

5-1-2014

# Enhancement to Camera Calibration: Representation, Robust Statistics, and 3D Calibration Tool

Qiaotian Li

*University of Wisconsin-Milwaukee*

Follow this and additional works at: <https://dc.uwm.edu/etd>



Part of the [Electrical and Electronics Commons](#)

---

## Recommended Citation

Li, Qiaotian, "Enhancement to Camera Calibration: Representation, Robust Statistics, and 3D Calibration Tool" (2014). *Theses and Dissertations*. 469.

<https://dc.uwm.edu/etd/469>

This Dissertation is brought to you for free and open access by UWM Digital Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of UWM Digital Commons. For more information, please contact [open-access@uwm.edu](mailto:open-access@uwm.edu).

ENHANCEMENT TO CAMERA CALIBRATION: REPRESENTATION,  
ROBUST STATISTICS, AND 3D CALIBRATION TOOL

by

Qiaotian LI

A Dissertation Submitted in

Partial Fulfillment of the

Requirements of the Degree of

Doctor of Philosophy

in Engineering

at

The University of Wisconsin–Milwaukee

May 2014



# ABSTRACT

## ENHANCEMENT TO CAMERA CALIBRATION: REPRESENTATION, ROBUST STATISTICS, AND 3D CALIBRATION TOOL

by

Qiaotian LI

The University of Wisconsin–Milwaukee, 2014

Under the Supervision of Professor Brian, S.R. Armstrong

This thesis demonstrates the **enhancement to camera calibration** in three aspects: representation of pose, robust statistics and 3D calibration tool. Camera calibration is the reconstruction of digital camera information based on digital images of an object in 3D space, since the digital images are 2D projections of a 3D object onto the camera sensor. Camera calibration is the estimation of the interior orientation (IO) parameters and exterior orientation (EO) parameters of a digital camera. Camera calibration is an essential part of image metrology. If the quality of camera calibration cannot be guaranteed, neither can the reliability of the subsequent analysis and applications based on digital images.

The first enhancement of camera calibration is in representation of pose. A formal definition of “**singularity of representation**” is given mathematically. An example is offered to show how singularity can lead to difficulty or failure in optimization. The **spherical coordinate system** is introduced as a representation method instead of other widely-used representations. The spherical coordinate system represents camera poses according to camera calibration tool images in digital image processing. With the introduction of the  $\nu$  frame in digital images, the singularities of spherical coordinate system are demonstrated mathematically.

The application of **robust statistics** in optimization is the second enhancement of camera calibration. In photogrammetry, it is typical to collect thousands of observed data points for bundle adjustment. Unexpected outliers in observed data are unavoidable, and thus, the algorithm accuracy may not reach our goal. The least squares estimator is a widely used estimation method in camera calibration, but its sensitivity to outliers makes the algorithm unreliable, and it can even fail to fit the observations. By closely analyzing and comparing the characteristics of the least squares estimator, robust estimators with alternative assumptions are shown to detect and de-weight outliers that are not well processed with the classical assumptions, and provide a reliable fit to the observations. Among all possible robust estimators, two robust estimators from M-estimator family are applied to optimization in existing camera calibration algorithm. The robustified method can considerably improve accuracy for camera calibration estimation.

A **new metric**  $\bar{D}$  is introduced, which is the distance between two camera calibrations considering all of the estimated camera IO parameters.  $\bar{D}$  can be used to evaluate the performance among various estimators. After applying the robust estimator, the system improves the accuracy and performance in camera calibration up to 25%. The influence of a robustified estimator modification is also considered. It is established that the modification has impact on the estimation accuracy.

The third enhancement is the design and application of a **3D calibration tool** for data collection. An all-new 3D calibration tool is designed to improve camera calibration accuracy over the 2D calibration tool. The comparison of the 3D and 2D calibration tools is conducted experimentally and theoretically. The experimental analysis is based on camera calibration results and the corresponding  $\bar{D}$  matrix, which shows that the 3D calibration tool improves accuracy. The mathematical analysis is based on the calculated covariance matrix of camera calibration without other impact factors. The experimental and theoretical analyses show that the 3D calibration tool can obtain more accurate calibration results compared with the 2D calibration tool, establishing that a carefully designed 3D

calibration tool will yield better estimates than a 2D calibration tool.

©Copyright by Qiaotian LI 2014

All Rights Reserved

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Overview of Thesis Topics . . . . .	1
1.2	Camera Calibration . . . . .	2
1.2.1	Overview of Camera Calibration . . . . .	2
1.2.2	Camera Calibration in Different Applications . . . . .	2
1.3	Bundle Adjustment (BA) for High-accuracy Camera Calibration . . . .	3
1.4	Research Purpose and Approach . . . . .	3
1.5	Thesis Organization . . . . .	5
<b>2</b>	<b>Literature Review</b>	<b>6</b>
2.1	Camera Geometries and Camera Models . . . . .	6
2.1.1	Coordinate Representations and Coordinate Transformation of Object in 3D space . . . . .	6
2.1.2	Camera Geometries . . . . .	11
2.1.3	Camera Models . . . . .	19
2.2	Camera Calibration . . . . .	28
2.2.1	Definition of Camera Calibration . . . . .	28
2.2.2	Camera Calibration Methods . . . . .	29
2.2.3	Camera Calibration Methods: Linear Algorithm . . . . .	30
2.2.4	Camera Calibration Methods: Non-linear and Hybrid Algorithm	32
2.3	Camera Calibration: Bundle Adjustment (BA) . . . . .	36
2.3.1	Bundle Adjustment: Optimization by the Newton-like Algorithm	37
2.3.2	Bundle Adjustment: Issues . . . . .	41
2.3.3	Bundle Adjustment: Evaluation and Improvement . . . . .	42

2.4	Summary	46
3	Representation Enhancement: Spherical Coordinate Representation	47
3.1	Singularity of Representation	47
3.1.1	Definition of “Singularity of Representation” (SoR)	47
3.1.2	Loss of One Degree of Freedom (DOF)	48
3.2	Singularity of Representation in Bundle Adjustment	49
3.2.1	The Jacobian Matrix and the Hessian Matrix	49
3.2.2	Singularity of Representation in Bundle Adjustment	52
3.2.3	An Example of Singularity of Representation	54
3.3	Singularity of Spherical Coordinate Representation	59
3.3.1	Spherical Coordinate Representation	59
3.3.2	Frames: the $t$ Frame, the $c$ Frame, the $i$ Frame, and the $v$ Frame	61
3.3.3	Structure of $J_{w_{S_{\hat{e}}}}$ in the $v$ Frame	65
3.3.4	Rank of ${}^v J_{w_{S_{\hat{e}}}}$	75
3.3.5	Nominal Verification of Singularity of Representation	77
3.4	Summary	79
4	Optimization Enhancement: Robust Statistics	81
4.1	Least Squares Estimation in Bundle Adjustment	81
4.1.1	Assumptions of the Least Squares Estimator	81
4.1.2	Non-Gaussian Distribution of Residuals in Observations	82
4.2	Robust Estimation in Bundle Adjustment	88
4.2.1	General Idea of Robust Estimation	88
4.2.2	Robustified Iterative Estimation	89
4.3	Robustified Iterative Formulas in Bundle Adjustment	93
4.3.1	Robustified Iterative Formulas	93

4.3.2	Specific Robustified Formulas . . . . .	95
4.4	$\bar{D}$ , a New Metric for Measuring the Distance Between Camera Calibrations . . . . .	102
4.5	Performance Analysis of Robustified Optimization . . . . .	105
4.5.1	Algorithm Performance Analysis . . . . .	105
4.5.2	Adjustment of the Robustified Estimators . . . . .	107
4.6	Summary . . . . .	112
5	Experimental Enhancement: 3D Calibration Tool	113
5.1	Camera Calibration Tool . . . . .	113
5.1.1	2D Camera Calibration Tool . . . . .	113
5.1.2	3D Camera Calibration Tool . . . . .	115
5.2	Camera Calibration with 2D and 3D Calibration Tools . . . . .	118
5.2.1	Camera Calibration Experiment Setup and Data Collection . .	118
5.2.2	Image Processing in Camera Calibration . . . . .	122
5.3	Camera Calibration Results Analysis . . . . .	127
5.3.1	Data Collection and Camera Calibration Results . . . . .	127
5.3.2	$\hat{\chi}^*$ , the Estimation of Ideal Camera Model . . . . .	127
5.3.3	Experimental Analysis of 2D and 3D Camera Calibration Tools	127
5.3.4	Mathematical Analysis of 2D and 3D Camera Calibration Tools . . . . .	132
5.4	Summary . . . . .	138
6	Final Chapter and Future Work	139
6.1	Key Findings of This Research . . . . .	139
6.2	Future Work . . . . .	140

## List of Figures

2.1	Euler angles of X, Y, Z axes: pitch $\omega$ , yaw $\phi$ and roll $\kappa$ . . . . .	9
2.2	Coordinate transformation: from the calibration tool (target) frame to the camera frame. . . . .	11
2.3	Single camera geometry: mapping ${}^tP_a \in R^3$ onto image plane to get ${}^iP_a \in R^2$ . . . . .	12
2.4	Dual-camera geometry: two cameras observe the same landmark. . . . .	17
2.5	Multi-station geometry: combination of $k = 1, 2, 3, \dots, n_c$ single camera geometries by $k = 1, 2, 3, \dots, n_c$ camera stations. . . . .	18
2.6	The pinhole camera model: mapping ${}^tP_a \in R^3$ to ${}^iP_a \in R^2$ without lens distortion. . . . .	20
2.7	Radial distortion (No.1) and Tangential distortion (No.2), distortion effects shown at $20\times$ magnification. . . . .	22
2.8	Camera model with lens distortion, showing ${}^iP_a$ (or ${}^qP_a$ ) and ${}^jP_a$ (or ${}^pP_a$ ). . . . .	23
3.1	The Jacobian matrix of $\hat{\mathbf{u}}$ (a) and the Jacobian matrix of each camera pose (b): the black area means a zero value and the white area means a non-zero value. . . . .	50
3.2	The Hessian matrix: the black area means a zero value and the white area means a non-zero value. . . . .	51
3.3	The Hessian matrix of camera pose: the black area means a zero value and the white area means a non-zero value. . . . .	52
3.4	A pose in spherical calibration tool coordinates. The bearing ray connects the root frame (in this case the calibration tool frame) to the object frame (in this case the camera frame). . . . .	60



3.5	The $t$ frame and its axes are on the calibration tool. The $v$ frame and its axes are in the image of calibration tool, which is defined by three landmarks defined by the image of three landmarks ${}^tP_{L_j}$ : $L_1$ is the origin; $L_1$ and $L_2$ give us $t_x$ ; $L_1$ and $L_3$ give us $t_y$ ; $t_z$ is perpendicular to $t_x$ and $t_y$ by $t_z = \text{cross}(t_x, t_y)$ . . . . .	63
3.6	A pose in spherical calibration tool coordinates and its projection on to the $v$ frame: $I_0$ is the nominal plane based on ${}^s\phi = 0^\circ$ , ${}^s\omega = 0^\circ$ , and ${}^s\kappa = 0^\circ$ ; $I_1, \dots, I_k$ is the planes based on other possible ${}^s\phi$ , ${}^s\omega$ , and ${}^s\kappa$ values. . . . .	69
4.1	Gaussian (normal) distribution. . . . .	82
4.2	A landmark that generates a large residual value in bundle adjustment. . . .	83
4.3	A landmark that generates a small residual value in bundle adjustment. . .	84
4.4	Residuals does not fit a normal distribution: four different camera calibration data sets collected by three individuals. The No.1 data set was collected by Prosilica GC1290 industrial camera with Schneider 10mm lens. The No.2, No.3, and No.4 data sets were collected by AVT-Stingray F033B industrial camera with Edmund 10mm lens. . . . .	86
4.5	Distribution of residual function and normal distribution: four different camera calibration data sets collected by three individuals. . . . .	87
4.6	$\rho(R(\mathbf{u}))$ of the least squares estimator. . . . .	92
4.7	$w(R(\mathbf{u}))$ of the least squares estimator. . . . .	92
4.8	$\rho(r_i(\mathbf{u}))$ of the Cauchy distribution based robust M-estimator. . . . .	96
4.9	$w(r_i(\mathbf{u}))$ of the Cauchy distribution based robust M-estimator. . . . .	97
4.10	$\rho(r_i(\mathbf{u}))$ of the Welsch distribution based robust M-estimator. . . . .	98
4.11	$w(r_i(\mathbf{u}))$ of the Welsch distribution based robust M-estimator. . . . .	98
4.12	Distribution of the normal distribution, the Cauchy distribution, and the Welsch distribution. . . . .	99

4.13	Weighting function of the normal distribution, the Cauchy distribution, and the Welsch distribution. . . . .	100
4.14	$\psi$ function of the Cauchy distribution. . . . .	101
4.15	$\psi$ function of the Welsch distribution. . . . .	101
4.16	Residual and weighting function of the Welsch distribution. . . . .	107
4.17	The Welsch distribution after $5\times$ magnification of $c$ : y axis scale is 0 to 10. . . . .	108
4.18	The Welsch distribution after $5\times$ magnification of $c$ : y axis scale is 0 to 20. . . . .	108
4.19	The Welsch distribution after $\frac{1}{5}\times$ magnification of $c$ : y axis scale is 0 to 0.1. . . . .	109
4.20	The Welsch distribution with tuning the $c$ parameters: $5\times$ , Original, $\frac{1}{5}\times$ . . . . .	109
4.21	The magnification of the Welsch distribution with tuning the $c$ parameters: $5\times$ , Original, $\frac{1}{5}\times$ . . . . .	110
4.22	The Cauchy distribution with tuning the $c$ parameters: $5\times$ , Original, $\frac{1}{5}\times$ . . . . .	111
4.23	The magnification of the Cauchy distribution with tuning the $c$ parameters: $5\times$ , Original, $\frac{1}{5}\times$ . . . . .	111
5.1	2D calibration tools: No.1 adopted from Remondino and Fraser [54]; No.2 adopted from Zhang [68]. . . . .	114
5.2	The 2D calibration tool with 1 white central starburst landmark around by four circles and 72 black starburst landmarks. . . . .	115
5.3	3D calibration tools: No.1 adopted from Abraham and Hau[2]; No.2 adopted from Remondino and Fraser [54]; No.3 adopted from Kunii and Chikatsu [36]; No.4 adopted from Heikkila [33]. . . . .	116
5.4	Design of a 3D calibration tool. . . . .	117
5.5	The 3D calibration tool with 1 white central landmark surrounded by four circles and 68 black starburst landmarks. . . . .	117
5.6	The 3D calibration tool with the raised middle area. . . . .	118
5.7	Camera unit (Left), Digital images displayed by PC monitor (Middle) and 2D calibration tool (Right). . . . .	119

5.8	Digital images of the 3D calibration tool at various positions: 1, 2, 3, 4, and 5 are in the front with the tool tilting towards different directions; 6, 7, 8, 9, and 10 are in the rear with the tool tilting towards different directions. .	121
5.9	Digital images of the 3D calibration tool with $\kappa$ angle rotation at $90^\circ$ , $180^\circ$ and $270^\circ$ : 1, 2, and 3 are collected in the front position; 4, 5, and 6 are collected in the rear position. . . . .	122
5.10	Distribution of landmarks of a image set: Data set was collected with Prosilica GC1290 camera and Edmund 18mm lens with the 2D calibration tool. . . . .	123
5.11	Distribution of landmarks of a image set: Data set was collected with Prosilica GC1290 camera and Edmund 18mm lens with the 3D calibration tool. . . . .	124
5.12	Diversity of camera poses in experimental camera calibration set: Data set was collected with Prosilica GC1290 camera and Edmund 18mm lens, azimuth and elevation v.s. $^s\kappa$ angle with the 2D calibration tool. . . . .	125
5.13	Diversity of camera poses in experimental camera calibration set: Data set was collected with Prosilica GC1290 camera and Edmund 18mm lens, azimuth and elevation v.s. range with the 2D calibration tool. . . . .	125
5.14	Diversity of camera poses in experimental camera calibration set: Data set was collected with Prosilica GC1290 camera and Edmund 18mm lens, azimuth and elevation v.s. $^s\kappa$ angle with the 3D calibration tool. . . . .	126
5.15	Diversity of camera poses in experimental camera calibration set: Data set was collected with Prosilica GC1290 camera and Schneider 18mm lens, azimuth and elevation v.s. range with the 3D calibration tool. . . . .	126

5.16	Difference values of STD of the 3D and the 2D calibration tools during camera calibration: a blue cross means the 3D calibration tool makes the standard deviation of camera calibration smaller than the 2D calibration tool; while a red circle means the 2D calibration tool makes STD smaller than the 3D calibration tool. . . . .	133
5.17	STD values of the 3D and 2D calibration tools corresponding to camera poses in 6 DOF: a blue cross is the STD values of the 3D calibration tool; a red circle is the STD values of the 2D calibration tool. . . . .	134
5.18	Difference of STD of the 3D and 2D calibration tools corresponding to camera poses in 6 DOF: a blue cross means the 3D calibration tool makes the STD of camera calibration smaller than the 2D calibration tool; while a red circle means the 2D calibration tool makes STD smaller than the 3D calibration tool. . . . .	135

## List of Tables

2.1	Coordinates on image plane affected by lens distortion. . . . .	24
4.1	Distances among 7 different camera calibration results with AVT-StingrayF033B industrial camera and Edmund 10mm optical lens. K1, K2, K3, K4, K5 were collected by one individual; R1 and R2 were collected by the other individual. . . . .	104
4.3	Performance Analysis of the LSE and Robustified Estimator. . . . .	105
4.2	Upper triangular part of $\bar{D}$ of least squares estimation in bundle adjustment, where $\bar{D}$ is a symmetrical matrix. . . . .	106
4.4	Performance Analysis of the LSE and Robustified Estimator. . . . .	112
5.1	Camera Calibration Results of the 2D camera calibration tool with focus of the lens around 1 meter. . . . .	128
5.2	Camera Calibration Results of the 3D camera calibration tool with focus of the lens around 1 meter. . . . .	129
5.3	The upper triangular part of the symmetric $\bar{D}_{2D}$ of the 2D calibration tool with focus of the lens around 1 meter. . . . .	130
5.4	The upper triangular part from the symmetric $\bar{D}_{3D}$ of the 3D calibration tool with focus of the lens around 1 meter. . . . .	130
5.5	Statistics of $\bar{D}$ based on the 2D and 3D calibration tools, 8 image sets for each calibration tool with focus of the lens around 1 meter. . . . .	130
5.6	Statistics of $\bar{D}$ based on the 2D and 3D calibration tools, 8 image sets for each calibration tool with focus of the lens around 2 meters. . . . .	131
5.7	Statistics of $\bar{D}$ based on the 2D and 3D calibration tools, 8 image sets for each calibration tool with focus of the lens around 2 meter. . . . .	131

5.8	Improvement of $\bar{D}$ based on the 2D and 3D calibration tools: simulated $\tilde{\chi}_j$ corresponding to covariance matrices of camera calibration. . . . .	137
-----	--	-----

## ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Brian S. R. Armstrong, for his guidance and care throughout my PhD work. I will always remember his edification on how to become a researcher, especially the time he spent with me on research during the weekends. He once told me “Give a man a fish and you feed him for a day. Teach a man to fish and you feed him for a lifetime” in front of the elevator at the second floor of EMS building. The training and practice I received in his lab have become the most essential part of my skill set for the future. There is a lot to say, but no words can truly describe my emotion! I would like to thank Dr. Chiu-Tai Law and Dr. Brian S. R. Armstrong who gave me the chance to be a teaching assistant during my studies. This great experience not only supports my life in the United States, but also has helped me to develop in many aspects. I have taught more than two hundred non-electrical engineering students electrical engineering! It has been fun!

I would like to thank my family, which has shaped the person I am today. My parents brought me to this world and raised me selflessly. My mother taught me to have a spirit of perseverance, and my father passed on his wisdom. I met my wife at this university after arriving in the United States. She and our daughter Sophia encourage me to continue my work when I feel unconfident and exhausted from PhD research. The support from my parents-in-law is certainly priceless and makes my life less stressful. Without my family, I cannot imagine where I would be right now.

Five years ago, I was just another Columbus, coming to a new continent with dreams and expectations. Everything was foreign, and there were no friends or family. Now, I am a married man who has a little angel. It has been such a memorable and wonderful journey. A PhD is not the end. It is only a new beginning. I still have dreams and adventures to keep me moving forward.

**I really appreciate everyone who came into my life with kindness and love. God bless you all!**

# 1 Introduction

## 1.1 Overview of Thesis Topics

In order to obtain information of an object in 3D space from its 2D digital images, we need to know the 10 interior orientation (IO) parameters and 6 exterior orientation (EO) parameters of a digital camera. **Camera Calibration** is the process by which we reconstruct the IO and EO of the camera. The reconstruction depends on analysis of the digital images, which involves image processing, image metrology, statistics and numerical methods. There are several ways to do camera calibration for various applications. In close range **photogrammetry**<sup>1</sup>, the most powerful method is **bundle adjustment (BA)**, which can take a very large sample and estimate IO and EO as a bundle [59].

The problem is that it is hard to get an accurate result, because BA leads to simultaneous estimation of 1000 or more parameters<sup>2</sup>, and it is preferable to determine the values of IO parameters to the fourth digit. A low-accuracy camera calibration result will affect the next stage application such as the compensation to the blur effect of human head motion during MRI scanning<sup>3</sup>. Reasons for performance difficulty are: possible singularity of representation; non-Gaussian distribution of residuals; and the need for a calibration experiment that better constrains the camera parameters.

- 
1. The word derived from Greek words, where *photo* means **light**, *gramma* means **something drawn or written**, *metron* means **to measure**. Thus, **photogrammetry** is the technology of obtaining reliable information about physical objects and environment by procedure of recording, measuring and interpreting photographic images.
  2. If 200 images of a calibration tool with 73 landmarks are in a data set, there will be  $200 \times 6$  parameters for camera pose,  $73 \times 3$  for landmarks and 10 camera interior parameters to be estimated. Thus, the total number of estimated parameters is more than 1000.
  3. This research is sponsored by NIH/NIDA R01DA021146 funding to develop a single camera motion tracking system for MRI.



## **1.2 Camera Calibration**

### **1.2.1 Overview of Camera Calibration**

Camera calibration is an essential part of image metrology. It offers the camera interior geometry parameters (IO) and the rotation and translation (EO) of the calibration tool relative to the camera. With IO and EO information, we can reconstruct the mapping from 3D landmarks on the calibration tool onto the camera's image sensor. The more accurate the estimated parameters, the better compensation can be performed for the next stage of the application.

In the data collection stage, a camera will take photos of a camera calibration tool. Digital images are the only data source. There are two basic sensor-object geometries: the single-camera geometry, which is also the base of multi-camera convergent geometry, and the dual-camera geometry. The particular geometry models the mapping from the landmarks on the calibration tool in 3D space to its 2D image on the image sensor, which is what to recover. The coordinates transformation among several coordinate systems is necessary, so the choice of representation for the coordinate system is an issue.

Based on camera geometry, there are linear methods, such as the Direct Linear Transformation, that offer a closed-form simplified solution. There are also non-linear methods, such as bundle adjustment, that offer the recovery of the full set of camera modeling parameters. Camera calibration itself is not necessarily linear or non-linear; the choice depends on the application requirements and accuracy. Thus, both linear and non-linear methods have been developed.

### **1.2.2 Camera Calibration in Different Applications**

Mainly, two communities are using camera calibration: the computer vision (CV) community and the photogrammetry community. There are common points and differences

due to varying requirements.

The photogrammetry community requires higher accuracy and more parameters to be obtained and considers high accuracy of full-range parameters to be one of its most important concerns. The difference between photogrammetry and close-range photogrammetry is that the latter is applied when the focal length is a perceptible part of the camera-subject separation.

The computer vision community may only need to recover part of the IO parameters. Some computer applications recover focal length only without any lens distortion modeling. However, in photogrammetry, it is customary to model and recover the full lens distortion parameters. The basic camera-object mapping geometries are used by both communities.

### **1.3 Bundle Adjustment (BA) for High-accuracy Camera Calibration**

Bundle adjustment is considered to be the most powerful tool for camera calibration for close-range photogrammetry. Bundle adjustment has the advantage of estimating IO and EO simultaneously and can accept data with no limitation on size, source, collecting time or collector compared to conventional camera calibration. The complete set of variables are updated and improved during the calculation, which is an optimal estimation. Bundle adjustment requires a non-linear estimator to finish the task. Newton-like iterative method is used as the numerical method; the least squares estimator is common. However the robust estimator is needed due to the non-Gaussian distribution of the residuals.

### **1.4 Research Purpose and Approach**

To enhance the performance and accuracy of camera calibration in photogrammetric applications, I address three challenges of accurate bundle adjustment in this thesis:

### 1. Singularity of representation (SoR)

Singularity of representation (SoR) exists in several research areas with usage of 3D space representation. When SoR occurs, it can significantly degrade the performance of algorithms. In this thesis, I discover the cases of singularity using the spherical-maker coordinate system in camera FOV.

### 2. Application of robust statistics

Robustified statistics is a powerful tool for handling of outliers. Data used in the BA algorithm can be collected at any time and from any source, and does not follow the normal distribution strictly. However, the assumption of normal distribution is a general precondition for classical statistics, like the least squares estimator. Thus, a suitable robustified cost function is necessary for the improvement of bundle adjustment. In this thesis, I find a suitable robust estimator for this estimation task and develop a novel algorithm for the improvement of camera calibration accuracy.

### 3. Development of a 3D camera calibration tool.

Camera calibration can use a 2D calibration tool or a 3D calibration tool. A 3D camera calibration tool will offer extra constraint in the collected data set. The extra information will improve the reliability of Newton-like methods in BA, which I establish mathematically. In this section, I will calculate the expected covariance matrix with a 3D calibration tool, conduct real experiments with the camera calibration system, and analyze the covariance matrix numerically alongside the covariance matrix from the 2D calibration tool.

## **1.5 Thesis Organization**

The thesis has six chapters. Chapter one is the introduction of this research. Chapter two is the literature review, which covers the basics of camera geometries, camera models, camera calibration methods, etc. Chapter three introduces a new coordinate system to represent 3D object. The singularity of this representation is addressed in our application. Chapter four addresses the improvement of camera calibration by introducing robust estimators. Chapter five covers the utilization of the 3D calibration tool designed. The advantages of the 3D calibration tool are analyzed and demonstrated in practice. In the chapter six, the major contributions of this thesis work are listed and future work is described.

## 2 Literature Review

### 2.1 Camera Geometries and Camera Models

#### 2.1.1 Coordinate Representations and Coordinate Transformation of Object in 3D space

An object in 3D space is **uniquely** determined by its position and orientation [39, 43].

- Positions: Cartesian coordinate system and others such as spherical coordinate system; polar coordinate system; homogeneous coordinate system, etc.
- Orientation: Euler angles, which can be transferred to rotation matrix.

#### Coordinate representation of object position in 3D space

There are several different coordinate representations available for the **position** of a landmark in 3D space [41, 43]. To represent the position of a landmark in 3D space, we need to choose a coordinate system. The Cartesian coordinate system (denoted by P) is most widely used [45].

If a transformation between coordinate systems is needed, the position of point  $a$  in the frame  $A$  represented in Cartesian coordinate system is given [14]:

$${}^A\mathbf{P}_a = \begin{bmatrix} {}^AX_a \\ {}^AY_a \\ {}^AZ_a \end{bmatrix} \in R^3 \quad (2.1)$$

where  ${}^AX_a, {}^AY_a, {}^AZ_a$  are the X, Y, Z components of  ${}^A\mathbf{P}_a$ . Thus, point  $a$  in the world coordinate system is

$${}^w\mathbf{P}_a = \begin{bmatrix} {}^wX_a \\ {}^wY_a \\ {}^wZ_a \end{bmatrix} \in R^3. \quad (2.2)$$

The coordinate system is also known as “frame” [14]. The position of a landmark represented by a three-dimensional coordinate representation is relative and depends on which frame the landmark is in. In close-range photogrammetry, there are the world coordinate frame, denoted by  ${}^wP$ ; the camera coordinate frame, denoted by  ${}^cP$ ; the calibration tool coordinate frame, denoted by  ${}^tP$ ; the image coordinate frame, denoted by  ${}^iP$ ; and the pixel coordinate frame, denoted by  ${}^pP$  [41].

### Coordinate representation of object orientation in 3D space

There are several different representations available for the **orientation** of a landmark in 3D space[41, 43]. In this representation, three angles ( $\omega$ ,  $\kappa$ ,  $\phi$ ) about three axes (X, Y, Z) are used to describe the orientation of a rigid body in 3D Euclidean space. Each would be converted to an elementary rotation matrix. The combination of these three elementary matrices gives us the rotation of the rigid body.

- If the axes move with the object, the rotations are in the order yaw ( $\phi$ )  $\rightarrow$  pitch ( $\omega$ )  $\rightarrow$  roll ( $\kappa$ ) . If the axes are fixed, then the rotations are roll  $\rightarrow$  pitch  $\rightarrow$  yaw. In aviation, a positive pitch means the plane nose moves up, and a positive yaw means it is turned to the right. A positive roll means that the right wing moves down. Thus, the right-hand rule is used for three axes relationship.
- The pitch about x ( $\omega$ ) axis:

$$R_{\omega} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_x & -S_x \\ 0 & S_x & C_x \end{bmatrix},$$

where  $S_x$  is  $\sin(x)$  and  $C_x$  is  $\cos(x)$ .

- The yaw ( $\varphi$ ) about y axis

$$R_{\varphi} = \begin{bmatrix} C_y & 0 & S_y \\ 0 & 1 & 0 \\ -S_y & 0 & C_y \end{bmatrix},$$

where  $S_y$  is  $\sin(y)$  and  $C_y$  is  $\cos(y)$ .

- The roll ( $\kappa$ ) about z axis

$$R_{\kappa} = \begin{bmatrix} C_z & -S_z & 0 \\ S_z & C_z & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where  $S_z$  is  $\sin(z)$  and  $C_z$  is  $\cos(z)$ .

### Coordinate transformation of object in 3D space

If we have the value of position of a landmark in 3D space in one frame and want to know the position of this landmark in 3D space in the other frame, it is necessary to do coordinate transformation from the current frame to the destination frame [14, 41]. The origin and orientation of the current frame in the destination frame for coordinate transformation are required.

#### Example:

To find  ${}^cP_a$  (the destination frame) from  ${}^tP_a$  (the current frame), we need:

- The origin of the calibration tool frame, expressed in the camera frame,

$${}^cP_i = \begin{bmatrix} {}^cX_i \\ {}^cY_i \\ {}^cZ_i \end{bmatrix} \quad (2.3)$$

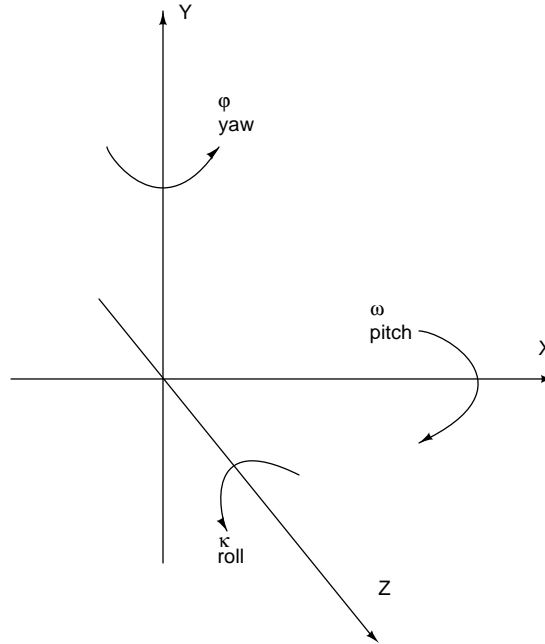


Figure 2.1: Euler angles of X, Y, Z axes: pitch  $\omega$ , yaw  $\varphi$  and roll  $\kappa$ .

- The orientation of the calibration tool frame with respect to the camera frame. The orientation of one coordinate frame with respect to another can be expressed either as:
  - (a) The Euler angles: pitch  $\omega$ , roll  $\kappa$  and yaw  $\varphi$  values, [degrees] or [radians]. See figure 2.1
  - (b) A rotation matrix,  ${}^cR$ .

This notation [14] is understood as

$$\begin{matrix} \text{To frame} \\ \text{From frame} \end{matrix} R \quad (2.4)$$

where the "From frame" is the frame from which the vectors are coming and the "To frame" is the frame to which the vectors are being rotated.

In calculation, a rotation matrix is preferred. However, Euler angles have the minimum parameterization needed for estimation, are easier for humans to read, and also good for



plotting. Euler angles and rotation matrices can be converted to each other. A rotation from one frame to another frame is constructed by three elementary rotation matrices corresponding to each Euler angle [14, 41] as below:

$$\begin{aligned}
 {}^cR &= R_{\kappa} \cdot R_{\varphi} \cdot R_{\omega} \\
 &= \begin{bmatrix} C_{\kappa} & -S_{\kappa} & 0 \\ S_{\kappa} & C_{\kappa} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_{\varphi} & 0 & S_{\varphi} \\ 0 & 1 & 0 \\ -S_{\varphi} & 0 & C_{\varphi} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_{\omega} & -S_{\omega} \\ 0 & S_{\omega} & C_{\omega} \end{bmatrix} \\
 &= \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}
 \end{aligned} \tag{2.5}$$

Transformation between frames is illustrated in figure 2.2. According to vector addition rule [14],

$$v_3 = v_2 + v_1, \tag{2.6}$$

where  $v_2 = {}^tP_a$  is landmark  $a$  expressed in the calibration tool frame,  $v_1 = {}^cP_i$  is the origin of the calibration tool frame  $P_o$ , which is located with respect to the camera frame,  $v_3$  is the landmark  $a$  expressed in the camera frame. We want to transform the expression of landmark  $a$  from the calibration tool frame to the camera frame. Note that  $v_2$  is expressed in the camera frame instead to do the addition. Thus, the rotation matrix  ${}^cR$  is needed for:

$${}^c(v_2) = {}^cR \cdot {}^t(v_2). \tag{2.7}$$

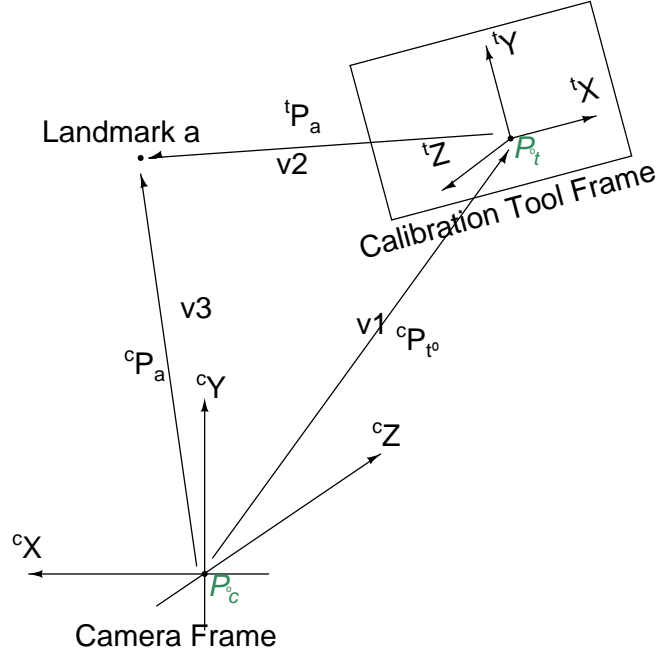


Figure 2.2: Coordinate transformation: from the calibration tool (target) frame to the camera frame.

Finally, from equation (2.6), the equation 2.8 is obtained as:

$${}^cP_a = {}^cR \cdot {}^tP_a + {}^cP_t, \quad (2.8)$$

which is called conformal transformation in [13], for transformation from  ${}^tP_a$  to  ${}^cP_a$  (the destination frame).

### 2.1.2 Camera Geometries

Simply speaking, the camera geometry describes the projection of a landmark  $a$  in 3D space onto the 2D image plane (the projection plane) along a straight line—principal axis [13]. The 3D landmark  $a$  and its corresponding image is a pair of correspondence points. There are three kinds of camera geometries accepted by researchers: single camera

geometry [13, 38], dual camera geometry [13, 38, 43] and multi-station geometry [13, 38, 44].

### Single camera geometry

The camera geometry with only one camera involved is based on central perspective projection [13, 38]. Single camera geometry is the mapping of a landmark  $a$  in 3D space onto the image plane. The mapping from 3D to 2D is along a straight line defined by  ${}^tP_a$  and the principal point of lens  $P_c$ . Single camera geometry is illustrated in figure 2.3.

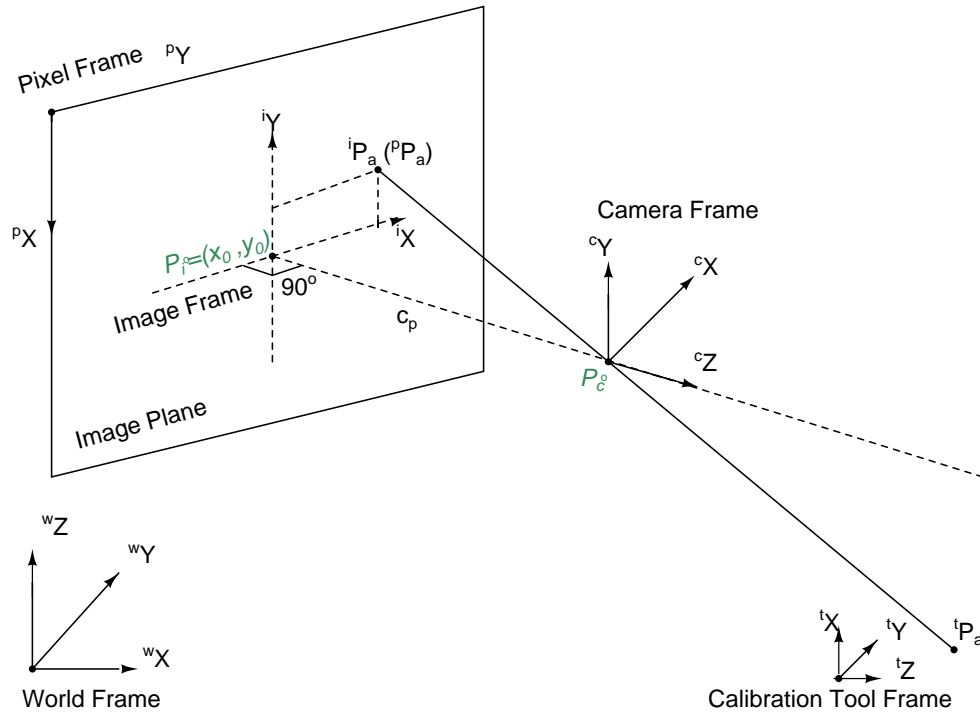


Figure 2.3: Single camera geometry: mapping  ${}^tP_a \in R^3$  onto image plane to get  ${}^iP_a \in R^2$ .

In figure 2.3, there are two principal points. One is the principal point of lens  $P_c$ , which is also the origin of camera coordinate. The other is the principal point of the imager  $P_i = (x_0, y_0)$ , which is also the origin of image coordinate. The two principal points defines a straight line called principal axis, which is perpendicular to the image plane. The  $Z$  axis of the camera frame lies along the principal axis. The distance from  $P_c$  to  $P_i$  is the effective focal length  $c_p$ .

In this geometry, there are world coordinates, target (calibration tool) coordinates, camera coordinates, image coordinates and pixel coordinates [41]. Landmark  $a$  in 3D space in calibration tool coordinates is expressed as

$${}^tP_a = \begin{bmatrix} {}^tX_a \\ {}^tY_a \\ {}^tZ_a \end{bmatrix} \in R^3. \quad (2.9)$$

The landmark  $a$  in world coordinates is

$${}^wP_a = \begin{bmatrix} {}^wX_a \\ {}^wY_a \\ {}^wZ_a \end{bmatrix} \in R^3. \quad (2.10)$$

The landmark  $a$  in camera coordinates is

$${}^cP_a = \begin{bmatrix} {}^cX_a \\ {}^cY_a \\ {}^cZ_a \end{bmatrix} \in R^3. \quad (2.11)$$

The image of landmark  $a$  in image coordinates is

$${}^iP_a = \begin{bmatrix} {}^iX_a \\ {}^iY_a \end{bmatrix} \in R^2. \quad (2.12)$$

The image of landmark  $a$  in pixel coordinates is

$${}^pP_a = \begin{bmatrix} {}^pX_a \\ {}^pY_a \end{bmatrix} \in R^2. \quad (2.13)$$

The starting frame (“From-frame”) is calibration tool coordinates, and the destination frame

(“To-frame”) is image or pixel coordinates. Recall the vector addition for coordinate transformation illustrated by figure 2.2 in section 2.1.1.

To clarify the problem, the variable with a **hat**  $\hat{\cdot}$  is denoted as the value from or relative to estimation based on the model; the variable with a **bar**  $\bar{\cdot}$  is denoted as the value from or relative to measurement based on image processing; the variable without an accent on it is the priori value for initial input of the algorithm. Therefore, starting from  ${}^tP_a$  and camera mapping geometry [13, 38], the transformation from the calibration tool frame ( ${}^tP_a \in R^3$ ) to the camera frame ( ${}^c\hat{P}_a \in R^3$ ) is:

$${}^c\hat{P}_a = {}^cR \cdot {}^tP_a + {}^c\hat{P}_t \quad (2.14)$$

Transformation from the camera frame ( ${}^cP_a \in R^3$ ) to the normalized image frame ( ${}^n\hat{P}_a \in R^3$ ):

$${}^n\hat{P}_a = \frac{{}^c\hat{P}_a}{{}^c\hat{Z}_a} \quad (2.15)$$

where  ${}^n\hat{P}_a = \begin{bmatrix} {}^n\hat{X}_a & {}^n\hat{Y}_a & 1 \end{bmatrix}^T$ . Transformation from the normalized image frame ( ${}^n\hat{P}_a \in R^3$ ) to the image frame ( ${}^iP_a \in R^2$ ):

$${}^i\hat{P}_a = -c_p \cdot {}^n\hat{P}_a(1:2) \quad (2.16)$$

where  $c_p [mm]$  is the effective focal length.

On the other hand, starting from measurement as the pixel frame ( ${}^p\bar{P}_a \in R^2$ ) based on image processing [40, 42], the relationship without counting lens distortion effect<sup>4</sup> is:

$$\begin{bmatrix} {}^i\bar{P}_a \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{k_x} & 0 & -\frac{x_0}{k_x} \\ 0 & \frac{1}{k_y} & -\frac{y_0}{k_y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^p\bar{P}_a \\ 1 \end{bmatrix} \quad (2.17)$$

---

4. See section 2.1.3 for more information about lens distortion.

where matrix

$$\begin{bmatrix} \frac{1}{k_x} & 0 & -\frac{x_0}{k_x} \\ 0 & \frac{1}{k_y} & -\frac{y_0}{k_y} \\ 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

is the camera interior orientation matrix and  ${}^i\bar{P}_a \in R^2$ . In the matrix,  $(x_0, y_0)$  is the principal point of the lens as well as the origin of image plane and image coordinates.  $(k_x, k_y)$  is the pixel density of the imager, which represents the number of pixels in the unit length (1 mm) of the imager in x direction and y direction.

Arrange equation (2.14), the equivalent equation is:

$${}^tP_a = {}^tR \cdot {}^c\hat{P}_a + {}^tP_{\hat{c}}, \quad (2.19)$$

then it becomes

$$\begin{aligned} {}^c\hat{P}_a &= {}^tR^{-1} \cdot ({}^tP_a - {}^tP_{\hat{c}}) \\ &= {}^cR \cdot ({}^tP_a - {}^tP_{\hat{c}}). \end{aligned} \quad (2.20)$$

In matrix format, it is

$$\begin{bmatrix} {}^c\hat{X}_a \\ {}^c\hat{Y}_a \\ {}^c\hat{Z}_a \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} {}^tX_a - {}^tX_{\hat{c}} \\ {}^tY_a - {}^tY_{\hat{c}} \\ {}^tZ_a - {}^tZ_{\hat{c}} \end{bmatrix}. \quad (2.21)$$

After multiplying them, we have

$${}^c\hat{X}_a = R_{11}({}^tX_a - {}^tX_{\hat{c}}) + R_{12}({}^tY_a - {}^tY_{\hat{c}}) + R_{13}({}^tZ_a - {}^tZ_{\hat{c}}) \quad (2.22)$$

$${}^c\hat{Y}_a = R_{21}({}^tX_a - {}^tX_{\hat{c}}) + R_{22}({}^tY_a - {}^tY_{\hat{c}}) + R_{23}({}^tZ_a - {}^tZ_{\hat{c}}) \quad (2.23)$$

and

$${}^c\hat{Z}_a = R_{31}({}^tX_a - {}^tX_{\hat{c}}) + R_{32}({}^tY_a - {}^tY_{\hat{c}}) + R_{33}({}^tZ_a - {}^tZ_{\hat{c}}). \quad (2.24)$$

Then, substituting equation (2.22), (2.23) and (2.24) into equation (2.15) and (2.16), we get

$${}^i\hat{X}_a = -c_p \cdot \frac{R_{11}({}^tX_a - {}^tX_{\hat{c}}) + R_{12}({}^tY_a - {}^tY_{\hat{c}}) + R_{13}({}^tZ_a - {}^tZ_{\hat{c}})}{R_{31}({}^tX_a - {}^tX_{\hat{c}}) + R_{32}({}^tY_a - {}^tY_{\hat{c}}) + R_{33}({}^tZ_a - {}^tZ_{\hat{c}})} \quad (2.25)$$

and

$${}^i\hat{Y}_a = -c_p \cdot \frac{R_{21}({}^tX_a - {}^tX_{\hat{c}}) + R_{22}({}^tY_a - {}^tY_{\hat{c}}) + R_{23}({}^tZ_a - {}^tZ_{\hat{c}})}{R_{31}({}^tX_a - {}^tX_{\hat{c}}) + R_{32}({}^tY_a - {}^tY_{\hat{c}}) + R_{33}({}^tZ_a - {}^tZ_{\hat{c}})} \quad (2.26)$$

Equations (2.25) and (2.26) let us estimate value of the landmark in image coordinates from camera mapping geometry, while equation (2.17) allows us to measure value of the landmark in image coordinates based on image processing technology. The equations (2.25) and (2.26) are called the collinearity equations [13, 38]. Because the equations are derived based on the collinearity of the landmark  ${}^tP_a \in R^3$ , the perspective center (the principal point of lens), and the corresponding image  ${}^iP_a \in R^2$ . Additionally, the rotation matrix  ${}^cR$  or  ${}^tR$  and translation term  ${}^tP_{\hat{c}}$  for transformation from the calibration tool frame ( ${}^tP_a$ ) to the camera frame ( ${}^cP_a$ ) are called the camera **exterior orientation (EO)**. The method of recovering camera EO by single-camera geometry is called resection [13]. However, the disadvantage to resection is that it requires a needs good initial estimation of the Euler angles of rotation for iterative convergence. This makes resection good only when followed by dual-camera intersection or multi-camera bundle adjustment.

### Dual camera geometry

In dual camera geometry [13, 44], there are two cameras that take images of the same landmark, which indicates that a landmark has two images projected onto two image sensors. This procedure is also called intersection, since the coplanarity of each landmark-camera relationship intersects at the landmark, which gives us the coplanarity equations. Dual camera geometry is also called epipolar geometry or stereo camera geometry.

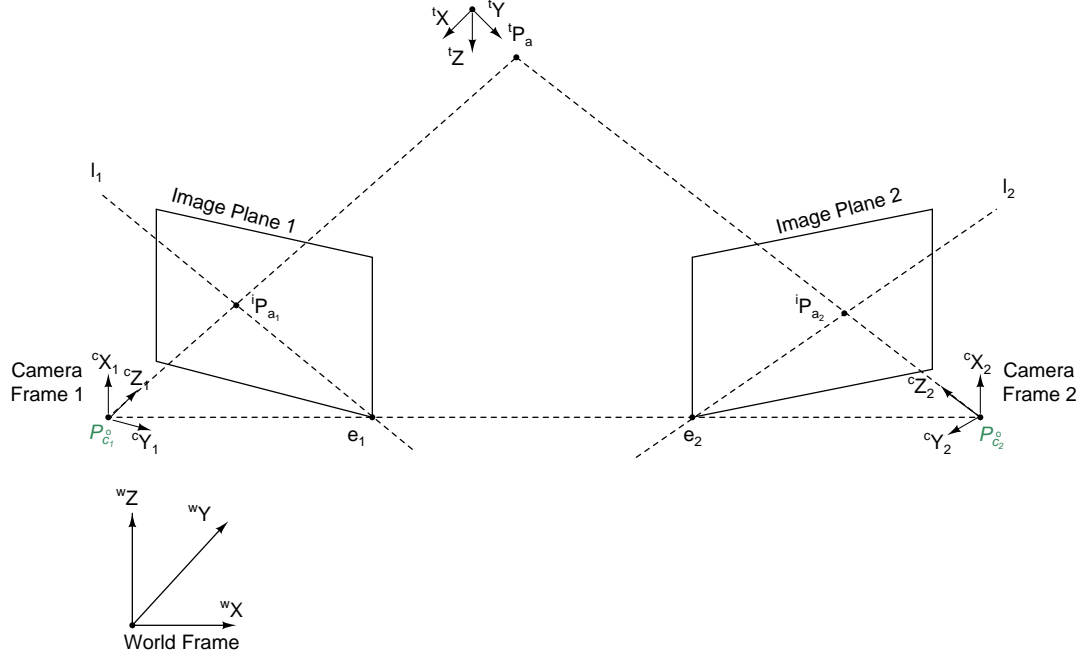


Figure 2.4: Dual-camera geometry: two cameras observe the same landmark.

Dual camera geometry is illustrated in figure 2.4. Origin of each camera coordinates  $P_{c1}$  and  $P_{c2}$  are principal point of camera 1 and camera 2, respectively. Landmark  $a$  is expressed by  ${}^tP_a$  in the calibration tool frame. Images of landmark  ${}^tP_a$  are  ${}^pP_{a1}$  in camera 1 and  ${}^pP_{a2}$  in camera 2. Points  $e_1$  and  $e_2$  are epipoles, which are the corresponding projection of  $P_{c1}$ ,  $P_{c2}$  onto its image plane. Landmark  ${}^tP_a$ ,  $P_{c1}$  and  ${}^pP_{a1}$  are on a straight line. Landmark  ${}^tP_a$ ,  $P_{c2}$  and  ${}^pP_{a2}$  are on a straight line.  $P_{c1}$ ,  $P_{c2}$  and  ${}^tP_a$  define a plane called the epipolar plane. Line  $l_1$  and  $l_2$  are the intersection of epipolar plane with image plane 1 and image plane 2. The coplanarity equations are derived based on the coplanarity of  $P_{c1}$ ,  $P_{c2}$ ,  ${}^tP_a$ ,  ${}^pP_{a1}$  and  ${}^pP_{a2}$ .

The difference between single camera geometry and dual camera geometry is that the latter does not need the given information of  ${}^tP_a$  in order to recover camera EO. The method for recovering camera EO by dual camera geometry is called intersection. One advantage of dual camera geometry is that decentering distortion can be ignored [21]. Since dual camera geometry is not used in bundle adjustment, detailed information will not be covered in this proposal. Cooper and Robson describe epipolar geometry [13]. The book by Ma and his



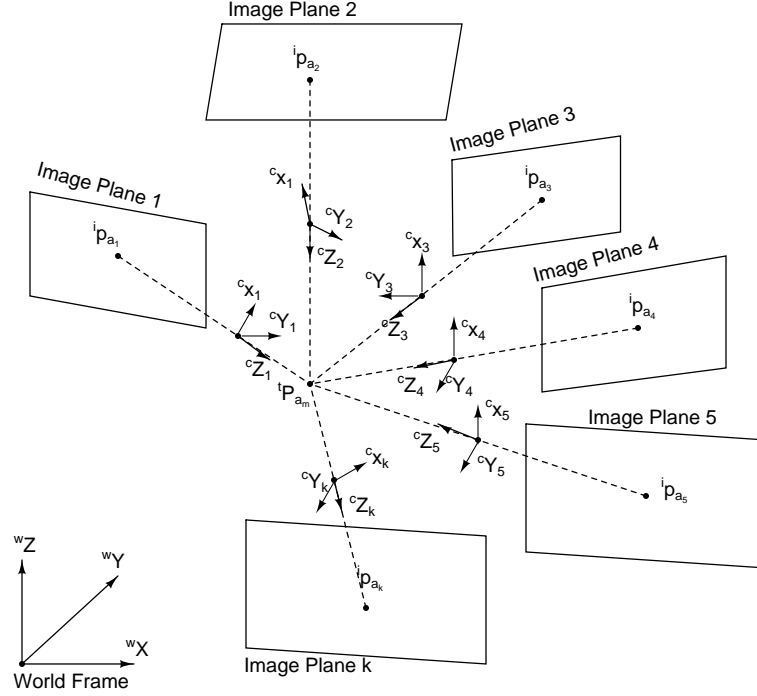


Figure 2.5: Multi-station geometry: combination of  $k = 1, 2, 3, \dots, n_c$  single camera geometries by  $k = 1, 2, 3, \dots, n_c$  camera stations.

colleges [44] is another detailed source.

### Multi-station geometry

Multi-station geometry is based on single camera geometry, which combines several single-camera geometries in 3D space. Thus, each camera station and landmark has two collinearity equations (2.25) and (2.26).

Multi-station geometry is the geometry used in bundle adjustment (BA). Multi-station geometry based on collinearity equations plus optimization method will gain bundle adjustment flexibility and improvement of accuracy [25]. Multi-camera geometry is shown in figure 2.5. There are, in total,  $k = 1, 2, 3, \dots, n_c$  camera positions. Each of them takes an image of the  $m = 1, 2, 3, \dots, n_l$  landmark  $a$ , so there are as many as  $n_c$  images of the  $m^{th}$  landmark  ${}^tP_{a_m} = [{}^tX_{a_m}, {}^tY_{a_m}, {}^tZ_{a_m}]^T$ . Each projection from landmark  $a_1, a_2, \dots, a_m$  to its image is the single camera perspective projection described in single camera geometry [13].

Note that there are two possibilities: 1) the camera is fixed and the calibration tool is moving; 2) the calibration tool is fixed and the camera is moving. With the first setup, we have a set of collinearity equations of the  $m^{th}$  landmarks  $a_m$ ,  $m = 1, 2, 3, \dots, n_l$  on the calibration tool from all possible  $k = 1, 2, 3, \dots, n_c$  camera poses as:

$$\hat{i}X_{a_m}^{(k)} = -c_p \cdot \frac{R_{11}^{(k)}({}^tX_{a_m} - {}^tX_{\hat{c}}^{(k)}) + R_{12}^{(k)}({}^tY_{a_m} - {}^tY_{\hat{c}}^{(k)}) + R_{13}^{(k)}({}^tZ_{a_m} - {}^tZ_{\hat{c}}^{(k)})}{R_{31}^{(k)}({}^tX_{a_m} - {}^tX_{\hat{c}}^{(k)}) + R_{32}^{(k)}({}^tY_{a_m} - {}^tY_{\hat{c}}^{(k)}) + R_{33}^{(k)}({}^tZ_{a_m} - {}^tZ_{\hat{c}}^{(k)})} \quad (2.27)$$

and

$$\hat{i}Y_{a_m}^{(k)} = -c_p \cdot \frac{R_{21}^{(k)}({}^tX_{a_m} - {}^tX_{\hat{c}}^{(k)}) + R_{22}^{(k)}({}^tY_{a_m} - {}^tY_{\hat{c}}^{(k)}) + R_{23}^{(k)}({}^tZ_{a_m} - {}^tZ_{\hat{c}}^{(k)})}{R_{31}^{(k)}({}^tX_{a_m} - {}^tX_{\hat{c}}^{(k)}) + R_{32}^{(k)}({}^tY_{a_m} - {}^tY_{\hat{c}}^{(k)}) + R_{33}^{(k)}({}^tZ_{a_m} - {}^tZ_{\hat{c}}^{(k)})} \quad (2.28)$$

### 2.1.3 Camera Models

#### Pinhole camera model

The pinhole perspective projection for a camera is called the pinhole camera model. There is no lens in the image setup, which makes the model ideal. The problem is that the convergent rays from the object is limited to pass pinhole and focus on the imager. The limitation of insufficient rays passing through the pin hole from the object leads to a very dark image of the object on the imager. Thus, the pinhole model is considered as the ideal camera model [40, 42], but is not physically realizable. However, researchers use the pinhole model to simplify the mathematical relationship of the transformation between  ${}^tP_a$  to  ${}^iP_a$ , which is more straightforward. See figure (2.6) for ideal camera pinhole model, which maps  ${}^tP_a \in R^3$  to  ${}^iP_a \in R^2$  without modeling the distortion of the optical lens.

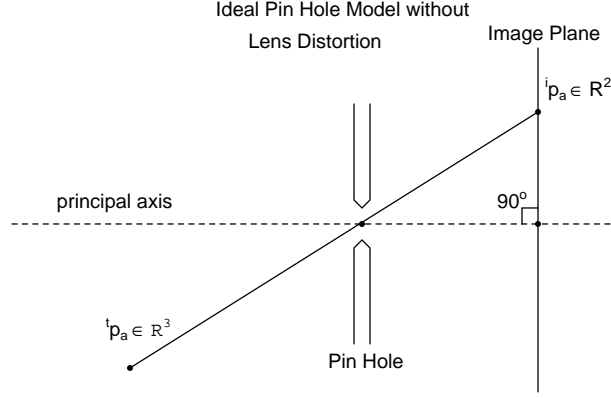


Figure 2.6: The pinhole camera model: mapping  ${}^tP_a \in R^3$  to  ${}^iP_a \in R^2$  without lens distortion.

### Camera model with lens distortion

Later, researchers introduced a lens to be mounted on a camera. While, a lens mounted on the camera will converge diverging rays from an object onto the imager and improve the brightness of the object's image, it will also introduce unavoidable distortion for the object's image due to its physical properties [5, 6, 40, 42]. Thus, it is necessary to introduce lens distortion of the optical lens for camera modeling. In lens distortion of a camera, the variation of magnification in angular of image of a landmark is called radial lens distortion [13, 46]; the displacement of the image of a landmark due to misalignment of the components of the lens is called decentering lens distortion (or tangential lens distortion) [5, 54].

In 1954, Magrill [46] described the relationship between lens distortion and the magnification of the image of the target (calibration tool). He developed a linear equation that can help researchers predict the level of lens distortion. However, he only modeled distortion for an object within sharp focus, but not less focused image of the landmark within field of view of the camera. To improve the modeling of lens distortion for applications, Brown [6] extended Magrill's formula in [46] from

$$\delta r_s = \delta r_{-\infty} - m_s \delta_{-\infty} \quad (2.29)$$

to

$$\delta r_s = \alpha_s \delta r_{s_1} + (1 + \alpha_s) \delta r_{s_2}, \quad (2.30)$$

where  $m_s = \frac{f}{s-f}$ ,  $s$  is the distance of the object plane on which the lens is focused;  $f$  is the lens focal length;  $\delta r_s$ ,  $\delta r_{\pm\infty}$  is the distortion function for focus on the object plane at distance  $s$  or at positive or negative infinity focus;  $a_s = \frac{s_2-s}{s_2-s_1} \frac{s_1-f}{s-f}$ ,  $s_1$  and  $s_2$  are two arbitrary distances of object planes;  $\delta r_{s_1}$  and  $\delta r_{s_2}$  are distortion functions;  $\delta r_{s,s'}$  is the distortion function corresponding to points in an object plane at distance  $s'$  for a lens focused at distance  $s$ .

Brown [5, 6] modeled the radial distortion function by a polynomial<sup>5</sup>:

$$\delta r = K_1 \cdot r^3 + K_2 \cdot r^5 + K_3 \cdot r^7 + \dots, \quad (2.31)$$

where  $r = ||^jP_a|| = \sqrt{{}^jX_a^2 + {}^jY_a^2}$  is the radius of the uncorrected image of the landmark and where  ${}^jP_a$  is introduced as the uncorrected point in image coordinates

$$\begin{bmatrix} {}^j\bar{P}_a \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{k_x} & 0 & -\frac{x_0}{k_x} \\ 0 & \frac{1}{k_y} & -\frac{y_0}{k_y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^p\bar{P}_a \\ 1 \end{bmatrix} \quad (2.32)$$

and  $[K_1, K_2, K_3]$  is the radial distortion coefficient. Then the distortions in x and y directions of radial lens distortion  $\delta {}^jP_{a\_radial}$  are:

$$\delta {}^jP_{a\_radial} = \begin{bmatrix} \delta r_x \\ \delta r_y \end{bmatrix} = \begin{bmatrix} \delta r \frac{{}^jX_a}{r} \\ \delta r \frac{{}^jY_a}{r} \end{bmatrix} = \begin{bmatrix} {}^jX_a(K_1 \cdot r^2 + K_2 \cdot r^4 + K_3 \cdot r^6 + \dots) \\ {}^jY_a(K_1 \cdot r^2 + K_2 \cdot r^4 + K_3 \cdot r^6 + \dots) \end{bmatrix} \quad (2.33)$$

---

5. Generally, three terms  $K_1, K_2, K_3$  in this polynomial are the most that can be determined accurately. If more accuracy is required, more terms can be added as user's need.

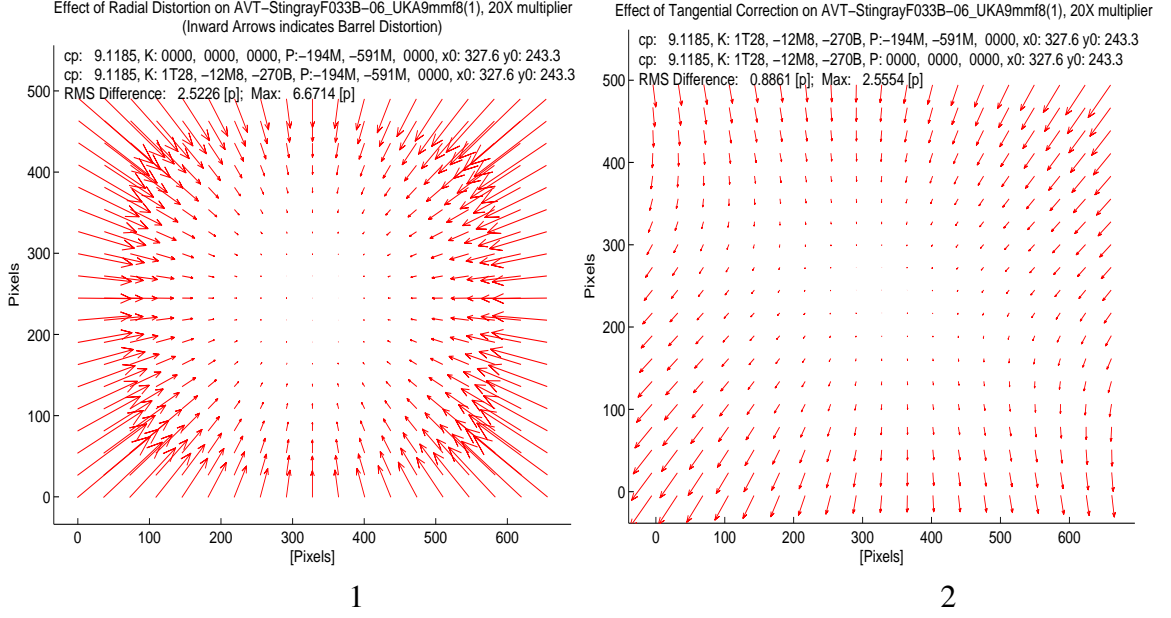


Figure 2.7: Radial distortion (No.1) and Tangential distortion (No.2), distortion effects shown at  $20\times$  magnification.

and modeled decentering distortion function by a polynomial:

$$\delta^j P_{a\_decen} = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix} = \begin{bmatrix} P_1(r^2 + jX_a^2) + 2P_2 jX_a jY_a \\ P_2(r^2 + jY_a^2) + 2P_1 jX_a jY_a \end{bmatrix} \quad (2.34)$$

where  $[P_1, P_2]$  are the lens decenterring distortion coefficients. The radial and tangential distortions of the experimental camera calibration data set collected by AVT-Stingray F033B industrial camera with Edmund 10mm lens are shown in figure 2.7.

The decentering distortion contains both radial and tangential components [5, 65]. Radial distortion changes with focus (depth of view) and field of view. The corresponding coefficient  $K_i$  is highly correlated in its group, but has a loose relationship with EO and other lens distortion coefficients [54].

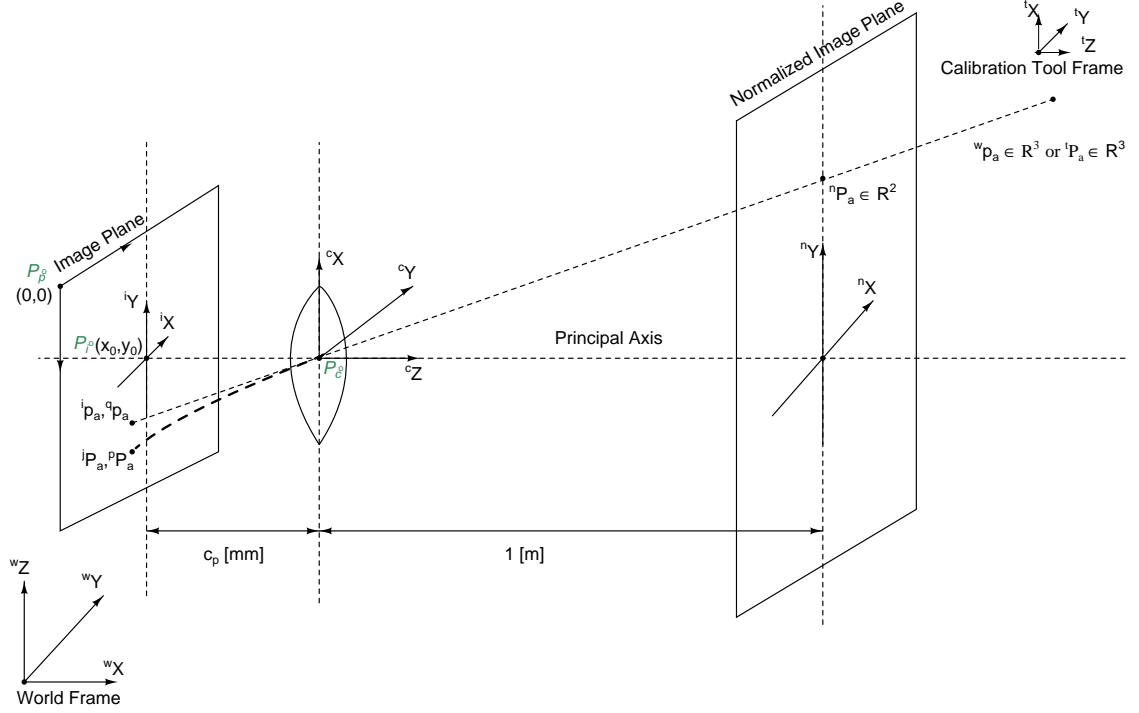


Figure 2.8: Camera model with lens distortion, showing  ${}^iP_a$  (or  ${}^qP_a$ ) and  ${}^jP_a$  (or  ${}^pP_a$ ).

A camera model with lens distortion is shown in figure 2.8: there are  ${}^tP_a$ ,  ${}^nP_a$ ,  ${}^cP_a$ ,  ${}^pP_a$ ,  ${}^jP_a$  and  ${}^iP_a$ . We discussed about  ${}^wP_a$ ,  ${}^tP_a$ ,  ${}^cP_a$ ,  ${}^iP_a$  and  ${}^pP_a$  in single camera geometry in section 2.1.2. The original light without lens distortion arrives at the point  ${}^iP_a$  in the image frame or  ${}^qP_a$  in the pixel frame. However, the lens distortion disturbs the straight path when lights from  ${}^tP_a$  go through the lens, physically. The physical position of image of  ${}^tP_a$  on image plane is  ${}^jP_a$  in the image frame or  ${}^pP_a$  in the pixel frame by following the curve in bold dashed line. The relationship between ideal image, distorted image and corrected image is illustrated in table 2.1.

	Physical Geometry with Lens Distortion	
	Coordinates in [pixels]	Coordinates in [mm]
Points on the image plane physically (Distorted)	${}^p\bar{P}_a$	${}^j\bar{P}_a$
	Geometry with Lens Distortion Correction	
	Coordinates in [pixels]	Coordinates in [mm]
Points data after lens distortion correction (Corrected)	${}^q\hat{P}_a$	${}^i\bar{P}_a$ or ${}^i\hat{P}_a$

Table 2.1: Coordinates on image plane affected by lens distortion.

In the table,  ${}^p\bar{P}_a$  and  ${}^j\bar{P}_a$  are from image metrology of digital images as observation, which includes lens distortion. Variable  ${}^i\bar{P}_a$  is data derived from image metrology with lens distortion correction. The value of  ${}^i\bar{P}_a$  is not ideal for landmark in pixel coordinates, but an estimation of ideal value can be made by correcting lens distortion. Therefore, to follow camera model with lens distortion and correction, it is necessary to rewrite equation (2.17), which did not account for the lens distortion effect, as equation (2.35) to transform distorted  ${}^p\bar{P}_a$  to distorted  ${}^j\bar{P}_a$ :

$$\begin{bmatrix} {}^j\bar{P}_a \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{k_x} & 0 & -\frac{x_0}{k_x} \\ 0 & \frac{1}{k_y} & -\frac{y_0}{k_y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^p\bar{P}_a \\ 1 \end{bmatrix}. \quad (2.35)$$

Next, compensation of the lens distortion effect gives the corrected landmark in pixel coordinates  ${}^i\bar{P}_a$  from distorted  ${}^j\bar{P}_a$ :

$${}^i\bar{P}_a = {}^j\bar{P}_a + \delta {}^jP_{a\_radial} + \delta {}^jP_{a\_decen}, \quad (2.36)$$

which can be expressed as

$${}^i\bar{P}_a = L_c({}^j\bar{P}_a) \quad (2.37)$$

where  $L_c$  denotes the forward correction model of lens distortion. **The above relationship**

**starts from the landmark in pixel coordinates , the observation, to the landmark in image coordinates.**

On the other hand, starting from  ${}^tP_a$  and following calculation through camera geometry by equations (2.14), (2.15), (2.16), we have the estimated landmark in pixel coordinates  $i\hat{P}_a$ . The  $q\hat{P}_a$  is a concept variable corresponding to  $i\hat{P}_a$ . A similar relationship compared to equation (2.35) is

$$\begin{bmatrix} i\hat{P}_a \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{k_x} & 0 & -\frac{x_0}{k_x} \\ 0 & \frac{1}{k_y} & -\frac{y_0}{k_y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q\hat{P}_a \\ 1 \end{bmatrix} \quad (2.38)$$

Note that **to transform the landmark in image coordinates back to the landmark in pixel coordinates requires an iterative solution.** The reason is the inverse lens distortion model  $L_c^{-1}$  does not exist in closed-form. This forces and allows us to apply the Newton-like method to do optimal camera calibration with lens distortion. More information about the Newton-like method for optimization is in section 2.3.1.

The coefficients for lens distortion modeling are called Additional Parameters (AP). Other distortion parameters, such as affinity distortion, shear distortion, linear distortion, prism distortion, out of plane distortion, etc, were also introduced by researchers [21, 39]. The reasons can be light sensitive rectangular elements of digital camera, non-orthogonal image axes, issues in lens design and assembly, focal plane unflatness [40, 54]. The full AP set consists of more than 20 parameters but only 10 thereof are used for digital cameras [54]. The items in AP for lens distortion compensation is freely selected by users according to their applications. But AP other than  $K_i, i = 1, 2, 3, \dots$  and  $P_i, i = 1, 2, \dots$  are either quite negligible or are covered by radial and decentering distortion. Lens distortions are not independent of each other, because they all from the lens itself. For example, the modeling of barrel distortion and prism distortion can be covered by radial distortion modeling [65]. The most widely accepted modern camera model in photogrammetry are effective focal length  $c_p$ , principal distance;  $(x_0, y_0)$ , principal point of imager;  $[K_1, K_2, K_3]$ , the radial



distortion coefficient; and  $[P_1, P_2]$ , the decentering distortion coefficient [6, 54].

We refer to  $c_p$ ,  $(k_x, k_y)$ , and  $(x_0, y_0)$  mentioned in equations (2.14), (2.15), (2.16), and (2.17), plus  $[K_1, K_2, K_3]$  and  $[P_1, P_2]$ , the lens distortion coefficients of the lens, as the camera **interior orientation (IO)**.

### Camera model function of bundle adjustment

As mentioned in section 2.1, multi-station geometry is used in bundle adjustment. The modeling function for bundle adjustment is written in below.

If we write collinearity equations (2.27) and (2.28) of multi-camera geometry again, which are for the  $m^{th}$  landmarks  $a_m, m = 1, 2, 3, \dots, n_l$  on the calibration tool from all possible  $k = 1, 2, 3, \dots, n_c$  camera poses:

$$\begin{cases} \hat{X}_{a_m}^{(k)} = -c_p \cdot \frac{R_{11}^{(k)}(tX_{a_m} - tX_{\hat{c}}^{(k)}) + R_{12}^{(k)}(tY_{a_m} - tY_{\hat{c}}^{(k)}) + R_{13}^{(k)}(tZ_{a_m} - tZ_{\hat{c}}^{(k)})}{R_{31}^{(k)}(tX_{a_m} - tX_{\hat{c}}^{(k)}) + R_{32}^{(k)}(tY_{a_m} - tY_{\hat{c}}^{(k)}) + R_{33}^{(k)}(tZ_{a_m} - tZ_{\hat{c}}^{(k)})} \\ \hat{Y}_{a_m}^{(k)} = -c_p \cdot \frac{R_{21}^{(k)}(tX_{a_m} - tX_{\hat{c}}^{(k)}) + R_{22}^{(k)}(tY_{a_m} - tY_{\hat{c}}^{(k)}) + R_{23}^{(k)}(tZ_{a_m} - tZ_{\hat{c}}^{(k)})}{R_{31}^{(k)}(tX_{a_m} - tX_{\hat{c}}^{(k)}) + R_{32}^{(k)}(tY_{a_m} - tY_{\hat{c}}^{(k)}) + R_{33}^{(k)}(tZ_{a_m} - tZ_{\hat{c}}^{(k)})} \end{cases} \quad (2.39)$$

The camera interior orientation and lens distortion correction coefficient are introduced for each camera pose  $k = 1, 2, 3, \dots, n_c$ . Note that the camera is fixed and the camera calibration tool is moving during data collection phase, thus there is only one set of camera IO but  $k$  sets of camera EO. Then, the modified equation of equation (2.39) is obtained as

follows:

$$\left\{ \begin{array}{l} jX_{a_m}^{(k)} + \frac{jX_{a_m}^{(k)}}{r_{a_m}^{(k)}} (K_1 \cdot (r_{a_m}^{(k)})^3 + K_2 \cdot (r_{a_m}^{(k)})^5 + K_1 \cdot (r_{a_m}^{(k)})^7) + P_1((r_{a_m}^{(k)})^2 + (jX_{a_m}^{(k)})^2) + 2P_2 \cdot jX_{a_m}^{(k)} jY_{a_m}^{(k)} \\ \dots = -c_p \cdot \frac{R_{11}^{(k)}(tX_{a_m} - tX_{\hat{c}}^{(k)}) + R_{12}^{(k)}(tY_{a_m} - tY_{\hat{c}}^{(k)}) + R_{13}^{(k)}(tZ_{a_m} - tZ_{\hat{c}}^{(k)})}{R_{31}^{(k)}(tX_{a_m} - tX_{\hat{c}}^{(k)}) + R_{32}^{(k)}(tY_{a_m} - tY_{\hat{c}}^{(k)}) + R_{33}^{(k)}(tZ_{a_m} - tZ_{\hat{c}}^{(k)})} \\ \\ jY_{a_m}^{(k)} + \frac{jY_{a_m}^{(k)}}{r_{a_m}^{(k)}} (K_1 \cdot (r_{a_m}^{(k)})^3 + K_2 \cdot (r_{a_m}^{(k)})^5 + K_1 \cdot (r_{a_m}^{(k)})^7) + P_2((r_{a_m}^{(k)})^2 + (jY_{a_m}^{(k)})^2) + 2P_1 \cdot jX_{a_m}^{(k)} jY_{a_m}^{(k)} \\ \dots = -c_p \cdot \frac{R_{21}^{(k)}(tX_{a_m} - tX_{\hat{c}}^{(k)}) + R_{22}^{(k)}(tY_{a_m} - tY_{\hat{c}}^{(k)}) + R_{23}^{(k)}(tZ_{a_m} - tZ_{\hat{c}}^{(k)})}{R_{31}^{(k)}(tX_{a_m} - tX_{\hat{c}}^{(k)}) + R_{32}^{(k)}(tY_{a_m} - tY_{\hat{c}}^{(k)}) + R_{33}^{(k)}(tZ_{a_m} - tZ_{\hat{c}}^{(k)})} \end{array} \right. \quad (2.40)$$

where  $r_{a_m}^{(k)} = ||jP_{a_m}^{(k)}|| = \sqrt{(jX_{a_m}^{(k)})^2 + (jY_{a_m}^{(k)})^2}$ ;  $m$  is for the  $m^{th}$  landmarks  $a_m, m = 1, 2, 3, \dots, n_l$  on the calibration tool; and  $k = 1, 2, 3, \dots, n_c$  is for the  $k^{th}$  camera station which captured the digital images of the landmarks on the calibration tool.

If the inverse of equation (2.35)

$$\begin{bmatrix} jX_a \\ jY_a \\ 1 \end{bmatrix} = \begin{bmatrix} jP_a \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{k_x} & 0 & -\frac{x_0}{k_x} \\ 0 & \frac{1}{k_y} & -\frac{y_0}{k_y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} pP_a \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{k_x}(pX_a - x_0) \\ \frac{1}{k_y}(pY_a - y_0) \\ 1 \end{bmatrix} \quad (2.41)$$

is substituted in to equation (2.40), we will have

$$\left\{ \begin{array}{l} \frac{1}{k_x}(pX_{a_m}^{(k)} - x_0) + \frac{1}{k_x}(pX_{a_m}^{(k)} - x_0) \cdot (r_{a_m}^{(k)})^{-1} \cdot (K_1 \cdot (r_{a_m}^{(k)})^3 + K_2 \cdot (r_{a_m}^{(k)})^5 + K_1 \cdot (r_{a_m}^{(k)})^7) + \dots \\ P_1((r_{a_m}^{(k)})^2 + (jX_{a_m}^{(k)})^2) + 2P_2 \cdot jX_{a_m}^{(k)} jY_{a_m}^{(k)} = -c_p \cdot \frac{R_{11}^{(k)}(tX_{a_m} - tX_{\hat{c}}^{(k)}) + R_{12}^{(k)}(tY_{a_m} - tY_{\hat{c}}^{(k)}) + R_{13}^{(k)}(tZ_{a_m} - tZ_{\hat{c}}^{(k)})}{R_{31}^{(k)}(tX_{a_m} - tX_{\hat{c}}^{(k)}) + R_{32}^{(k)}(tY_{a_m} - tY_{\hat{c}}^{(k)}) + R_{33}^{(k)}(tZ_{a_m} - tZ_{\hat{c}}^{(k)})} \\ \\ \frac{1}{k_y}(pY_{a_m}^{(k)} - y_0) + \frac{1}{k_y}(pY_{a_m}^{(k)} - y_0) \cdot (r_{a_m}^{(k)})^{-1} \cdot (K_1 \cdot (r_{a_m}^{(k)})^3 + K_2 \cdot (r_{a_m}^{(k)})^5 + K_1 \cdot (r_{a_m}^{(k)})^7) + \dots \\ P_2((r_{a_m}^{(k)})^2 + (jY_{a_m}^{(k)})^2) + 2P_1 \cdot jX_{a_m}^{(k)} jY_{a_m}^{(k)} = -c_p \cdot \frac{R_{21}^{(k)}(tX_{a_m} - tX_{\hat{c}}^{(k)}) + R_{22}^{(k)}(tY_{a_m} - tY_{\hat{c}}^{(k)}) + R_{23}^{(k)}(tZ_{a_m} - tZ_{\hat{c}}^{(k)})}{R_{31}^{(k)}(tX_{a_m} - tX_{\hat{c}}^{(k)}) + R_{32}^{(k)}(tY_{a_m} - tY_{\hat{c}}^{(k)}) + R_{33}^{(k)}(tZ_{a_m} - tZ_{\hat{c}}^{(k)})} \end{array} \right. \quad (2.42)$$

The equation (2.42) can be written as

$$y = f(\hat{\mathbf{u}}, x, c), \quad (2.43)$$

where  $y$  is the observed data<sup>6</sup> or estimated data<sup>7</sup> as the dependent variable;  $x$  is the vector of independent variables;  $c$  is a vector of constants; and  $\hat{\mathbf{u}}$  is the IO, EO and lens distortion coefficient parameters to be estimated<sup>8</sup>. Equation (2.42) is the explicit model for camera calibration by bundle adjustment, while equation (2.43) is the implicit model.

## 2.2 Camera Calibration

### 2.2.1 Definition of Camera Calibration

**Camera calibration** is the process of recovering the relationship between a landmark  $a$  and its image based on camera geometry in 3D space by estimating camera interior orientation parameters, lens distortion correction coefficients, and camera exterior orientation parameters.

**The interior orientation of a camera is:**

1. Principal distance  $c_p$ : the distance from the perspective center to image plane.  
 $c_p = \frac{s \cdot f}{s - f}$ , where  $f$  is the focal length of the lens as a fixed number for a certain lens and  $s$  is object distance, which is the distance from object to lens.  $c_p$  is also called adjusted focal length or effective focal length.
2. Principal point of the lens  ${}^pP_i = ({}^pX_i, {}^pY_i)$  ( or  $(x_0, y_0)$ , or  $(x_p, y_p)$ ): the origin of the image plane and image coordinates. It is also called the offset of the imager.

---

6. Observed data is the  ${}^p\bar{P}_a$  based on image processing of digital image of calibration tool.

7. Estimated data is the  ${}^p\hat{P}_a$  based on the camera model equations.

8. In linear regression point of view,  $u$  is  $\beta$ .

3. Pixel density of the imager ( $k_x, k_y$ ): the pixel density of the imager in x direction and y direction, which is the number of pixels in the unit length (1 mm).

**The lens distortion parameters are:**

1. Radial distortion coefficient [ $K_1, K_2, K_3$ ]: the coefficient for lens distortion modeling polynomial.  $K_i$  is highly correlated and varies with focal length.  $K_1$  also modeled the barrel distortion.
2. Decentering distortion coefficient [ $P_1, P_2$ ]: the coefficient for lens distortion modeling polynomial. Due to a lack of centering of lens elements along the optical axis,  $P_i$  varies with focal length, but to a lower extent than  $K_i$ ; but both  $K_i$  and  $P_i$  will be worse for long focal length.

The camera calibration interior model is denoted by  $\chi$ . So, the  $\chi_k$  of the  $k^{th}$  camera is  $\chi_k = \begin{bmatrix} c_p & k_x & k_y & x_0 & y_0 & K_1 & K_2 & K_3 & P_1 & P_2 & P_3 \end{bmatrix}^T$ , and the units of each elements of  $\chi_i$  in order is  $mm, \frac{pixel}{mm}, \frac{pixel}{mm}, pixel, pixel, mm^{-2}, mm^{-4}, mm^{-6}, mm^{-1}, mm^{-1}$  and  $mm^{-2}$ .

**The exterior orientation of camera is:**

1. Rotation matrix  ${}^cR$  or  ${}^tR$  and translation term  ${}^tP_c$  for transformation from the calibration tool frame ( ${}^tP_a \in R^3$ ) to the camera frame ( ${}^cP_a \in R^3$ ).

## 2.2.2 Camera Calibration Methods

Camera calibration methods can be divided into two types: conventional camera calibration and self-calibration (auto-calibration). **Conventional camera calibration** needs a test range, which may contain hundreds of landmarks. The landmarks are on an object with known Euclidean structure. These coordinated landmarks are measured and determined by images from a metric camera. Camera calibration should always be

conducted using a test range [22]. The maintenance of the test range is relatively high, and the size of target can be very large. **Self-calibration (auto-calibration)** is the development of on-the-job calibration<sup>9</sup>. On-the-job calibration does not require taking the camera to the test range for calibration. Accurate coordinated landmarks are not needed as priori knowledge. A minimum of seven landmarks are required. It is more typical to use 50-100 landmarks on the calibration tool. An expensive metric camera is not needed for maintenance of the test range; a non-metric camera is enough, but it still needs a calibration tool with landmarks. The key difference between traditional calibration and self-calibration is the former needs Euclidean information of landmarks, which is the used constraints, while self-calibration does need an object but does not use Euclidean information. Instead, the IO constraints of the camera are used in self-calibration.

Camera calibration methods can also be classified by parameter estimation and optimization algorithm other than whether the camera is calibrated “automatically”, which are linear camera calibration methods and non-linear camera calibration methods. Originally, camera calibration algorithm was developed in the field of photogrammetry area first, but later it caught the attention from computer vision community, which processes digital images collected by camera.

### 2.2.3 Camera Calibration Methods: Linear Algorithm

Abdel-Aziz and Karara [1] developed a method called **direct linear transformation (DLT)**. DLT is designed for low accuracy and reduction of data in close range photogrammetry. After arranging equation(2.14), (2.15), and (2.16) , they gave us

$$\begin{cases} iX_a + \frac{l_1 \cdot {}^tX_a + l_2 \cdot {}^tY_a + l_3 \cdot {}^tZ_a + l_4}{l_9 \cdot {}^tX_a + l_{10} \cdot {}^tY_a + l_{11} \cdot {}^tZ_a + 1} = 0 \\ iY_a + \frac{l_5 \cdot {}^tX_a + l_6 \cdot {}^tY_a + l_7 \cdot {}^tZ_a + l_8}{l_9 \cdot {}^tX_a + l_{10} \cdot {}^tY_a + l_{11} \cdot {}^tZ_a + 1} = 0 \end{cases} \quad (2.44)$$

---

9. Sometimes, on-the-job calibration is considered as self-camera calibration

where  $l_1, l_2, \dots, l_{11}$  are parameters of coordinate transformation and are independent. Here, we do not use an iterative method for the 11 parameters, but continue to arrange the equations to get

$$\begin{cases} {}^iX_a(l_9 \cdot {}^tX_a + l_{10} \cdot {}^tY_a + l_{11} \cdot {}^tZ_a + 1) - l_1 \cdot {}^tX_a - l_2 \cdot {}^tY_a - l_3 \cdot {}^tZ_a - l_4 = 0 \\ {}^iY_a(l_9 \cdot {}^tX_a + l_{10} \cdot {}^tY_a + l_{11} \cdot {}^tZ_a + 1) - l_5 \cdot {}^tX_a - l_6 \cdot {}^tY_a - l_7 \cdot {}^tZ_a - l_8 = 0 \end{cases} \quad (2.45)$$

The equations above show us that DLT approximates the non-linear estimation by linear estimation under distortion-free assumption for conversion between  ${}^tP_a \in R^3$  and  ${}^iP_a \in R^2$ . This method is simple and easy to apply but not accurate, since it is applied by the linearization of a non-linear model. DLT is quite sensitive to observed data noise. Later, Marzan and Karara [48] improved the rigorousness of DLT. DLT was modified by Faugeras and Toscani [17] and Melen [50]. Chen and his colleges [9] evaluated the accuracy of the DLT method experimentally, suggesting that it is good to distribute control points evenly on the target for accuracy improvement.

Caprile and Torre [8] introduced a camera calibration method using **vanishing points**. This method focused on the application of computer vision by two steps. First,  $c_p$  and  $(x_0, y_0)$  of camera interior orientation was calculated based on a single image of a cube. Next, camera exterior orientation was calculated based on dual-camera geometry by matching the corresponding vanishing points in images of  ${}^tP_a$  and triangulation constraint. No lens distortion effect was considered.

All in all, the absence of lens distortion makes linear camera calibration more popular in research areas that do not require the rigorous output of camera calibration, such as kinematics analysis [10, 15, 32]. The closed-form solution by linear camera calibration is utilized to yield initial estimated data for a non-linear iterative algorithm in many two-step hybrid camera calibration algorithms described below.

### 2.2.4 Camera Calibration Methods: Non-linear and Hybrid Algorithm

Since the linear method is only good for situations with low accuracy requirements and does not count lens distortion effect in the modeling, techniques in this group must contain lens distortion and use a non-linear iterative method to refine calibration results. Unlike the linear method, there is no closed-form solution for the non-linear method; the non-linear method uses the iteration strategy to finish the task. Least squares estimation is widely used to minimize cost function:

$$\sum r(x)^2 \quad (2.46)$$

where  $r(x) = \hat{y} - \bar{y} = \hat{P}_a - \bar{P}_a$  is the residual function of the difference of estimation  $\hat{y}$  from equations (2.14), (2.15), (2.16) and observation  $\bar{y}$  from equation (2.17).

Researchers used to use the non-linear optimization method by guessing the initial value of parameters. However, the property of the Newton-like iterative method with the least squares estimator requires the initial value to be within the basin of attraction for successful and correct convergence. In other words, only a good initial guess can let the algorithm converge to a correct result; otherwise, it will converge extremely slowly, fail to converge or converge to a false result. Therefore, the hybrid method was developed. “hybird” refers to the algorithm containing the linear method as the first phase and the non-linear method as the second phase by using the output of linear phase for the initial guess. This way, the convergence of algorithm is smoother and faster. Moreover, there is a unique method that estimates lens distortion parameters only. Brown [6] introduced the plumb-line method for calibrating camera with a test field, which can extract radial and decentering distortion only but not for principal distance  $c_p$ , the principal point/offset  $(x_p, y_p)$  and EO of camera. Most methods will estimate camera interior orientation together.

Several hybrid methods were developed. Tsai [60] presented a hybrid method that is considered a classical camera calibration method in computer vision. He studied the constraint called the radial alignment constraint, which is under single camera perspective

geometry. He assumed some parameters such as the principal point; the pixel density in  $x, y$  directions is given. The system inputs were those offered parameters,  $\bar{y} = {}^pP_a$  measured from digital image and  ${}^tP_a$  measured by microscope. Tsai considered these data to be accurate and did not estimate them further. By the linear closed-form equations<sup>10</sup> mentioned above, he calculated camera EO parameters (rotation matrix  $R$  and translation  $T_x, T_y$ ) as the final value. After solving the initial value of  $f$  and  $T_z$ , he used LSE to estimate the only counted lens distortion parameter  $K_1$  with improved  $f$  and  $T_z$ . Tsai used at least eight points of correspondence to finish calculation<sup>11</sup>. Later, Tsai and Lenz [37] counted the principal point of the image plane as estimated parameters based on their previous theory [60]. The method is not self-calibration, since  ${}^tP_a$  is measured by telescope as priori knowledge.

The eight-point algorithm [28, 29, 30, 31] is considered the classical representation of the direct relative orientation methods, which use a fundamental matrix and an essential matrix based on dual camera geometry<sup>12</sup>. Later there are the 7-point algorithm, the 6-point algorithm and the 5-point relative orientation algorithm [52, 57]. Hartley [28, 29, 31] only recovered camera exterior orientation (3D reconstruction). In [31], he researched on calibrating camera by so-called eight control points algorithm. He discussed the ambiguity of Euclidean reconstruction from an essential matrix, which maps correspondence from  ${}^tP_a$  to  ${}^iP_a$ . Later, he [28, 29] used projective reconstruction to find landmark-camera ( ${}^iP_{a_j} \cdot M_i = {}^tP_{a_j}$ ) correspondence projection, quasi-affine transformation to find camera interior parameter matrix  $K_i$ <sup>13</sup> and  $v$ , and a Euclidean reconstruction to find the rotation matrix  $R_i$  of camera EO by solving an 8 parameters problem. After collecting values of  $K_i, R_i, {}^tP_{a_j}$ , Hartley used the Levenberg-Marquart algorithm to refine all of those values. The first reconstruction is a linear closed-form solution and is performed to obtain the initial value for later non-linear refinement. His technique can also accept multi-views

10. The relationship of  $\bar{y}$  and  ${}^tP_a$  based on single camera geometry.

11. This is not our current technique, where the inputs are  ${}^pP_a$  and outputs are  $\hat{u}$ .

12. See dual camera geometry subsection in 2.1.3

13. The inverse of the matrix in equation (2.17).



and improve estimation quality. In [30], he further discussed the performance of this classic eight-point algorithm. Note that there are no any lens distortion parameters in this mathematical model. Thus, the disadvantage is that the non-linear lens distortion parameters in the camera model are not easily estimated under the linear method. However, this kind of method can be used to offer an initial guess of the rotation matrix and translation from  ${}^iP_a$  to  ${}^cP_a$  for the non-linear method.

Weng and his colleges [65] also offered a two-step calibration procedure for a stereo camera setup. The first calculation yields camera EO and some camera IO based on a distortion-free (pinhole) model and then improves all the parameters under a camera model with lens distortion. The differences between Weng's method and Tsai's method are: a) Weng's linear step gives camera EO and IO, but Tsai only computes part of them; and b) Weng's non-linear step can solve for all the parameters of IO and EO, but Tsai's only solves  $K_1$  and improves  $f$  and  $T_z$ . Weng and his colleges analyzed the influence of the involvement of types of additional parameters, represented by polar coordinates, in the camera model. They introduced a way to evaluate quality of camera calibration by image resolution-normalized stereo camera error. Like Tsai's method above, they treated a priori knowledge of  ${}^iP_a$  as the true value for the input of their algorithm calculation.

Heikkila and Silven [34] introduced a hybrid method for camera calibration. In the first phase, they used the DLT method introduced by Abdel-Aziz and Karara [1] to offer the initial estimation for the camera model. In the second phase, the Levenberg-Marquart (LM) algorithm was employed as the non-linear iterative estimation method to compute lens distortion parameters and optimize IO and EO from DLT. In their article, the method was described as a four-step calibration procedure. The first step was to apply the DLT method; the second step uses the LM algorithm; the third step corrects distortion of circular features; and the fourth step is to correct the distorted image coordinates  ${}^jP_a$  to  ${}^iP_a$  by the an implicit correction model for lens distortion parameters. They reached the accuracy level that "residual error less than 0.01 pixel unit" in statistics. In [33], Heikkila analyzed

lens distortion modeling and handling in more detail than the analysis in [34]. In their theory, a matrix of circular landmarks is used as the target and developed corresponding a perspective projection geometry of the circular landmarks. He demonstrated his claim by analyzing synthetic images for the forward distortion model and reverse distortion model. He also mentioned other error sources including insufficient projection models, illumination changes, camera electronics, and calibration tools.

Zhang [68] focused on a low-cost, flexible camera calibration for a desktop vision system. He did not measure  ${}^tP_a$  with extra instruments as in Tsai's method [60] or Weng's algorithm [65]. This was the flexible part, since only a planar calibration tool printed by laser printer is enough for the application. His technique needed at least three views of a planar object in different orientations. He used the homography relationship of projective correspondence of  ${}^tP_a \in R^3$  and its image  ${}^iP_a \in R^2$ . A closed-form solution is offered for a subset of camera IO and the full set of camera EO at the first step. Then, a non-linear estimation—the Levenberg-Marquardt algorithm is utilized to find the radial distortion  $K_i$  of lens distortion modeling. Note that he did not recover decentering distortion coefficients  $P_i$ .

Wang and his colleges [63, 64] developed a new model of lens distortion. According to their claim, the model has fewer parameters to be calibrated and more explicit physical meaning than the classical model [6]. He covered three types of distortion—radial distortion, decentering distortion, and thin prism distortion—and the coupling relationship among each of these three types. They express tangential distortion with a transformation consisting of rotation and translation. The calculation procedure is from  $P_c({}^cP_a)$  to  $P_i({}^iP_a)$  to  $P_d({}^jP_a)$  to  $\begin{bmatrix} x_{di} & y_{di} & z_{di} \end{bmatrix}^T$  to  $P_r({}^jP_a) = \begin{bmatrix} U & V \end{bmatrix}^T$  to  ${}^pP_a = \begin{bmatrix} u & v \end{bmatrix}^T$ . Based on experimental analysis, they claimed that the new model reduces calculation time consumption by 20%<sup>14</sup> and that the longer focal length of the lens would reduce the lens distortion effect. They also claimed that less differences in the performance of the lens distortion model.

14. In their estimation, they only estimates  $K_1$  and  $K_2$ , but tangential distortion coefficient  $P_1$  and  $P_2$  are covered in the conversion form  $\begin{bmatrix} x_{di} & y_{di} & z_{di} \end{bmatrix}^T \rightarrow P_r({}^jP_a)$  by equation (15).

Rahman and Krouglic [53] describe a camera calibration method that minimizes the error between the theoretical image points from camera model and those points determined experimentally in distortion-free space. The so-called distortion-free method is based on the pinhole model or after lens distortion correction. As they claim, the new method offers improved accuracy, robustness, and efficiency over a wide range lens distortion. They use quaternion representation for spatial orientation. In the algorithm, they use the Gauss-Newton iterative method to minimize the difference of  ${}^iP_a^*$  and  ${}^iP_a$ , where  ${}^iP_a^*$  is the undistorted<sup>15</sup> landmark coordinates in the image frame and is obtained by applying perspective projection under the pinhole model (just from calculation),  ${}^iP_a$  is the landmark coordinates in the image frame after lens distortion correction<sup>16</sup>. Ideal data  ${}^iP_a^*$  can only be estimated and can not be collected by anyone under any methods. Finally, they test the influence of higher order terms in the lens distortion model on estimation.

None of the above methods developed by researchers involve bundle adjustment (BA), even though their methods include non-linear estimation, because camera IO and EO are not estimated as a bundle or only part of camera IO parameters are recovered. Moreover, during the data collection stage, the bundle adjustment (BA) considers the minimum number of parameter values to be fixed and estimates everything other than the three landmarks as 7 constraints by minimizing the objective function in Newton-like numerical method.

### 2.3 Camera Calibration: Bundle Adjustment (BA)

Currently, bundle adjustment (BA) is a standard method in the photogrammetry community<sup>17</sup>. From the camera calibration point of view, bundle adjustment estimates both the IO and EO of a camera simultaneously. From the computation point of view, bundle adjustment is a large scale parameter estimation problem involving 1000 or more

15. Ideal situation without lens distortion in the model.

16.  ${}^iP_a = {}^iP_a + \text{distortion correction term}$ , where  ${}^iP_a$  is the distorted image coordinates—observed data.

17. The algorithms in section 2.2.3, 2.2.4, 2.2.4 are not bundle adjustment for camera calibration in close-range photogrammetry, since the full set parameters of the lens distortion in IO is not re covered on purpose and IO and EO are not estimated simultaneously.

parameters for estimation. From the numerical method point of view, bundle adjustment uses a modified Gauss-Newton algorithm to finish the optimal estimation, which is a optimal estimation method. From experimental point of view, bundle adjustment in close-range photogrammetry utilizes a high-level instrument, such as a high-resolution industrial digital camera. The non-linear optimization of bundle adjustment is finished by the Newton-like method.

### 2.3.1 Bundle Adjustment: Optimization by the Newton-like Algorithm

All of the Newton-like methods, regardless of their names, are based on Gauss-Newton method. Here, “Newton-like” means each algorithm in this branch is the modification of Gauss-Newton method described below. The goal of modification is to make calculation more efficiently and find the real minimizer (including the avoidance of saddle point). These algorithms modify the Gauss-Newton method in the following aspects:

- How to handle the  $2^{nd}$  order term in the Hessian matrix: a) Ignore the  $2^{nd}$  order term or b) do other approximation instead of  $2^{nd}$  order term;
- How to adjust the convergence direction  $\delta x$ , such as steepest descent direction or conjugate direction;
- How to do the step control for step size  $\lambda$ .

#### The Gauss-Newton method

In section 2.1.3, the implicit camera model is

$$y = f(x, \mathbf{u}, c), \quad (2.47)$$

which has a cost function  $g(x)$ . The Gauss-Newton method uses the least squares estimator (LSE), which requires a SSE (sum of squared error) cost function. Thus, the corresponding

$g(x)$  in Gauss-Newton method is

$$g(\mathbf{u}) = \frac{1}{2} \sum_{i=1}^M \|r_i(\mathbf{u})\|_2^2 = \frac{1}{2} R(\mathbf{u})^T R(\mathbf{u}), \quad (2.48)$$

where

$$R(\mathbf{u}) = \bar{y} - \hat{y} = {}^p\bar{P}_a - {}^p\hat{P}_a \quad (2.49)$$

is the residual function. Variable  $\bar{y} = {}^p\bar{P}_a$  is the observation from the system physically;  $\hat{y} = {}^p\hat{P}_a$  is the estimation of the output from system modeling. So, with the statements about camera model with lens distortion in section (2.1.3), the objective of the optimization is to minimize the difference between experimental observation and mathematical modeling values of the same variable. In other words, the optimal estimation for camera IO and EO represented by  $u$  is found by minimizing the residual. However, the cost function of a complicated system is quite expensive to compute. Thus, Taylor Series expression is used to approximate the cost function as below:

$$g(\mathbf{u}) = g(\mathbf{u}_c) + (\mathbf{u} - \mathbf{u}_c)g'(\mathbf{u}_c) + \frac{1}{2}(\mathbf{u} - \mathbf{u}_c)^2 g''(\mathbf{u}_c) + \dots, \quad (2.50)$$

where  $\mathbf{u}_c$  is the solution of equation (2.50).

The Gauss-Newton method chooses to keep the first three terms for precise approximation of  $g(\mathbf{u})$ :

$$g(\mathbf{u}) = g(\mathbf{u}_c) + (\mathbf{u} - \mathbf{u}_c)g'(\mathbf{u}_c) + \frac{1}{2}(\mathbf{u} - \mathbf{u}_c)^2 g''(\mathbf{u}_c). \quad (2.51)$$

An alternative way to write the above equation is

$$g(\mathbf{u}) = g(\mathbf{u}_c) + \nabla g(\mathbf{u}_c)^T (\mathbf{u} - \mathbf{u}_c) + \frac{1}{2}(\mathbf{u} - \mathbf{u}_c)^T \nabla^2 g(\mathbf{u}_c) (\mathbf{u} - \mathbf{u}_c). \quad (2.52)$$

To minimize function  $g(\mathbf{u})$  is to solve  $g'(\mathbf{u}^*) = 0$  to find the corresponding  $\mathbf{u}_c$  as the estimation of  $\mathbf{u}^*$ , which is to minimize the residual function  $g(\mathbf{u})$  for optimization:

$$\frac{\partial g(\mathbf{u})}{\partial \mathbf{u}} = \nabla g(\mathbf{u}_c)^T + \nabla^2 g(\mathbf{u}_c)(\mathbf{u} - \mathbf{u}_c) = 0. \quad (2.53)$$

If the termination condition of iteration is reached, the current  $\mathbf{u}_c$  is the estimation of  $\mathbf{u}^*$ , or  $\hat{\mathbf{u}} = \mathbf{u}_c$ , the estimated minimizer. The iterative part is the  $c^{th}$  solution  $\mathbf{u}$  is used as the  $\mathbf{u}_{c+1}$  for next iterative step before the end of iteration. So, the iteration formula is

$$\mathbf{u}_{c+1} = \mathbf{u}_c - \frac{\nabla g(\mathbf{u}_c)}{\nabla^2 g(\mathbf{u}_c)} = \mathbf{u}_c - \frac{1}{\nabla^2 g(\mathbf{u}_c)} \nabla g(\mathbf{u}_c) = \mathbf{u}_c + \delta \mathbf{u}, \quad (2.54)$$

where  $\delta \mathbf{u}$  is called the Gauss-Newton descent direction as well as the updated term for the  $c^{th}$  solution  $\mathbf{u}$ .

Substituting equation (2.48) into equation (2.54), we get

$$\mathbf{u}_{c+1} = \mathbf{u}_c - \nabla \left\{ \frac{1}{2} R(\mathbf{u}_c)^T R(\mathbf{u}_c) \right\}^T \left\{ \nabla^2 \left\{ \frac{1}{2} R(\mathbf{u}_c)^T R(\mathbf{u}_c) \right\} \right\}^{-1}. \quad (2.55)$$

The numerator term in equation (2.55) gives us:

$$\nabla \left\{ \frac{1}{2} R(\mathbf{u}_c)^T R(\mathbf{u}_c) \right\} = R'(\mathbf{u})^T R(\mathbf{u}) = J^T \boldsymbol{\varepsilon}, \quad (2.56)$$

where

$$J = R'(\mathbf{u})_{i,j} = \frac{\partial r_i(\mathbf{u})}{\partial \mathbf{u}_j} \in R^{n_{ob} \times n_{\hat{u}}} \quad (2.57)$$

is the **Jacobian matrix** of cost function  $g(\mathbf{u})$ ,  $n_{\hat{u}}$  is the number of parameters to be estimated,  $j = 1, 2, 3, \dots, n_{\hat{u}}$ ,  $n_{ob}$  is the number of observed data points, and  $i = 1, 2, 3, \dots, n_{ob}$ .

$$\boldsymbol{\varepsilon} = \bar{y} - \hat{y} = R(\mathbf{u}) \quad (2.58)$$

is the residual function. The denominator term in equation (2.55) gives us the **Hessian**

**matrix:**

$$\begin{aligned}
 H = \nabla^2 g(\mathbf{u}) &= \nabla^2 \left\{ \frac{1}{2} R(\mathbf{u}_c)^T R(\mathbf{u}_c) \right\} = R'(\mathbf{u})^T R'(\mathbf{u}) + \sum_{j=1}^M r_j(\mathbf{u})^T \nabla^2 r_j(\mathbf{u}) \\
 &= R'(\mathbf{u})^T R'(\mathbf{u}) + R''(\mathbf{u})^T R(\mathbf{u}), \quad (2.59)
 \end{aligned}$$

where  $R''(\mathbf{u})$  is called tensor. The problem is that this cost function is expensive to minimize in closed-form, even with an up-to-date computer. Thus, the Gauss-Newton approximation is performed by ignoring the second order term  $R''(\mathbf{u})^T R(\mathbf{u})$  to get the simplified Hessian matrix:

$$H = \nabla^2 f(\mathbf{u}) = R'(\mathbf{u})^T R'(\mathbf{u}) = J^T J \in R^{n_{\hat{u}} \times n_{\hat{u}}}. \quad (2.60)$$

This equation allows for faster calculation, since we use the first order Jacobian to construct the second order Hessian. But it sacrifices the convergence rate from 2nd-order convergence down to first order. However, if  $R(\mathbf{u}^*) \approx 0$ , the Gauss-Newton method can also achieve second order convergence.

The approximation of ignoring  $R''(\mathbf{u})^T R(\mathbf{u})$  in equation (2.59) gives us the iteration formula

$$\mathbf{u}_{c+1} = \mathbf{u}_c - (J^T J)^{-1} J^T \boldsymbol{\varepsilon} = \mathbf{u}_c + \delta \mathbf{u}, \quad (2.61)$$

where  $\delta \mathbf{u}$  is

$$\delta \mathbf{u} = -(J^T J)^{-1} \nabla f(\mathbf{u}) = -(J^T J)^{-1} J^T \boldsymbol{\varepsilon}. \quad (2.62)$$

Equation (2.62) is the so-called normal equation and is also known as the left pseudo-inverse solution.

### 2.3.2 Bundle Adjustment: Issues

The self-calibrating bundle adjustment was first introduced in the 1970s. Self-calibrating bundle adjustment needs a minimal constraint to define network datum such as the IO constraint. Therefore, an uncalibrated<sup>18</sup> object with landmarks is needed. The important factors in self-calibration are the geometrical arrangement of the camera stations, the intersection angles of rays from object points to cameras, the number of landmarks seen from a diversity of camera locations, and the spread of landmarks across the image format, [22]. Self-camera calibration is currently widely used, since it finishes the calculation automatically and does not require the user to be an expert in close-range photogrammetry, which is a very flexible method.

Granshaw [25] systemically discussed the theory of bundle adjustment. He demonstrated the improvement of estimation accuracy of using multi-station geometry based on single camera geometry instead of dual camera geometry by analysis of the propagation of errors during camera calibration. He also mentioned the error-free control points for avoiding datum defects (rank deficient of the Hessian matrix), which are the fixed (non-updated) seven parameters for absolute orientation of a camera. Without definition of absolute orientation by seven fixed parameters, only the shape of object can be recovered, not the position of the object with respect to the camera in 3D space. However, free network bundle adjustment based on multi-camera geometry, which extends the Hessian matrix in a normal equation and finds estimation by minimizing the trace of covariance matrix (the inverse of the Hessian matrix) is promising. He showed that the performance of multi-station bundle adjustment without control points is as good as the performance with control points. He also gave the definition of self-calibrating bundle adjustment as the procedure of the recovery of camera IO, EO and additional parameters without special compensation (control points) of lens distortion for system error. While it is used to be popular to add additional parameters (APs) to camera interior parameters to model the

---

18. *Uncalibrated* means non-priori measurement of  ${}^tP_a$ .



lens distortion effect, researchers [54] found that unreasonable additional parameters would lower the accuracy, and Brown's certain parameters [6] are the most popular choice.

Triggs *et al.* [59] did a great job in their review of bundle adjustment (BA). They described bundle adjustment as “the problem of refining a visual reconstruction to produce jointly optimal 3D structure and viewing parameter (camera pose) estimates”. The word *optimal* indicates bundle adjustment estimate parameters by minimizing model fitting error. The word *jointly* indicates the estimation of camera IO, EO and lens distortion simultaneously. Several issues in the application of bundle adjustment were covered by Triggs *et al.* For the representation of the rotation matrix, they suggested the use of quaternion or inclusion of a small local perturbation term  $\delta R$  for rotation matrix  $R \rightarrow R \cdot \delta R$ . For modeling cost function in non-linear optimization, they recommended using a robust estimator to lower the influence of outliers due to large-scale estimation. Here, a large-scale problem means that the bundle adjustment has hundreds or thousands of images with tens of thousands of landmarks on images and thousands of parameters to be estimated. The sparse method for a matrix is helpful with reasonable ordering of the matrix. For the Hessian matrix in normal equation, they stated the first order method by the approximation of the Hessian matrix offers much slower convergence than the second order method in bundle adjustment. Moreover, variable scaling or preconditioning is important to get the good-conditioned Hessian and also improves the first order problem. They pointed out that discarding the  $2^{nd}$  order term is not necessary to lower the  $2^{nd}$  convergence rate of Newton-like method to the  $1^{st}$  order method. The real reason is the significant ignorance of the off-diagonal element of the Hessian matrix, which is possible by discarding the  $2^{nd}$  order term in the Hessian matrix. They recommended the use of network design before the experiment and quality control to evaluate the result of bundle adjustment.

### 2.3.3 Bundle Adjustment: Evaluation and Improvement

Over the years, researchers in the camera calibration area have worked to evaluate and

improve camera calibration accuracy.

### **The evaluation of bundle adjustment**

It is important to evaluate the accuracy of camera calibration after calculation. Gruen [26] discussed the precision and reliability issues in BA. He stated that accuracy includes two aspects: the precision, the statistical quality of the estimation and reliably, the ability to detect errors (outliers in the data). For precision measurement, he used the variance of each element by extracting the trace of the covariance matrix:  $\sigma_x^2 = \frac{tr(Cov_x)}{n_x}$ ,  $\sigma_y^2 = \frac{tr(Cov_y)}{n_y}$ ,  $\sigma_z^2 = \frac{tr(Cov_z)}{n_z}$ . He divided reliability into internal reliability, the ability of detecting an outlier, and external reliability, the influence of the outliers on the final estimation. The experimental analysis recommended the rejection of non-determinable APs to improve estimation accuracy as well as a larger convergence angle in the network can also help to maintain reliability. He recommended that a basic configuration of network design is four stations with two bases perpendicular to each other for avoidance of pure epipolar plane observations. A larger base improves the external reliability significantly. However, they use the least squares estimator but not the robust estimator.

Granshaw [25] divided the examination into two parts: the first changes in the number, position, and orientation of the photographs, the network geometry, and the second is in the object space control. He inspected the covariance matrix (the inverse of the Hessian matrix) by applying matrix partitioning.

Actually, almost every researchers evaluated their claim in statistics [1, 3, 6, 8, 9, 11, 12, 17, 23, 24, 26, 33, 34, 37, 48, 50, 53, 54, 60, 65, 64, 63, 62, 68, 67].

### **The improvement of bundle adjustment**

Beyond improvement of camera modeling and calculation algorithm, **network design** had proven as an effective tool to improve the accuracy of camera calibration in photogrammetry. Simply speaking, it addressed how to place a camera in 3D space so it can collect better images for camera calibration.

Fraser [18, 19, 20] formally introduced the network design and stated that the purpose of network design, as well as network geometry, is to design the photogrammetric imaging configuration, which involves finding the best placement of the camera in 3D space to collect image data with the calibration tool. A good network design would improve the final estimation accuracy. The general idea is to a) try different camera station positions for camera calibration. b) use the calibration result to validate which network is the best one. This will give us the general rule of camera placement. During the design, three aspects should be considered: selection of an image scale; number of camera stations; and the relative geometry relationship of the camera stations. Fraser gave the classification of design stages: zero-order design (ZOD), the datum problem for optimal landmarks coordinates; first-order design (FOD), the configuration problem for optimal network geometry; and second-order design (SOD), the weight problem for optimal observational precision. FOD is a major network design problem for close-range photogrammetry.

About improvement on the estimation, he concluded that: a) it is important that each basic camera station see all the landmarks on the calibration tool. There is a trade-off between the FOV of the camera and the focal length; b) if make  $k > 1$ , which would take more than one at a station, this would improve the accuracy; c) the strong basic stations configuration keeps camera calibration at a high accuracy level; and d) more camera stations aside from the basic stations does not necessarily generate a better result, but it does improve the precision of landmark in image coordinates.

Kenneth and Fraser[16] introduced the OLT (on-line triangulation) with sequential estimation<sup>19</sup> to single sensor vision metrology in application, which unlike simultaneous least squares estimation, can monitor object precision during data acquisition and offer estimation quality evaluation before the completion of estimation. The estimation allows system to deal with added image during image acquisition. However, for non-linear estimation, the sequential estimation is less powerful, which will cause the estimated

---

19. Sequential estimation is also called as process identification or adjustment-in-steps or phased adjustment.

parameters vector to drift as data are added. Thus, significantly accurate initial values for parameters are required, and a simultaneous estimation is needed for double check. They re-emphasized that good network design will enhance the parameter recovery. In their experiment, they collected four images ( $\kappa = 0^\circ, 90^\circ, 180^\circ, 270^\circ$ ) at each of six camera stations. In the experimental analysis, they compared the performance of simultaneous adjustment and sequential adjustment in computation time and projective point precision. They also covered the additional close-range photogrammetry applications such as the network design.

Saadatseresht *et al.* [55] were looking for a method of enhancing the accuracy performance of photogrammetric system based on network design, which can select additional camera stations automatically (VUS and VUF automatic calculation). Under two assumptions, they introduced visibility uncertainty prediction<sup>20</sup> (VUP) and visibility uncertainty spheres<sup>21</sup> (VUS) for analysis. The authors found out that during the network design, there are certain constraints that can impact the pick-up of the camera station and accuracy fulfillment (AF) of the camera station. Their strategy was to redo BA for camera calibration based on data deducted from the network and then calculate VUS and VUP<sup>22</sup>. A car door mounted with a retro-reflective target and a box as the obstruction were used in first test. If compared to previous conclusions in photogrammetry network design, claims in this article are that 1) new camera station might offer better network geometry than repeatedly grabbed images at the same camera station; 2) uniform distribution of camera stations in the original networks would improve the visibility modeling; and 3) better VUS and VUF are available with more camera stations.

---

20. By fuzzy visibility value  $v$ ,  $v = 1$  means perfectly visible;  $v = 0$  means invisible;  $0 < v < 1$  means the percentage possibility in visibility.

21. VUS is used to graphically show the visible and non-visible spot for VUP in 3D space.

22. VUP (the number of AF camera) is calculated by least-squares-based interpolation to predict the AF.

## 2.4 Summary

In this chapter, literature about camera calibration is reviewed. At the beginning, the fundamental knowledge of coordinate system is introduced, including coordinate representation and coordinate transformation of 3D object. Next, the different types of camera geometries and camera models are introduced. Based on the definition of camera calibration, various camera calibration methods are reviewed in the field, such as linear algorithms, non-linear and hybrid algorithms, and bundle adjustment. Bundle adjustment is considered to be the most powerful technology for camera calibration calculation in application. Detailed information about bundle adjustment algorithm is provided, and the limitations of bundle adjustment are offered. Other researchers' methodologies for the improvement of bundle adjustment are also reviewed.

## 3 Representation Enhancement: Spherical Coordinate Representation

### 3.1 Singularity of Representation

According to section 2.1, it is clear that we need to represent the position and orientation of landmark in 3D space first and then do the transformation to apply camera geometry for camera calibration. The representation is required to represent the landmark uniquely. However, singularity of representation leads to failure in the unique representation of poses and the camera calibration calculation.

Singularity of representation arises in several research areas with usage of 3D space representation such as kinematic control [11, 24], hand-eye problems [56, 61], and camera calibration [59]. In kinematic control area, Gosselin and Angeles [24] classified singularity into three types by matrix determinant in closed-loop kinematic chains. Chiaverini [11] primarily discussed singularity and the handling of singularity in the task-priority strategy. He introduced the augmented Jacobian matrix and developed a damped least squares solution instead of the classic inverse kinematic solution to handle the occurrence of singularity.

#### 3.1.1 Definition of “Singularity of Representation” (SoR)

Singularity of representation is an important issue in camera calibration [53, 58, 59]. Triggs *et al.* [59] pointed out that “parametrization singularities cause ill-conditioning and erratic numerical behaviour.” of bundle adjustment. Ill-conditioning of the **Hessian matrix** leads to failure or false convergence of the algorithm, which downgrades corresponding reliability of the iterative algorithm. While not giving a formal definition of singularity, Triggs *et al.* [59] stated that singularity prevents the representation of 3D features from

covering the whole parameter space uniformly. The mathematical definition for singularity of representation is:

**Definition:** Singularity of Representation (SoR) is defined as that a Jacobian matrix  $J \in R^{n_{ob} \times n_{\hat{u}}}$  does not have full rank  $\text{rank}(J) \neq n_{\hat{u}}$ , but  $\text{rank}(J) < n_{\hat{u}}$ , where the  $n_{ob}$  is the number of observations and  $n_{\hat{u}}$  is the number of parameters to be estimated,  $n_{ob} \geq n_{\hat{u}}$ .

### 3.1.2 Loss of One Degree of Freedom (DOF)

The source of singularity is in representation of pose and is shown in the **Jacobian matrix** during calculation. Actually, this definition is also applicable to the singularity in kinematics. In 3D space, singularity is demonstrated as the loss of one degree of freedom, which is equal to the rank deficiency of the Jacobian matrix<sup>23</sup>.

Based on an Euler angles representation, in 3D rotation, three dimensions move freely by an asymmetric<sup>24</sup> order combination. For example, a common order is in y-axis→x axis→z axis by right hand rule. Note that in the rotation order y-axis→x axis→z axis, the movement around the y axis is the parent movement of the x and z axes. The movement of the x axis is the parent movement of the z axis. If the rotation axis is not fixed but makes the rotation about the axis locally, the parent movement would lead to the sub axis movement at the same time, then to the local movement about sub axes. So movement of the y axis affects movements of the x and z axes. Movement of the x axis affects movement of the z axis. However, movement of the z axis can not affect movement of the x and y axes.

The problem is that the free movement would make two axes in one direction. Then there is no difference when you move one axis or the other axis (because rotation is on one plane). The effect is that a pose cannot be reached by movement around the x, y, and z axes. The loss of one degree of freedom is also known as gimbal lock [27]. Gimbal lock cannot be avoided, and different orders of representation of rotation have different poses in gimbal lock. In other words, not every rotation can be realized by a change in Euler angles.

23. An active DOF counts one rank of the corresponding matrix.

24. The symmetric order has another six possible combinations.

## 3.2 Singularity of Representation in Bundle Adjustment

### 3.2.1 The Jacobian Matrix and the Hessian Matrix

From the definition of singularity of representation (SoR) given above, we know that the SoR in bundle adjustment arises with the Jacobian matrix. If the SoR gives us the ill-conditioned Jacobian and Hessian matrices, the camera calibration may have difficulty keeping reliability or even converging.

The Jacobian matrix of implicit camera model  $y = f(\mathbf{u}, x, c)$  mentioned in section 2.3.1 is written as:

$$J = \begin{bmatrix} \frac{d\hat{y}}{d\hat{\chi}} & \frac{d\hat{y}}{d^w\hat{\mathcal{P}}_i} & \frac{d\hat{y}}{d^w\hat{\mathcal{S}}_c} \end{bmatrix} \in R^{n_{ob} \times n_{\hat{\mathbf{u}}}}, \quad (3.1)$$

where  $\hat{y}$  is the  ${}^p\hat{P}_a$ ,  $\hat{\chi}$  is the estimated camera model,  ${}^w\hat{\mathcal{S}}_c$  is the estimated camera pose expressed in spherical coordinates, and  ${}^w\hat{\mathcal{P}}_i$  is the estimated landmark position expressed in Cartesian coordinates<sup>25</sup>. If a singularity raises, it happens in the sub-matrix corresponding to each camera pose:

$$J_{w\mathcal{S}_c} = \frac{d{}^p\hat{P}_a}{d^w\hat{\mathcal{S}}_c} = \begin{bmatrix} \frac{d{}^p\hat{P}_a}{d\varphi} & \frac{d{}^p\hat{P}_a}{d\omega} & \frac{d{}^p\hat{P}_a}{d^sZ_a} & \frac{d{}^p\hat{P}_a}{d^s\varphi} & \frac{d{}^p\hat{P}_a}{d^s\omega} & \frac{d{}^p\hat{P}_a}{d^s\kappa} \end{bmatrix} \in R^{n_{ob} \times 6}, \quad (3.2)$$

where  $\hat{y}$  is the  ${}^pP_a$ , the landmark  $a$  in pixel coordinates, and  $\frac{n_{ob}}{2}$  is the number of landmarks observed by camera<sup>26</sup>.

An example of the Jacobian and Hessian matrices calculated by an experimental data set that only contains three images is shown in figures 3.1 and 3.2. The black area means a zero value, and the white area means a non-zero value, in both figures. Most spots in the matrix are zero; thus, a sparse algorithm can save the space of storage and time of calculation. Generally, there are many more than three images in a data set for bundle adjustment. In figure 3.1, the part (a) is the full Jacobian matrix, derived by equation (3.1).

You can see that the parameter ordering of the full version of the Jacobian matrix in the

25. Landmarks are all points in 3D space, thus no orientation is needed.

26. Note that  ${}^pP_a \in R^2$ , thus the number of observations is multiplied by 2.



row from left to right is parameters of camera model, calibration tool pose, and camera pose. Thus, the sub matrix for camera pose can be extracted from the Jacobian matrix, which is shown in (b) of figure 3.1. The part (b) is the three sub-matrices corresponding to each camera pose from the digital image, derived by equation (3.2). The  $J_{w\hat{\mathcal{S}}_c}$  matrix is constructed by column vectors and has  $73 \times 2$  rows for each camera pose based on the fact that each calibration tool has 73 landmarks on it. So the size of structure in part (b) of figure 3.1 is  $438 \times 6 = 73 \cdot 2 \cdot 3 \times 6$ .

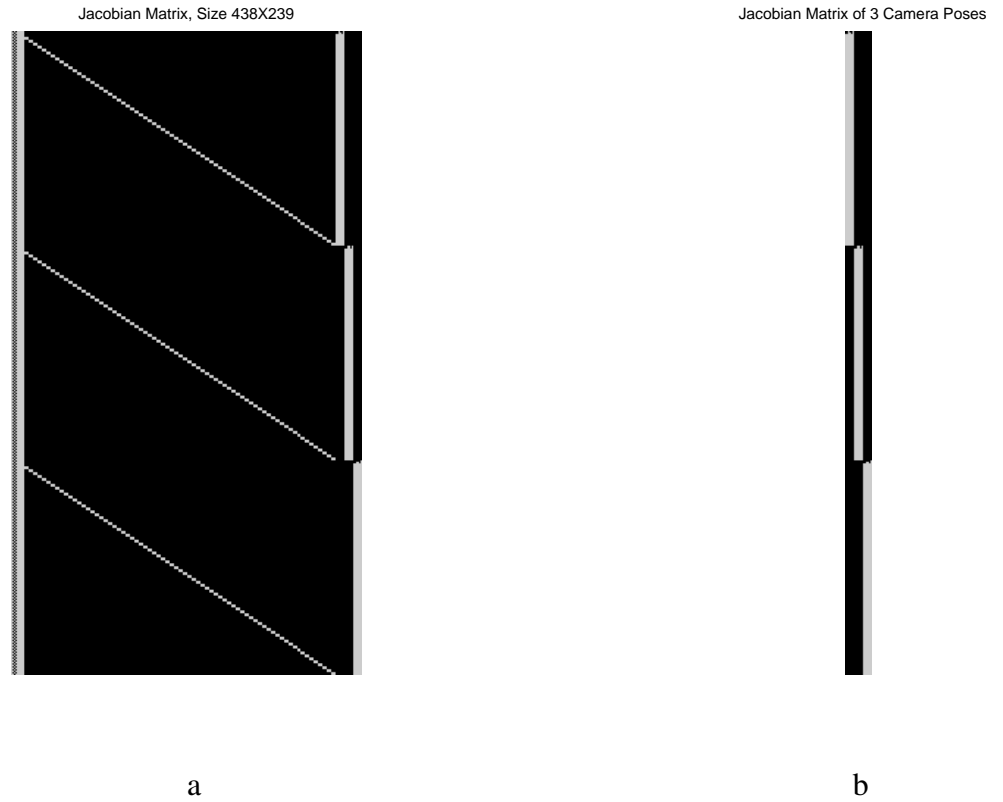


Figure 3.1: The Jacobian matrix of  $\hat{\mathbf{u}}$  (a) and the Jacobian matrix of each camera pose (b): the black area means a zero value and the white area means a non-zero value.

Figure 3.2 shows the Hessian matrix. In the row from left to right are parameters of the camera model, calibration tool pose, and camera pose, derived by

$$H = J^T \cdot J \in R^{n_{\hat{\mathbf{u}}} \times n_{\hat{\mathbf{u}}}} . \quad (3.3)$$

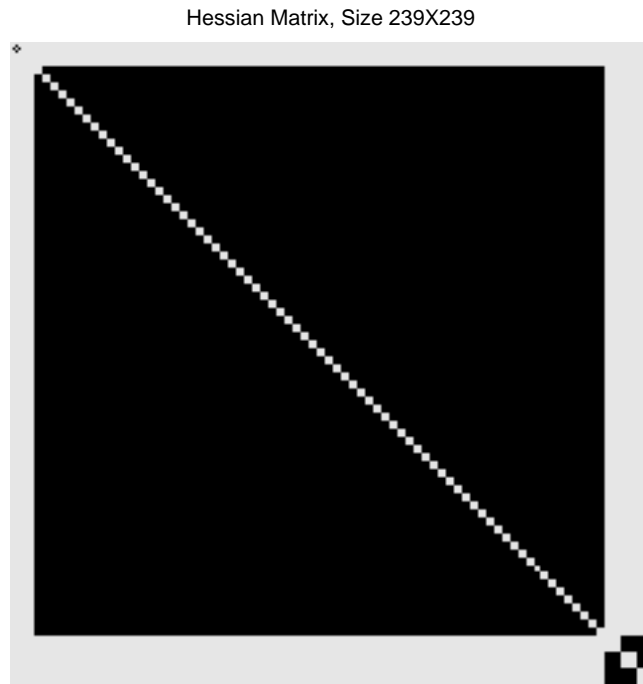


Figure 3.2: The Hessian matrix: the black area means a zero value and the white area means a non-zero value.

Note that there are three  $6 \times 6$  square sub matrices corresponding to each digital image, which is shown in figure 3.3, and derived by

$$H_{w_{\mathcal{S}_c}} = J_{w_{\mathcal{S}_c}}^T \cdot J_{w_{\mathcal{S}_c}} \in R^{6 \times 6}. \quad (3.4)$$

Hessian Matrix of 3 Camera Poses



Figure 3.3: The Hessian matrix of camera pose: the black area means a zero value and the white area means a non-zero value.

### 3.2.2 Singularity of Representation in Bundle Adjustment

Mathematically, singularity would impact the bundle adjustment algorithm, because we need to solve the normal equation:

$$J^T J \cdot \delta x = J^T \cdot \epsilon. \quad (3.5)$$

If performing the left pseudo-inverse for an over-determined question, we have equation (2.62):

$$\delta x = (J^T J)^{-1} \cdot J^T \cdot \epsilon. \quad (3.6)$$

Buss and Kim [7] pointed out that the pseudo-inverse method tends to have problems in the neighborhoods of singularities. Due to an unachievable position with loss of one degree of freedom, the Jacobian matrix no longer has full column rank. If the Euler angles

are extracted from a rotation matrix which is near to a singularity position, then the pseudo-inverse method will lead to very large and sudden changes in angles.

Singular value decomposition<sup>27</sup> (SVD) is a sub-algorithm in bundle adjustment. If applying SVD to the Jacobian matrix, we have:

$$J = U \cdot \Sigma \cdot V^H = \sum_{i=1}^r \sigma_i \cdot u_i \cdot v_i^H, \quad (3.7)$$

where  $J \in R^{n_{ob} \times n_{\hat{u}}}$  and

$$U = \sum_{i=1}^r u_i \in R^{n_{ob} \times n_{ob}} \quad (3.8)$$

is an orthogonal matrix,

$$\Sigma = \sum_{i=1}^r \sigma_i \in R^{n_{ob} \times n_{\hat{u}}} \quad (3.9)$$

is a matrix with non-negative real numbers on the diagonal, and

$$V = \sum_{i=1}^r v_i \in R^{n_{\hat{u}} \times n_{\hat{u}}} \quad (3.10)$$

is an orthogonal matrix. Correspondingly, the Hessian matrix is:

$$H = J^T \cdot J = (V^H)^T \Sigma^T U^T \cdot U \Sigma V^H = V \cdot \Sigma^T \Sigma \cdot V^H. \quad (3.11)$$

Since  $\Sigma$  is a diagonal matrix, then

$$H = J^T \cdot J = V \cdot \Sigma^2 \cdot V^H. \quad (3.12)$$

In a non-singularity situation, the rank of singular value matrix  $\Sigma$  is equal to rank of the Hessian matrix  $H$ :

$$\text{rank}(\Sigma) = \text{rank}(H),$$

---

27. Note that this name has no relationship with the singularity of representation.

where the  $\text{rank}(\Sigma)$  is the number of non-zero diagonal entries of  $\Sigma$  matrix. In a singularity situation, the Hessian matrix shown in equation (3.11) is rank deficient. Specifically, the number of non-zero diagonal elements of  $\Sigma$  is less than it is in the non-singularity situation. In other words, there are **zeros** in diagonal position of  $H$ . Correspondingly, forming the inverse of the Hessian matrix as  $H^{-1} = (J^T J)^{-1}$  is impossible. Thus, singularity leads to the failure of estimation in the bundle adjustment algorithm.

### 3.2.3 An Example of Singularity of Representation

The observation model transforms landmark (denoted by  $a$ ) position in calibration tool coordinates (denoted by  $t$ ),  ${}^tP_a$ , to landmark in pixel coordinates (denoted by  $q$ ),  ${}^qP_a$ . Thus, the transformation from  ${}^tP_a$  to  ${}^qP_a$  is done through equations (2.14), (2.15), (2.16), and (2.17):

$$\begin{bmatrix} {}^qP_a \\ 1 \end{bmatrix} = \begin{bmatrix} -c_p k_x & 0 & -c_p x_0 \\ 0 & -c_p k_y & -c_p y_0 \\ 0 & 0 & 1 \end{bmatrix} \frac{{}^cP_a}{cP_a(3)}. \quad (3.13)$$

The  ${}^cP_a$  is calculated by

$${}^cP_a = {}^cR({}^tP_a - {}^tP_{\hat{c}}), \quad (3.14)$$

where the rotation matrix  ${}^cR$  is expressed in spherical coordinates by introducing  ${}_sR$ :

$${}_tR = {}_sR \cdot {}_tR = {}_sR \cdot {}_sR^T = (R_y^{180^\circ} \cdot R_{\varphi} \cdot R_{\omega} \cdot R_{\kappa}) \cdot (R_{\omega}^T \cdot R_{\varphi}^T) \quad (3.15)$$

and

$$\begin{aligned}
{}^cP_{\hat{t}} &= -{}^cR {}^tP_{\hat{c}} \\
&= -{}^cR {}^sR {}^tR {}^sR \begin{bmatrix} 0 \\ 0 \\ {}^sZ \end{bmatrix} = -{}^cR \begin{bmatrix} 0 \\ 0 \\ {}^sZ \end{bmatrix} = -(R_y^{180^\circ} \cdot R_{\varphi} \cdot R_{\omega} \cdot R_{\kappa}) \cdot \begin{bmatrix} 0 \\ 0 \\ {}^sZ_{\hat{c}} \end{bmatrix} \\
&= - \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} C_{\varphi} & 0 & S_{\varphi} \\ 0 & 1 & 0 \\ -S_{\varphi} & 0 & C_{\varphi} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_{\omega} & -S_{\omega} \\ 0 & S_{\omega} & C_{\omega} \end{bmatrix} \begin{bmatrix} C_{\kappa} & -S_{\kappa} & 0 \\ S_{\kappa} & C_{\kappa} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ {}^sZ_{\hat{c}} \end{bmatrix} \\
&= - \begin{bmatrix} -(C_{\varphi}C_{\kappa} + S_{\omega}S_{\varphi}S_{\kappa}) & S_{\kappa}C_{\varphi} - S_{\omega}S_{\varphi}C_{\kappa} & -C_{\omega}S_{\varphi} \\ C_{\omega}S_{\kappa} & C_{\omega}C_{\kappa} & -S_{\omega} \\ S_{\varphi}C_{\kappa} - C_{\varphi}S_{\omega}S_{\kappa} & -(S_{\varphi}S_{\kappa} + C_{\varphi}S_{\omega}C_{\kappa}) & -C_{\varphi}C_{\omega} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ {}^sZ_{\hat{c}} \end{bmatrix} \\
&= \begin{bmatrix} C_{\omega}S_{\varphi} \\ S_{\omega} \\ C_{\varphi}C_{\omega} \end{bmatrix} \cdot {}^sZ_{\hat{c}} \tag{3.16}
\end{aligned}$$

is the origin of spherical calibration tool expressed in the camera frame. Then, substituting equations (3.14) into (3.15) , we have

$$\begin{aligned}
{}^cP_a &= (R_y^{180^\circ} \cdot R_{\varphi} \cdot R_{\omega} \cdot R_{\kappa}) \cdot (R_{\omega}^T \cdot R_{\varphi}^T) \cdot ({}^tP_a - {}^tP_{\hat{c}}) \\
&= (R_y^{180^\circ} \cdot R_{\varphi} \cdot R_{\omega} \cdot R_{\kappa}) \cdot (R_{\omega}^T \cdot R_{\varphi}^T \cdot {}^tP_a - \begin{bmatrix} 0 \\ 0 \\ {}^sZ_{\hat{c}} \end{bmatrix}) \\
&= {}^cR \left( R_{\omega}^T \cdot R_{\varphi}^T \cdot {}^tP_a - \begin{bmatrix} 0 \\ 0 \\ {}^sZ_{\hat{c}} \end{bmatrix} \right) \tag{3.17}
\end{aligned}$$

Therefore, there is

$${}^qP_a = \begin{bmatrix} -c_p k_x \frac{{}^cP_a(1)}{{}^cP_a(3)} - c_p x_0 \\ -c_p k_y \frac{{}^cP_a(2)}{{}^cP_a(3)} - c_p y_0 \end{bmatrix} \quad (3.18)$$

To calculate the Jacobian matrix corresponding to each camera pose, we can substitute equation (3.18) into equation (3.2) and apply the derivative of  ${}^qP_a$  with respect to each element of  ${}^wS_{\hat{c}}$  in equation (3.2). The derivative of observation  $\frac{d{}^q\hat{P}_a}{d{}^wS_{\hat{c}}}$  is:

$$J_{wS_{\hat{c}}} = \frac{d{}^q\hat{P}_a}{d{}^w\hat{S}_{\hat{c}}} = \begin{bmatrix} \frac{d{}^q\hat{P}_a}{d\phi} & \frac{d{}^q\hat{P}_a}{d\omega} & \frac{d{}^q\hat{P}_a}{d^sZ_{\hat{c}}} & \frac{d{}^q\hat{P}_a}{d^s\phi} & \frac{d{}^q\hat{P}_a}{d^s\omega} & \frac{d{}^q\hat{P}_a}{d^s\kappa} \end{bmatrix}, \quad (3.19)$$

where each column vector of  $\frac{d{}^q\hat{P}_a}{d{}^wS_{\hat{c}}}$  is:

$$\frac{d{}^q\hat{P}_a}{d\phi} = -c_p \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \left( \frac{1}{{}^cP_a(3)} \frac{d{}^cP_a(1:2)}{d\phi} - \left( \frac{{}^cP_a(1:2)}{{}^cP_a^2(3)} \right) \frac{d{}^cP_a(3)}{d\phi} \right) \quad (3.20)$$

$$\frac{d{}^q\hat{P}_a}{d\omega} = -c_p \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \left( \frac{1}{{}^cP_a(3)} \frac{d{}^cP_a(1:2)}{d\omega} - \left( \frac{{}^cP_a(1:2)}{{}^cP_a^2(3)} \right) \frac{d{}^cP_a(3)}{d\omega} \right) \quad (3.21)$$

$$\frac{d{}^q\hat{P}_a}{d^sZ_{\hat{c}}} = -c_p \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \left( \frac{1}{{}^cP_a(3)} \frac{d{}^cP_a(1:2)}{d^sZ} - \left( \frac{{}^cP_a(1:2)}{{}^cP_a^2(3)} \right) \frac{d{}^cP_a(3)}{d^sZ_{\hat{c}}} \right) \quad (3.22)$$

$$\frac{d{}^q\hat{P}_a}{d^s\phi} = -c_p \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \left( \frac{1}{{}^cP_a(3)} \frac{d{}^cP_a(1:2)}{d^s\phi} - \left( \frac{{}^cP_a(1:2)}{{}^cP_a^2(3)} \right) \frac{d{}^cP_a(3)}{d^s\phi} \right) \quad (3.23)$$

$$\frac{d{}^q\hat{P}_a}{d^s\omega} = -c_p \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \left( \frac{1}{{}^cP_a(3)} \frac{d{}^cP_a(1:2)}{d^s\omega} - \left( \frac{{}^cP_a(1:2)}{{}^cP_a^2(3)} \right) \frac{d{}^cP_a(3)}{d^s\omega} \right) \quad (3.24)$$

$$\frac{d^q \hat{P}_a}{d^s \kappa} = -c_p \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \left( \frac{1}{{}^c P_a(3)} \frac{d^c P_a(1:2)}{d^s \kappa} - \left( \frac{{}^c P_a(1:2)}{{}^c P_a^2(3)} \right) \frac{d^c P_a(3)}{d^s \kappa} \right). \quad (3.25)$$

The  $\left[ \begin{array}{cccccc} \frac{d^c P_a}{d\varphi} & \frac{d^c P_a}{d\omega} & \frac{d^c P_a}{d^s Z_{\hat{c}}} & \frac{d^c P_a}{d^s \varphi} & \frac{d^c P_a}{d^s \omega} & \frac{d^c P_a}{d^s \kappa} \end{array} \right]$  for each column vector in equations (3.20)-(3.24) and (3.25) is:

$$\frac{d^c P_a}{d\varphi} = {}^s R \cdot R_{\omega}^T \cdot \frac{dR_{\varphi}^T}{d\varphi} \cdot {}^t P_a \quad (3.26)$$

$$\frac{d^c P_a}{d\omega} = {}^s R \cdot \frac{dR_{\omega}^T}{d\omega} \cdot R_{\varphi}^T \cdot {}^t P_a \quad (3.27)$$

$$\frac{d^c P_a}{d^s Z_{\hat{c}}} = -{}^s R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.28)$$

$$\frac{d^c P_a}{d^s \varphi} = (R_y^{180^\circ} \cdot \frac{dR_{s\varphi}}{d^s \varphi} \cdot R_{s\omega} \cdot R_{s\kappa}) \cdot (R_{\omega}^T \cdot R_{\varphi}^T \cdot {}^t P_a - \begin{bmatrix} 0 \\ 0 \\ {}^s Z_{\hat{c}} \end{bmatrix}) \quad (3.29)$$

$$\frac{d^c P_a}{d^s \omega} = (R_y^{180^\circ} \cdot R_{s\varphi} \cdot \frac{dR_{s\omega}}{d^s \omega} \cdot R_{s\kappa}) \cdot (R_{\omega}^T \cdot R_{\varphi}^T \cdot {}^t P_a - \begin{bmatrix} 0 \\ 0 \\ {}^s Z_{\hat{c}} \end{bmatrix}) \quad (3.30)$$

$$\frac{d^c P_a}{d^s \kappa} = (R_y^{180^\circ} \cdot R_{s\varphi} \cdot R_{s\omega} \cdot \frac{dR_{s\kappa}}{d^s \kappa}) \cdot (R_{\omega}^T \cdot R_{\varphi}^T \cdot {}^t P_a - \begin{bmatrix} 0 \\ 0 \\ {}^s Z_{\hat{c}} \end{bmatrix}) \quad (3.31)$$



One example of singularity is that if  ${}^s\omega = +\frac{\pi}{2}$ , in equation (3.29), the derivative term in the three elementary rotation matrix

$$\begin{aligned} \frac{dR_{{}^s\varphi}}{d{}^s\varphi} \cdot R_{{}^s\omega} \cdot R_{{}^s\kappa} &= \begin{bmatrix} -S_{{}^s\varphi} & 0 & C_{{}^s\varphi} \\ 0 & 1 & 0 \\ -C_{{}^s\varphi} & 0 & -S_{{}^s\varphi} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_{{}^s\omega} & -S_{{}^s\omega} \\ 0 & S_{{}^s\omega} & C_{{}^s\omega} \end{bmatrix} \begin{bmatrix} C_{{}^s\kappa} & -S_{{}^s\kappa} & 0 \\ S_{{}^s\kappa} & C_{{}^s\kappa} & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} -S_{{}^s\varphi}C_{{}^s\kappa} + S_{{}^s\omega}C_{{}^s\varphi}S_{{}^s\kappa} & S_{{}^s\kappa}S_{{}^s\varphi} + S_{{}^s\omega}C_{{}^s\varphi}C_{{}^s\kappa} & C_{{}^s\omega}C_{{}^s\varphi} \\ C_{{}^s\omega}S_{{}^s\kappa} & C_{{}^s\omega}C_{{}^s\kappa} & -S_{{}^s\omega} \\ -C_{{}^s\varphi}C_{{}^s\kappa} - S_{{}^s\varphi}S_{{}^s\omega}S_{{}^s\kappa} & C_{{}^s\varphi}S_{{}^s\kappa} - S_{{}^s\varphi}S_{{}^s\omega}C_{{}^s\kappa} & -S_{{}^s\varphi}C_{{}^s\omega} \end{bmatrix} \quad (3.32) \end{aligned}$$

becomes

$$\frac{dR_{{}^s\varphi}}{d{}^s\varphi} \cdot R_{{}^s\omega} \cdot R_{{}^s\kappa} = \begin{bmatrix} -S_{{}^s\varphi}C_{{}^s\kappa} + C_{{}^s\varphi}S_{{}^s\kappa} & S_{{}^s\kappa}S_{{}^s\varphi} + C_{{}^s\varphi}C_{{}^s\kappa} & 0 \\ 0 & 0 & -1 \\ -C_{{}^s\varphi}C_{{}^s\kappa} - S_{{}^s\varphi}S_{{}^s\kappa} & C_{{}^s\varphi}S_{{}^s\kappa} - S_{{}^s\varphi}C_{{}^s\kappa} & 0 \end{bmatrix} = \begin{bmatrix} -S_{{}^s\varphi-{}^s\kappa} & C_{{}^s\varphi-{}^s\kappa} & 0 \\ 0 & 0 & -1 \\ -C_{{}^s\varphi-{}^s\kappa} & -S_{{}^s\varphi-{}^s\kappa} & 0 \end{bmatrix}, \quad (3.33)$$

which cannot give us a unique angle value for  ${}^s\varphi$  and  ${}^s\kappa$ , since the cofactor matrix depends on  ${}^s\varphi - {}^s\kappa$ , but not  ${}^s\varphi$  or  ${}^s\kappa$ . Therefore, the Jacobian matrix of a camera pose

$$J_{wS_e} = \frac{d{}^p\hat{P}_a}{d{}^w\hat{S}_e} = \begin{bmatrix} \frac{d{}^p\hat{P}_a}{d{}^s\varphi} & \frac{d{}^p\hat{P}_a}{d{}^s\omega} & \frac{d{}^p\hat{P}_a}{d{}^sZ_a} & \frac{d{}^p\hat{P}_a}{d{}^s\varphi} & \frac{d{}^p\hat{P}_a}{d{}^s\omega} & \frac{d{}^p\hat{P}_a}{d{}^s\kappa} \end{bmatrix} \quad (3.34)$$

has a column vector corresponding to  ${}^s\varphi$  or  ${}^s\kappa$ , which are not independent with the column vector corresponding to  ${}^s\kappa$  or  ${}^s\varphi$ , respectively. This is to say that Jacobian matrix will be **rank deficient**.

### 3.3 Singularity of Spherical Coordinate Representation

#### 3.3.1 Spherical Coordinate Representation

In geometry, we want to **uniquely** determine and represent the position of a point or an object by a representation system. For example, the number line system is used with a zero as the origin for a point in one dimensional space while the Cartesian coordinate system is used to represent positions in both two dimensional space and three dimensional space. Euler angles can be used to express orientation in three dimensional space. Thus, an object in 3D space has totally six degrees of freedom after combining position information and orientation information:

$$\begin{bmatrix} X \\ Y \\ Z \\ \omega \\ \kappa \\ \varphi \end{bmatrix} \in R^6, \quad (3.35)$$

where  $\omega$ ,  $\varphi$ , and  $\kappa$  are the rotation angles about the X, Z, Y axes, respectively.

The spherical coordinate system is applied for representation in our application. A pose in spherical coordinate systems is illustrated in figure 3.4.

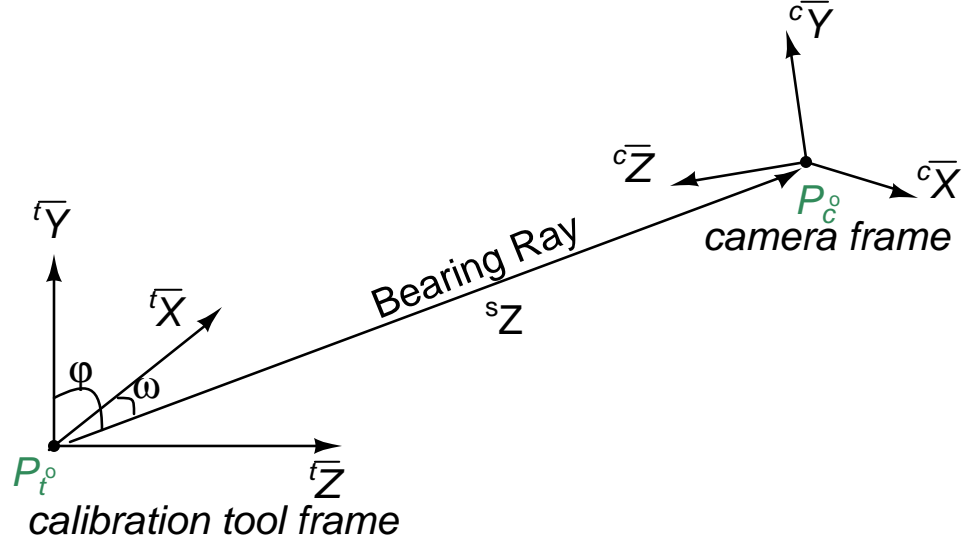


Figure 3.4: A pose in spherical calibration tool coordinates. The bearing ray connects the root frame (in this case the calibration tool frame) to the object frame (in this case the camera frame).

The spherical coordinate system is used to represent position, and Euler angles are used to represent the orientation<sup>28</sup>. If the object (digital camera) in 3D space is expressed in the spherical coordinate system, the camera pose in the calibration tool frame is expressed in the spherical coordinates as:

$${}^tS_c = \begin{bmatrix} \varphi \\ \omega \\ sZ \\ s\omega \\ s\kappa \\ s\varphi \end{bmatrix} = \begin{bmatrix} {}^tS_c \\ s\omega \\ s\kappa \\ s\varphi \end{bmatrix} \in R^6, \quad (3.36)$$

28. Euler angles representation is explained in section 2.1.1.

where

$${}^tS_c = \begin{bmatrix} \varphi \\ \omega \\ {}^sZ_c \end{bmatrix} \in R^3 \quad (3.37)$$

is the position of camera origin in the calibration tool frame expressed in spherical coordinates:  $\varphi$  is called azimuth,  $\omega$  is called elevation,  ${}^sZ_c$  is the distance between the calibration tool frame and the camera frame;  $\begin{bmatrix} {}^s\omega & {}^s\kappa & {}^s\varphi \end{bmatrix}$  expresses the orientation of the camera. A rotation matrix  ${}_s^tR = R_\varphi \cdot R_\omega$  defines the rotation from the bearing to the calibration tool frame with two elementary rotation matrices  $R_\omega$  and  $R_\varphi$ , as shown in figure 3.4. If the spherical representation is used to represent a point in target coordinates, these coordinates are referred to as spherical target coordinates. For example,  ${}^tS_c$  is a point in camera coordinates defined in target coordinates using the spherical representation. As an alternative way of representing an object in 3D space, spherical coordinate representation introduces two more angles,  $\varphi$  and  $\omega$ , compared to the expression in equation (3.35).

### 3.3.2 Frames: the $t$ Frame, the $c$ Frame, the $i$ Frame, and the $v$ Frame

As shown in figure 2.8, bundle adjustment after image processing of calibration tool images requires the conversion of the landmark in target coordinates  ${}^tP_a$  into the landmark in camera coordinates  ${}^cP_a$  and then to the landmark in image coordinates  ${}^iP_a$ . Here, three landmarks in the target coordinate frame are selected:

$${}^tP_{L1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, {}^tP_{L2} = \begin{bmatrix} l \\ 0 \\ 0 \end{bmatrix}, {}^tP_{L3} = \begin{bmatrix} 0 \\ l \\ 0 \end{bmatrix} \quad (3.38)$$

Then ,  ${}^iP_a$  is converted to  ${}^cP_a$  by

$$\begin{aligned} {}^cP_a &= {}^cR \cdot {}^iP_a + {}^cP_i \\ &= (R_y^{180^\circ} \cdot R_{s_\varphi} \cdot R_{s_\omega} \cdot R_{s_\kappa})(R_\omega^T \cdot R_\varphi^T) \cdot {}^iP_a + {}^cP_i, \end{aligned} \quad (3.39)$$

where

$${}^cR = (R_y^{180^\circ} \cdot R_{s_\varphi} \cdot R_{s_\omega} \cdot R_{s_\kappa}). \quad (3.40)$$

Last,  ${}^cP_a$  is converted to  ${}^iP_a$  by

$${}^iP_a = -c_p \cdot \begin{bmatrix} \frac{{}^cP_a(1)}{{}^cP_a(3)} \\ \frac{{}^cP_a(2)}{{}^cP_a(3)} \end{bmatrix}. \quad (3.41)$$

An additional frame: the  $v$  frame is introduced for searching for possible singularities in spherical coordinate system,. The  $v$  frame gives a special structure in the Jacobian matrix. Accordingly, there is a space that contains the projection of landmarks represented in the  $i$  frame onto the  $v$  frame. All effective landmarks in digital images of the calibration tool can be represented in the  $v$  frame. The geometry is shown in figure 3.5.

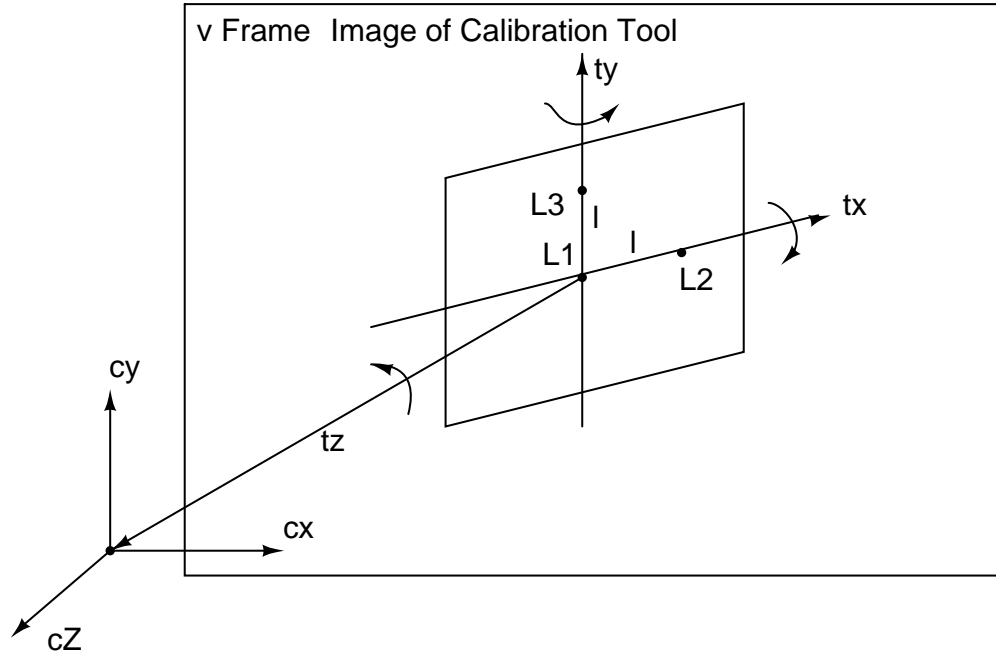


Figure 3.5: The  $t$  frame and its axes are on the calibration tool. The  $v$  frame and its axes are in the image of calibration tool, which is defined by three landmarks defined by the image of three landmarks  ${}^tP_{L_j}$ :  $L_1$  is the origin;  $L_1$  and  $L_2$  give us  $t_x$ ;  $L_1$  and  $L_3$  give us  $t_y$ ;  $t_z$  is perpendicular to  $t_x$  and  $t_y$  by  $t_z = \text{cross}(t_x, t_y)$ .

In figure 3.5, it is clear that  $L_1$  is the origin of the new frame;  $L_1$  and  $L_2$  give us  $t_x$  axis;  $L_1$  and  $L_3$  give us  $t_y$  axis;  $t_z$  axis is the perpendicular direction to  $t_x$  and  $t_y$  by  $t_z = \text{cross}(t_x, t_y)$ . It is important to remember the landmark position in the  $v$  frame is based on the transformation and projection from landmarks in the  $t$  frame due to rotation movement of the calibration tool in 3D space. Thus, the image of  ${}^tP_{L_1}$  in the  $v$  frame is the origin of the  $v$  frame. The image of  ${}^tP_{L_2}$  in the  $v$  frame is on the  $v_x$  axis which is the image of the  $t_x$  axis in the  $v$  frame. The image of  ${}^tP_{L_3}$  in the  $v$  frame is on the  $v_y$  axis which is the image of the  $t_y$  axis in the  $v$  frame. The position change of  ${}^tP_{L_j}$  in the  $v$  frame is done by moving calibration tool during the data collection stage. Note that the  $v_x$  and  $v_y$  axes will not in general be orthogonal.

The  ${}^iP_a$  obtained from  ${}^tP_a$  are:

$${}^iP_{L_1} = \begin{bmatrix} {}^iX_{L_1} \\ {}^iY_{L_1} \end{bmatrix}, \quad {}^iP_{L_2} = \begin{bmatrix} {}^iX_{L_2} \\ {}^iY_{L_2} \end{bmatrix}, \quad {}^iP_{L_3} = \begin{bmatrix} {}^iX_{L_3} \\ {}^iY_{L_3} \end{bmatrix}, \quad (3.42)$$

so the vectors of  $v_1$  and  $v_2$  directions are:

$$v_1 = {}^iP_{L_2} - {}^iP_{L_1} = \begin{bmatrix} {}^iX_{L_2} \\ {}^iY_{L_2} \end{bmatrix} - \begin{bmatrix} {}^iX_{L_1} \\ {}^iY_{L_1} \end{bmatrix} \quad (3.43)$$

$$v_2 = {}^iP_{L_3} - {}^iP_{L_1} = \begin{bmatrix} {}^iX_{L_3} \\ {}^iY_{L_3} \end{bmatrix} - \begin{bmatrix} {}^iX_{L_1} \\ {}^iY_{L_1} \end{bmatrix} \quad (3.44)$$

By normalization, the normalized directions are used as the basis of the  $v$  frame:

$$v_1^N = \frac{v_1}{||v_1||} = \begin{bmatrix} \frac{{}^iX_{L_2} - {}^iX_{L_1}}{||{}^i v_1||} \\ \frac{{}^iY_{L_2} - {}^iY_{L_1}}{||{}^i v_1||} \end{bmatrix} \quad (3.45)$$

$$v_2^N = \frac{v_2}{||v_2||} = \begin{bmatrix} \frac{{}^iX_{L_3} - {}^iX_{L_1}}{||{}^i v_2||} \\ \frac{{}^iY_{L_3} - {}^iY_{L_1}}{||{}^i v_2||} \end{bmatrix} \quad (3.46)$$

which can be written in matrix form:

$${}_i^v m = \begin{bmatrix} v_1^N & v_2^N \end{bmatrix}^{-1} = \begin{bmatrix} \frac{{}^iX_{L_2} - {}^iX_{L_1}}{||{}^i v_1||} & \frac{{}^iX_{L_3} - {}^iX_{L_1}}{||{}^i v_2||} \\ \frac{{}^iY_{L_2} - {}^iY_{L_1}}{||{}^i v_1||} & \frac{{}^iY_{L_3} - {}^iY_{L_1}}{||{}^i v_2||} \end{bmatrix}^{-1} = \begin{bmatrix} m_{11} & m_{21} \\ m_{12} & m_{22} \end{bmatrix} \quad (3.47)$$

Then the transformation from the image frame to the  $v$  frame is performed by matrix:

$${}^v_i M = \begin{bmatrix} m_{11} & m_{21} & 0 & 0 & 0 & 0 \\ m_{12} & m_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & m_{11} & m_{21} & 0 & 0 \\ 0 & 0 & m_{12} & m_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 & m_{11} & m_{21} \\ 0 & 0 & 0 & 0 & m_{12} & m_{22} \end{bmatrix} \quad (3.48)$$

Since the  $v_1^N$  and  $v_2^N$  are the basis vectors, the matrix  ${}^v_i m$  is full rank, and thus, the matrix  ${}^v_i M$  is full rank. This allows us to interpret the Jacobian matrix  $J_{w_{S_{\hat{e}}}}$  in terms of the  $v$  frame shown in figure 3.5.

### 3.3.3 Structure of $J_{w_{S_{\hat{e}}}}$ in the $v$ Frame

#### The Transformation of the Jacobian Matrix

The  $J_{w_{S_{\hat{e}}}}$  is represented in the  $i$  frame as

$$J_{w_{S_{\hat{e}}}} = \begin{bmatrix} \frac{d^i P_{L1}}{d^w S_{\hat{e}}} \\ \frac{d^i P_{L2}}{d^w S_{\hat{e}}} \\ \frac{d^i P_{L2}}{d^w S_{\hat{e}}} \\ \frac{d^i P_{L2}}{d^w S_{\hat{e}}} \end{bmatrix} = \begin{bmatrix} \frac{d^i X_{L1}}{d\varphi} & \frac{d^i X_{L1}}{d\omega} & \frac{d^i X_{L1}}{d^s Z} & \frac{d^i X_{L1}}{d^s \omega} & \frac{d^i X_{L1}}{d^s \kappa} & \frac{d^i X_{L1}}{d^s \varphi} \\ \frac{d^i Y_{L1}}{d\varphi} & \frac{d^i Y_{L1}}{d\omega} & \frac{d^i Y_{L1}}{d^s Z} & \frac{d^i Y_{L1}}{d^s \omega} & \frac{d^i Y_{L1}}{d^s \kappa} & \frac{d^i Y_{L1}}{d^s \varphi} \\ \frac{d^i X_{L2}}{d\varphi} & \frac{d^i X_{L2}}{d\omega} & \frac{d^i X_{L2}}{d^s Z} & \frac{d^i X_{L2}}{d^s \omega} & \frac{d^i X_{L2}}{d^s \kappa} & \frac{d^i X_{L2}}{d^s \varphi} \\ \frac{d^i Y_{L2}}{d\varphi} & \frac{d^i Y_{L2}}{d\omega} & \frac{d^i Y_{L2}}{d^s Z} & \frac{d^i Y_{L2}}{d^s \omega} & \frac{d^i Y_{L2}}{d^s \kappa} & \frac{d^i Y_{L2}}{d^s \varphi} \\ \frac{d^i X_{L3}}{d\varphi} & \frac{d^i X_{L3}}{d\omega} & \frac{d^i X_{L3}}{d^s Z} & \frac{d^i X_{L3}}{d^s \omega} & \frac{d^i X_{L2}}{d^s \kappa} & \frac{d^i X_{L3}}{d^s \varphi} \\ \frac{d^i Y_{L3}}{d\varphi} & \frac{d^i Y_{L3}}{d\omega} & \frac{d^i Y_{L3}}{d^s Z} & \frac{d^i Y_{L3}}{d^s \omega} & \frac{d^i Y_{L3}}{d^s \kappa} & \frac{d^i Y_{L3}}{d^s \varphi} \end{bmatrix} \quad (3.49)$$

The basis transformation gives us



$$\begin{aligned}
{}^v J_{wS_{\hat{c}}} &= {}^v_i M \cdot J_{wS_{\hat{c}}} \\
&= \begin{bmatrix} \frac{d^v X_{L1}}{d\phi} & \frac{d^v X_{L1}}{d\omega} & \frac{d^v X_{L1}}{d^s Z} & \frac{d^v X_{L1}}{d^s \omega} & \frac{d^v X_{L1}}{d^s \kappa} & \frac{d^v X_{L1}}{d^s \phi} \\ \frac{d^v Y_{L1}}{d\phi} & \frac{d^v Y_{L1}}{d\omega} & \frac{d^v Y_{L1}}{d^s Z} & \frac{d^v Y_{L1}}{d^s \omega} & \frac{d^v Y_{L1}}{d^s \kappa} & \frac{d^v Y_{L1}}{d^s \phi} \\ \frac{d^v X_{L2}}{d\phi} & \frac{d^v X_{L2}}{d\omega} & \frac{d^v X_{L2}}{d^s Z} & \frac{d^v X_{L2}}{d^s \omega} & \frac{d^v X_{L2}}{d^s \kappa} & \frac{d^v X_{L2}}{d^s \phi} \\ \frac{d^v Y_{L2}}{d\phi} & \frac{d^v Y_{L2}}{d\omega} & \frac{d^v Y_{L2}}{d^s Z} & \frac{d^v Y_{L2}}{d^s \omega} & \frac{d^v Y_{L2}}{d^s \kappa} & \frac{d^v Y_{L2}}{d^s \phi} \\ \frac{d^v X_{L3}}{d\phi} & \frac{d^v X_{L3}}{d\omega} & \frac{d^v X_{L3}}{d^s Z} & \frac{d^v X_{L3}}{d^s \omega} & \frac{d^v X_{L3}}{d^s \kappa} & \frac{d^v X_{L3}}{d^s \phi} \\ \frac{d^v Y_{L3}}{d\phi} & \frac{d^v Y_{L3}}{d\omega} & \frac{d^v Y_{L3}}{d^s Z} & \frac{d^v Y_{L3}}{d^s \omega} & \frac{d^v Y_{L3}}{d^s \kappa} & \frac{d^v Y_{L3}}{d^s \phi} \end{bmatrix} \quad (3.50)
\end{aligned}$$

To start the transformation of the Jacobian matrix to the  $v$  frame, we need convert  ${}^t P_a$  to  ${}^c P_a$  to  ${}^i P_a$  and see what the Jacobian matrix is. Correspondingly, there is<sup>29</sup>

$$\begin{aligned}
{}^c P_{L_j} &= {}^c R \cdot {}^t P_{L_j} + {}^c P_i \\
&= {}^c_s R \cdot {}^s_t R \cdot {}^t P_{L_j} + {}^c P_i \\
&= {}^c_s R \left( R_{\omega}^T \cdot R_{\phi}^T \cdot {}^t P_{L_j} - \begin{bmatrix} 0 \\ 0 \\ {}^s Z_{\hat{c}} \end{bmatrix} \right), \quad (3.51)
\end{aligned}$$

where

$${}^c_s R = \begin{bmatrix} -(C_{s\phi} C_{s\kappa} + S_{s\omega} S_{s\phi} S_{s\kappa}) & S_{s\kappa} C_{s\phi} - S_{s\omega} S_{s\phi} C_{s\kappa} & -C_{s\omega} S_{s\phi} \\ C_{s\omega} S_{s\kappa} & C_{s\omega} C_{s\kappa} & -S_{s\omega} \\ S_{s\phi} C_{s\kappa} - C_{s\phi} S_{s\omega} S_{s\kappa} & -(S_{s\phi} S_{s\kappa} + C_{s\phi} S_{s\omega} C_{s\kappa}) & -C_{s\phi} C_{s\omega} \end{bmatrix} \quad (3.52)$$

---

29. Note that the equations are obtained by assuming  ${}^t Z_a = 0$  for each landmark.

$${}^sR = R_{\omega}^T \cdot R_{\varphi}^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_{\omega} & S_{\omega} \\ 0 & -S_{\omega} & C_{\omega} \end{bmatrix} \begin{bmatrix} C_{\varphi} & 0 & -S_{\varphi} \\ 0 & 1 & 0 \\ S_{\varphi} & -S_{\omega} & C_{\varphi} \end{bmatrix} = \begin{bmatrix} C_{\varphi} & 0 & -S_{\varphi} \\ S_{\omega}S_{\varphi} & C_{\omega} & S_{\omega}C_{\varphi} \\ C_{\omega}S_{\varphi} & -S_{\omega} & C_{\omega}C_{\varphi} \end{bmatrix}. \quad (3.53)$$

Substituting landmarks  $L_1$ ,  $L_2$ , and  $L_3$ , the first landmark,  ${}^iP_{L_1}$ , becomes

$${}^iP_{L_1} = -c_p \cdot \frac{{}^cP_{L_1}(1:2)}{{}^cP_{L_1}(3)} = -c_p \begin{bmatrix} \frac{S_{s_{\varphi}}}{C_{s_{\varphi}}} \\ \frac{S_{s_{\omega}}}{C_{s_{\varphi}}C_{s_{\omega}}} \end{bmatrix}, \quad (3.54)$$

where

$$\begin{aligned} {}^cP_{L_1} &= {}^sR \begin{bmatrix} 0 \\ 0 \\ -{}^sZ_{\hat{c}} \end{bmatrix} \\ &= \begin{bmatrix} -(C_{s_{\varphi}}C_{s_{\kappa}} + S_{s_{\omega}}S_{s_{\varphi}}S_{s_{\kappa}}) & S_{s_{\kappa}}C_{s_{\varphi}} - S_{s_{\omega}}S_{s_{\varphi}}C_{s_{\kappa}} & -C_{s_{\omega}}S_{s_{\varphi}} \\ C_{s_{\omega}}S_{s_{\kappa}} & C_{s_{\omega}}C_{s_{\kappa}} & -S_{s_{\omega}} \\ S_{s_{\varphi}}C_{s_{\kappa}} - C_{s_{\varphi}}S_{s_{\omega}}S_{s_{\kappa}} & -(S_{s_{\varphi}}S_{s_{\kappa}} + C_{s_{\varphi}}S_{s_{\omega}}C_{s_{\kappa}}) & -C_{s_{\varphi}}C_{s_{\omega}} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -{}^sZ_{\hat{c}} \end{bmatrix} \\ &= \begin{bmatrix} C_{s_{\omega}}S_{s_{\varphi}}{}^sZ_{\hat{c}} \\ S_{s_{\omega}}{}^sZ_{\hat{c}} \\ C_{s_{\varphi}}C_{s_{\omega}}{}^sZ_{\hat{c}} \end{bmatrix}. \end{aligned} \quad (3.55)$$

Next, the second landmark,  ${}^iP_{L_2}$ , is

$${}^iP_{L_2} = -c_p \cdot \frac{{}^cP_{L_2}(1:2)}{{}^cP_{L_2}(3)} = -c_p \cdot \begin{bmatrix} \frac{{}^cP_{L_2}(1)}{{}^cP_{L_2}(3)} \\ \frac{{}^cP_{L_2}(2)}{{}^cP_{L_2}(3)} \end{bmatrix}, \quad (3.56)$$

where

$$\begin{aligned}
{}^cP_{L_2} &= {}^c_sR \left( R_{\omega}^T \cdot R_{\varphi}^T \cdot \begin{bmatrix} l \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ {}^sZ_{\hat{c}} \end{bmatrix} \right) \\
&= {}^c_sR \left( \begin{bmatrix} C_{\varphi} & 0 & -S_{\varphi} \\ S_{\omega}S_{\varphi} & C_{\omega} & S_{\omega}C_{\varphi} \\ C_{\omega}S_{\varphi} & -S_{\omega} & C_{\omega}C_{\varphi} \end{bmatrix} \cdot \begin{bmatrix} l \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ {}^sZ_{\hat{c}} \end{bmatrix} \right) \\
&= {}^c_sR \begin{bmatrix} C_{\varphi} \cdot l \\ S_{\omega}S_{\varphi} \cdot l \\ C_{\omega}S_{\varphi} \cdot l - {}^sZ_{\hat{c}} \end{bmatrix}.
\end{aligned} \tag{3.57}$$

Then, the third landmark,  ${}^iP_{L_3}$ , is

$${}^iP_{L_3} = -c_p \cdot \frac{{}^cP_{L_3}(1:2)}{{}^cP_{L_3}(3)} = -c_p \begin{bmatrix} \frac{{}^cP_{L_3}(1)}{{}^cP_{L_3}(3)} \\ \frac{{}^cP_{L_3}(2)}{{}^cP_{L_3}(3)} \end{bmatrix}, \tag{3.58}$$

where

$$\begin{aligned}
{}^cP_{L_3} &= {}^c_sR \left( R_{\omega}^T \cdot R_{\varphi}^T \cdot \begin{bmatrix} 0 \\ l \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ {}^sZ_{\hat{c}} \end{bmatrix} \right) \\
&= {}^c_sR \left( \begin{bmatrix} C_{\varphi} & 0 & -S_{\varphi} \\ S_{\omega}S_{\varphi} & C_{\omega} & S_{\omega}C_{\varphi} \\ C_{\omega}S_{\varphi} & -S_{\omega} & C_{\omega}C_{\varphi} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ l \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ {}^sZ_{\hat{c}} \end{bmatrix} \right) \\
&= {}^c_sR \begin{bmatrix} 0 \\ C_{\omega} \cdot l \\ -S_{\omega} \cdot l - {}^sZ_{\hat{c}} \end{bmatrix}.
\end{aligned} \tag{3.59}$$

### The Structure of the Jacobian Matrix

To simplify the problem, consider the case of  ${}^s\varphi = 0^\circ$ ,  ${}^s\omega = 0^\circ$ , and  ${}^s\kappa = 0^\circ$ , which gives us:

$${}^cR = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}. \quad (3.60)$$

This simplification shows us the plane  $I_0$  in figure 3.6. Other possible angle values of  ${}^s\varphi$ ,  ${}^s\omega$ , and  ${}^s\kappa$  in  ${}^cR$  will tilt the image of calibration tool in 3D space, which is shown as the plane  $I_1, \dots, I_k$  in figure 3.6.

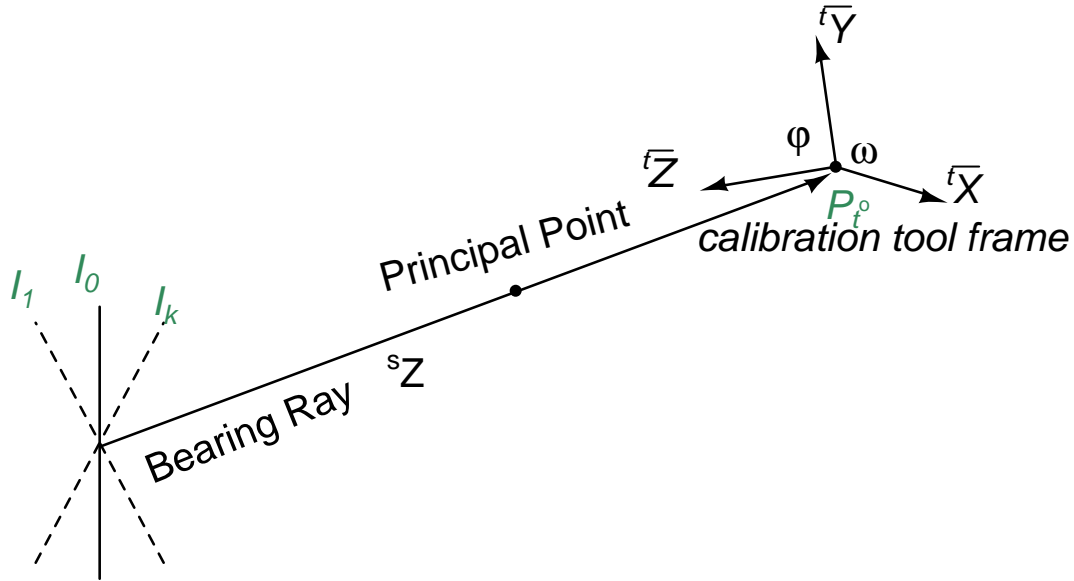


Figure 3.6: A pose in spherical calibration tool coordinates and its projection on to the  $v$  frame:  $I_0$  is the nominal plane based on  ${}^s\varphi = 0^\circ$ ,  ${}^s\omega = 0^\circ$ , and  ${}^s\kappa = 0^\circ$ ;  $I_1, \dots, I_k$  is the planes based on other possible  ${}^s\varphi$ ,  ${}^s\omega$ , and  ${}^s\kappa$  values.

Because the range of tilt of  $I_1, \dots, I_k$  is restricted<sup>30</sup>, the mapping of  $I_0$  to  $I_1, \dots, I_k$  is a **one-to-one projection**, and the result of the above simplified case on plane  $I_0$  will also be true to other cases on plane  $I_1, \dots, I_k$ .

Representing the Jacobian matrix in the  $v$  frame and assigning  ${}^s\varphi = 0^\circ$ ,  ${}^s\omega = 0^\circ$ , and

30. The FOV of a camera will limit the acceptable tilting angles to obtain usable digital images.

${}^s\kappa = 0^\circ$  will offer a special structure in the Jacobian matrix  ${}^vJ_{wS_\varepsilon}$ . Mathematically, the straightforward calculation of equations gives us the Jacobian matrix  ${}^vJ_{wS_\varepsilon}$  as:

$$\begin{aligned}
 {}^vJ_{wS_\varepsilon} &= \begin{bmatrix} \frac{d{}^vP_{L_j}}{d\varphi} & \frac{d{}^vP_{L_j}}{d\omega} & \frac{d{}^vP_{L_j}}{d{}^sZ} & \frac{d{}^vP_{L_j}}{d{}^s\omega} & \frac{d{}^vP_{L_j}}{d{}^s\kappa} & \frac{d{}^vP_{L_j}}{d{}^s\varphi} \end{bmatrix} \\
 &= \begin{bmatrix} 0 & 0 & 0 & \frac{d{}^vX_{L_1}}{d{}^s\omega} & 0 & \frac{d{}^vX_{L_1}}{d{}^s\varphi} \\ 0 & 0 & 0 & \frac{d{}^vY_{L_1}}{d{}^s\omega} & 0 & \frac{d{}^vY_{L_1}}{d{}^s\varphi} \\ \frac{d{}^vX_{L_2}}{d\varphi} & \frac{d{}^vX_{L_2}}{d\omega} & \frac{d{}^vX_{L_2}}{d{}^sZ} & \frac{d{}^vX_{L_2}}{d{}^s\omega} & \frac{d{}^vX_{L_2}}{d{}^s\kappa} & \frac{d{}^vX_{L_2}}{d{}^s\varphi} \\ \frac{d{}^vY_{L_2}}{d\varphi} & \frac{d{}^vY_{L_2}}{d\omega} & 0 & \frac{d{}^vY_{L_2}}{d{}^s\omega} & \frac{d{}^vY_{L_2}}{d{}^s\kappa} & \frac{d{}^vY_{L_2}}{d{}^s\varphi} \\ 0 & 0 & 0 & \frac{d{}^vX_{L_3}}{d{}^s\omega} & \frac{d{}^vX_{L_3}}{d{}^s\kappa} & \frac{d{}^vX_{L_3}}{d{}^s\varphi} \\ 0 & \frac{d{}^vY_{L_3}}{d\omega} & \frac{d{}^vY_{L_3}}{d{}^sZ} & \frac{d{}^vY_{L_3}}{d{}^s\omega} & \frac{d{}^vY_{L_3}}{d{}^s\kappa} & \frac{d{}^vY_{L_3}}{d{}^s\varphi} \end{bmatrix}. \tag{3.61}
 \end{aligned}$$

Substituting equations (3.54), (3.56), and (3.58) into equation (3.50), the structure in the above equation is found. First, equation (3.54) does not contain variables  $\varphi$ ,  $\omega$ ,  ${}^sZ$ , and  ${}^s\kappa$ , so the first two rows of  ${}^vJ_{wS_\varepsilon}$  have eight zeros in corresponding spots. Next, if we derive equations for  $L_1$  in terms of  $\frac{d{}^vP_{L_j}}{d{}^s\omega}$  and  $\frac{d{}^vP_{L_j}}{d{}^s\varphi}$ , there are

$$\frac{d{}^vP_{L_1}}{d{}^s\omega} = {}^v_i m \cdot \frac{d{}^iP_{L_1}}{d{}^s\omega} = -c_p \cdot {}^v_i m \cdot \begin{bmatrix} 0 \\ \frac{1}{C_{s_\omega}^2 C_{s_\varphi}} \end{bmatrix} = -c_p \cdot {}^v_i m \cdot \begin{bmatrix} 0 \\ q \end{bmatrix} \tag{3.62}$$

and

$$\frac{d{}^vP_{L_1}}{d{}^s\varphi} = {}^v_i m \cdot \frac{d{}^iP_{L_1}}{d{}^s\varphi} = -c_p \cdot {}^v_i m \cdot \begin{bmatrix} \frac{1}{C_{s_\varphi}^2} \\ \frac{S_{s_\varphi} S_{s_\omega}}{C_{s_\omega} C_{s_\varphi}^2} \end{bmatrix} = -c_p \cdot {}^v_i m \cdot \begin{bmatrix} r \\ w \end{bmatrix}, \tag{3.63}$$

where  ${}^v_i m$  is a full rank  $2 \times 2$  matrix,

$$q = \frac{1}{C_{s_\omega}^2 C_{s_\varphi}} \tag{3.64}$$

and

$$\begin{bmatrix} r \\ w \end{bmatrix} = \begin{bmatrix} \frac{1}{C_{s\phi}^2} \\ \frac{S_{s\phi}S_{s\omega}}{C_{s\omega}C_{s\phi}^2} \end{bmatrix}. \quad (3.65)$$

Accordingly, the determinant of  ${}^v J_{w_{S_{\hat{e}}}}$  is:

$$\begin{aligned} \det({}^v J_{w_{S_{\hat{e}}}}) &= \frac{d^v X_{L1}}{d^s \phi} \cdot \frac{d^v Y_{L1}}{d^s \omega} \cdot \begin{bmatrix} \frac{d^v X_{L2}}{d\phi} & \frac{d^v X_{L2}}{d\omega} & \frac{d^v X_{L2}}{d^s Z} & \frac{d^v X_{L2}}{d^s \kappa} \\ \frac{d^v Y_{L2}}{d\phi} & \frac{d^v Y_{L2}}{d\omega} & 0 & \frac{d^v Y_{L2}}{d^s \kappa} \\ 0 & 0 & 0 & \frac{d^v X_{L3}}{d^s \kappa} \\ 0 & \frac{d^v Y_{L3}}{d\omega} & \frac{d^v Y_{L3}}{d^s Z} & \frac{d^v Y_{L3}}{d^s \kappa} \end{bmatrix} \\ &= q \cdot r \cdot \det \left( \begin{bmatrix} a & b & c & d \\ e & f & 0 & g \\ 0 & 0 & 0 & k \\ 0 & n & o & p \end{bmatrix} \right), \end{aligned} \quad (3.66)$$

where letters are used to simplify the representation of each term as  $a = \frac{d^v X_{L2}}{d\phi}$ ,  $b = \frac{d^v X_{L2}}{d\omega}$ ,  $c = \frac{d^v X_{L2}}{d^s Z}$ ,  $d = \frac{d^v X_{L2}}{d^s \kappa}$ ,  $e = \frac{d^v Y_{L2}}{d\phi}$ ,  $f = \frac{d^v Y_{L2}}{d\omega}$ ,  $g = \frac{d^v Y_{L2}}{d^s \kappa}$ ,  $k = \frac{d^v X_{L3}}{d^s \kappa}$ ,  $n = \frac{d^v Y_{L3}}{d\omega}$ ,  $o = \frac{d^v Y_{L3}}{d^s Z}$ , and  $p = \frac{d^v Y_{L3}}{d^s \kappa}$  again.

Then, the equations of term  $a$  and  $e$  are:

$$\begin{aligned} \begin{bmatrix} a \\ e \end{bmatrix} &= \frac{d^v P_{L2}}{d\phi} = {}^v_i m \cdot \frac{d^i P_{L2}}{d\phi} \\ &= -c_p \cdot {}^v_i m \cdot \left( \frac{1}{c P_a(3)} \frac{d^c P_a(1:2)}{d\phi} - \left( \frac{c P_a(1:2)}{c P_a^2(3)} \right) \frac{d^c P_a(3)}{d\phi} \right) \\ &= -c_p \cdot {}^v_i m \cdot \begin{bmatrix} \frac{l \cdot C_{\omega} - {}^s Z \cdot S_{\phi}}{Z_2} \\ \frac{(l \cdot S_{\omega} + {}^s Z) \cdot S_{\omega}}{Z_2 \cdot C_{\phi} \cdot C_{\omega}} \end{bmatrix}, \end{aligned} \quad (3.67)$$

where

$$\begin{aligned}
 \frac{d^c P_{L_2}}{d\phi} &= {}^c_s R \cdot R_{\omega}^T \cdot \frac{dR_{\phi}^T}{d\phi} \cdot {}^t P_a \\
 &= {}^c_s R \left( \begin{bmatrix} -S_{\phi} & 0 & -C_{\phi} \\ S_{\omega} C_{\phi} & 0 & -S_{\omega} S_{\phi} \\ C_{\omega} C_{\phi} & 0 & -C_{\omega} S_{\phi} \end{bmatrix} \cdot \begin{bmatrix} l \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ {}^s Z_{\hat{c}} \end{bmatrix} \right) \\
 &= {}^c_s R \cdot \begin{bmatrix} -S_{\phi} \cdot l \\ S_{\omega} C_{\phi} \cdot l \\ C_{\omega} C_{\phi} \cdot l \end{bmatrix}
 \end{aligned} \tag{3.68}$$

and

$$Z_2 = {}^s Z - l \cdot C_{\omega} \cdot S_{\phi}. \tag{3.69}$$

The equations of term  $b$  and  $f$  are

$$\begin{aligned}
 \begin{bmatrix} b \\ f \end{bmatrix} &= \frac{d^v P_{L_2}}{d\omega} = {}^v_i m \cdot \frac{d^i P_{L_2}}{d\omega} \\
 &= -c_p \cdot {}^v_i m \cdot \left( \frac{1}{{}^c P_a(3)} \frac{d^c P_a(1:2)}{d\omega} - \left( \frac{{}^c P_a(1:2)}{{}^c P_a^2(3)} \right) \frac{d^c P_a(3)}{d\omega} \right) \\
 &= -c_p \cdot {}^v_i m \cdot \begin{bmatrix} -\frac{l \cdot S_{\phi} S_{\omega}}{Z_2} \\ \frac{(l \cdot S_{\omega} + {}^s Z) \cdot S_{\phi}}{Z_2} \end{bmatrix},
 \end{aligned} \tag{3.70}$$

where

$$\begin{aligned}
 \frac{d^c P_{L_2}}{d\omega} &= {}^cR \cdot \frac{dR_{\omega}^T}{d\omega} \cdot R_{\varphi}^T \cdot {}^tP_a \\
 &= {}^cR \cdot \left( \begin{bmatrix} 0 & 0 & 0 \\ C_{\omega}S_{\varphi} & -S_{\omega} & C_{\omega}C_{\varphi} \\ -S_{\omega}S_{\varphi} & -C_{\omega} & -S_{\omega}C_{\varphi} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ l \\ 0 \end{bmatrix} \right) \\
 &= {}^cR \cdot \begin{bmatrix} 0 \\ C_{\omega}S_{\varphi} \cdot l \\ -S_{\omega}S_{\varphi} \cdot l \end{bmatrix}.
 \end{aligned} \tag{3.71}$$

The term  $c$  is as below

$$\begin{aligned}
 \begin{bmatrix} c \\ 0 \end{bmatrix} &= \frac{d^v P_{L_2}}{d^s Z} = {}^v_i m \cdot \frac{d^i P_{L_2}}{d^s Z} \\
 &= -c_p \cdot {}^v_i m \cdot \left( \frac{1}{cP_a(3)} \frac{d^c P_a(1:2)}{d^s Z} - \left( \frac{cP_a(1:2)}{cP_a^2(3)} \right) \frac{d^c P_a(3)}{d^s Z} \right) \\
 &= -c_p \cdot {}^v_i m \cdot \begin{bmatrix} -\frac{1}{Z_2} \\ 0 \end{bmatrix},
 \end{aligned} \tag{3.72}$$

where

$$\frac{d^c P_{L_2}}{d^s Z} = -{}^cR \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -C_{\omega}S_{\varphi} \\ -S_{\omega} \\ -C_{\varphi}C_{\omega} \end{bmatrix}. \tag{3.73}$$

It is easy to write

$$\frac{d^v P_{L_3}}{d\varphi} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{3.74}$$



Term  $n$  and  $o$  are:

$$\begin{aligned}
 \begin{bmatrix} 0 \\ n \end{bmatrix} &= \frac{d^v P_{L_3}}{d\omega} = {}^v_i m \cdot \frac{d^i P_{L_3}}{d\omega} \\
 &= -c_p \cdot {}^v_i m \cdot \left( \frac{1}{c P_a(3)} \frac{d^c P_a(1:2)}{d\omega} - \left( \frac{{}^c P_a(1:2)}{{}^c P_a^2(3)} \right) \frac{d^c P_a(3)}{d\omega} \right) \\
 &= -c_p \cdot {}^v_i m \cdot \begin{bmatrix} 0 \\ \frac{l + {}^s Z \cdot S_\omega}{(l \cdot S_\omega + {}^s Z) \cdot C_\omega} \end{bmatrix}, \tag{3.75}
 \end{aligned}$$

where

$$\frac{d^c P_{L_2}}{d\omega} = {}^c_s R \cdot \begin{bmatrix} 0 \\ -S_\omega \cdot l \\ -C_\omega \cdot l \end{bmatrix}, \tag{3.76}$$

and

$$\begin{aligned}
 \begin{bmatrix} 0 \\ o \end{bmatrix} &= \frac{d^v P_{L_3}}{d^s Z} = {}^v_i m \cdot \frac{d^i P_{L_3}}{d\omega} \\
 &= -c_p \cdot {}^v_i m \cdot \left( \frac{1}{c P_a(3)} \frac{d^c P_a(1:2)}{d^s Z} - \left( \frac{{}^c P_a(1:2)}{{}^c P_a^2(3)} \right) \frac{d^c P_a(3)}{d^s Z} \right) \\
 &= -c_p \cdot {}^v_i m \cdot \begin{bmatrix} 0 \\ -\frac{1}{l \cdot S_\omega + {}^s Z} \end{bmatrix}, \tag{3.77}
 \end{aligned}$$

where

$$\frac{d^c P_{L_3}}{d^s Z} = {}^c_s R \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -C_{\omega} S_{\varphi} \\ -S_{\omega} \\ -C_{\varphi} C_{\omega} \end{bmatrix}. \tag{3.78}$$

Last, the equation of term  $k$  is

$$\begin{aligned}
 \begin{bmatrix} k \\ - \end{bmatrix} &= \frac{d^v P_{L_3}}{d^s \kappa} = {}^v_i m \cdot \frac{d^i P_{L_3}}{d^s \kappa} \\
 &= -c_p \cdot {}^v_i m \cdot \left( \frac{1}{c P_a(3)} \frac{d^c P_a(1:2)}{d^s \kappa} - \left( \frac{c P_a(1:2)}{c P_a^2(3)} \right) \frac{d^c P_a(3)}{d \text{Roll}} \right) \\
 &= -c_p \cdot {}^v_i m \cdot \begin{bmatrix} -\frac{C_\omega \cdot Z_2}{(l \cdot S_\omega + s Z) \cdot C_\phi} \\ - \end{bmatrix}, \tag{3.79}
 \end{aligned}$$

where  $\frac{d^c P_{L_3}}{d^s \kappa}$  is shown in equation (3.31).

Equation (3.66) allows us consider the determinant of a  $4 \times 4$  matrix as:

$$\begin{aligned}
 {}^v J_{reduced} &= \begin{bmatrix} \frac{d^v P_{L_j}}{d\phi} & \frac{d^v P_{L_j}}{d\omega} & \frac{d^v P_{L_j}}{d^s Z} & \frac{d^v P_{L_j}}{d^s \kappa} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{d^v X_{L_2}}{d\phi} & \frac{d^v X_{L_2}}{d\omega} & \frac{d^v X_{L_2}}{d^s Z} & \frac{d^v X_{L_2}}{d^s \kappa} \\ \frac{d^v Y_{L_2}}{d\phi} & \frac{d^v Y_{L_2}}{d\omega} & 0 & \frac{d^v Y_{L_2}}{d^s \kappa} \\ 0 & 0 & 0 & \frac{d^v X_{L_3}}{d^s \kappa} \\ 0 & \frac{d^v Y_{L_3}}{d\omega} & \frac{d^v Y_{L_3}}{d^s Z} & \frac{d^v Y_{L_3}}{d^s \kappa} \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & 0 & g \\ 0 & 0 & 0 & k \\ 0 & n & o & p \end{bmatrix} \tag{3.80}
 \end{aligned}$$

instead of a  $6 \times 6$  matrix in equation (3.61).

### 3.3.4 Rank of ${}^v J_{w_{S_{\hat{c}}}}$

Next, calculating the matrix determinant of equation (3.80), we obtain

$$\det({}^v J_{reduced}) = -k \times (a \cdot f \cdot o + c \cdot e \cdot n - b \cdot e \cdot o) \tag{3.81}$$

By substituting equations (3.67), (3.70), (3.72), (3.75), (3.77), and (3.79) from section 3.3.3 into equation (3.81), we obtain

$$\det({}^v J_{reduced}) = \frac{-\left\{ \left( C_\varphi C_\omega^2 S_\varphi^2 + S_\omega^2 \right) \cdot {}^s Z - \left( C_\varphi C_\omega^3 S_\varphi + C_\omega S_\varphi S_\omega^2 - S_\omega \right) \cdot l \right\}}{C_\varphi^2 C_\omega \cdot ({}^s Z - l \cdot C_\omega \cdot S_\varphi) \cdot ({}^s Z + l \cdot S_\omega)} \quad (3.82)$$

The range  ${}^s Z$  is the distance from the camera to the calibration tool, which is at least one meter in application, while  $l$  is the distance between landmarks  ${}^t P_{L_j}$  on the calibration tool, which is in millimeters. Thus, there is

$${}^s Z \gg l. \quad (3.83)$$

so we get

$$({}^s Z - l \cdot C_\omega \cdot S_\varphi) \gg 0 \quad (3.84)$$

$$({}^s Z + l \cdot S_\omega) \gg 0. \quad (3.85)$$

Then, equation (3.82) leads to rank deficient cases as

$$\left( C_\varphi C_\omega^2 S_\varphi^2 + S_\omega^2 \right) \cdot {}^s Z - \left( C_\varphi C_\omega^3 S_\varphi + C_\omega S_\varphi S_\omega^2 - S_\omega \right) \cdot l = 0 \quad (3.86)$$

$$C_\varphi = 0 \quad (3.87)$$

$$C_\omega = 0 \quad (3.88)$$

Solving the above three equations, the conditions of singularity are

$$\varphi = 0^\circ \quad \text{and} \quad \omega = \sin^{-1}\left(-\frac{l}{{}^s Z}\right) \quad (3.89)$$

$$\omega = 0^\circ \quad \text{and} \quad \varphi = \sin^{-1}\left(\frac{l}{{}^s Z}\right) \quad (3.90)$$

$$\varphi = 0^\circ \quad \text{and} \quad \omega = 0^\circ \quad (3.91)$$

$$\varphi = 90^\circ \quad (3.92)$$

$$\omega = 90^\circ \quad (3.93)$$

In the above conditions of singularity, we have

$$\det({}^v J_{w\mathcal{S}_{\hat{e}}}) = 0 \quad (3.94)$$

If rewriting equation (3.50), there is:

$${}^v J_{w\mathcal{S}_{\hat{e}}} = {}^v_i M \cdot J_{w\mathcal{S}_{\hat{e}}}, \quad (3.95)$$

where  ${}^v_i M$  is a full rank  $6 \times 6$  matrix. Correspondingly, we can conclude that

$$\det(J_{w\mathcal{S}_{\hat{e}}}) = 0 \quad (3.96)$$

when the above angles in conditions of singularity are achieved.

**Given the definition of SoR in section 3.1, there are five positions of singularity in spherical coordinate systems shown in equation (3.89)-(3.92) and (3.93).**

### 3.3.5 Nominal Verification of Singularity of Representation

After finding the five conditions of SoR in equations (3.89)- (3.93), the SoR cases can be verified numerically.

If we assume  ${}^s Z = 1.1$  m and  $l = 0.05$  m, in the case of  $\varphi = 0^\circ$  and  $\omega = \sin^{-1}(-\frac{l}{{}^s Z})$ ,

then

$${}^v J_{wS_{\hat{e}}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -22.00 \\ 0 & 0 & 0 & 21.98 & 0 & 0 \\ 0.05 & 0 & -0.91 & 0 & 0 & -22.05 \\ -0.05 & 0 & 0 & 21.98 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -22.00 \\ 0 & 0 & -0.91 & 22.02 & 0 & 0 \end{bmatrix}. \quad (3.97)$$

This matrix is rank 5. And in the case of  $\omega = 0^\circ$  and  $\varphi = \sin^{-1}(\frac{l}{\sqrt{2}})$ , the Jacobian matrix is:

$${}^v J_{wS_{\hat{e}}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -21.98 \\ 0 & 0 & 0 & 22.00 & 0 & 0 \\ 0 & 0 & -0.91 & 0 & 0 & -22.02 \\ 0 & 0.05 & 0 & 22.00 & 1.00 & 0 \\ 0 & 0 & 0 & 0 & -1.00 & -21.98 \\ 0 & -0.05 & -0.91 & 22.05 & 0 & 0 \end{bmatrix}, \quad (3.98)$$

which has rank 5. In the third case of  $\varphi = 0^\circ$  and  $\omega = 0^\circ$ , there is:

$${}^v J_{wS_{\hat{e}}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -22.00 \\ 0 & 0 & 0 & 22.00 & 0 & 0 \\ 0.05 & 0 & -0.91 & 0 & 0 & -22.05 \\ 0 & 0 & 0 & 22.00 & 1.00 & 0 \\ 0 & 0 & 0 & 0 & -1.00 & -22.00 \\ 0 & 0.05 & -0.91 & 22.05 & 0 & 0 \end{bmatrix}, \quad (3.99)$$

which also has a rank of 5. In the fourth case of  $\phi = 90^\circ$ , we get:

$${}^v J_{w_{\mathcal{S}_e}} = 1.0 \times 10^{17} \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -8.8189 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.1633 & 0 & 0 & 0 & 0 & -8.8189 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.1603 & -8.8189 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (3.100)$$

with rank 3. The last case of  $\omega = 90^\circ$  yields:

$${}^v J_{w_{\mathcal{S}_e}} = 1.0 \times 10^{17} \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9.1455 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1663 & 0 & 0 & 9.1455 & 0.1663 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.163 & 0 & 9.1455 & 0 & 0 \end{bmatrix} \quad (3.101)$$

with rank 3. The determinate value of all the above numerical matrices is zero.

### 3.4 Summary

In this chapter, I give the formal definition of singularity of representation and demonstrate cases of SoR. An example is offered to show how singularity can lead to difficulty or failure in optimization. A new representation system, the spherical coordinate system is introduced in bundle adjustment. With the introduction of the  $v$  frame, the Jacobian matrix  ${}^v J_{w_{\mathcal{S}_e}}$  expressed in the  $v$  frame is analyzed instead of  $J_{w_{\mathcal{S}_e}}$  in the image frame. After substitution and calculation of equations, the determinant of  ${}^v J_{w_{\mathcal{S}_e}}$  leads us to a simplified  $4 \times 4$  matrix  ${}^v J_{reduced}$ . Next, I establish that the corresponding Jacobian matrix is not full

rank. Furthermore, the five conditions of singularity in application are given, under which spherical coordinate representation will reach singularity. Finally, the numerical examples are offered corresponding to each conditions of singularity.

## 4 Optimization Enhancement: Robust Statistics

### 4.1 Least Squares Estimation in Bundle Adjustment

#### 4.1.1 Assumptions of the Least Squares Estimator

As stated in section 2.3.1, the non-linear estimation calculation strategy of the Gauss-Newton method is to minimize the least squares cost function  $g(\mathbf{u})$  by equation (2.48). The iterative term  $\delta\mathbf{u}$  updates the estimated parameter vectors by minimizing SSE of residual. The assumptions required for the least squares estimator to be optimal are [4, 51]:

1. Residual has constant variance;
2. Residual distribution is normal distribution;
3. Data in the sample are independent, identical distribution (i.i.d.).

A typical normal distribution PDF function is shown in figure 4.1. The normal distribution has a very small tail area.



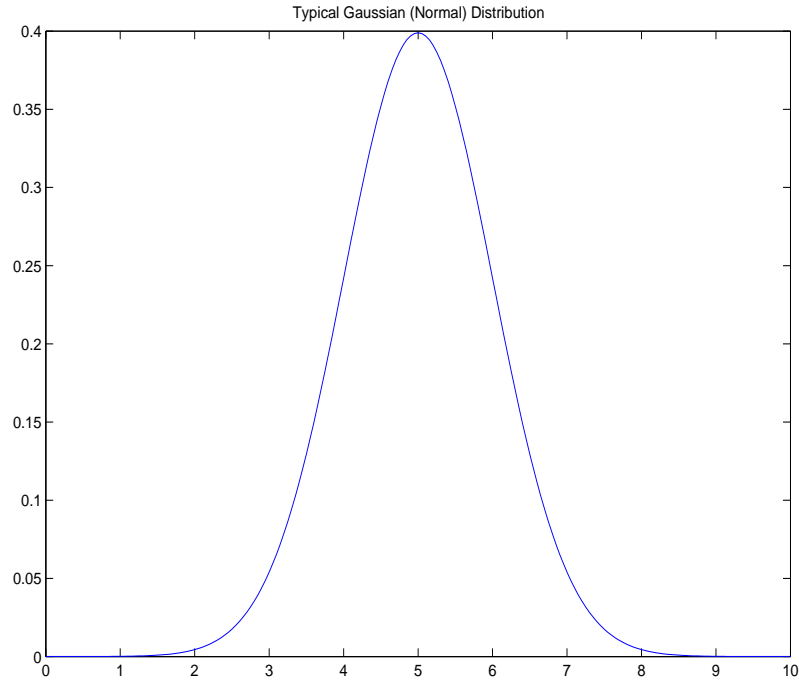


Figure 4.1: Gaussian (normal) distribution.

#### 4.1.2 Non-Gaussian Distribution of Residuals in Observations

Researchers including Triggs *et al.* [59] worked on the error modeling problem. Triggs argued that the outliers<sup>31</sup> are unavoidable in the image data used for camera calibration and that outliers can have a significant impact on the accuracy of estimation. During the optimization phase of camera calibration, the landmarks on the digital image of the camera calibration tool<sup>32</sup> with largest and smallest residual values can be identified for the establishment of the impact from the outliers in data set. Figure 4.2 demonstrates a landmark with large residual which has serious impact on the optimization. Figure 4.3 demonstrates a landmark with low residual.

31. Imperfectly grabbed features such as feature correspondence errors, a specularity, a shadow, poor focus of camera, motion blur, etc.

32. This camera calibration tool is a 2D camera calibration tool, which is shown in figure 5.2 of section 5.1.1.

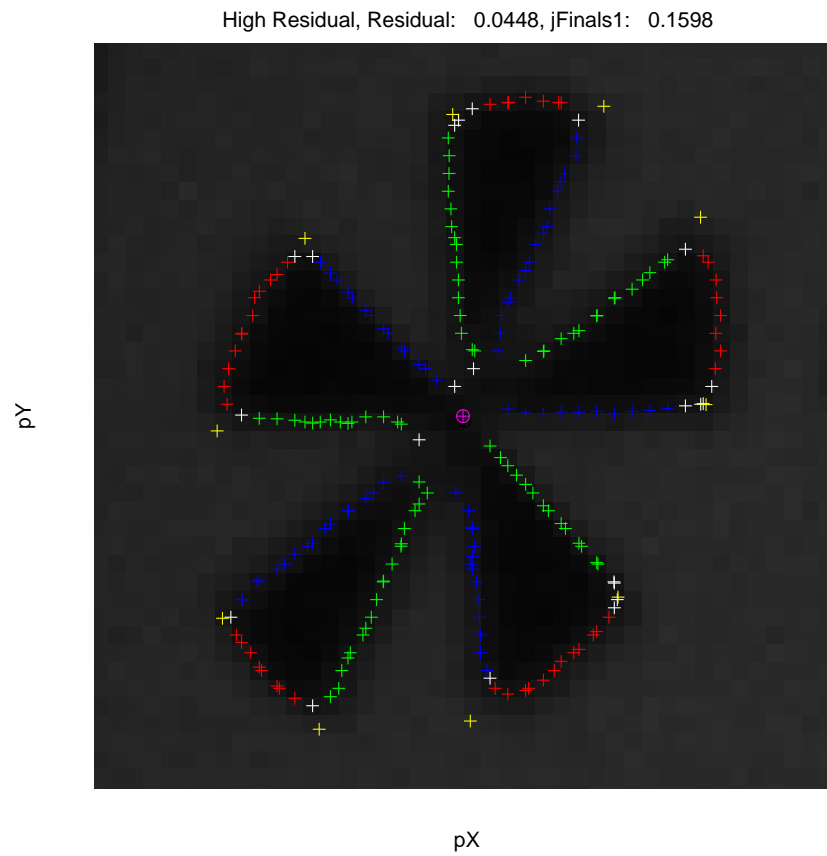


Figure 4.2: A landmark that generates a large residual value in bundle adjustment.

In figure 4.2, the pixel information yielded by the landmark is not ideal and contains an observable curve at the edge of landmark spokes. This is where the large residual comes from.

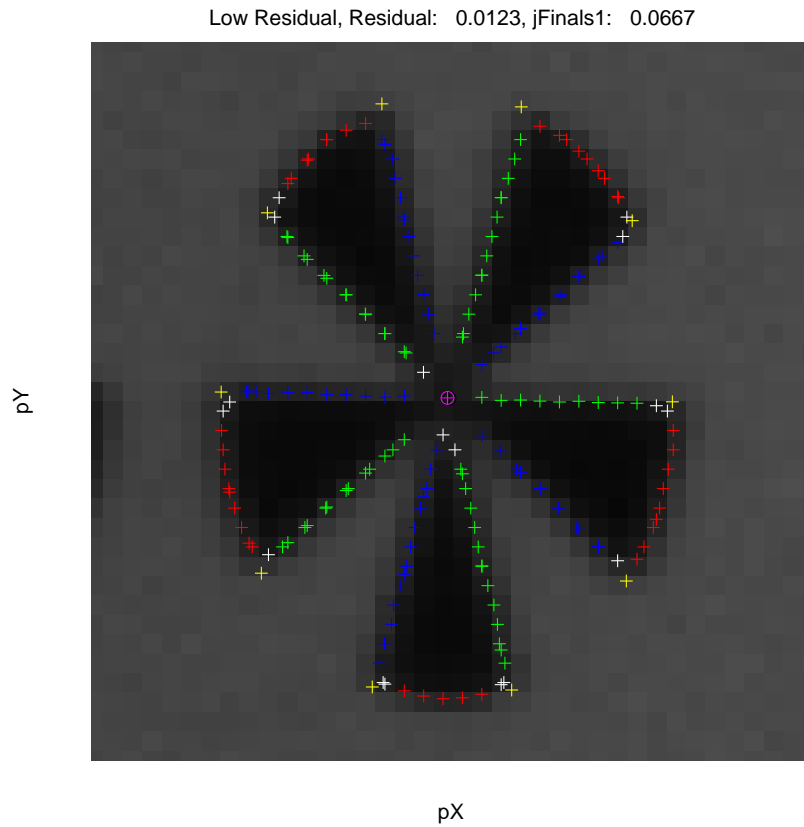


Figure 4.3: A landmark that generates a small residual value in bundle adjustment.

The landmark in figure 4.3 is processed more accurately with accurate edge detection. One essential part of camera calibration is the modeling of those outlines during the camera calibration calculation. This illustrates how the robust estimator improves robustness by de-weighting outliers created in the image process.

For least squares estimation, the standard assumption of normal distribution only has a small tail area, as shown in figure 4.1. A distribution with a small tail area poorly models outliers and so can have poor robustness. Thus, the widely-used least squares estimator may be very unreliable with the influence of outliers. In bundle adjustment, thousands of observations are collected, and more than a thousand parameters are estimated. The observations are the measurements of landmark information expressed in pixel coordinates— $p\vec{P}_a$ . The task is done by applying the image processing algorithm for

digital images. As discussed before, the distribution of residuals

$$R(\mathbf{u}) = {}^p\hat{P}_a - {}^p\bar{P}_a \quad (4.1)$$

does not follow normal distribution.

Figure 4.4 contains four normal probability plots of four different experimental camera calibration data sets collected by three individuals. Each probability plot compares the residuals of an experimental data set and a normal distribution, which can be used to judge whether the data follow a normal distribution. Figure 4.4 indicates that residuals significantly depart from a normal distribution [4, 51]. The No.1 data set was collected by a Prosilica GC1290 industrial camera with Schneider 10mm lens. The No.2, No.3, and No.4 data sets were collected by AVT-Stingray F033B industrial camera with Edmund 10mm lens. A total of 200 images were collected for each of the four data sets with the 2D calibration tool shown in figure 5.2 of the next section.

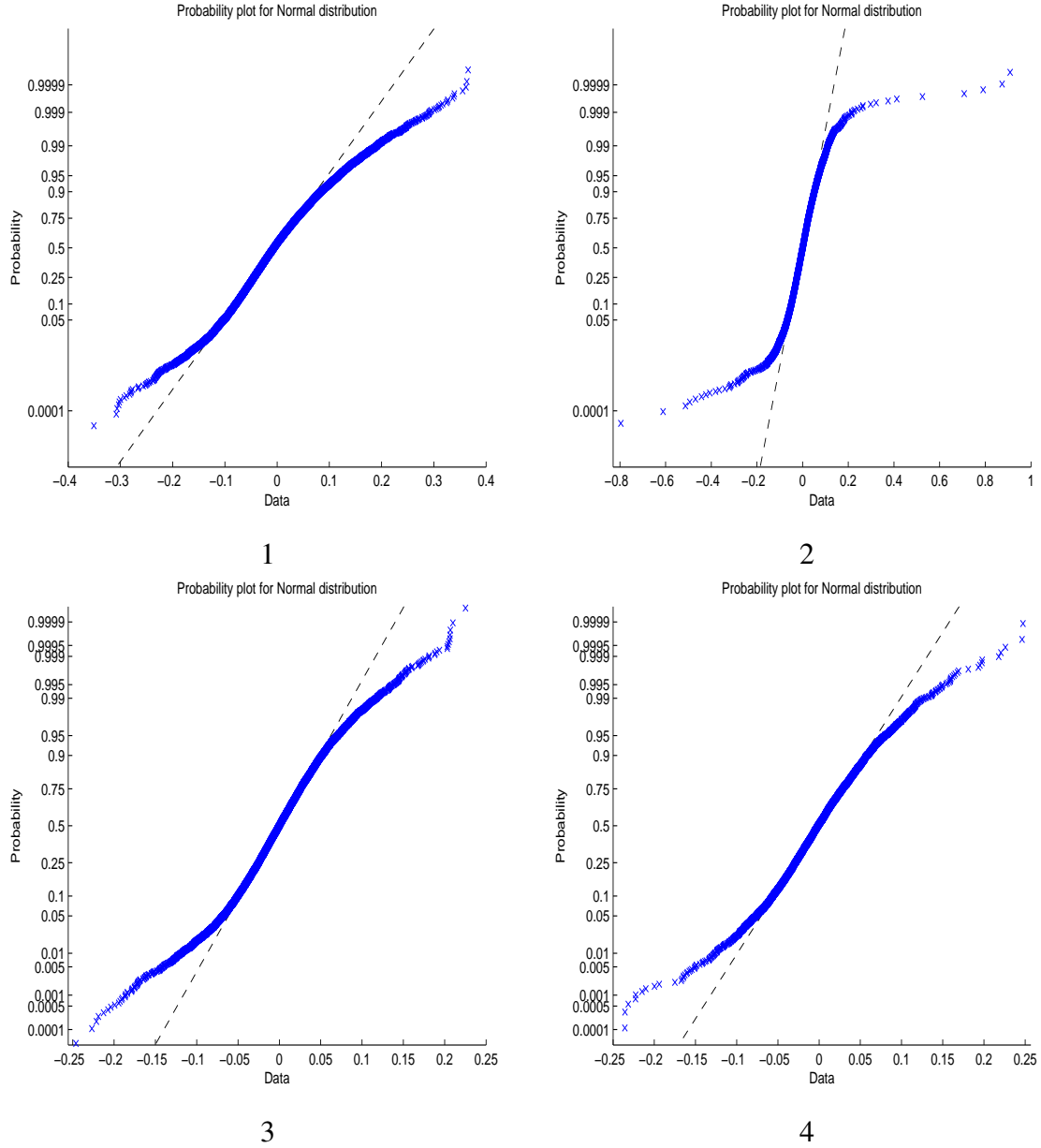


Figure 4.4: Residuals does not fit a normal distribution: four different camera calibration data sets collected by three individuals. The No.1 data set was collected by Prosilica GC1290 industrial camera with Schneider 10mm lens. The No.2, No.3, and No.4 data sets were collected by AVT-Stingray F033B industrial camera with Edmund 10mm lens.

Figure 4.5 is the comparison of the residuals of the above four experimental data sets with a normal distribution on a logarithmic scale. Hundreds of data points in the tail area lie outside the normal distribution. The distribution of residuals is the logarithmic histogram

in blue, and the normal distribution is the curve in red shown in figure 4.5. The inadequate modeling of the residual function influences the performance of bundle adjustment.

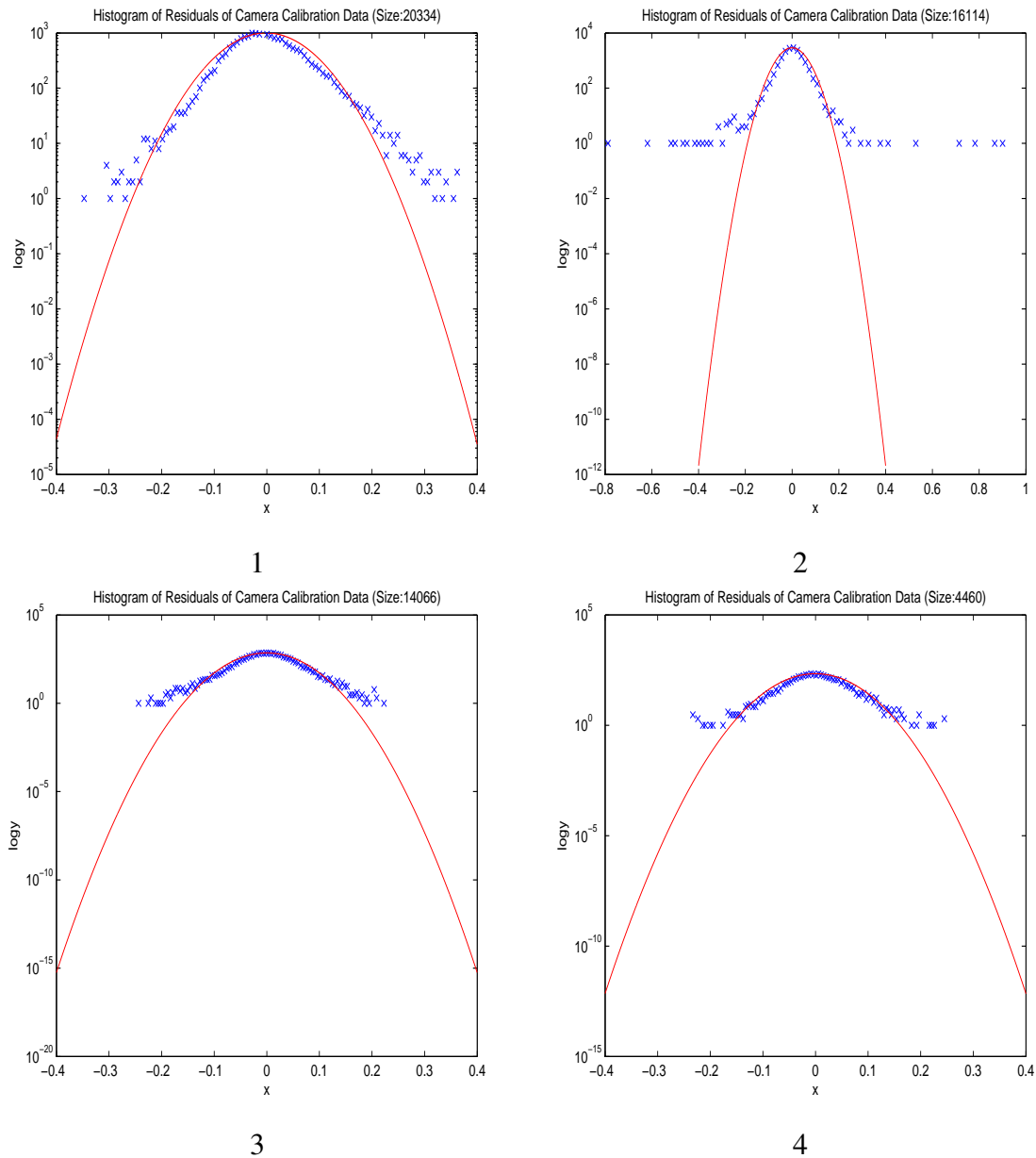


Figure 4.5: Distribution of residual function and normal distribution: four different camera calibration data sets collected by three individuals.

## 4.2 Robust Estimation in Bundle Adjustment

Triggs *et al.*[59] pointed out that blunders in the data may affect one or more of the observations without affecting other observations. This fact allows a **robust estimator**, with assumptions other than normality distribution, to take care of outliers by de-weighting or eliminating them.

### 4.2.1 General Idea of Robust Estimation

A robustified distribution has a constant tail area compared to the normal distribution. This property of the density function that the outlier area is constant reduces the influence of outliers on the estimation. In robust estimation, researchers [47, 49, 35, 66] introduce a  $\rho$  function and  $\psi$  function (if  $\rho$  is differentiable,  $\psi = \rho'$ ) to modify the cost function of the least squares estimator and check robust statistic parameters instead. There are several types of robust estimators: the M-estimator, which uses the maximum likelihood function; the L-estimator, which uses the linear combination of order statistics; the R-estimator, which uses rank transformation; the RM estimator, which uses the repeated median; and the LMS estimator, which uses the least median of squares.

Here, the functions from the M-estimator family, which contains a group of estimators, are selected. A  $\rho$  function in M-estimators is defined with the following properties [47]:

- $\rho(\mathbf{u})$  is a non-decreasing function of  $|\mathbf{u}|$ .
- $\rho(0) = 0$ .
- $\forall \mathbf{u} > 0$ ,  $\rho(\mathbf{u})$  is increasing. Thus,  $\rho(\mathbf{u}) < \rho(\infty)$ .
- If  $\rho$  is bounded, then there is an assumption as  $\rho(\infty) = 1$ .

Correspondingly, we define the  $\psi$  function as the derivative of the  $\rho$  function in the M-estimators:

- $\psi$  is odd.
- $\forall \mathbf{u} \geq 0, \psi(\mathbf{u}) \geq 0$

The functions  $\rho$  and  $\psi$  can be chosen out of the M-estimator family to provide desirable properties in terms of efficiency and performance of bundle adjustment.

#### 4.2.2 Robustified Iterative Estimation

By introducing a  $\rho =$  function, the cost function is as below:

$$\rho = \frac{1}{2} \sum_{j=1}^M r_j^2(\mathbf{u}) = \frac{1}{2} R(\mathbf{u})^T R(\mathbf{u}), \quad (4.2)$$

where  $R(\mathbf{u}) = \bar{y} - \hat{y}$  is the residual function and  $j = 1, 2, \dots, n_{ob}$ ;  $n_{ob}$  is the number of observed data points in a camera calibration data set by image processing. The  $\rho$  function is a function of residual  $R(\mathbf{u})$ . The corresponding  $\psi$  function and its derivative are:

$$\psi(R(\mathbf{u})) = \rho'(R(\mathbf{u})) = J^T \cdot \varepsilon, \quad (4.3)$$

where  $\varepsilon = R(\mathbf{u})$  is the residual function, a vector;

$$\psi'(R(\mathbf{u})) = (J^T)' \cdot \varepsilon + J^T \cdot J, \quad (4.4)$$

where  $(J^T)' \cdot \varepsilon$ , the second order term, is ignored by Gauss-Newton approximation and  $J^T J$  is an approximation to the Hessian matrix. Then, we have

$$\psi'(R(\mathbf{u})) = J^T \cdot J. \quad (4.5)$$

Correspondingly, the original formula (2.54) for iteration in Newton-like iterative algorithm becomes:

$$\hat{\mathbf{u}}_{c+1} = \hat{\mathbf{u}}_c - \frac{\nabla \rho(R(\mathbf{u}))}{\nabla^2 \rho(R(\mathbf{u}))} = \mathbf{u}_c + \delta \mathbf{u}, \quad (4.6)$$



where

$$\delta \mathbf{u} = -\frac{\nabla \rho(R(\mathbf{u}))}{\nabla^2 \rho(R(\mathbf{u}))} \quad (4.7)$$

is the update term<sup>33</sup> for each iteration in optimization. Note that the term dimensions are

$$R(\mathbf{u}) \in R^{n_{\hat{\mathbf{u}}} \times 1},$$

$$\nabla \rho(R(\mathbf{u})) \in R^{n_{\hat{\mathbf{u}}} \times 1},$$

and

$$\nabla^2 \rho(R(\mathbf{u})) \in R^{n_{\hat{\mathbf{u}}} \times n_{\hat{\mathbf{u}}}}.$$

Substituting equation (4.3) and (4.2) into (4.7), the equation (2.62) is given back as:

$$\delta \mathbf{u} = -\frac{\nabla \rho(R(\mathbf{u}))}{\nabla^2 \rho(R(\mathbf{u}))} = -\frac{J^T \cdot \boldsymbol{\varepsilon}}{J^T \cdot J} \in R^{n_{\hat{\mathbf{u}}} \times 1}. \quad (4.8)$$

As discussed in section 2.3.1, if using the least squares estimator in bundle adjustment, the algorithm is trying to minimize:

$$\min \{\rho(R(\mathbf{u}))\} = \min \left\{ \frac{1}{2} R(\mathbf{u})^T R(\mathbf{u}) \right\}. \quad (4.9)$$

After defining a weighting function as:

$$w(R(\mathbf{u})) = \frac{\psi(R(\mathbf{u}))}{R(\mathbf{u})}, \quad (4.10)$$

---

33. The equation (4.7) is better written as:

$$\delta \mathbf{u} = (\nabla^2 \rho(R(\mathbf{u})))^{-1} \cdot \nabla \rho(R(\mathbf{u}))$$

for the calculation involving matrices. The fraction representation is used in equation (4.7) instead of the matrix inversion representation here, because the former representation makes the equations clearer than the latter.

the robustified iteration is to minimize

$$w(R(\mathbf{u})) \cdot R^T(\mathbf{u})R(\mathbf{u}) \quad (4.11)$$

instead of equation (4.9). This is how the  $\rho$  and  $\psi$  functions robustify the optimization by applying a weighting function. Further, by substituting equation (4.2) into equation (4.10), we have

$$w(R(\mathbf{u})) = 1 \quad (4.12)$$

Plugging equation (4.12) into equation (4.11) gives equation (4.9) back.

Thus, another way to consider the Gauss-Newton method is that it adds the  $\rho(\cdot) = \frac{\mathbf{u}^2}{2}$  function to the original cost function which minimizes  $R(\mathbf{u}) = \|\bar{P}_a - \hat{P}_a\|$ . So, the Gauss-Newton method is to minimize  $R^2(\mathbf{u}) = \|\bar{P}_a - \hat{P}_a\|^2$  instead of  $R(\mathbf{u}) = \|\bar{P}_a - \hat{P}_a\|$ . Figures 4.6 and 4.7 show the plots of  $\rho(r_i(\mathbf{u}))$  and  $w(r_i(\mathbf{u}))$  when  $\rho = \frac{1}{2}R(\mathbf{u})^T R(\mathbf{u})$  for the least squares estimator. The weighting function demonstrates that all residuals would have equivalent weight during optimization phase of camera calibration. Thus, the outliers intend to seriously impact the update term during estimation in the optimization phase.

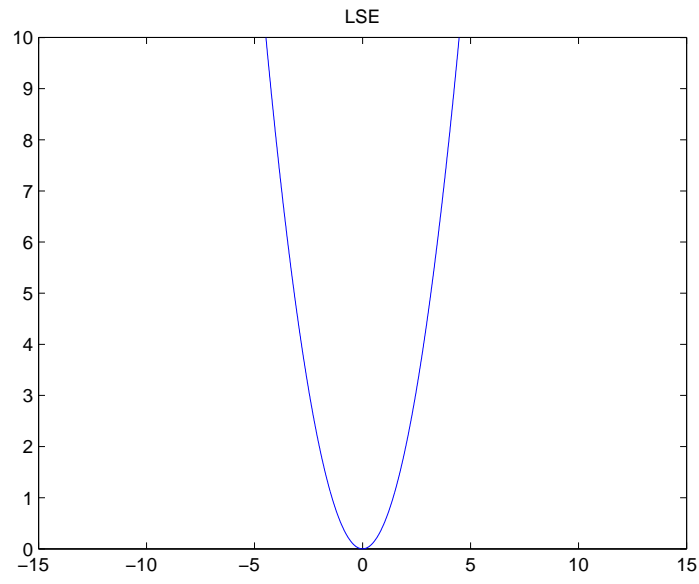


Figure 4.6:  $\rho(R(\mathbf{u}))$  of the least squares estimator.

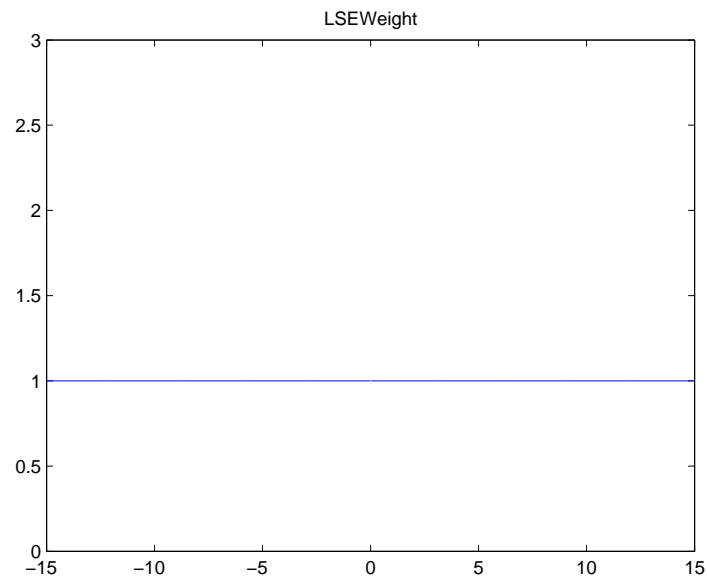


Figure 4.7:  $w(R(\mathbf{u}))$  of the least squares estimator.

### 4.3 Robustified Iterative Formulas in Bundle Adjustment

#### 4.3.1 Robustified Iterative Formulas

To avoid ending up with the least squares estimator, the  $\rho$  function is applied to equation (4.2). The robustified cost function becomes:

$$f_i(\mathbf{u}) = \rho\left(\frac{1}{2} \sum_{i=1}^M r_i^2(\mathbf{u})\right) = \rho\left(\frac{1}{2} R_i(\mathbf{u})^T R_i(\mathbf{u})\right), \quad (4.13)$$

where  $i$  is the independent observations number<sup>34</sup>. Based on equation (4.13), the gradient  $\nabla f(\mathbf{u})$  is:

$$\begin{aligned} \nabla f_i(\mathbf{u}) &= \nabla \left\{ \frac{1}{2} \rho(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u})) \right\} = \frac{1}{2} \rho'(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u})) \cdot (R_i(\mathbf{u})^T \cdot R_i(\mathbf{u}))' \\ &= \rho'(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u})) \cdot \left( \frac{1}{2} (R_i(\mathbf{u})^T)' \cdot R_i(\mathbf{u}) + \frac{1}{2} R_i(\mathbf{u})^T \cdot R_i'(\mathbf{u}) \right) \\ &= \rho'(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u})) \cdot (R_i(\mathbf{u})^T)' \cdot R_i(\mathbf{u}) = \rho'(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u})) \cdot J_i^T \cdot \varepsilon_i \\ &= \rho' \cdot J_i^T \cdot \varepsilon_i \end{aligned} \quad (4.14)$$

and the Hessian  $\nabla^2 f(\mathbf{u})$  is:

$$\begin{aligned} \nabla^2 f_i(\mathbf{u}) &= \nabla^2 \left\{ \frac{1}{2} \rho(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u})) \right\} = \nabla \left\{ \rho'(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u})) \cdot ((R_i'(\mathbf{u}))^T \cdot R_i(\mathbf{u})) \right\} \\ &= \nabla \left\{ \rho'(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u})) \right\} \cdot (R_i'(\mathbf{u}))^T \cdot R_i(\mathbf{u}) + \rho'(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u})) \cdot \nabla \left\{ (R_i'(\mathbf{u}))^T \cdot R_i(\mathbf{u}) \right\} \\ &= \rho''(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u})) \cdot \nabla(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u})) \cdot (R_i'(\mathbf{u}))^T \cdot R_i(\mathbf{u}) \\ &\quad + \rho'(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u})) \cdot \{ (R_i''(\mathbf{u}))^T \cdot R_i(\mathbf{u}) + R_i'(\mathbf{u})^T \cdot R_i'(\mathbf{u}) \} \\ &= \rho''(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u})) \cdot 2(R_i'(\mathbf{u}))^T \cdot R_i(\mathbf{u}) \cdot (R_i'(\mathbf{u}))^T \cdot R_i(\mathbf{u}) \\ &\quad + \rho'(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u})) \cdot \{ (R_i''(\mathbf{u}))^T \cdot R_i(\mathbf{u}) + R_i'(\mathbf{u})^T \cdot R_i'(\mathbf{u}) \}. \end{aligned} \quad (4.15)$$

---

34. The independent group is one single landmark data as  $\begin{bmatrix} iP_x \\ iP_y \end{bmatrix}$ . We would accept, weight, or reject them as a whole, without influence on other groups.

By neglecting the second order term  $(R_i'(\mathbf{u}))^T \cdot R_i(\mathbf{u})$  and using  $\rho', \rho''$  for  $\rho'(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u}))$  and  $\rho''(R_i(\mathbf{u})^T \cdot R_i(\mathbf{u}))$  respectively, we get

$$\begin{aligned}
\nabla^2 f_i(\mathbf{u}) &\approx \rho'' \cdot 2(R_i'(\mathbf{u})^T \cdot R_i(\mathbf{u})) \cdot (R_i'(\mathbf{u}))^T \cdot R_i(\mathbf{u}) + \rho' \cdot \{R_i'(\mathbf{u})\}^T \cdot R_i'(\mathbf{u})\} \\
&\approx \rho'' \cdot 2(J_i^T \cdot R_i(\mathbf{u})) \cdot J_i^T \cdot R_i(\mathbf{u}) + \rho' \cdot J_i^T \cdot J_i \\
&\approx 2\rho'' \cdot J_i^T \cdot R_i(\mathbf{u}) \cdot (R_i(\mathbf{u}))^T \cdot J_i + \rho' \cdot J_i^T \cdot J_i \\
&\approx J_i^T \cdot \{\rho' + 2\rho'' \cdot R_i(\mathbf{u}) \cdot (R_i(\mathbf{u}))^T\} \cdot J_i \\
&\approx J_i^T \cdot \{\rho' + 2\rho'' \cdot \varepsilon_i \cdot \varepsilon_i^T\} \cdot J_i.
\end{aligned} \tag{4.16}$$

Equations (4.14) and (4.16) form a very large group of equations corresponding to each independent landmark. Thus, the iterative formula will be

$$\mathbf{u}_{c+1} = \mathbf{u}_c - \frac{\nabla f(\mathbf{u}_c)}{\nabla^2 f(\mathbf{u}_c)} = \mathbf{u}_c - \frac{\rho' \cdot J_i^T \cdot \varepsilon_i}{J_i^T \cdot \{\rho' + 2\rho'' \cdot \varepsilon_i \cdot \varepsilon_i^T\} \cdot J_i}. \tag{4.17}$$

Given  $n_c$  as the number of camera poses,  $n_l$  as the number of landmarks per image,  $n_{ob}$  as the number of observed data points,  $\frac{n_{ob}}{2}$  as the number of landmarks in images, and  $n_{\hat{\mathbf{u}}}$  as the number of parameters to be estimated, the sizes of each symbol in the above equations are:

$$R_i(\mathbf{u}) \in C^{2 \times 1}, \tag{4.18}$$

$$\nabla f_i(\mathbf{u}) \in C^{n_{\hat{\mathbf{u}}} \times 1} \quad \text{and} \quad \nabla f(\mathbf{u}) \in C^{n_{\hat{\mathbf{u}}} \times 1}, \tag{4.19}$$

$$\nabla^2 f_i(\mathbf{u}) \in C^{n_{\hat{\mathbf{u}}} \times n_{\hat{\mathbf{u}}}} \quad \text{and} \quad \nabla^2 f(\mathbf{u}) \in C^{n_{\hat{\mathbf{u}}} \times n_{\hat{\mathbf{u}}}}, \tag{4.20}$$

$$R_i(\mathbf{u}) = \varepsilon_i \in C^{2 \times 1} \quad \text{and} \quad R(\mathbf{u}) = \varepsilon \in C^{n_{ob} \times 1}, \tag{4.21}$$

$$\rho_i(\cdot) \in C^{2 \times 1} \quad \text{and} \quad \rho(\cdot) \in C^{n_{ob} \times 1}, \tag{4.22}$$

$$\rho_i'(\cdot) \in C^{2 \times 2} \quad \text{and} \quad \rho'(\cdot) \in C^{n_{ob} \times n_{ob}}, \tag{4.23}$$

$$\rho_i''(\cdot) \in C^{2 \times 2} \quad \text{and} \quad \rho''(\cdot) \in C^{n_{ob} \times n_{ob}}, \tag{4.24}$$

$$J_i \in C^{2 \times n_{\hat{\mathbf{u}}}} \quad \text{and} \quad J \in C^{n_{ob} \times n_{\hat{\mathbf{u}}}}, \quad (4.25)$$

$$J_i^T \in C^{n_{\hat{\mathbf{u}}} \times 2} \quad \text{and} \quad J^T \in C^{n_{\hat{\mathbf{u}}} \times n_{ob}}, \quad (4.26)$$

where each group  $i$  has two data points as  $iP_a = \begin{bmatrix} iP_x \\ iP_y \end{bmatrix}$  and  $C$  represents the real number space.

Some people consider M-estimators to be an alternative method of parameter estimation instead of the least squares estimator. Actually, the least squares estimator can be considered as a special case of M-estimator [47, 66]. If we express equation (4.2) in terms of equation (4.13), equation (4.2) becomes

$$\rho = \mathbf{u} \quad (4.27)$$

Correspondingly, the  $\rho' = 1$  and  $\rho'' = 0$ , which brings us back to equations (2.56) and (2.60) in section 2.3.1.

#### 4.3.2 Specific Robustified Formulas

In the M-estimator family, there are many options for the implementation of robust estimators in bundle adjustment. One type of the robust estimator model for handling outliers is the heavy-tailed or fat-tailed distribution, in which the tail tends to zero more slowly than in a normal distribution. One example is the Cauchy distribution:

$$\rho(R_i(\mathbf{u})) = \frac{c^2}{2} \log \left( 1 + \frac{R_i(\mathbf{u})^T R_i(\mathbf{u})}{c^2} \right). \quad (4.28)$$

The equation (4.14) becomes

$$\nabla f_i(\mathbf{u}) = \frac{1}{\ln(10) \left( 1 + \frac{R_i(\mathbf{u})^T R_i(\mathbf{u})}{c^2} \right)} \cdot J_i^T \cdot \boldsymbol{\varepsilon}, \quad (4.29)$$

where

$$\rho' = \frac{1}{\ln(10) \left(1 + \frac{R_i(\mathbf{u})^T R_i(\mathbf{u})}{c^2}\right)}. \quad (4.30)$$

and equation (4.16) becomes

$$\nabla^2 f_i(\mathbf{u}) \approx J_i^T \cdot \left\{ \frac{1}{\ln(10) \left(1 + \frac{R_i(\mathbf{u})^T R_i(\mathbf{u})}{c^2}\right)} + 2 \cdot \left( \frac{2}{\ln(10) \cdot c^2 \cdot \left(1 + \frac{R_i(\mathbf{u})^T R_i(\mathbf{u})}{c^2}\right)^2} \right) \cdot \boldsymbol{\varepsilon}_i \cdot \boldsymbol{\varepsilon}_i^T \right\} \cdot J_i, \quad (4.31)$$

where

$$\rho'' = \frac{2}{\ln(10) \cdot c^2 \cdot \left(1 + \frac{R_i(\mathbf{u})^T R_i(\mathbf{u})}{c^2}\right)^2}. \quad (4.32)$$

Figures 4.8 and 4.9 are plots of the Cauchy distribution and its weighting function. As shown in figure 4.8, the Cauchy distribution is bell-shaped curve within the central region.

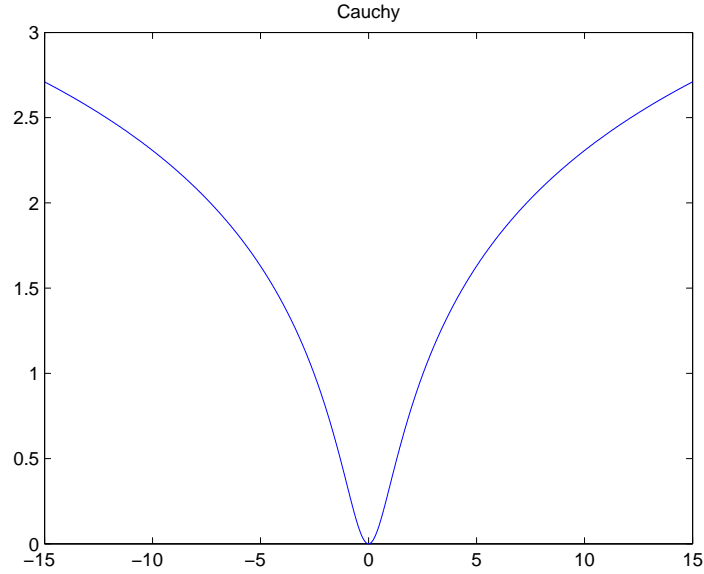


Figure 4.8:  $\rho(r_i(\mathbf{u}))$  of the Cauchy distribution based robust M-estimator.

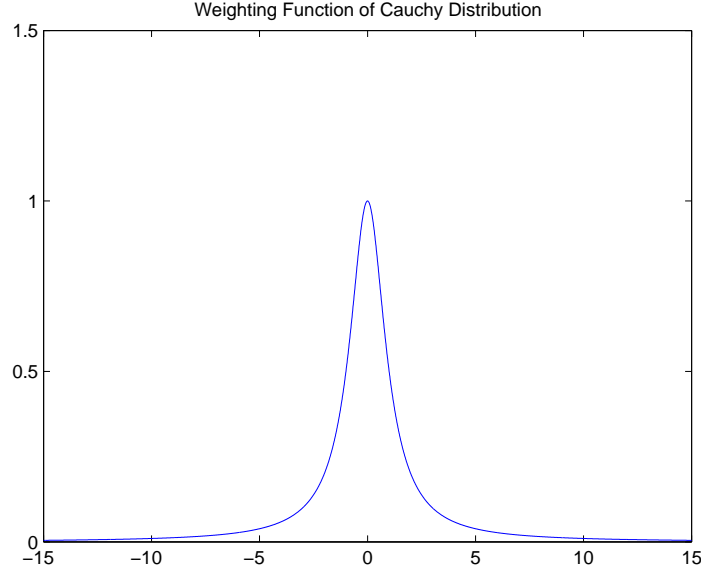


Figure 4.9:  $w(r_i(\mathbf{u}))$  of the Cauchy distribution based robust M-estimator.

Another possible example is the Welsch distribution:

$$\rho(R_i(\mathbf{u})) = \frac{c^2}{2} \left( 1 - e^{-\frac{R_i(\mathbf{u})^T R_i(\mathbf{u})}{c^2}} \right). \quad (4.33)$$

Equation (4.14) becomes

$$\nabla f_i(\mathbf{u}) = e^{-\frac{R_i(\mathbf{u})^T R_i(\mathbf{u})}{c^2}} \cdot J_i^T \cdot \boldsymbol{\varepsilon}, \quad (4.34)$$

where

$$\rho' = e^{-\frac{R_i(\mathbf{u})^T R_i(\mathbf{u})}{c^2}}, \quad (4.35)$$

and equation (4.16) becomes

$$\nabla^2 f_i(\mathbf{u}) \approx J_i^T \cdot \left\{ e^{-\frac{R_i(\mathbf{u})^T R_i(\mathbf{u})}{c^2}} + 2 \left( -\frac{2}{c^2} e^{-\frac{R_i(\mathbf{u})^T R_i(\mathbf{u})}{c^2}} \right) \cdot \boldsymbol{\varepsilon}_i \cdot \boldsymbol{\varepsilon}_i^T \right\} \cdot J_i, \quad (4.36)$$

where

$$\rho'' = -\frac{2}{c^2} e^{-\frac{R_i(\mathbf{u})^T R_i(\mathbf{u})}{c^2}}. \quad (4.37)$$

Figures 4.10 and 4.11 show the plot of a Welsch distribution and its weighting function.



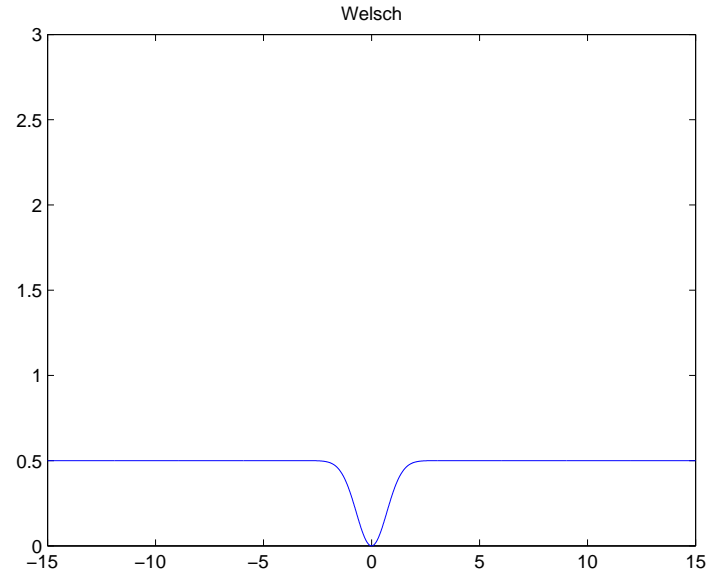


Figure 4.10:  $\rho(r_i(\mathbf{u}))$  of the Welsch distribution based robust M-estimator.

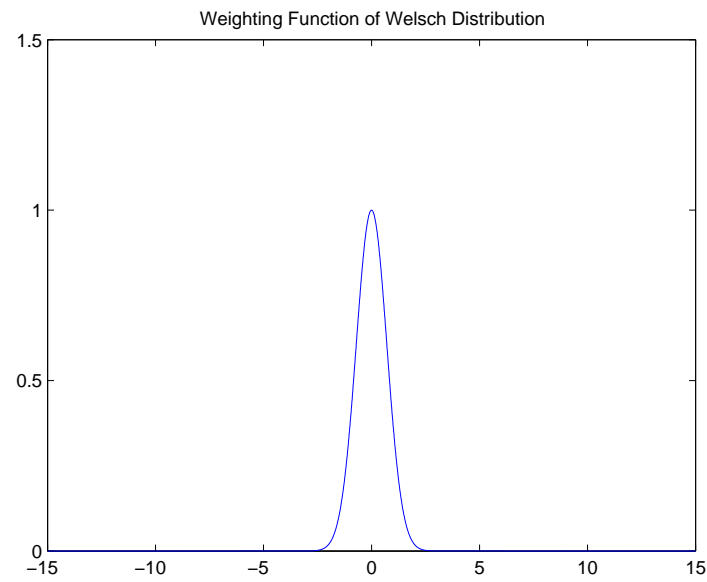


Figure 4.11:  $w(r_i(\mathbf{u}))$  of the Welsch distribution based robust M-estimator.

Figure 4.12 shows the plots of the normal distribution, the Cauchy distribution and the Welsch distribution. As you can see, within the central region, the curve of the robustified  $\rho$  function overlaps the curve of the normal distribution. This means the robust estimator

and the least squares estimator have the same behavior within a certain range of data.

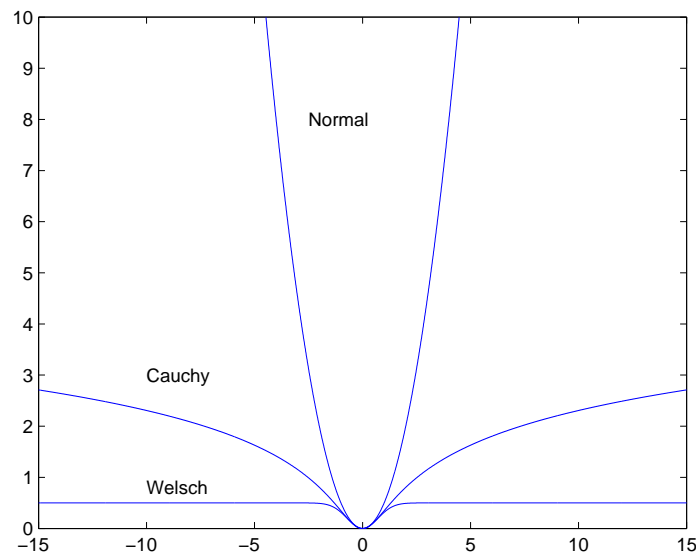


Figure 4.12: Distribution of the normal distribution, the Cauchy distribution, and the Welsch distribution.

Figure 4.13 is the plot of weighting functions corresponding to the LSE, Cauchy and Welsch distributions. The weighting curve of the Cauchy and Welsch distributions is not a straight line at 1 but a bell-shaped curve, which indicates that outside a certain range, the robustified estimator will substantially de-weight outliers. Note that the bell-shaped weighting functions of the Cauchy and Welsch distributions would apply to equation (4.11) for each update term during iteration.

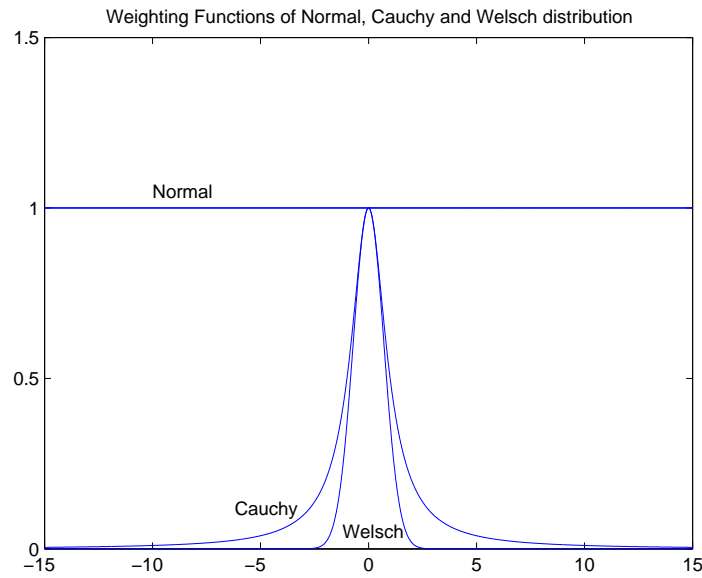


Figure 4.13: Weighting function of the normal distribution, the Cauchy distribution, and the Welsch distribution.

Additionally, according to Ricardo *et al.* [47], if  $\psi$  is re-descending, this will lead us to the re-descending M-estimator, which has decent robustness performance corresponding to those large outliers. Figures 4.14 and 4.15 are the plots of  $\psi$  functions of the Cauchy and Welsch distributions.

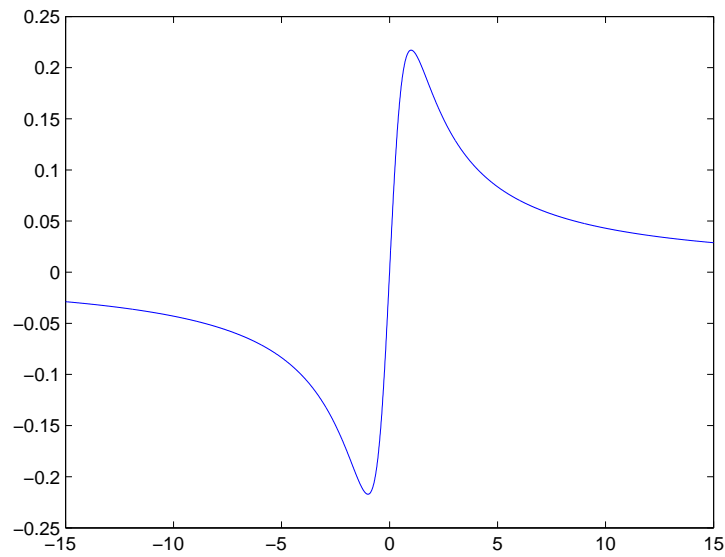


Figure 4.14:  $\psi$  function of the Cauchy distribution.

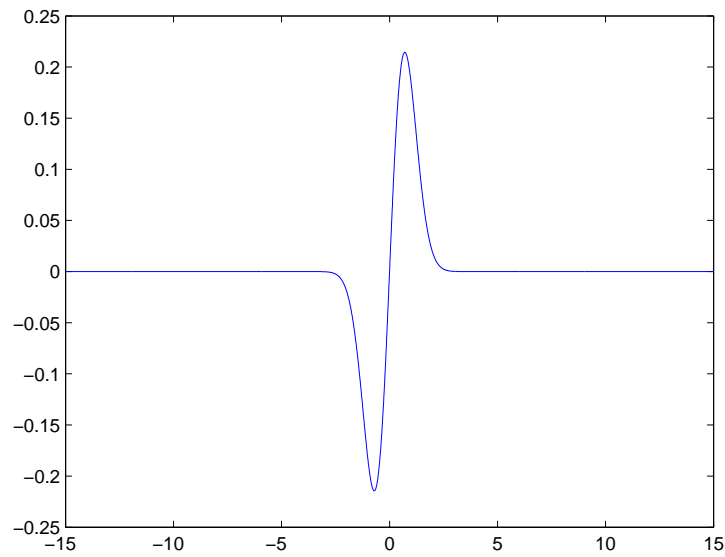


Figure 4.15:  $\psi$  function of the Welsch distribution.

Both of the Cauchy and Welsch distributions are re-descending. However, the re-descending feature of the Welsch distribution is stronger than the re-descending feature of the Cauchy distribution, since the curve of the Welsch distribution goes to zero faster

than the curve of the Cauchy distribution. The re-descending feature that tends to zero at  $\infty$  is preferable, as it implies that for a large sample, the  $\rho$  function increases relatively more slowly.

#### 4.4 $\bar{D}$ , a New Metric for Measuring the Distance Between Camera Calibrations

In the existing literature, no satisfactory metric has been proposed to express camera calibration accuracy among several results of camera calibration with consideration of all camera IO parameters. To measure the distance between camera calibration results, I introduce a variable  $\bar{D}$ .

To define  $\bar{D}$ , first the  $d_k^{(i,j)}$  is designed as:

$$d_k^{(i,j)} = ||^p P_k^{(i)} - ^p P_k^{(j)}||_2, \quad (4.38)$$

where  $d_k^{(i,j)}$  is the 2-norm of the distance between  $k^{th}$  landmark in two camera calibrations, whose value is in pixels;  $k$  is an index of landmark points in the image of simulated grid of points;  $i$  and  $j$  are index of camera models  $\chi_i$  and  $\chi_j$ ; and

$$^p P_k^{(i)} = \begin{bmatrix} ^p P_x \\ ^p P_y \end{bmatrix} \in R^{2 \times k}, \quad (4.39)$$

where the  $^p P_k^{(i)}$  is calculated by equations (2.14)-(2.17) and the unit is pixel. Section 2.1.3 talks about how to perform the calculation from  ${}^t P_a$  data by following camera geometry. Here, the  ${}^t P_a$  data is generated by simulated grid of points. Thus, the only experimental information in  $^p P_k^{(i)}$  calculation is each estimated camera model. This will make  $\bar{D}$  focus on the evaluation of estimated camera models.

Then, the distances vector is:

$$D_{i,j} = \begin{bmatrix} d_1^{(i,j)} \\ d_2^{(i,j)} \\ \vdots \\ d_k^{(i,j)} \end{bmatrix} \in R^{n_k \times 1}. \quad (4.40)$$

Finally,  $\bar{D}$ , the distance between two camera calibrations, is:

$$\bar{D}_{i,j} = \bar{D}(\chi_i, \chi_j) = ||D_{i,j}||_{\infty}, \quad (4.41)$$

where  $\hat{\chi}$  is the camera model from each camera calibration.

For example, if there is relationship:

$$\bar{D}(\hat{\chi}_1, \chi^*) < \bar{D}(\hat{\chi}_2, \chi^*), \quad (4.42)$$

where  $\chi^*$  is the true camera model, as the reference, then we say  $\hat{\chi}_1$  is more accurate than  $\hat{\chi}_2$ . In table 4.1,  $\bar{D}$  is used to calculate the pair-wise distance between camera calibrations among seven different camera calibration results, which were collected by two individuals (K1, K2, K3, K4, K5 were collected by one individual; R1 and R2 were collected by the other individual) with an AVT-StingrayF033B industrial camera and Edmund 10mm optical lens. The  $\bar{D}$  table is **symmetrical** with a value of 0 for all diagonal elements, since the comparison of the  $i^{th}$  calibration itself yields zero. In table 4.1, the root mean square (RMS) value of each column gives the minimum and maximum distance among camera calibration results. The camera model corresponding to the minimum RMS value is identified as the confident good camera calibration result  $\hat{\chi}^*$ , which is the estimation of ideal camera model, whereas the estimated camera model corresponding to the maximum

RMS value is considered the worst camera calibration result. If there is a result considered an outlier out of all camera calibration results according to  $\bar{D}$  matrix, the camera model should be eliminated from the comparison group before identification of the confident good camera calibration result  $\hat{\chi}^*$ .

Data Sets	K1	K2	K3	K4	K5	R1	R2
K1	0	0.114	0.487	1.112	0.224	0.312	0.213
K2	0.114	0	0.539	1.152	0.250	0.349	0.178
K3	0.487	0.539	0	1.592	0.317	0.250	0.556
K4	1.112	1.152	1.592	0	1.314	1.411	1.296
K5	0.224	0.250	0.317	1.314	0	0.222	0.256
R1	0.312	0.349	0.250	1.411	0.222	0	0.406
R2	0.212	0.178	0.556	1.296	0.256	0.406	0

Table 4.1: Distances among 7 different camera calibration results with AVT-StingrayF033B industrial camera and Edmund 10mm optical lens. K1, K2, K3, K4, K5 were collected by one individual; R1 and R2 were collected by the other individual.

## 4.5 Performance Analysis of Robustified Optimization

### 4.5.1 Algorithm Performance Analysis

The setup of data collection is shown in section 5.2.1. Under the same hardware configuration, 15 image sets were collected using a 2D calibration tool with the depth change between front and rear positions during data collection. First, the camera calibration was performed with the least squares estimator, and the corresponding  $\bar{D}$  matrix is computed and listed in table 4.2. Next, the least squares estimator is replaced by the two selected robustified estimators in the parameter algorithm. Then, the two  $15 \times 15$   $\bar{D}$  matrices of the Welsch distribution and the Cauchy distribution are calculated accordingly.

With the  $15 \times 15$   $\bar{D}$  matrices corresponding to the least squares estimator, the Cauchy distribution and the Welsch distribution, the root mean square values for each  $\bar{D}$  matrix are formed and listed in table 4.3. This table demonstrates that the performance of the robustified estimator in bundle adjustment gains up to 20% reduction of  $\bar{D}$ .

	RMS of $\bar{D}$ (pixel)	Improvement Percentage
LSE	0.489	—
Welsch	0.391	20.04%
Cauchy	0.418	14.52%

Table 4.3: Performance Analysis of the LSE and Robustified Estimator.

Figure 4.16 shows the residual values generated by the robustified estimator with the Welsch distribution and the weighting function of the Welsch distribution. As mentioned in section 4.3, compared to the flat line weighting function of the least squares estimator shown in figures 4.7 and 4.13, the Welsch distribution gains robustness by de-weighting outliers in residual values, while the least squares estimator heavily weights outliers in data.



	Set 1	Set 2	Set 3	Set 4	Set 4	Set 6	Set 7	Set 8	Set 9	Set 10	Set 11	Set 12	Set 13	Set 14	Set 15
Set 1	0	0.697	0.641	0.556	0.658	1.070	0.562	1.090	0.346	0.912	1.191	0.614	0.356	0.571	0.408
Set 2	—	0	0.294	0.268	0.176	0.380	0.231	0.397	0.396	0.220	0.500	0.145	0.534	0.183	0.316
Set 3	—	—	0	0.254	0.255	0.454	0.515	0.469	0.352	0.289	0.574	0.181	0.312	0.316	0.359
Set 4	—	—	—	0	0.153	0.521	0.273	0.547	0.240	0.364	0.647	0.142	0.262	0.121	0.158
Set 5	—	—	—	—	0	0.414	0.263	0.436	0.350	0.255	0.537	0.098	0.420	0.089	0.268
Set 6	—	—	—	—	—	0	0.524	0.269	0.780	0.189	0.252	0.475	0.753	0.514	0.698
Set 7	—	—	—	—	—	—	0	0.552	0.383	0.411	0.632	0.344	0.532	0.204	0.303
Set 8	—	—	—	—	—	—	—	0	0.811	0.188	0.159	0.499	0.787	0.537	0.730
Set 9	—	—	—	—	—	—	—	—	0	0.595	0.874	0.304	0.225	0.255	0.082
Set 10	—	—	—	—	—	—	—	—	—	0	0.288	0.308	0.587	0.349	0.533
Set 11	—	—	—	—	—	—	—	—	—	—	0	0.601	0.887	0.641	0.830
Set 12	—	—	—	—	—	—	—	—	—	—	—	0	0.387	0.144	0.229
Set 13	—	—	—	—	—	—	—	—	—	—	—	—	0	0.379	0.238
Set 14	—	—	—	—	—	—	—	—	—	—	—	—	—	0	0.238
Set 15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0

Table 4.2: Upper triangular part of  $\bar{D}$  of least squares estimation in bundle adjustment, where  $\bar{D}$  is a symmetrical matrix.

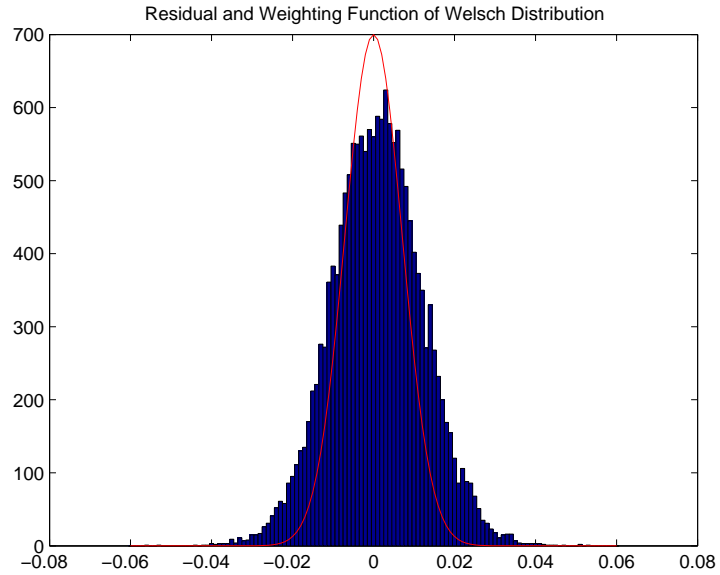


Figure 4.16: Residual and weighting function of the Welsch distribution.

#### 4.5.2 Adjustment of the Robustified Estimators

The parameters  $c$  is tuned by  $5\times$  magnification. Figures 4.17 and 4.18 are the plots of Welsch distribution after  $5\times$  magnification of  $c$ . It is obvious that in the figure 4.17 is the Welsch distribution that has the least-squares-like curve within central region. Figure 4.18 is just in different scale, by which you can see that the tuning of parameters  $c$  delays the position of bending a least-squares-like curve by comparing with figure 4.10

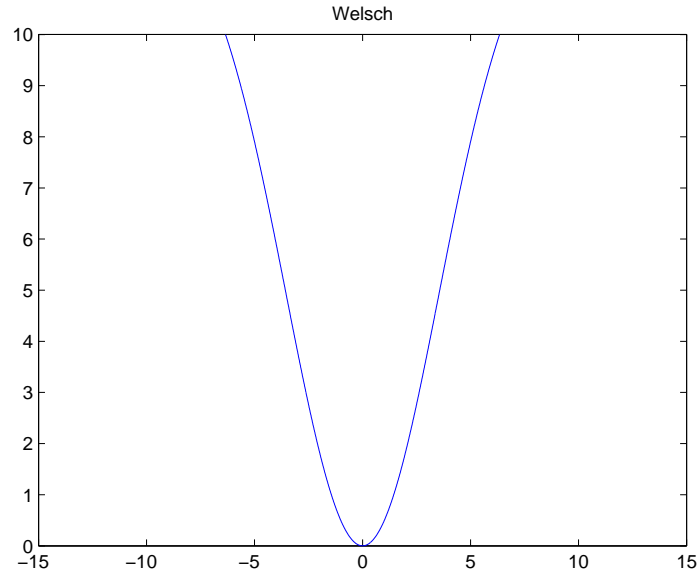


Figure 4.17: The Welsch distribution after  $5\times$  magnification of  $c$ : y axis scale is 0 to 10.

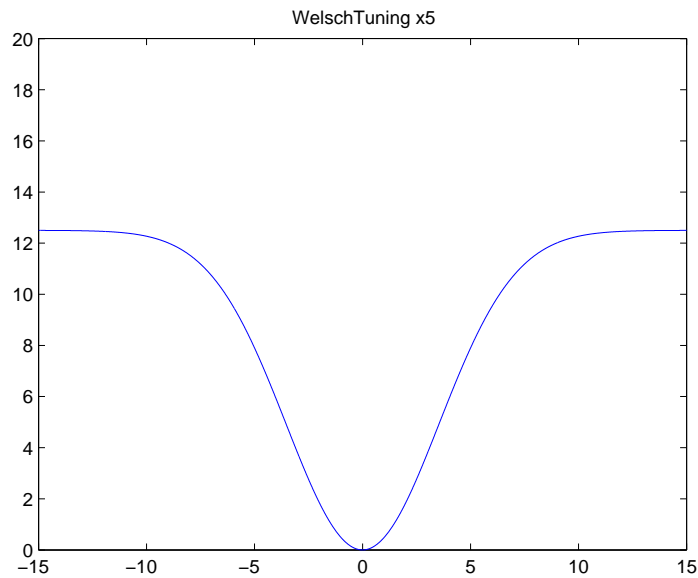


Figure 4.18: The Welsch distribution after  $5\times$  magnification of  $c$ : y axis scale is 0 to 20.

Figure 4.19 is the plot of the Welsch distribution after  $\frac{1}{5}\times$  magnification of  $c$ . As you can see, it is a very sharp quadratic curve.

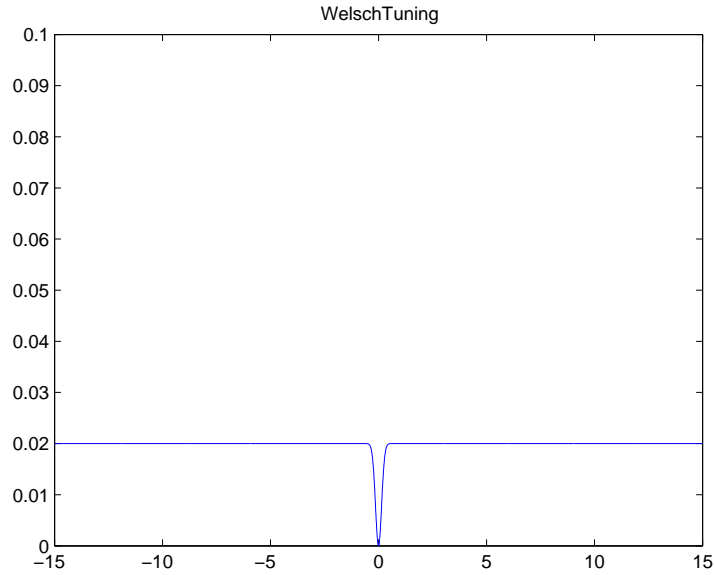


Figure 4.19: The Welsch distribution after  $\frac{1}{5} \times$  magnification of  $c$ : y axis scale is 0 to 0.1.

Figures 4.20 and 4.21 are the combined plot of the Welsch distribution with adjustment of the  $c$  parameters.

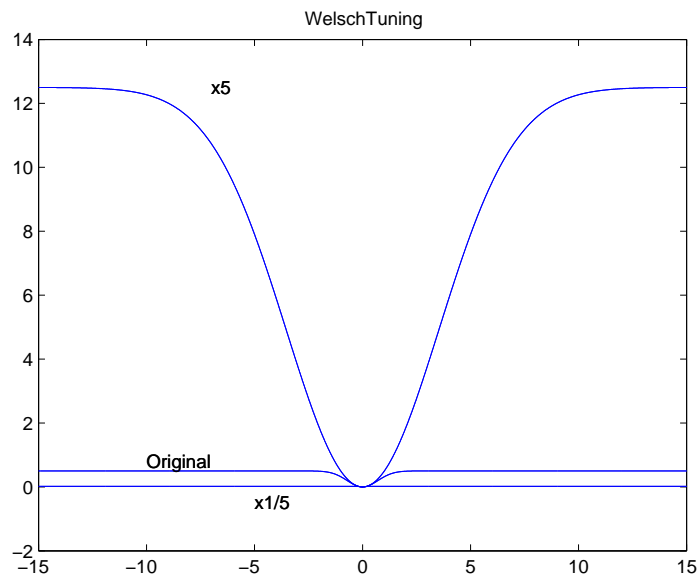


Figure 4.20: The Welsch distribution with tuning the  $c$  parameters:  $5 \times$ , Original,  $\frac{1}{5} \times$ .

Note that figure 4.21 is the magnified version of figure 4.20. Figure 4.21 is for better

observing the overlapped curves of the modified the Cauchy distributions, since the  $\frac{1}{5}$  modification seems like a straight line in figure 4.20.

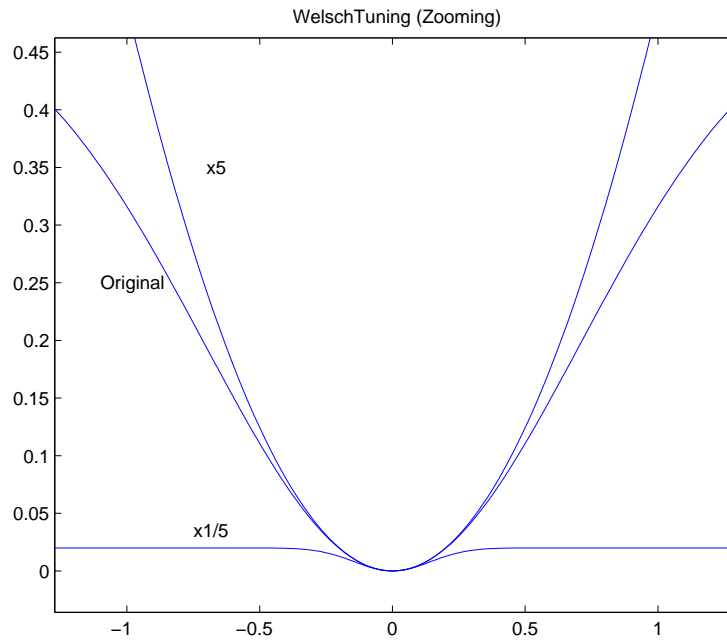


Figure 4.21: The magnification of the Welsch distribution with tuning the  $c$  parameters:  $5\times$ , Original,  $\frac{1}{5}\times$ .

Similarly, figures 4.22 and 4.23 are the combined plots of the Cauchy distribution with adjustment of the  $c$  parameters. Figure 4.23 zooms figure 4.22 for better observation of the overlapped curves from modified the Cauchy distributions.

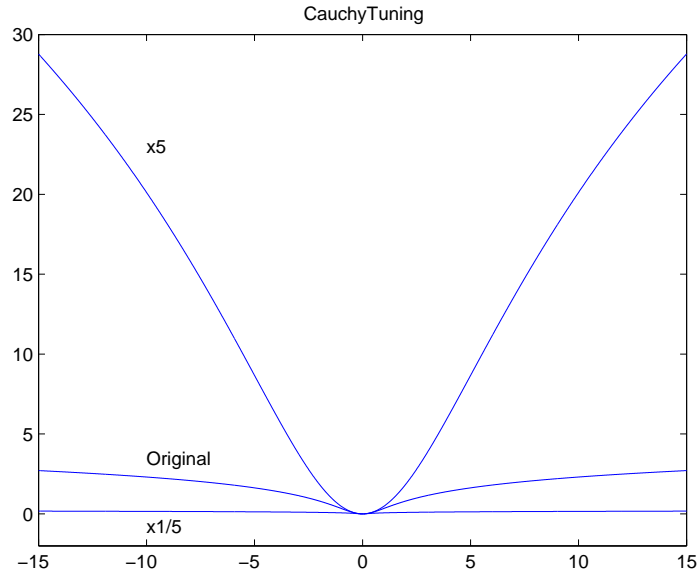


Figure 4.22: The Cauchy distribution with tuning the  $c$  parameters:  $5\times$ , Original,  $\frac{1}{5}\times$ .

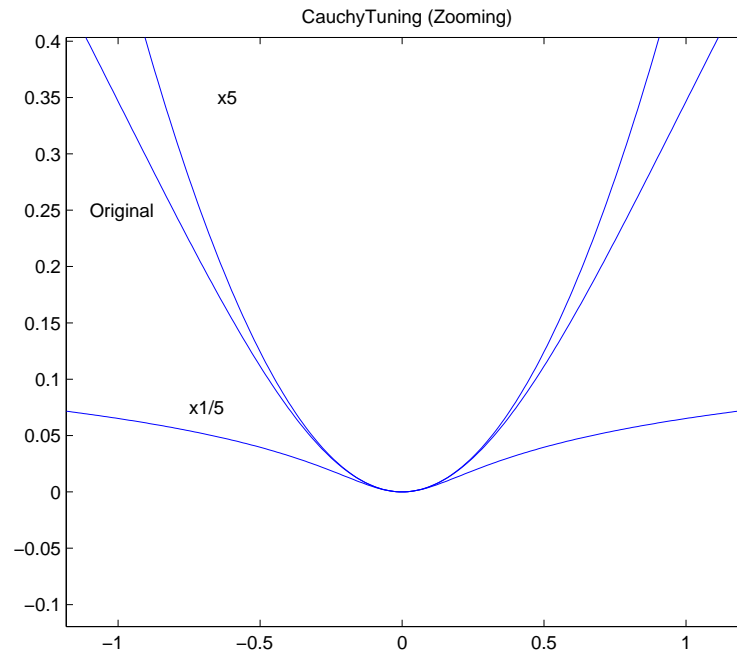


Figure 4.23: The magnification of the Cauchy distribution with tuning the  $c$  parameters:  $5\times$ , Original,  $\frac{1}{5}\times$ .

By adjusting as above, we can find whether the change of a robustified estimator would

impact the calculation accuracy. Next, I redo the camera calibration for all 15 image sets and  $\bar{D}$  matrix calculation. Finally, applying RMS for all values in the  $\bar{D}$  matrix after tuning robustified estimator, table 4.4 is obtained.

	RMS of $\bar{D}$ (pixel)	Improvement Percentage
LSE	0.486	—
Welsch	0.391	20.04% (better)
Welsch $5 \times c$	0.361	25.72% (better)
Welsch $\frac{1}{5} \times c$	0.772	58.85% (worse)
Cauchy	0.418	14.52% (better)
Cauchy $5 \times c$	0.367	24.49% (better)
Cauchy $\frac{1}{5} \times c$	0.772	58.85% (worse)

Table 4.4: Performance Analysis of the LSE and Robustified Estimator.

This table demonstrates that the performance improvement of the robustified estimator in bundle adjustment can be better if tuning  $c$  in the right way. Or, if the  $c$  is not appropriate, the robustified estimator would lose its expected robustness.

## 4.6 Summary

In this chapter, the disadvantage of the least squares estimator is analyzed first. The outliers significantly impact the estimation accuracy under least squares estimation in optimization. Next, the advantage of robust estimator is demonstrated, and the outliers are de-weighted by a robustified estimator. The impact of outliers is considerably less under robustified estimation procedure in optimization. Two distributions from M-estimator family are selected and applied to bundle adjustment. A new metric  $\bar{D}$  is introduced, which is the distance between two camera calibrations with consideration of all estimated camera IO parameters.  $\bar{D}$  is used to evaluate the performance among the least squares estimator and robust estimators. After applying a robust estimator, the system improves the accuracy and performance in camera calibration up to 25% with the analysis of experimental camera calibration calculation results. Lastly, it is demonstrated that if the robustified estimator is tuned properly, the accuracy can be improved further.

## 5 Experimental Enhancement: 3D Calibration Tool

Bundle adjustment is performed with the goal of high accuracy estimation for camera calibration. The bundle adjustment algorithm accepts a huge amount of observation data and does large scale optimization. In addition to avoidance of singularity in section 3 and improvement of outlier handling in section 4, a revolution of calibration tool used in camera calibration is also necessary for accuracy improvement of camera calibration results.

### 5.1 Camera Calibration Tool

#### 5.1.1 2D Camera Calibration Tool

##### Overview of 2D calibration tools

In general, a 2D calibration tool is used for camera calibration. The drawback is that the corresponding Hessian matrix may be ill-conditioned, which is a major reason for failure of numerical calculation in camera calibration. Various 2D calibration tools are shown in figure 5.1. The first one, from Remondino and Fraser [54], is a planar checker board, which is a very common planar calibration tool. The second one, from Zhang [68], is a checker board with special dots and is used to identify the correspondence of reference points.



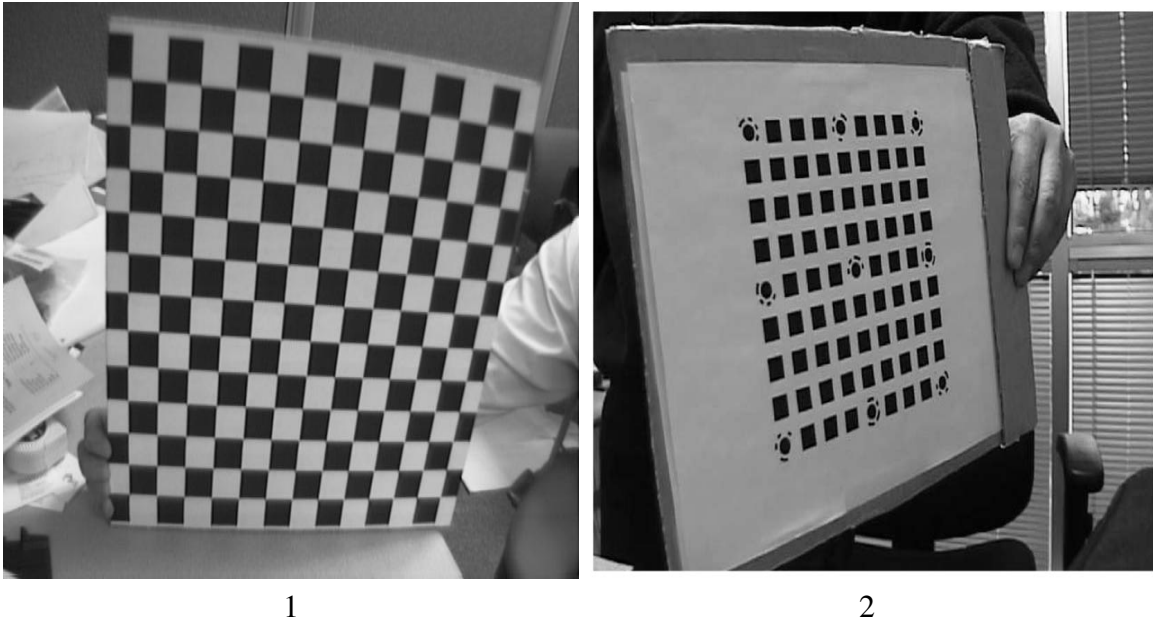


Figure 5.1: 2D calibration tools: No.1 adopted from Remondino and Fraser [54]; No.2 adopted from Zhang [68].

### Artwork of 2D calibration tool

The **2D calibration tool**, which relies on the starburst image processing algorithm, is used in this thesis. The 2D calibration tool is shown in figure 5.2.

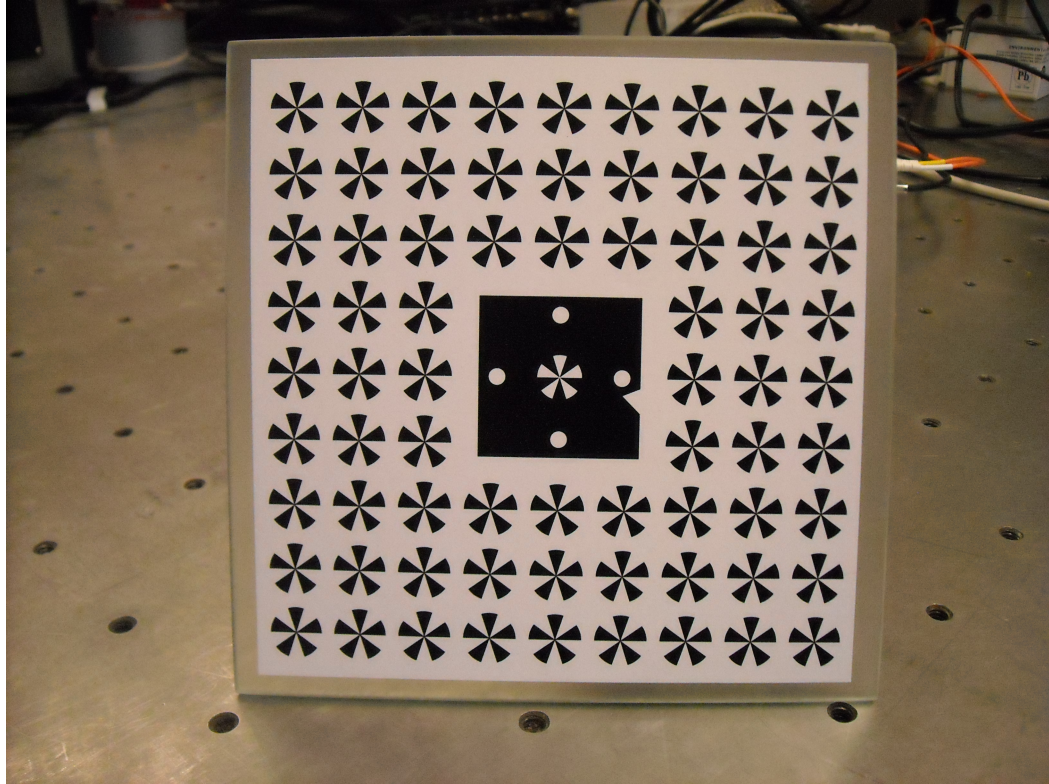


Figure 5.2: The 2D calibration tool with 1 white central starburst landmark around by four circles and 72 black starburst landmarks.

There is one white starburst landmark surrounded by four circles in the center of the calibration tool and seventy-two black starburst landmarks around the white starburst. In total, there are 73 starburst landmarks on the 2D calibration tool.

### 5.1.2 3D Camera Calibration Tool

#### Overview of 3D calibration tool

A 3D calibration tool will introduce additional constraints in the parameter estimation, yielding a better-conditioned Hessian matrix and a smaller norm value of corresponding covariance matrix. Several 3D calibration tools used by other researchers are shown in figure 5.3. The first is from Abraham and Hau[2]; the second is from Remondino and Fraser [54]; the third is from Kunii and Chikatsu [36]; and the fourth is from Heikkila [33]. Various designs of a 3D calibration tool have been used. The choice of landmark features

is up to the researchers and their system setup.

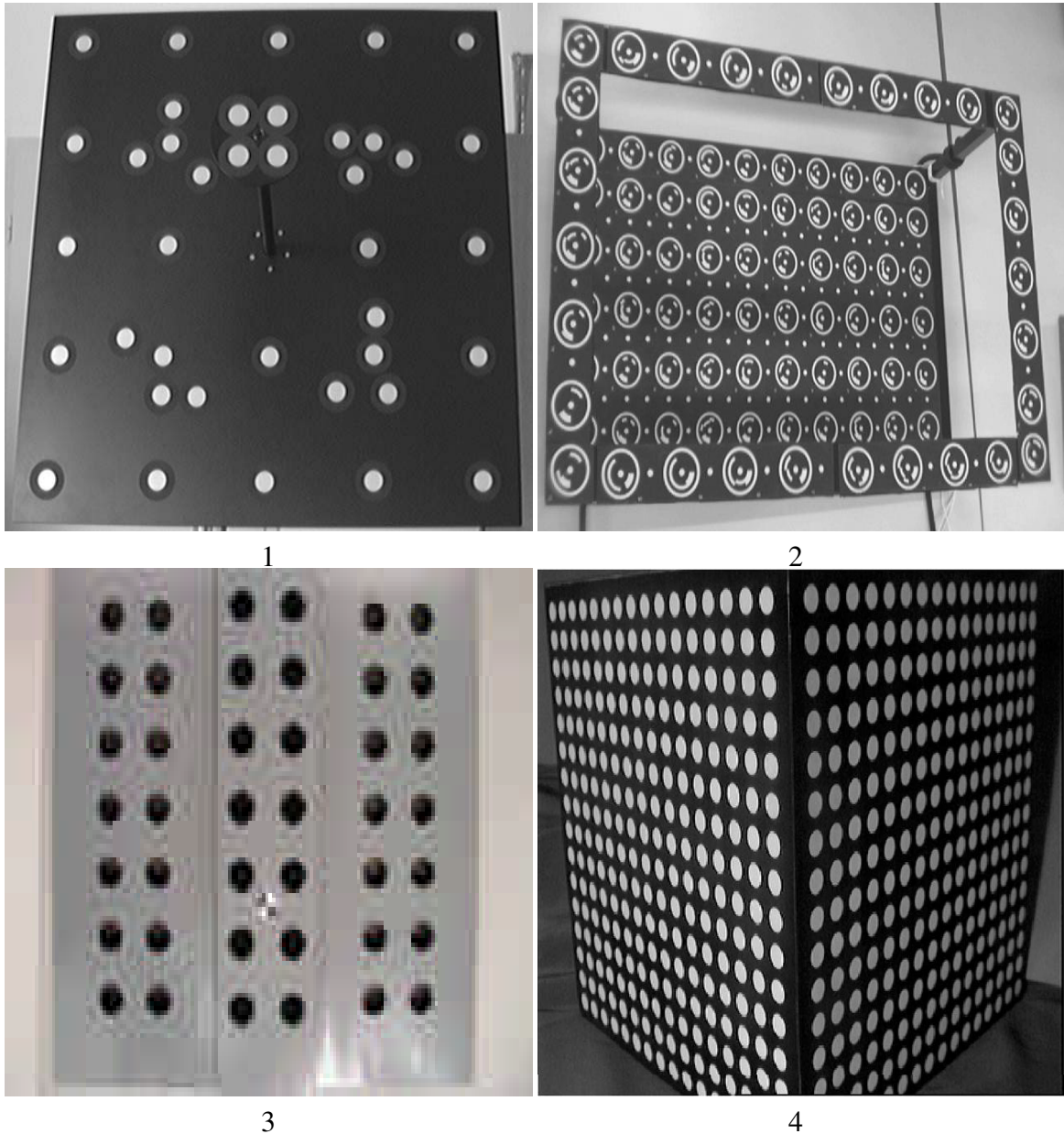


Figure 5.3: 3D calibration tools: No.1 adopted from Abraham and Hau[2]; No.2 adopted from Remondino and Fraser [54]; No.3 adopted from Kunii and Chikatsu [36]; No.4 adopted from Heikkila [33].

### Artwork of 3D calibration tool

Like the 2D calibration tool, the 3D calibration tool used in this thesis is designed to work with the starburst image processing algorithm. It is shown in figures 5.4, 5.5 and 5.6. The

investigated 3D calibration tool has a raised area in the center. The bundle adjustment achieves ray intersection to within  $\pm 3$  microns on the calibration tool, so elevating some landmarks by 3 to 6 mm will be significant. Figure 5.4 shows a line drawing of the design.

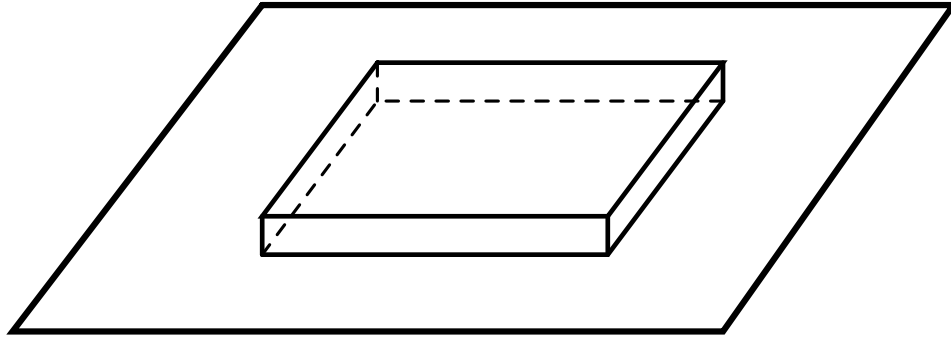


Figure 5.4: Design of a 3D calibration tool.

The practical **3D calibration tool** is shown in figures 5.5 and 5.6.

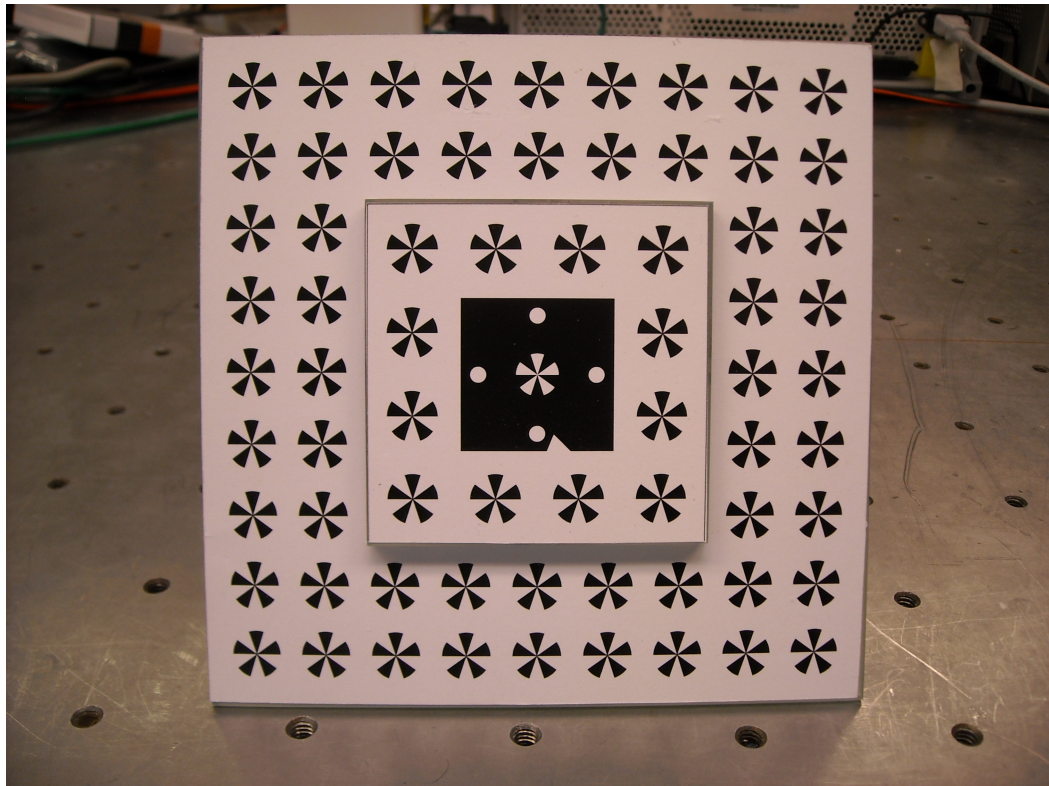


Figure 5.5: The 3D calibration tool with 1 white central landmark surrounded by four circles and 68 black starburst landmarks.

As you can see, there is one white starburst landmark surrounded by four circles in the



center of the calibration tool and sixty-eight black starburst landmarks around the white starburst. Thus, there are a total of 69 starburst landmarks on the 3D calibration tool. For each starburst landmark, the center point is the intersection of the five spokes of the starburst. Figure 5.6 illustrates the raised area in the middle of 3D calibration tool, which is measured at 5.88 mm<sup>35</sup> higher than the rest of 3D calibration tool.

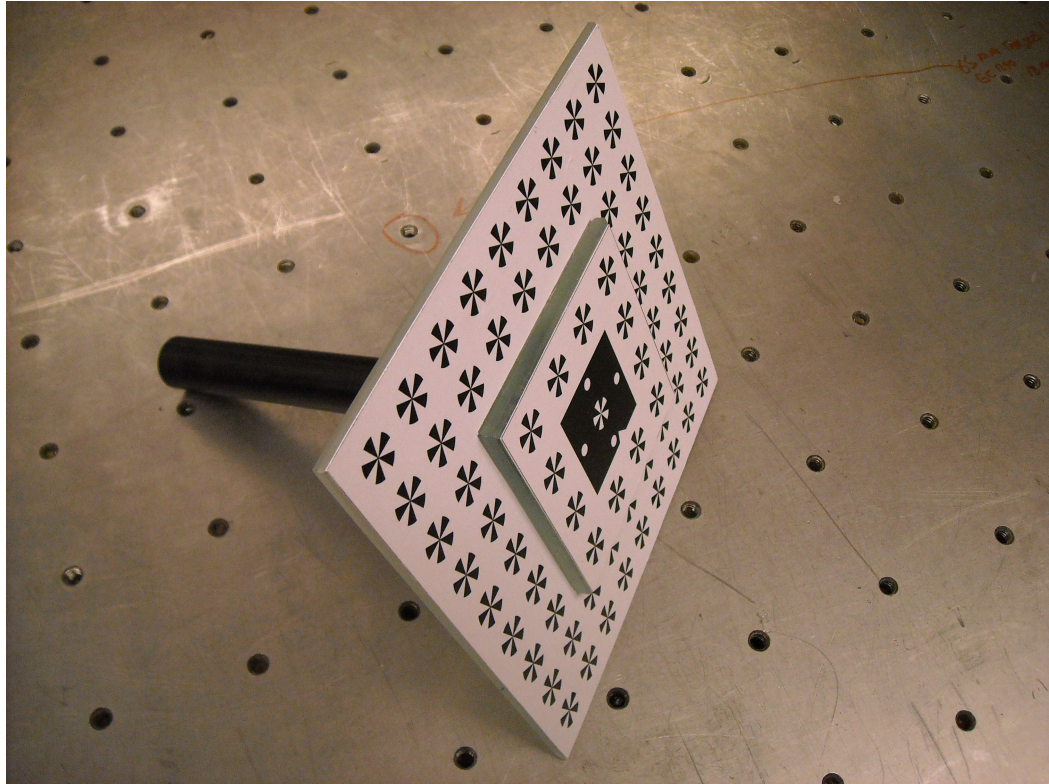


Figure 5.6: The 3D calibration tool with the raised middle area.

## 5.2 Camera Calibration with 2D and 3D Calibration Tools

### 5.2.1 Camera Calibration Experiment Setup and Data Collection

#### Experiment setup

Data sets used in this thesis were collected by the camera unit with a 2D calibration tool and 3D calibration tool shown in figures 5.2, 5.5 and 5.6. The system is used to calibrate

35. Measured by a digital caliper. This value is offered during camera calibration calculation as one of those initial values for optimization.

motion tracking systems for prospective motion correction in MRI. The setup of camera unit and calibration tool is shown in figure 5.7.

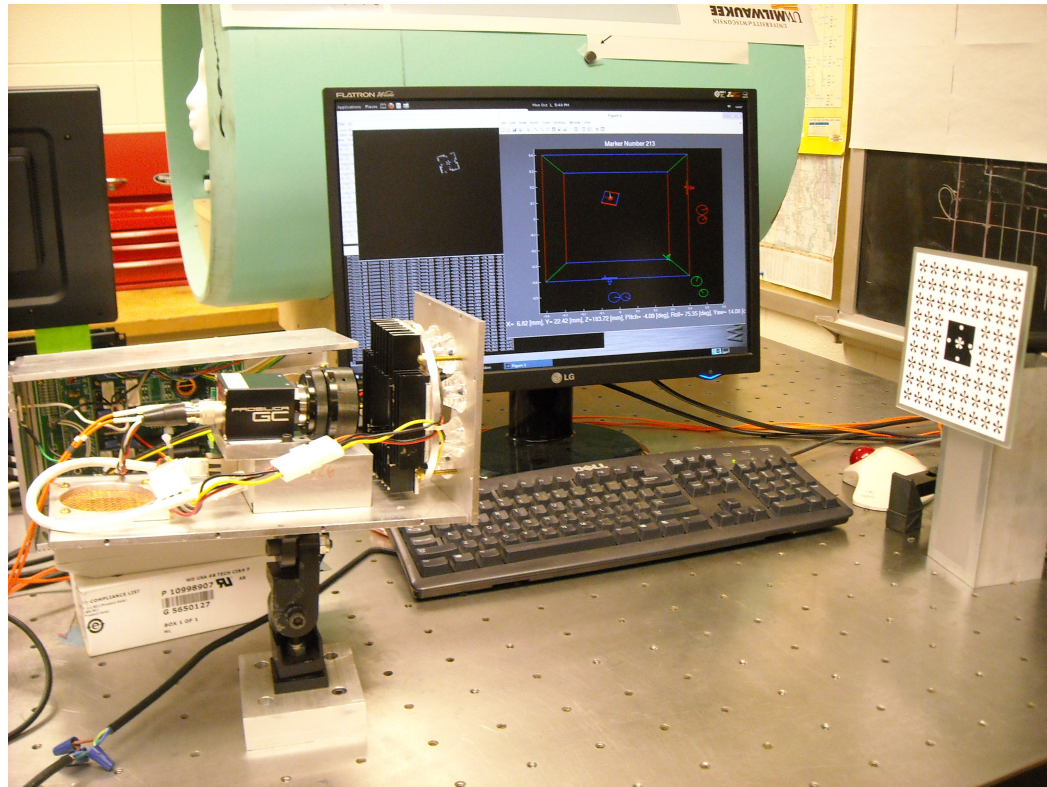


Figure 5.7: Camera unit (Left), Digital images displayed by PC monitor (Middle) and 2D calibration tool (Right).

The camera unit contains a digital high performance industrial camera—Prosilica GC1290 from Allied Vision Technologies with an Edmund 18mm lens and an LED lighting section. The camera is a  $1280 \times 960$  resolution mega pixel CCD camera and incorporates a high-quality Sony EXview HAD CCD sensor. The pixel size is  $3.75 \times 3.75 \mu\text{m}$ . The maximum frame rate at full resolution is 32 fps. The lighting part offers extra lighting for calibration tool observation by camera sensor.

During the experiment, the user moves the calibration tool by following a certain procedure to collect images of the calibration tool at different desired positions. Digital images will be recorded by an optical cable connection between the camera unit and computer. The user can see what the camera has recorded in LCD monitor and may adjust

the calibration tool to the desired position in 3D space.

### **Data collection strategy**

Currently, the collection of 200 images for each camera calibration is performed. Figures 5.8 and 5.9 show the motion strategy of the calibration tool in 3D space. The same motion strategy is used for both the 3D and 2D calibration tools. Figure 5.8 shows that, during image collection stage, the 3D calibration tool is tilted towards 5 directions of the camera: the front, the top, the bottom, the right and the left in the position closer to camera unit within the FOV of the camera unit. Then the direction adjustment is repeated by moving backward within the FOV of the camera unit. Figure 5.9 shows that the 3D calibration tool is also rotated at  $\kappa$  angle<sup>36</sup> by  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$ . Every time after rotating the  $\kappa$  angle, the user needs to repeat the movement demonstrated in figure 5.8.

---

36.  $\kappa = 0^\circ$  is the initial position shown in figure 5.8.

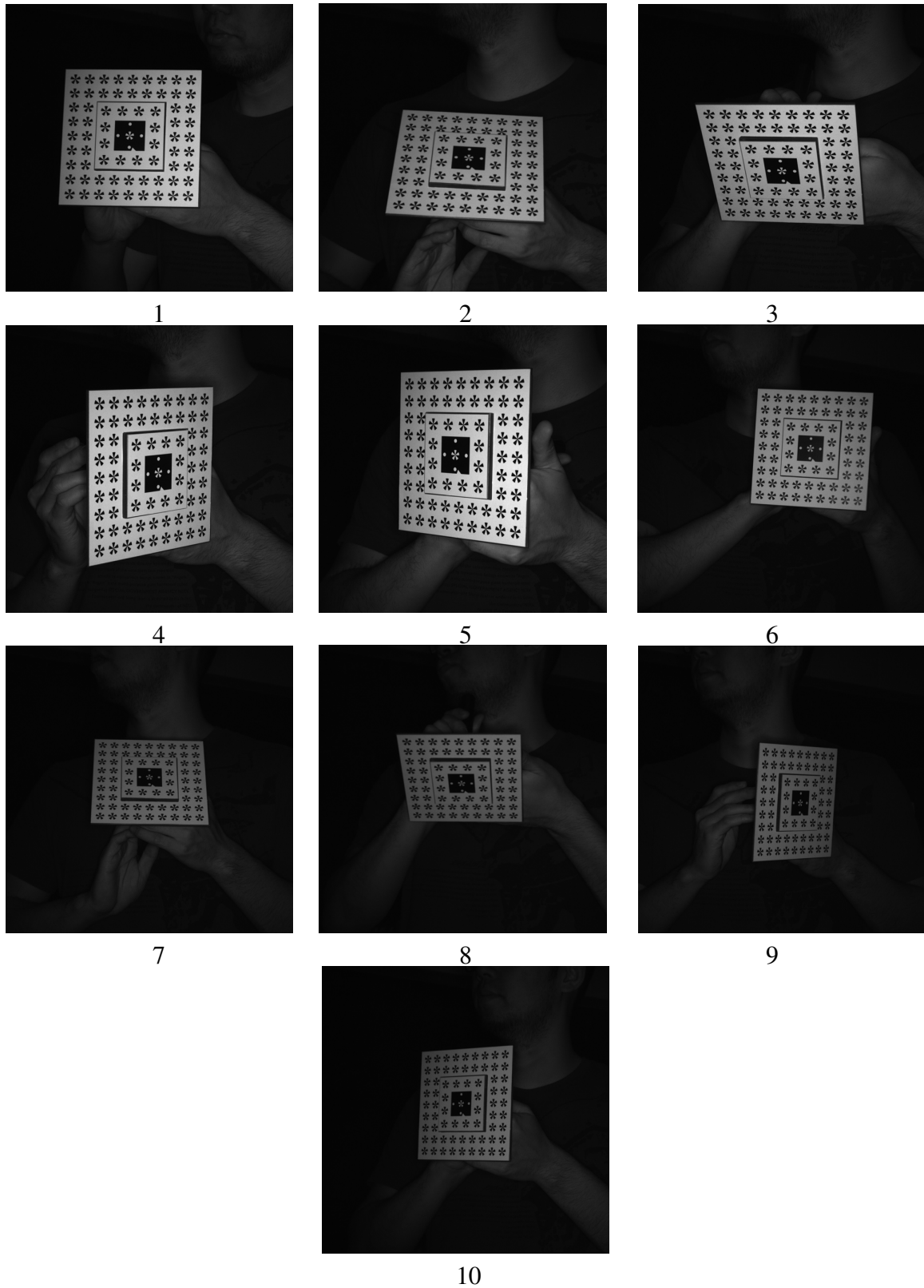


Figure 5.8: Digital images of the 3D calibration tool at various positions: 1, 2, 3, 4, and 5 are in the front with the tool tilting towards different directions; 6, 7, 8, 9, and 10 are in the rear with the tool tilting towards different directions.



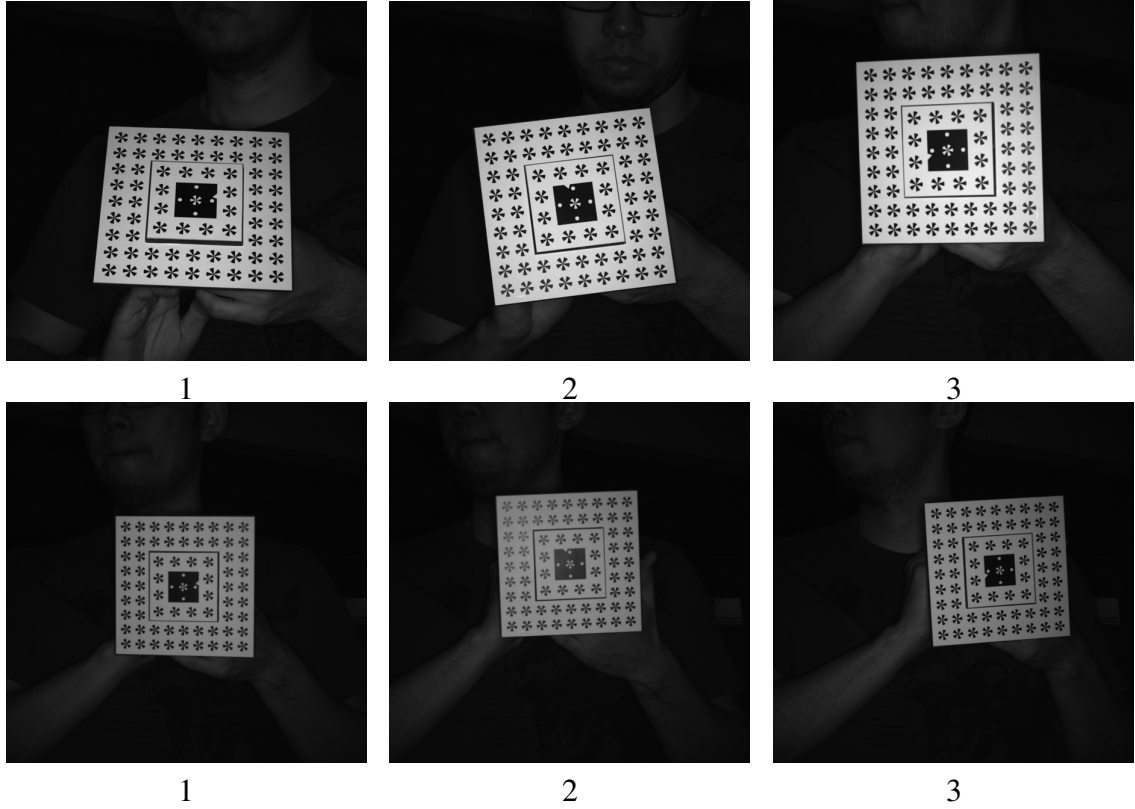


Figure 5.9: Digital images of the 3D calibration tool with  $\kappa$  angle rotation at  $90^\circ$ ,  $180^\circ$  and  $270^\circ$ : 1, 2, and 3 are collected in the front position; 4, 5, and 6 are collected in the rear position.

### 5.2.2 Image Processing in Camera Calibration

The image sets collected by following the above pattern are processed by the camera calibration software system. The user runs image processing first for processing the digital images—the extraction of  ${}^pP_a \in R^2$  for each landmark from each image, which can be as many as  $200 \times 73 \times 2 = 29200$  measurement values for the 2D calibration tool or  $200 \times 69 \times 2 = 27600$  measurement values for the 3D calibration tool. Next, the measured  ${}^pP_a = \bar{Y}$  and initial values of camera IO parameters are used as input for the bundle adjustment algorithm to do camera calibration, which can estimate camera IO, EO and lens distortion parameters simultaneously. Regarding the workload of each camera calibration by bundle adjustment, the above setup can be taken as an example. There are 200 images of distinct calibration tool positions. Each image contains 73 landmarks for the 2D calibration

tool, or 69 landmarks for the 3D calibration tool. Thus, the algorithm needs to estimate 10 camera IO plus  $200 \times 6$  camera EO plus  $73 \times 3$  landmark positions for a 2D calibration tool or  $69 \times 3$  landmark positions for a 3D calibration tool, which equals 1422 parameters. Particularly, a total of 1417 parameters need to be estimated, after subtraction of the 7 parameters mentioned by Ganshaw [25]. After image processing, a typical distribution of  ${}^pP_a$  in the field of view (FOV) of a camera calibration data set is shown in figures 5.10 and 5.11. The image set of the 2D and 3D calibration tools contains around 200 usable images, which almost cover the FOV of the digital camera, as shown in figures 5.11 and 5.10.

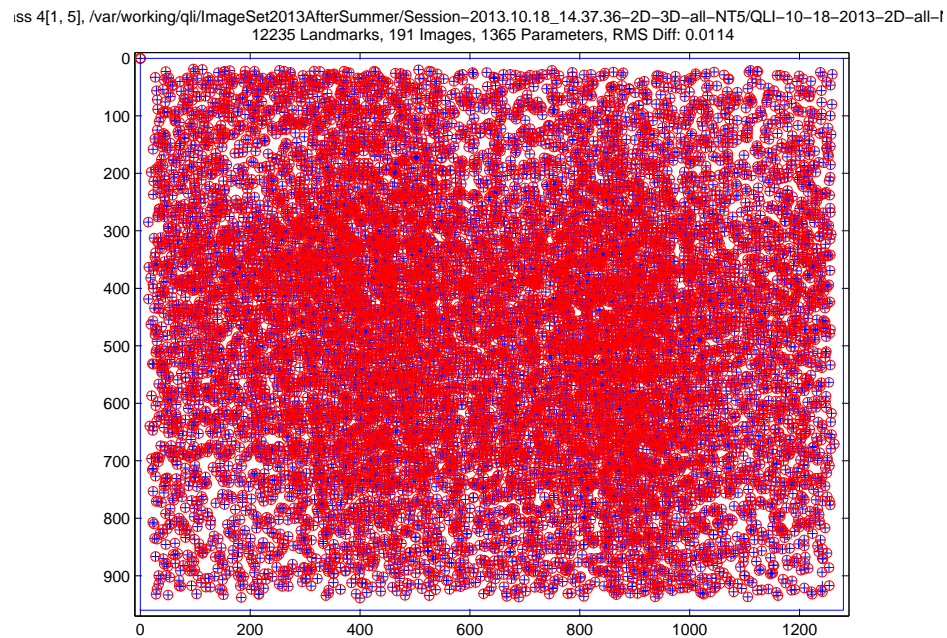


Figure 5.10: Distribution of landmarks of a image set: Data set was collected with Prosilica GC1290 camera and Edmund 18mm lens with the 2D calibration tool.

iss 4[1, 5], /var/working/qli/ImageSet2013AfterSummer/Session-2013.10.18\_14.37.36-2D-3D-all-NT5/QLI-10-18-2013-3D-all-I  
12116 Landmarks, 203 Images, 1425 Parameters, RMS Diff: 0.0149

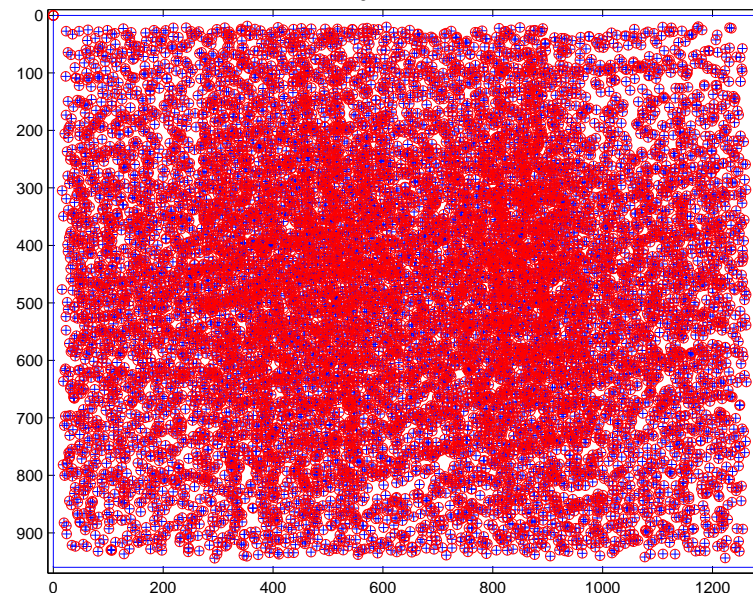


Figure 5.11: Distribution of landmarks of a image set: Data set was collected with Prosilica GC1290 camera and Edmund 18mm lens with the 3D calibration tool.

The diversity of the camera poses captured in one camera calibration data set is shown in figures 5.12, 5.13, 5.14, and 5.15. For the 2D calibration tool, figure 5.12 shows groups of the camera poses acquired by rotating  $\kappa$  angle at  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$ ; Figure 5.13 shows groups of camera poses acquired by adjusting the range from the camera to the calibration tool. Similarly, for the 3D calibration tool, figure 5.14 includes groups of camera poses found by rotating  $\kappa$  angle at  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$  and figure 5.15 includes groups of the camera poses found by adjusting the range from the camera to the calibration tool. Figures 5.10, 5.12, and 5.13 indicate that the collection of around 200 images is enough to cover possible camera poses with a 2D calibration tool. Figures 5.11, 5.14, and 5.15 describe the same situation with the 3D calibration tool.

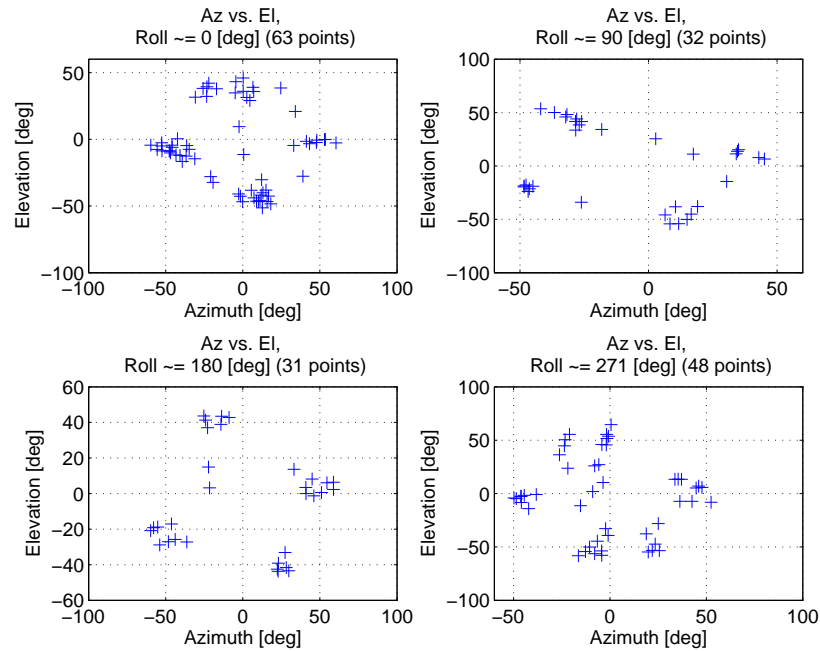


Figure 5.12: Diversity of camera poses in experimental camera calibration set: Data set was collected with Prosilica GC1290 camera and Edmund 18mm lens, azimuth and elevation v.s.  $\phi$  angle with the 2D calibration tool.

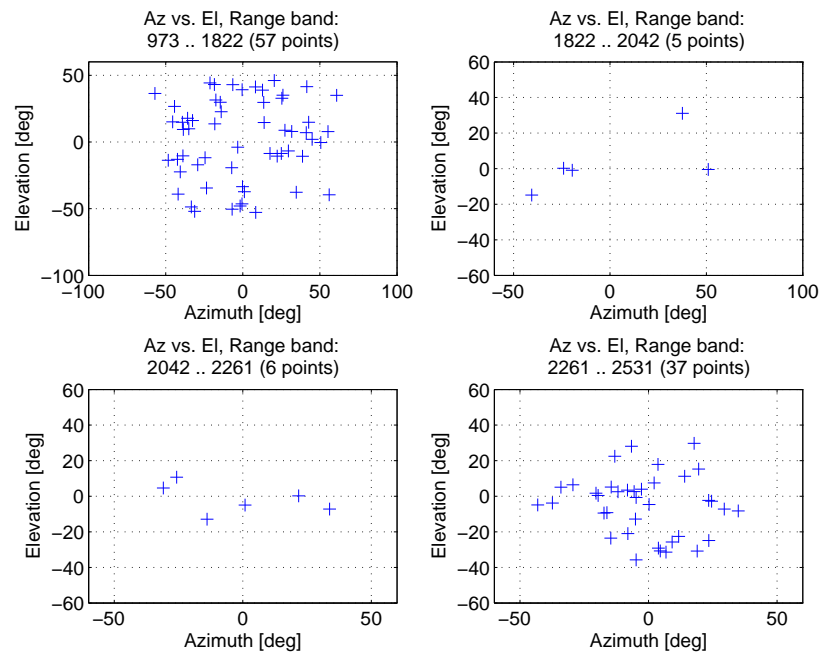


Figure 5.13: Diversity of camera poses in experimental camera calibration set: Data set was collected with Prosilica GC1290 camera and Edmund 18mm lens, azimuth and elevation v.s. range with the 2D calibration tool.

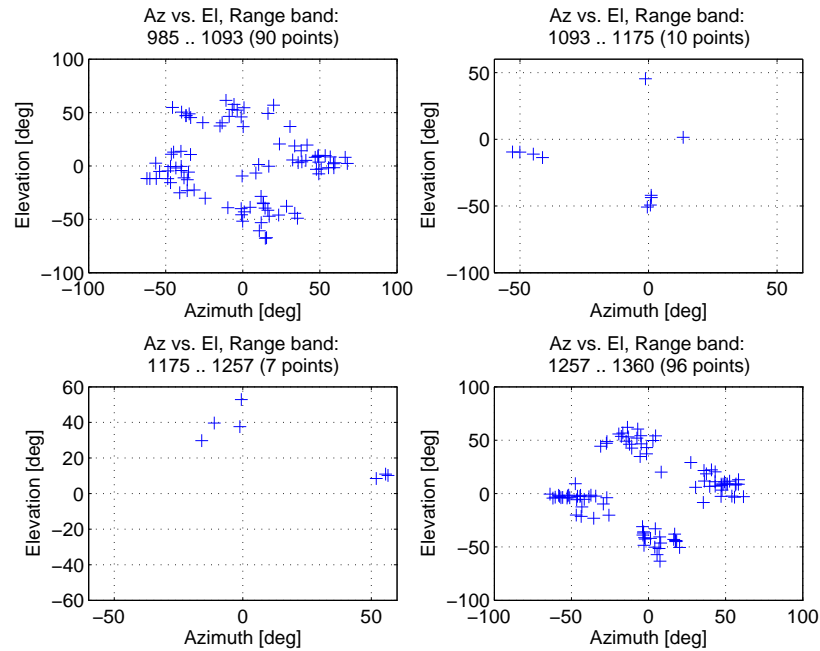


Figure 5.14: Diversity of camera poses in experimental camera calibration set: Data set was collected with Prosilica GC1290 camera and Edmund 18mm lens, azimuth and elevation v.s.  $\kappa$  angle with the 3D calibration tool.

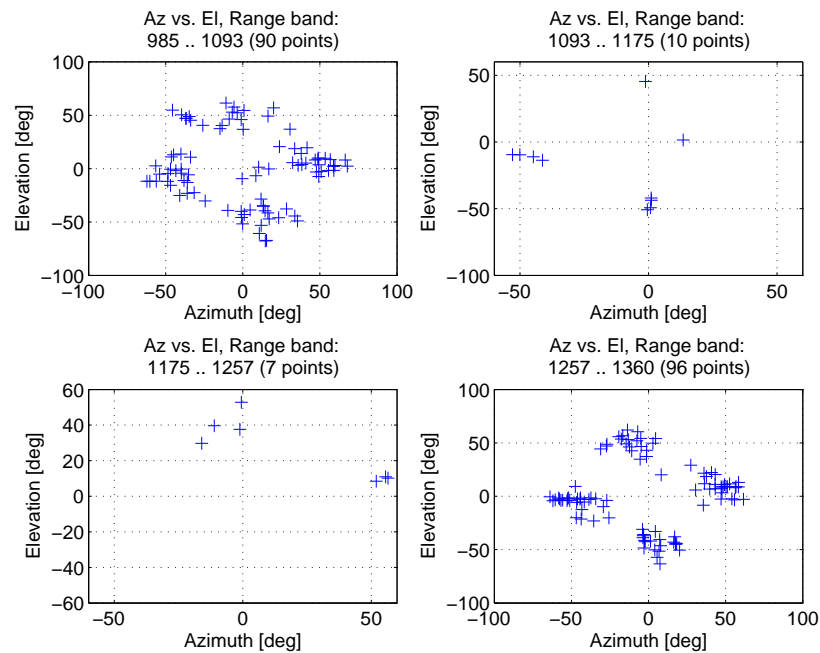


Figure 5.15: Diversity of camera poses in experimental camera calibration set: Data set was collected with Prosilica GC1290 camera and Schneider 18mm lens, azimuth and elevation v.s. range with the 3D calibration tool.

## 5.3 Camera Calibration Results Analysis

### 5.3.1 Data Collection and Camera Calibration Results

The setup of data collection is shown in section 5.2.1. By using the same hardware configuration and setup focus of the lens around 1 meter, 8 image sets are collected with the 2D calibration tool with the depth change between front and rear positions, and 8 image sets are collected with the 3D calibration tool with the depth change between the front and rear positions. Using the same strategy, 8 image sets by the 2D calibration tool and 8 image sets by the 3D calibration tool are collected when setting focus of the lens around 2 meters. After the bundle adjustment optimization of all the image sets, there is a total of 32 estimated camera EO and IO based on each image set. Tables 5.1 and 5.2 list the estimated camera IO parameters for the 2D and 3D calibration tools with focus of the lens around 1 meter.

### 5.3.2 $\hat{\chi}^*$ , the Estimation of Ideal Camera Model

The confident good camera calibration model  $\hat{\chi}^*$  is found using the  $\bar{D}_{i,j}$  matrix as described above. Based on a  $\bar{D}_{i,j}$  matrix, the camera model with the smallest root mean square distance of all the estimated camera models is considered as the  $\hat{\chi}^*$ . Among the image sets collected with focus of the lens at around 1 meter in section 5.3.1, the  $\hat{\chi}^*$  is identified as the camera calibration result of the 4<sup>th</sup> image set by the 3D calibration tool. Note that there will be a different  $\hat{\chi}^*$  for the image sets collected with focus of the lens around 2 meters.

### 5.3.3 Experimental Analysis of 2D and 3D Camera Calibration Tools

After finishing the camera calibration calculation for the 16 image sets collected with focus of the lens around 1 meter, the  $\bar{D}_{2D}$  and  $\bar{D}_{3D}$  are calculated. The  $\bar{D}_{i,j}$  matrix and  $\hat{\chi}^*$  are calculated as mentioned in section 5.3.2. Correspondingly, table 5.3 lists the  $\bar{D}_{2D}$ .

	Set1	Set2	Set3	Set4	Set5	Set6	Set7	Set8
$c_p$	18.762	18.769	18.772	18.767	18.766	18.764	18.765	18.760
$k_x$	266.667	266.667	266.667	266.667	266.667	266.667	266.667	266.667
$k_y$	266.687	266.691	266.684	266.690	266.682	266.693	266.687	266.709
$x_0$	646.698	654.326	656.079	650.591	655.028	653.195	653.037	651.617
$y_0$	490.055	488.571	484.932	489.939	486.787	487.032	486.807	487.533
$K_1 (\times 10^{-4})$	2.900	2.906	2.905	2.902	2.895	2.887	2.905	2.895
$K_2 (\times 10^{-7})$	-6.310	-13.926	-29.698	-21.330	-22.984	-16.168	-18.308	-7.277
$K_3 (\times 10^{-7})$	-1.319	-0.483	1.209	0.372	0.644	0.100	0.152	-1.036
$P_1 (10^{-4})$	2.687	-20.048	-21.960	-9.759	-22.078	-16.376	-16.989	-13.774
$P_2 (10^{-4})$	1.655	1.720	1.830	1.674	1.765	1.759	1.748	1.721
$P_3$	0	0	0	0	0	0	0	0

Table 5.1: Camera Calibration Results of the 2D camera calibration tool with focus of the lens around 1 meter.

	Set1	Set2	Set3	Set4	Set5	Set6	Set7	Set8
$c_p$	18.766	18.765	18.762	18.764	18.764	18.766	18.763	18.763
$k_x$	266.667	266.667	266.667	266.667	266.667	266.667	266.667	266.667
$k_y$	266.643	266.644	266.647	266.649	266.649	266.652	266.655	266.644
$x_0$	608.792	605.923	610.310	611.886	606.427	616.807	619.584	608.007
$y_0$	501.936	493.435	501.897	503.058	495.280	509.916	497.606	497.668
$K_1 (\times 10^{-4})$	2.845	2.903	2.916	2.918	2.877	2.886	2.854	2.795
$K_2 (\times 10^{-7})$	1.337	-18.831	-19.091	-14.935	-10.549	-10.467	-0.726	4.834
$K_3 (\times 10^{-7})$	-1.873	-0.076	0.084	-0.620	-0.855	-0.801	-1.707	-1.730
$P_1 (10^{-4})$	1.126	1.169	1.039	0.995	1.164	0.914	0.806	1.143
$P_2 (10^{-4})$	1.346	1.553	1.320	1.304	1.528	1.129	1.478	1.483
$P_3$	0	0	0	0	0	0	0	0

Table 5.2: Camera Calibration Results of the 3D camera calibration tool with focus of the lens around 1 meter.



2D	Set1	Set2	Set3	Set4	Set5	Set6	Set7	Set8
Set1	0	0.447	0.612	0.279	0.326	0.219	0.272	0.194
Set2	—	0	0.165	0.165	0.185	0.253	0.192	0.388
Set3	—	—	0	0.327	0.293	0.388	0.335	0.550
Set4	—	—	—	0	0.172	0.209	0.153	0.311
Set5	—	—	—	—	0	0.108	0.054	0.271
Set6	—	—	—	—	—	0	0.062	0.163
Set7	—	—	—	—	—	—	0	0.217
Set8	—	—	—	—	—	—	—	0

Table 5.3: The upper triangular part of the symmetric  $\bar{D}_{2D}$  of the 2D calibration tool with focus of the lens around 1 meter.

Table 5.4 lists the  $\bar{D}_{3D}$ .

3D	Set1	Set2	Set3	Set4	Set5	Set6	Set7	Set8
Set1	0	0.159	0.222	0.154	0.182	0.217	0.311	0.178
Set2	—	0	0.295	0.227	0.053	0.374	0.288	0.113
Set3	—	—	0	0.139	0.249	0.319	0.308	0.188
Set4	—	—	—	0	0.207	0.180	0.199	0.203
Set5	—	—	—	—	0	0.395	0.237	0.061
Set6	—	—	—	—	—	0	0.281	0.377
Set7	—	—	—	—	—	—	0	0.217
Set8	—	—	—	—	—	—	—	0

Table 5.4: The upper triangular part from the symmetric  $\bar{D}_{3D}$  of the 3D calibration tool with focus of the lens around 1 meter.

Then, the root mean square values for the two  $\bar{D}$  matrices in tables 5.3 and 5.4 are computed and shown in table 5.5.

Experimental Results	RMS of $\bar{D}$ (pixel)	Improvement Percentage
$\bar{D}_{2D}$	0.291	—
$\bar{D}_{3D}$	0.242	16.87%

Table 5.5: Statistics of  $\bar{D}$  based on the 2D and 3D calibration tools, 8 image sets for each calibration tool with focus of the lens around 1 meter.

Based on the above table, according to image sets collected under focus of the lens around 1 meter, the 3D calibration tool offers the camera calibration result with accuracy improved by around 16%. The camera calibration results of the 3D calibration tool are

closer to the ideal  $\chi^*$  than the camera calibration results of the 2D calibration tool. The improvement of the camera calibration results accuracy are repeatable based on different image sets collected by different individuals.

By the same calculation, table 5.6 is of image sets collected under focus of lens around 2 meter. As can be seen, the improvement of the the 3D calibration tool over the 2D calibration tool is 26.91%.

Experimental Results	RMS of $\bar{D}$ (pixel)	Improvement Percentage
$\bar{D}_{2D}$	0.470	—
$\bar{D}_{3D}$	0.343	26.91%

Table 5.6: Statistics of  $\bar{D}$  based on the 2D and 3D calibration tools, 8 image sets for each calibration tool with focus of the lens around 2 meters.

The other 16 image sets (8 for each type calibration tool) are collected by focus of the lens around 2 meters on another day, which offers a different results. As shown in table 5.7, the improvement performance of the 3D calibration tool over the 2D calibration tool is around 3%. Thus, the performance of the 3D and the 2D calibration tools are quite close according to this particular experiment.

Experimental Results	RMS of $\bar{D}$ (pixel)	Improvement Percentage
$\bar{D}_{2D}$	0.538	—
$\bar{D}_{3D}$	0.523	2.79%

Table 5.7: Statistics of  $\bar{D}$  based on the 2D and 3D calibration tools, 8 image sets for each calibration tool with focus of the lens around 2 meter.

With further experiment, the unstable results with setting focus of the lens around 2 meter appears inconsistent. The unrepeatable results of improvement are believed to be generated by motion blur in the digital images. With a longer distance of focus situation, camera sensor is easier to capture motion blur when the user moves the camera calibration tool in 3D space, since the longer focus of the lens means a larger  $f/\#$  and slower shutter speed.

### 5.3.4 Mathematical Analysis of 2D and 3D Camera Calibration Tools

Mathematical analysis is used to simulate camera poses of the 3D calibration tool based on camera poses of the 2D calibration tool. This would reduce other possible factors<sup>37</sup> that would impact the camera calibration results, and thus focus on the analysis of the performance improvement according to the different designs between the 3D and 2D calibration tools.

#### **Covariance matrices analysis: camera calibration estimation**

Figures 5.16, 5.17, and 5.18 offer us the performance analysis of the 2D calibration tool and the 3D calibration tool by simulating the same camera poses for both tools. Figure 5.16 is the difference of standard deviation between the 3D calibration tool and the 2D calibration tool during camera calibration. Figure 5.17 shows the standard deviation values in the 3D calibration tool and the 2D calibration tool corresponding to each camera pose in 6 DOF, represented in spherical coordinate system. Figure 5.18 compares the change of standard deviation values in the 3D calibration tool and the 2D calibration tool corresponding to each camera pose in 6 DOF, represented in spherical coordinate system. All three figures indicate that the 3D calibration tool offers less uncertainty in the camera calibration optimization.

---

37. For example, it is not piratical to have two image sets that contain exact same camera poses collected by the 2D and 3D calibration tools. However, simulated the 3D calibration tool from the 2D calibration tool can achieve this.

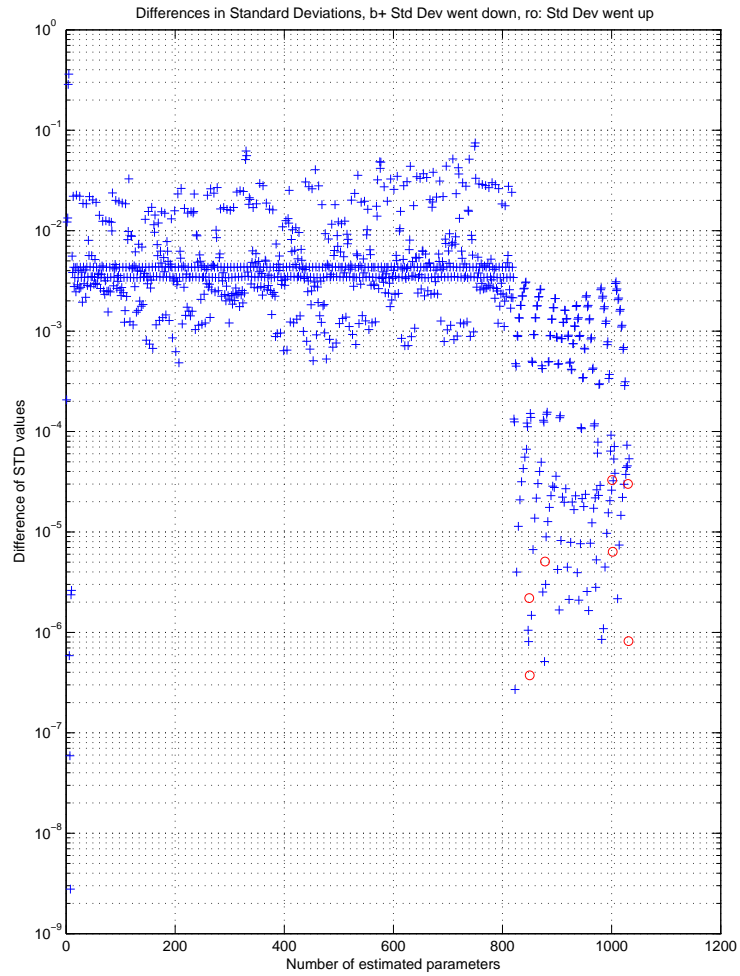


Figure 5.16: Difference values of STD of the 3D and the 2D calibration tools during camera calibration: a blue cross means the 3D calibration tool makes the standard deviation of camera calibration smaller than the 2D calibration tool; while a red circle means the 2D calibration tool makes STD smaller than the 3D calibration tool.

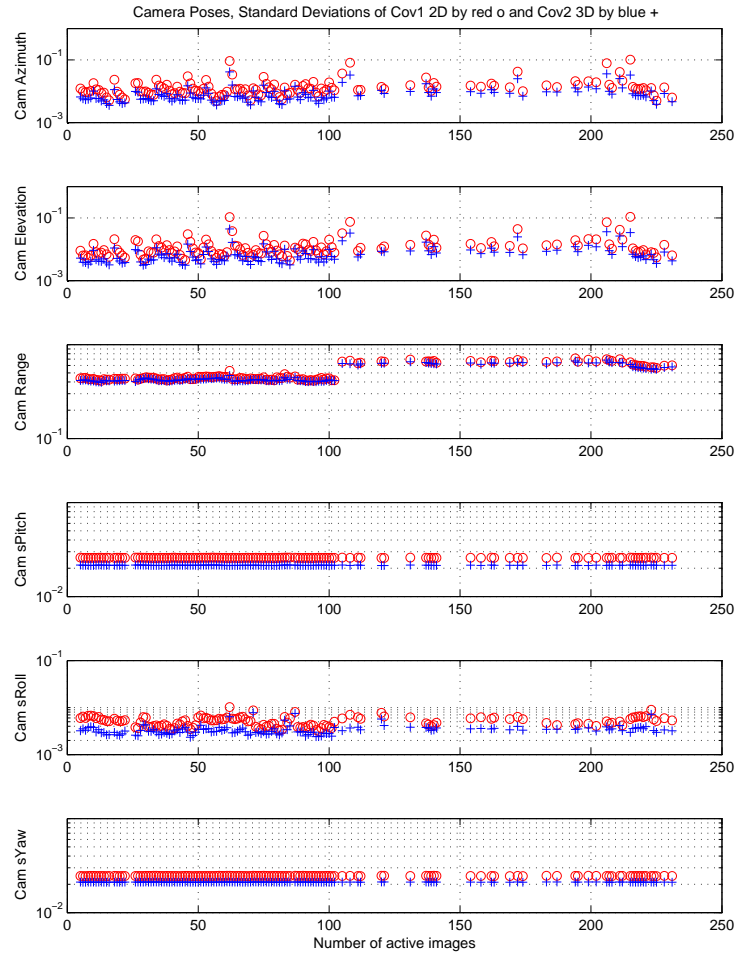


Figure 5.17: STD values of the 3D and 2D calibration tools corresponding to camera poses in 6 DOF: a blue cross is the STD values of the 3D calibration tool; a red circle is the STD values of the 2D calibration tool.

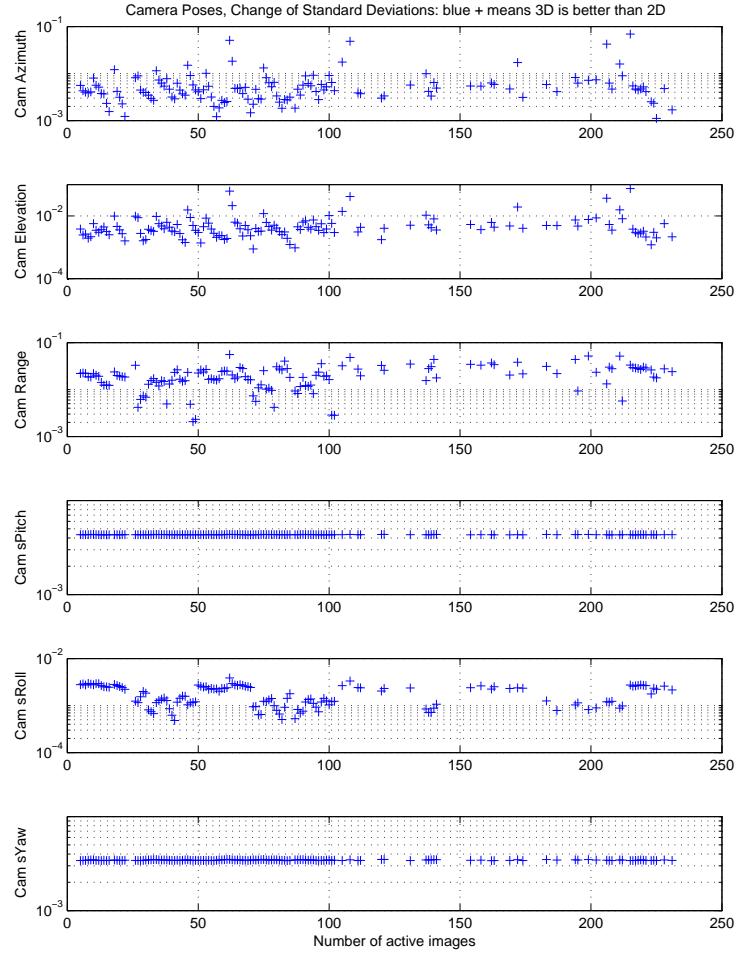


Figure 5.18: Difference of STD of the 3D and 2D calibration tools corresponding to camera poses in 6 DOF: a blue cross means the 3D calibration tool makes the STD of camera calibration smaller than the 2D calibration tool; while a red circle means the 2D calibration tool makes STD smaller than the 3D calibration tool.

### Covariance matrices analysis: camera calibration results

The idea is to simulate the covariance matrix of the 3D calibration tool and generate the camera calibration estimation. The calculation starts from the covariance matrix generated by the camera calibration of the 2D calibration tool image sets. We are looking for the camera model of each camera calibration result  $\chi_j$ :

$$\chi_j = f(\chi^*, Cov_{\chi_j}), \quad (5.1)$$

such that

$$Cov_{\chi_j} = E\{(\chi_j - \chi^*) \cdot (\chi_j - \chi^*)^T\}, \quad (5.2)$$

where  $\chi^*$  is the ideal camera model as the reference;  $\chi_j$  is the estimated camera model corresponding to each image set; and  $j = 1, 2, \dots$  is the index. In the experimental data collection,  $j = 1, 2, \dots, 8$ , since eight 2D calibration tool image sets are collected;  $Cov_{\chi_j}$  is the covariance matrix of camera calibration corresponding to each image set.

To compute  $\chi_j$  is to simulate observed data form  $Cov_{\chi_j}$  as follows:

1. Construct a normal distributed matrix  $A$  whose dimension is  $12 \times \text{a large number}$ . In my computation,  $A$  is  $12 \times 10^5$
2. SVD decomposition of  $Cov_{\chi_j}$  to get

$$[U, \Sigma, V] = \text{svd}(Cov_{\chi_j}) \quad (5.3)$$

If matrix  $Cov_{\chi_j} \in C^{m \times n}$ , the SVD of  $Cov_{\chi_j}$  gives us

$$Cov_{\chi_j} = U \Sigma V^H = \sum_{i=1}^r \sigma_i u_i v_i^H,$$

where  $U = (u_1, \dots, u_m)$ ,  $V = (v_1, \dots, v_n)$  and  $\Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix}$ .

3. QR decomposition

$$[Q, R] = \text{qr}(\sqrt{\Sigma} \times V^T) \quad (5.4)$$

If  $B = \sqrt{\Sigma} \times V^T$ , then QR decomposition gives us

$$B = Q \begin{pmatrix} R \\ 0 \end{pmatrix},$$

where  $Q \in C^{m \times m}$  is an orthogonal matrix<sup>38</sup> and  $R \in C^{n \times n}$  is called R factor of matrix  $B$ , which is an upper triangular with non-negative diagonal entries.

4.  $R$  performs as a transformation matrix to introduce the covariance relationship contained in  $Cov_{\chi_j}$  into matrix  $A$ :

$$\tilde{\chi}_j = R^T \times A, \quad (5.5)$$

where  $\tilde{\chi}_j$  is the transformed data which contains the covariance information<sup>39</sup>.

5. Constructing the camera model  $\chi_j$ :

$$\hat{\chi}_j = \hat{\chi}^* + \tilde{\chi}_j. \quad (5.6)$$

By the Monte Carlo method, the 100 samples from  $\tilde{\chi}_j$  in equation (5.6) are selected to provide simulated  $\chi_j^1, \chi_j^2, \dots, \chi_j^k$ .  $\bar{D}_{i,j}$  matrix is calculated for each 3D calibration tool and 2D calibration tool:  $\bar{D}_{3D}$  and  $\bar{D}_{2D}$ . Next, the root mean square values of  $\bar{D}_{3D}$  and  $\bar{D}_{2D}$  are computed. Table 5.8 gives us the results:

Simulated Results	RMS of $\bar{D}$ (pixel)	Improvement Percentage
$\bar{D}_{2D}$	0.093	—
$\bar{D}_{3D}$	0.083	10.42%

Table 5.8: Improvement of  $\bar{D}$  based on the 2D and 3D calibration tools: simulated  $\tilde{\chi}_j$  corresponding to covariance matrices of camera calibration.

The RMS value of  $\bar{D}_{2D}$  and  $\bar{D}_{3D}$  indicates that for each image set, the simulated  $\bar{D}$  of the 2D and the 3D calibration tools is

$$\bar{D}_{3D} < \bar{D}_{2D}, \quad (5.7)$$

38. If  $Q$  is an orthogonal matrix, then  $Q^H Q = I$ .

39. Each column of the  $\tilde{\chi}_j$  is a sample of  $\tilde{\chi}$ .



which means that a camera calibration based on image sets of a 3D camera calibration tool introduces less correlation into estimation.

If we compare the improvement percentage values in tables 5.8 and 5.5 in section 5.3.3, the improvement of a 3D calibration tool is around 11%, which demonstrates the consistency of improvement between experimental result and simulated result.

## 5.4 Summary

In this chapter, a new 3D calibration tool is designed to collect observations for camera calibration. It is demonstrated that a carefully designed calibration tool will improve estimation results compared to the estimation from a 2D calibration tool. By analyzing the experimental results, data sets collected with the 3D calibration tool will lead to the improvement mathematically, higher calibration accuracy, as expected. By introducing the covariance matrices and the Monte Carlo method, the advantage of the 3D calibration tool is analyzed without other impact factors involved as much as possible. The analysis shows that the residual values generated by the 3D calibration tool is smaller than the covariance matrix generated by the 2D calibration tool, as expected. This result is repeatable when processed data sets are collected with focus of the lens around 1 meter, compared to setting focus of the lens around 2 meters. The 3D calibration tool is superior to the 2D calibration tool in its improvement of camera calibration accuracy.

## 6 Final Chapter and Future Work

This research makes three primary contributions: (i) development of singularity of representation definition and confirmation of new conditions for singularity in spherical coordinate representation in application; (ii) application of the robust estimators in bundle adjustment, experimental and theoretical evaluation of the performance of robustified estimators; (iii) design and validation of a new 3D calibration tool. These are enhancements to high-accuracy camera calibration technology.

### 6.1 Key Findings of This Research

The first contribution of this thesis is a formal definition of singularity of representation that is offered mathematically and goes beyond a description of singularity. The spherical coordinate system is introduced as a representation and is used to represent camera pose during camera calibration. This coordinate representation method does contain singularities with respect to camera FOV. This is demonstrated mathematically, laying the foundation for avoiding singularities during camera calibration.

The second contribution is an analysis of the characteristics of the least squares and robustified estimators. Classical estimators, with the normal distribution assumption, can be strongly influenced by outliers, giving inaccurate results. In photogrammetry, thousands of data points are collected for bundle adjustment. The outliers in data sets will impact the camera calibration accuracy, which will fail to satisfy the application requirements. Robust estimators can de-weight outliers that cannot be modeled using the classical estimator. The good fit to the observations by robust estimators provides reliable camera calibration results. By applying specific robustified estimator to optimization of bundle adjustment, the robustified method is proven to significantly improve camera calibration accuracy.

The third part is the design and application of a 3D calibration tool for data collection. 16 image sets are collected, 8 for the 3D calibration tool and 8 for the 2D calibration tool.

The experimental analysis is based on camera calibration results and the corresponding  $\bar{D}$  matrix, which shows that the 3D calibration tool gives us accuracy improvement. The mathematical analysis is based on simulated results generated by a covariance matrix of camera calibration, which shows that the 3D calibration tool gets similar accuracy improvement to the 2D calibration tool compared to experimental analysis.

Moreover, I introduce  $\bar{D}$ , a new metric created to measure and evaluate the distance between camera calibrations. The utilization of this quantity offers a way to evaluate the performance of robustified estimator, find the confident good camera calibration result as reference, and evaluate the performance of calibration tool for data collection and camera calibration computing.

## 6.2 Future Work

This research can be expanded to several sub-topics. Regarding singularity of representation situations, it is important to check the impact of the singularity cases on the processing system. For example, if camera calibration data sets contain a singularity, the impact on the accuracy of the camera calibration results should be determined, and the degradation of optimization in bundle adjustment needs to be addressed. Also, the application should be enhanced to find the data with singularity and eliminate them, or the system can remind users that they hit the particular conditions of singularity in those positions in the data collection stage. This will reduce the likelihood of difficulties in optimization calculation during bundle adjustment.

In the robustified optimization step, the algorithm can be optimized towards the efficiency of the robustified calculation. An adaptive algorithm that can adjust itself according to analysis of experimental camera calibration data will enhance the performance and reliability of bundle adjustment. For example, it will be more optimal that the adaptive algorithm can automatically tune the parameters of the robustified object function based on the experimental camera calibration data. In chapter four, I demonstrate that tuning of the

parameters has the potential to improve the estimation further. This will gain us the best result according to specific experimental camera calibration data.

As for the 3D calibration tool, the work can be expanded to investigate the performance of other landmarks layout design. For example, in my current development, there is a raised area in the center of the 3D calibration tool; it would be interesting to design a 3D calibration tool with raised border area instead of central area. Then, we could compare the performance of the two versions of the 3D calibration tool. Moreover, since the 3D calibration tool introduces extra constraint compared to a 2D calibration tool, it would also be meaningful to investigate if camera calibration based on experimental data sets with a 3D calibration tool would need fewer images to achieve the same accuracy. This would reduce the time-cost of data collections and accelerate camera calibration calculation speed for the users.

## References

- [1] A. Y. Abdel and H. M. Karara. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *American Society of Photogrammetry Symposium on Close-range Photogrammetry*, pages 1–18, 1971.
- [2] S. Abraham and T. Hau. Towards autonomous high-precision calibration of digital cameras. *Proceedings of SPIE Videometrics V*, 3174:82–93, 1997.
- [3] H. A. Beyer. *Geometric and Radiometric Analysis of a CCD-Camera Based Photogrammetric Close-Range System*. PhD thesis, ETH-Zurich, 1992.
- [4] B. L. Bowerman, R. O’Connell, and A. Koehler. *Forecasting, Time Series, and Regression (4th Edition)*. South-Western College, 2004.
- [5] D. C. Brown. Decentering distortion of lenses. *Photogrammetric Engineering*, XXXII(3):444–462, 1965.
- [6] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.
- [7] S. R. Buss and J. S. Kim. Selectively damped least squares for inverse kinematics. *Journal of Graphics Tools*, 10(3):37–49, 2005.
- [8] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–139, 1990.
- [9] L. Chen, C. W. Armstrong, and D. D. Raftopoulos. An investigation on the accuracy of three-dimensional space reconstruction using the direct linear transformation technique. *Journal of Biomechanics*, 27(4):493–500, 1994.

- [10] L. Chiaria, U. D. Croce, A. Leardini, and A. Cappozzo. Human movement analysis using stereophotogrammetry - part 2: Instrumental errors. *Gait and Posture*, 21(2):197–211, 2005.
- [11] S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. *Robotics and Automation, IEEE Transactions on*, 13(3):398–410, 1997.
- [12] T. A. Clarke and J. G. Fryer. The development of camera calibration methods and models. *The Photogrammetric Record*, 16(91), 1998.
- [13] M. A. R. Cooper and S. Robson. Theory of close range photogrammetry. In Keith B. Atkinson, editor, *Close range photogrammetry and machine vision*, pages 9–51. Whittles, 1996.
- [14] J. J. Craig. *Introduction to Robotics : Mechanics and Control (3rd Edition)*. Prentice Hall, 2005.
- [15] J. S. Day, D. J. Murdoch, and G. A. Dumas. Calibration of position and angular data from a magnetic tracking device. *Journal of Biomechanics*, 33(8):1039–1045, 2000.
- [16] K. Edmundson and C. S. Fraser. A practical evaluation of sequential estimation for vision metrology. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(5):272–285, 1998.
- [17] O. D. Faugeras and G. Toscani. Camera calibration for 3d computer vision. *Int. Workshop on Machine Vision and Machine Intelligence*, pages 240–247, 1987.
- [18] C. S. Fraser. Limiting error propagation in network design. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(5):487–493, 1987.

- [19] C. S. Fraser. Photogrammetric measurement to one part in a million. *Photogrammetric Engineering and Remote Sensing*, 58(3):305–310, 1992.
- [20] C. S. Fraser. Network design. In Keith B. Atkinson, editor, *Close range photogrammetry and machine vision*, pages 256–279. Whittles, 1996.
- [21] C. S. Fraser. Digital camera self-calibration. *ISPRS Journal of Photogrammetry and Remote Sensing*, 52(4):149–159, 1997.
- [22] J. G. Fryer. Camera calibration. In Keith B. Atkinson, editor, *Close range photogrammetry and machine vision*, pages 156–175. Whittles, 1996.
- [23] G. Ganci and M. R. Shortis. A comparison of the utility and efficiency of digital photogrammetry and industrial theodolite systems. *ISPRS Congress Technical Commission V: Close Range Techniques and Machine Vision*, 31(5):182–187, 1996.
- [24] C. Gosselin and J. Angeles. Singularity analysis of closed-loop kinematic chains. *Robotics and Automation, IEEE Transactions on*, 6(3):281–290, 1990.
- [25] S. I. Granshaw. Bundle adjustment methods in engineering photogrammetry. *The Photogrammetric Record*, 10(56):181–207, 1980.
- [26] A. Gruen. Precision and reliability aspects in close range photogrammetry. *Photogrammetric Journal of Finland*, 8(2):117–132, 1981.
- [27] J. V. Harrison. An algorithm providing all-attitude capability for three-gimballed inertial systems. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-7(3):532–543, 1971.
- [28] R. I. Hartley. Euclidean reconstruction from uncalibrated views. In D. Forsyth J. L. Mundy, A. Zisserman, editor, *Applications of Invariance in Computer Vision*, volume 825, pages 187–202. Springer, 1993.

- [29] R. I. Hartley. An algorithm for self calibration from several views. *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94, IEEE Computer Society Conference on*, pages 908–912, 1994.
- [30] R. I. Hartley. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(6):580–593, 1997.
- [31] R. I. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 761–764. 1992.
- [32] T. L. Hedrick. Software techniques for two- and three-dimensional kinematic measurements of biological and biomimetic systems. *Bioinspiration and Biomimetics*, 3(3):034001, 2008.
- [33] J. Heikkila. Geometric camera calibration using circular control points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10):1066–1077, 2000.
- [34] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. *Computer Vision and Pattern Recognition, 1997 IEEE Computer Society Conference on*, pages 1106–1112, 1997.
- [35] P. J. Huber and E. M. Ronchetti. *Robust Statistics*. Wiley, 2009.
- [36] Y. Kunii and H. Chikatsu. On the application of 3 million consumer pixel camera to digital photogrammetry. *Proceedings of SPIE Videometrics VII*, 4039:278–287, 2001.
- [37] R. K. Lenz and R. Y. Tsai. Techniques for calibration of the scale factor and image center for high accuracy 3-d machine vision metrology. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(5):713–720, 1988.



- [38] T. Luhmann, S. Robson, S. Kyle, and I. Harley. Analytical methods. In *Close Range Photogrammetry: Principles, Techniques and Application*, pages 201–315. Wiley, 2007.
- [39] T. Luhmann, S. Robson, S. Kyle, and I. Harley. *Close Range Photogrammetry: Principles, Techniques and Applications*. Wiley, 2007.
- [40] T. Luhmann, S. Robson, S. Kyle, and I. Harley. Imaging technology. In *Close Range Photogrammetry: Principles, Techniques and Application*, pages 97–195. Wiley, 2007.
- [41] T. Luhmann, S. Robson, S. Kyle, and I. Harley. Mathematical fundamentals. In *Close Range Photogrammetry: Principles, Techniques and Application*, pages 31–95. Wiley, 2007.
- [42] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. Image formation. In *An Invitation to 3-D Vision: From Images to Geometric Models*, pages 36–56. Springer, 2010.
- [43] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer, 2010.
- [44] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. Reconstruction from two calibrated views. In *An Invitation to 3-D Vision: From Images to Geometric Models*, pages 77–107. Springer, 2010.
- [45] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. Representation of a three dimensional moving scene. In *An Invitation to 3-D Vision: From Images to Geometric Models*, pages 9–35. Springer, 2010.
- [46] A. A. Magrill. Variation in distortion with magnification. *JOSA Journal of The Optical Society of America*, 45(3):148–149, 1954.

- [47] R. A. Maronna, D. R. Martin, and V. J. Yohai. *Robust Statistics: Theory and Methods*. Wiley, 2006.
- [48] G. T. Marzan and H. M. Karara. A computer program for direct linear transformation solution of the collinearity condition, and some applications of it. In *Proceedings of the symposium on close-range photogrammetric systems*, pages 420–476. 1975.
- [49] P. Meer, D. Mintz, A. Rosenfeld, and D. Kim. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6(1):59–70, 1991.
- [50] T. Melen. *Geometrical modelling and calibration of video cameras for underwater navigation*. PhD thesis, Inst. for teknisk kybernetikk, Norges tekniske hogskole, 1994.
- [51] D. C. Montgomery, E. A. Peck, and G. G. Vining. *Introduction to Linear Regression Analysis (4th Edition)*. Wiley, 2007.
- [52] D. Nister. An efficient solution to the five-point relative pose problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6):756–770, 2004.
- [53] T. Rahman and N. Krouglicof. An efficient camera calibration technique offering robustness and accuracy over a wide range of lens distortion. *Image Processing, IEEE Transactions on*, 21(2):626–637, 2012.
- [54] C. S. Remondino, F. and Fraser. Digital camera calibration methods: considerations and comparisons. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVI(5):266–272, 2006.
- [55] M. Saadatseresht, C. S. Fraser, F. Samadzadegan, and A. Azizi. Visibility analysis in vision metrology network design. *The Photogrammetric Record*, 19(107):219–236, 2004.

- [56] Y. C. Shiu. Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form  $ax=xb$ . *Robotics and Automation, IEEE Transactions on*, 5(1):16–29, 1989.
- [57] H. Stewenius, C. Engels, and D. Nister. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294, 2006.
- [58] J. Stuelpnagel. On the parametrization of the three-dimensional rotation group. *SIAM*, 6(4):422–430, 1964.
- [59] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Vision algorithms: theory and practice*, volume 1883, pages 298–372. Springer Berlin Heidelberg, 2000.
- [60] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344, 1987.
- [61] R. Y. Tsai and R. K. Lenz. A new technique for fully autonomous and efficient 3d robotics hand-eye calibration. *Robotics and Automation, IEEE Transactions on*, 5(3):345–358, 1989.
- [62] C. C. Wang. Extrinsic calibration of a vision sensor mounted on a robot. *Robotics and Automation, IEEE Transactions on*, 2(8):161–175, 1992.
- [63] J. H. Wang, F. H. Shi, J. Zhang, and Y. C. Liu. A new calibration model and method of camera lens distortion. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5713–5718. IEEE, Beijing, 2006.

- [64] J. H. Wang, F. H. Shi, J. Zhang, and Y. C. Liu. A new calibration model of camera lens distortion. *Pattern Recognition*, 41(2):607–615, 2008.
- [65] J. Weng, P. Cohen, and M. G. Herniou. Camera calibration with distortion models and accuracy evaluation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(10):965–980, 1992.
- [66] Z. Y. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing Journal*, 15(1):59–76, 1997.
- [67] Z. Y. Zhang. Camera calibration with one-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.
- [68] Z. Y. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):892–899, 2004.

## CURRICULUM VITAE

Qiaotian LI

Place of Birth: Changchun, Jilin, China

Education:

B.Eng. with honors, Changchun University of Sci & Tech 2002–2006

Major: Electrical Engineering

Master, Changchun University of Sci & Tech 2006–2008

Major: Electrical Engineering

Dissertation Title: Enhancement to camera calibration: representation, robust statistics,  
and 3D Calibration Tool

Internship Experience:

MATLAB Graphics Engineering Internship, The MathWorks Inc., 2013

Teaching Experience:

EE 306–Introduction to Electrical Engineering, Spring 2009 to Fall 2013, Electrical  
Engineering Department, University of Wisconsin-Milwaukee

EE 305–Electrical Circuits II, Summer 2012, Electrical Engineering Department,  
University of Wisconsin-Milwaukee