

**REDUCING THE VARIANCE OF INTRINSIC CAMERA CALIBRATION RESULTS IN
THE ROS CAMERA_CALIBRATION PACKAGE**

by

GEOFFREY NELSON CHIOU, B.S.

THESIS

Presented to the Graduate Faculty of
The University of Texas at San Antonio
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

COMMITTEE MEMBERS:

Pranav Bhounsule, Ph.D., Chair

Christopher Lewis, Ph.D.

Amir Jafari, Ph.D.

THE UNIVERSITY OF TEXAS AT SAN ANTONIO
College of Engineering
Department of Mechanical Engineering
December 2017

DEDICATION

This thesis is dedicated to my mother, who will surely ask me when I will be starting my Ph.D.

ACKNOWLEDGEMENTS

First, I would like to thank my advisor Dr. Pranav Bhounsule for always giving good advice and putting up with my antics for the past few years. I would also like to thank Dr. Christopher Lewis from Southwest Research Institute for mentoring me on camera calibration. Dr. Lewis took many hours out of his day to explain calibration concepts that would otherwise have taken months to understand. Alex Goins and Jonathan Meyer from SwRI were also instrumental in my success and completion of this project. Both would never hesitate to assist me in any problems or questions I had about calibration and programming. Lastly, I would like to thank Paul Hvass (formerly of SwRI) and Austin Deric from SwRI who managed my projects, and Dr. Harry Millwater from UTSA who introduced me to SwRI.

December 2017

REDUCING THE VARIANCE OF INTRINSIC CAMERA CALIBRATION RESULTS IN THE ROS CAMERA_CALIBRATION PACKAGE

ABSTRACT

Geoffrey Chiou, M.S.
The University of Texas at San Antonio, 2017

Supervising Professor: Pranav Bhounsule, Ph.D.

The intrinsic calibration of a camera is the process in which the internal optical and geometric characteristics of the camera are determined. If accurate intrinsic parameters of a camera are known, the ray in 3D space that every point in the image lies on can be determined. Pairing with another camera allows for the position of the points in the image to be calculated by intersection of the rays. Accurate intrinsics also allow for the position and orientation of a camera relative to some world coordinate system to be calculated. These two reasons for having accurate intrinsic calibration for a camera are especially important in the field of industrial robotics where 3D cameras are frequently mounted on the ends of manipulators.

In the ROS (Robot Operating System) ecosystem, the camera_calibration package is the default standard for intrinsic camera calibration. Several researchers from the Industrial Robotics & Automation division at Southwest Research Institute have noted that this package results in large variances in the intrinsic parameters of the camera when calibrating across multiple attempts. There are also open issues on this matter in their public repository that have not been addressed by the developers. In this thesis, we confirm that the camera_calibration package does indeed return different results across multiple attempts, test out several possible hypotheses as to why, identify the reason, and provide simple solution to fix the cause of the issue.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
LIST OF TABLES	vi
LIST OF FIGURES	vii
CHAPTER ONE: INTRODUCTION.....	1
1.1 EXISTING CALIBRATION FRAMEWORKS IN ROS.....	3
1.2 THESIS CONTRIBUTIONS.....	4
1.3 OUTLINE OF THESIS.....	4
CHAPTER TWO: BACKGROUND AND LITERATURE REVIEW	6
2.1 PINHOLE CAMERA MODEL	6
2.2 DISTORTION PARAMETERS	9
2.3 INTRINSIC CAMERA CALIBRATION PROCESS	12
2.5 VERIFYING RESULTS OF A CALIBRATION.....	14
2.5 HYPOTHESIS	15
CHAPTER THREE: METHODOLOGY	18
3.1 DATA COLLECTION	18
3.2 DATA ANALYSIS PROCESS	22
CHAPTER FOUR: RESULTS AND DISCUSSION.....	24
4.1 DATA SET ONE RESULTS AND DISCUSSION.....	24
4.2 DATA SET TWO RESULTS AND DISCUSSION.....	26
4.3 DATA SET THREE RESULTS AND DISCUSSION.....	28
CHAPTER FIVE: CONCLUSION.....	31
5.1 CONCLUSION.....	31
5.2 FUTURE WORK.....	32
VITA	

LIST OF TABLES

Table 1. Intrinsic Parameters of Calibration for Data Set 1 using OpenCV Full	24
Table 2. Intrinsic Parameters of Calibration for Data Set 1 using Ceres	24
Table 3. Intrinsic Parameters of Calibration for Data Set 1 using Ceres with 2-Step Solve	25
Table 4. Intrinsic Parameters of Calibration for Data Set 2 using OpenCV	26
Table 5. Intrinsic Parameters of Calibration for Data Set 2 using Ceres	27
Table 6. Intrinsic Parameters of Calibration for Data Set 2 using Ceres with 2-Step Solve	27
Table 7. Intrinsic Parameters of Calibration for Data Set 3 using OpenCV	28
Table 8. Intrinsic Parameters of Calibration for Data Set 3 using Ceres	29
Table 9. Intrinsic Parameters of Calibration for Data Set 2 using Ceres with 2-Step Solve	29
Table 11. Intrinsic Parameters of Calibration for Data Set 1 using OpenCV Full	33
Table 12. Intrinsic Parameters of Calibration for Data Set 1 using Ceres Full	33
Table 13. Intrinsic Parameters of Calibration for Data Set 1 using Ceres 2-Step Solve Full	34
Table 14. Intrinsic Parameters of Calibration for Data Set 2 using OpenCV Full	35
Table 15. Intrinsic Parameters of Calibration for Data Set 2 using Ceres Full	36
Table 16. Intrinsic Parameters of Calibration for Data Set 2 using Ceres 2-Step Solve Full	37
Table 17. Intrinsic Parameters of Calibration for Data Set 3 using OpenCV Full	38
Table 18. Intrinsic Parameters of Calibration for Data Set 3 using Ceres Full	39
Table 19. Intrinsic Parameters of Calibration for Data Set 2 using Ceres 2-Step Solve Full	40
Table 20. Intrinsic Parameters of Calibration for Data Set 3 using Ceres Circle Correction Full	41

LIST OF FIGURES

Figure 1. Diagram of an Extrinsic Calibration.....	2
Figure 2. Diagram of Ideal Pinhole Camera	6
Figure 3. Diagram Following a Ray of Light in the Pinhole Camera Model	7
Figure 4. Visualization of Two Different Types of Radial Distortion.....	9
Figure 5. Diagram Explaining What Causes Tangential Distortion	10
Figure 6. Residual Vectors of Radial Distortion Model	16
Figure 7. Residual Vectors of Tangential Distortion Model.....	17
Figure 8. Asus Xtion Pro Depth Sensor.....	18
Figure 9. 9 x 12 Modified Circle Grid Calibration Pattern.....	18
Figure 10. Screenshot of camera_calibration User Interface.....	19
Figure 11. 11 x 15 Modified Circle Grid Calibration Pattern Mounted on a Wall.....	20
Figure 12. Circle Grid Finder with Dots Drawn at the First, Origin, and Last Point	21
Figure 13. Plot of Focal Length Results of Xtion Intrinsic Calibration Using Three Methods....	25
Figure 14. Plot of Focal Length Results of Xtion Intrinsic Calibration Across All Data Sets	29

CHAPTER ONE: INTRODUCTION

Industrial automation has become ubiquitous with long assembly lines of robotic manipulators assembling cars. These manipulators are all programmed to do a specific task repeatedly at each stage of the manufacturing process. If a car arrives in a state unknown to the robot, it cannot adapt. While this is fine for assembling cars, what if the robot needs to remove casting defects from a cast iron part that is different every time? The next level of industrial automation involves giving these manipulators vision. Sensors such as 2D and 3D cameras can be mounted on the robot to provide it with a set of eyes. Through vision processing, these sensors allow the robot to adapt to a wide variety of situations depending on what the camera perceives.

If a camera is attached to the end of a manipulator, the location of the camera housing relative to the robot can be measured. However, images taken by the camera bear no relation to the housing. Therefore, finding the relationship between the optical frame of the camera and robot is important. The optical frame of the robot is located at the focal point, which exists somewhere inside or outside the camera housing and cannot be easily measured. Establishing a common frame of reference between the optical frame of the camera and robot is done through a process called extrinsic calibration. Extrinsic calibration of a camera to a robot is used to determine the position and orientation of a camera optical frame relative to a frame of reference known to the robot. Mounting a camera on a robot allows the camera to be moved by robot and scan oddly shaped parts. Figure 1 provides an example of what an extrinsic calibration might look like when a camera is mounted on the tool of the robot. In this case, extrinsic calibration is used to solve for the relationship between the tool frame and camera optical frame. To perform an extrinsic calibration, a robot with a camera mounted on the tool will move around and take several images of a

calibration pattern while recording the joint states of the robot at every image. Every point on the target is known in the frame of the target. Those points are transformed into the base frame of the robot using an educated guess of the transform between the target frame and base frame. The transform between the base frame of the robot and the tool frame is known through the kinematics of the robot. The target points are then transformed into the tool frame of the robot. Finally, using a guess for the transform between the tool frame and camera optical frame, the target points are transformed into the camera optical frame.

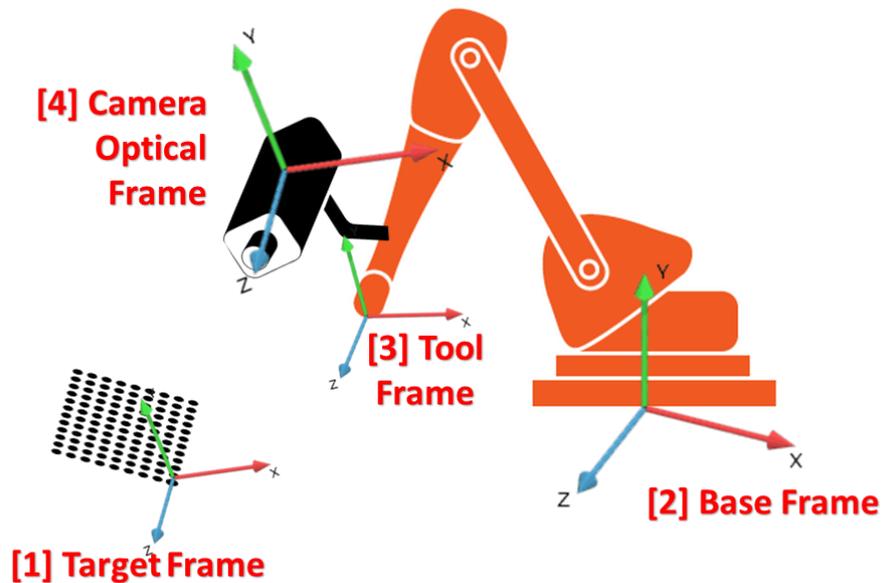


Figure 1. Diagram of an Extrinsic Calibration

Through a process called perspective projection, the target points are projected into a two-dimensional image plane. These points are compared to the points in the images captured earlier. The residual between the projected point and observed point is minimized by refining the transform between the target and base and the transform between the tool and camera optical frame.

Performing a perspective projection requires the intrinsic parameters of a camera. Therefore, obtaining an accurate extrinsic calibration requires an accurate set of intrinsic parameters, which includes the camera projection model and distortion coefficients. Since the camera projection model relates the 2D points in the image to the 3D points in the scene, an incorrect model will result in an incorrect extrinsic calibration. The intrinsic parameters are obtained through a process called intrinsic camera calibration. Intrinsic camera calibration is solved using bundle adjustment, which takes known 3D feature locations, projects them into the 2D image plane using the camera parameters, and minimizes the reprojection error by refining the camera parameters.

This thesis will explore our attempts to obtain a consistent and accurate intrinsic calibration using the *camera_calibration* package [1] that ships with ROS (Robot Operating System) and improving upon those results using the *industrial_calibration* package [2] which is published by ROS-I (ROS-Industrial).

1.1 EXISTING CALIBRATION FRAMEWORKS IN ROS

In the ROS ecosystem, there are two main packages that provide tools for intrinsic camera calibration. The first and most commonly used is the *camera_calibration* package [1], which exists in the *image_pipeline* repository published by ROS-Perception. The *camera_calibration* package provides a user interface that allows users to collect calibration images automatically and indicates when sufficient data has been collected through progress bars. Then it passes the data to an OpenCV backend which processes the data and runs the non-linear least squares optimization algorithm to calibrate the intrinsic parameters [2]. The *camera_calibration* page has been shown to provide results with high variances in focal length and principal point across multiple calibrations.

The second package is called *industrial_calibration* and is published by the ROS-Industrial Consortium [3]. This library provides a wide assortment of tools and cost functions to perform intrinsic and extrinsic calibrations that would be useful in the field of industrial robotics. Like OpenCV, the *industrial_calibration* package also calibrates through solving a non-linear least squares optimization problem. This package differs from OpenCV's implementation through the fact that it uses Ceres-Solver as a back end. Ceres-Solver is an open source C++ library initially developed by Google to solve large scale bundle adjustment problems [4]. This package was originally written to perform extrinsic calibrations, but after discovering that the *camera_calibration* package produced large variances in the focal length and principal points across multiple calibrations of the same sensor, Dr. Lewis decided to develop a new calibration method using a linear rail.

1.2 THESIS CONTRIBUTIONS

During the completion of this thesis, an open source fork of the *industrial_calibration* package temporarily called *IC2* was developed to separate the calibration functional into a C++ library separate from ROS. This gave us the flexibility to run many different experiments on calibration images. We also developed several programs to sort through calibration images and remove images that suffered from poor feature detection. A tool was also written to collect calibration images manually. The findings in thesis also allowed us to provide instructions on how to achieve consistent results when using the *camera_calibration* page.

1.3 OUTLINE OF THESIS

The first thing we do is review the background knowledge required to understand how intrinsic calibration of a camera is performed. This includes review of the pinhole camera model, how a perspective projection works, the distortion model for lenses. Then we discuss a few important papers that define the current process. Finally, we will define our initial hypothesis and the methodology used to collect data. Finally we explain how we analyzed the data, and discuss our results.

CHAPTER TWO: BACKGROUND AND LITERATURE REVIEW

2.1 PINHOLE CAMERA MODEL

The pinhole camera model is a model of an ideal pinhole camera as shown in Figure 2. The idea pinhole camera is essentially a box where light cannot enter except through a small hole with no lens called the aperture. When rays of light pass through the aperture, they project an inverted image on the opposite of the hole [5].

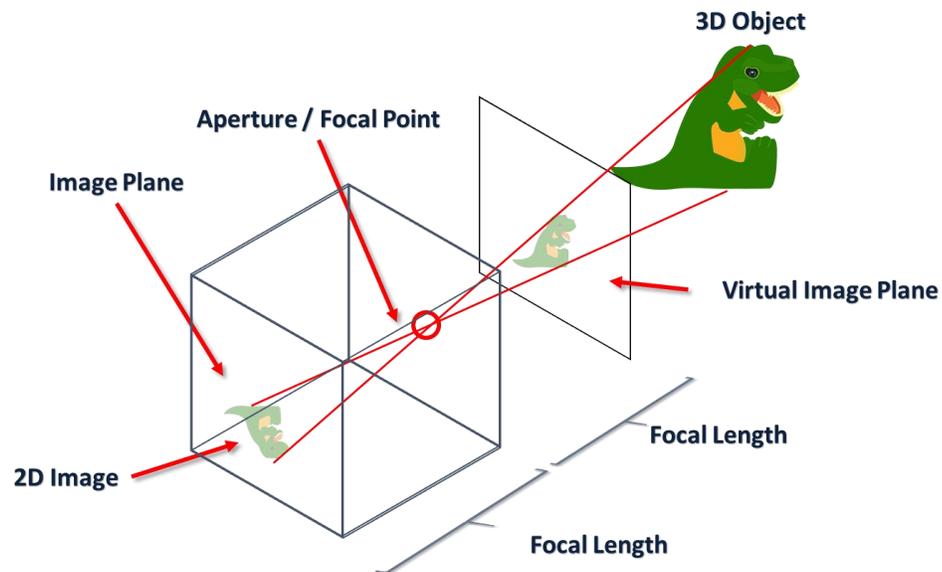


Figure 2. Diagram of Ideal Pinhole Camera

The pinhole camera model describes the relationship between the three-dimensional scene and how they are projected into the two-dimensional image plane with a perspective transformation [6]. The diagram below describes the pinhole camera model.

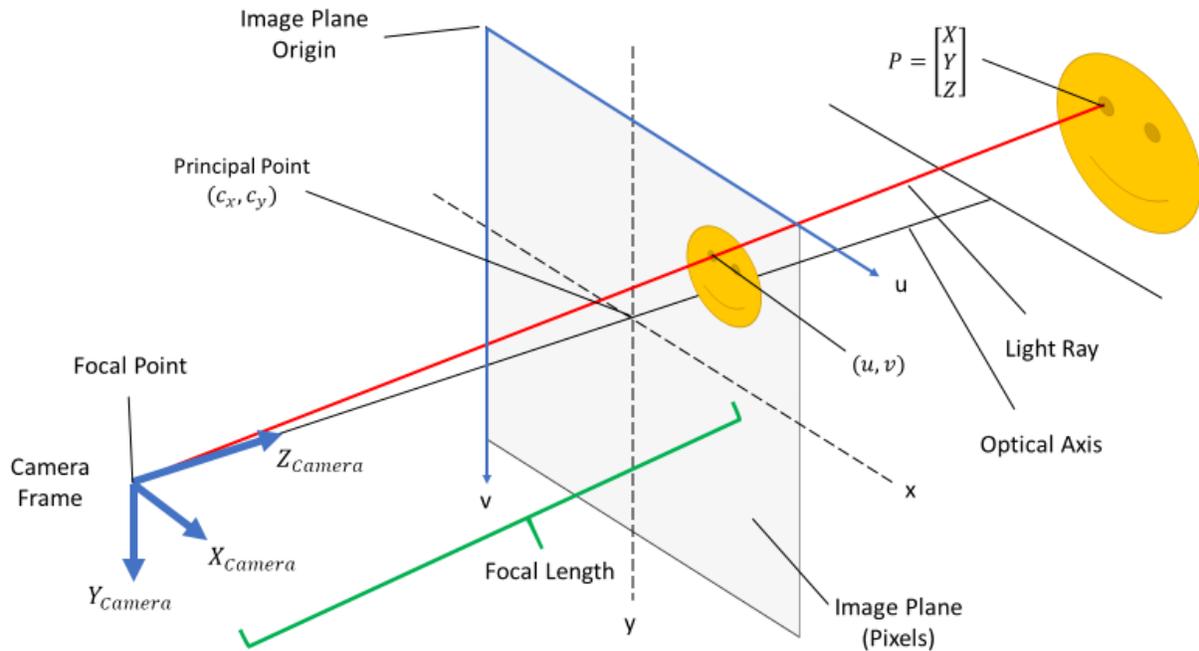


Figure 3. Diagram Following a Ray of Light in the Pinhole Camera Model

Starting from the right side of Figure 3, a ray of light from the 3D point P with coordinates of $[X, Y, Z]^T$ is projected into the image plane to the point (u, v) in pixels with a perspective projection. The line normal to the image plane is called the optical axis and is collinear with Z_{camera} . The intersection between the optical axis and the image plane is called the principal point. The light rays all eventually intersect at the camera frame. The distance between the focal point and the image plane is the focal length [7].

The intrinsic parameters of an ideal pinhole camera can be described with the camera matrix shown below.

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

From the camera matrix: f_x and f_y are the focal lengths of the x and y in pixels. These are derived from the focal length in millimeters multiplied by the size of size of the pixel in pixels per millimeter in the x and y directions respectively. In an idea pinhole camera, these values should be the same. In modern cameras, these values will be very close but can differ for a variety of reasons. Different values of f_x and f_y results on non-square pixels. c_x and c_y describe the location of the principal point offset, which is the location of the principal point relative to the image frame. The perspective projection between the 3D point P from the scene and the 2D image plane is given by the following equation [6].

$$m' = A[R|t]M'$$

In this perspective projection: m' is the point in 2D image coordinates, M' is the point in 3D scene coordinates, A is the matrix of intrinsic camera parameters, and R and t describe the rotation and translation of the camera, also known as the extrinsic parameters. The R and t matrices transform points from P into the coordinate system of the camera. Expanding the equation above where $m' = [u, v, 1]^T$ and $P = M' = [X, Y, Z, 1]^T$ gives the following:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

The above equation can be expanded and reorganized into something easier to understand. The X, Y, and Z points of the scene are rotated and translated into the camera frame. Then those points are scaled. The scaled points are multiplied by the focal length and added to the principal point. The equations below show this simplification [6].

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

$$\text{If } z \neq 0 \rightarrow x' = \frac{x}{z}, \quad \text{Else} \rightarrow x' = x$$

$$\text{If } z \neq 0 \rightarrow y' = \frac{y}{z}, \quad \text{Else} \rightarrow y' = y$$

$$u = f_x x' + c_x$$

$$v = f_y y' + c_y$$

2.2 DISTORTION PARAMETERS

Most real cameras require a lens to focus light into the image sensor. The ideal pinhole camera does not have a lens; therefore, the camera model is not an accurate representation of an actual camera since it does not account for lens distortion. The distortion caused by a lens can be modeled as radial distortion and tangential distortion. For most lenses, the distortion is mostly radial and a tiny bit of tangential distortion.

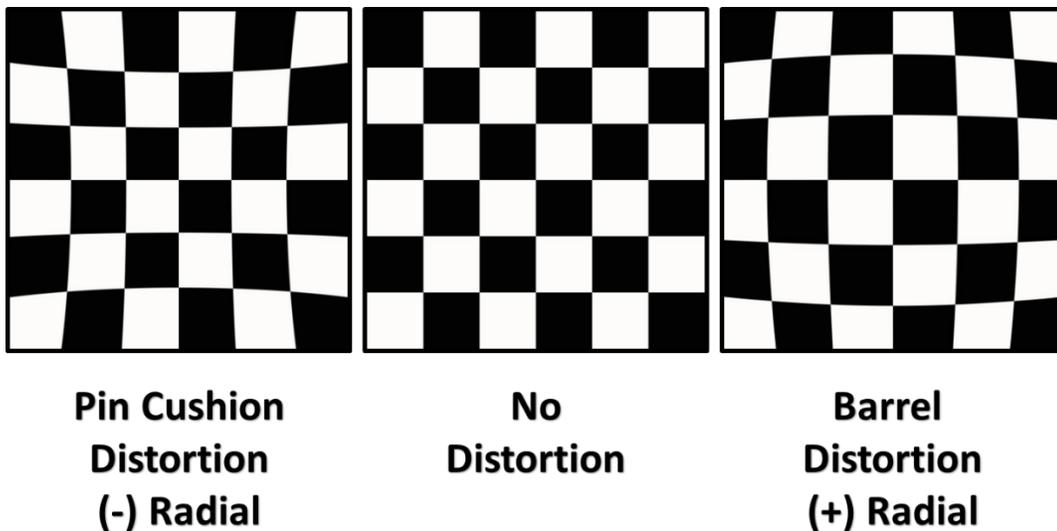


Figure 4. Visualization of Two Different Types of Radial Distortion

The two types of radial distortion are known as pincushion and barrel. A pincushion radial distortion is a negative amount of radial distortion where the light rays near the edge of the image

are bent towards the optical center. A barrel radial distortion is a positive amount of radial distortion where the light rays near the edge of the image are bent away from the optical center [5]. The figure above shows an example of the two types of radial distortion simulated on a checkerboard. The checkerboard on the left is an example of a pincushion distortion, the checkerboard in the center has no radial distortion, and the checkerboard on the right has barrel distortion. The two distorted images were generated with Adobe Photoshop® by taking the original checkerboard image and correcting for lens distortion. Adding a positive correction resulted in a negative pin cushion distortion and adding a negative correction resulted in a positive barrel distortion.

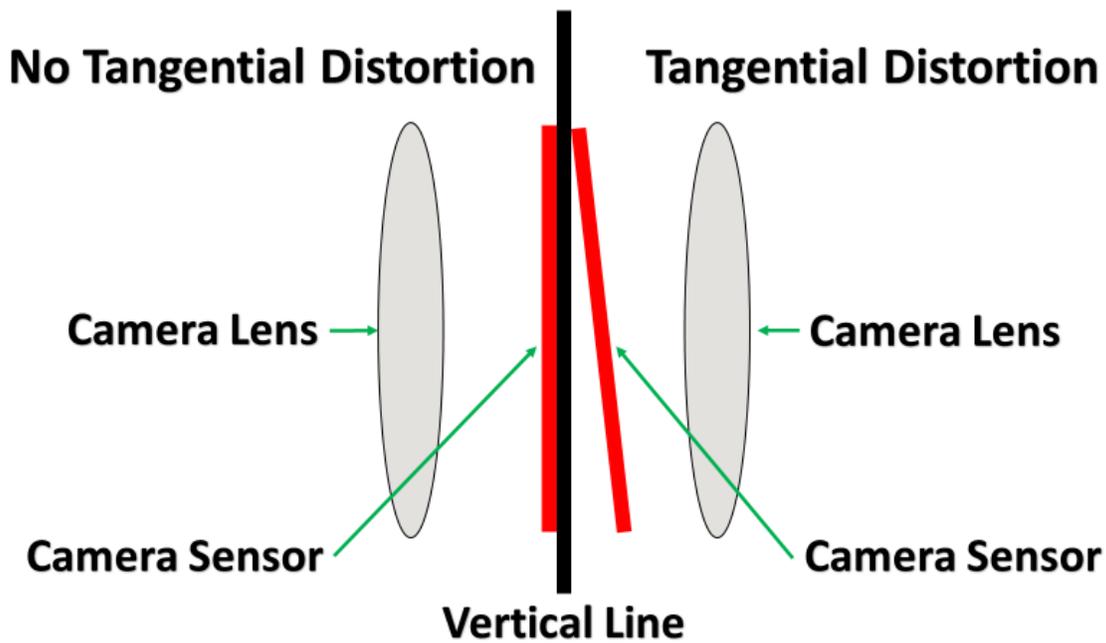


Figure 5. Diagram Explaining What Causes Tangential Distortion

Tangential distortion is caused by having a lens that is not parallel to the image plane [5]. The figure above shows an example of when tangential distortion can occur. On the left side of the figure, the lens is parallel to the image plane. This results in a scenario where the image only has

radial distortion and zero tangential distortion. The right side of the figure has an image plane that is not parallel to the lens. This situation can manifest when the physical elements in the lens are not aligned properly. In this case, tangential distortion exists along with radial distortion. Brown calls this decentering distortion, which introduces both tangential distortion and asymmetric radial distortion [8].

Correcting for radial and tangential distortion is performed using the Brown—Conrady distortion model [8]. The radial components are fit to a n th order polynomial where n is a multiple of two. In most cases, the radial distortion is dominated by the lower order polynomials with the higher order coefficients having a negligible effect. OpenCV allows for a 12th order polynomial fit, which introduces six radial distortion coefficients. The Industrial Calibration package allows for a 6th order fit which introduces three radial distortion coefficients but can easily be extended to allow for a higher order if required. Zhang states that most of the distortion is dominated by the first distortion coefficient, and more elaborate models would introduce numerical instability [9]. In [9], Zhang only solves for the first two radial distortion coefficients and none of the tangential distortion coefficients.

In this thesis, we modeled the radial distortion with a 6th order polynomial which introduces the radial distortion coefficients K_1 , K_2 , and K_3 . If we let $r^2 = (x')^2 + (y')^2$ where x' and y' are the undistorted points, the model for the distorted point only accounting for radial distortion produces the equations [5]:

$$x_{distorted} = x'(1 + K_1r^2 + K_2r^4 + K_3r^6)$$

$$y_{distorted} = y'(1 + K_1r^2 + K_2r^4 + K_3r^6)$$

Using the Brown—Conrady model to correct for tangential distortion introduces the tangential distortion coefficients P1 and P2. The correction equation for only tangential distortion produces the equations:

$$x_{distorted} = x' + [2P_1x'y' + P_2(r^2 + 2(x')^2)]$$

$$y_{distorted} = y' + [P_1(r^2 + 2(y')^2) + 2P_2x'y']$$

The radial and tangential distortion corrections can be joined into a single equation for each of the x and y as shown below:

$$x'' = \underbrace{x' + K_1r^2x' + K_2r^4x' + K_3r^6x'}_{Radial} + \underbrace{[2P_1x'y' + P_2(r^2 + 2(x')^2)]}_{Tangential}$$

$$y'' = \underbrace{y' + K_1r^2y' + K_2r^4y' + K_3r^6y'}_{Radial} + \underbrace{[P_1(r^2 + 2(y')^2) + 2P_2x'y']}_{Tangential}$$

The x' and y' values from the perspective transform equation shown earlier can be replaced with x'' and y'' which results in the new perspective transform which accounts for distortion:

$$u = f_x x'' + c_x$$

$$v = f_y y'' + c_y$$

2.3 INTRINSIC CAMERA CALIBRATION PROCESS

In Zhang's paper [9], he breaks the calibration procedure into five easy steps summarized below.

1. Print out a calibration pattern and attach it to a flat surface.
2. Take a few images of the calibration from different orientations by moving the camera or pattern.
3. Detect the features of the calibration target from the images. For a chessboard, this would be the corners, for a circle grid, this would be the centers of the circles.

4. Estimate the intrinsic and extrinsic parameters of the camera for each image using closed form solutions he derived.
5. Estimate the complete set of intrinsic parameters and distortion coefficients by minimizing the following function:

$$\sum_{i=1}^n \sum_{j=1}^m \left\| m_{ij} - \tilde{m}(A, K_1, K_2, R_i, t_i M_j) \right\|^2$$

which is a nonlinear minimization problem solved using the Levenberg-Marquardt Algorithm (LMA). The initial guess for the K_1 and K_2 parameters can be set to 0.

In our case, we used a modified version of the minimization function that includes the third radial distortion coefficient and the two tangential distortion coefficients. The function was solved with the LMA solver included in Ceres-Solver [4] instead of *Minpack* used by Zhang [9]. The function is shown below.

$$\sum_{i=1}^n \sum_{j=1}^m \left\| m_{ij} - \tilde{m}(A, K_1, K_2, P_1, P_2, K_3, R_i, t_i M_j) \right\|^2$$

In case it was not clear from looking at the formula, n represents the number of images and m represents the number of features in the calibration pattern from each image. For example, if 10 images of a calibration pattern were taken and the pattern had a circle grid of 10 x 10 dots, then n would equal 10, and m would be 100. Since the locations of each feature point are known (M_j), \tilde{m} is the projection of M into the image plane while accounting for distortion using the equations shown in the previous section. The algorithm loops through every image and every feature point on every image and minimizes the residual (expected value – actual value) by adjusting the camera matrix, distortion parameters, and extrinsic parameters of the camera.

2.5 VERIFYING RESULTS OF A CALIBRATION

Upon completion of a calibration, the final average reprojection error of the calibration is computed as the final cost of the calibration divided by the total number of observations. The average reprojection error can be used to give an estimation on the precision of the calibrated intrinsic parameters. Although a large reprojection can be interpreted as having a bad calibration, a low reprojection error does not necessarily mean the calibration is good. Determining if the solution of the non-linear least squares calibration is of good quality requires the generation of the covariance of the solution. Ceres-Solver conveniently allows a user to easily compute the blocks of the covariance they need instead of generating the entire covariance matrix which can be extremely time consuming. I will not go into detail on how the covariance is estimated but Ceres-Solver explains it in their documentation for those who are interested [4]. For purposes of camera calibration, we only need to interpret the results of the covariance matrix.

The covariance matrix we refer to earlier is a variance-covariance matrix which is symmetric. The diagonals of the matrix are the variances (square of standard deviation) for each of the variables, which indicates how much the data is scattered about the mean. The off-diagonal values of the matrix are the covariance of each pair of variables, which is measure of how the two variables change together. When both variables increase or decrease together, the covariance has a positive value. If one variable increases as the other decreases, the covariance will have a negative value. An example of the variance-covariance matrix of the intrinsic parameters is shown below.

	f_x	f_y	c_x	c_y	k_1	k_2	k_3	p_1	p_2
f_x	$\sigma_{f_x}^2$	$c(f_x, f_y)$	$c(f_x, c_x)$	$c(f_x, c_y)$	$c(f_x, k_1)$	$c(f_x, k_2)$	$c(f_x, k_3)$	$c(f_x, p_1)$	$c(f_x, p_2)$
f_y	$c(f_y, f_x)$	$\sigma_{f_y}^2$	$c(f_y, c_x)$	$c(f_y, c_y)$	$c(f_y, k_1)$	$c(f_y, k_2)$	$c(f_y, k_3)$	$c(f_y, p_1)$	$c(f_y, p_2)$
c_x	$c(c_x, f_x)$	$c(c_x, f_y)$	$\sigma_{c_x}^2$	$c(c_x, c_y)$	$c(c_x, k_1)$	$c(c_x, k_2)$	$c(c_x, k_3)$	$c(c_x, p_1)$	$c(c_x, p_2)$
c_y	$c(c_y, f_x)$	$c(c_y, f_y)$	$c(c_y, c_x)$	$\sigma_{c_y}^2$	$c(c_y, k_1)$	$c(c_y, k_2)$	$c(c_y, k_3)$	$c(c_y, p_1)$	$c(c_y, p_2)$
k_1	$c(k_1, f_x)$	$c(k_1, f_y)$	$c(k_1, c_x)$	$c(k_1, c_y)$	$\sigma_{k_1}^2$	$c(k_1, k_2)$	$c(k_1, k_3)$	$c(k_1, p_1)$	$c(k_1, p_2)$
k_2	$c(k_2, f_x)$	$c(k_2, f_y)$	$c(k_2, c_x)$	$c(k_2, c_y)$	$c(k_2, k_1)$	$\sigma_{k_2}^2$	$c(k_2, k_3)$	$c(k_2, p_1)$	$c(k_2, p_2)$
k_3	$c(k_3, f_x)$	$c(k_3, f_y)$	$c(k_3, c_x)$	$c(k_3, c_y)$	$c(k_3, k_1)$	$c(k_3, k_2)$	$\sigma_{k_3}^2$	$c(k_3, p_1)$	$c(k_3, p_2)$
p_1	$c(p_1, f_x)$	$c(p_1, f_y)$	$c(p_1, c_x)$	$c(p_1, c_y)$	$c(p_1, k_1)$	$c(p_1, k_2)$	$c(p_1, k_3)$	$\sigma_{p_1}^2$	$c(p_1, p_2)$
p_2	$c(p_2, f_x)$	$c(p_2, f_y)$	$c(p_2, c_x)$	$c(p_2, c_y)$	$c(p_2, k_1)$	$c(p_2, k_2)$	$c(p_2, k_3)$	$c(p_2, p_1)$	$\sigma_{p_2}^2$

There is no method that we know of to easily produce the exact intrinsic parameters of a camera to compare against our solution. However, there are ways to check the results. One method used by Chris Lewis from Southwest Research Institute [3] is to mount the camera on a linear rail placed perpendicular to a calibration pattern and capture two images of the calibration pattern with a known distance traveled between both images. Using the calibrated intrinsic parameters, the algorithm would solve for the t vector. The accuracy of the calibration is determined by subtracting the t vector of the first image from the t vector from the second image. This is the distance that it theoretically moved. Compare this to the known value that the linear rail moved and you obtain the percent error between the two distances traveled.

2.5 HYPOTHESIS

The motivation for this research arose when engineers from the Robotics and Automation Engineering section at Southwest Research Institute had trouble obtaining accurate and consistent calibration results when performing intrinsic camera calibration using the ROS *camera_calibration* package which is the standard package for camera calibration in the ROS ecosystem. One engineer stated that it required hundreds of images of the calibration pattern to achieve a good calibration. Chris Lewis performed multiple calibrations on a single camera and

obtained results high variances in the center point and focal length. This led to many man hours and dollars spent on developing new calibration techniques.

The hypothesis theorized that the method in which we solved for the intrinsic parameters of the camera was flawed. A vector field of the residuals for the radial components of the distortion model is shown in the Figure 8. The residual field shows that the residuals are indeed radial. However, when we plot the residual field for the tangential distortion model shown in Figure 9, there are also radial components. This led to the hypothesis that the method in which we solve for the distortion parameters is incorrect. Instead of solving for all 9 intrinsic parameters at once, the theory is to solve for the focal lengths, principal points, and radial distortion coefficients first, removing all radial components from the tangential coefficients, then hold those parameters constant and solve for the tangential distortion coefficients second. This should theoretically decrease the amount of variance in the principal point and focal lengths since solving for the principal point requires some radial distortion because the radial distortion emanates from the principal point.

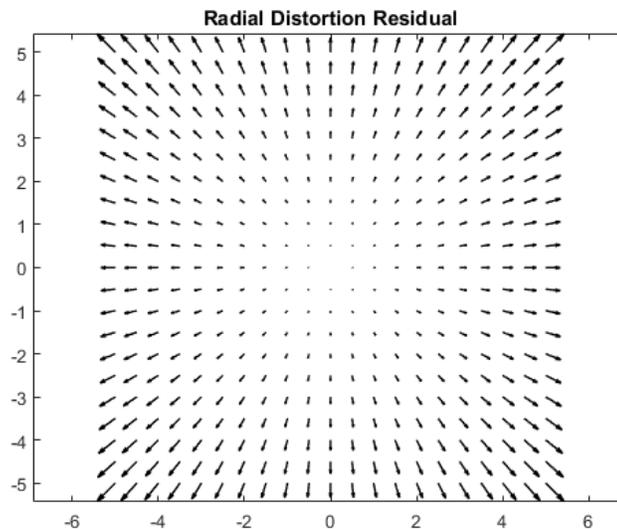


Figure 6. Residual Vectors of Radial Distortion Model

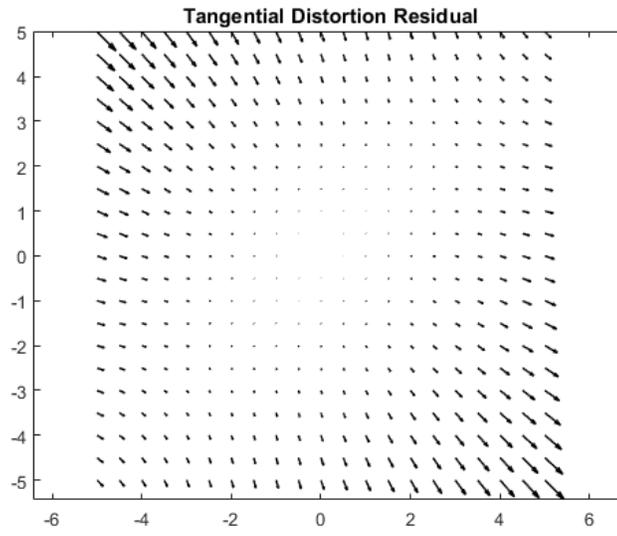


Figure 7. Residual Vectors of Tangential Distortion Model

CHAPTER THREE: METHODOLOGY

3.1 DATA COLLECTION

All of the calibration image data in this research was collected using a single camera. We chose to use the RGB sensor inside of the Asus Xtion Pro shown below [10]. The Xtion produces an RGB image of 640 by 480 pixels and was chosen as the camera for this research for a few reasons. The camera communicates with the computer through a USB interface (no need for POE) and easy to use existing ROS drivers. This camera is also readily available in the lab and sees lots of use in the industry due to the low cost.



Figure 8. Asus Xtion Pro Depth Sensor

The first data set was collected automatically using the GUI that ships with *camera_calibration* package. We used a 9 by 12 modified circle grid calibration pattern shown below. When printed, the calibration pattern had a circle diameter of 0.124 meters and a circle spacing of 0.01779 meters.

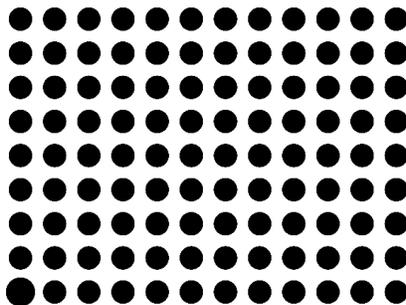


Figure 9. 9 x 12 Modified Circle Grid Calibration Pattern

The user interface for the *camera_calibration* package is a view window showing the camera feed with meters on the side that fill up as data is collected. The instructions [1] indicate that the calibration pattern needs to be placed in the camera's left, right, top and bottom fields of view. The X bar indicates when there is enough data in the left and right, the Y bar indicates when there is enough data in the top and bottom, and the size bar indicates when there is enough data towards and away from the camera as well as tilt. The instructions also indicate that the pattern should fill the entire field of view at some point. As the pattern is moved around the field of view of the camera, the indicator bars will increase in length, and a *calibrate* button lights up when the software determines that there is enough data for a calibration. The image below is a screenshot of the user interface described above taken from the *camera_calibration* tutorial on the ROS wiki.

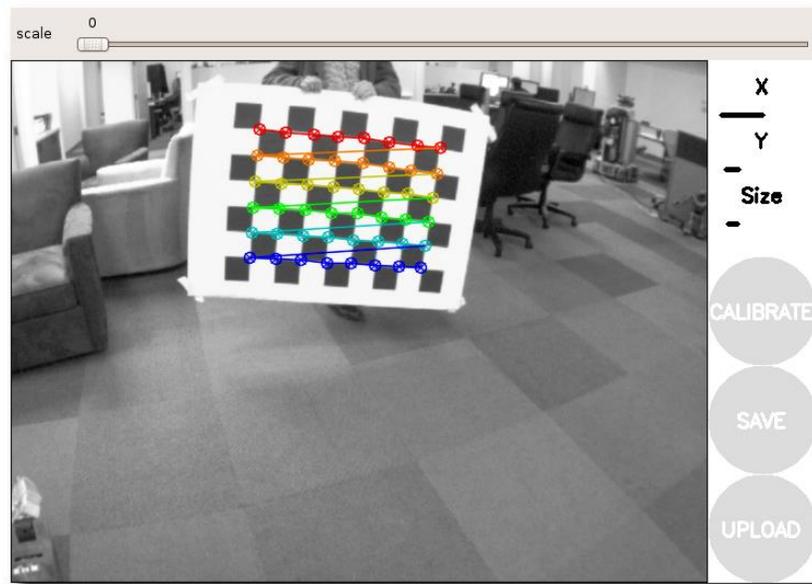


Figure 10. Screenshot of *camera_calibration* User Interface

The first data set included 15 sets of images collected automatically using the *camera_calibration* GUI. The calibration pattern was taped to a piece of cardboard and attached

to a tripod. We attached the Xtion to a UR3 collaborative robot placed in front of the calibration pattern and dragged the UR3 to different poses by hand until the meters of the calibration GUI filled and the calibration button appeared. In total, the UR3 was taught about 50 different poses at different amounts of skew in different areas of the field of view of the camera. For the first 10 sets, we replayed the robot poses for the calibration GUI as it automatically collected data without moving the target. For the last five, the target was translated and rotated slightly between each calibration while the robot maintained the same motion as the previous 10. Collection of this data required roughly six hours to complete since we needed to wait for each calibration to solve before starting a new set. The first 10 sets averaged roughly 90 images per set while the last five had reduced images because the pattern finder could not find a pattern in those images as easily. Near the end, we noticed that some of the poses were rotated too much causing the pattern finder to find the points of the target in the incorrect order. After the data was collected, we wrote a custom node to sort through all the data and throw out the worst offenders. After analysis of calibration results of this data which is described in the next chapter we determined that the automatic data collection of the *camera_calibration* GUI was unreliable and switched to a manual data collection process.

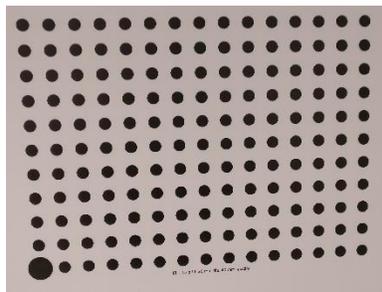


Figure 11. 11 x 15 Modified Circle Grid Calibration Pattern Mounted on a Wall

The second data set was collected using the wall mounted modified circle grid calibration pattern in the figure shown below. This time the pattern had circles arranged in an 11 by 15 grid with a circle diameter of 0.04 meters and a circle spacing of 0.02 meters. During our analysis, we

noticed that we did not fully fill the field of view of the camera for most data sets. This time instead of using a robot, we held the Xtion by hand and moved it around the target until every bar of the *camera_calibration* GUI was filled and the calibrate button lit up. This resulted in over 100 images. Once again, we sorted through the data and deleted anywhere the circle grid finder had trouble seeing the dots. The result was around 94 images. We further reduced this data set to 30 images by deleting any redundant images. Since we were not about to wave the camera at the target manually 15 times again, we decided to run the calibration 30 times using 29 of the images and swapping out an image for each calibration.

Due to what we learned from results of the second data set, we stopped using the *camera_calibration* GUI and wrote our own GUI for manual data collection for the third data set. The modified GUI only has a window displaying the camera view. When a target is detected, lines will be drawn over the detected points. We also drew dots over the first, origin, and last point to indicate if it was oriented correctly. This precluded us from the need to sort through the data in the end to check for bad images. Every time we were satisfied with the image we saw in the view window, we hit the ‘s’ key on the keyboard and it would automatically save the image.

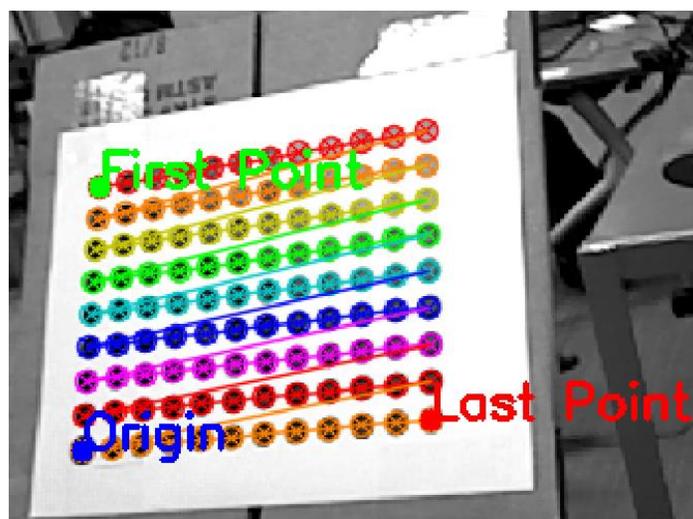


Figure 12. Circle Grid Finder with Dots Drawn at the First, Origin, and Last Point

The camera for this data set was mounted in a tripod. We placed the tripod at different positions in front of the target and collected 30 images at each position moving the orientation of the camera a tiny bit for each image so that they were all slightly different. In total, we moved the camera to 30 different positions resulting in a total of 900 images. This means that we would have 30 sets of 30 images to calibrate for this data set.

3.2 DATA ANALYSIS PROCESS

This section explains how we performed the calibrations on each of the data sets. The tables we refer to as “results” in this section take the calibration parameters results for each set of images within the entire data set and calculates the mean, median, and standard deviations. Full tables with all values can be found in the appendix.

First, we establish that the two calibration solvers used in the *camera_calibration* package and the *industrial_calibration* package obtain similar results. The *camera_calibration* package uses the camera calibration functionality provided by OpenCV to solve for the intrinsic camera parameters. Our *industrial_calibration* package uses Ceres-Solver as the LMA solver to solve for the intrinsic parameters. We establish that both methods achieve similar results because it is easier to modify the calibration algorithm using Ceres by changing the cost function instead of digging through the OpenCV source code and rebuilding the modifications.

We start by looping through all 15 data sets and crawling through the directory and loading all images. We extract the observations from the calibration patterns for every image, and passed them to the OpenCV camera calibration function. We also had to pass flags to use an intrinsic guess and to fix the K4, K5, K6 parameters since they were unused. The results were saved and shown in Table 1 in the next section.

The setup process for solving for camera intrinsics with *industrial_calibration* is like OpenCV. We extracted the observations and passed them to the calibration object and ran the calibration. Although the guesses for the poses of the camera at each image can be estimated analytically, to save processing time, we passed the poses calculated during the OpenCV solve in as the guesses for the Ceres solve. The analysis of the results of the calibration using Ceres for all 15 data sets using Ceres-Solver are shown in Table 2 in the next section.

Theory one was implemented using a function like the one above. Instead of calling solve once, we set the tangential distortion coefficient parameter blocks constant, solve for focal length, principal points, the radial distortion parameters, and the extrinsic rotations and translations. We then set all those parameters constant and solve again for the tangential distortion parameters only. The results of the calibration for all 15 data sets are shown in Table 3 in the next section.

In the second data set, the process for solving for the calibration parameters was similar. The only thing we changed was how the data was passed into the calibration function. This data set cycled through the same images changing only one every time. The tables for these can be found in the next section as well.

The results for the final data set were generated in a similar process as mentioned above. The last calibration we performed on data set three was to correct for the shifts in center points of the circle. We replaced the function that projected target points into the image plane and solved for the residual with a function that projected the target points into the image plane, corrected for the shift in center point, then solved for the residual.

CHAPTER FOUR: RESULTS AND DISCUSSION

4.1 DATA SET ONE RESULTS AND DISCUSSION

The table below shows the mean, median, and standard deviation of the calibration results for all 15 sets of images from data set one when calibrated using the intrinsic camera calibration features in OpenCV. We note that there is an average reprojection error of 0.25 pixels and the standard deviation of the reprojection error is relatively low. We also note that the standard deviations of the focal lengths and principal points are extremely high at around four to five pixels.

Table 1. Intrinsic Parameters of Calibration for Data Set 1 using OpenCV Full

Data Set	OpenCV									
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	Reprj Error
Mean	537.0658	536.1501	325.4949	231.8829	0.068379	-0.21432	-0.00126	-0.00011	0.177491	0.257201802
Median	538.3928	537.2339	328.0747	231.6251	0.06845	-0.218	-0.00151	0.001378	0.184089	0.257082027
Std.Dev	4.474733	4.092078	4.84634	3.077894	0.0074	0.02665	0.001275	0.002704	0.044171	0.03925095

The intrinsic parameters from this data set, when solved with Ceres, have very similar means when compared to OpenCV as shown below in Table 2. This validates that although implemented differently, our method of solving produces similar results with OpenCV. We note that the reprojection errors are eight times less when using Ceres compared to OpenCV. This is to be expected since Ceres is a better solver than whatever OpenCV uses in the backend. The standard deviations in the focal lengths and optical center points remain high. This is consistent with the results from Table 1 and is the issue we are trying to resolve.

Table 2. Intrinsic Parameters of Calibration for Data Set 1 using Ceres

Data Set	Ceres									
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	Reprj Error
Mean	537.0703	536.1547	325.5078	231.8852	0.068393	-0.21432	-0.00126	-0.0001	0.177455	0.03360186
Median	538.3818	537.2246	328.1836	231.6129	0.068486	-0.218	-0.00152	0.001386	0.1839	0.032926854
Std.Dev	4.465886	4.083868	4.841077	3.081418	0.0074	0.026677	0.001289	0.00271	0.044181	0.010661583

The next table contains the results of using a two-step solve where we hold the p distortion parameters constant, solve for all the other parameters, hold those constant, and only solve for the p values. The first thing we notice is that the parameters have been reduced to nothing and have a very low standard deviation. This is consistent with the fact that there is usually very little tangential distortion. Most of the radial components of the tangential distortion seem to have been accounted for in the k parameters. The other change that pops out is the reduction in the standard deviation of the C_x value.

Table 3. Intrinsic Parameters of Calibration for Data Set 1 using Ceres with 2-Step Solve

Data Set	Ceres 2-Step									
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	Reprj Error
Mean	536.7143	535.821	325.6737	233.4829	0.067181	-0.21793	-1.7E-05	-2.1E-06	0.183787	0.033980238
Median	537.9964	536.847	326.2907	232.9204	0.067723	-0.22264	-1.8E-05	4.87E-06	0.192516	0.033106432
Std.Dev	5.103357	4.63207	2.298207	4.18509	0.007261	0.02711	8.76E-06	1.36E-05	0.04223	0.010878865

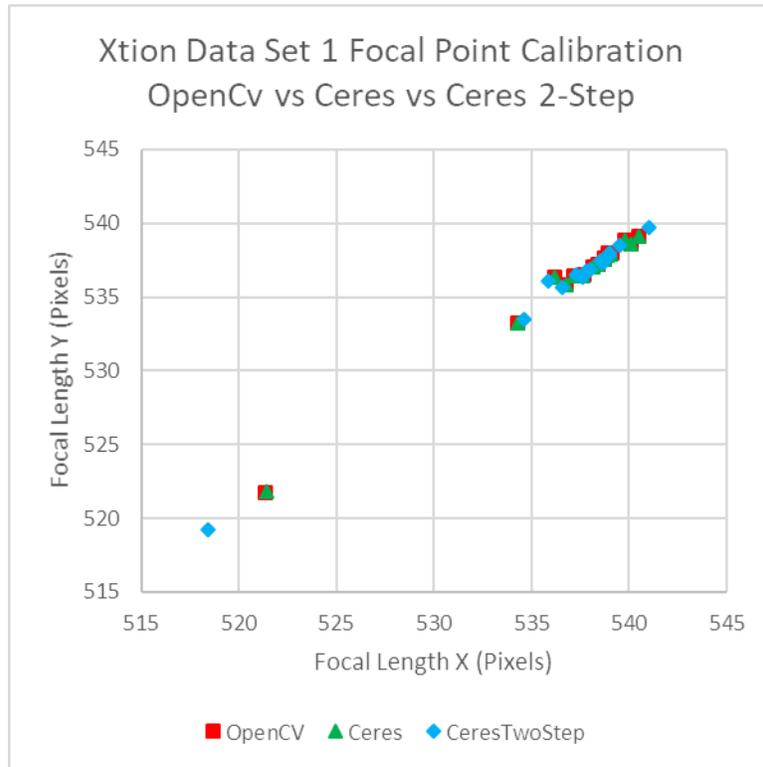


Figure 13. Plot of Focal Length Results of Xtion Intrinsic Calibration Using Three Methods

The calibration results of this data set had a disturbingly high standard deviation in the focal lengths and principal points as we expected. We hypothesized that the issue could be in our data. Looking through the data, we noticed that many images did not fill the entire field of view with the calibration pattern. The focal lengths in x and y are plotted against each other in Figure 13 for all three methods. The outliers are of the last couple image sets where the calibration pattern had been moved further back.

For the next data set, we made sure to collect enough images where the entire field of view was filled as well as enough images with significant amounts of skew. This time we did not collect 15 sets of calibration images as with data set one. Instead, we swapped out a single image out of 30 images to have 30 sets of 29 images similar Zhang, except he used only five [9].

4.2 DATA SET TWO RESULTS AND DISCUSSION

From the table below, we see that the residuals have increased compared to the first data set when we calibrate with OpenCV. The standard deviations of the focal lengths and principal point are significantly lower compared to data set one. This can most likely be explained by the fact that we just used the same 30 images over and over while swapping one out. The biggest issue is that the two answers are different. The mean of the focal lengths in x and y for the first data set were at 537.06 and 636.15 respectively. In this data set, the means of the focal lengths in x and y are 533.28 and 534.34 respectively. That is a near four pixels of increase in the x. The results of the principal points are very different between the two as well.

Table 4. Intrinsic Parameters of Calibration for Data Set 2 using OpenCV

Data Set	OpenCV									Reprj Error
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	
Mean	533.2851	534.3429	316.4282	233.2138	0.052171	-0.23268	0.000455	0.001049	0.237757	0.455218433
Median	533.2745	534.3235	316.502	233.1885	0.051919	-0.23215	0.000455	0.001061	0.23767	0.4595605
Std.Dev	0.813208	0.813384	0.422609	0.473307	0.001324	0.005833	0.000147	0.000263	0.008834	0.012787092

Again, the reprojection errors are lower when solving with Ceres compared to OpenCV as shown in the table below. The standard deviations of the focal lengths and principal points also remained relatively similar. We notice that the mean of the reprojection errors has tripled compare to the reprojection errors in Table 2.

Table 5. Intrinsic Parameters of Calibration for Data Set 2 using Ceres

Data Set	Ceres									
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	Reprj Error
Mean	533.2873	534.3451	316.4317	233.2107	0.052166	-0.23269	0.000454	0.00105	0.23783	0.103693607
Median	533.277	534.326	316.5045	233.1855	0.051929	-0.23214	0.000456	0.001063	0.237693	0.105598
Std.Dev	0.812531	0.812651	0.418823	0.471011	0.001324	0.005822	0.000147	0.000262	0.008807	0.005558342

Using a two-step solve had a very similar effect on this data set as it did with data set one. We see this in the table below where the standard deviation of the C_x component is lower compared to the previous table. The tangential distortion parameters have once again been reduced to nothing.

Table 6. Intrinsic Parameters of Calibration for Data Set 2 using Ceres with 2-Step Solve

Data Set	Ceres 2-Step									
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	Reprj Error
Mean	533.2917	534.3537	314.8996	232.6026	0.051203	-0.22615	2.54E-05	2.24E-05	0.227083	0.1041277
Median	533.277	534.331	314.9405	232.5705	0.050918	-0.22557	2.56E-05	2.28E-05	0.22777	0.106045
Std.Dev	0.811556	0.811568	0.274396	0.480788	0.001369	0.005877	8.18E-06	5.36E-06	0.008676	0.005515804

The results of this data set introduce the questions of what happened that changed our values so significantly and why have the reprojection errors increased? While investigating how this could have happened, we looked at the images collected automatically using the *camera_calibration* package. We noticed that many of the images had motion blur. This led us to be more careful when collecting our data and not using the GUI provided with the ROS *camera_calibration* package. Instead we wrote our own data collection program that allowed for manual data collection to avoid any motion blur. We also added in indicators to show if the circle

grid finder was finding the points correctly. The circle grid finder uses the large dot of the modified circle grid to determine orientation of the target. When there is too much skew, the grid finder cannot tell which dot is the large dot, causing the grid points to be returned incorrectly. Using this new data collection program, we collected 30 sets of 30 images by moving a tripod around in front of the calibration pattern to 30 different poses and moving the camera slightly at each pose so the images are not all the same. The result is the data set of 900 images.

4.3 DATA SET THREE RESULTS AND DISCUSSION

The results of the intrinsic calibration using the third data set using OpenCV is shown in the table below. The first thing we notice is that the values of focal length and principal points are close to the values obtained in Table 4 for data set two. The reprojection errors are also lower. In fact, the reprojection errors for this solve using OpenCV are on par with the reprojection errors from Table 5 when Ceres was used. The next thing we note is that there is very little variance in the focal lengths and principal point values. This is a very good sign since the goal of the first hypothesis was to eliminate large variances in the focal length and principal point.

Table 7. Intrinsic Parameters of Calibration for Data Set 3 using OpenCV

OpenCV Solve										
Set	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	Reprj Error
Mean	533.5156	534.2188	315.1511	232.2743	0.057415	-0.24467	-0.00045	0.000157235	0.243113	0.111240033
Median	533.513	534.219	315.179	232.2715	0.057631	-0.24493	-0.00043	0.000177383	0.243794	0.1111215
StdDev	0.062229	0.057637	0.126214	0.059501	0.000838	0.00552	7.48E-05	0.000120527	0.009494	0.001020407

The results of solving the third data set using Ceres once again produces a much lower reprojection error. The standard deviations are essentially the same as the OpenCV result. Since the variance in the focal lengths and principal points are already very low, doing a two-step calibration should have little to no effect.

Table 8. Intrinsic Parameters of Calibration for Data Set 3 using Ceres

Ceres Solve										
Set	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	Reprj Error
Mean	533.5158	534.219	315.1523	232.2752	0.057409	-0.24464	-0.00045	0.00015804	0.243073	0.006187686
Median	533.513	534.219	315.1805	232.2725	0.057624	-0.24491	-0.00043	0.00017819	0.243756	0.00617398
StdDev	0.062114	0.057584	0.126057	0.059514	0.000838	0.005522	7.48E-05	0.00012051	0.009497	0.0001138

The table below with the results of the two-step solve show that doing a two-step solve is not required since the values remain essentially the same.

Table 9. Intrinsic Parameters of Calibration for Data Set 2 using Ceres with 2-Step Solve

Ceres 2-Step Solve										
Set	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	Reprj Error
Mean	533.5291	534.2124	314.9268	232.8847	0.0574	-0.24551	-2.2925E-05	3.688E-06	0.244713	0.006272537
Median	533.5195	534.21	314.9155	232.873	0.057541	-0.24534	-2.1598E-05	3.554E-06	0.244819	0.00627153
StdDev	0.05693	0.055141	0.061078	0.068533	0.000848	0.005369	4.6708E-06	2.496E-06	0.009002	0.00011488

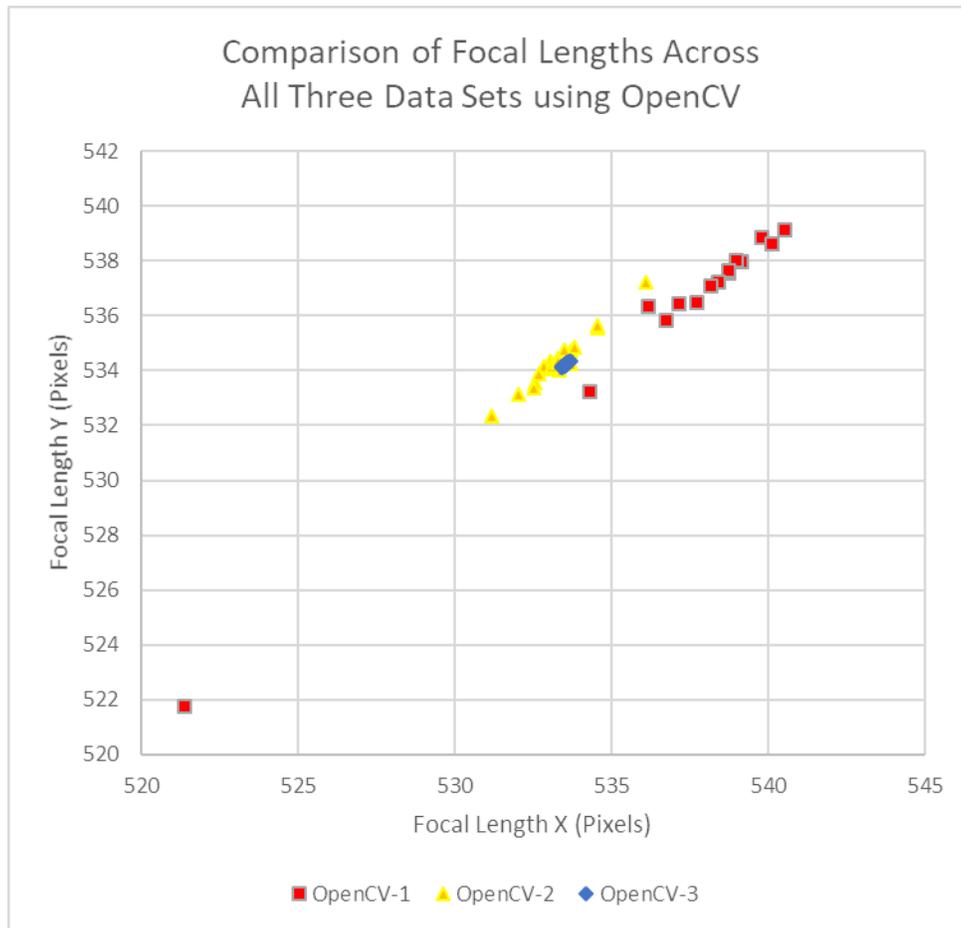


Figure 14. Plot of Focal Length Results of Xtion Intrinsic Calibration Across All Data Sets

We compare the focal lengths using OpenCV across all three data sets in Figure 14. We see that data set three has the lowest deviation amongst all three data sets. From the results of data set three we learn a couple of things. The first thing we learned is that our first hypothesis was wrong. There is nothing wrong with how we solve for the distortion parameters. If we capture the right amount and type of data and ensure there is no motion blur, the focal lengths and principal points have a very small variance across multiple calibrations. The second thing we learn is that unless you want a different calibration every time you run a calibration, do not use the automatic data collection that ships with the ROS *camera_calibration* package.

CHAPTER FIVE: CONCLUSION

5.1 CONCLUSION

In this thesis, we attempted to improve the results of intrinsic camera calibration of the *camera_calibration* package by testing the hypothesis that a two-step solve would decrease the variance in the focal lengths and principal points. A two-step solve means that we hold the tangential distortion parameters constant and solving for the focal lengths, principal point coordinates, and radial distortion parameters first, then holding those parameters constant and solving for the tangential components last. This proved to be unnecessary as shown in the third data set. The variances in the focal length and principal points were caused by not having enough of the right type of data and too much motion blur in some of the data.

To fix this problem, we propose the following guidelines when using the ROS *camera_calibration* package. Make sure data is being collected in a well-lit room. Use a static target instead of a static camera. Cover the lens of the camera while it is in motion to prevent the software from collecting data while the camera is in motion. Make sure to capture images where the calibration pattern fills the entire field of view of the camera. Capture more images from this distance while skewing to the left, right, top, and bottom. Move back until the calibration pattern is approximately half the size compared to when it filled the field of view. Capture data from the different skews again. Then move the camera so that the target fills the top left, top center, top right, bottom left, bottom center, bottom right, left center, and right center capturing data from all the skews at each location as well. Capturing this data without any motion blur should result in consistent calibrations.

5.2 FUTURE WORK

Although we suggested poses from which to capture data, these are based off of personal experience with camera calibration and intuition. It would be interesting to research the least number of images required and their exact poses to capture enough data for a calibration with good results. We also plan to modify the *camera_calibration* package to not capture data while the camera is in motion to avoid motion blur and attempt to push it back to the public repository. Tools to perform manual data collect data for this thesis and we plan to release that with IC2 as well.

APPENDIX

Table 10. Intrinsic Parameters of Calibration for Data Set 1 using OpenCV Full

Data Set	OpenCV									Reprj Error
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	
1	537.7097	536.4955	328.9036	230.3843	0.071099	-0.22231	-0.00164	0.001704	0.207288	0.288927922
2	537.1628	536.4434	329.3325	232.811	0.06845	-0.22011	-0.00069	0.002134	0.184927	0.257923333
3	538.7228	537.5716	328.216	230.7882	0.072514	-0.24749	-0.00154	0.001378	0.251105	0.257335926
4	539.1191	537.9649	329.2799	230.5537	0.069168	-0.22144	-0.00155	0.001703	0.183699	0.256332179
5	538.9678	538.0276	329.4303	231.6251	0.067829	-0.1998	-0.00121	0.001957	0.15495	0.259630639
6	538.7129	537.6616	328.0747	230.7397	0.067651	-0.21471	-0.0016	0.000946	0.184089	0.267401059
7	538.3928	537.2339	327.3839	231.3589	0.068042	-0.21326	-0.00174	0.000749	0.178513	0.249364986
8	539.8049	538.833	329.1993	231.5774	0.070135	-0.22794	-0.00116	0.001637	0.201847	0.256619528
9	536.7549	535.8375	328.9169	232.1939	0.06678	-0.20888	-0.00129	0.001552	0.169872	0.256962253
10	538.1515	537.0732	327.5397	232.0631	0.067624	-0.218	-0.00151	0.000956	0.186384	0.250851766
11	540.5275	539.1459	315.8886	233.6881	0.06895	-0.16433	-0.00366	-0.00376	0.095483	0.193196265
12	540.1037	538.6223	317.3202	223.6189	0.044844	-0.16451	-0.00027	-0.00499	0.098084	0.206502743
13	536.1762	536.3404	316.6176	233.8819	0.065736	-0.24744	-0.00038	-0.00445	0.238175	0.200529637
14	534.3156	533.2348	323.3679	233.6195	0.074836	-0.18513	0.00231	0.001757	0.116522	0.297221169
15	521.3646	521.766	322.9521	239.339	0.082022	-0.25943	-0.00302	-0.00494	0.211423	0.359227625
Mean	537.0658	536.1501	325.4949	231.8829	0.068379	-0.21432	-0.00126	-0.00011	0.177491	0.257201802
Median	538.3928	537.2339	328.0747	231.6251	0.06845	-0.218	-0.00151	0.001378	0.184089	0.257082027
Std.Dev	4.474733	4.092078	4.84634	3.077894	0.0074	0.02665	0.001275	0.002704	0.044171	0.03925095

Table 11. Intrinsic Parameters of Calibration for Data Set 1 using Ceres Full

Data Set	Ceres									Reprj Error
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	
1	537.6577	536.4508	328.9222	230.4324	0.071092	-0.22231	-0.00163	0.001722	0.20728	0.041739734
2	537.1701	536.4497	329.3082	232.7883	0.068486	-0.22025	-0.0007	0.002118	0.185106	0.033262267
3	538.7123	537.5628	328.2259	230.807	0.072493	-0.24742	-0.00153	0.001386	0.251026	0.033110923
4	539.0805	537.9321	329.271	230.5739	0.06918	-0.22143	-0.00156	0.001703	0.183582	0.03285316
5	538.9708	538.0302	329.4138	231.6129	0.067852	-0.19988	-0.00121	0.001946	0.155047	0.033704069
6	538.8727	537.7996	328.1836	230.5886	0.067818	-0.21477	-0.00164	0.001009	0.1839	0.03207822
7	538.3818	537.2246	327.3954	231.377	0.068022	-0.21321	-0.00173	0.000757	0.178466	0.031091482
8	539.7873	538.8182	329.1706	231.5668	0.070179	-0.22803	-0.00117	0.001622	0.201881	0.032926854
9	536.7624	535.8438	328.8956	232.1741	0.06681	-0.20896	-0.0013	0.001538	0.169937	0.033014834
10	538.1352	537.0598	327.5225	232.0646	0.067641	-0.218	-0.00152	0.000947	0.18628	0.03146335
11	540.5178	539.1372	315.8922	233.6972	0.068952	-0.16439	-0.00365	-0.00376	0.095636	0.01866245
12	540.102	538.6216	317.3243	223.6219	0.044841	-0.16451	-0.00027	-0.00499	0.098115	0.021321733
13	536.1709	536.3383	316.6174	233.8919	0.065717	-0.24734	-0.00038	-0.00445	0.238022	0.020106114
14	534.331	533.2502	323.533	233.7613	0.074743	-0.18473	0.002386	0.001867	0.115951	0.044170393
15	521.4022	521.8011	322.9408	239.32	0.082071	-0.2596	-0.00302	-0.00495	0.21159	0.064522314
Mean	537.0703	536.1547	325.5078	231.8852	0.068393	-0.21432	-0.00126	-0.0001	0.177455	0.03360186
Median	538.3818	537.2246	328.1836	231.6129	0.068486	-0.218	-0.00152	0.001386	0.1839	0.032926854
Std.Dev	4.465886	4.083868	4.841077	3.081418	0.0074	0.026677	0.001289	0.00271	0.044181	0.010661583

Table 12. Intrinsic Parameters of Calibration for Data Set 1 using Ceres 2-Step Solve Full

Data Set	Ceres 2-Step									
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	Reprj Error
1	537.6405	536.3853	326.4415	232.2131	0.070408	-0.2328	-2E-05	8.09E-06	0.220755	0.041996538
2	537.3284	536.5504	326.2517	233.2344	0.06867	-0.22967	-9.6E-06	8.48E-06	0.197101	0.033453326
3	538.6016	537.4289	326.2306	232.6003	0.071664	-0.25587	-1.9E-05	6.19E-06	0.262561	0.033313851
4	539.0101	537.8199	326.8003	232.347	0.068371	-0.22987	-2E-05	7.06E-06	0.192516	0.033100531
5	538.9904	537.9876	326.5889	232.8084	0.067723	-0.21006	-1.6E-05	8.72E-06	0.166913	0.033933335
6	538.5703	537.4934	326.7089	232.683	0.066819	-0.22225	-1.9E-05	4.49E-06	0.193205	0.032258067
7	537.9964	536.847	326.2907	233.6661	0.06676	-0.2189	-1.9E-05	3.46E-06	0.184787	0.031253422
8	539.5363	538.5239	326.7475	232.9204	0.069801	-0.23544	-1.4E-05	7.9E-06	0.210155	0.033106432
9	536.5924	535.639	326.621	233.7247	0.065939	-0.21318	-1.7E-05	6.99E-06	0.171706	0.033208453
10	537.8352	536.7458	326.1232	234.0222	0.066559	-0.22264	-1.7E-05	4.87E-06	0.190012	0.031614722
11	538.653	537.3292	321.3509	238.6453	0.066217	-0.19126	-3E-05	-2E-05	0.154739	0.019211501
12	541.0281	539.7101	325.1936	224.1689	0.043515	-0.14064	-1.8E-05	-2.9E-05	0.07287	0.022057095
13	535.8816	536.0895	322.7507	234.6572	0.06271	-0.22515	-1.6E-05	-2.2E-05	0.207862	0.02060732
14	534.5999	533.5243	320.7777	229.7385	0.076884	-0.18981	1.05E-05	-1.3E-06	0.1252	0.044239538
15	518.4506	519.2401	330.2288	244.8137	0.075669	-0.25144	-2.8E-05	-2.5E-05	0.206429	0.066349441
Mean	536.7143	535.821	325.6737	233.4829	0.067181	-0.21793	-1.7E-05	-2.1E-06	0.183787	0.033980238
Median	537.9964	536.847	326.2907	232.9204	0.067723	-0.22264	-1.8E-05	4.87E-06	0.192516	0.033106432
Std.Dev	5.103357	4.63207	2.298207	4.18509	0.007261	0.02711	8.76E-06	1.36E-05	0.04223	0.010878865

Table 13. Intrinsic Parameters of Calibration for Data Set 2 using OpenCV Full

Data Set	OpenCV									Reprj Error
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	
1	533.085	534.198	316.507	233.538	0.050501	-0.22503	0.000493	0.001072	0.227783	0.457135
2	532.037	533.126	316.06	233.993	0.050031	-0.22468	0.000556	0.000975	0.228345	0.458803
3	533.675	534.742	316.457	232.768	0.05126	-0.22819	0.000419	0.001057	0.232029	0.462812
4	533.022	534.078	316.475	233.381	0.051893	-0.23194	0.000403	0.001099	0.239667	0.462706
5	533.309	534.034	315.599	233.454	0.053039	-0.23248	0.000372	0.000333	0.232595	0.432749
6	534.531	535.559	316.566	233.311	0.051604	-0.23448	0.000419	0.001059	0.244186	0.459891
7	533.116	534.211	316.292	232.989	0.053361	-0.23832	0.000463	0.00106	0.245557	0.461454
8	533.272	534.322	316.497	233.198	0.052069	-0.23199	0.000439	0.001062	0.236623	0.463037
9	532.564	533.569	316.507	234.126	0.052394	-0.23151	0.000489	0.000975	0.234514	0.453299
10	533.048	534.325	315.834	233.039	0.054847	-0.24093	0.000156	0.000481	0.247002	0.446937
11	533.705	534.253	317.498	232.673	0.055478	-0.24926	0.000106	0.001777	0.260696	0.397295
12	533.811	534.841	316.174	232.926	0.051932	-0.23093	0.000467	0.00102	0.23523	0.461944
13	533.024	534.103	316.882	233.128	0.052717	-0.23558	0.000436	0.001059	0.242109	0.459978
14	531.166	532.325	315.885	232.22	0.051907	-0.23389	0.00046	0.001096	0.241955	0.452099
15	533.282	534.434	316.207	233.236	0.051641	-0.2297	0.000479	0.000813	0.233603	0.460864
16	532.836	534.164	316.622	233.609	0.053617	-0.24217	0.000576	0.001139	0.253178	0.45379
17	536.099	537.239	315.831	232.397	0.049721	-0.22167	0.000471	0.000953	0.223874	0.452732
18	533.1	534.14	316.749	234	0.050933	-0.22287	0.000242	0.001418	0.213877	0.455755
19	532.522	533.371	316.037	233.489	0.051664	-0.2287	0.000486	0.001331	0.233194	0.458543
20	533.29	534.35	316.442	233.126	0.052732	-0.23541	0.000437	0.001052	0.241636	0.463134
21	533.591	534.432	316.358	233.498	0.052788	-0.23985	0.000999	0.000977	0.248527	0.444202
22	532.667	533.875	315.504	233.839	0.055159	-0.23855	0.000358	0.000543	0.24052	0.447191
23	533.354	534.432	316.548	233.179	0.052848	-0.23587	0.000447	0.001072	0.242397	0.462843
24	533.314	534.325	316.56	233.162	0.051949	-0.23217	0.00041	0.001157	0.237732	0.462353
25	533.266	534.404	316.708	233.256	0.051952	-0.23213	0.00055	0.001213	0.237607	0.46202
26	533.25	534.319	316.79	233.177	0.051382	-0.22974	0.000462	0.001124	0.234604	0.463012
27	533.277	534.337	316.797	233.412	0.051392	-0.22899	0.000625	0.001327	0.231174	0.46265
28	534.544	535.651	316.895	232.043	0.051193	-0.2276	0.00045	0.001161	0.230779	0.45923
29	533.295	534.366	316.716	233.111	0.051496	-0.23256	0.000442	0.001095	0.240184	0.462704
30	533.501	534.761	316.85	233.136	0.051645	-0.23322	0.000531	0.000971	0.241537	0.455391
Mean	533.2851	534.3429	316.4282	233.2138	0.052171	-0.23268	0.000455	0.001049	0.237757	0.455218433
Median	533.2745	534.3235	316.502	233.1885	0.051919	-0.23215	0.000455	0.001061	0.23767	0.4595605
Std.Dev	0.813208	0.813384	0.422609	0.473307	0.001324	0.005833	0.000147	0.000263	0.008834	0.012787092

Table 14. Intrinsic Parameters of Calibration for Data Set 2 using Ceres Full

Data Set	Ceres									Reprj Error
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	
1	533.088	534.201	316.51	233.533	0.050492	-0.22502	0.000492	0.001073	0.227831	0.104486
2	532.042	533.131	316.065	233.986	0.050026	-0.22473	0.000555	0.000976	0.228498	0.10525
3	533.676	534.744	316.462	232.766	0.051257	-0.22822	0.000419	0.001058	0.232122	0.107097
4	533.025	534.081	316.477	233.377	0.051884	-0.23193	0.000403	0.0011	0.239691	0.107048
5	533.311	534.036	315.61	233.453	0.053044	-0.23259	0.000373	0.000338	0.23286	0.0936358
6	534.533	535.56	316.566	233.308	0.051599	-0.23447	0.000418	0.001058	0.244188	0.10575
7	533.118	534.213	316.298	232.988	0.053361	-0.23837	0.000463	0.001063	0.245708	0.10647
8	533.274	534.325	316.499	233.195	0.052063	-0.23199	0.000439	0.001063	0.236682	0.107202
9	532.568	533.573	316.51	234.119	0.052387	-0.23153	0.000487	0.000976	0.234608	0.10274
10	533.051	534.327	315.843	233.038	0.05485	-0.24101	0.000157	0.000485	0.247216	0.0998765
11	533.706	534.254	317.493	232.669	0.055462	-0.24917	0.000104	0.001774	0.260535	0.0789215
12	533.813	534.843	316.181	232.924	0.051933	-0.231	0.000467	0.001023	0.235398	0.106696
13	533.025	534.105	316.879	233.124	0.052703	-0.2355	0.000435	0.001057	0.242	0.10579
14	531.167	532.326	315.897	232.223	0.051924	-0.23405	0.000462	0.001102	0.242302	0.102197
15	533.285	534.436	316.212	233.234	0.051639	-0.22974	0.000479	0.000815	0.233732	0.106198
16	532.839	534.167	316.623	233.603	0.053603	-0.24213	0.000574	0.001139	0.253163	0.102963
17	536.099	537.239	315.846	232.402	0.049728	-0.2218	0.000474	0.00096	0.224198	0.102483
18	533.104	534.144	316.75	233.993	0.050912	-0.2228	0.00024	0.001417	0.213802	0.103856
19	532.525	533.375	316.042	233.484	0.051659	-0.22874	0.000485	0.001332	0.233331	0.105131
20	533.292	534.352	316.446	233.124	0.052728	-0.23543	0.000437	0.001054	0.241715	0.107246
21	533.593	534.434	316.357	233.494	0.052774	-0.2398	0.000998	0.000976	0.248482	0.0986576
22	532.671	533.879	315.512	233.835	0.055164	-0.23866	0.000357	0.000546	0.24078	0.0999898
23	533.356	534.434	316.55	233.176	0.052842	-0.23588	0.000446	0.001073	0.24244	0.107112
24	533.316	534.327	316.562	233.159	0.051942	-0.23217	0.000409	0.001158	0.237779	0.106885
25	533.269	534.406	316.709	233.252	0.051942	-0.23211	0.000549	0.001213	0.237606	0.106731
26	533.251	534.32	316.791	233.173	0.05137	-0.2297	0.00046	0.001124	0.234585	0.10719
27	533.28	534.34	316.795	233.406	0.051378	-0.22894	0.000623	0.001325	0.231123	0.107022
28	534.543	535.65	316.896	232.045	0.051195	-0.22761	0.000451	0.001162	0.230802	0.105446
29	533.297	534.368	316.717	233.107	0.051484	-0.23253	0.000441	0.001095	0.240181	0.107047
30	533.503	534.763	316.852	233.132	0.051636	-0.2332	0.00053	0.000972	0.241548	0.103691
Mean	533.2873	534.3451	316.4317	233.2107	0.052166	-0.23269	0.000454	0.00105	0.23783	0.103693607
Median	533.277	534.326	316.5045	233.1855	0.051929	-0.23214	0.000456	0.001063	0.237693	0.105598
Std.Dev	0.812531	0.812651	0.418823	0.471011	0.001324	0.005822	0.000147	0.000262	0.008807	0.005558342

Table 15. Intrinsic Parameters of Calibration for Data Set 2 using Ceres 2-Step Solve Full

Data Set	Ceres 2-Step									
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	Reprj Error
1	533.096	534.214	314.944	232.873	0.049422	-0.21791	2.82E-05	2.31E-05	0.216269	0.104933
2	532.072	533.165	314.632	233.233	0.049114	-0.21848	3.14E-05	2.10E-05	0.218407	0.105659
3	533.739	534.814	314.893	232.169	0.050324	-0.22193	2.26E-05	2.22E-05	0.221726	0.107477
4	533.077	534.136	314.887	232.832	0.051132	-0.22674	2.23E-05	2.42E-05	0.230885	0.107475
5	533.309	534.04	315.123	232.97	0.052828	-0.23052	2.13E-05	7.78E-06	0.229413	0.0937284
6	534.536	535.567	315.021	232.744	0.050543	-0.22746	2.37E-05	2.26E-05	0.232704	0.106158
7	533.122	534.221	314.751	232.372	0.052469	-0.23213	2.62E-05	2.27E-05	0.235368	0.1069
8	533.289	534.343	314.946	232.602	0.051053	-0.2252	2.58E-05	2.29E-05	0.225535	0.107621
9	532.566	533.576	315.093	233.484	0.051545	-0.22547	2.84E-05	2.12E-05	0.22455	0.103124
10	533.051	534.33	315.139	232.832	0.054421	-0.23814	8.84E-06	1.02E-05	0.242494	0.0999548
11	533.719	534.264	314.882	232.503	0.05357	-0.23785	5.20E-06	3.66E-05	0.241786	0.0798624
12	533.864	534.895	314.669	232.268	0.050914	-0.22417	2.54E-05	2.13E-05	0.224254	0.107073
13	533.035	534.118	315.338	232.539	0.051565	-0.22822	2.43E-05	2.26E-05	0.230319	0.106205
14	531.182	532.345	314.307	231.617	0.050798	-0.22622	2.59E-05	2.34E-05	0.229373	0.102646
15	533.275	534.433	315.025	232.599	0.050979	-0.22501	2.72E-05	1.78E-05	0.225752	0.106477
16	532.848	534.18	314.959	232.833	0.052499	-0.23451	3.16E-05	2.38E-05	0.240674	0.10344
17	536.114	537.259	314.437	231.757	0.049071	-0.2166	2.64E-05	2.03E-05	0.215188	0.102844
18	533.014	534.044	314.715	233.68	0.049706	-0.21509	1.29E-05	3.05E-05	0.202113	0.104481
19	532.528	533.387	314.121	232.849	0.050374	-0.22043	2.63E-05	2.79E-05	0.219421	0.105723
20	533.303	534.368	314.906	232.534	0.051746	-0.2288	2.46E-05	2.26E-05	0.23084	0.107659
21	533.59	534.45	314.945	232.189	0.051955	-0.23286	5.53E-05	2.16E-05	0.237284	0.0993364
22	532.67	533.885	314.716	233.369	0.054692	-0.23516	2.01E-05	1.19E-05	0.235113	0.100131
23	533.361	534.441	314.982	232.579	0.051817	-0.22901	2.51E-05	2.29E-05	0.231196	0.107543
24	533.316	534.332	314.868	232.608	0.050923	-0.22539	2.32E-05	2.53E-05	0.226574	0.107347
25	533.279	534.414	314.937	232.526	0.050744	-0.22407	3.08E-05	2.54E-05	0.224502	0.107276
26	533.242	534.317	315.17	232.547	0.050317	-0.22276	2.51E-05	2.39E-05	0.22321	0.107641
27	533.291	534.356	314.854	232.562	0.050099	-0.22027	3.25E-05	2.72E-05	0.216908	0.107623
28	534.496	535.605	315.18	231.468	0.050118	-0.22051	2.61E-05	2.48E-05	0.219158	0.105932
29	533.273	534.349	315.113	232.505	0.050441	-0.22567	2.44E-05	2.35E-05	0.228966	0.107485
30	533.493	534.762	315.435	232.436	0.050905	-0.22783	3.01E-05	2.10E-05	0.232522	0.104076
Mean	533.2917	534.3537	314.8996	232.6026	0.051203	-0.22615	2.54E-05	2.24E-05	0.227083	0.1041277
Median	533.277	534.331	314.9405	232.5705	0.050918	-0.22557	2.56E-05	2.28E-05	0.22777	0.106045
Std.Dev	0.811556	0.811568	0.274396	0.480788	0.001369	0.005877	8.18E-06	5.36E-06	0.008676	0.005515804

Table 16. Intrinsic Parameters of Calibration for Data Set 3 using OpenCV Full

Set	OpenCV Solve									Reprj Error
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	
0	533.504	534.209	314.964	232.326	0.057863	-0.24968	-0.00039	-1.26E-06	0.252962	0.112636
1	533.445	534.157	314.933	232.299	0.058228	-0.25042	-0.00042	-3.20E-05	0.253095	0.111165
2	533.463	534.173	314.988	232.25	0.058095	-0.25079	-0.00043	-3.13E-07	0.253182	0.111206
3	533.513	534.208	315.062	232.345	0.058591	-0.25239	-0.00041	9.53E-05	0.254337	0.110599
4	533.473	534.167	314.991	232.288	0.058337	-0.25113	-0.00045	6.03E-05	0.252089	0.111845
5	533.42	534.128	315.066	232.347	0.057694	-0.24898	-0.00044	4.94E-05	0.251282	0.112727
6	533.41	534.12	315.134	232.241	0.057371	-0.24797	-0.00047	1.00E-04	0.250599	0.112229
7	533.428	534.142	315.004	232.272	0.058232	-0.25411	-0.00045	1.91E-05	0.262539	0.114034
8	533.459	534.17	315.051	232.232	0.058784	-0.25569	-0.00044	4.37E-06	0.263927	0.112
9	533.422	534.139	315.026	232.351	0.058002	-0.24883	-0.00039	1.40E-05	0.25074	0.11098
10	533.577	534.29	314.997	232.347	0.057572	-0.24624	-0.0003	-2.49E-05	0.246308	0.111131
11	533.575	534.286	315.011	232.369	0.056505	-0.24053	-0.00036	7.38E-05	0.237331	0.111854
12	533.655	534.365	315.065	232.276	0.056122	-0.23782	-0.00043	0.000101338	0.232779	0.11056
13	533.513	534.232	315.174	232.34	0.056689	-0.23816	-0.00039	0.000164366	0.232128	0.110794
14	533.568	534.262	315.172	232.301	0.055457	-0.23405	-0.00039	0.000149858	0.227912	0.109374
15	533.524	534.2	315.258	232.333	0.056508	-0.2389	-0.00034	0.000211947	0.233801	0.109938
16	533.562	534.246	315.184	232.348	0.056287	-0.2378	-0.00039	0.000197397	0.231699	0.110412
17	533.559	534.259	315.214	232.269	0.05691	-0.24156	-0.00041	0.00023159	0.236404	0.111663
18	533.511	534.197	315.274	232.268	0.056629	-0.23909	-0.00041	0.000303277	0.23222	0.110279
19	533.528	534.226	315.284	232.178	0.05769	-0.24376	-0.00051	0.000324795	0.238222	0.110209
20	533.478	534.194	315.28	232.242	0.057136	-0.23971	-0.00041	0.000367764	0.234934	0.110566
21	533.671	534.344	315.445	232.3	0.058358	-0.24679	-0.00044	0.000376296	0.243918	0.112934
22	533.474	534.183	315.271	232.247	0.058207	-0.2461	-0.00047	0.000296573	0.244037	0.109769
23	533.555	534.262	315.253	232.204	0.058056	-0.24649	-0.00052	0.000263989	0.245485	0.110129
24	533.515	534.233	315.277	232.144	0.057263	-0.24447	-0.00059	0.000247848	0.244198	0.111607
25	533.511	534.229	315.263	232.177	0.057857	-0.24539	-0.00058	0.000241198	0.24367	0.111112
26	533.506	534.228	315.239	232.182	0.057793	-0.24356	-0.0006	0.000232588	0.237951	0.111105
27	533.546	534.249	315.256	232.271	0.057414	-0.24204	-0.00056	0.000246461	0.235525	0.110627
28	533.564	534.253	315.204	232.262	0.056724	-0.24053	-0.00055	0.000211471	0.237465	0.111721
29	533.539	534.212	315.192	232.219	0.056081	-0.23721	-0.00053	0.0001904	0.232654	0.111566
Mean	533.5156	534.2188	315.1511	232.2743	0.057415	-0.24467	-0.00045	0.000157235	0.243113	0.111240033
Median	533.513	534.219	315.179	232.2715	0.057631	-0.24493	-0.00043	0.000177383	0.243794	0.1111215
StdDev	0.062229	0.057637	0.126214	0.059501	0.000838	0.00552	7.48E-05	0.000120527	0.009494	0.001020407

Table 17. Intrinsic Parameters of Calibration for Data Set 3 using Ceres Full

Set	Ceres Solve									
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	Reprj Error
0	533.505	534.209	314.966	232.327	0.057858	-0.24966	-0.00039	-3.97E-07	0.252934	0.00634336
1	533.445	534.158	314.934	232.3	0.058223	-0.2504	-0.00042	-3.11E-05	0.253067	0.00623289
2	533.464	534.174	314.989	232.251	0.058089	-0.25076	-0.00043	5.14E-07	0.253152	0.00618337
3	533.513	534.208	315.063	232.346	0.058586	-0.25237	-0.00041	9.61E-05	0.254306	0.00611604
4	533.473	534.167	314.992	232.289	0.058332	-0.25111	-0.00045	6.11E-05	0.25206	0.00625466
5	533.42	534.128	315.068	232.348	0.057688	-0.24895	-0.00044	5.02E-05	0.251247	0.00635366
6	533.411	534.12	315.135	232.242	0.057365	-0.24794	-0.00047	0.0001008	0.250559	0.00629764
7	533.428	534.143	315.006	232.273	0.058225	-0.25408	-0.00045	2.01E-05	0.262495	0.00650185
8	533.46	534.17	315.052	232.233	0.058778	-0.25566	-0.00044	5.27E-06	0.263885	0.00627201
9	533.422	534.14	315.027	232.351	0.057996	-0.2488	-0.00039	1.48E-05	0.250702	0.00615827
10	533.577	534.29	314.999	232.348	0.057566	-0.24621	-0.0003	-2.41E-05	0.246266	0.00617503
11	533.575	534.286	315.012	232.37	0.056499	-0.2405	-0.00035	7.46E-05	0.237289	0.00625567
12	533.655	534.366	315.067	232.277	0.056115	-0.23778	-0.00043	0.00010217	0.232727	0.00611177
13	533.513	534.233	315.176	232.341	0.056684	-0.23814	-0.00039	0.00016513	0.232091	0.0061377
14	533.568	534.262	315.173	232.302	0.055452	-0.23403	-0.00039	0.0001506	0.227874	0.00598129
15	533.524	534.2	315.259	232.334	0.056502	-0.23887	-0.00034	0.00021269	0.233756	0.00604319
16	533.562	534.246	315.185	232.349	0.05628	-0.23777	-0.00039	0.00019818	0.231647	0.0060954
17	533.559	534.259	315.215	232.27	0.056902	-0.24152	-0.00041	0.00023239	0.236348	0.00623434
18	533.512	534.197	315.275	232.269	0.056622	-0.23906	-0.00041	0.00030403	0.232168	0.00608076
19	533.529	534.226	315.285	232.179	0.057683	-0.24373	-0.00051	0.00032557	0.238171	0.00607298
20	533.478	534.194	315.281	232.243	0.05713	-0.23968	-0.00041	0.00036854	0.234892	0.0061124
21	533.671	534.344	315.446	232.301	0.058352	-0.24676	-0.00044	0.00037715	0.243875	0.00637704
22	533.474	534.183	315.272	232.247	0.058201	-0.24608	-0.00047	0.00029737	0.244	0.00602461
23	533.555	534.262	315.254	232.204	0.05805	-0.24647	-0.00052	0.00026479	0.245448	0.00606418
24	533.515	534.233	315.278	232.145	0.057257	-0.24445	-0.00059	0.00024868	0.244163	0.00622804
25	533.511	534.23	315.264	232.178	0.057851	-0.24537	-0.00058	0.00024201	0.243636	0.00617293
26	533.506	534.228	315.24	232.183	0.057787	-0.24353	-0.0006	0.00023336	0.237909	0.00616603
27	533.546	534.249	315.257	232.272	0.057408	-0.24201	-0.00056	0.00024723	0.235486	0.00611918
28	533.564	534.253	315.205	232.263	0.056719	-0.24051	-0.00055	0.00021232	0.237431	0.00624082
29	533.539	534.212	315.194	232.22	0.056075	-0.23718	-0.00052	0.00019125	0.232612	0.00622347
Mean	533.5158	534.219	315.1523	232.2752	0.057409	-0.24464	-0.00045	0.00015804	0.243073	0.006187686
Median	533.513	534.219	315.1805	232.2725	0.057624	-0.24491	-0.00043	0.00017819	0.243756	0.00617398
StdDev	0.062114	0.057584	0.126057	0.059514	0.000838	0.005522	7.48E-05	0.00012051	0.009497	0.0001138

Table 18. Intrinsic Parameters of Calibration for Data Set 2 using Ceres 2-Step Solve Full

Set	Ceres 2-Step Solve									
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	Reprj Error
0	533.534	534.213	314.975	232.86	0.057886	-0.25064	-1.74E-05	7.88E-07	0.254	0.00639809
1	533.484	534.17	314.991	232.862	0.058261	-0.25151	-1.86E-05	2.11E-07	0.254322	0.00629411
2	533.495	534.178	314.994	232.835	0.058044	-0.25146	-1.96E-05	1.56E-06	0.253897	0.00624895
3	533.545	534.215	314.923	232.89	0.05854	-0.2531	-1.99E-05	3.08E-06	0.255452	0.00617792
4	533.506	534.172	314.903	232.888	0.058295	-0.25198	-2.15E-05	2.81E-06	0.253305	0.00632738
5	533.446	534.125	314.989	232.951	0.057579	-0.24958	-2.11E-05	2.37E-06	0.252372	0.00642472
6	533.439	534.121	314.988	232.876	0.057219	-0.24828	-2.29E-05	2.18E-06	0.2513	0.00638073
7	533.46	534.147	314.976	232.881	0.058018	-0.25425	-2.15E-05	1.08E-06	0.263011	0.0065739
8	533.497	534.182	315.042	232.818	0.058481	-0.25536	-2.07E-05	6.85E-07	0.263816	0.00634025
9	533.449	534.145	314.998	232.869	0.057785	-0.2486	-1.85E-05	6.96E-07	0.250475	0.00621215
10	533.596	534.293	315.032	232.75	0.057414	-0.24597	-1.46E-05	-1.02E-06	0.245866	0.00620741
11	533.595	534.29	314.907	232.846	0.056357	-0.24039	-1.83E-05	9.11E-07	0.237156	0.00630341
12	533.671	534.361	314.927	232.856	0.055967	-0.2379	-2.17E-05	1.51E-06	0.233083	0.00618179
13	533.517	534.219	314.944	232.866	0.056775	-0.23957	-1.93E-05	2.75E-06	0.234306	0.00620047
14	533.571	534.25	314.958	232.82	0.055491	-0.23516	-1.99E-05	2.91E-06	0.229794	0.00604219
15	533.522	534.188	314.962	232.793	0.056628	-0.24041	-1.79E-05	4.03E-06	0.236358	0.00609852
16	533.556	534.23	314.91	232.876	0.056371	-0.2391	-2.03E-05	4.18E-06	0.234079	0.00616187
17	533.557	534.245	314.895	232.828	0.056946	-0.24274	-2.27E-05	5.68E-06	0.238872	0.00631326
18	533.507	534.183	314.86	232.828	0.056756	-0.24077	-2.28E-05	6.65E-06	0.235391	0.00616955
19	533.534	534.218	314.837	232.87	0.057817	-0.24555	-2.73E-05	7.82E-06	0.241918	0.00619936
20	533.484	534.19	314.758	232.8	0.057503	-0.24256	-2.19E-05	7.44E-06	0.239454	0.00620938
21	533.683	534.345	314.909	232.904	0.058702	-0.24952	-2.37E-05	7.89E-06	0.248452	0.00648608
22	533.475	534.17	314.84	232.892	0.058455	-0.24828	-2.54E-05	6.59E-06	0.247868	0.00612828
23	533.548	534.241	314.862	232.921	0.058108	-0.24736	-2.78E-05	5.20E-06	0.247577	0.00618118
24	533.511	534.209	314.907	232.958	0.057154	-0.24458	-3.10E-05	5.12E-06	0.245374	0.00636774
25	533.503	534.202	314.907	232.981	0.057689	-0.24514	-3.11E-05	4.96E-06	0.244263	0.00630806
26	533.514	534.211	314.897	233.014	0.057815	-0.24444	-3.24E-05	6.39E-06	0.239919	0.00630763
27	533.554	534.23	314.897	233.045	0.057479	-0.24316	-3.05E-05	6.93E-06	0.237702	0.00624525
28	533.572	534.234	314.894	233.021	0.056612	-0.24117	-2.98E-05	4.69E-06	0.239439	0.00635711
29	533.548	534.194	314.921	232.943	0.05585	-0.23691	-2.80E-05	4.55E-06	0.232577	0.00632937
Mean	533.5291	534.2124	314.9268	232.8847	0.0574	-0.24551	-2.2925E-05	3.688E-06	0.244713	0.006272537
Median	533.5195	534.21	314.9155	232.873	0.057541	-0.24534	-2.1598E-05	3.554E-06	0.244819	0.00627153
StdDev	0.05693	0.055141	0.061078	0.068533	0.000848	0.005369	4.6708E-06	2.496E-06	0.009002	0.00011488

Table 19. Intrinsic Parameters of Calibration for Data Set 3 using Ceres Circle Correction Full

Set	Ceres Circle Solve									Reprj Error
	Fx	Fy	Cx	Cy	k1	k2	p1	p2	k3	
0	539.871	540.175	316.064	231.492	0.013209	-0.10213	-0.00056	0.00045668	0.091905	0.0422353
1	539.924	540.234	316.063	231.421	0.014259	-0.1075	-0.00058	0.00040497	0.099399	0.0423188
2	539.82	540.108	316.121	231.346	0.01258	-0.1005	-0.0006	0.00046086	0.087923	0.0414042
3	540.186	540.457	316.273	231.349	0.011772	-0.08559	-0.00062	0.00045229	0.051051	0.0419875
4	540.132	540.445	316.291	231.392	0.010722	-0.08137	-0.00055	0.00053746	0.046364	0.0420739
5	539.914	540.31	316.399	231.41	0.008724	-0.07835	-0.00061	0.00051196	0.055951	0.0422253
6	539.868	540.254	316.31	231.305	0.009894	-0.08613	-0.00062	0.00054267	0.06724	0.0425206
7	539.872	540.181	316.135	231.234	0.012167	-0.10422	-0.00068	0.0004789	0.10254	0.0421998
8	539.93	540.298	316.262	231.181	0.012848	-0.10464	-0.00062	0.00056176	0.101286	0.0418036
9	539.928	540.336	316.404	231.304	0.011304	-0.08995	-0.00066	0.00060212	0.072292	0.0413141
10	540.262	540.707	316.05	231.558	0.01023	-0.08225	-0.00037	0.00027196	0.058035	0.0420963
11	540.466	540.875	316.049	231.445	0.007325	-0.06681	-0.00059	0.00039358	0.034328	0.0419976
12	540.507	540.949	316.069	231.382	0.006469	-0.06426	-0.00067	0.00051091	0.030779	0.0419699
13	540.014	540.469	316.285	231.364	0.009394	-0.07658	-0.0008	0.00065917	0.043935	0.0417109
14	540.211	540.615	316.235	231.277	0.008424	-0.07296	-0.00089	0.00060646	0.039676	0.0416762
15	539.742	540.039	316.273	231.28	0.010094	-0.07852	-0.00092	0.00062002	0.042322	0.0418116
16	539.682	539.996	316.173	231.329	0.009694	-0.07723	-0.00097	0.00053806	0.040691	0.0415423
17	539.837	540.072	316.323	231.344	0.010688	-0.08404	-0.00103	0.00052356	0.050851	0.0414381
18	539.705	539.89	316.396	231.357	0.009951	-0.07956	-0.00101	0.0006751	0.042917	0.0413765
19	539.532	539.719	316.519	231.165	0.011528	-0.08651	-0.00121	0.00071245	0.053803	0.0408689
20	539.462	539.704	316.771	231.411	0.010879	-0.08164	-0.00107	0.00081571	0.046815	0.0416902
21	539.704	539.927	316.93	231.408	0.010881	-0.08092	-0.00116	0.00076986	0.045745	0.0410912
22	539.386	539.678	316.704	231.385	0.010337	-0.07986	-0.00113	0.00081113	0.046073	0.0402987
23	539.494	539.802	316.643	231.401	0.00821	-0.0716	-0.00112	0.00077795	0.038593	0.0399818
24	539.334	539.613	316.646	231.373	0.007434	-0.07175	-0.00116	0.00078371	0.04297	0.0399871
25	539.493	539.782	316.547	231.46	0.007826	-0.07318	-0.00109	0.00079737	0.042921	0.040013
26	539.575	539.824	316.551	231.44	0.008739	-0.0747	-0.001	0.0007859	0.038929	0.0400056
27	539.821	540.066	316.53	231.525	0.007431	-0.06681	-0.00096	0.00078255	0.026597	0.0390337
28	539.822	539.962	316.458	231.484	0.006362	-0.06723	-0.00097	0.00073875	0.037219	0.0392986
29	539.699	539.851	316.467	231.511	0.006767	-0.06854	-0.00087	0.00076551	0.040068	0.0409245
Mean	539.8398	540.1446	316.3647	231.3778	0.009871	-0.08151	-0.00084	0.00061165	0.053974	0.041296527
Median	539.8295	540.09	316.3165	231.3835	0.010022	-0.07971	-0.00088	0.00060429	0.045909	0.04160925
StdDev	0.289908	0.347652	0.229164	0.09278	0.002065	0.01184	0.000232	0.00014784	0.021319	0.000932555

REFERENCES

- [1] P. M. James Bowman, "camera_calibration," ROS.
- [2] Itseez, "The OpenCV Reference Manual," <http://opencv.org>, 2014.
- [3] C. Lewis, "ROS Industrial Calibration Library," ROS-Industrial.
- [4] K. M. e. a. Sameer Agarwal, "Ceres Solver," <http://ceres-solver.org>.
- [5] MathWorks, "What is Camera Calibration," MathWorks, 2017. [Online]. Available: <https://www.mathworks.com/help/vision/ug/camera-calibration.html>.
- [6] OpenCV, "Camera Calibration and 3D Reconstruction," OpenCV, [Online]. Available: https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html.
- [7] Y. Morvan, "Acquisition, Compression and Rendering of Depth and Texture for Multi-View video," epixea.com, 2009.
- [8] D. C. Brown, "Decentering Distortion of Lenses," *Photogrammetric Engineering*, vol. 32, no. 3, pp. 444-462, 1966.
- [9] Z. Zhang, "A Flexible New Technique for Camera Calibration," Microsoft Research, Redmond Washington, 1998.
- [10] Asus, Artist, *Asus Xtion Pro*. [Art]. Asus.
- [11] Itseez, "Open Source Computer Vision Library," <https://github.com/itseez/opencv>, 2015.
- [12] C. L. Geoffrey Chiou, "IC2".
- [13] R. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 323-344, 1987.

VITA

Geoffrey Chiou is from San Antonio, TX. He studied mechanical engineering and earned his Bachelor's in Mechanical Engineering from The University of Texas at San Antonio. He will earn his Master's in Mechanical Engineering from The University of Texas at San Antonio after he submits this document. His future plans include not attending a Ph.D. program.