \hat{y}\hat{y})^2 \quad (3.5c)$$

$$F = \sum_{i=1}^{\infty} m_i (x^2 + y^2)^i (\hat{x}^2 + \hat{y}^2) \quad (3.5d)$$

$$D = \sum_{i=1}^{\infty} k_i (x^2 + y^2)^i (\hat{x}\hat{x} + \hat{y}\hat{y}) \quad (3.5e)$$

If we consider just the first terms of each summand, we will see that these represent the standard five Seidel sums, or *monochromatic aberrations* [34]. The series represent spherical aberration, coma, astigmatism, curvature of field and distortion, respectively. The higher order terms of the series each contribute solely to their respective aberration; however, when higher order terms (*chromatic aberrations*) are considered it must be noted that there is also contributing effects from a convolution of the standard aberrations to the total aberration of an image [34, 35, 32].

We will now expand the wave aberration function, Eq. (3.4), as a power series. If the terms are rearranged properly, this will yield [32]

$$\begin{aligned} W(x^2 + y^2, \hat{x}^2 + \hat{y}^2, x\hat{x} + y\hat{y}) &= a_0 + b_1(\hat{x}^2 + \hat{y}^2) + b_2(x\hat{x} + y\hat{y}) \\ &\quad + S + C + A + F + D + \dots \text{other terms} \end{aligned} \quad (3.6)$$

where the *other terms* are the convolution terms and terms dependent only on the object coordinates. We neglect terms dependent only on the object coordinates because the wave aberration function is defined to vanish at the principal point $(\hat{x}, \hat{y}) = (0, 0)$. Consequently, any term that has no dependence on the image coordinates must be defined as

zero. We define a_0 and all constants of the *other terms* dependent on the object coordinates to be zero [32].

Although we would assume the linear terms, b_1 and b_2 , are the most significant contributors to the wave aberration, these terms may also be neglected. The b_1 and b_2 terms result from a choice of the reference point on the image plane and from the tilt of the reference sphere, respectively [32, 36]. Thus, both terms may be corrected by proper image focusing and choice of magnification. For a more thorough reasoning of this conclusion, I direct the reader to [32] and [35]. The wave aberration function can now be described by:

$$W(x^2 + y^2, \hat{x}^2 + \hat{y}^2, x\hat{x} + y\hat{y}) = S + C + A + F + D \quad (3.7)$$

When developing a practical distortion map, we are concerned with the transverse ray aberrations in the camera reference frame rather than the wave aberration. According to [32], the ray aberrations may be well approximated by differentiating the wave aberration function by the respective image coordinate. Ergo, we have ray aberrations given by [32]

$$\Delta_x = \frac{\partial W}{\partial \hat{x}} \quad (3.8)$$

$$\Delta_y = \frac{\partial W}{\partial \hat{y}} \quad (3.9)$$

Note Gross [32] includes a constant, $-\frac{R}{n'}$, that we absorb into the constants of the series.

Recall that the Seidel sums are defined by the third order ray aberrations. After differentiating we see that, assuming a proper choice of constants, the first terms of Eqs. (3.5) are equivalent to the five Seidel sums [34].

Radial distortion displaces the image point without a loss of focus, whereas the other four aberrations are a blurring of the image that do not necessarily displace the image point [34, 35]. Thus, for the purpose of navigation and reconstruction, we are only concerned about the radial component. By neglecting the non-radial terms, we arrive at our final

model of the radial distortion given by:

$$\Delta_x = \frac{\partial D}{\partial \hat{x}} = [k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots] x \quad (3.10a)$$

$$\Delta_y = \frac{\partial D}{\partial \hat{y}} = [k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots] y \quad (3.10b)$$

where $r^2 = (x^2 + y^2)$. This model has become the standard model of radial distortion in the imaging community (*e.g.* [30, 2, 33]).

Decentered Distortion

A lens system is considered decentered if the optical axis does not intersect the center of each component, or if any component of the system is not orthogonal to the camera boresight. The latter can also be considered as a requirement that all components be parallel to the detector array. All practical cameras will suffer from some degree of misalignment [1].

An image taken by a decentered system will have an axis of maximum tangential distortion, M . At a constant radial distance from the boresight, the tangential distortion is proportional to the maximum tangential distortion at that radius [28, 2]. See Figure 3.5.

We can see from the figure that $\delta_t = P \cos(\theta - \theta_0)$. Define P as the maximum tangential distortion at some radius, given by [2].

$$P \equiv P(r) = J_1 r^2 + h. o. t. \text{ in powers of } r^2 \quad (3.11)$$

As shown by Conrady [28], the corresponding radial and tangential components of decentered distortion are:

$$\delta_t = P \cos(\theta - \theta_0) \quad (3.12)$$

$$\delta_r = 3P \sin(\theta - \theta_0) \quad (3.13)$$

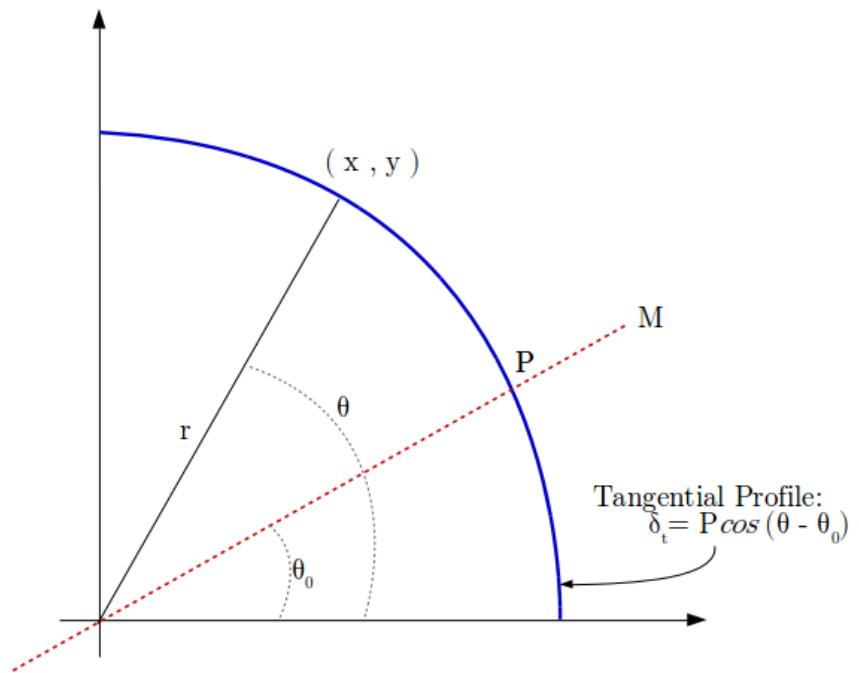


Figure 3.5: Decentering Tangential Distortion (Image reproduced from [2])

As in the case of radial distortion, we are concerned with ray aberrations in the camera coordinate frame. Thus, make a frame transformation provided by [2]

$$\delta_x = \delta_r \cos \theta - \delta_t \sin \theta \quad (3.14)$$

$$\delta_y = \delta_r \sin \theta + \delta_t \cos \theta \quad (3.15)$$

Now, substitute Eqs. (3.12) and (3.13) into Eqs. (3.14) and (3.15). Details of the algebra will be left to the reader; however, after the proper application of trigonometric identities provide a more concise model of δ_x and δ_y given by [2],

$$\delta_x = -P[(1 + 2\frac{x^2}{r^2}) \sin \theta_0 - 2\frac{xy}{r^2} \cos \theta_0] \quad (3.16)$$

$$\delta_y = -P[2\frac{xy}{r^2} \sin \theta_0 - (1 + 2\frac{y^2}{r^2} \cos \theta_0)] \quad (3.17)$$

If a more comprehensive model is sought, one must consider higher order terms in the profile function P . Conrady, however, does not derive any higher order effects in the construction of his model. According to [2], higher order effects can be reasonably modelled by adding higher powers of r^2 . Thus, the profile function can be defined as

$$P = J_1 r^2 + J_2 r^4 + J_3 r^6 + \dots \quad (3.18)$$

Substitute Eq. (3.18) into Eqs. (3.16) and (3.17) to find the higher order model [2]:

$$\delta_x = -[J_1 r^2 + J_2 r^4 + J_3 r^6 + \dots][\frac{(r^2 + 2x^2)}{r^2} \sin \theta_0 - \frac{2xy}{r^2} \cos \theta_0] \quad (3.19)$$

$$\delta_y = -[J_1 r^2 + J_2 r^4 + J_3 r^6 + \dots][\frac{2xy}{r^2} \sin \theta_0 - \frac{(r^2 + 2y^2)}{r^2} \cos \theta_0] \quad (3.20)$$

For the purpose of readability and simplification, let us now make a change of variable

given by:

$$\begin{aligned}
p_3 &= \frac{J_2}{J_1} \\
p_1 &= -J_1 \cos \theta_0 & p_4 &= \frac{J_3}{J_1} \\
p_2 &= -J_1 \sin \theta_0 & p_5 &= \frac{J_4}{J_1} \\
&&&\vdots
\end{aligned}$$

Factor out a $-J_1 r^2$ and apply these substitutions to Eqs. (3.19) and (3.20) to arrive at the final model of decentering distortion as given by [2]:

$$\delta_x = [1 + p_3 r^2 + p_4 r^4 + p_5 r^6 + \dots][(r^2 + 2x^2)p_1 + 2p_2 xy] \quad (3.21)$$

$$\delta_y = [1 + p_3 r^2 + p_4 r^4 + p_5 r^6 + \dots][2xyp_1 + (r^2 + 2y^2)p_2] \quad (3.22)$$

Now that both the radial and decentering components of distortion have been derived, the model describing the net distortion of an image can finally be written as [2]:

$$\begin{bmatrix} \Psi_{net_x} \\ \Psi_{net_y} \end{bmatrix} = (k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \begin{bmatrix} x \\ y \end{bmatrix} + (1 + p_3 r^2 + p_4 r^4 + p_5 r^6 + \dots) \begin{bmatrix} (r^2 + 2x^2)p_1 + 2p_2 xy \\ 2xyp_1 + (r^2 + 2y^2)p_2 \end{bmatrix} \quad (3.23)$$

The distortion of an image is merely the displacement of the image point from the ideal, $(\hat{x} - x, \hat{y} - y)$. Ergo, the distorted image coordinates can be found from [2]

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \begin{bmatrix} x \\ y \end{bmatrix} + (1 + p_3 r^2 + p_4 r^4 + p_5 r^6 + \dots) \begin{bmatrix} (r^2 + 2x^2)p_1 + 2p_2 xy \\ 2xyp_1 + (r^2 + 2y^2)p_2 \end{bmatrix} \quad (3.24)$$

Note that any distortion effects on incoming light are modeled on the image plane. Consequently, the image points found using Eq. 3.24 must be converted to pixel coordinates using Eq. 3.2 to generate a real distorted image.

3.2 Omnidirectional Cameras

Before I begin a description of the omnidirectional camera model, allow me to note that this section is based on the model developed by Davide Scaramuzza [37]. As such, unless otherwise cited, all claims made in the following section are taken directly from [37] or are summaries of my understanding of the work done by [37].

3.2.1 Previous Work

The perspective projection model discussed in the previous section holds true for cases in which the mapping from the 3-D object space into the image plane is linear. In other words, perspective projection describes images that are created by intersecting a plane with the set of lines penetrating the scene and coinciding at the projection center. This projection does not hold in camera systems with high distortions introduced by a wide field-of-view. Such cameras can be generalized to two constructions: a combination of mirrors and lenses (*catadioptric*) and extreme wide angle lenses (*dioptric* or fisheye). Depictions of these two lens classes can be seen in [37] by the interested reader.

Cameras of these structures can be considered a central single viewpoint projection. This is ideal. Due to the single viewpoint constraint, every pixel in the camera's CCD array corresponds to one particular direction. Under perspective projection, a pixel geometrically relates to two different directions: the ray pointing in front of the camera, and the ray pointing behind it; however, due to the physical constraints of the system (limited field-of-view) the rear pointing ray can be neglected. Considering a camera with a field-of-view of 180° or greater, we see that these physical constraints no longer uniformly hold, thereby motivating the need for a new projection model.

The original concept of a central omnidirectional camera was put forth in 1637 by René Descartes [38]. He showed that conical lenses and mirrors focus light into a single point if the illumination direction is properly chosen. This idea was expanded upon and popularized

by Baker and Nayar [39, 40, 41]. They model a set of specific classes of catadioptric cameras which are constructed using a single lens and a single mirror to create the single viewpoint geometry. They presented four configurations to create a central omnidirectional camera: 1) a paraboloidal mirror coupled with an orthographic camera [42], a perspective camera coupled with 2) a hyperboloidal mirror, 3) an ellipsoidal mirror, and 4) a planar mirror.

Then, in 2000, Geyer and Daniilidis [21] were the first to propose a theory for all central catadioptric cameras. Their theory proved that all central catadioptric camera systems projections are isomorphic to the projective mapping of a sphere, which is centered at the viewpoint, to a plane such that the projection center lies on the intersection of the plane and the normal to the plane that intersects the viewpoint. The theory of Geyer and Daniilidis was extended to certain dioptric lenses by Ying and Hu in 2004 [43].

A unified model that represents all central cameras (catadioptric, dioptric, and perspective) was finally developed by Micusik [44]. He formalized a method to describe the projection of the 3-D world onto the image using two separate projections. Unfortunately, the definitions of the two projections are dependent upon the specific types of lenses and/or mirrors used in the imager. Essentially, the first projection defines the profile of the lens or mirror and the second projection defines a perspective projection of the pseudo-image plane generated from the first projection into a true image.

3.2.2 The Taylor Model

This idea of the double reference system persists in the *Taylor Model* developed by Scaramuzza; however, Scaramuzza's formulation does not require any knowledge on the construction of the imager. The first reference system defines what will be denoted as the sensor plane. This plane is the hypothetical plane that is centered at the camera's optical center \mathbf{O}_c and is orthogonal to the mirror or dioptric lens axis (the pseudo-image plane mentioned in the review of Micusik's work). The second reference system is the image plane,

which is similar to the generated image in a perspective projection in that it coincides with the focal plane array and has coordinates expressed in pixels.

Recalling the discussion of perspective cameras, manufacturing alignment errors cause the image plane of a real camera to be, to a small degree, non-orthogonal and off-center of the lens/mirror system's optical axis. Additionally, the discretization of light by the CCD array must be considered by some *camera intrinsic matrix*. The rotation of the image plane to the ideal \mathbf{R}_c can be combined with the *intrinsic matrix* \mathbf{K}_c into a homography $\mathbf{H}_c = \mathbf{K}_c \mathbf{R}_c$ that transforms the image plane to the sensor plane.

Rather than begin the definition of the central camera model with the 3-D object coordinate as was the case with a perspective camera model, we will first consider the observed image. Let the 2×1 vector \mathbf{u}_i describe the pixel location of the i^{th} point in an image. Then, the homography \mathbf{H}_c can be well approximated by the affine transformation in Eq. 3.25 when the manufacturing errors are small.

$$\mathbf{x}_i = \mathbf{A}\mathbf{u}_i + \mathbf{t} \quad (3.25)$$

such that $\mathbf{x}_i = [x, y]^T$ is the i^{th} coordinate in the sensor plane and

$$A = \begin{bmatrix} c & d \\ e & 1 \end{bmatrix} \quad t = \begin{bmatrix} u_p \\ v_p \end{bmatrix}$$

such that $[u_p, v_p]^T$ is the center of the image.

Scaramuzza now defines a projection function g , which is used to transform the sensor plane into 3-D coordinates, as Eq. 3.26

$$g(\rho) = a_0 + a_1\rho + a_2\rho^2 + \dots + a_n\rho^n \quad (3.26)$$

where

$$\rho = (x_i^2 + y_i^2)^{\frac{1}{2}}$$

and the degree on the polynomial n is dictated by the user to determine the number of calibration parameters a_0, a_1, \dots, a_n . There is an additional constraint on the projection such that the derivative of g must fade at the origin of the sensor plane (Eq. 3.27).

$$\frac{dg}{d\rho} \Big|_{\rho=0} = 0 \quad (3.27)$$

Consequently, $a_1 = 0$ and Eq. 3.26 can be reduced to n parameters as expressed in Eq. 3.28.

$$g(\rho) = a_0 + a_2\rho^2 + \dots + a_n\rho^n \quad (3.28)$$

The projection g can be used to project the sensor plane into 3-D object space by Eq. 3.29.

$$\lambda \mathbf{p}_i = \lambda \begin{bmatrix} \mathbf{x}_i \\ g(\rho) \end{bmatrix} = \begin{bmatrix} X_{C_i} \\ Y_{C_i} \\ Z_{C_i} \end{bmatrix} \quad (3.29)$$

such that \mathbf{p}_i is the vector pointing to the i^{th} world coordinate as expressed in the camera frame $[X_{C_i}, Y_{C_i}, Z_{C_i}]^T$ with a scaling λ .

Recall that an ideal central projection can be considered as a projection of the unit sphere onto the sensor plane. Consider the normalization of \mathbf{p}_i in Eq. 3.30.

$$\mathbf{q}_i = \frac{\mathbf{p}_i}{\|\mathbf{p}_i\|} \quad (3.30)$$

We see that, with the application of Eq. 3.29, this relates the unit sphere to the sensor plane. Ideally, this projection to the image plane will produce a circular image such that lines in the scene appear as circular geodesics in the image; however, this is not realizable in a physical system. The misalignments and manufacturing errors will create an elliptical projection into the image plane. Thus, \mathbf{A} can be thought of as the stretch matrix that contorts the ideal circle into the observed ellipse. In a well built system, \mathbf{A} should be near identity.

3.3 Camera-LiDAR System

Use of LiDAR systems is prevalent in the autonomous navigation community, especially for simultaneous localization and mapping applications. These sensors, such as Velodyne® scanning LiDAR systems, are rigorously factory calibrated and as such, calibration of the sensor by the consumer is unnecessary. The interested reader is directed to [37] or [45] for descriptions of LiDAR models and calibration methods.

It is often desirable to improve the density of a LiDAR scan with image data, or conversely to improve depth estimates achieved from image data with supplemental LiDAR information [46]. This can only be achieved if accurate knowledge of the rigid body transformation between the visual sensor and the LiDAR is known. Such a transformation is given by Eq. 3.31,

$$\mathbf{e}_{\ell_C} = s (\mathbf{R} \mathbf{v}_\ell + \mathbf{t}) \quad (3.31)$$

where \mathbf{v}_ℓ is a 3-D measurement in the LiDAR frame, \mathbf{R} and \mathbf{t} are the respective rotation and translation from the LiDAR frame to the camera frame, $s = \frac{1}{\|\mathbf{R} \mathbf{v}_\ell + \mathbf{t}\|}$ is the normalization of the rotated LiDAR vector, and \mathbf{e}_{ℓ_C} is the corresponding unit line-of-sight in the camera frame.

Chapter 4

Technical Approach

4.1 Perspective Camera Calibration

There are a variety of techniques used to approximate the parameters of a perspective camera model [47, 48, 49, 50]. The basic premise in all of these techniques is to take an image of a target, or targets, that allows for simple fiducial (corners, edges, bright points, etc) location. The target is also commonly structured to provide uniform field-of-view coverage and to permit for intuitive calculation of ideal pinhole locations.

Using the observed image fiducials in conjunction with approximate ideal pinhole locations, the parameters of a camera model can be estimated via an appropriate optimizer, such as the Levenberg-Marquardt algorithm (LMA) [51, 52]. The following sections will overview three techniques used to calibrate a perspective camera: calibration via checkerboards, goniometric calibration, calibration via star field images.

4.1.1 Checkerboard Calibration

Calibration of a perspective camera using images of a planar checkerboard captured at varying angles — an algorithm originally proposed in [53] — has become a ubiquitous technique in the computer vision community [54, 55, 9, 56, 57]. The work presented in

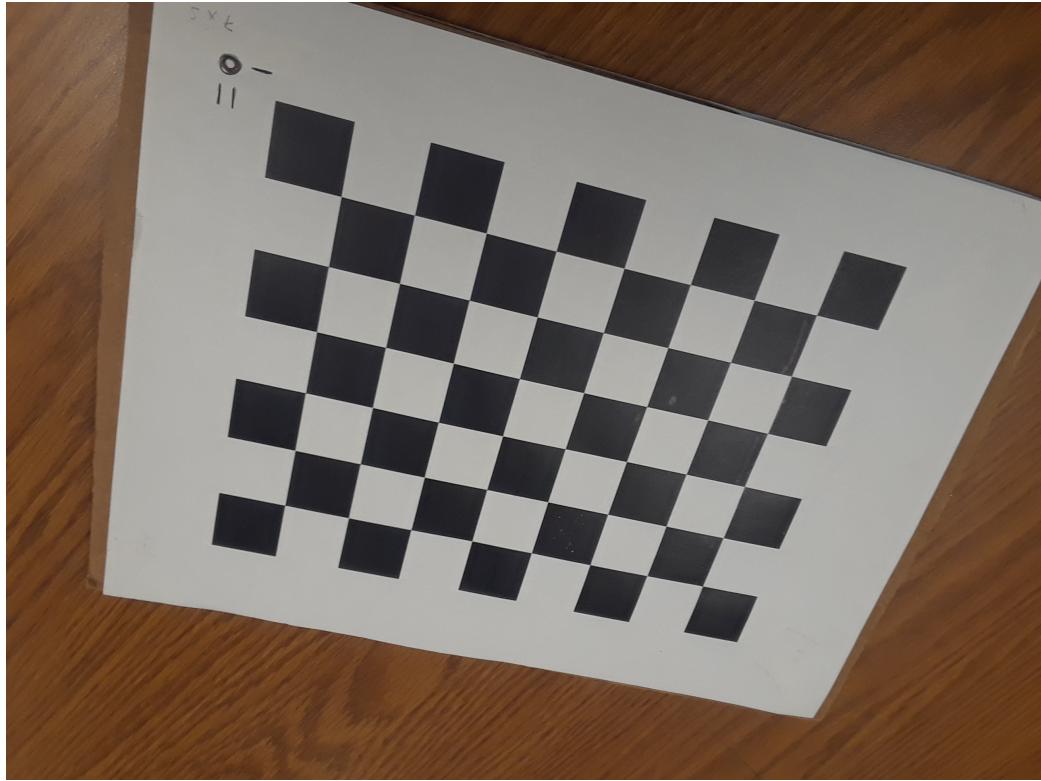


Figure 4.1: Sample of a checkerboard image used for calibration.

this thesis utilizes one such method that was implemented into a MATLAB[®] toolbox and OpenCV implementation by Bouguet. This toolbox is very well documented and the interested reader is directed to [47] for access to the toolbox, background documentation, and procedural examples. For completeness, the following section provides a brief overview of the toolbox.

The first step to using Bouguet's toolbox is to capture a series of checkerboard images, such as that in Figure 4.1. The goal of this process is to capture images of the board at a variety of different angles in such a way that the entire field-of-view has relatively uniform coverage over the course of the image ensemble, and relatively uniform exposure between images. Relatively uniform field-of-view coverage is necessary for the model to account for distortions in the entirety of the image, rather than yield a bias towards some direction. Similar exposure from one image to another within the ensemble ensures that the edge detector and corner extraction procedures perform comparably. The number of images

needed to obtain proper coverage is dependent upon the board size, the camera’s field-of-view, and the focal distance; however, the recommended minimum is 10-15 images.

This image ensemble is then loaded using a graphical user interface, and the exterior corners of each checkerboard are selected by the user. Using the selected corners and user provided checker dimensions, an autonomous corner extraction algorithm then counts the number of squares within the grid, and locates the remaining interior corners by estimating a planar homography. In the case of a lens with high distortion effects, an initial guess of the *Brown Model* parameters may be required to estimate the checkerboard corners.

Employing Zhang’s method [53], the observed corners and checkerboard dimensions are used to estimate both the 6 DOF rotation-translation of the board plane relative to the camera frame in each image and the 9 DOF camera model —the camera intrinsic matrix and the *Brown Model* parameters. Note that the default of the Bouguet algorithm ignores the pixel array skew factor α , which is a common convention in modern calibration techniques due to the negligible non-orthogonality of contemporary focal plane arrays.

4.1.2 Goniometric Calibration

While sufficient for many computer vision processes [9, 57, 19], calibration techniques using checkerboards at unknown poses do not guarantee sub-pixel accuracies (see Chapter 5). The simultaneous estimation of unknown 3-D world coordinates, which are dependent on the estimated board pose, and camera parameters inflict a degree of unreliability on the checkerboard method [58]. Techniques using known target coordinates, such as goniometric calibration [59], can yield higher reliability; though they generally require a more elaborate setup of expensive equipment [58].

Goniometric camera calibration is performed using a camera mounted to a 2-axis gimbal to rotate a camera through a series of images at a collimated light source target. A collimator consists of a light source, aperture and lens system that refracts the light source in such a way that a beam of parallel light is produced. A simplistic example can be seen

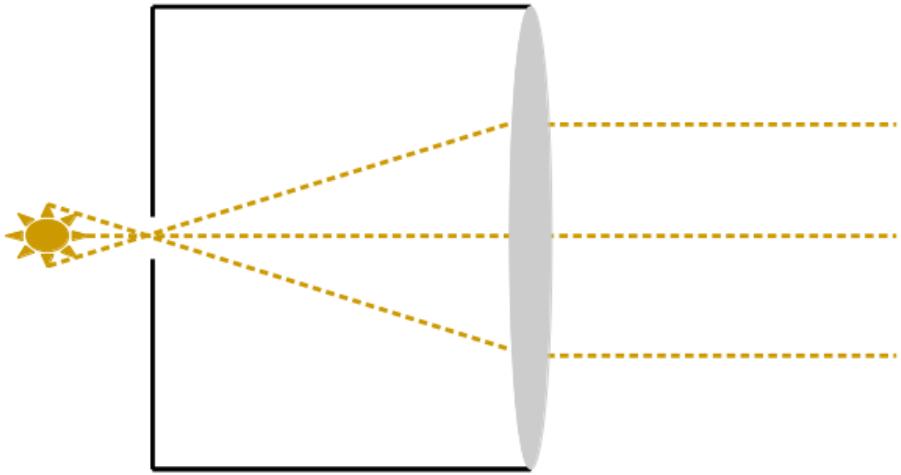


Figure 4.2: Depiction of simple collimating lens system (Image reproduced from [3]).

in Figure 4.2.

A collimated light source simulates an infinite point source, such as a star. Thus, goniometric calibration techniques allow for a camera to be calibrated at a wide range of focal lengths, which makes it a common technique for aerial imagery camera [60]. Precision calibration can be achieved by mounting a camera onto high precision 2-axis gimbal. The camera is then rotated through an arrangement of viewpoints, capturing an image at each viewpoint, which is commonly some uniform grid covering the camera’s field-of-view. For the purpose of this research, the camera is assumed to be mounted such that its optical center coincides with the gimbal’s center of rotation and the camera and gimbal frames align.

The most common technique to capture images for calibration is to rotate the pan/tilt such that the series of illuminated image points lie on a regular grid [59]. A regular grid is achieved by defining uniformly spaced azimuth γ and elevation ϕ angles. The gimbal is

rotated through each of these angles and an image is captured at each interval. Thus for each image, the true line-of-sight to the light source is defined in Eq. 4.1.

$$\mathbf{l}_k = \begin{bmatrix} \frac{\tan \gamma_i}{\cos \phi_j} \\ \tan \phi_j \\ 1 \end{bmatrix}$$

$$\mathbf{e}_k = \frac{\mathbf{l}_k}{\|\mathbf{l}_k\|} \quad (4.1)$$

Note that $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, M$ generates an $M \times N$ array of fiducials. The image ensemble is then compiled to create a single image with an array of fiducial point sources, an example of which is presented in Figure 4.3. This is done to alleviate computational complexity when the calibration algorithm is implemented, as only one image will need to be processed rather than an ensemble of $M \cdot N$ images.

Since each image is of a single point source captured at different pointing directions, the compiled image can be generated by adding intensities across all images. However, a CCD array experiences fixed pattern noise, a random intensity biases even on unilluminated pixels, known as the *dark frame*. These biases must be accounted for prior to generating the compiled image or the effects of the noise are compounded.

A dark frame can be calculated in several manners. If the camera has a mechanical shutter, the image created by capturing with a closed shutter is the dark frame of the camera. Another technique, which is the technique employed in this research, is to approximate the dark frame by calculating the median of a series of evenly spaced images within the point source ensemble. Thus, for each image, a dark frame correction is performed as

$$\mathcal{I}_i = \mathcal{I}_i^o - \mathcal{D} \quad (4.2)$$

where \mathcal{I}_i^o is the i^{th} image of the ensemble, \mathcal{D} is the estimated dark frame of the camera, and \mathcal{I}_i is the i^{th} corrected image. The compiled image \mathcal{C} can now be generated by summing

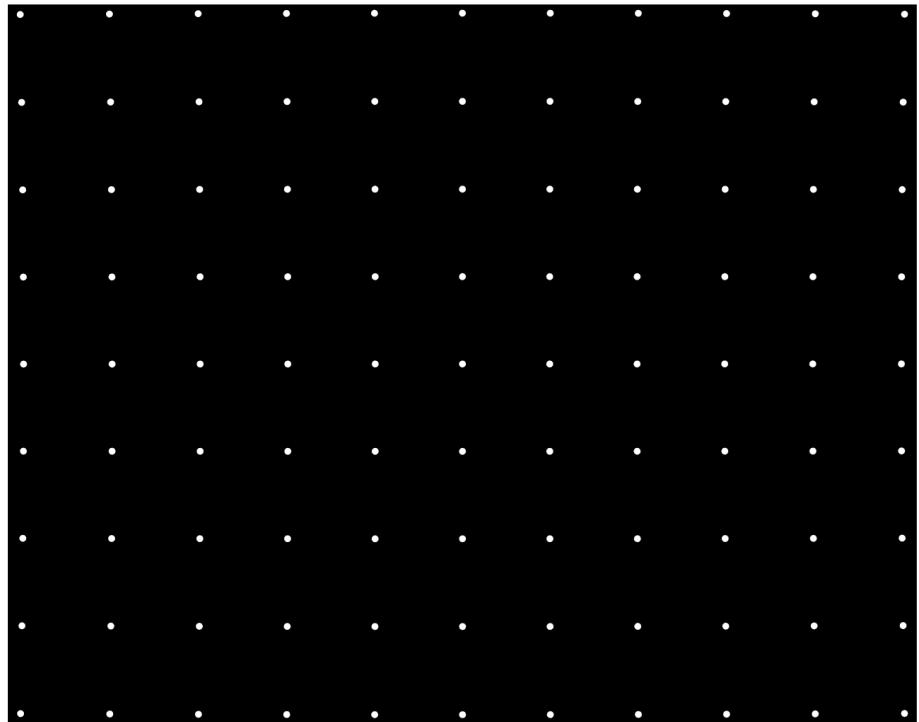


Figure 4.3: Example of a compound image generated by an ensemble of collimated point source images. Note that the fiducials have been marked for visibility.

over each corrected image of the ensemble.

$$\mathcal{C} = \sum_{i=1}^{M \cdot N} \mathcal{I}_i \quad (4.3)$$

Point source light will spread over a region of pixels described by a camera's point spread function [61, 1]. With the proper focus, point source image fiducials can be located through a variety of centroiding techniques, such as the weighted center of intensity method [1] provided by

$$u_{c_i} = \frac{1}{\mathcal{I}_{tot}} \sum_{\mathcal{M}_i} u \mathcal{I}(u, v) \quad v_{c_i} = \frac{1}{\mathcal{I}_{tot}} \sum_{\mathcal{M}_i} v \mathcal{I}(u, v) \quad (4.4)$$

where

$$\mathcal{I}_{tot} = \sum_{\mathcal{M}_i} \mathcal{I}(u, v) \quad (4.5)$$

such that \mathcal{M}_i is the set of pixels belonging to the i^{th} illuminated region of the image \mathcal{I} . The a priori fiducial lines-of-sight from Eq. 4.1 are used to initialize the process. Using factory provided parameters, the lines-of-sight are converted to pixel locations (Eq. 3.2) around which a 31×31 pixel region of interest is built. The center is then shifted to the brightest pixel within the region, and a mask is constructed using all pixels within a three pixel radius that are above some intensity threshold.

Considering the image \mathcal{C} , we recognize that we can calculate an array of observed image centroids corresponding to respective lines-of-sight from Eq 4.1. Thus, we now have the necessary information to utilize in a nonlinear optimizer to solve for the camera model found in Eq. 3.24. Choice of nonlinear optimizer is up to the reader's discretion; however, this work implemented a LMA [51, 52]. Implementation of the LMA routine for camera calibration can be found in Appendix A.

4.1.3 Star Field Calibration

As discussed, camera calibration requires a series of observed image and control point matches. If the lines-of-sight from the camera to the control points are known to high precision, a more accurate solution can be achieved using less images than when the viewpoints are unknown [62]. Stellar lines-of-sight have been extensively studied and accurately catalogued [63], which makes stars ideal candidates for calibration target points.

Star field images are obtained with a zenith pointing camera. The goal of this method is to capture images that contain as many bright stars as possible without saturating any pixels within each respective illuminated pixel region. Similar to the method discussed in the previous section, these illuminated regions are centroided and the centroids recorded as observed points of interest.

Obtaining corresponding lines-of-sight requires knowledge of the camera's attitude relative to the Earth-centered-Earth-fixed frame used by a stellar catalog [63]. The camera's attitude was estimated via an adaptation of the geometric hashing technique posed in [4] to build an index of inter-star angles. While the idea behind the algorithm is similar to that of a star tracker, this method utilized groups of four stars (quads) rather than pairs [64].

To generate the hashcode index, a unit sphere is discretized by 300 evenly spaced points. A region of interest is then considered about each of these 300 lines-of-sight. Within each region, star quad indices are generated using [4]. The basic premise is to consider four stars, such as Figure 4.4.

Then we define the following sets of angles:

$$\mathcal{P} = \{\theta_{12}, \theta_{13}, \theta_{14}, \theta_{23}, \theta_{24}, \theta_{34}\} \quad (4.6)$$

$$\mathcal{Q} = \{\theta_{ik}, \theta_{il}, \theta_{jk}, \theta_{jl} \mid \theta_{ij} = \max(\mathcal{P})\} \quad (4.7)$$

The quad is defined according to these steps:

1. Find the two stars that form the largest angle in set \mathcal{P} .

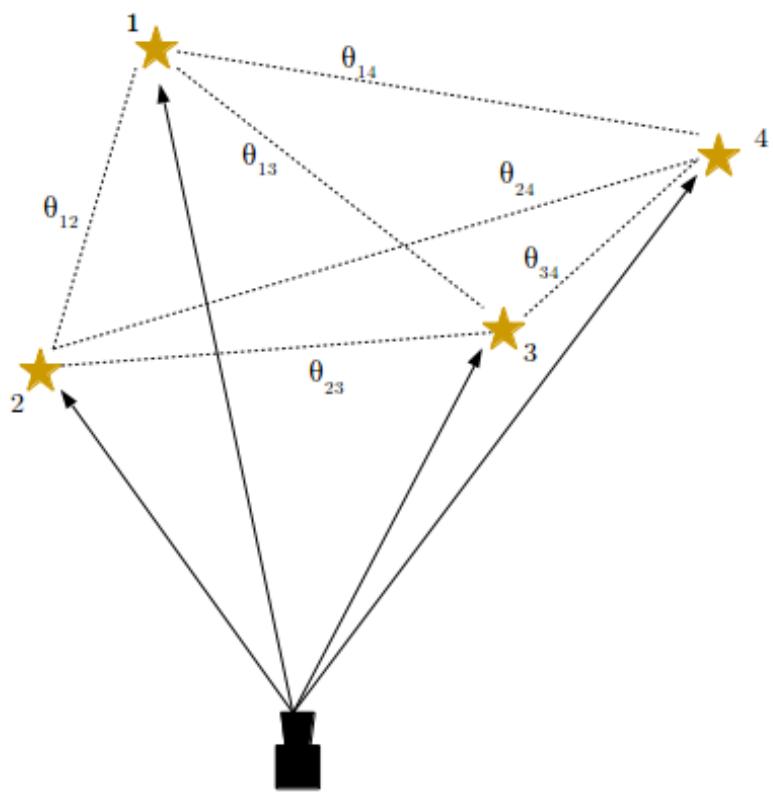


Figure 4.4: Example of the construction of a star quad using interior star angles (Image reproduced from [4]).

2. Find the stars that correspond to the smallest angle in \mathcal{Q} .
3. Record the repeated star in Steps 1 and 2.
4. Rinse and repeat until all stars are labelled in the quad.

This process is repeated for each unique four star set in the discretized window, and the process is completed at each of the 300 windows.

When an image is captured, star centroids are converted to unit lines-of-sight using an initial guess in the inverse procedure in Eqs. 3.2 and 3.24. Four lines-of-sight are arbitrarily selected and their inter-star angles computed, such that

$$\theta = \arccos \mathbf{e}_i \cdot \mathbf{e}_j$$

A quad is formed using the procedure above, and the index is searched for a match. If a match is found, the attitude is approximated using the star catalog entries and then verified using two independent stars randomly selected from the image. Note that matches are found using an error window since we are only using an initial guess for the camera parameters. If a match is not found, another four stars are selected and the process repeats until an attitude has been verified.

Once an attitude has been computed for each image in the ensemble, calibration can be performed. Using the attitudes, star catalog provided lines-of-sight can be converted to image coordinates to be used as expected star locations. These are matched with centroids found in a region of interest. As in Section 4.1.2, matched line-of-sight and observed pixel pairs are implemented into a nonlinear optimizer — such as the Levenberg-Marquardt routine shown in Appendix A — to estimate the *Brown Model* parameters. The calculated camera model is then applied to the centroid locations, and the attitude estimate can be refined.

This technique requires a significant amount of time since the Earth’s rotation is used to provide the camera’s attitude variation, which drifts the fiducials across the scene providing

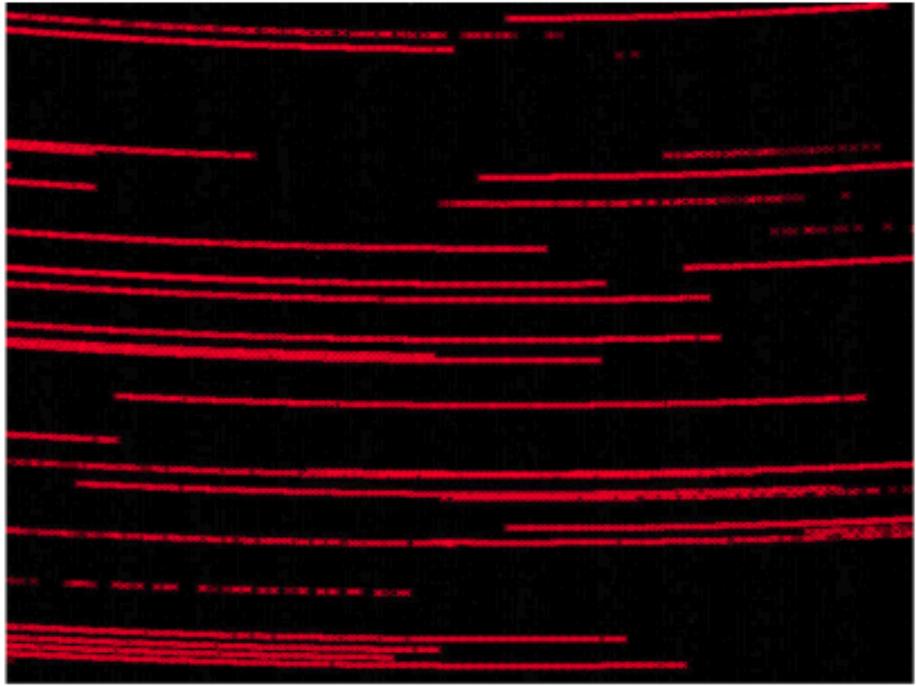


Figure 4.5: Example of the stellar drift across an image rotated by Earth rate over a one hour period [1].

an increased coverage of the field-of-view. An example, taken from [1], can be seen in Figure 4.5.

One caveat worth mentioning about this particular method is the introduction of atmospheric refraction. While atmospheric refraction will change the apparent lines-of-sight to some degree [65], thereby inducing some additional apparent lens distortion, the effect should be minute for a zenith pointing perspective camera.

4.2 Omnidirectional Camera Calibration

The method of omnidirectional camera calibration that was implemented for this research was proposed and thoroughly documented by Scaramuzza in [37]. I will repeat the convention used in Section 3.2. Unless specifically cited, the following section will provide an overview of the method proposed in [37] or a summary of my understanding of the work.

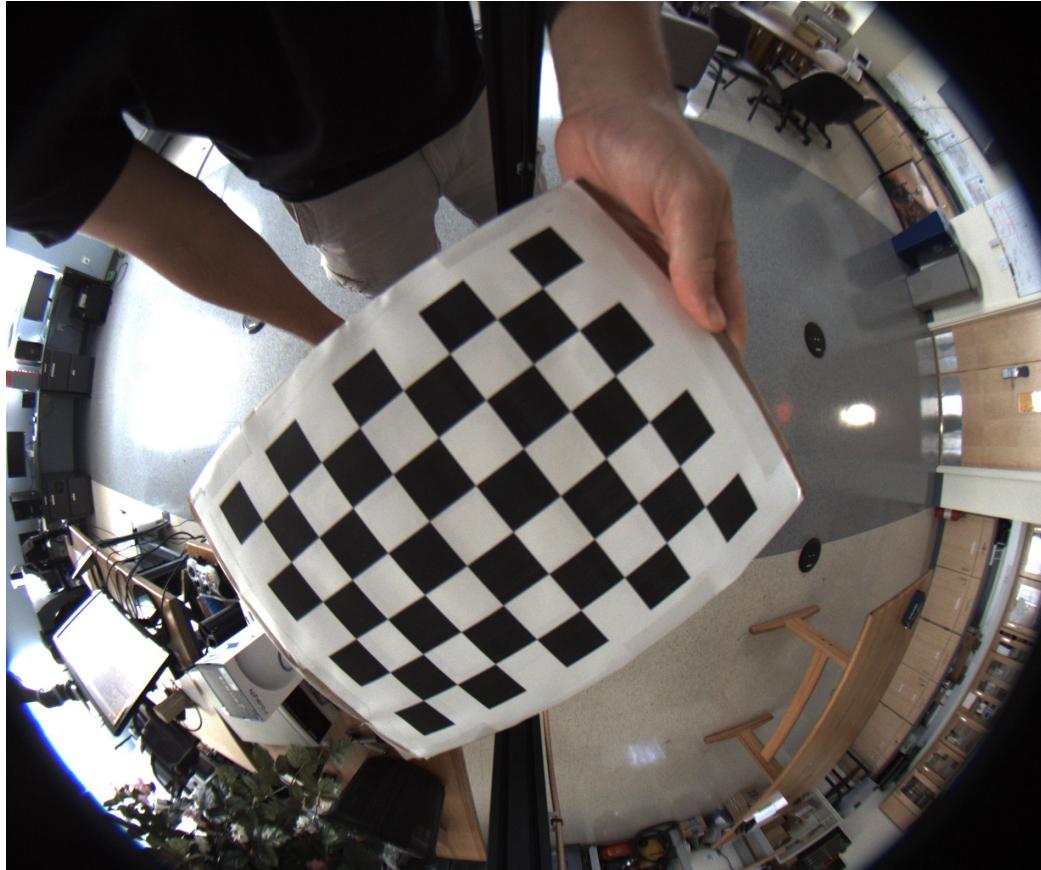


Figure 4.6: Example checkerboard image for calibration of a camera with a 180° fisheye lens.

The calibration procedure proposed by Scaramuzza yields estimates for the Taylor polynomial parameters a_0, a_2, \dots, a_n from Eq. 3.26, the elements of \mathbf{A} and the center of projection (x_p, y_p) in Eq. 3.25. The algorithm was developed into a user-friendly MATLAB® toolbox by Scaramuzza, and I refer the reader to [66] for implementation instructions; however, in this manuscript I will overview the driving mathematics presented in [37].

The process begins with a collection of checkerboard images similar to that proposed for perspective cameras (Fig. 4.6). The ensemble should contain images of the checkerboard at variety of different poses such that the camera’s entire field-of-view is sampled. That is, the relative rotation \mathbf{R}_i and translation \mathbf{T}_i between the board and camera should differ between images, which will produce a different set of 3-D observed corners $M_i =$

$[X_i^j, Y_i^j, Z_i^j]$ (the j^{th} corner of the i^{th} image, \mathcal{I}_i expressed in the board's reference frame).

The corresponding pixel coordinates for each of these corners will be denoted by $m_i^j = [u_i^j, v_i^j]$.

The parameter estimation process is two-fold. The first phase of the procedure is to estimate the Taylor polynomial parameters. This iterative procedure assumes that $\mathbf{A} = \mathbf{I}$ and $\mathbf{t} = \mathbf{0}$, which implies that the image plane and the sensor plane are identical [37]. Recalling Eq. 3.25, we see that such an assumption means that $\mathbf{x} = \mathbf{u}$. This substitution is applied to Eq. 3.29 to get Eq. 4.2.

$$\lambda \mathbf{p}_i^j = \lambda \begin{bmatrix} u_i \\ v_i \\ g(\rho_{\mathbf{u}_i}) \end{bmatrix} = [\mathbf{R}_i \mid \mathbf{T}_i] \begin{bmatrix} \mathbf{X}_i^j \\ 1 \end{bmatrix} \quad (4.8)$$

where $\mathbf{u}_i = (u_i, v_i)$ denotes the i^{th} observed pixel coordinate and $g(\rho_{\mathbf{u}_i})$ is the Taylor polynomial expansion of said image point (Eq. 3.26). Now, we are able to estimate the Taylor parameters a_0, a_2, \dots, a_n .

Without loss of generality, we may let $Z_i^j = 0$ due to planar nature of the target board [37]. Thus, can be simplified to Eq. 4.9

$$[\mathbf{R}_i \mid \mathbf{T}_i] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = [\mathbf{r}_i^1 \ \mathbf{r}_i^2 \ \mathbf{r}_i^3 \mid \mathbf{T}_i] \begin{bmatrix} X_i^j \\ Y_i^j \\ 0 \\ 1 \end{bmatrix}$$

$$= [\mathbf{r}_i^1 \ \mathbf{r}_i^2 \mid \mathbf{T}_i] \begin{bmatrix} X_i^j \\ Y_i^j \\ 1 \end{bmatrix} \quad (4.9)$$

where \mathbf{r}_i^k is the k^{th} column of the rotation matrix of the i^{th} checkerboard. It should now be apparent that this technique is similar to the Zhang method [53] described in Section 4.1.1 in the need to simultaneously estimate the pose of the board and the Taylor parameters.

Estimation of the Board Pose

Solving for the pose estimation requires us to first eliminate the scaling factor λ . We do this by leveraging the fact that the cross product of a vector with itself is zero. Hence, we have the relation in Eq. 4.10 [37].

$$\lambda (\mathbf{p}_i^j \times \mathbf{p}_i^j) = \mathbf{p}_i^j \times [\mathbf{r}_i^1 \ \mathbf{r}_i^2 \ | \ \mathbf{T}_i] \begin{bmatrix} X_i^j \\ Y_i^j \\ 1 \end{bmatrix} = 0 \quad (4.10)$$

Substituting Eq. 4.2 into the relation above, we have (Eq. 4.11).

$$\begin{bmatrix} u_i^j \\ v_i^j \\ a_0 + a_2(\rho_i^j)^2 + \dots + a_n(\rho_i^j)^n \end{bmatrix} \times [\mathbf{r}_i^1 \ \mathbf{r}_i^2 \ \mathbf{T}_i] \begin{bmatrix} X_i^j \\ Y_i^j \\ 1 \end{bmatrix} = 0 \quad (4.11)$$

We can now expand this relation to three homogenous equations (Eqs. 4.12 - 4.14) by considering each single observation (\mathbf{p}_i) independently [37].

$$v_i^j (r_{i31}X_i^j + r_{i32}Y_i^j + t_{i3}) - g(\rho_i^j) (r_{i21}X_i^j + r_{i22}Y_i^j + t_{i2}) = 0 \quad (4.12)$$

$$g(\rho_i^j) (r_{i11}X_i^j + r_{i12}Y_i^j + t_{i1}) - u_i^j (r_{i31}X_i^j + r_{i32}Y_i^j + t_{i3}) = 0 \quad (4.13)$$

$$u_i^j (r_{i21}X_i^j + r_{i22}Y_i^j + t_{i2}) - v_i^j (r_{i11}X_i^j + r_{i12}Y_i^j + t_{i1}) = 0 \quad (4.14)$$

Recall that the observed image coordinates (u_i^j, v_i^j) and the checkerboard reference points (X_i^j, Y_i^j) are known. Thus, we can rewrite Eq. 4.14 as a system of K linear equations — corresponding to K image points — to solve for the unknown rigid body transform parameters [37] as Eq. 4.15

$$\mathbf{F}\mathbf{H} = \mathbf{0} \quad (4.15)$$

where

$$\mathbf{F} = \begin{bmatrix} -v_i^1 X_i^1 & -v_i^1 Y_i^1 & u_i^1 X_i^1 & u_i^1 Y_i^1 & -v_i^1 & u_i^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -v_i^K X_i^K & -v_i^K Y_i^K & u_i^K X_i^K & u_i^K Y_i^K & -v_i^K & u_i^K \end{bmatrix} \quad (4.16)$$

and the unknowns are stacked in \mathbf{H} yielding

$$\mathbf{H} = [r_{i_{11}} \ r_{i_{12}} \ r_{i_{21}} \ r_{i_{22}} \ t_{i_1} \ t_{i_2}]^T \quad (4.17)$$

This system can be solved via a singular value decomposition minimized $\|\mathbf{FH}\|^2$ subject to $\|\mathbf{H}\|^2 = 1$. Thus, we are able to solve for all parameters of the rigid body transform with the exception of z -axis translation parameter t_{i_3} [37].

Estimation of the Taylor Expansion Parameters

Now that we have the solution from Eq. 4.15, we can substitute into Eqs. 4.12 and 4.13 to solve for the unknown Taylor expansion parameters and the remaining translation parameter. Again, we stack a system of K equations relating K observed points Eq. 4.18 [37].

$$\begin{bmatrix} B_i^1 & B_i^1(\rho_i^1)^2 & \cdots & B_i^1(\rho_i^1)^n & -v_i^1 & 0 & \cdots & 0 \\ C_i^1 & C_i^1(\rho_i^1)^2 & \cdots & C_i^1(\rho_i^1)^n & -u_i^1 & 0 & \cdots & 0 \\ \vdots & \vdots \\ B_i^K & B_i^K(\rho_i^K)^2 & \cdots & B_i^K(\rho_i^K)^n & -v_i^K & 0 & \cdots & 0 \\ C_i^K & C_i^K(\rho_i^K)^2 & \cdots & C_i^K(\rho_i^K)^n & -u_i^K & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_2 \\ \vdots \\ a_n \\ t_{1_3} \\ t_{2_3} \\ \vdots \\ t_{K_3} \end{bmatrix} = \begin{bmatrix} D_i^1 \\ E_i^1 \\ \vdots \\ D_i^K \\ E_i^K \\ \vdots \end{bmatrix} \quad (4.18)$$

where we define the expressions for B , C , D , and E to be

$$\begin{aligned} B_i^j &= r_{i_{21}} X_i^j + r_{i_{22}} Y_i^j + t_{i_2} \\ C_i^j &= v_i^j (r_{i_{31}} X_i^j + r_{i_{32}} Y_i^j) \\ D_i^j &= r_{i_{11}} X_i^j + r_{i_{12}} Y_i^j + t_{i_1} \\ E_i^j &= u_i^j (r_{i_{31}} X_i^j + r_{i_{32}} Y_i^j) \end{aligned}$$

Now, Eq. 4.18 can be solved for a_0 , a_2 , ..., a_n , and t_{i_3} using a pseudo-inverse method for the solution to the normal equations [37]. Scaramuzza notes that the optimal degree of the solution can be obtained by starting at $n = 2$ and increasing unitarily until the average reprojection error is below a defined threshold.

Refinement of the Estimates

Recall that we previously assumed $\mathbf{A} = \mathbf{I}$ and $\mathbf{t} = 0$, which is a reasonable approximation for a quality omnidirectional camera system [37]. In the next section, we will overview a nonlinear refinement procedure that will improve our estimates of \mathbf{A} , \mathbf{t} , \mathbf{R} , \mathbf{T} , and the Taylor polynomial parameters.

Assume we have N images of a calibration pattern containing K corners. Also assume

independent noise corrupts each of these image points. Then, we can find a maximum likelihood estimate for the aforementioned parameters by minimizing the sum of all Euclidean distances between each world point, projected into the image, and the corresponding observed image point for each of the K point correspondencies across all N images. Scaramuzza chose to use the Levenberg-Marquardt nonlinear optimizer [51, 52] to solve the optimization problem as initialized by the estimates obtained in Eqs. 4.15 and 4.18.

As discussed, this procedure was implemented into the OcamCalib Toolbox in MATLAB[®] which is very well documented at [66]. I recommend the interested reader to [66] for implementation instructions.

4.3 Camera-LiDAR Camera Calibration

Though the literature contains autonomous approaches to LiDAR-camera calibration [24] and processes that do not require the use of a special target [46], such methods are only readily applicable under a subset of conditions. This work has chosen to use a more user-heavy approach. The process again begins with a collection of fiducial board data. The ensemble should contain a collection of matching image and LiDAR point cloud sets of static scenes containing the fiducial board, an example of which can be seen in Figure 4.7. As discussed in the previous calibration techniques, the board should be repositioned in each scene such that the fiducials are uniformly spread across the entirety of the camera, and potentially LiDAR, field-of-view.

The next step is to locate the matching fiducial centroids in the image and LiDAR point cloud sets. First, LiDAR fiducial points will be located. To start this process, depth clustering is performed on the point cloud using the MATLAB[®] *clusterLidar* function. Removal of points corresponding to the floor will allow for greater ease of clustering. Then, the user retains only the cluster of points pertaining to the calibration board.

Now that the board has been isolated in the data, clustering can be repeated using both

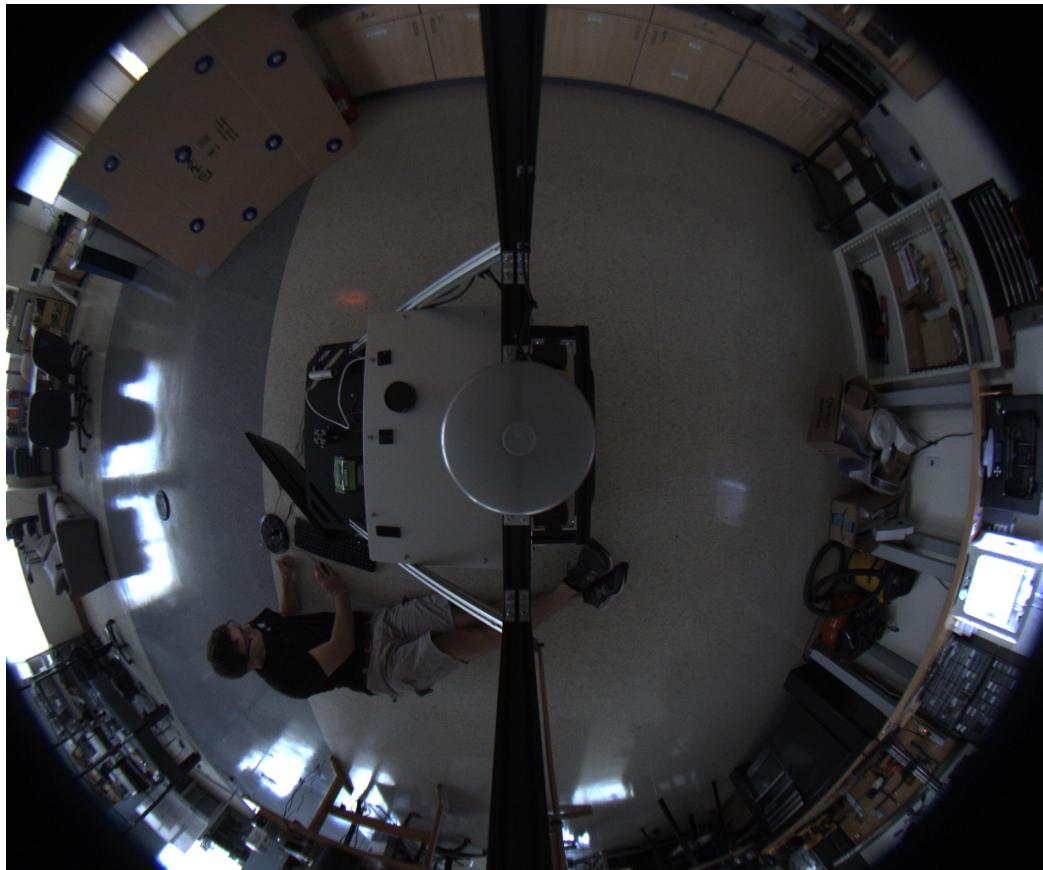


Figure 4.7: Example image of a calibration board containing reflective fiducials. Image was captured with a downward facing 180° fisheye lens.

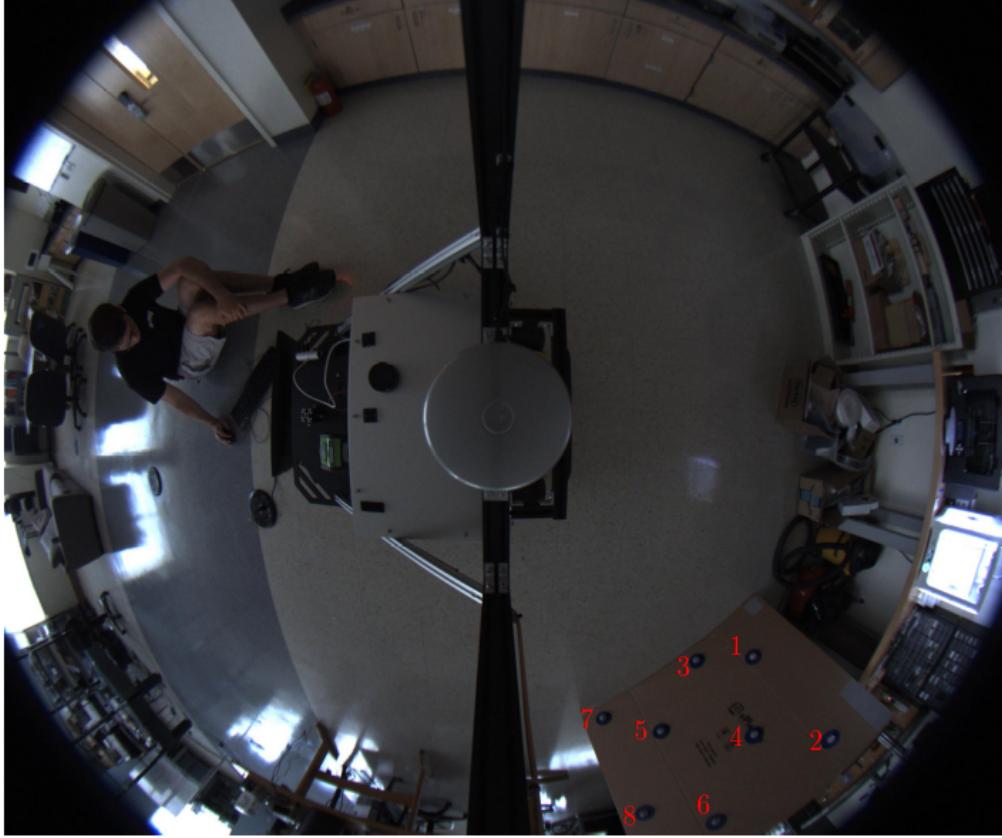


Figure 4.8: Example arrangement of fiducial locations for easy matching between LiDAR points and image centroids.

depth and reflectivity values. Depending on the environment in which the data was collected, it could be advantageous to perform this step on the brightest subset of the overall data, the size of which must be chosen according to each application. Again, the *clusterLidar* function is implemented; however, this time the reflectivity values are included in the input data. The output of *clusterLidar* will be sets of points corresponding to the reflective markers on the calibration board. Each of these sets is averaged to estimate the centroid of the respective fiducial markers. These points must be ordered according to some specific pattern, such as that displayed in Figure 4.8, to be manually matched with respective image coordinates. This process is repeated for each point cloud in the data ensemble, and the fiducial sets stacked into a single set of data for future use.

The next step is to locate the markers in the corresponding images. This can be per-

formed in a variety of ways, but this research chose to implement the MATLAB® *detect-SURFFeatures* function on the region of each image containing the board. The returned features corresponding to the fiducial centers are manually selected by the user, and stored in the same arrangement as that chosen to structure the LiDAR points to ensure markers are matched with the appropriate data. Note that any missing LiDAR or image points must be denoted in such a way that the data can be removed from both LiDAR and image sets where a full match is not located.

Now that both data sets — 3-D LiDAR points and 2-D image centroids — have been obtained, they can be used to estimate the rigid body transformation. If a perspective camera is used, then the solution can be obtained through the well-known Perspective- n -Point (PnP) problem [26]. If an omnidirectional camera is implemented, however, then the PnP solution will not be sufficient [46].

The function being minimized in PnP is classically the image reprojection error Eq. 4.19 [46].

$$\min_{\mathbf{R}, \mathbf{t}} \frac{1}{2} \sum_{i=1}^n \|\mathbf{u}_i - \hat{\mathbf{u}}_i(\mathbf{R}, \mathbf{t}, \mathbf{v}_\ell)\|^2 \quad (4.19)$$

where \mathbf{u}_i are observed image centroids and $\hat{\mathbf{u}}_i(\mathbf{R}, \mathbf{t}, \mathbf{v}_\ell)$ is the corresponding 3-D LiDAR vector projected onto the image. However, such minimization of the Euclidean distance assumes that there is uniform resolution across the entire image, which is not the case for omnidirectional cameras. Omnidirectional cameras, as discussed in Section 3.2, uniformly relate the world to a unit sphere. Thus, an error function that is associated to spatial distribution on a sphere is more advantageous to consider than Euclidean distance in the image. The Riemann metric on a sphere, provided in Eq. 4.20, is a better cost function to optimize for omnidirectional cameras [46, 67].

$$\min_{\mathbf{R}, \mathbf{t}} \frac{1}{2} \sum_{i=1}^n \left\| \arccos \left(\mathbf{e}_{C_i}^T \cdot \mathbf{e}_{\ell_{C_i}} \right) \right\|^2 \quad (4.20)$$

where \mathbf{e}_{C_i} is the i^{th} image centroid projected to the unit sphere (from Eqs. 3.25, 3.29 and 3.30) and $\mathbf{e}_{\ell_{C_i}}$ is the normalized LiDAR measurement in the camera frame (Eq. 3.31). This cost function is utilized in an interior point routine [68] using the nonlinear optimization function *lsqnonlin* in MATLAB[®] to obtain estimates for the rotation \mathbf{R} and translation \mathbf{t} of the LiDAR frame to the camera frame.

Chapter 5

Results

5.1 Perspective Cameras

The calibration techniques listed in Section 4.1 were tested with a PixelLink PL-D735 camera with a 35 mm focal length lens. The checkerboard calibration method was tested using an ensemble of 60 checkerboard images captured in a windowless room with fluorescent lighting. Calibration was performed on three different partitions of the data set: the first half, the second half, and the entire ensemble. A mesh was generated and converted both to a pinhole digital image (Eq. 3.2) and a distorted perspective image (Eqs. 3.2 and 3.24). The difference between the ideal pinhole and perspective images was computed, and contour plots of the estimated distortion are shown in Figures 5.1 - 5.2.

Careful consideration of the distortion maps shows that this method is not repeatable to any high degree of accuracy. This variability is a side effect of Zhang's algorithm [53], which simultaneously estimates the relative pose of the checkerboard in each image and the *Brown model* parameters. The pose of the checkerboard naturally dictates the true 3-D locations of each corner, on which estimation of the camera model is inherently dependent. Consequently, for as the image set is changed, so are the board poses and the corresponding camera model estimates as the error in the pose estimates leaks into the camera model. I

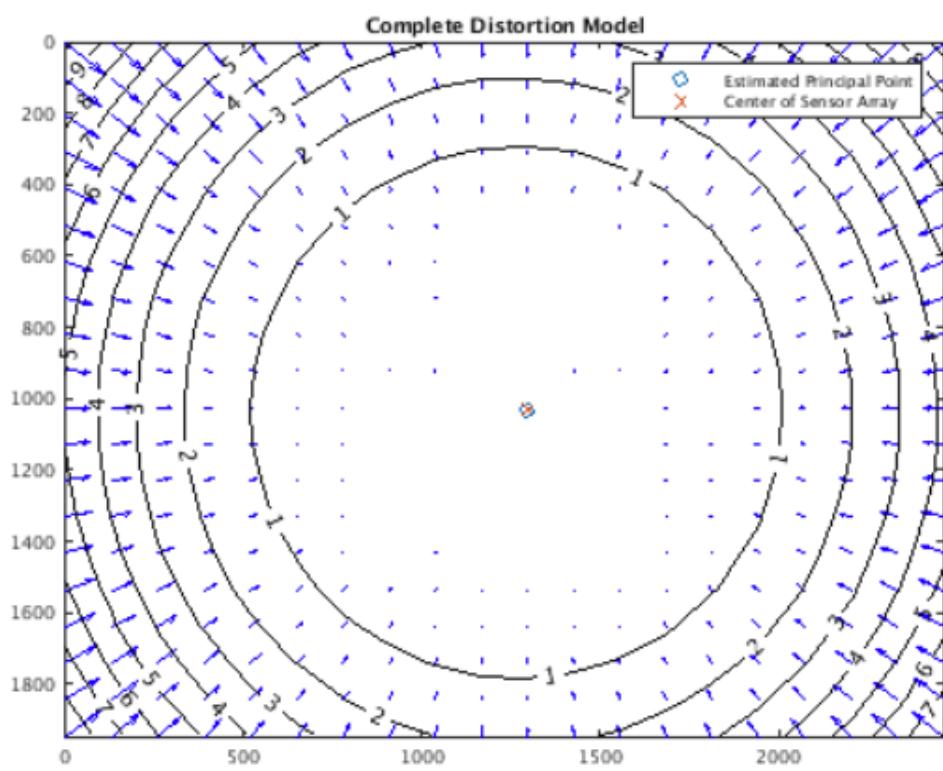


Figure 5.1: Distortion map calculated from the first 30 images of the image ensemble [3].

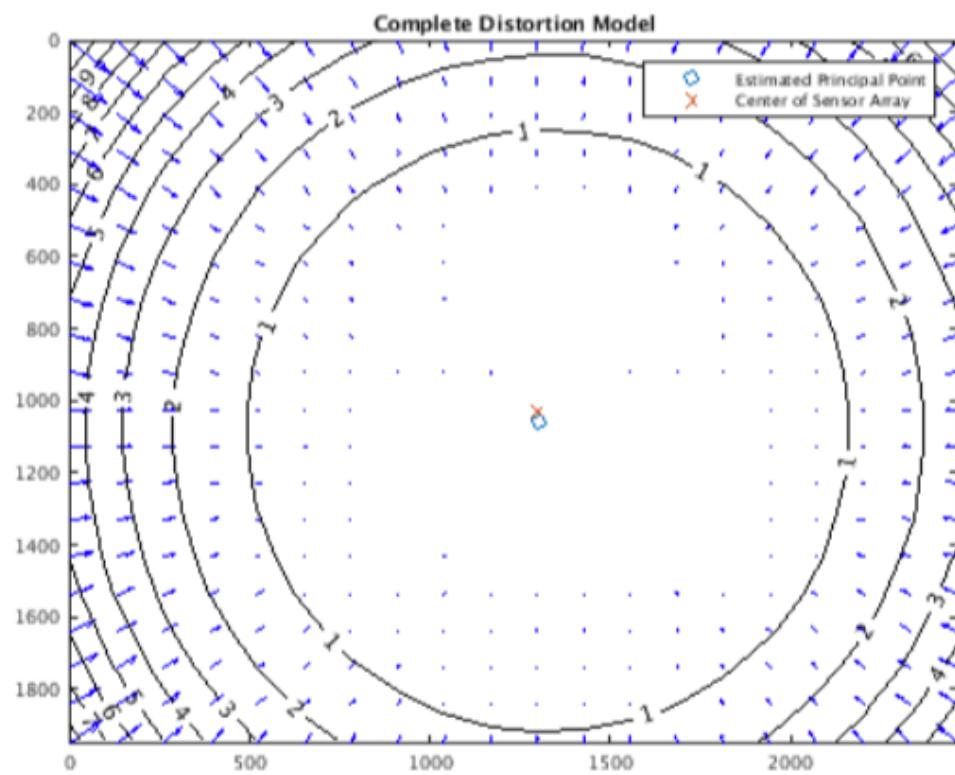


Figure 5.2: Distortion map calculated from the second half of the image ensemble [3].

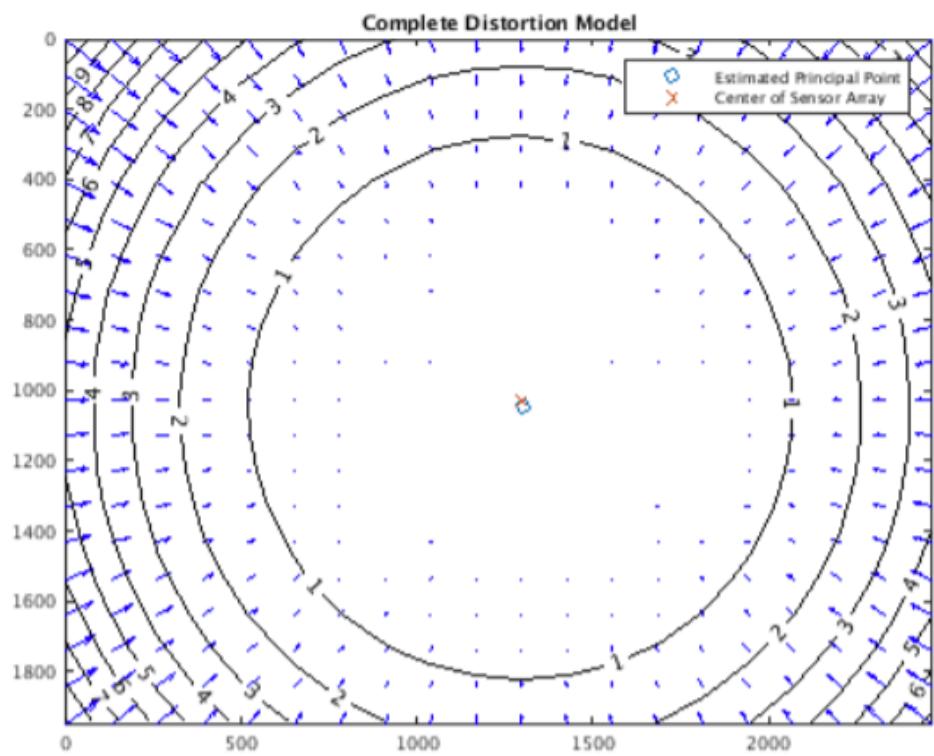


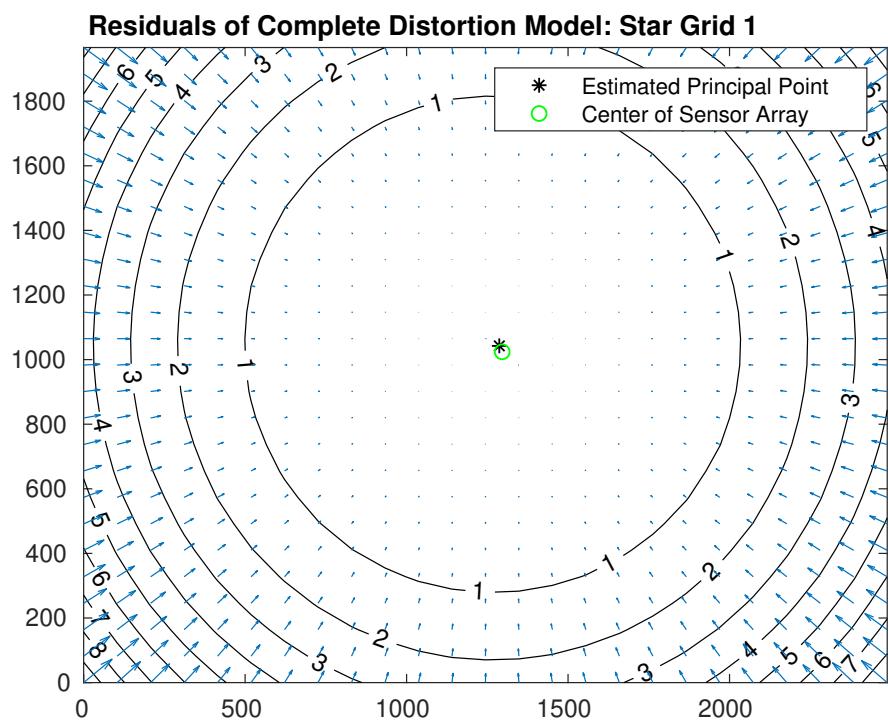
Figure 5.3: Distortion map calculated from the entire 60 images of image ensemble [3].

am, by no means, claiming that the checkerboard method is insufficient for it's designed task, which was to correct an image for visual error. This calibration method is even plenty sufficient for many computer vision and robotic tasks [19]. It is a well documented, easy to approach method for camera calibration. I am, however, suggesting that a different approach is needed to meet high accuracy requirements, such as those found in spacecraft optical navigation [17].

The next experiment reviewed in this work is an analysis of goniometric calibration performed on a granite optical bench. A compound image of a 9×11 grid of collimated point sources was generated. Two separate iterations were performed to capture both a training and testing image. Figures 5.4 and 5.5 display distortion maps achieved via calibration using each separate grid. Similar to above, the maps were created by converting the point source lines-of-sight to a pinhole digital image and a distorted image and calculating the difference.

Comparing Figures 5.4 and 5.5, we see that this technique is far more repeatable than the checkerboard method. Intuitively, this makes sense. If a high precision gimbal is used to rotate a carefully placed camera, then the lines-of-sight to each fiducial location should be well known; thus, the calibration process need only account for the camera parameters relative to an assumed truth. The precision of this technique can be seen in Figures 5.6 and 5.7, which were calculated on a second grid using a calibration mapping estimated via the first grid.

The effect of lens distortion on an individual fiducial is displayed in Figure 5.8, where the ideal pinhole location was distorted to the observed. This was done for presentation's sake, with no difference to the accuracy of the solution, since the computation to distort an ideal image using Eq. 3.24 is more computationally efficient than the undistorting an observed image.



75

Figure 5.4: Distortion map calculated using the first grid [3].

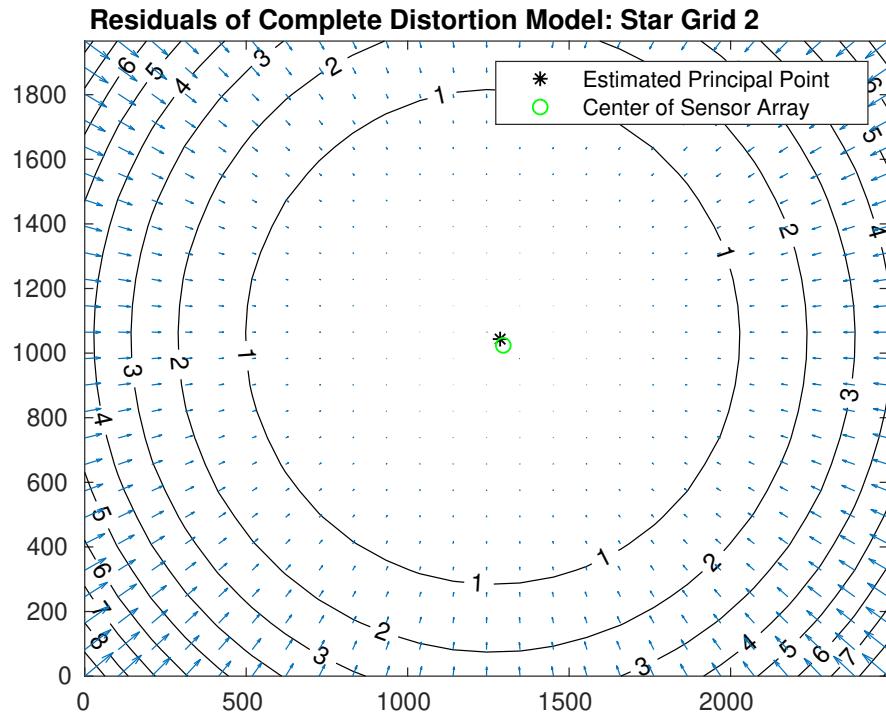


Figure 5.5: Distortion map calculated using the second grid [3].

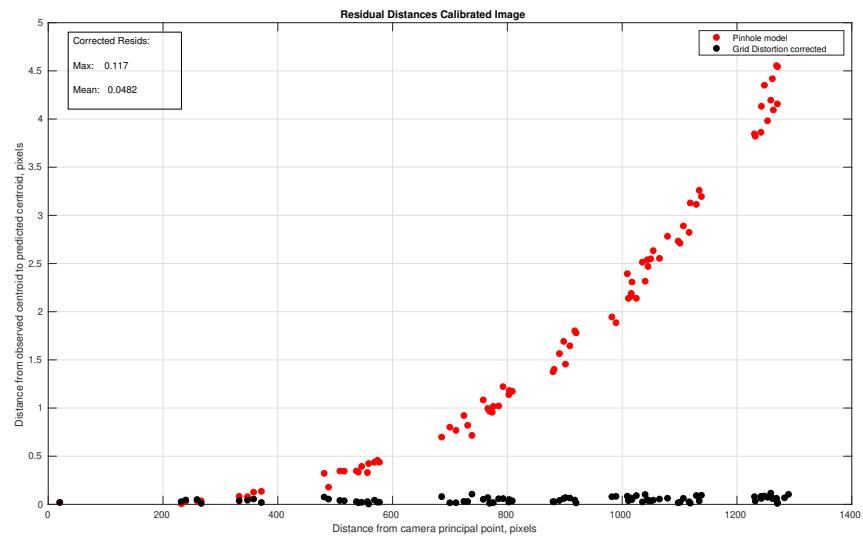


Figure 5.6: Net residuals between estimated image locations and observed image locations [3].

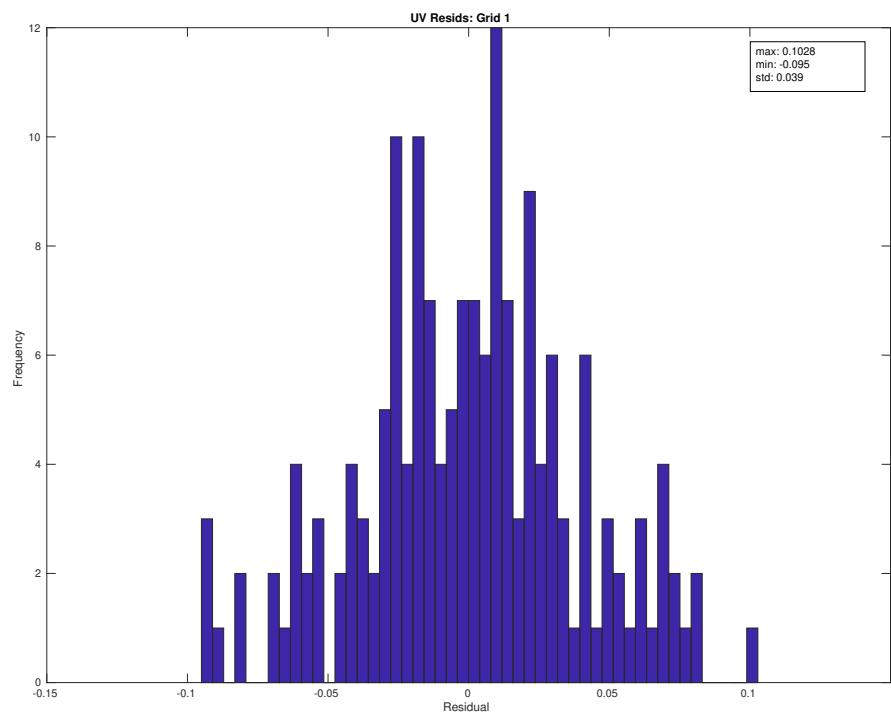


Figure 5.7: Magnitude of the (u, v) residuals between estimated image locations and observed image locations [3].

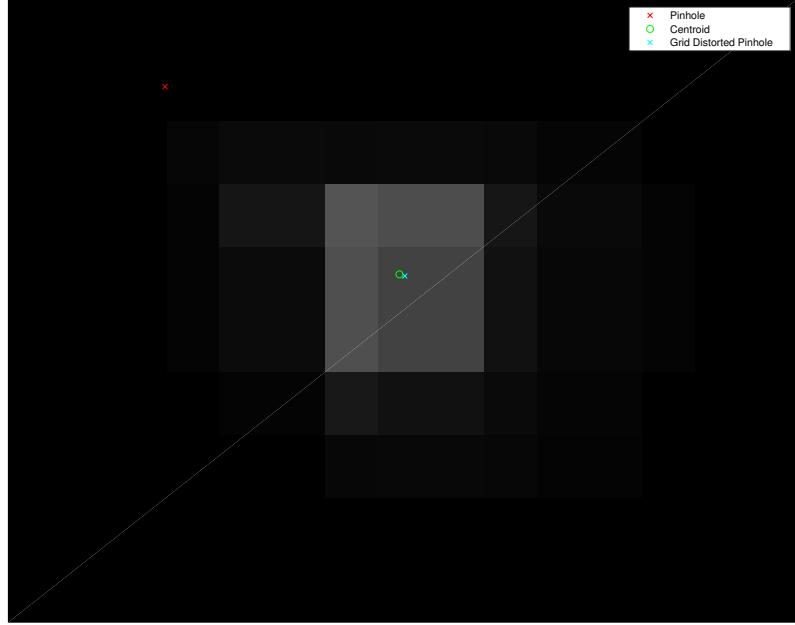


Figure 5.8: Magnified view of an observed point source showing the effects of lens distortion [3].

The final perspective camera experiment reported in this work was performed on an ensemble of 240 night sky images captured from a zenith pointing camera in Houston, TX. Calibration was performed on a set of 15 evenly spaced images throughout the ensemble. Figure 5.9 shows the residuals computed on these training images.

The computed camera parameters were applied to the stars found in the entire image ensemble. Similar to the method mentioned previously, the catalog provided lines-of-sight were converted distorted and converted to a digital image. The residuals between the distorted catalog values and the observed image points are compared in Figures 5.10 and 5.11.

Clearly, if one is willing to invest the funds for an accurate collimating lens and precision gimbal then goniometric calibration is the preferred method of calibration. However, stellar lines-of-sight are known to high accuracy and, in most cases, atmospheric refraction should not impose a major impact on the quality of the catalog estimates. Thus, calibration via stellar imagery provides the most “bang for the buck”, so to speak, as long as the user

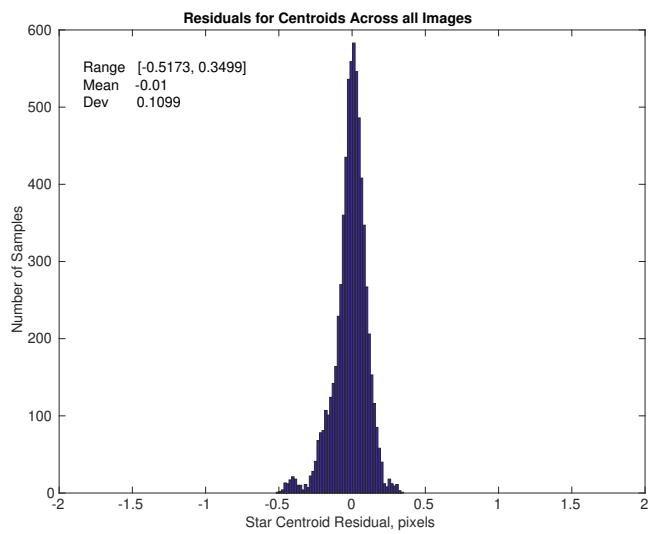


Figure 5.9: Magnitude of post-calibration directional residuals across the training images [3].

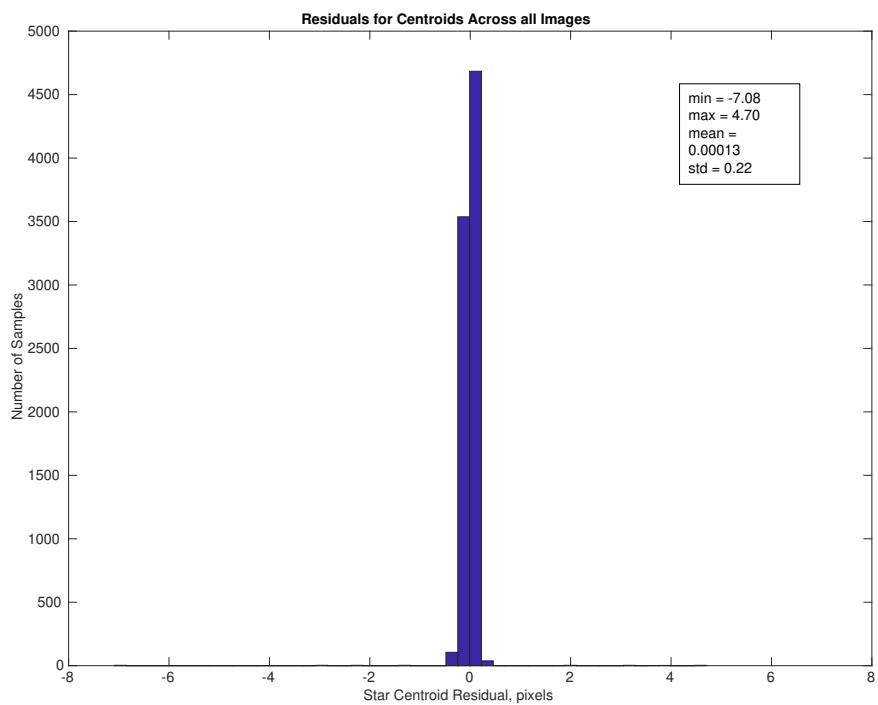


Figure 5.10: Magnitude of post-calibration directional residuals across the entire image ensemble [3].

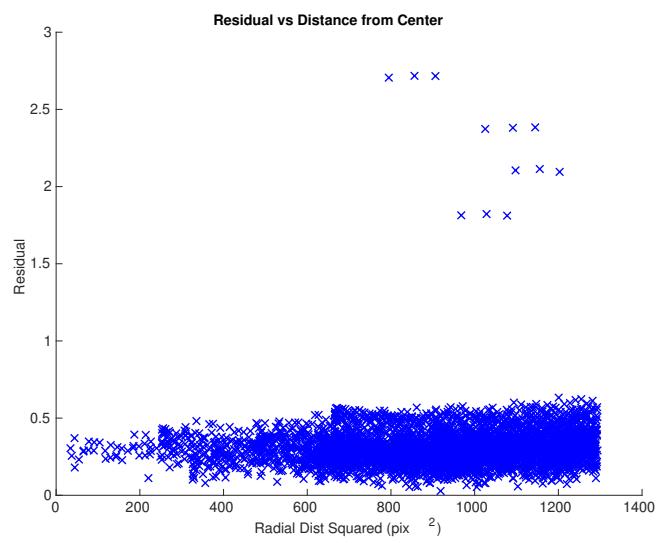


Figure 5.11: Magnitude of post-calibration residuals across the entire image ensemble [3].

has access to a sturdy tripod and an area that does not suffer from much light pollution. The lowest quality solution, though readily approachable by any user, can be obtained using checkerboard images. This method is inexpensive, quick, and sufficiently accurate for near-Earth ranges; thus, it is the ubiquitous method in the robotics community.

5.2 Omnidirectional Cameras

The omnidirectional camera calibration algorithm described in Section 4.2 was experimentally tested with a FLIR® 5MP Blackfly-S camera outfitted with a 180° fisheye lens. An ensemble of 24 checkerboard images (*e.g.* Figure 4.6) was used with the OCamCalib MATLAB® toolbox to estimate a 4th degree projection function. Figure 5.12 displays the solution of $g(\rho)$ — denoted by $f(\rho)$ by the author of the toolbox — and the relation of ρ to the off-horizontal angle of the incoming light.

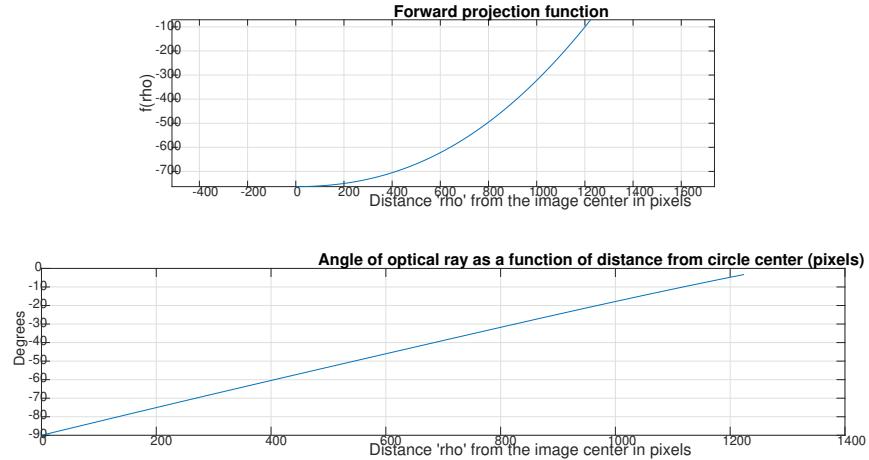


Figure 5.12: Solution to the *Taylor* projection function related to the distance from the image center and the angle of the incoming ray.

The estimated pose of the board in each image can be confirmed in Figure 5.13. While it is impossible to quantitatively verify the pose estimates, such a plot shows that the solutions are at least observably realistic.

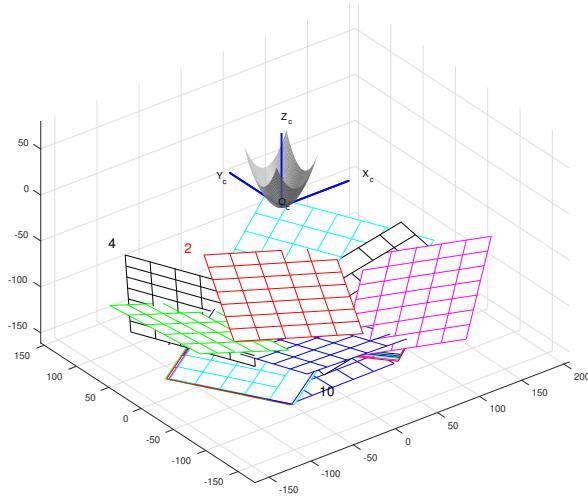


Figure 5.13: Estimates of the checkerboard pose in each image.

The OCamCalib toolbox also includes a corner refinement step, which is performed after the initial calibration. The improvement on the reprojection error when the corner refinement is implemented is shown in Figure 5.14. Observe that this technique is similar to the checkerboard method discussed for perspective cameras. Consequently, it suffers from the same flaws. However, as shown in this section, a relatively accurate solution can be achieved using this method, considering the high distortions initially present in an omnidirectional camera image. Again, this method provides a solution that is accurate enough for close range (meter level) applications.

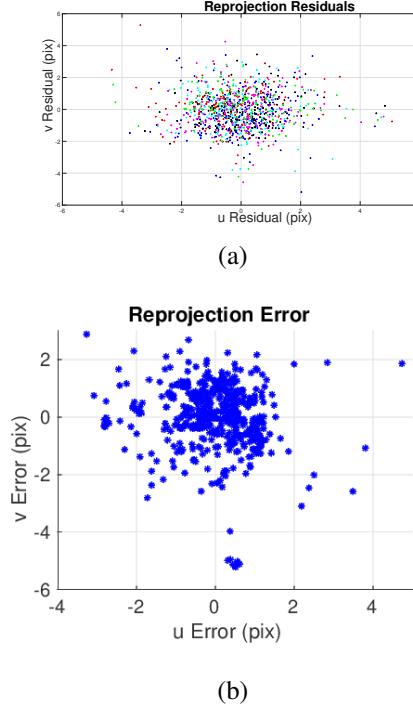


Figure 5.14: (a) Reprojection error before corner refinement. (b) Reprojection error after corner refinement.

5.3 Camera-LiDAR System

Calibration between a fisheye camera and LiDAR system was tested with a downward facing FLIR® 5MP Blackfly-S mounted above a Velodyne® 32-channel 360° scanning LiDAR. A hand measured rigid body transform — LiDAR to camera — was found to yield rotation angles (ZYX) of $[-90^\circ, 0^\circ, 180^\circ]$ and a translation of $[0, 0, 0.275]$ meters. A depiction of the setup is seen in Figure 5.15.

Four images were captured of a board containing 8 reflective fiducial markers. The images were captured such that the board was approximately centered in each quadrant of the camera frame (Figure 5.16).

Calibration was then performed using the *interior point* method provided in MATLAB®. Constraints of [measured $\pm 60^\circ$] and [measured ± 1 m] were imposed on the euler angle rotation solutions and $[x, y, z]$ translation solutions, respectively. Figure 5.17 provides a visual comparison of the intial measured transformation and the calibration solution by

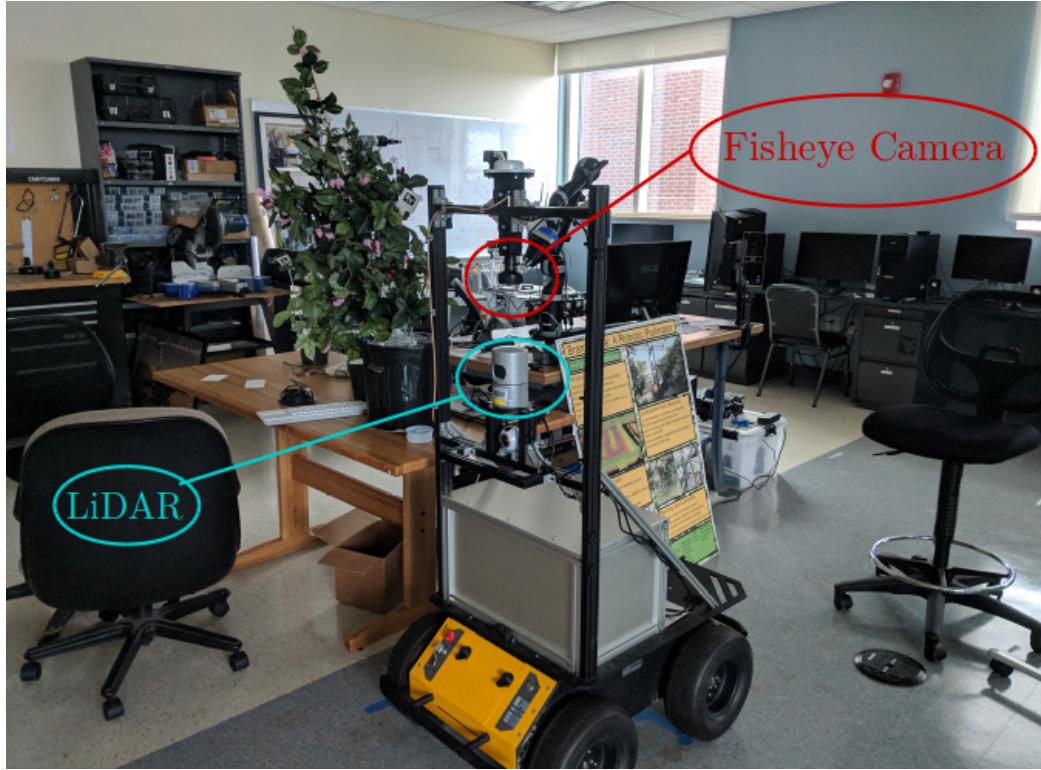


Figure 5.15: Depiction of the experimental set up used for this manuscript.

transforming the LiDAR data to the camera frame via the respective estimate and projecting the vectors into the image. A quantitative comparison of the measured transformation and calibration solution can be seen in Figure 5.18, which reports the post-calibration angular errors between the transformed LiDAR lines-of-sight and image centroid vectors projected to the unit sphere. The calibration solution clearly improves upon the measured transformation. The improvement is only slight due to the simple sensor arrangement present for the experimental procedure in this manuscript. Assuming poor knowledge of the arrangement (necessary for the imposed constraint discussed above), Figure 5.19 shows the calibration result if the transformation is initialized by the identity.

Further testing on the need for a good initial estimate was performed using a 2000 iteration Monte Carlo. At each iteration, the intial rotation was assumed to correspond to three euler angles, each chosen uniformly random from the interval $[-180^\circ, 180^\circ]$, and the initial translation components were each uniformly chosed from the interval $[-5, 5]$ me-

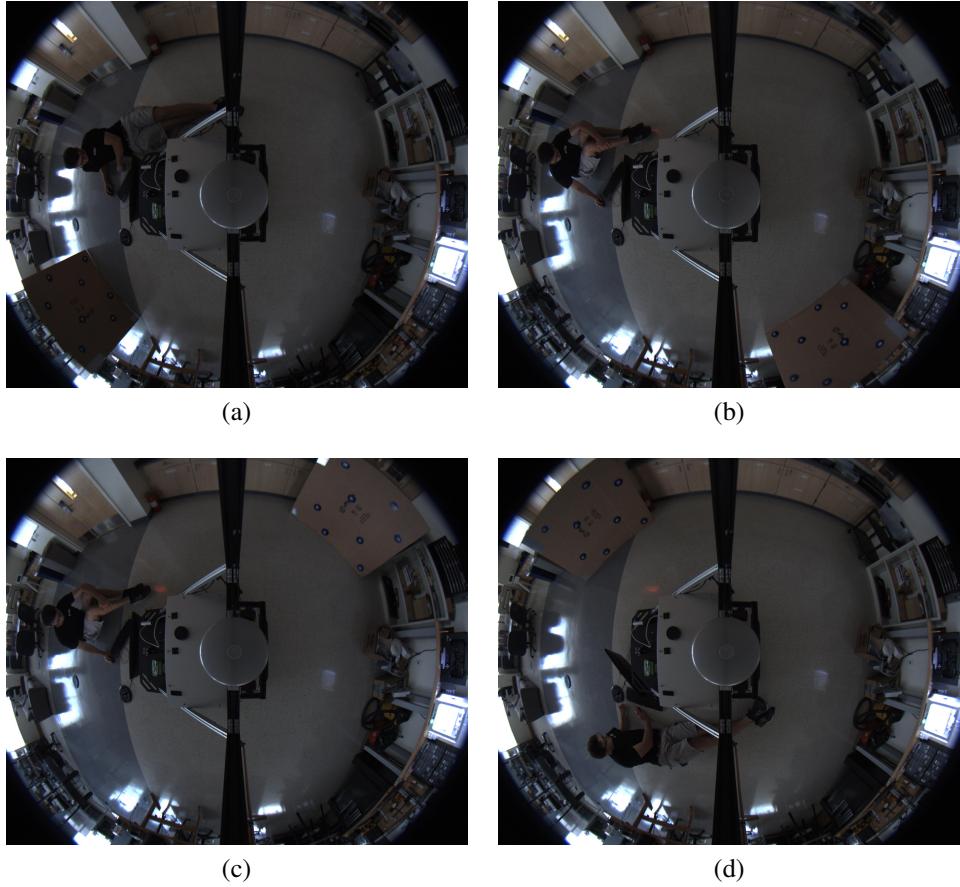


Figure 5.16: Example ensemble of calibration board images

ters. Figures 5.20 and 5.21 show the respective components of the rotation and translation solutions, in degrees and meters. The angular errors at each fiducial over the course of the simulation are reported in Figure 5.22. This study shows that with very loose knowledge of the arrangement between an omnidirectional camera and a LiDAR, a constrained optimization problem can be solved to achieve a valid estimate of the rigid body transformation between the two sensors.

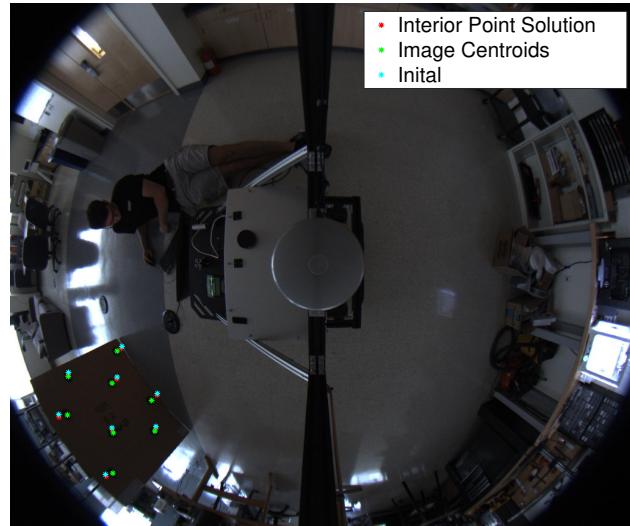


Figure 5.17: Comparison of hand measured transformation and the interior point solution.

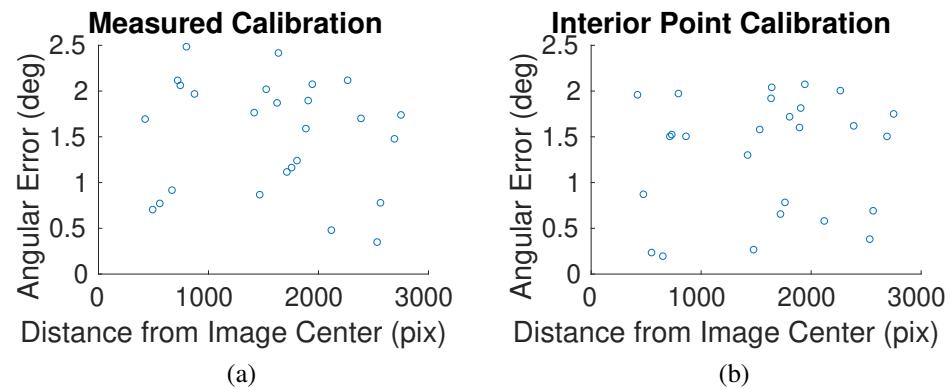


Figure 5.18: Comparison of the angular error (in degrees) between the transformed LiDAR vectors and their corresponding image vectors using the (a) measured transformation and (b) interior point solution.

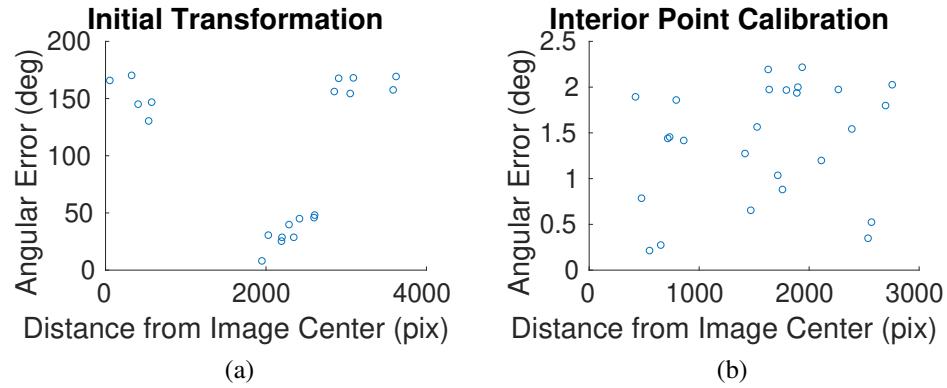


Figure 5.19: Comparison of the angular error (in degrees) between the transformed LiDAR vectors and their corresponding image vectors using the (a) identity transformation and (b) interior point solution.

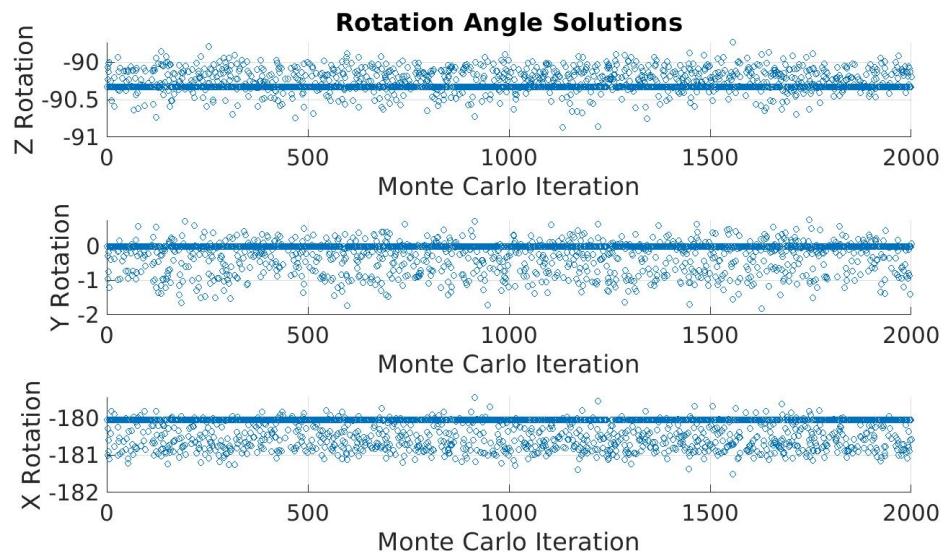


Figure 5.20: Euler angle rotation solutions, in degrees, found via the interior point optimization over 2000 random initializations.

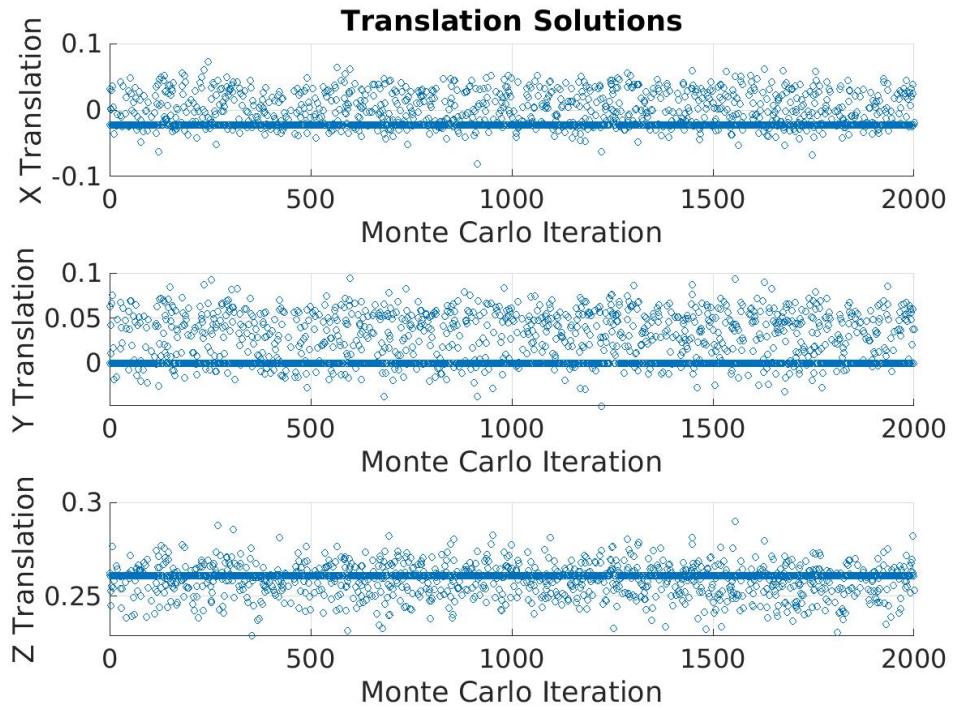


Figure 5.21: Translation solution elements, in meters, found via the interior point optimization over 2000 random initializations.

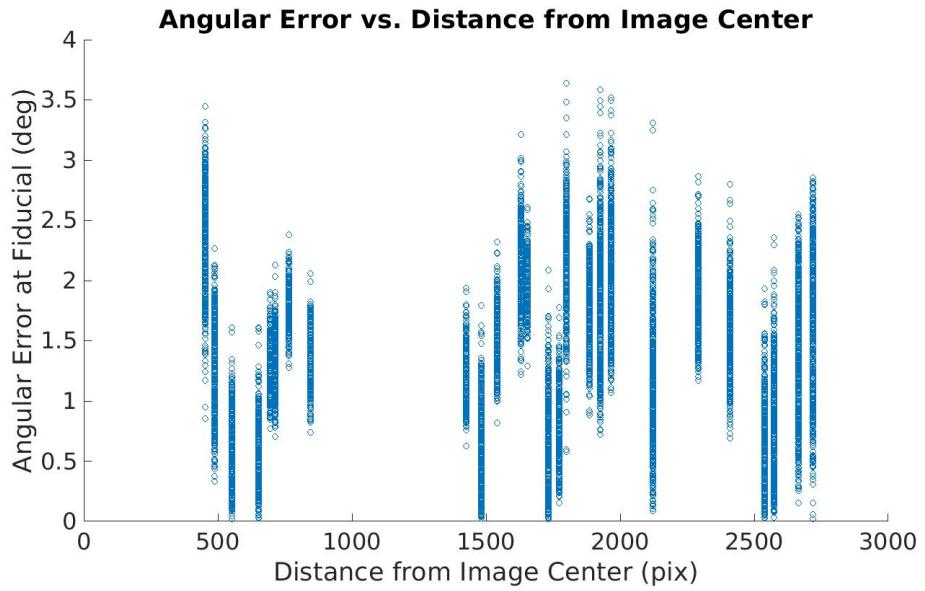


Figure 5.22: Post-calibration angular errors at each fiducial over 2000 random initializations.

Chapter 6

Conclusions and Future Work

The previous chapters have discussed the calibration of optical sensors for applications in autonomous navigation. The following chapter will provide a brief summary of the manuscript and outline potential future work in the area.

The use of three classes of optical sensors was discussed: perspective cameras, omnidirectional cameras, and LiDAR. Mathematical models, the *Brown Model* [2] and the *Taylor Model* [37], of both perspective cameras and omnidirectional cameras, respectively, were derived. Calibration procedures for perspective cameras were outlined. Goniometric calibration, though expensive, was shown to outperform calibration using stellar fields and greatly outperform calibration using checkerboards. A checkerboard technique for calibration of an omnidirectional camera was also described and shown to perform on a level similar to that of checkerboard calibration of a perspective camera. Finally, estimation of the rigid body transformation between a LiDAR and camera using a planar fiducial board was shown.

Much of the implementation of perspective cameras is not yet fully modelled using omnidirectional cameras. Supplemental work to this manuscript is needed to utilize the described omnidirectional image geometry in computer vision applications. Additionally, improvements on clustering techniques used to locate the LiDAR and image fiducials could

be investigated to improve the extrinsic calibration solution.

Bibliography

- [1] Christian, J. A., Benhacine, L., Hikes, J., and D’Souza, C., “Geometric Calibration of the Orion Optical Navigation Camera using Star Field Images,” *The Journal of the Astronautical Sciences*, Vol. 63, No. 4, 2016, pp. 335–353.
- [2] Brown, D., “Decentering Distortion of Lenses,” *Photogrammetric Engineering*, Vol. 32, No. 3, 1966, pp. 444–462.
- [3] Hikes, J., “A Summary of Calibration Techniques for Optical Navigation Cameras,” Tech. rep., 2018.
- [4] Gerhard, J., *A Geometric Hashing Technique for Star Pattern Recognition*, Master’s thesis, West Virginia University, 2016.
- [5] Brown, B., “The Social Life of Autonomous Cars,” *IEEE Computer*, Vol. 50, No. 2, 2017, pp. 92–96.
- [6] Mademlis, I., “Overview of Drone Cinematography for Sports Filming,” European Conference on Visual Media Production, 2017.
- [7] Manley, J. E. and Smith, J., “Rapid Development and Evolution of a Micro-UUV,” OCEANS, Anchorage, AK, 2017.
- [8] Ohi, N. and *et. al.*, “Design of an Autonomous Precision Pollination Robot,” IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain, 2018.

- [9] Nistér, D., “An Efficient Solution to the Five-Point Relative Pose Problem,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 6, 2004, pp. 756–770.
- [10] Caruso, D., Engel, J., and Cremers, D., “Large-Scale Direct SLAM for Omnidirectional Cameras,” *IEEE International Conference on Intelligent Robots and Systems*, 2015.
- [11] Shin, Y.-S., Park, Y. S., and Kim, A., “Direct Visual SLAM Using Sparse Depth for Camera-LiDAR Systems,” IEEE International Conference on Robotics and Automation, Brisbane, QLD, Australia, 2018.
- [12] Hassaballah, M., *Studies in Computational Intelligence: Image Feature Detection, Description and Matching*, Vol. 630, Springer, Cham.
- [13] Yilmaz, A., Javed, O., and Shah, M., “Object Tracking: A Survey,” *ACM Computing Surveys*, Vol. 38, No. 13, 2006.
- [14] Zorin, D. and Barr, A. H., “Correction of Geometric Perceptual Distortions in Pictures,” *Proceedings of SIGGRAPH*, 1995, pp. 257–264.
- [15] Kubovy, M., *The Psychology of Perspective and Renaissance Art*, Cambridge University Press, New York, 1986.
- [16] Rosen, J. and Ma, J., “Autonomous Operation in Surgical Robotics,” *ASME Mechanical Engineering*, Vol. 137, No. 9, 2015, pp. 15–18.
- [17] Christian, J. A. and Robinson, S., “Non-Iterative Horizon-Based Optical Navigation by Cholesky Factorization,” *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 12, 2016, pp. 2757–2765.
- [18] Golub, G. and Loan, C. F. V., *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 4th ed., 2012.

- [19] Fuentes-Pancheco, J., Ruiz-Ascencio, J., and Rendón-Mancha, J. M., “Visual Simultaneous Localization and Mapping: A Survey,” *Artificial Intelligence Review*, Vol. 43, No. 1, 2015, pp. 55–81.
- [20] Corke, P. I., “Spherical Image-Based Visual Servo and Structure Estimation,” IEEE International Conference on Robotics and Automation, Anchorage, AK, 2010.
- [21] Geyer, C. and Daniilidis, K., “A Unifying Theory for Central Panoramic Systems and Practical Applications,” Proceedings European Conference on Computer Vision, 2000.
- [22] Hansen, P., Boles, W., and Corke, P., “Spherical Diffusion for Scale Invariant Key-point Detection in Wide Angle Images,” *Digital Image Computing: Techniques and Applications*, 2008, pp. 525–532.
- [23] Geiger, A., Lenz, P., and Urtasun, R., “Are we Ready for Autonomous Driving? The Kitti Vision Benchmark Suite,” IEEE Conference on Computer Vision and Pattern Recognition, 2012.
- [24] Pandey, G., McBride, J. R., Savarese, S., and Eustice, R. M., “Automatic Extrinsic Calibration of Vision and LiDAR by Maximizing Mutual Information,” *Journal of Field Robotics, Special Issue on Calibration for Field Robotics*, Vol. 32, No. 5, 2015, pp. 696–722.
- [25] Scaramuzza, D., Harati, A., and Siegwart, R., “Extrinsic Self-Calibration of a Camera and a 3D Laser Range Finder from Natural Scenes,” IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007.
- [26] Quan, L. and Lan, Z., “Linear N-Point Camera Pose Determination,” *IEEE Transactions on PAMI*, Vol. 21, No. 7, 1999.

- [27] Zhang, Z., “Iterative Point Matching for Registration of Free-Form Curves,” *IRA Rapports de Recherche No 1658 Programme 4 Robotique, Image et Vision*, 1992.
- [28] Conrady, A., “Decentered Lens Systems,” *Monthly Notices of the Royal Astronomical Society*, Vol. 79, 1919, pp. 384–390.
- [29] Strong, J., *Concepts of Classical Optics*, W. H. Freeman and Co., 1958.
- [30] Weng, J., Cohen, P., and Herniou, M., “Camera Calibration with Distortion Models and Accuracy Evaluation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 10, 1992, pp. 965–980.
- [31] Wei, G.-Q. and Ma, S. D., “Implicit and Explicit Camera Calibration: Theory and Experiments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 5, 1994, pp. 469–480.
- [32] Gross, H., editor, *Handbook of Optical Systems: Aberration Theory and Correction of Optical Systems*, Vol. 3, Wiley-VCH Verlag GmbH & Co. KGaA, 2007.
- [33] Slama, C. C., Theurer, C., and Henriksen, S. W., editors, *Manual of Photogrammetry*, American Society of Photogrammetry, 4th ed., 1980.
- [34] Jenkins, F. A., *Fundamentals of Optics*, McGraw-Hill, Inc., 1976.
- [35] Longhurst, R. S., *Geometrical and Physical Optics*, Longman Group Limited, 3rd ed., 1973.
- [36] Benhacine, L., *Optical Aberrations and their Effect on the Centroid Location of Unresolved Objects*, Master’s thesis, West Virginia University, 2017.
- [37] Scaramuzza, D., *Omnidirectional Vision: From Calibration to Robot Motion Estimation*, Ph.D. thesis, ETH Zurich, 2008.

- [38] Descartes, R. and Smith, D., *The Geometry of Renè Descartes*, Dover, New York, Originally published in Discours de la Methode, 1637.
- [39] Nayar, S., “Catadioptric Omnidirectional Camera,” *IEEE CVPR’97*, 1997, pp. 482–488.
- [40] “A Theory of Catadioptric Image Formation,” *ICCV’98*, 1998, pp. 35–42.
- [41] Baker, S. and Nayar, S., “A Theory of Single-Viewpoint Catadioptric Image Formation,” *International Journal of Computer Vision*, Vol. 35, No. 2, 1999, pp. 175–196.
- [42] Park, J.-H., Baantseren, G., Kim, N., Park, G., Kang, J.-M., and Lee, B., “View Image Generation in Perspective and Orthographic Projection Geometry Based on Integral Imaging,” *Optics Express*, Vol. 16, No. 12, 2008, pp. 8800–8813.
- [43] Ying, X. and Hu, Z., “Can we Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model?” European Conference on Computer Vision, 2004, pp. 442–355.
- [44] Micusik, B., *Two View Geometry of Omnidirectional Cameras*, Ph.D. thesis, Center for Machine Perception, Czech Technical University, Prague, Czech Republic, 2004.
- [45] Muhammad, N. and Lacroix, S., “Calibration of a Rotating Multi-Beam LiDAR,” IEEE International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 2010.
- [46] Scaramuzza, D., Harati, A., and Siegwart, R., “Extrinsic Self Calibration of a Camera and a 3D Laser Range Finder from Natural Scenes,” IEEE International Conference on Intelligent Robots and Systems, San Diego, CA, 2007.
- [47] Bouguet, J.-Y., “Camera Calibration Toolbox for MATLAB®,” http://www.vision.caltech.edu/bouguetj/calib_doc/, 2015, Online; accessed 2016.

- [48] Remondino, F. and Fraser, C., “Digital Camera Calibration Methods: Considerations and Comparisons,” *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2006, pp. 266–272.
- [49] Rhodes, A., Christian, J. A., and Robinson, S., “Geometric Camera Calibration Using Near-Field Images of the ISS Centerline Docking Plate,” AAS/AIAA Astrodynamics Specialist Conference, Stevenson, WA, 2017.
- [50] Ahmed, M. and Farag, A., “Nonmetric Calibration of Camera Lens Distortion: Differential Methods and Robust Estimation,” *IEEE Transactions on Image Processing*, Vol. 14, No. 8, 2005, pp. 1215–1230.
- [51] Levenberg, K., “A Method for the Solution of Certain Non-Linear Problems in Least Squares,” *Quarterly of Applied Mathematics*, Vol. 2, No. 2, 1944, pp. 164–168.
- [52] Marquardt, D. W., “An Algorithm for Least-Squares Estimation of Nonlinear Parameters,” *Journal of the Society for Industrial and Applied Mathematics*, Vol. 11, No. 2, 1963, pp. 431–441.
- [53] Zhang, Z., “Flexible Camera Calibration by Viewing a Plane from Unknown Orientations,” 7th IEEE Conference on Computer Vision, Kerkyra, Greece, 1999.
- [54] Bradski, G. and Kaehler, A., *Learning OpenCV: Computer Vision with the OpenCV Library*, O’Reilly Media, Inc., 2008.
- [55] Szeliski, R., *Computer Vision: Algorithms and Applications*, Springer Science & Business Media, 2011.
- [56] Herrera, D., Kannala, J., and Heikkilä, J., “Joint Depth and Color Camera Calibration with Distortion Correction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 10, 2012, pp. 2058–2064.

- [57] Wöhler, C., *3D Computer Vision: Efficient Methods and Applications*, Springer Science & Business Media, 2013.
- [58] Shu, F., *High-Precision Calibration Approaches to Robot Vision Systems*, Ph.D. thesis, University of Hamburg, 2009.
- [59] Hieronymus, J., “Comparison of Methods for Geometric Camera Calibration,” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 39, 2012, pp. 595–599.
- [60] Clarke, T. A. and Fryer, J. G., “The Development of Camera Calibration Methods and Models,” *The Photogrammetric Record*, Vol. 16, No. 91, 1998, pp. 51–66.
- [61] Welch, S. S., “Effects of Window Size and Shape on Accuracy of Subpixel Centroid Estimation of Target Images,” Tech. rep., 2003.
- [62] Klaus, A., Bauer, J., Karner, K., Elbischger, P., Perko, R., and Bischof, H., “Camera Calibration from a Single Night Sky Image,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, D.C., 2004.
- [63] ESA, “The Hipparcos and Tycho Catalogues,” 1997, ESA SP-1200.
- [64] Dikmen, S., *Development of Star Tracker Attitude and Position Determination System for Spacecraft Maneuvering and Docking Facility*, Master’s thesis, Luleå University of Technology, 2016.
- [65] Thomas, M. and Joseph, R., “Astronomical Refraction,” *Johns Hopkins APL Tech. Dig.*, Vol. 17, No. 3, 1996.
- [66] Scaramuzza, D., “Omnidirectional Camera Calibration Toolbox for MATLAB[®],” <https://sites.google.com/site/scarabotix/ocamcalib-toolbox>, 2006, Online; accessed 2017.

- [67] Mei, C. and Rives, P., “Calibration Between a Central Catadioptric Camera and a Laser Range Finder for Robotic Applications,” ICRA, Orlando, FL, 2006.
- [68] Nemirovski, A. and Todd, M., “Interior-point Methods for Optimization,” *Acta Numerica*, Vol. 17, 2008, pp. 191–234.

Appendix A

Levenberg-Marquardt Routine for Perspective Camera Calibration

With the exception of a few notation changes and simplifications, this appendix is directly taken from the journal paper by Christian, Benhacine, and Hikes [1].

Define the camera model parameters as

$$\begin{aligned}\mathbf{k} &= \begin{bmatrix} d_x & \alpha & d_y & u_p & v_p \end{bmatrix}^T \\ \boldsymbol{\xi} &= \begin{bmatrix} k_1 & k_2 & k_3 & p_1 & p_2 \end{bmatrix}^T\end{aligned}$$

where the entire state vector is denoted by $\boldsymbol{\beta} = [\mathbf{k}^T \ \boldsymbol{\xi}^T]^T$. Then, we use the iterative LMA to solve the nonlinear estimation problem

$$\text{Minimize } J(\boldsymbol{\beta}) = \|\tilde{\mathbf{y}} - \mathbf{h}(\mathbf{e}_I, \boldsymbol{\beta})\|^2 \quad (\text{A.1})$$

where $\tilde{\mathbf{y}}$ is the matrix of all the measured centroids from the compiled image and $\mathbf{h}(\mathbf{e}_I, \boldsymbol{\beta})$ is the corresponding nonlinear measurement model (see Eq. 3.24). Observe that \mathbf{e}_I is a matrix formed by stacking the lines-of-sight found in Eq. 4.1 as columns.

Now, let us quickly define the Jacobian at the j -th iteration of the LMA algorithm as

$${}^{(j)}\mathbf{J} = \frac{\partial \mathbf{h}(\mathbf{e}_I, {}^{(j)}\boldsymbol{\beta})}{\partial {}^{(j)}\boldsymbol{\beta}} \quad (\text{A.2})$$

and \mathbf{N} as

$$\mathbf{N} = \mathbf{J}^T \mathbf{J} \quad (\text{A.3})$$

The equations to compute each of the entries in \mathbf{J} are presented in the following subsection. Finally, let $\text{diag}(\mathbf{N})$ define a square matrix the same size as \mathbf{N} that contains only the diagonal elements of \mathbf{N} and is zero elsewhere.

Thus, the LMA algorithm is as follows:

1. If available, get an initial guess for \mathbf{k} and $\boldsymbol{\xi}$ from previous calibrations.
2. Collect $M \cdot N$ images of a collimated source at known attitudes.
3. Construct an initial guess of the Jacobian, ${}^{(0)}\mathbf{J}$, from in Eq. A.2 and compute ${}^{(0)}\mathbf{N}$ from Eq. A.3.
4. Find the mean value of the diagonal entries of \mathbf{N} and define this as \bar{n} , then set ${}^{(0)}\lambda = 0.001\bar{n}$.
5. Solve the following linear system for $\boldsymbol{\delta}$

$$[{}^{(j)}\mathbf{N} + {}^{(j)}\lambda \text{diag}({}^{(j)}\mathbf{N})] \boldsymbol{\delta} = {}^{(j)}\mathbf{J}^T [\tilde{\mathbf{y}} - \mathbf{h}(\mathbf{e}_I, {}^{(j)}\boldsymbol{\beta})] \quad (\text{A.4})$$

6. Compute $\boldsymbol{\gamma} = {}^{(j)}\boldsymbol{\beta} + \boldsymbol{\delta}$, and compare the residuals $[\tilde{\mathbf{y}} - \mathbf{h}(\mathbf{e}_I, {}^{(j)}\boldsymbol{\beta})]$ and $[\tilde{\mathbf{y}} - \mathbf{h}(\mathbf{e}_I, \boldsymbol{\gamma})]$.

Then do one of the following:

- (a) If $\boldsymbol{\gamma}$ produces a reduction in the measurement residuals compared with ${}^{(j)}\boldsymbol{\beta}$, then $\boldsymbol{\gamma}$ constitutes an improvement to the estimate and so we set ${}^{(j+1)}\boldsymbol{\beta} = \boldsymbol{\gamma}$. Then reduce λ by an order of magnitude by setting ${}^{(j+1)}\lambda = {}^{(j)}\lambda/10$.
- (b) If $\boldsymbol{\gamma}$ produces a larger measurement residual compared with ${}^{(j)}\boldsymbol{\beta}$, then increase λ by an order of magnitude by setting ${}^{(j)}\lambda = 10 {}^{(j)}\lambda$ and return to Step # 5 and

try again.

7. Continue iteration until the residual $\|\tilde{\mathbf{y}} - \mathbf{h}(\mathbf{e}_I, {}^{(j)}\boldsymbol{\beta})\| < tol$ or a maximum number of iterations is reached.

The end result is the best estimate of $\boldsymbol{\beta}$, which includes all the camera calibration parameters.

Calculation of Jacobian Matrix at each LMA Iteration

This section presents the equations for all the terms in the Jacobian matrix required for the LMA algorithm. The Jacobian is defined as

$$\mathbf{J} = \frac{\partial \mathbf{h}(\mathbf{e}_I, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \begin{bmatrix} \frac{[\partial \mathbf{u}'_1]}{\partial \mathbf{k}} & \frac{[\partial \mathbf{u}'_1]}{\partial \boldsymbol{\xi}} \\ \frac{[\partial \mathbf{u}'_2]}{\partial \mathbf{k}} & \frac{[\partial \mathbf{u}'_2]}{\partial \boldsymbol{\xi}} \\ \vdots & \vdots \\ \frac{[\partial \mathbf{u}'_{MN}]}{\partial \mathbf{k}} & \frac{[\partial \mathbf{u}'_{MN}]}{\partial \boldsymbol{\xi}} \end{bmatrix} \quad (\text{A.5})$$

where $[\mathbf{u}'_i]$ describes the i -th star measurement in the image. The partials are as follows (and have been numerically checked via forward finite differencing),

$$\frac{\partial \mathbf{u}'_i}{\partial \mathbf{k}} = \begin{bmatrix} x'_i & y'_i & 0 & 1 & 0 \\ 0 & 0 & y'_i & 0 & 1 \end{bmatrix} \in \mathbb{R}^{2 \times 5} \quad (\text{A.6})$$

$$\frac{\partial \mathbf{u}'_i}{\partial \boldsymbol{\xi}} = \frac{\partial \mathbf{u}'_i}{\partial \mathbf{x}'_i} \frac{\partial \mathbf{x}'_i}{\partial \boldsymbol{\xi}} \in \mathbb{R}^{2 \times 5} \quad (\text{A.7})$$

The two partials that make up Eq. A.7 are as follows,

$$\frac{\partial \mathbf{u}'_i}{\partial \mathbf{x}'_i} = \begin{bmatrix} d_x & \alpha \\ 0 & d_y \end{bmatrix} \quad (\text{A.8})$$

$$\frac{\partial \mathbf{x}'_i}{\partial \boldsymbol{\xi}} = \begin{bmatrix} x_i r_i^2 & x_i r_i^4 & x_i r_i^6 & 2x_i y_i & (r_i^2 + 2x_i^2) \\ y_i r_i^2 & y_i r_i^4 & y_i r_i^6 & (r_i^2 + 2y_i^2) & 2x_i y_i \end{bmatrix} \quad (\text{A.9})$$