CALIBRATING NON-OVERLAPPING CAMERAS WITH A LASER RAY

by

Wuhe Zou

A dissertation submitted in partial

fulfillment of the requirements

for the degree of

Doctor of Philosophy

Electrical and Electronic Engineering

Tottori University

March, 2015

Dissertation written by

Zou Wuhe

PH.D, Tottori University, Japan, 2015

TABLE OF CONTENTS

LIST OF FIGURES V

LIST OF TABLES IX

DEDICATION X

ABSTRACT XI

ACKNOWLEDGEMENTS XIV

CHAPTER 1 INTRODUCTION 1

1.1 BACKGROUND 1

1.2 CONTRIBUTION 2

1.3 ORGANIZATION 4

CHAPTER 2 RELATIVE RESEARCH 6

2.1 CALIBRATING CAMERAS WITH OVERLAPPING VIEW 6

2.2 CALIBRATING CAMERAS WITH NON-OVERLAPPING VIEW 7

2.3 STATE-OF-THE-ART METHOD FOR CALIBRATING NON-OVERLAPPING CAMERAS 8

CHAPTER 3 FUNDAMENTALS 11

3.1 PINHOLE CAMERA MODEL 11

3.2 INTRINSIC PARAMETERS 11

3.2.1 Perspective projection using homogeneous coordinates *13*

3.2.2 Principal-point offset *13*

3.2.3 Image-sensor characteristics *15*

3.2.4 Radial lens distortion *16*

3.3 EXTRINSIC PARAMETERS 19

3.3.1 Direct mapping (from world to camera coordinate systems) 19

3.3.2 Indirect mapping (from camera to world coordinate systems) 20

3.4 STEREOVISION SYSTEM 21

3.5 THE STEREO CORRESPONDENCE E PROBLEM 22

CHAPTER 4 ESTIMATING THE POSE OF THE LASER POINTER RELATIVE TO THE CALIBRATION PATTERN 25

4.1 NOTATION 25

4.2 PROBLEM DEFINITION 26

4.3 COMPUTATION OF LASER POINTER PARAMETERS 27

4.4 EXPERIMENTS 28

4.5 ESTIMATION OF THE COMPUTATIONAL ERROR 29

4.6 CONCLUSION AND CONTRIBUTION 30

CHAPTER 5 CALIBRATING NON-OVERLAPPING CAMERAS WITH ONE CALIBRATION PATTERN 31

5.1 QUATERNION FOR ROTATION 31

5.2 LEVENBERG-MARQUARDT METHOD FOR NONLINEAR OPTIMIZATION 34

5.3 PLÜCKER COORDINATES 37

5.4 COPLANAR CONSTRAINT METHOD 38

5.4.1 Problem definition 38

5.4.2 Process 40

5.4.3 Coplanar constraint for relative pose via a laser ray 40

5.4.4 Relative pose estimation 42

5.4.5 Parameters refinement 43

5.5 EXPERIMENTS AND RESULTS 43
5.5.1 Evaluation on synthetic data *43*
5.5.2 Evaluation on real-world data *48*
5.6 CONCLUSION AND CONTRIBUTION 50
CHAPTER 6 CALIBRATING NON-OVERLAPPING CAMERAS WITH TWO CALIBRATION PATTERNS 52
6.1 COLLINEAR CONSTRAINT METHOD 52
6.1.1 Problem definition *53*
6.1.2 Process *53*
6.1.3 Linear estimation of the relative pose *54*
6.1.4 Nonlinear estimation of the relative pose *56*
6.1.5 Parameter refinement *57*
6.2 EXPERIMENTS AND RESULTS 58
6.2.1 Simulations *59*
6.2.2 Real-world experiments *65*
6.3 CONCLUSION AND CONTRIBUTION 71
CHAPTER 7 CALIBRATING NON-OVERLAPPING RGB-D CAMERAS WITH ONE PATTERN 73
7.1 INTRODUCTION 73
7.2 RELATED WORKS 75
7.3 INTRODUCTION TO KINECT 77
7.3.1 IR image *77*
7.3.2 RGB image *78*

7.3.3 Depth image *78*

7.3.4 Depth resolution *78*

7.3.5 Shift between IR image and depth image *79*

7.4 GEOMETRICAL MODEL OF KINECT 80

7.5 CALIBRATING ONE KINECT 81

7.6 3D POINT RECONSTRUCTION FROM ITS 2D RGB IMAGE PROJECTION 85

7.7 CALIBRATING TWO KINECTS WITH NON-OVERLAPPING VIEWS 87

7.7.1 Problem definition *87*

7.7.2 Process *89*

7.7.3 Linear estimation of relative pose *90*

7.7.4 Nonlinear estimation of relative pose *91*

7.7.5 Parameter refinement *92*

7.8 EXPERIMENTS AND RESULTS 93

7.8.1 Simulations *93*

7.8.2 Real-world experiments *98*

7.9 CONCLUSION AND CONTRIBUTION 100

CHAPTER 8 CONCLUSIONS AND CONTRIBUTIONS 103

8.1 CONCLUSION 103

8.2 PUBLICATIONS 106

REFERENCES 108

LIST OF FIGURES

Figure 1.1 :A driving assistance system of detecting a driver's attention. By mapping the head/gaze direction to the coordinate system of the outward camera, we can estimate what area of scene the driver pays his/her attention to. 2

Figure 1.2: Calibration of in-vehicle non-overlapping cameras with a laser pointer. (a) The laser pointer is mounted on a calibration board. (b) Calibrating non-overlapping cameras with a laser pointer and one pattern. (c) Calibrating non-overlapping cameras with a laser pointer and two patterns. 3

Figure 2.1: (a) Using a mirror to calibrate cameras with a big baseline facing in almost the opposite direction, the image of the virtual pattern captured by the camera is rather small due to the distance between it and Camera 2. (b) Example of the pattern's image captured by the outside camera via a mirror. 9

Figure 3.1: The ideal pinhole camera model describes the relationship between a 3D point $(X,Y,Z)^T$ and its corresponding 2D projection (u,v) onto the image plane. 12

Figure 3.2: (a) The image (x,y) and camera (u,v) coordinate system. (b) Non-ideal image sensor with non-square, skewed pixels. 14

Figure 3.3: (a) Distorted image. (b) Corresponding corrected image using the inverted mapping method. 18

Figure 3.4: The relationship between the camera and world coordinate system is defined by the translation t and rotation R of the camera. 20

Figure 3.5: Epipolar geometry. The epipolar plane is defined by the point P and the two camera centers O_l and O_r. 23

Figure 4.1: Calibrating the pose of the laser pointer. (a) Diagram of the calibration. (b) A laser spot appears in the image when the laser pointer points to the optical center of the camera. 26

Figure 4.2: Calibrating the pose of the laser pointer. (a) Diagram of the calibration. (b) A laser spot appears in the image when the laser pointer points to the optical center of the camera. 28

Figure 4.3: Estimating the upper error limit of the laser pointer's orientation. 30

Figure 5.1: Calibrating non-overlapping cameras with a laser pointer. 39

Figure 5.2: Calibrating non-overlapping cameras with a laser pointer. 39

Figure 5.3: Results with synthetic data of different number of laser spots: extrinsic calibration error and reprojection epipolar error with respect to noise level. Note that,

quantizing error of digital image has been considered before adding noise. And with real data, the noise standard deviation is below 0.1 pixel. 46
Figure 5.4: Example images. Left is the pattern captured by Camera 1 inside the vehicle. Right is the laser spot captured by Camera 2 outside the vehicle. 48

Figure 6.1: Calibrating non-overlapping cameras with a laser pointer and two patterns. 53
Figure 6.2: Block diagram of the proposed calibration algorithm based on the collinear constraint method. 54
Figure 6.3: Evaluation of the accuracy for varying data volumes, noise levels and baselines for the laser and mirror based methods. 62
Figure 6.4: Example images. Left is the pattern captured by Camera 1 inside the vehicle. Right is the laser spot captured by Camera 2 outside the vehicle. 66
Figure 6.5: Setup used for the comparison experiment. 68
Figure 6.6: Experiments in five different lighting conditions: environment (left), pattern inside vehicle (middle), pattern outside vehicle (right). 71

Figure 7.1: Systems with two non-overlapping Kinects 74
Figure 7.2: Kinect consists of Infrared (IR) projector, IR camera and RGB camera (illustration from [52]). 77
Figure 7.3: Geometrical model of Kinect. 79

Figure 7.4: RGB and IR images of a calibration pattern 82

Figure 7.5: A sketch for calibrating non-overlapping Kinects with laser pointer 87

Figure 7.6: A sketch for calibrating non-overlapping Kinects with laser pointer 89

Figure 7.7: Evaluate accuracy against data volume and noise level for Mirror-Method, Laser-Method and Laser-Depth-Method. 97

Figure 7.8: Compare two calibration methods..... 101

LIST OF TABLES

Table 5.1: Experiment results corresponding to different data volume (48, 96 and 192 laser spots). 50

Table 6.1: Results for in-vehicle experiment 67

Table 6.2: Results of comparison experiment 69

Table 7.1: Intrinsic parameters for Kinect1 83

Table 7.2: Intrinsic parameters for Kinect2 84

Table 7.3: Result of real-world experiments 100

DEDICATION

I would like to dedicate this dissertation to my wife, Yun Huang. I want to thank her for being so patient with me and supportive through these years of my studying. Her love and wisdom enabled me to face up to any difficulties and make me the person I am today. Also this thesis is dedicated to my parents, Caiping Jin and Guocai Zou for all the support and encouragement throughout these years of study abroad. I never could have done this without you. I'd also like to thank respectable teachers and all my dear friends, instructors as well. They have helped me a lot on my way to pursue my dreams. All great wishes to you.

ABSTRACT

This thesis focuses on the topics of calibrating non-overlapping cameras. Multi-camera system has been extensively applied in action recognition, Simultaneous Localization and Mapping, robot navigation, scene understanding and so on. It is necessary to calibrate relative poses between internally calibrated cameras. However, sometimes the cameras of the system may have non-overlapping view. Since the FOVs (field of view) of the two cameras do not overlap, the conventional stereo camera calibration methods cannot be applied directly. This thesis deals with such a problem. The contributions of this thesis are two-fold: calibration of non-overlapping conventional RGB cameras and calibration of non-overlapping RGB-D cameras, e.g. Kinects.

For part 1 (calibration of non-overlapping conventional RGB cameras), this thesis aims at calibrating two in-vehicle cameras mounted back to back. Road sign detection is carried out by an outward-facing camera, and the monitoring of a driver's behavior is carried out by an inward-facing camera. To assess a driver's attention level, we need not only measure the gaze direction of the driver using the inward camera, but also map the estimated gaze direction to the coordinate system of the outward camera to determine what the driver is looking at. However, the mapping of gaze direction cannot be done without knowing the relative pose between the inward and outward-facing cameras. Since the FOVs of the two cameras do not overlap, the conventional stereo camera calibration

methods cannot be applied directly. Since one camera can measure the vehicle's motion while the other cannot, the consistency of motion cannot be used for camera calibration in this case. Since one of the cameras is situated inside the vehicle, it is difficult to set up a mirror with which the two cameras can observe the same scenes. This paper proposes a new approach to calibrate such non-overlapping cameras using a laser pointer, which can overcome the above problems. A laser pointer is mounted on a calibration board so that its pose within the coordinate system of the calibration board can be obtained. While one of the cameras, such as the inward camera, observes the calibration board with the laser pointer, the intersection of the laser ray and the outside scenery is observed by the other camera. Thus, the FOV of both cameras are connected and the relative pose between the cameras can be estimated. Two algorithms are presented based on this concept. In the first algorithm, the intersection of the laser ray and the object in the driver's FOV is observed by the second camera. In this case, the relative pose between the cameras can be estimated through a coplanar constraint. In the second algorithm, the second camera observes another calibration board set up outside the vehicle, with which the laser ray intersects. Thus, a more rigid constraint, namely, the collinear constraint, can be used to estimate the relative pose between the cameras. We compare the performance of the two proposed algorithms with the conventional mirror-based method through simulations and experiments. Finally the laser-based collinear method is applied to the calibration of an in-vehicle camera system. In contrast to other methods, the proposed method is simple,

practical, and especially well suited to the calibration of non-overlapping in-vehicle cameras in a factory or garage.

For part 2 (calibration of non-overlapping conventional RGB-D cameras), this thesis aims at calibrating two Kinects mounted on a mobile robot. The two Kinects monitor front and back respectively, so that the robot can go forward or backward anytime. To reconstruct the environment in a unified coordinate system, the relative pose of two Kinects should be calibrated. However, in order to avoid interference between infrared rays of two Kinects, two Kinects should never have overlapping views. This thesis deals with the above problem. Until now, several methods of calibrating non-overlapping RGB cameras have been reported. However, if the methods for RGB cameras are used directly to RGB-D cameras, the depth cues of RGB-D cameras is ignored without being exploited for the pose calibration of RGB-D cameras. In this thesis, we cope with the problem of calibrating non-overlapping RGB-D cameras, such as Kinects, by exploiting the depth cues. A laser pointer is fixed at one calibration board so that its pose at the coordinate system of the calibration board can be obtained easily. While one of the RGB-D cameras observes the calibration board fixed with the laser pointer, the laser pointer project a spot to the scene which is observed by the other. Thus, two 3D points, respectively located in the field of views of the two RGB-D cameras, are connected by a laser ray. The relative pose of two RGB-D cameras can be estimated through this collinear constraint. The effectiveness of the proposed method has been proved by results of simulation and real-world experiments.

ACKNOWLEDGEMENTS

Joining the robot vision group at Tottori University was one of the best decisions of my life. I have to thank Prof. Li for the continuous support and for the many things he has taught me. He had a project ready for me as soon as I arrived at Tottori University, enabling me to jump into research immediately. His openness to meet at any time of day or night was crucial to my learning process and coming up with new ideas. His insights made debugging and implementing complicated algorithms relatively easy.

Additionally, this work would not have been possible without the help of Ning Dai and Lei Zhang who supervised me in my master learning at the Shanghai Institute of Technical Physics of the Chinese Academy of Science. They taught me the foundations of research which I further built upon throughout my PhD. I had the opportunity to work with Xuebin Qin directly for the first two years of my PhD as well. His patience with my persistent questions was essential to my progress in the early years of the program. I also had the pleasure of working with Hanchao Jia. His intuition about what works in computer vision inspired me in the later portion of my PhD to think outside the box and produce some state-of-the-art results.

I'd especially like to thank Zheng Jiao and Jianfeng Li for doing me a lot of favor with practical experiments. Especially thank Prof. Nakanishi for the engaging chats and group meetings throughout my years at Tottori University. His advises are extremely useful to my research. His rigorous work attitude and enthusiasm to students moved me a

lot. I would also like to thank all the students and teachers at our lab for the interesting discussions and research they have all done.

I thank my family for their continuing support in my studies and helping me with all the tough decisions along the way. Finally, I thank my wife Yun Huang whose patience with my long hours of work and her loving support helped me get through the thick and thin of my PhD. Despite the exciting research I had the opportunity to do, it was her comedic upbeat attitude and encouragement that kept me going. Without her this would not have been possible.

Wuhe Zou

2015.3.18, Tottori, Japan

CHAPTER 1

Introduction

1.1 Background

Cameras are widely used in driving assistance [1, 2, 3, 4]. For example, road scene analysis, such as lane marking detection and road sign detection, is carried out by an outward-facing camera, and the monitoring of a driver's behavior is carried out by an inward-facing camera [2]. To assess a driver's attention level, we need not only measure the gaze direction of the driver using the inward camera, but also map the estimated gaze direction to the coordinate system of the outward camera to determine what the driver is looking at. However, the mapping of gaze direction cannot be done without knowing the relative pose between the inward and outward-facing cameras [2, 3].

As shown in Fig. 1.1, there is no overlapping field of view (FOV) between the inward camera observing the driver's behavior, Camera 1, and the outward camera, Camera 2. Moreover, the inward camera is mounted inside the vehicle, meaning that the outward camera records a moving image, while the inward camera observes a static scene when the car is moving. The calibration of the extrinsic parameters (relative pose) of the outward and inward cameras is the problem addressed in this research.

Since the FOVs of the two cameras do not overlap, the conventional stereo camera calibration methods cannot be applied directly [1, 5, 6, 7]. Since one camera can measure the vehicle's motion while the other cannot, the consistency of motion

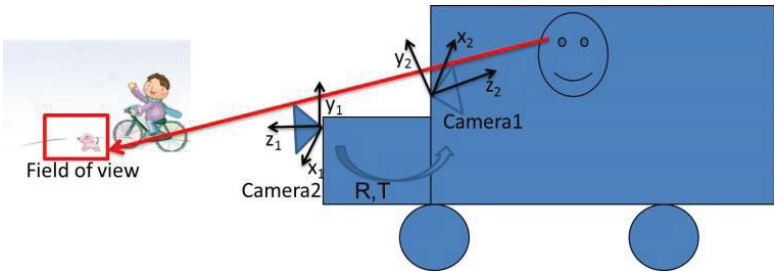


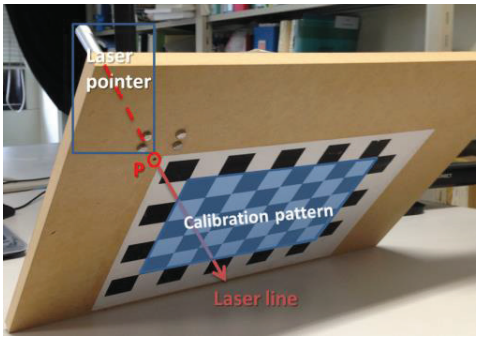
Figure 1.1 :A driving assistance system of detecting a driver's attention. By mapping the head/gaze direction to the coordinate system of the outward camera, we can estimate what area of scene the driver pays his/her attention to.

mentioned discussed in [9, 13] cannot be used for camera calibration in this case. Since one of the cameras is situated inside the vehicle, it is difficult to set up a mirror with which the two cameras can observe the same scenes [11, 14, 15, 16, 20, 21].

1.2 Contribution

This research proposes new approaches to calibrating such non-overlapping cameras using a laser pointer, which can overcome the above problems. A laser pointer is mounted on a calibration board so that its pose within the coordinate system of the calibration board can be obtained as shown in Fig. 1.2(a). While one of the cameras, such as the inward camera, observes the calibration board with the laser pointer, the intersection of the laser ray and the outside scenery is observed by the other camera, as shown in Figs. 1.2(b) and (c). Thus, the FOV of both cameras are connected and the relative pose between the cameras can be estimated. The contribution includes calibration of non-overlapping RGB cameras and calibration of non-overlapping RGB-D cameras.

For calibrating non-overlapping RGB cameras, two algorithms are presented based on this concept. In the first algorithm, the intersection of the laser ray and the object in the driver's FOV is observed by Camera 2, as shown in Fig. 1.2(b). In this case, the relative pose between the cameras can be estimated through a coplanar constraint. In



(a)

(b) (c)

Figure 1.2: Calibration of in-vehicle non-overlapping cameras with a laser pointer. (a) The laser pointer is mounted on a calibration board. (b) Calibrating non-overlapping cameras with a laser pointer and one pattern. (c) Calibrating non-overlapping cameras with a laser pointer and two patterns.

the second algorithm, Camera 2 observes another calibration board set up outside the vehicle, with which the laser ray intersects, as shown in Fig. 1.2(c). Thus, a more rigid constraint, namely, the collinear constraint, can be used to estimate the relative pose between the cameras. We compare the performance of these two algorithms with the conventional mirror-based method presented in [15] through simulations and experiments.

On the other hand, we cope with the problem of calibrating non-overlapping RGB-D cameras, such as Kinects, by exploiting the depth cues. A laser pointer is fixed at one calibration board so that its pose at the coordinate system of the calibration board can be obtained easily, as shown in Fig. 1.2. While one of the RGB-D cameras observes the calibration board fixed with the laser pointer, the laser pointer project a spot to the scene which is observed by the other. Thus, two 3D points, respectively located in the field of views of the two RGB-D cameras, are connected by a laser ray. The relative pose of two RGB-D cameras can be estimated through this collinear constraint.

1.3 Organization

The rest of this thesis is organized as follows. Chapter 2 reviews previous works on external calibration of cameras, giving particular attention to calibration methods for non-overlapping cameras. Particularly, the state-of-the-art method for calibrating non-overlapping cameras is introduced, which use a mirror to calibrate non-overlapping cameras. Chapter 3 introduces basic knowledge of camera model, including intrinsic parameters, extrinsic parameters, projection matrix and stereo geometry. The background knowledge makes our methods easier to understand. Chapter 4 introduces how to

estimate the pose of the laser pointer relative to the calibration pattern. An error analysis has been carried out to show the effectiveness of this technic. The pose of the laser pointer will be used as initial value throughout this dissertation. Chapter 5 introduces how to calibrate non-overlapping RGB cameras with one calibration pattern using a coplanar constraint method. Chapter 6 introduces how to calibrate non-overlapping RGB cameras with tow calibration patterns using a collinear constraint method. Chapter 7 introduces how to calibrate non-overlapping RGB-D cameras with one calibration pattern. Finally, conclusions are presented in Chapter 8.

CHAPTER 2

Relative Research

Camera calibration is a fundamental problem in the field of computer vision. The relative positions and orientations of cameras have to be considered in many computer/robot vision applications such as 3D reconstruction, motion estimation, augmented reality, visual surveillance, and intelligent transportation systems. In these applications, it is necessary to calibrate cameras in a single coordinate frame.

2.1 Calibrating cameras with overlapping view

Various methods have been proposed to calibrate cameras using calibration objects. Generally there are two classes of methods: the first class uses a calibration object and provides the external camera parameters within the coordinate system of the calibration object as a byproduct of the internal calibration; the second class does not use any calibration object and delivers an external camera calibration up to scale.

We first review the methods that can be used in conjunction with our method. A variety of methods use fixed 3D geometry [30, 39]. Orthonormal properties of the rotation matrix have been used in plane-based approaches [32, 31, 37, 7, and 38]. A planar pattern viewed from at least three different orientations is used in [7]. Other calibration objects include: spheres [26, 40], circles [33], surfaces of revolution [29],

shadows [28] and even fixed stars in the night sky [35]. Most of these methods are based on the constraints provided by vanishing points along perpendicular directions.

The method in [12] uses a moving plate to calibrate multi-camera environments and does not require a 3D calibration object with known 3D coordinates. Kitahara et al. [34] calibrated their large scale multi-camera environment by using a classical direct method [39]. The necessary 3D points are collected by a combined use of a calibration board and a 3D laser-surveying instrument. Svoboda et al. [24] have calibrated a system of at least three cameras by obtaining a set of virtual 3D points made by waving a bright spot throughout the working volume. Baker and Aloimonos [27] proposed a calibration method for a multi-camera network which requires a planar pattern with a precise gird. Lee et al. [36] established a common coordinate frame for sparse set of cameras so that all cameras observe a common dominant plane. They tracked objects moving in this plane and from their trajectories they estimated the external parameters of the cameras in a single coordinate system. Sinha et al. [10] have calibrated a camera network by using epipolar geometry constraints derived from dynamic silhouettes. All above conventional methods for camera network calibration assume that the cameras have overlapping views such that points lying in these views can be used to determine the relative positions of the cameras. Here, we focus on the calibration of cameras with non-overlapping views.

2.2 Calibrating cameras with non-overlapping view

An approach to aligning non-overlapping image sequences was proposed by Caspi and Irani [8] for the case of multi-camera systems sharing the same projection

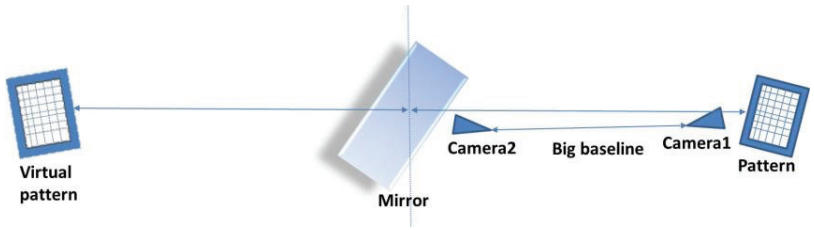
center. In this approach, a calibration object is not used. However, the requirements of a common optical center and rigid motion are strict limitations. Thus, this approach is inapplicable to camera calibration with a large baseline separation.

Esquivel [9] proposed an approach to calibrate a pair of cameras with non-overlapping views based on the motion consistency of rigid coupling. This approach used image sequences synchronously captured by internally calibrated cameras with non-overlapping views to calibrate the cameras' extrinsic parameters. The approach computes the camera motion up-to-scale independently for each camera. Then, all reconstructions are aligned with a similarity transformation. However, this approach is not suitable for the calibration of cameras which do not satisfy motion consistency, such as the two vehicle-mounted cameras shown in Fig. 1.1.

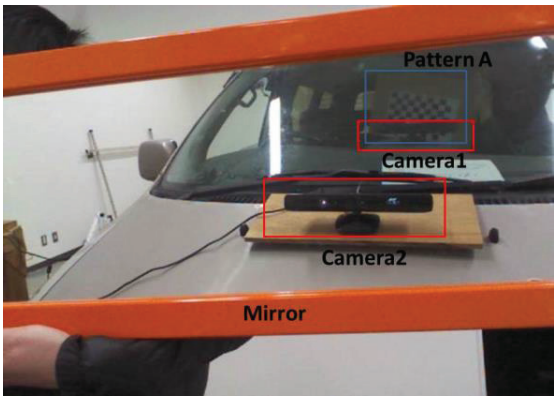
2.3 State-of-the-art method for calibrating non-overlapping cameras

State-of-the-art methods of calibrating non-overlapping cameras using mirrors have been proposed by Kumar [11], Hesch [14] [21], Rodrigues [15], Agrawal [16], and Sturm [20]. The extrinsic parameters of the camera network are estimated from the mirrored camera poses by formulating constraints between the cameras. The calibration object provides a unique reference frame for computing a family of mirrored camera poses. However, it may be difficult to apply this approach to the estimation of the relative pose between the outward and inward-facing cameras in our setup, due to the following considerations.

- Because Camera 1 is inside the vehicle, the setup space is limited. A simpler and more practical method is desirable.
- When the baseline between the two cameras is large, the virtual pattern inside the mirror will be far from Camera 2, as shown in Fig. 2.1(a). A sample image using this method for our setup is shown in Fig.2.1 (b). It is apparent that the image of the mirrored pattern captured by Camera 2 becomes small. This influences the accuracy of detecting the corners of the checkered



(a)



(b)

Figure 2.1: (a) Using a mirror to calibrate cameras with a big baseline facing in almost the opposite direction, the image of the virtual pattern captured by the Camera2 is rather small due to the distance between it and Camera2. (b) Example of the pattern's image captured by the outside camera via a mirror.

calibration pattern. In an extreme case, if we want to calibrate cameras set up in a stadium, it will be nearly impossible to apply the mirror-based approach.

- Because the cameras are pointing in opposite directions to each other, when Camera 1 observes the calibration board, the calibration board image inside the mirror observed by Camera 2 can be occluded by Camera 1, as shown in Fig. 2.1(b).
- The windscreen may blur the image; the light reflected by the windscreen may also obstruct the observation of the calibration pattern, as shown in Fig. 2.1(b).

CHAPTER 3

Fundamentals

As the subject of this thesis is calibrating non-overlapping cameras, we start with physical model of single camera and stereovision in this chapter. Some basic mathematic fundamentals for camera calibration will also be mentioned, which will make this thesis easy to understand. In this chapter, we firstly introduce pinhole camera model. Then the intrinsic and extrinsic parameters will be introduced. Finally stereovision system and its epipolar problem will be introduced.

3.1 Pinhole camera model

In this section, we describe the image acquisition process known as the pinhole camera model, which is regularly employed as a basis in this thesis. More specifically, we first discuss the model that integrates the internal or intrinsic camera parameters, such as the focal length and the lens distortion. Secondly, we extend the presented simple camera model to integrate external or extrinsic camera parameters corresponding to the position and orientation of the camera.

3.2 Intrinsic parameters

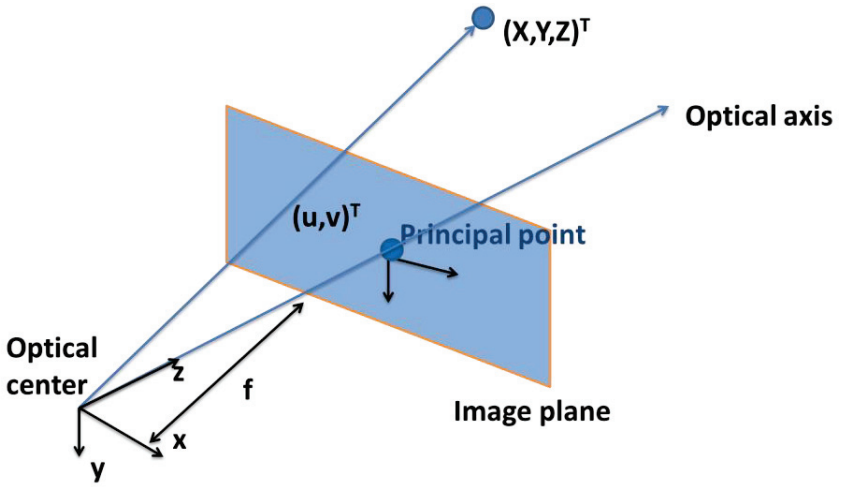


Figure 3.1: The ideal pinhole camera model describes the relationship between a 3D point $(X,Y,Z)^T$ and its corresponding 2D projection (u,v) onto the image plane.

The pinhole camera model defines the geometric relationship between a 3D point and its 2D corresponding projection onto the image plane. When using a pinhole camera model, this geometric mapping from 3D to 2D is called a perspective projection. We denote the center of the perspective projection (the point in which all the rays intersect) as the optical center or camera center and the line perpendicular to the image plane passing through the optical center as the optical axis (see Fig. 3.1). Additionally, the intersection point of the image plane with the optical axis is called the principal point. The pinhole camera that models a perspective projection of 3D points onto the image plane can be described as follows.

3.2.1 Perspective projection using homogeneous coordinates

Let us consider a camera with the optical axis being collinear to the Zcam-axis and the optical center being located at the origin of a 3D coordinate system (see Fig. 3.1). The projection of a 3D world point $(X,Y,Z)^T$ onto the image plane at pixel position $(u,v)^T$ can be written as:

$$\begin{cases} u = \dfrac{X \cdot f}{Z} \\ v = \dfrac{Y \cdot f}{Z} \end{cases} \qquad (3.1)$$

Where, f denotes the focal length. To avoid such a non-linear division operation, the previous relation can be reformulated using the projective geometry framework, as

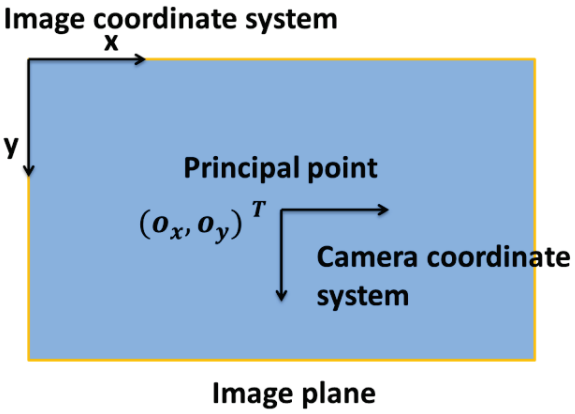
$$(\lambda u, \lambda v, \lambda)^T = (Xf, Yf, Z)^T . \qquad (3.2)$$

This relation can be expressed in matrix notation by

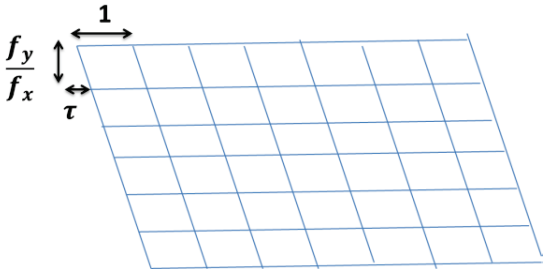
$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \qquad (3.3)$$

Where, $\lambda = Z$ is the homogeneous scaling factor.

3.2.2 Principal-point offset



(a)



(b)

Figure 3.2: (a) The image (x,y) and camera (u,v) coordinate system. (b) Non-ideal image sensor with non-square, skewed pixels.

Most of the current imaging systems define the origin of the pixel coordinate system at the top-left pixel of the image. However, it was previously assumed that the origin of the pixel coordinate system corresponds to the principal point $(o_x,o_y)^T$, located at the center of the image (see Fig. 3.2(a)). A conversion of the coordinate systems is thus necessary. Using homogeneous coordinates, the principal-point position can be readily integrated into the projection matrix. The perspective projection equation becomes now

$$\lambda\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & o_x & 0 \\ 0 & f & o_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \qquad (3.4)$$

3.2.3 Image-sensor characteristics

To derive the relation described by Equation (3.4), it was implicitly assumed that the pixels of the image sensor are square, i.e., aspect ratio is 1:1 and pixels are not skewed. However, both assumptions may not always be valid. In practice, the pixel aspect ratio is often provided by the image-sensor manufacturer. Second, pixels can potentially be skewed, especially in the case that the image is acquired by a frame grabber. In this particular case, the pixel grid may be skewed due to an inaccurate synchronization of the pixel-sampling process. Both previously mentioned imperfections of the imaging system can be taken into account in the camera model, using the parameters f_y/f_x and τ, which model the pixel aspect ratio and skew of the pixels, respectively (see Fig. 3.2(b)). The projection mapping can be now updated as

$$\lambda\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & \tau & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = [\mathbf{M} \quad \mathbf{0}]P \qquad (3.5)$$

With $P = (X, Y, Z, 1)^T$ being a 3D point defined with homogeneous coordinates. In practice, when employing recent digital cameras, it can be safely assumed that pixels are square (f_x=f_y) and non-skewed ($\tau = 0$). The projection matrix that incorporates the

intrinsic parameters is denoted as **M** throughout this thesis. The all zero element vector is denoted by **0** .

3.2.4 Radial lens distortion

Real camera lenses typically suffer from non-linear lens distortion. In practice, radial lens distortion causes straight lines to be mapped as curved lines. The radial lens distortion appears more visible at the image edges, where the radial distance is high. A standard technique to model the radial lens can be described as follows.

Let $(x_u,y_u)^T$ and $(x_d,y_d)^T$ be the corrected and the measured distorted pixel positions, respectively. The relation between an undistorted and distorted pixel can be modeled with a polynomial function and can be written as

$$\begin{pmatrix} x_u - o_x \\ y_u - o_y \end{pmatrix} = L(r_d) \begin{pmatrix} x_d - o_x \\ y_d - o_y \end{pmatrix}, \qquad (3.6)$$

Where,

$$L(r_d) = 1 + k_1 r_d^{\,2} \text{ and } r_d^{\,2} = (x_d - o_x)^2 + (y_d - o_y)^2 . \qquad (3.7)$$

In the case $k_1=0$, it can be noted that $x_u=x_d$ and $y_u=y_d$, which corresponds to the absence of radial lens distortion.

It should be noted that Equation (3.6) provides the correct pixel position using a function of the distorted pixel position. However, to generate an undistorted image, it would be more convenient to base the function L(r) on the undistorted pixel position. This technique is usually known as the inverse mapping method. The inverse mapping technique consists of scanning each pixel in the output image and re-sampling and

interpolating the correct pixel from the input image. To perform an inverse mapping, the inversion of the radial lens distortion model is necessary and can be described as follows. First, similar to the second part of Equation (3.7), we define

$$r_u^2 = (x_u - o_x)^2 + (y_u - o_y)^2 \qquad (3.8)$$

Then, taking the norm of Equation (3.6), it can be derived that

$$(x_u - o_x)^2 + (y_u - o_y)^2 = L(r_d) \cdot ((x_d - o_x)^2 + (y_d - o_y)^2), \qquad (3.9)$$

Which is equivalent to

$$r_u = L(r_d) \cdot r_d . \qquad (3.10)$$

When taking into account Equation (3.7), this equation can be rewritten as a cubic polynomial:

$$r_d^3 + \frac{1}{k_1} r_d - \frac{r_u}{k_1} = 0 . \qquad (3.11)$$

The inverted lens distortion function can be derived by substituting Equation (3.10) into Equation (3.6) and developing it from the right-hand side:

$$\begin{pmatrix} x_d - o_x \\ y_d - o_y \end{pmatrix} = \frac{r_d}{r_u} \begin{pmatrix} x_u - o_x \\ y_u - o_y \end{pmatrix}, \qquad (3.12)$$

Where r_d can be calculated by solving the cubic polynomial function of Equation (3.11). This polynomial can be solved using Cardano's method, by first calculating the discriminant $\Delta = q^2 + 4 / 27\ p^3$ where $p = 1 / k_1$ and $q = - r_u / k_1$. Depending on the sign of the discriminant, three sets of solutions are possible.

If $\Delta > 0$, then the equation has one real root r_{d1} defined as

$$r_{d1} = \sqrt[3]{\frac{-q+\sqrt{\Delta}}{2}} + \sqrt[3]{\frac{-q-\sqrt{\Delta}}{2}}\,. \qquad (3.13)$$

If $\Delta>0$, then the equation has three real roots r_{dk} defined as

$$r_{dk} = 2\sqrt{\frac{-p}{3}}\cos\left(\frac{\arccos\left(\frac{-q}{2}\sqrt{\frac{27}{-p^3}}\right)+2k\pi}{3}\right), \qquad (3.14)$$

For k={0, 1, 2}, where an appropriate solution r_{dk} should be selected such that $r_{dk} > 0$ and $r_{dk} < r_{uk}$. However, only one single radius corresponds to the practical solution. Therefore, the second case $\Delta < 0$ should not be encountered. The third case with $\Delta = 0$ is also impractical. In practice, we have noticed that, indeed, these second and third cases never occur.

(a) (b)

Figure 3.3: (a) Distorted image. (b) Corresponding corrected image using the inverted mapping method.

As an example, Fig. 3.3 depicts a distorted image and the corresponding corrected image using the inverted mapping method, with $\Delta > 0$.

The discussed lens-distortion correction method requires knowledge of the lens parameters, i.e., k_1 and $(o_x,o_y)^T$. The estimation of the distortion parameters can be performed by minimizing a cost function that measures the curvature of lines in the distorted image. To measure this curvature, a practical solution is to detect feature points belonging to the same line on a calibration rig, e.g., a checkerboard calibration pattern. Each point belonging to the same line in the distorted image forms a bended line instead of a straight line. By comparing the deviation of the bended line from the theoretical straight line model, the distortion parameters can be calculated.

3.3 Extrinsic parameters

As opposed to the intrinsic parameters that describe internal parameters of the camera (focal distance, radial lens parameters), the extrinsic parameters indicate the external position and orientation of the camera in the 3D world. Mathematically, the position and orientation of the camera is defined by a 3×1 vector **t** and by a 3×3 rotation matrix **R** (see Figure 3.4).

3.3.1 Direct mapping (from world to camera coordinate systems)

To obtain the pixel position $p = (x,y,1)^T$ of a 3D-world homogeneous point P, the camera should be first translated to the world coordinate origin and second, rotated. This can be mathematically written as

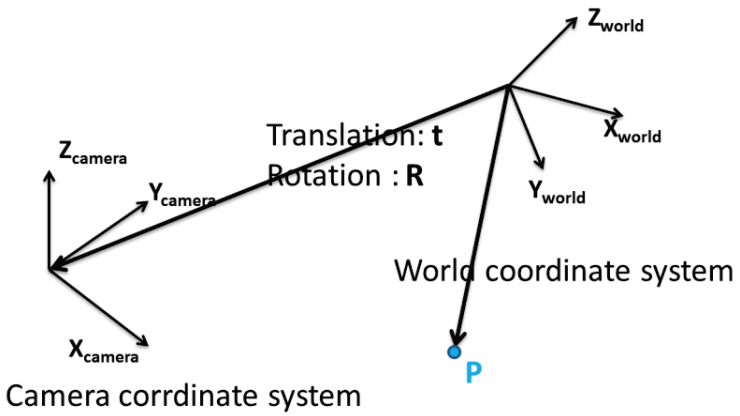


Figure 3.4: The relationship between the camera and world coordinate system is defined by the translation t and rotation R of the camera.

$$\lambda \mathbf{p} = \left[\mathbf{M}|\mathbf{0}\right] \begin{bmatrix} \mathbf{R} & 0 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{0}^T & -\mathbf{t} \\ 0 & 1 \end{bmatrix} P \,. \qquad (3.15)$$

Alternatively, when combining matrices, Equation (3.15) can be reformulated as

$$\lambda \mathbf{p} = \left[\mathbf{M}|\mathbf{0}\right] \begin{bmatrix} \mathbf{R} & -\mathbf{Rt} \\ \mathbf{0} & 1 \end{bmatrix} P = \mathbf{MR} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - \mathbf{MRt} \,. \qquad (3.16)$$

3.3.2 Indirect mapping (from camera to world coordinate systems)

Previously, the process of projecting a 3D point onto the 2D image plane was described. We now present how a 2D point can be back-projected to the 3D space and derive the corresponding coordinates. Considering a 2D point p in an image, there exists a collection of 3D points that are mapped and projected onto the same point p. This collection of 3D points constitutes a ray connecting the camera center $\mathbf{t} = (tx,ty,tz)^T$ and p

= $(x,y,1)^T$. From Equation (3.16), the ray $\mathbf{P}(\lambda)$ associated to a pixel $\mathbf{p} = (x,y,1)^T$ can be defined as

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \mathbf{t} + \lambda \mathbf{R}^{-1}\mathbf{M}^{-1}\mathbf{p}\,, \qquad (3.17)$$

Where λ is the positive scaling factor defining the position of the 3D point on the ray. In the case Z is known, it is possible to obtain the coordinates X and Y by calculating λ using the relation

$$\lambda = \frac{Z - t_z}{z_3}, where, (z_1, z_2, z_3)^T = \mathbf{R}^{-1}\mathbf{M}^{-1}\mathbf{p}\,. \qquad (3.18)$$

The back-projection operation is important for depth estimation and image rendering, which will be extensively addressed later in this thesis.

3.4 Stereovision system

In the previous section, we have described the geometry of a single camera. We now examine the case of two camera views capturing the same scene from two different viewpoints. Given two images, we are interested in estimating the 3D structure of the scene. The estimation of coordinates of a 3D point P can be performed in two steps. First, given a selected pixel p_l in the image I_l, the position of the corresponding pixel p_r in image I_r is estimated. Similarly to the previous section, a pair of corresponding points p_l and p_r is called a point-correspondence. This correspondence in points p_l and p_r comes from the projection of the same point P onto both images I_l and I_r. Second, the position of P is calculated by triangulation of the two corresponding points, using the geometry of

the two cameras. The geometry of the two cameras relates to the respective position and orientation and internal geometry of each individual camera. The underlying geometry that describes the relationship between both cameras is known as the epipolar geometry. The estimation process of the epipolar geometry is known as weak calibration, as opposed to the strong calibration. The epipolar geometry addresses (among others) the following two aspects.

- Geometry of point-correspondence: considering a point in an image, the epipolar geometry provides a constraint on the position of the corresponding point.
- Scene geometry: given point-correspondences and the epipolar geometry of both cameras, a description of the scene structure can be recovered.

3.5 The stereo correspondence e problem

Let us now describe the geometry of two images and define their mutual relation. Consider a 3D point P that is projected through the camera centers O_l and O_r onto two images at pixel positions p_l and p_r, respectively (see Fig. 3.5).

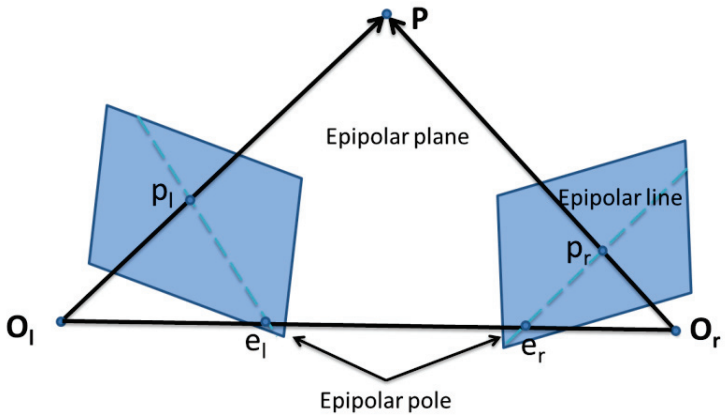


Figure 3.5: Epipolar geometry. The epipolar plane is defined by the point P and the two camera centers O_l and O_r.

Obviously, the 3D points P, O_l, O_r and the projected points p_l, p_r are all located within one common plane. This common plane is known as the epipolar plane. The epipolar plane is fully determined by the back-projected ray going through O_l and p_l and the camera center O_r. The property that the previously specified points belong to the epipolar plane provides a constraint for searching point-correspondences. More specifically, considering the image point p_l, a point p_r lies on the intersection of the epipolar plane with the second image plane. The intersection of both planes corresponds to a line known as the epipolar line. Therefore, the search of point-correspondences can be limited to a search along the epipolar line instead of an exhaustive search in the image. Additionally, it is interesting to note that this constraint is independent of the scene structure, but instead, uniquely relies on the epipolar geometry. The epipolar geometry can be described using a 3×3 rank-2 matrix, called the fundamental matrix **F**.

We now introduce some terminology related to the epipolar geometry employed further in this thesis.

- The epipolar plane is the plane defined by a 3D point and the two cameras centers.
- The epipolar line is the line determined by the intersection of the image plane with the epipolar plane.
- The baseline is the line going through the two cameras centers.
- The epipolar pole is the image-point determined by the intersection of the image plane with the baseline. Also, the epipolar pole corresponds to the projection of the first camera center (O_l) onto the second image plane, or vice versa.

CHAPTER 4

Estimating the Pose of the Laser Pointer Relative to the Calibration Pattern

A laser pointer mounted on a checkerboard (Board A) is used in both the coplanar and collinear calibration methods (introduced in chapter 5, chapter 6 and chapter 7 respectively). Therefore, in this chapter, I preliminarily estimate the pose of the laser ray. The parameters of the laser ray consist of its orientation and the location of its starting point in the coordinate system of Board A.

4.1 Notation

In this thesis, we denote 2D points by minuscule, e.g., q ,3D points by majuscule, e.g., Q, vectors by bold minuscule, e.g., $\mathbf{n}$, matrices by bold majuscule, e.g., $\mathbf{R}$, and 3D lines of Plücker coordinates by ℓ and $\boldsymbol{\delta}$. Unless otherwise stated, homogeneous coordinates are used for points and other geometric entities. Equality between vectors and matrices, up to a scalar factor, is denoted by $\cong$. The cross-product of two vectors $\mathbf{a}$ and $\mathbf{b}$ is written as $\mathbf{a}\times\mathbf{b}$. A coordinate system is denoted by a symbol, e.g., C_1. Superscripts are used to identify the reference frame that the points or vectors belong to. For example, Q^1 denotes the 3D point Q in coordinate system C_1. A transformation between two coordinate systems is denoted such that Q^1 can be represented by Q^2 via the transformation $\mathbf{T}_{2\_1}$: $Q^1 = \mathbf{T}_{2\_1}Q^2$, where

$$\mathbf{T}_{2\_1} = \begin{bmatrix} \mathbf{R}_{2\_1} & \mathbf{t}_{2\_1} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

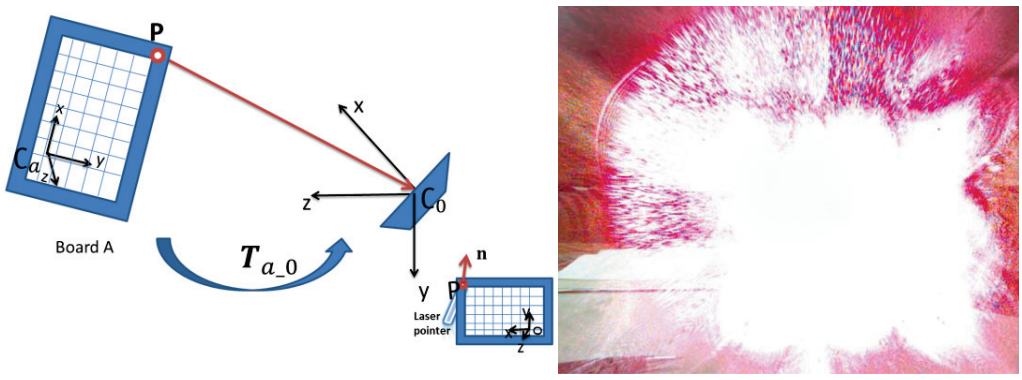
To project a 3D point Q to its image projection q, a perspective projection model is used.

4.2 Problem definition

The front of Board A has a checkered pattern used for calibration. The laser pointer is mounted almost vertically on the back of the board such that its ray passes through a small hole in the board.

The main idea is that, because of the ideal central projecting model and the unidirectional property of the laser, when the laser is viewed by the camera, the laser should be pointing to the optical central of the camera.

A sketch (see Fig. 4.1) is used to define the problem. The coordinate systems of



(a) (b)

Figure 4.1: Calibrating the pose of the laser pointer. (a) Diagram of the calibration. (b) A laser spot appears in the image when the laser pointer points to the optical center of the camera.

Board A and the camera are denoted by C_a and C_0. The transformation from C_a to C_0 is $\mathbf{T}_{a\_0}$. The laser pointer's pose is described by the location P and the direction vector $\mathbf{n}$.

4.3 Computation of laser pointer parameters

Normally you don't have a single heading 2 or 3. With the fundamentals mentioned before, the following relations can be draw:

$$p^{board} = \mathbf{H} p^{image} \tag{4.1}$$

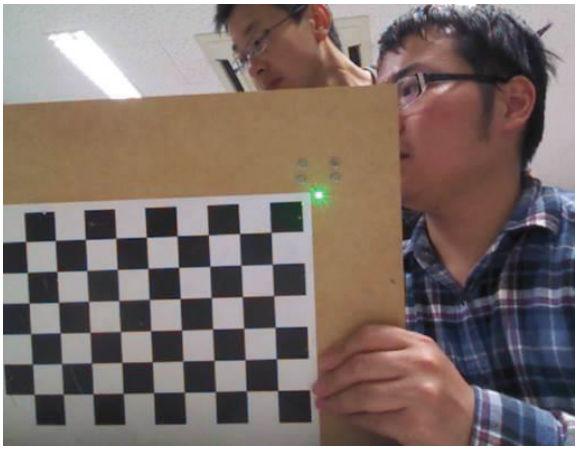
$$P^0 = \mathbf{T}_{a\_0} P^a \tag{4.2}$$

$$\mathbf{n}^0 \cong -P^0 \tag{4.3}$$

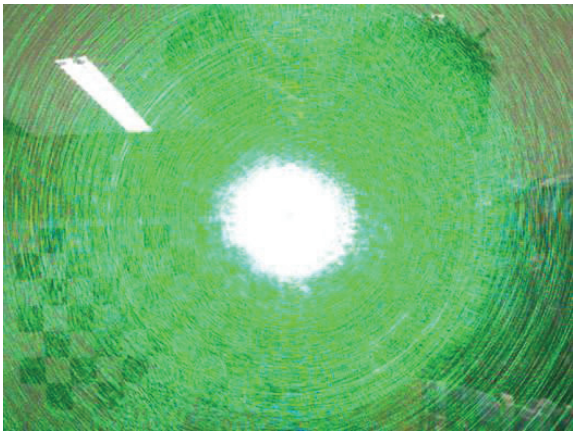
$$\mathbf{n}^a = \mathbf{R}_{a\_0}{}^{-1} \mathbf{n}^0 \tag{4.4}$$

Here, p^{board} and p^{image} are the 2D point p on the plane of Board A and the image plane, respectively. $\mathbf{H}$ is a homography matrix between the board and image planes. In Equation (4.3), P^0 is a nonhomogeneous coordinate. Using the method proposed by Zhang [7, 12], the transformation $\mathbf{T}_{a\_0}$ and the homography matrix $\mathbf{H}$ can be calculated. Thus, with the image of the checkered pattern, $\mathbf{H}$, p^{image}, $\mathbf{R}_{a\_0}$, and $\mathbf{t}_{a\_0}$ can be obtained. Since the board plane is the plane $z = 0$ in C_a, the 3D point P^a can be obtained from the 2D point p^{board}. Therefore, using Equations (4.1), (4.2), (4.3), and (4.4), the pose of the laser pointer, (P^a, $\mathbf{n}^a$), can be estimated.

4.4 Experiments



(a)



(b)

Figure 4.2: Calibrating the pose of the laser pointer. (a) When the laser pointer does not point to the optical center of the camera. (b) A laser spot appears in the image when the laser pointer points to the optical center of the camera.

To calibrate the pose of the laser pointer, the laser is projected in the general direction of the camera as shown in Fig. 4.1. Then, the ray's direction is tuned by

adjusting the orientation of Board A so that the ray can be seen by the camera lens; that is, the laser pointer is pointing exactly at the optical center of the camera as required. The result is a laser spot as shown in Fig. 4.2 (b) appearing in the image captured by the camera. Then, the laser is turned off and an image is captured with the checkered pattern and the hole p. Even when the laser ray is deviating a little from the optical center of the camera, an image likes Fig. 4.2 (a) instead of Fig. 4.2 (b) will be captured by the camera.

4.5 Estimation of the computational error

The following estimations of the relative pose parameters rely on the estimation of the laser pointer's pose. Thus, the accuracy of the latter has a significant effect on the overall precision. Ideally, the laser should be pointing at the camera's optical center when the laser spot appears in the center of the image. However, in the actual experiments, the estimation will be slightly influenced by the lens's refraction, introducing experimental error. Moreover, during operation, it is difficult to confirm that a laser spot such as that in Fig. 4.2(b) is located exactly in the center of the image. However, we can estimate the upper error limit of the laser pointer's orientation which is denoted by $\Delta\Theta_{max}$.

When a laser spot appears in the image, it is confirmed that the laser has at least passed through the lens, even if it is not projected directly onto the optical center. Thus, the radius of the camera lens can be used in the calculation of the upper error limit. As shown in Fig. 4.3, the radius of the camera lens is denoted by r and the distance between the board and the camera is L. L is approximately perpendicular to r in Fig. 4.3; thus,

$$\Delta\Theta_{\max} = \arctan\left(\frac{r}{L}\right)$$

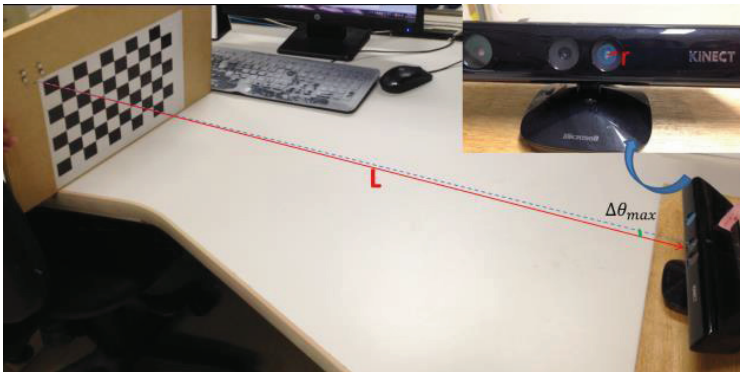


Figure 4.3: Estimating the upper error limit of the laser pointer's orientation.

where $r \approx 0.5$ cm in our experiment. In order to capture a clear, large image of the pattern, L is usually set to about 0.8 m in the experiment. In this case, the upper error limit is just $\Delta\Theta_{\max} \approx 0.36^{\circ}$. Note that here the initial values of the parameters (${}^{a}P_0, {}^{a}\mathbf{n}_0$) of the laser ray are estimated. They are refined in subsequent processes.

4.6 Conclusion and contribution

In this chapter, a novel technique is proposed to estimate the pose of a laser pointer, which is fixed on a calibration board. It is based on the principals that according to laser's unidirectional property and ideal central perspective camera model, laser ray should be pointing to the optical center of camera when it is viewed by the camera. It is a practical and flexible method to calibrate the pose of laser pointer. Moreover, it is very effective and used throughout this thesis.

CHAPTER 5

Calibrating Non-overlapping Cameras with One Calibration Pattern

In this chapter, we propose a novel method of calibrating the relative pose of two non-overlapping cameras with a laser pointer. A laser pointer is fixed at a calibration board so that its pose at the coordinate system of the calibration board can be obtained. While one of the cameras observes the calibration board, the intersection of the ray of the laser pointer with scenes is observed by the other. Thus, the view field of both cameras is connected, and the relative pose between the cameras can be estimated. In contrast with other methods, the proposed method is simple and practical, and especially suitable to the calibration of non-overlapping in-vehicle cameras at a factory or a garage.

This chapter is organized as follows. Firstly the fundamentals about quaternion and Levenberg-Marquardt method will be introduced beforehand. Then the key principle of our novel calibration method of non-overlapping cameras is drawn. Next, the results and performance of this method are shown on both synthetic data and real-world experiment. Finally, come to a conclusion.

5.1 Quaternion for rotation

Up until now we represent a rotation about an axis through the origin by a 3x3 orthogonal matrix with determinant 1. However, the matrix representation seems redundant because only four of its nine elements are independent. Also the geometric interpretation of such a matrix is not clear until we carry out several steps of calculation

to extract the rotation axis and angle. Furthermore, to compose two rotations, we need to compute the product of the two corresponding matrices, which requires twenty-seven multiplications and eighteen additions. So first of all, we would like to introduce a little about the quaternion, which will be used in this thesis.

Quaternions are very efficient for analyzing situations where rotation are involved. A quaternion is a 4-tuple, which is a more concise representation than a rotation matrix. Its geometric meaning is also more obvious as the rotation axis and angle can be trivially recovered. The quaternion algebra to be introduced will also allow us to easily compose rotation. This is because quaternion composition takes merely sixteen multiplications and twelve additions.

The development of quaternions is attributed to W. R. Hamilton in 1843. Legend has it that Hamilton was walking with his wife Helen at the Royal Irish Academy when he was suddenly struck by the idea of adding a fourth dimension in order to multiply triples. Excited by this breakthrough, he carved the newfound quaternion equations $i^2 = j^2 = k^2 = ijk = -1$ into the stone of the bridge.

A common notation for writing a quaternion is: $q = (s, \mathbf{v})$ where s is a scalar and v a vector. From this it is easy to see that a real number x can be written as the following quaternion $q = (x, \mathbf{0})$. Similarly a vector v is also the quaternion $q = (0, \mathbf{v})$. Operations can be defined on quaternions:

- addition $q + q' = (s + s', \mathbf{v} + \mathbf{v}')$

- multiplication $q \times q' = (ss' - \mathbf{v} \cdot \mathbf{v}',\ s'\mathbf{v} + s\mathbf{v}' + \mathbf{v} \times \mathbf{v}')$

- conjugate $\overline{q} = (s, -\mathbf{v})$

- norm $\|q\|^2 = \overline{q} \times q = q \times \overline{q} = s^2 + \|\mathbf{v}\|^2$

- inverse $q^{-1} = \dfrac{\overline{q}}{\|q\|^2}$

It can be proven that unit quaternions represent 3D rotation. If the rotation is represented by the quaternion q, then a point $p = (0, \mathbf{p})$ is rotated into a point p' given by: $p' = q \times p \times \overline{q} = q \times p \times q^{-1}$.

To convert from an axis and angle representation to quaternion, use the following formula where u is a unit vector describing the axis and θ is the rotation angle around u:

$$q = (\cos(\theta/2),\ \sin(\theta/2)\ \mathbf{u})$$

To convert from Euler angles to quaternion, just apply the above formula and respect the order of multiplications. For example with the convention ω, Φ and κ:

$$q = q_\omega \times q_\phi \times q_\kappa$$

$$q_\omega = (\cos(\omega/2),\ (\sin(\omega/2), 0, 0))$$

$$q_\phi = (\cos(\phi/2),\ (0, \sin(\phi/2), 0))$$

$$q_\kappa = (\cos(\kappa/2),\ (0, 0, \sin(\kappa/2)))$$

To convert from a rotation matrix **M** to quaternion q can be worked on using these formulas:

$$q = (w,\ (x, y, z))$$

$$\mathbf{M} = \begin{bmatrix} 1-2y^2-2z^2 & 2xy-2wz & 2xz+2wy \\ 2xy+2wz & 1-2z^2-2x^2 & 2yz-2wx \\ 2xz-2wy & 2yz+2wx & 1-2x^2-2y^2 \end{bmatrix}$$

Checking of course that the norm of the quaternion is 1, i.e. $w^2 + x^2 + y^2 + z^2 = 1$.

5.2 Levenberg-Marquardt method for nonlinear optimization

As the non-linear optimization method of Levenberg-Marquardt method is extensive used later, we firstly give a short introduction to it. Levenberg and later Marquardt suggested to use a damped Gauss-Newton method. The step $\mathbf{h}_{lm}$ is defined by the following equation:

$(\mathbf{J}^T\mathbf{J} + u\mathbf{I})\mathbf{h}_{lm} = -\mathbf{g}$ with $\mathbf{g} = \mathbf{J}^T f$ and $u \geq 0$

Here, $\mathbf{J} = \mathbf{J}(\mathbf{x})$ and $\mathbf{f} = \mathbf{f}(\mathbf{x})$. The damping parameter u has several effects:

(1) For all u>0 the coefficient matrix is positive definite, and this ensures that $\mathbf{h}_{lm}$ is a descent direction.

(2) For large values of u we get $\mathbf{h}_{lm} \approx -\frac{1}{u}\mathbf{g} = -\frac{1}{u}\mathbf{F}'(\mathbf{x})$ is a short step in the steepest descent direction. This is good if the current iterate is far from the solution.

(3) If u is very small, then $\mathbf{h}_{lm} \approx \mathbf{h}_{gn}$, which is a good step in the final stages of the iteration, when $\mathbf{x}$ is close to $\mathbf{x}^*$. If $F(\mathbf{x}^*) = 0$ (or very small), then we can get (almost) quadratic final convergence.

Thus, the damping parameter influences both the direction and the size of the step, and this leads us to make a method without a specific line search. The choice of initial u-value should be related to the size of the elements in $\mathbf{A}_0 = \mathbf{J}(\mathbf{x}_0)^T \mathbf{J}(\mathbf{x}_0)$, eg by letting

$$u_0 = \tau \cdot \max_i \{a_{ii}^0\},$$

Where τ is chosen by the user. During iteration the size of u can be updated. The updating is controlled by the gain ratio

$$\sigma = \frac{F(\mathbf{x}) - F(\mathbf{x} + \mathbf{h}_{lm})}{L(\mathbf{0}) - L(\mathbf{h}_{lm})}$$

Where the denominator is the gain predicted by a linear model.

$$\begin{aligned} L(\mathbf{0}) - L(\mathbf{h}_{lm}) &= -\mathbf{h}_{lm}{}^T \mathbf{J}^T \mathbf{f} - \frac{1}{2} \mathbf{h}_{lm}{}^T \mathbf{J}^T \mathbf{J} \mathbf{h}_{lm} \\ &= -\frac{1}{2} \mathbf{h}_{lm}{}^T (2\mathbf{g} + (\mathbf{J}^T \mathbf{J} + u\mathbf{I} - u\mathbf{I}) \mathbf{h}_{lm}) \\ &= \frac{1}{2} \mathbf{h}_{lm}{}^T (u\mathbf{h}_{lm} - \mathbf{g}) \end{aligned}$$

Note that both $\mathbf{h}_{lm}{}^T \mathbf{h}_{lm}$ and $-\mathbf{h}_{lm}{}^T \mathbf{g}$ are positive, so $L(\mathbf{0}) - L(\mathbf{h}_{lm})$ is guaranteed to be positive.

A large value of σ indicated that $L(\mathbf{h}_{lm})$ is a good approximation to $F(\mathbf{x} + \mathbf{h}_{lm})$, and we can decrease u so that the next Levenberg-Marquardt step is closer to the Gauss-

Newton step. If σ is small (maybe even negative), then $L(\mathbf{h}_{lm})$ is a poor approximation, and we should increase u with the twofold aim of getting closer to the steepest descent direction and reducing the step length. These goals can be met in different ways.

The stopping criteria for the algorithm should reflect that at a global minimizer we have $\mathbf{F'(x^*)} = \mathbf{g(x^*)} = \mathbf{0}$, so we can use

$\|\mathbf{g}\|_{\infty \leq} \varepsilon_1$

Where ε_1 is a small, positive number, chosen by the user. Another relevant criterion is to stop if the change in $\mathbf{x}$ is small,

$\|\mathbf{x}_{new} - \mathbf{x}\| \leq \varepsilon_2(\|\mathbf{x}\| + \varepsilon_2)$

This expression gives a gradual change from relative step size ε_2 when $\|\mathbf{x}\|$ is large to absolute step size ${\varepsilon_2}^2$ if $\mathbf{x}$ is close to $\mathbf{0}$. Finally, as in all iterative processes we need a safeguard against an infinite loop,

$k \geq k_{\max}$

Also ε_2 and $k_{\max}$ are chosen by the user.

The last two criteria come into effect eg if ε_1 is chosen so small that effects of rounding errors have large influence. This will typically reveal itself in a poor accordance between the actual gain in F and the gain predicted by the linear model, and will result in u being augmented in every step. The strategy for augmenting u implies that in this case u grows

fast, resulting in small $\|\mathbf{h}_{lm}\|$, and the process will be stopped by the stopping criteria $\|\mathbf{g}\|_{\infty\leq}\varepsilon_1$.

The algorithm is summarized below.

Levenberg–Marquardt method

begin
  $k := 0$; $\nu := 2$; $\mathbf{x} := \mathbf{x}_0$
  $\mathbf{A} := \mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x})$; $\mathbf{g} := \mathbf{J}(\mathbf{x})^\top \mathbf{f}(\mathbf{x})$
  found $:= (\|\mathbf{g}\|_\infty \leq \varepsilon_1)$; $\mu := \tau * \max\{a_{ii}\}$
  while (**not** *found*) **and** $(k < k_{\max})$
    $k := k+1$; Solve $(\mathbf{A} + \mu\mathbf{I})\mathbf{h}_{\text{lm}} = -\mathbf{g}$
    if $\|\mathbf{h}_{\text{lm}}\| \leq \varepsilon_2(\|\mathbf{x}\| + \varepsilon_2)$
      found $:=$ **true**
    else
      $\mathbf{x}_{\text{new}} := \mathbf{x} + \mathbf{h}_{\text{lm}}$
      $\varrho := (F(\mathbf{x}) - F(\mathbf{x}_{\text{new}}))/(L(\mathbf{0}) - L(\mathbf{h}_{\text{lm}}))$
      if $\varrho > 0$ {step acceptable}
        $\mathbf{x} := \mathbf{x}_{\text{new}}$
        $\mathbf{A} := \mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x})$; $\mathbf{g} := \mathbf{J}(\mathbf{x})^\top \mathbf{f}(\mathbf{x})$
        found $:= (\|\mathbf{g}\|_\infty \leq \varepsilon_1)$
        $\mu := \mu * \max\{\frac{1}{3}, 1 - (2\varrho - 1)^3\}$; $\nu := 2$
      else
        $\mu := \mu * \nu$; $\nu := 2 * \nu$
end

5.3 Plücker coordinates

Three-dimensional lines are represented either by two distinct 3D points, or by 6-vectors of so-called Plücker coordinates. We use the following convention. Let $\mathbf{A}$ and $\mathbf{B}$ be two 3D points, in homogeneous coordinates. The Plücker coordinates of the line spanned by them, are then given as:

$$\begin{pmatrix} B_4\overline{\mathbf{A}} - A_4\overline{\mathbf{B}} \\ \overline{\mathbf{A}} \times \overline{\mathbf{B}} \end{pmatrix}$$

Where $\overline{\mathbf{A}}$ is the 3-vector consisting of the first three coordinates of $\overline{\mathbf{A}}$ and likewise for $\overline{\mathbf{B}}$.

The action of displacements on Plücker coordinates is as follows. Let $\mathbf{t}$ and $\mathbf{R}$ be a translation vector and rotation matrix that map points according to:

$$\mathbf{Q} \mapsto \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{Q}$$

Plücker coordinates are then mapped according to:

$$\mathbf{L} \mapsto \begin{pmatrix} \mathbf{R} & \mathbf{0} \\ -[\mathbf{t}]_\times \mathbf{R} & \mathbf{R} \end{pmatrix} \mathbf{L}$$

Where $\mathbf{0}$ is the 3x3 matrix composed of zeroes.

Two lines cut one another exactly if

$$\mathbf{L}_2^T \begin{pmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \\ \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \end{pmatrix} \mathbf{L}_1 = 0$$

5.4 Coplanar constraint method

In this section, the first calibration method proposed is explained in detail. Two non-overlapping cameras are calibrated with one pattern and a laser pointer. A coplanar constraint is used to estimate the relative pose of the two cameras.

5.4.1 Problem definition

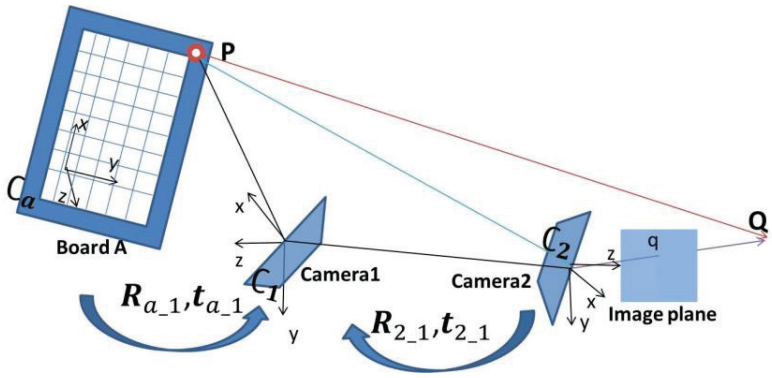


Figure 5.1: Calibrating non-overlapping cameras with a laser pointer.

The problem is illustrated in Fig. 5.1. The two non-overlapping cameras are labeled Camera 1 and Camera 2. The coordinate systems of Board A, Camera 1, and Camera 2 are C_a, C_1, and C_2, respectively. After the cameras' internal parameters are computed using the conventional method [7], the extrinsic parameters $(\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1})$ between the two cameras are estimated.

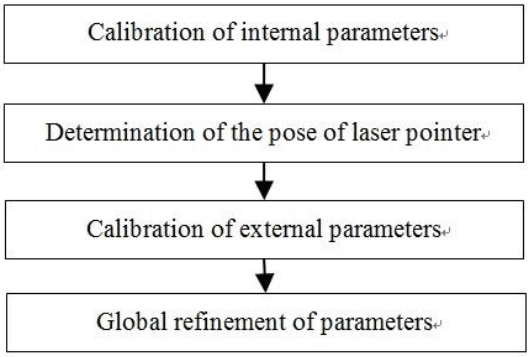


Figure 5.2: Calibrating non-overlapping cameras with a laser pointer.

5.4.2 Process

The cameras' intrinsic parameters are calibrated beforehand using a calibration board and the conventional method described in [7]. The initial pose of the laser pointer mounted on the calibration board has been estimated using the method mentioned in chapter 4. Then, we calibrate the extrinsic parameters (relative pose) of the two cameras using the laser ray connecting the FOVs of the two cameras as a reference. The method, which uses a coplanar constraint, is proposed to compute the extrinsic parameters and the laser pointer's pose based on this concept. A block diagram of the process is presented in Fig. 5.2.

5.4.3 Coplanar constraint for relative pose via a laser ray

We consider a ray of light $\overrightarrow{PQ}$ starting from point P that intersects with an external feature at point Q; Q is observed by Camera 2 as a laser dot q. The Plücker coordinates ℓ^a of the laser ray in C_a is estimated in Chapter 4. The transformation $(\mathbf{R}_{a\_1}, \mathbf{t}_{a\_1})$ can be calculated using the image of Board A captured by Camera 1 using the method described in [12]. Thus, the Plücker coordinates ℓ^1 of the laser ray in C_1 can be computed by multiplying ℓ^a by a 6×6 matrix [17]:

$$\ell^1 = \begin{pmatrix} \mathbf{R}_{a\_1} & \mathbf{0} \\ -[\mathbf{t}_{a\_1}]_\times & \mathbf{R}_{a\_1} \end{pmatrix} \ell^a \qquad (5.1)$$

The position of the image of the laser dot q gives the back-projection ray $\mathbf{q}^2$ in the second view (Camera 2) that must be incident with ℓ^1.

$$\mathbf{q}^2 \cong dist^{-1}\left(\mathbf{M}^{-1}q\right) \qquad (5.2)$$

Here, $\mathbf{M}$ is a perspective projection matrix and $dist^{-1}$ is a reverse processing of the distortion function with respect to Camera 2's internal parameters.

Next, δ^2 is used to denote the Plücker coordinates of the laser dot's back projection ray in the reference frame C_2, which are expressed as $\delta^2 = (\mathbf{q}^2; \mathbf{0} \times \mathbf{q}^2)$. The Plücker coordinates δ^1 of the back-projection ray in C_1 can be computed by multiplying δ^2 by a 6×6 matrix that contains the relative pose $\left(\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1}\right)$.

$$\delta^1 = \begin{pmatrix} \mathbf{R}_{2\_1} & \mathbf{0} \\ -\left[\mathbf{t}_{2\_1}\right]_\times & \mathbf{R}_{2\_1} \end{pmatrix} \delta^2 \qquad (5.3)$$

Based on the fact that the laser ray and laser dot's back-projection rays intersect at point Q, ℓ^1 and δ^1 should be coplanar. Thus, the two Plücker coordinates ℓ^1 and δ^1 satisfy the following necessary and sufficient condition [18]:

$$\left(\delta^1 \middle| \ell^1\right) = 0 \qquad (5.4)$$

Substituting ℓ^a and δ^2 with $(\mathbf{n}^a; P^a \times \mathbf{n}^a)$ and $(\mathbf{q}^2; \mathbf{0} \times \mathbf{q}^2)$, and consolidating Equations (5.1), (5.2) and (5.3), allows Equation (5.4) to be expanded as

$$\left(dist^{-1}\left(\mathbf{M}^{-1}q\right)\right)^T \mathbf{R}_{2\_1}{}^{-1}\left[\mathbf{t}_{2\_1} - \mathbf{R}_{a\_1}P^a + \mathbf{t}_{a\_1}\right]_\times \mathbf{R}_{a\_1}\mathbf{n}^a = 0 \qquad (5.5)$$

where P^a is used as a nonhomogeneous coordinate and $\left[\mathbf{t}_{2\_1}-\mathbf{R}_{a\_1}P^a+\mathbf{t}_{a\_1}\right]_\times$ is a 3×3 skew-symmetric matrix.

5.4.4 Relative pose estimation

For Equation (5.5), $\mathbf{M}$, P^a, and $\mathbf{n}^a$ are calibrated beforehand. $\mathbf{R}_{a\_1}$, $\mathbf{t}_{a\_1}$, and q can be obtained from each pattern image and corresponding laser spot images. $\mathbf{R}_{2\_1}$ and $\mathbf{t}_{2\_1}$ are unknown parameters.

By changing Board A's position and orientation, we obtain new observations $\left(\mathbf{R}_{a\_1},\mathbf{t}_{a\_1},q\right)_{(k)}$ and corresponding equations:

$$F_{1(k)}=\left(dist^{-1}\left(\mathbf{M}^{-1}q_{(k)}\right)\right)^T\mathbf{R}_{2\_1}{}^{-1}[\mathbf{t}_{2\_1}-\mathbf{R}_{a\_1(k)}P^a+\mathbf{t}_{a\_1(k)}]_\times\mathbf{R}_{a\_1(k)}\mathbf{n}^a \qquad (5.6)$$

Finally, the relative pose between the two cameras can be computed by minimizing the following objective function:

$$f\left(\mathbf{R}_{2\_1},\mathbf{t}_{2\_1}\right)=\sum_{k=1}^{K}\left(F_{1(k)}{}^2+\lambda\left(1-a^2-b^2-c^2-d^2\right)^2\right) \qquad (5.7)$$

where K is the total number of observations; $P^a,\mathbf{n}^a$ is fixed as $P^a{}_0,\mathbf{n}^a{}_0$; and the rotation $\mathbf{R}_{2\_1}$ is represented in the form of a quaternion $(a;b,c,d)$.

The nonlinear objective function above is optimized using the Levenberg-Marquardt method for preciseness.

5.4.5 Parameters refinement

The laser ray's pose is estimated in Chapter 4. The cameras' relative pose is estimated by fixing the laser ray's pose in last section (Section 5.4.4). As explained in Chapter 4, experimental errors may be introduced in the process of determining the laser ray's pose (P^a , $\mathbf{n}^a$). In this section, the laser ray's pose and the cameras' relative pose will be refined together using their initial values estimated in Chapter 4 and last subsection, respectively. Specifically, we minimize the following objective function using the Levenberg-Marquardt method.

$$f\left(\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1}, P^a, \mathbf{n}^a\right) = \sum_{k=1}^{K}\left({F_{1(k)}}^2 + \lambda\left(1 - a^2 - b^2 - c^2 - d^2\right)^2\right) \tag{5.8}$$

λ in Equations (5.7) and (5.8) is a coefficient, which maintains a balance between the coplanar constraint and the rotation matrix constraint. An empirical value of $\lambda = 0.01$ is used.

5.5 Experiments and results

In this section we show the experimental results of the proposed method. We first evaluate the performance of our method on synthetic data. Then, we apply this calibration method in an in-vehicle camera system which can be used for extracting driver's viewpoint.

5.5.1 Evaluation on synthetic data

We take the intrinsic parameters calibrated from real ones as the intrinsic parameters of synthetic cameras. We set the image resolution to 640×480 . Then we

create a synthetic pattern with 9×6 corners. The side length of pattern's gridding is set to be 0.026m. We set the laser pointer perpendicular to the chessboard and located nearly at the center of pattern. Concretely the source point of laser ray is $(0.117m,0.065m,0m)$ in the coordinate system of pattern. The extrinsic parameters' ground truth of two cameras is set as: translation $\mathbf{t}_{2\_1}=(0.1,0.1,-0.5)$ and roll, pitch, yaw of rotation $\mathbf{R}_{2\_1}=(159.61^{\circ},9.32^{\circ},173.55^{\circ})$, which is approximate to the real one of in-vehicle experiment configuration. During the simulation process, the pose $(\mathbf{R}_{a\_1},\mathbf{t}_{a\_1})_{(i)}$ of the pattern keeps changing. $\mathbf{t}_{a\_1(i)}$ is randomly chosen from these position: (-0.03m,-0 .02m,0.3m) , (-0.05m,-0 .03m,0.32m) , (-0.06m,-0. 01m,0.40m) , (-0.06m,0m, 0.40m) , (-0.08m,0.0 2m,0.35m) , (-0.01m,-0. 06m,0.35m) , (-0.06m,-0.05m,0.36m) , ((-0.03m,0 .02m,0.36m)) to make sure the pattern is seen by Camera1. Initially, pattern is looking towards Camera1. $\mathbf{R}_{a\_1(i)}$ is set to anticlockwise rotate θ_i around an unit vector $\mathbf{n}_i$ from the initial pose. The unit vector is randomly chosen in a uniform sphere. θ_i is randomly chosen from $(0.1,\pi/2)$ to make sure the pattern observed by Camera1. For every pose $(\mathbf{R}_{a\_1},\mathbf{t}_{a\_1})_{(i)}$ of the pattern, corners are projected to image of Camera1. To simulate the quantizing of digital image, the pixel coordinate of image points are changed to integer. Afterwards, the image points are attached with a White Gaussian Noise (AWGN) with a standard deviation σ. Then these image points are used to calculate the observation of pattern's pose $(\mathbf{R}_{a\_1'(i)},\mathbf{t}_{a\_1'(i)})$ using the method proposed by Zhang [7, 12].

So the observation of laser's pose in the coordinate of Camera1 can be calculated. On the other hand, a synthetic wall is created in front of Camera2 at the distance of 0.6m. The ground truth of laser's pose, pattern's pose and extrinsic parameters of cameras is used to calculate the laser spot's position on the wall. The laser spot is projected to image of Camera2. Quantizing and AWGN are also applied to the laser spot's image point to get its observation. Finally, our algorithm is applied on the noisy data. This method can adapt a wide range of initial value, so the initial value can be randomly given as long as the initial $\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1}$ corresponding to two cameras with non-overlapping FOV.

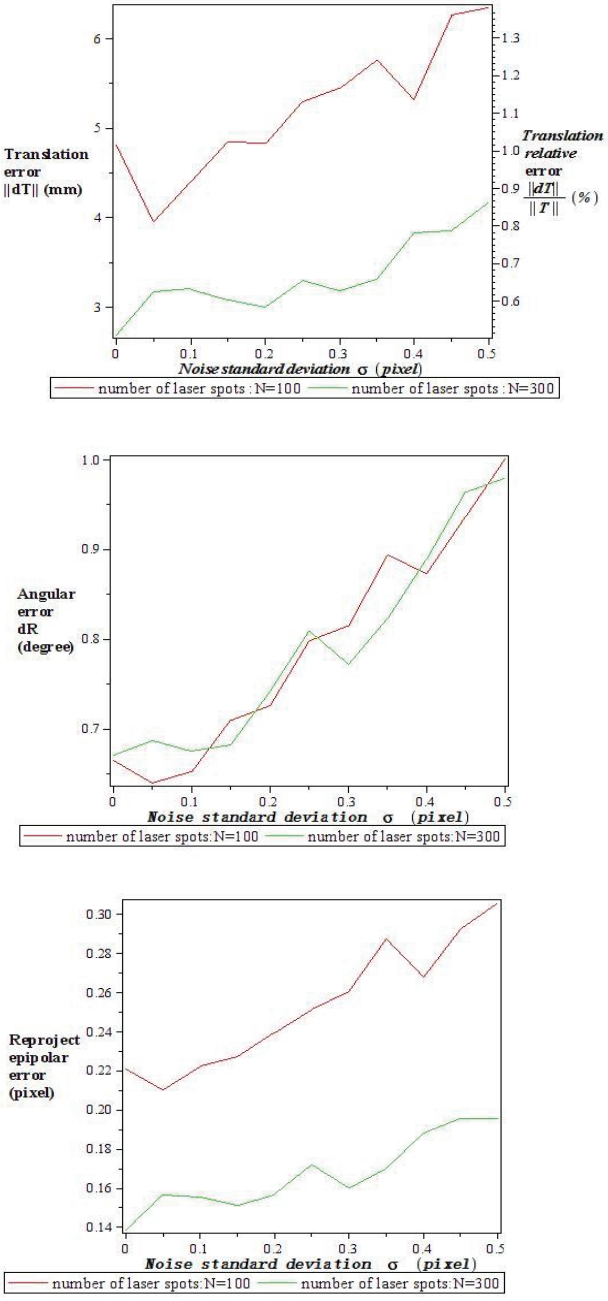


Figure 5.3: Results with synthetic data of different number of laser spots: extrinsic calibration error and reprojection epipolar error with respect to noise level. Note that, quantizing error of digital image has been considered before adding noise. And with real data, the noise standard deviation is below 0.1 pixel.

For every measure, N laser spots are observed. And for every σ value, 100 measures have been repeated. The calibration accuracy is given by the reprojection epipolar error, the norm $\|d\mathbf{T}\|$ of the translation error vector (and also $\|d\mathbf{T}\|/\|\mathbf{T}\|$ express as percentage of the magnitude of the camera baseline) and the angular error dR whose expression is given by:

$$d\mathbf{R} = d(\hat{\mathbf{R}}, \mathbf{R}^*) = \arccos\left(\frac{trace(\hat{\mathbf{R}}^T \mathbf{R}^*) - 1}{2}\right), \qquad (5.9)$$

Where $\mathbf{R}^*$ is the ground truth rotation and $\hat{\mathbf{R}}$ is the estimated rotation. And the reprojection epipolar error is explained latter during the evaluation of real data. Fig.5.3 shows the performances of the algorithm.

If the quantizing error and image noise is not considered, the translation relative error, angular error and reprojected epipolar error are $(4.70404\text{e-}005\,\%, 0.00433635^{o}, 1.7355\text{e-}005\,\text{pixel})$ and $(3.08981\text{e-}005\ \%, 0.00489588^{o}, 1.4638\text{e-}005\,\text{pixel})$, respectively, for 100 and 300 laser spots. After considering the quantizing error and image noise, the calibration results are shown in Fig.5.3. Comparing to the results proposed in [13], the accuracy in Fig.5.3 is a little lower. But here we have considered the quantizing error, which can lead to a uniform random error from -0.5 to 0.5 pixels. For example, in Fig.5.3, when the laser spot number is 100, translation error and epipolar error corresponding to standard deviation $\sigma = 0$ are bigger than that of standard deviation $\sigma = 0.05$. It is thought to be

caused by the quantizing error. If only the image noise ($\sigma = 0.1$) is considered, the translation relative error, angular error and reprojection epipolar error can reach $\left(0.199841\%, 0.211341^{o}, 0.0542977\,\text{pixel}\right)$ for 300 laser spots in our simulation. Besides, as shown in Fig.5.3, the calibration error is influenced by the number of laser spots used for calibration, especially for the translation relative error and reprojection epipolar error. So the calibration accuracy is expected to further improve by increasing the laser spot number.

5.5.2 Evaluation on real-world data

We applied our method to calibrate an in-vehicle camera system. This system consists of two Kinects. Here, we calibrate the relative pose of the RGB camera of the two Kinects. As shown in Fig. 1.2 (b), Camera1 was inside the vehicle and used to observe the driver. And Camera2 was outside the vehicle and used to observe the front of

Figure 5.4: Example images. Left is the laser source on the pattern captured by Camera 1 inside the vehicle. Right is the laser spot captured by Camera 2 outside the vehicle (in an ordinary scene).

the vehicle. The RGB resolution was set as 640×480. We used a 9×6 chessboard pattern with 54 corner points. The pattern was printed on an A4 paper. The two cameras were fixed on the vehicle. The chessboard was held in the Camera1's field of view. The laser line got through the windscreen and projected to the Camera2's field of view. Fig. 5.4 shows example images seen by the two cameras.

Since the ground truth of the relative pose between the cameras is unknown in this real-world experiment, we evaluate the experimental results based on the statistical indices of relative pose, ($\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1}$), and the value of objective function of equation (5.8).

Note that equation (5.8) is induced from the coplanar constraint of $\overrightarrow{PO_2}, \overrightarrow{PQ}, \overrightarrow{O_2Q}$, as shown in Fig. 5.1. It is the same as the epipolar constraint of conventional stereo. Here, we compute the epipolar errors, which measure the distance of the laser point, observed by camera 2 from the epipolar line.

With the synthetic data we have shown that the calibration accuracy can be greatly improved by increasing the number of observations. To check this, we compare the experiment results of different data volume, 48 spots, 96 spots and 192 spots respectively. For every data volume, 6 measures have been performed. The averages and standard deviations of relative position, relative orientation and reprojection error for different data volume are shown in Table 5.1. Although ground truth is unknown, when the number of laser spots increases, standard deviation obviously decreases. So we come to a conclusion that, the calibration accuracy can be improved by increasing the data volume. For 192 laser spots, the standard deviation of baseline is about 1.8 cm (3.08%)

and the average reprojection epipolar error is 0.7602 pixel with a standard deviation 0.0548 pixel. Comparing to the paper [11], where the epipolar error can reach 0.1 pixel after a global bundle-adjustment together with intrinsic parameters, our epipolar accuracy has yet to be improved. As we only concern about extrinsic parameters' calibration in this paper, the calibration accuracy is influenced by the error of intrinsic parameters during real world experiment. So a global bundle-adjustment together with intrinsic parameters is one of our future works.

Table 5.1: Experiment results corresponding to different data volume (48, 96 and 192 laser spots).

Data volume	Average and standard deviation	Baseline (m)	Relative orientation roll pitch yaw (degree)			Reprojection epipolar error (pixel)
48 laser spots	AVG	0.5923	159.627	9.374	173.863	0.8028
	STDEV	0.0368	0.480	1.597	0.513	0.1705
96 laser spots	AVG	0.6051	159.615	9.622	173.827	0.7676
	STDEV	0.0273	0.273	0.417	0.075	0.1366
192 laser spots	AVG	0.6029	159.381	9.503	173.766	0.7602
	STDEV	0.0179	0.092	0.139	0.039	0.0548

5.6 Conclusion and contribution

In this chapter, we proposed a novel method of calibrating the relative pose of cameras with non-overlapping field of views using a laser pointer fixed on calibration pattern. We show the effectiveness of this method on both synthetic data and an in-

vehicle camera system. Compared to other calibration methods of non-overlapping cameras, our proposed method is simple, practical, and especially suitable to the calibration of non-overlapping in-vehicle cameras at a factory or a garage, where the existing methods are difficult to be applied. For a camera network consist of more than two non-overlapping cameras, relative transformation between arbitrary two non-overlapping cameras can be estimated with our proposed method.

CHAPTER 6

Calibrating Non-overlapping Cameras with Two Calibration Patterns

This chapter proposes a new approach to calibrating inward and outward-facing in-vehicle cameras using a laser pointer and two calibration patterns. Since inward cameras, which observe the driver's behavior, and outward cameras, which observe the driver's field of view, do not have overlapping views, the conventional calibration methods for stereo cameras cannot be applied directly. A method is presented in which the two cameras can be connected by a laser emitted from a laser pointer. The laser pointer is mounted on a calibration board so that the laser ray's pose within the calibration board's coordinate system can be calculated. The collinear method is presented based on this idea. We compare the performance of the proposed collinear algorithm with both coplanar algorithm (proposed in Chapter 5) and the conventional mirror-based method through simulations and experiments. Finally the laser-based collinear method is applied to the calibration of an in-vehicle camera system. In contrast to other methods, the proposed method is simple, practical, and especially well suited to the calibration of non-overlapping in-vehicle cameras in a factory or garage.

6.1 Collinear constraint method

In this section, the second calibration method is proposed. We calibrate two non-overlapping cameras with two patterns and a laser pointer. A collinear constraint is used to estimate the relative pose of the two cameras.

6.1.1 Problem definition

The problem is illustrated in Fig. 6.1. There are two cameras, Camera 1 and Camera 2, which observe calibration boards A and B, respectively. A laser pointer is mounted on Board A. The coordinate systems of Board A, Board B, Camera 1, and Camera 2 are denoted by C_a, C_b, C_1, and C_2, respectively. The intrinsic parameters of the two cameras and the transformations $(\mathbf{R}_{a\_1}, \mathbf{t}_{a\_1})$ and $(\mathbf{R}_{b\_2}, \mathbf{t}_{b\_2})$ are computed using Zhang's method [7, 12].

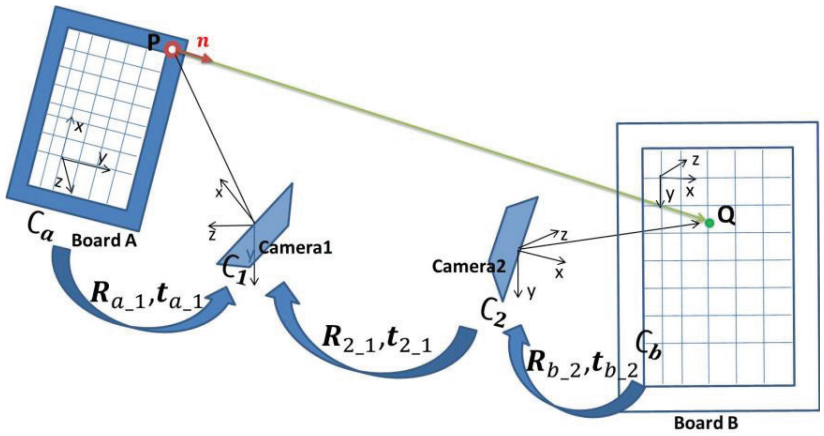


Figure 6.1: Calibrating non-overlapping cameras with a laser pointer and two patterns.

6.1.2 Process

Next, the method used to estimate the extrinsic parameters $(\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1})$ between the two cameras through a collinear constraint is explained. A block diagram of our calibration method is shown in Fig. 6.2.

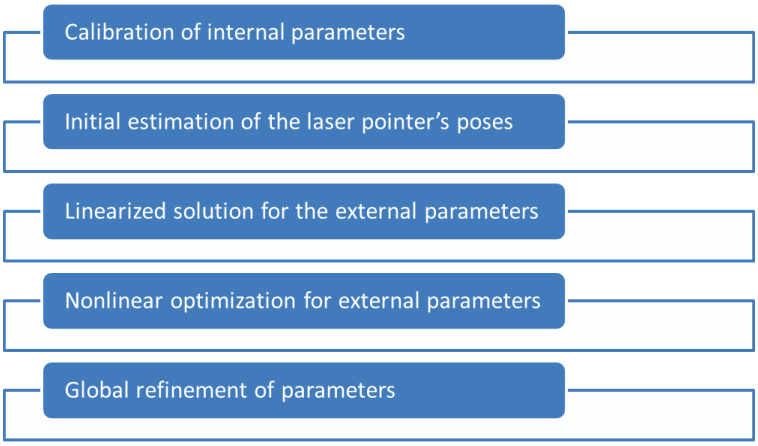


Figure 6.2: Flow diagram of the proposed calibration algorithm based on the collinear constraint method.

6.1.3 Linear estimation of the relative pose

To calibrate the relative pose between the two cameras, Board A and Board B are connected by a laser ray. The laser ray connects point P on Board A to point Q on Board B. As described in the last subsection, the Plücker coordinates ℓ^1 of the laser ray in C_1 can be represented by $\ell^a, \mathbf{R}_{a\_1}, \mathbf{t}_{a\_1}$ in terms of Equation (5.1). The 3D location Q^2 of the laser spot in C_2 can be computed similarly to the computation of P^0 in Chapter 4. Moreover, the coordinate Q^1 can be computed by multiplying Q^2 by a transformation matrix that contains the relative pose $(\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1})$.

$$Q^1 = \begin{bmatrix} \mathbf{R}_{2\_1} & \mathbf{t}_{2\_1} \\ \mathbf{0}^T & 1 \end{bmatrix} Q^2. \qquad (6.1)$$

Based on the fact that the 3D laser dot Q^1 lies in the laser ray ℓ^1, the following necessary and sufficient condition is obtained [18]:

$$\ell^1 \vee Q^1 = \left[\ell^1{}_\vee\right] Q^1 = 0. \qquad (6.2)$$

If

$$\ell^1 = (\mathbf{u}; \mathbf{v})$$

Then

$$\left[\ell^1{}_\vee\right] \overset{def}{=} \begin{bmatrix} [\mathbf{u}]_\times & \mathbf{v} \\ -\mathbf{v}^T & 0 \end{bmatrix}$$

where $[\mathbf{u}]_\times$ is the 3×3 skew-symmetric matrix.

Let $\ell^1 = (\mathbf{n}^1; P^1 \times \mathbf{n}^1)$, where P^1 is a nonhomogeneous coordinate in C_1; $\mathbf{n}^1 = [n_x{}^1, n_y{}^1, n_z{}^1]^T$, $P^1 = [P_x{}^1, P_y{}^1, P_z{}^1]^T$, $Q^2 = [Q_x{}^2, Q_y{}^2, Q_z{}^2]^T$, $\mathbf{t}_{2\_1} = [t_x, t_y, t_z]^T$, and $\mathbf{R}_{2\_1} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$.

Then, Equation (6.2) can be expanded as

$$\begin{cases} (r_{11} \cdot Q_x{}^2 + r_{12} \cdot Q_y{}^2 + r_{13} \cdot Q_z{}^2 + t_x - P_x{}^1) - \\ \dfrac{n_x{}^1}{n_y{}^1} \cdot (r_{21} \cdot Q_x{}^2 + r_{22} \cdot Q_y{}^2 + r_{23} \cdot Q_z{}^2 + t_y - P_y{}^1) = 0 \\ (r_{11} \cdot Q_x{}^2 + r_{12} \cdot Q_y{}^2 + r_{13} \cdot Q_z{}^2 + T_x - P_x{}^1) - \\ \dfrac{n_x{}^1}{n_z{}^1} \cdot (r_{31} \cdot Q_x{}^2 + r_{32} \cdot Q_y{}^2 + r_{33} \cdot Q_z{}^2 + t_z - P_z{}^1) = 0 \end{cases} \qquad (6.3)$$

where $P^a, \mathbf{n}^a$ is fixed as $P^a{}_0, \mathbf{n}^a{}_0$, and $\mathbf{n}^1, P^1$ can be calculated from $P^a, \mathbf{n}^a$ through the transformation $(\mathbf{R}_{a\_1}, \mathbf{t}_{a\_1})$. Q^2 can be calculated from the image of

Board B. In the two linear constraint equations of (6.3), the relative pose $\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1}$ is the only unknown. When a pair of images is captured by Camera 1 and Camera 2, one group of data $\left(\mathbf{n}^1, P^1, Q^2\right)_{(i)}$ is obtained.

Equation (6.3) is a basic linear equation for the relative pose of the two cameras. During the calibration process, we change the pose of Board A to obtain a series of data $\left(\mathbf{n}^1, P^1, Q^2\right)_{(i)}$, yielding two linear equations. However, there are 12 parameters in the relative pose $\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1}$. Thus, theoretically speaking, at least six sets of data are needed to solve the problem linearly. In practice, however, a solution is obtained through an over-constrained linear system of equations.

After solving the over-constrained linear system of equations for $\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1}$, SVD is used to approximate the estimated linear rotation matrix $\mathbf{R}_{2\_1}$ with the closest rotation matrix $\mathbf{R}_0$. Then, the rotation matrix $\mathbf{R}_0$ is fixed and is substituted back into Equations (6.3). Consequently, an over-constrained linear system of equations about $\mathbf{t}_{2\_1}$ is obtained. Next, $\mathbf{t}_0$ is solved through a least-squares approach and $\mathbf{R}_0$ is parameterized into a quaternion. The linear solution $\mathbf{R}_0, \mathbf{t}_0$ is used as the initial value in the latter non-linear optimization.

6.1.4 Nonlinear estimation of the relative pose

During the linear estimation of the relative pose, we did not strictly consider the constraint of the rotation matrix. However, it is useful to consider the collinear constraint

together with the constraint of the rotation matrix and represent the rotation matrix $\mathbf{R}_{2\_1}$ as a quaternion $\mathbf{R}_{2\_1} = (a;b,c,d)$. We arrive at the following constraint conditions:

$$\begin{cases} \ell^1 \vee Q^1 = 0 \\ a^2 + b^2 + c^2 + d^2 = 1 \end{cases} \tag{6.4}$$

An error function can be derived from Equations (6.4):

$$f\left(\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1}\right) = \sum_{i=1}^{N} \begin{pmatrix} \left(\left(\mathbf{T}_{2\_1} \cdot Q^2{}_{(i)} - P^1{}_{(i)}\right) \bullet \mathbf{n}^1{}_{(i)} - \left|\mathbf{T}_{2\_1} \cdot Q^2{}_{(i)} - P^1{}_{(i)}\right|\right)^2 \\ + \lambda\left(a^2 + b^2 + c^2 + d^2 - 1\right)^2 \end{pmatrix} \tag{6.5}$$

where N is the total number of groups of observations.

As described in Subsection 6.1.3, by changing the checkerboard's position and orientation, we obtain a data series $\left(\mathbf{n}^1, P^1, Q^2\right)_{(i)}$. The relative pose $\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1}$ is the only unknown in Equation (6.5).

Once the initial $\mathbf{R}_0, \mathbf{t}_0$ has been found via the linearity estimation described in Subsection 6.1.3, nonlinear optimization can be used to estimate the relative pose. The error function is minimized using the Levenberg-Marquardt method.

6.1.5 Parameter refinement

First, the laser ray's pose is estimated in Chapter 4. Then the cameras' relative pose is estimated by fixing the laser ray's pose as discussed in last section (Section 6.1.4). In Chapter 4, it was explained that in the process of determining the pose (P^a , $\mathbf{n}^a$) of the laser pointer, an experimental error may be introduced. Therefore, the pose of the laser pointer is refined together with the relative pose between the two cameras using their

initial values $\mathbf{R}_0, \mathbf{t}_0, \mathbf{n}^a{}_0, P^a{}_0$ (estimated in Chapter 4 and Section 6.1.4). This is achieved by minimizing the following objective function using the Levenberg-Marquardt method:

$$f\left(\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1}, P^a, \mathbf{n}^a\right) = \sum_{i=1}^{N} \left(\begin{array}{l} \left(\begin{array}{l} \left(\mathbf{R}_{2\_1} \cdot Q^2{}_{(i)} + \mathbf{t}_{2\_1} - \left(\mathbf{R}_{a\_1(i)} \cdot P^a + \mathbf{t}_{a\_1(i)}\right)\right) \bullet \left(\mathbf{R}_{a\_1(i)} \cdot \mathbf{n}^a\right) \\ - \left| \mathbf{R}_{2\_1} \cdot Q^2{}_{(i)} + \mathbf{t}_{2\_1} - \left(\mathbf{R}_{a\_1(i)} \cdot P^a + \mathbf{t}_{a\_1(i)}\right) \right| \end{array} \right)^2 \\ + \lambda\left(a^2 + b^2 + c^2 + d^2 - 1\right)^2 \end{array} \right) \tag{6.6}$$

The 3D point $Q^2{}_{(i)}$ is computed by $\mathbf{H}_{b\_2(i)}, \mathbf{T}_{b\_2(i)}$ in the same way that P^0 is calculated in Chapter 4. As in Equation (6.5), λ is a coefficient that maintains a balance between the collinear constraint and the rotation matrix constraint. If λ is too big, the collinear constraint will be too weak for convergence to occur. If λ is too small, the rotation constraint will be too weak, allowing a solution to be obtained that does not strictly meet the requirement of the rotation matrix. In this thesis, an empirical value of $\lambda = 0.0001$ was used here.

6.2 Experiments and results

In this section, the experimental results for the proposed collinear method are presented. First, we evaluated and compared the performance of the two proposed collinear method against the coplanar method and the mirror-based method using synthetic data. Then, the proposed method was applied to an in-vehicle camera system and was compared again with the mirror-based method using real-world data and a

configuration that models an in-vehicle camera system. Finally an experiment was carried out to prove the robustness of our methods against lighting condition.

6.2.1 Simulations

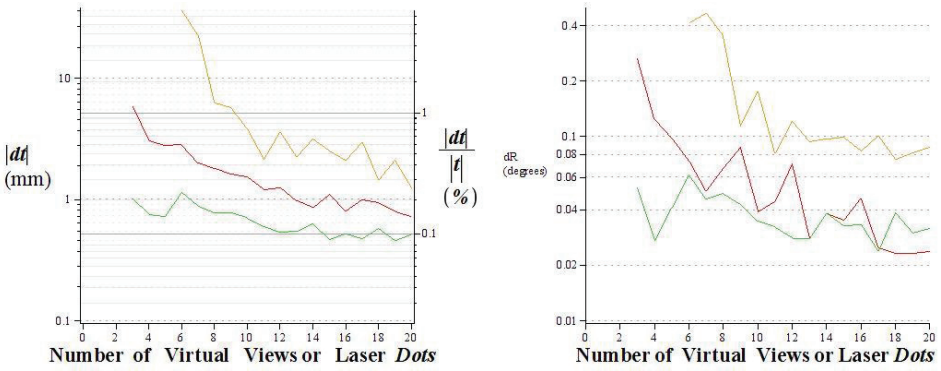
The experimental intrinsic parameters for the synthetic cameras (Camera 1 and Camera 2) are given in this section. The resolution of the two cameras was set as 640×480. Two synthetic patterns (Pattern A and Pattern B) with 9×6 grids were created virtually. The side length of a square in Pattern A was 0.026 m and that in Pattern B was 0.075 m. The laser pointer was located near the center of Pattern A and aligned perpendicularly to it. The laser ray's origin was thus $(0.117, 0.065, 0)$ m in the coordinate system of Pattern A. The extrinsic parameters' ground truth between the two cameras was set as the following: translation $\mathbf{t}_{2\_1} = (0.1, 0.1, -0.5)$ m and rotation $\mathbf{R}_{2\_1}$ expressed as roll, pitch, and yaw $(159.61^{\circ}, 9.32^{\circ}, 173.55^{\circ})$. These values were set according to the actual in-vehicle experimental configuration. Pattern A and Pattern B were viewed by Camera 1 and Camera 2, respectively. The following simulation setup was valid for both the coplanar and collinear method. However, for the coplanar constraint, Pattern B was considered a blank surface.

To simulate the experimental environment, Pattern B was set to face Camera 2 at a distance of 0.6 m. $\mathbf{t}_{b\_2}$ and $\mathbf{R}_{b\_2}$ were set as $\mathbf{t}_{b\_2} = (-0.3, -0.2, 0.6)$ m and $\mathbf{R}_{b\_2} = (0; \sqrt{2}/2, \sqrt{2}/2, 0)$, where the rotation is given as a quaternion. Pattern A was set 0.2 m from Camera 1 such that their relative translation $\mathbf{t}_{a\_1(i)}$ was

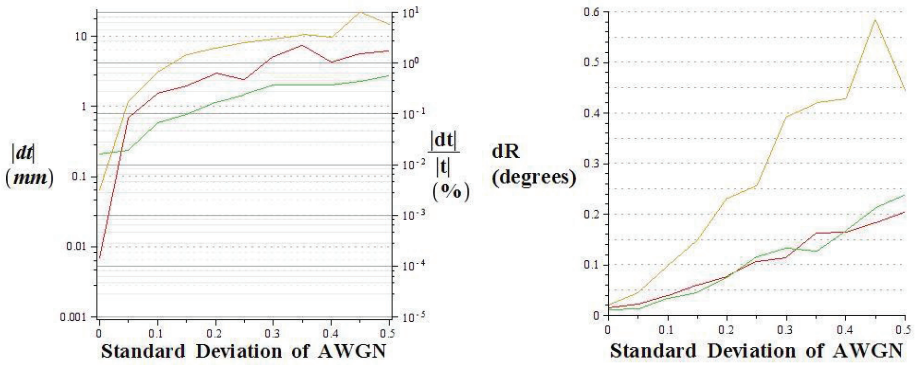
$\mathbf{t}_{a\_1} = (-0.104, -0.065, 0.2)$ m. During the simulation, the orientation of Pattern A relative to Camera 1, $\mathbf{R}_{a\_1(i)}$, evolved temporally as follows: Initially, Pattern A faced Camera 1 directly. Then, an anti-clockwise rotation angle $\theta_{A(i)}$ around a unit vector $\mathbf{n}_{A(i)}$ was introduced. The unit vector $\mathbf{n}_{A(i)}$ was randomly chosen from a unit sphere. The rotation angle $\theta_{A(i)}$ was randomly (uniformly) chosen from $(0.1, \pi/2)$ to ensure that Pattern A could be observed by Camera 1. For every pose $(\mathbf{R}_{a\_1}, \mathbf{t}_{a\_1})_{(i)}$ of Pattern A, Pattern A's corners were projected onto Image 1 of Camera 1, while Pattern B's corners were projected onto Image 2 of Camera 2. The point of intersection of the laser ray and Pattern B was calculated and was also projected onto Image 2 of Camera 2. Then, the position values of the image points of Image 1 and Image 2 (including the laser spot) were summed with an additive white Gaussian noise (AWGN) with a standard deviation σ. Finally, our algorithm was applied to this noise-inclusive data to provide an estimate of $\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1}$. Note that, for the collinear constraint method, the images of Pattern B's corner points captured after the noise was added were used to calculate the 3D coordinates of the laser spots.

To compare with the mirror-based method proposed by Rodrigues [15], a second setup was created. A synthetic mirror was added to our existing simulation environment in front of Camera 1. Via the mirror, Camera 1 could view the virtual image of Pattern B. The mirror was parameterized with $(\mathbf{n}_m, d_m)$, where d_m was the distance from Camera

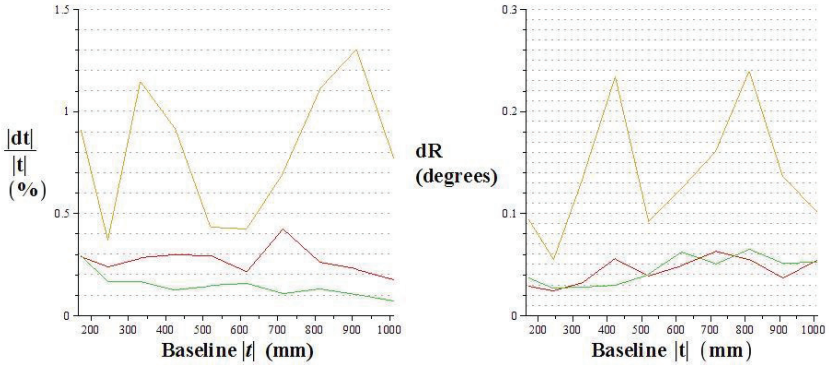
1's optical center to the mirror plane and $\mathbf{n}_m$ was the surface normal vector of the mirror plane in the coordinate system C_1. During the simulation, the pose of the mirror relative to Camera 1 changed over time. $d_{m(i)}$ was randomly (uniformly) chosen from $[0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5]$ m. $\mathbf{n}_{m(i)}$ was generated by rotating the vector $[0,0,1]^T$ by an angle $\theta_{m(i)}$ around a unit vector $\mathbf{n}_{m0(i)}$ in the anti-clockwise direction. The unit vector $\mathbf{n}_{m0(i)}$ was randomly chosen from a unit sphere. $\theta_{m(i)}$ was randomly (uniformly) chosen from $(0.1, \pi/6)$. The corner points of the mirrored Pattern B were created according to the mirror's pose $(\mathbf{n}_m, \theta_m)_{(i)}$ via a symmetry transformation [15]. Then, the corners of the mirrored Pattern B were projected onto Image 1 of Camera 1. As before, the position values of the image points of Image 1 and Image 2 were summed with an AWGN with a standard deviation σ. Finally, the mirror-based algorithm [15], for which the code was made publicly available by the authors, was applied to the noise-inclusive data to estimate $\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1}$.



(a) Influence of data volume



(b) Influence of noise level



(c) Influence of baseline

Figure 6.3: Evaluation of the accuracy for varying data volumes, noise levels and baselines for the laser and mirror based methods.

The calibration accuracy is given by the norm $\|\mathbf{dt}\|$ of the translation error vector

(and also $\|\mathbf{dt}\|/\|\mathbf{t}\|$ expressed as a percentage of the magnitude of the camera baseline) and the angular error dR which is given by [13]

$$dR = d(\hat{\mathbf{R}}, \mathbf{R}^*) = \arccos\left(\frac{trace(\hat{\mathbf{R}}^T \mathbf{R}^*) - 1}{2}\right) \tag{6.7}$$

where rotation $\mathbf{R}^*$ is the ground truth and $\hat{\mathbf{R}}$ is the estimated rotation. Fig. 6.3 shows the performances of the algorithms. For convenience, the three algorithms introduced in this thesis, the mirror-based algorithm, the laser-based collinear algorithm, and the laser-based coplanar algorithm, are denoted as Mirror, Laser nPQ, and Laser nPq, respectively.

To test the sensitivity to the volume of data for different algorithms, we set the translation $\mathbf{t}_{2\_1} = (0.1, 0.1, -0.5)$ m and the rotation $\mathbf{R}_{2\_1}$, $(159.61^{\circ}, 9.32^{\circ}, 173.55^{\circ})$ according to the actual in-vehicle experimental configuration. The standard deviation of the AWGN was fixed at 0.1 pixels (the noise standard deviation for real-world data is usually below 0.1 pixels [13]). In order to compare with Mirror, the data volume (the number of virtual patterns and laser dots) was changed from 3 to 20 which is exactly the same as the setup used in the simulation in [15]. For every data volume, the test was carried out 50 times and the average accuracy was obtained, as shown in Fig. 6.3 (a). For Mirror and Laser nPQ, solutions can be acquired with few data (N>=3) due to the linearity of the constraints. Since the constraint of Laser nPq is nonlinear, more data are necessary (N>=6). As shown in Fig. 6.3 (a), the orientation accuracy of Mirror and Laser

nPQ are similar, but Laser nPQ outperforms Mirror in terms of translation accuracy. It is also shown that Laser nPq is influenced significantly by the data volume as expected. In spite of slightly reduced accuracy, Laser nPq is also a new approach different from the Mirror method. Because Laser nPq does not use the second checkerboard outside the vehicle, it is more convenient than Laser NPQ and is thus an attractive method when the accuracy requirements are not high.

Next, the sensitivity to the noise level for the three algorithms was examined. The translation and orientation were set to the same as before. The data volume used was ten. σ was varied from 0.0 to 0.5. For every standard deviation, the test was carried out 50 times and the average accuracy was taken as shown in Fig. 6.3 (b). The accuracy of Laser nPq was restricted by the data volume. When there was no noise ($\sigma = 0$), Mirror obtained the highest accuracy. However, it was outperformed by Laser nPQ in terms of translation accuracy when noise was added ($\sigma > 0$). The reason is that the effect of the noise is amplified (especially for translation) during the mirroring projection when the two cameras' baseline is large (see Fig. 2.1(a)).

Finally, the sensitivity to the baseline for the three algorithms was tested. A data volume of ten was used again and σ was fixed at 0.1. $\mathbf{R}_{2\_1}$ was set to the same as before and $\mathbf{t}_{2\_1} = (0.1, 0.1, -z)$ m. z was varied from 0.1 to 1.0 (the baseline changed from 173.205 mm to 1009.95 mm). For every baseline, the test was carried out 50 times and the average accuracy was obtained as shown in Fig. 6.3 (c). As before the accuracy of Laser nPq was restricted by the data volume. The orientation accuracy for Mirror and

Laser nPQ are similar. The translation accuracy shows that Laser nPQ outperforms Mirror, especially when the baseline is large. However, there are fluctuations in the curves of Fig. 6.3 due to the addition of synthetic data with random noise (AWGN). Despite this, the curves reflect the overall trends.

6.2.2 Real-world experiments

Apply the laser-based method to in-vehicle cameras

Since the collinear method outperforms the coplanar method as demonstrated by the simulation, the collinear constraint method was applied to calibrate an in-vehicle camera system. This system consisted of two cameras (Camera 1 and Camera 2) positioned as shown in Fig. 1.2 (c). Here, the relative pose $\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1}$ of the two cameras was calibrated. As shown in Fig. 1.2 (c), Camera 1 was inside the vehicle and used to observe the driver and Camera 2 was outside the vehicle and was used to observe the driver's FOV. The RGB resolution was 640×480. Two 9×6 checkerboard patterns (Pattern A and Pattern B) were used. The side lengths of the squares in Pattern A and Pattern B were 0.026 m and 0.075 m, respectively. Pattern A was positioned in Camera 1's FOV and Pattern B was positioned in Camera 2's FOV (see Fig. 1.2 (c)). The laser ray was directed through the windscreen and onto Pattern B. Fig. 6.4 shows example images captured simultaneously by the two cameras.

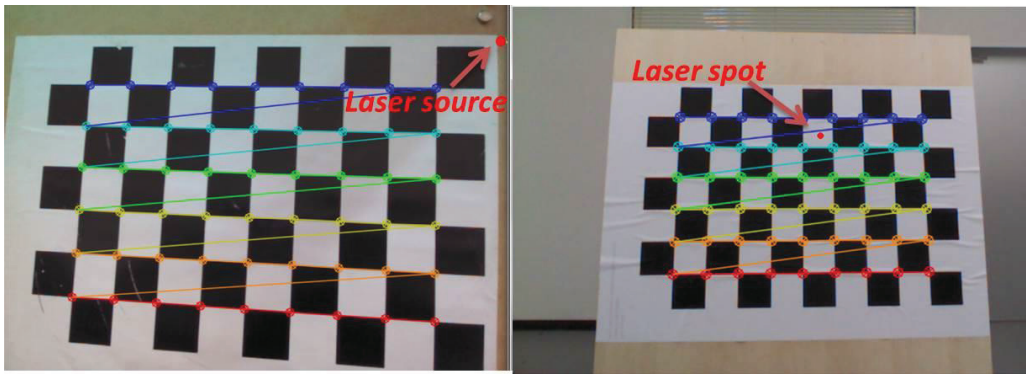


Figure 6.4: Example images. Left is the laser source captured by Camera 1 inside the vehicle. Right is the laser spot captured by Camera 2 outside the vehicle.

Since the ground truth of the relative pose between the cameras is unknown in the real-world experiments, the experimental results was evaluated based on the statistical indices of the relative pose ($\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1}$) and the mean reprojection error (MRE).

MRE is defined as follows: using the calibrated relative pose $\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1}$, the laser ray can be determined in the coordinate system of Camera 2. Thus, the point of intersection of the laser ray and Pattern B, $\hat{Q}$, can be calculated, as can the 3D coordinate of point Q. The Euclidean distance between these two points is denoted by d. The distance, $d_{(i)}$, for every group of observations, $\left(\mathbf{n}^1, P^1, Q^2\right)_{(i)}$, is obtained. MRE is thus defined as the average value of $d_{(i)}$ for every group of observations.

The experiment was carried out ten times, and each time 48 spots were used. The calibration results and corresponding MRE are shown in Table 6.1. When the

Table 6.1: Results for in-vehicle experiment

Data volume	Average and standard deviation	Relative position (m)			Relative orientation (Quaternion form)				Reprojection error (m)
48 Spots	AVG	0.0325	0.0704	0.8670	-0.1141	0.0513	0.9793	0.1589	0.0021
	STDEV	0.0095	0.0141	0.0099	0.0055	0.0016	0.0017	0.0084	0.0004

displacement between the two cameras is about 90 cm, the standard deviation is about 1 cm and the average reprojection error is 2.1 mm.

When carrying out the comparative experiment with the Mirror method, we were confronted by a problem existing in real-world environments: the calibration pattern image appeared smaller and blurred due to the windscreen, as shown in Fig. 2.1 (b). Thus, another comparative experiment was designed to imitate the configuration of in-vehicle cameras as described in the next subsection.

Comparative experiment using real-world data

Fig. 6.5 shows the camera configuration for the comparative experiment. To imitate the in-vehicle camera system, two intrinsic-parameter-calibrated cameras (Camera 1 and Camera 2) were set about 80 cm apart, pointing in opposite directions. Two positions, Position A and Position B, were chosen such that the two tips of Position A and Position B were able to be connected by two 20 cm parallel lines on a planar board, as shown in Fig. 6.5. Since the ground truth of the two cameras' relative pose was unknown, Camera 1 was moved from Position A to Position B along the parallel lines. The displacement was estimated using the methods under evaluation before and after the movement.

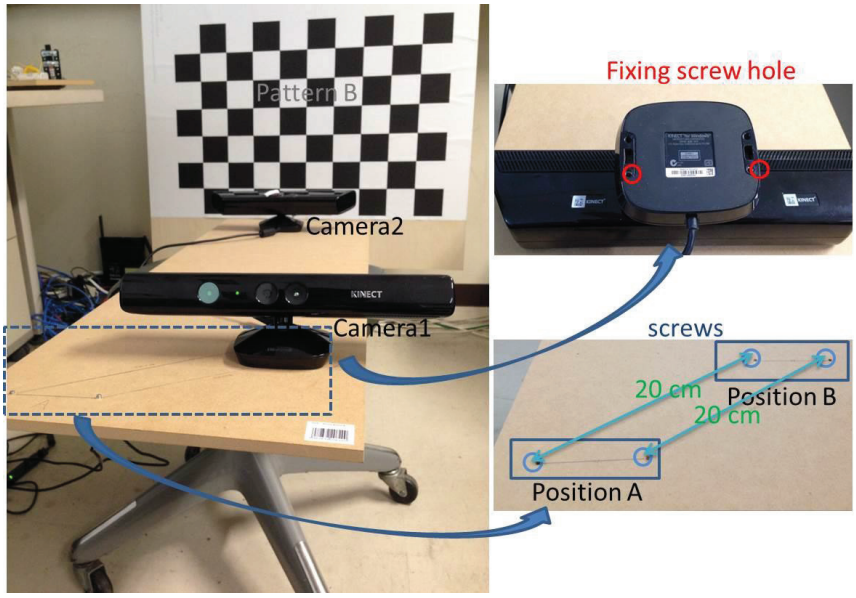


Figure 6.5: Setup used for the comparison experiment.

The two cameras were set with a certain angle so that Pattern B could be viewed by Camera 1 via the mirror without occlusion. Camera 2 and Pattern B's positions were fixed during the experiments. First, Camera 1 was set at Position A. The transformation $(\mathbf{R}_{2\_1}, \mathbf{t}_{2\_1})$ from Camera 2 to Camera 1 was estimated using our collinear method, and the transformation $(\mathbf{R}_{b\_1}, \mathbf{t}_{b\_1})$ from Pattern B to Camera 1 was estimated using the mirror-based method employing the code mentioned previously [15]. After moving Camera 1 from Position A to Position B, the transformations, and thus the displacements, were estimated again using the two methods for comparison. The test was carried out ten times and the results were averaged, as shown in Table 6.2. The result obtained using the proposed collinear method was closer to the ground truth and had a smaller deviation. As shown in Fig. 2.1 (a), when the baseline between the two cameras was large, the image of

the calibration pattern became small, greatly influencing the accuracy of the pose estimation.

Table 6.2: Results of comparison experiment

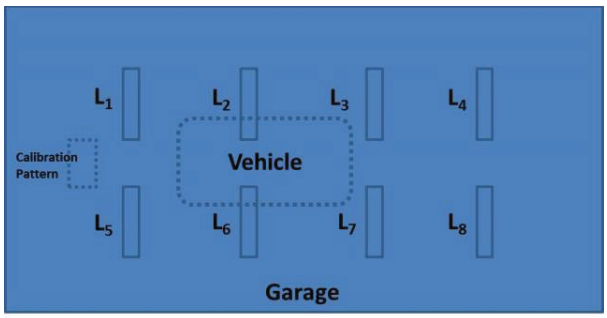
Methods	Movement (mm)
Laser based collinear method	194.509 ± 11.012
Mirror based method	191.266 ± 23.241
Ground truth	199 ± 2

Robustness under different lighting conditions

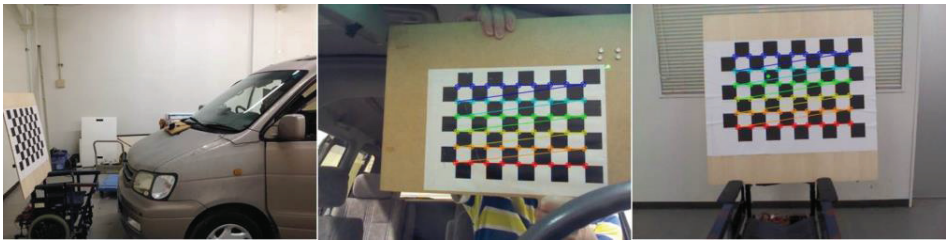
The performance of vision systems is usually dependent on the lighting conditions, so we also carried out experiments on the calibration of our in-vehicle camera system to elucidate this dependence. The input data in our method are obtained from two features: the corner points of the calibration patterns and the extracted laser spots. Therefore, we evaluate the robustness of the technique with respect to these two features.

Example experimental environments and corresponding results are shown in Fig. 6.6. Eight fluorescent lamps (L_1–L_8) were distributed around the garage as in Fig. 6.6 (a). There were five different lighting conditions: all eight fluorescent lamps in the room were turned on (Fig. 6.6 (b)); four fluorescent lamps (L_3, L_4, L_7, L_8) were turned on (Fig. 6.6 (c)); only two fluorescent lamps (L_3, L_4) were turned on (Fig. 6.6 (d)); no fluorescent lamps were turned on but the vehicle's head lamps and interior lights were used (Fig. 6.6 (e)); and finally outdoor daytime conditions (Fig. 6.6 (f)).

The results show that our method is robust in the lighting conditions of Figs. 6.6 (b), (c), (d), (e), and (f) (both corner points and laser spots can be recognized). Even in darkness, calibration was possible using the vehicle's head lamps and interior lights. Therefore, the proposed method is robust to a variety of lighting conditions, making it



(a) The lamps's distribution in the garage (topview).

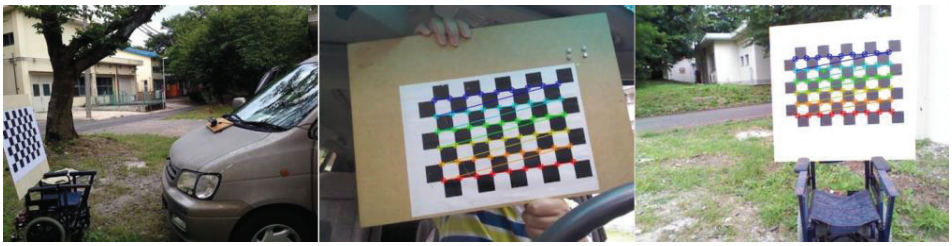


(b) All the eight fluorescent lamps in the room were turned on (indoor).

(c) Four fluorescent lamps (L_3, L_4, L_7, L_8) in the room were turned on (indoor).

(d) Only two fluorescent lamps (L_3, L_4) in the room were turned on (indoor).

(e) No fluorescent lamps but the vehicle head lamps and the lamps inside vehicle were used (indoor).



(f) Outdoor and daytime.

Figure 6.6: Experiments in five different lighting conditions: environment (left), pattern inside vehicle (middle), pattern outside vehicle (right).

suitable for use in real environments such as garages.

6.3 Conclusion and contribution

In this Chapter, a novel approach was proposed for the calibration of cameras with non-overlapping fields of view using two calibration patterns and a laser pointer

mounted on a calibration pattern. A collinear constraint algorithm, were presented and tested with synthetic data and for an in-vehicle camera system. In comparison with the conventional mirror-based method [15], the collinear constraint algorithm achieved higher accuracy in terms of the estimation of the displacement between the two cameras. The technique was also found to be robust in a range of different lighting conditions. The proposed laser-based approach is simple and practical, and especially well suited to the calibration of the relative pose between inward and outward-facing in-vehicle cameras in a factory or garage.

In future work we will carry out further real-world experiments with ground truth. In-vehicle camera systems using calibration techniques such as ours will undoubtedly be used to monitor driver behavior in the future.

CHAPTER 7

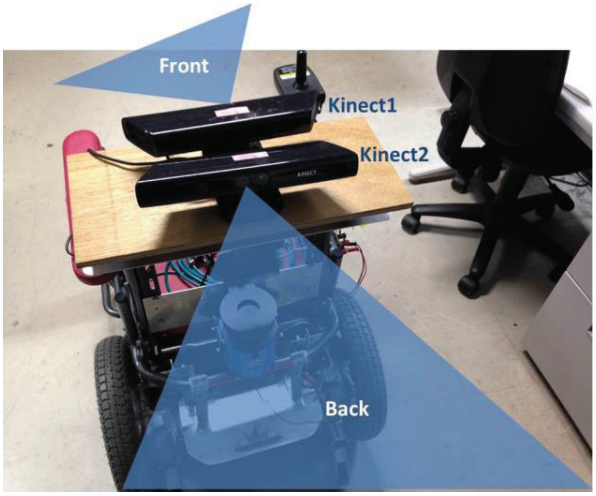
Calibrating Non-overlapping RGB-D Cameras with One Pattern

In this chapter, a novel method for calibrating non-overlapping RGB-D cameras is proposed using one pattern and a laser pointer. Compared to the two algorithms mentioned previously (in Chapter 5 and Chapter 6), the depth cues have been exploited. Two 3D points, respectively located in the field of views of the two RGB-D cameras, are connected by a laser ray. The relative pose of the two RGB-D cameras can be estimated through this collinear constraint.

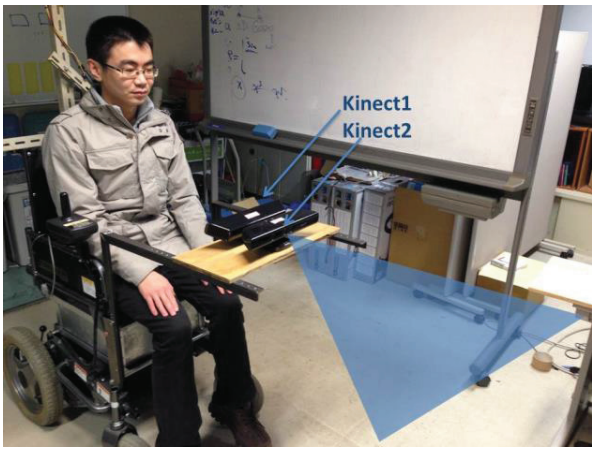
This chapter is organized as follows. Firstly we give an introduction to the background and the related work. And the fundamentals about camera model and calibration of Kinect will be introduced beforehand. Then the key principle of our novel calibration method of non-overlapping Kinects is drawn. Next, the results and performance of this method are shown on both synthetic data and real-world experiment. Finally, come to a conclusion.

7.1 Introduction

As a low-cost RGB-D camera, Kinect, has become a hot topic. It has been applied in action recognition [41], SLAM (Simultaneous Localization and Mapping) [42], robot navigation [43], scene understanding [44] and so on. In some cases, multiple Kinects are used in a system, and we need to calibrate their relative pose.



(a) Robot navigation



(b) Driving assistance

Figure 7.1: Systems with two non-overlapping Kinects

Fig. 7.1 (a) shows a mobile robot mounted with two Kinects. The two Kinects monitor front and back respectively, so that the robot can go forward or backward anytime. To reconstruct the environment in a unified coordinate system, the relative pose of two Kinects should be calibrated. Also, when use two Kinects in the motor chair for the disabled to construct a driving assistance system, one Kinect analyses the front scene

and the other Kinect monitors the user's behavior, as shown in Fig. 7.1 (b). When we want to capture the user's attention, we need not only to measure the gaze direction of the user, but also to map the gaze direction estimated from the user-looking Kinect to the coordinate system of the scene-looking Kinect so that the scene that the user is observing can be determined. Note that in the two cases above, both the two Kinects have not overlapping views.

Until now, several methods of calibrating non-overlapping RGB cameras have been reported. However, if the methods for RGB cameras are used directly to RGB-D cameras, the depth cues of RGB-D cameras is ignored without being exploited for the pose calibration of RGB-D cameras.

This Chapter copes with the problem of calibrating non-overlapping RGB-D cameras, such as Kinects, by exploiting the depth cues. A laser pointer is fixed at one calibration board so that its pose at the coordinate system of the calibration board can be obtained easily, as shown in Fig. 1.2 (a). While one of the RGB-D cameras observes the calibration board fixed with the laser pointer, the laser pointer project a spot to the scene which is observed by the other. Thus, two 3D points, respectively located in the field of views of the two RGB-D cameras, are connected by a laser ray. The relative pose of two RGB-D cameras can be estimated through this collinear constraint.

7.2 Related works

A Kinect consists of a RGB camera and a depth camera. To use a Kinect, we first need to calibrate its parameters, which include the intrinsic parameters of both RGB

camera and the depth camera, and the relative pose between them. For a single Kinect, many researches on the calibration are reported [45] [46].

If we use multiple Kinects simultaneously, we also need to calibrate the relative pose among them. Since the depth camera of a Kinect uses a structure light, the light pattern of the different cameras can interfere with one another causing incorrect depth measurements for the commonly visible areas. Besides the applications shown in Fig. 7.1, a simple method of setting up multiple Kinects is not to allow them with overlapping views. In these cases, we need to calibrate the relative pose of non-overlapping Kinects.

A straight method is first to determine the Kinects' relative pose via their RGB cameras. Several research on the calibration of non-overlapping RGB cameras have been reported, such as using a mirror to allow the different RGB cameras observing the same calibration pattern [11] [16]. However, it's often difficult to change the orientation of the pattern from one snapshot to another when calibrating using a mirror. Moreover, it's not easy to make both pattern and virtual image of pattern respectively seen by two cameras pointing to the opposite directions. Especially, for cameras with large volume, such as Kinect, the occlusion of the calibration pattern is difficult to avoid. Above all, this approach of calibrating Kinects' relative pose via their RGB cameras ignores the depth cues of the Kinects. That is, the depth cues are not exploited for the pose calibration along with their RGB cues.

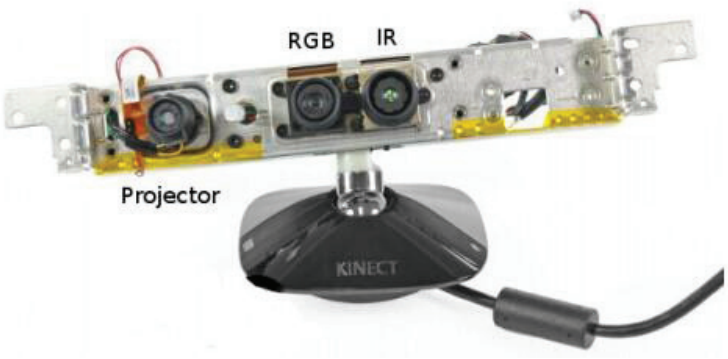


Figure 7.2: Kinect consists of Infrared (IR) projector, IR camera and RGB camera (illustration from [52]).

7.3 Introduction to Kinect

Kinect (see Fig. 7.2) is a composite device consisting of an IR projector of a pattern and IR camera, which are used to triangulate points in space. It works as a depth camera, and a color (RGB) camera, which can be used to recognize image content and texture 3D points. As a measuring device, Kinect delivers three outputs: IR image, RGB image, and (inverse) Depth image.

7.3.1 IR image

IR (1280x1024 pixels for 57x45 degrees FOV, 6.1 mm focal length, 5.2 um pixel size) camera is used to observe and decode the IR projection pattern to triangulate 3D scene. According to [45], if suitably illuminated by a halogen lamp [53, 55] and with the IR projector blocked, it can be reliably calibrated by [49] using the same calibration pattern used for the RGB camera. However, in this thesis, we calibrate the IR camera

without a halogen lamp. We use a tissue to block the IR projector. Then a smoother IR image can be captured and used to extract the corner points of the same calibration pattern. The camera exhibits non-negligible radial and tangential distortions.

7.3.2 RGB image

RGB (1280 x 1024 pixels for 63x50 degrees FOV, 2.9 mm focal length, 2.8 um pixel size) camera delivers medium quality images. It can be calibrated by [49] and used to track the camera motion by SfM system, e.g. [54, 52].

7.3.3 Depth image

The main raw output of Kinect is an image that corresponds to the depth in the scene. Rather than providing the actual depth z, Kinect returns "inverse depth" d. Taking into account the depth resolution achievable with Kinect, we adopted the model Equation 5 suggested in [50]. The Depth image is constructed by triangulation from the IR image and the projector and hence it is "carried" by the IR image, Equation 5.

7.3.4 Depth resolution

The resolution in depth is a function of the distance. The size of the quantization step q, which is the distance between the two consecutive recorded values, is the following function of the depth z

$$q(z) = 2.73z^2 + 0.74z - 0.58[mm]. \qquad (7.1)$$

With z in meters. According to the experiments of [45], the values of q at the beginning, resp. at the end, of the operational range were $q(0.50m) = 0.65mm$, resp. $q(15.7m) = 685mm$.

7.3.5 Shift between IR image and depth image

IR and Depth images are found to be shifted. That's because the Depth image is generated from the IR image through a convolution with a window of certain size. To determine the shift $[u_0, v_0]^T$, [45] captured several different targets in IR and Depth images. Then the contrast target was segmented out from the background and the shifts were determined bringing the segmented shapes in the best alignment. Finally the shift was estimated as the mean over all experiments. Their result suggests using a correlation window of size 7x7 pixels in the depth calculation process. And a 9x9 window size was estimated in [52].

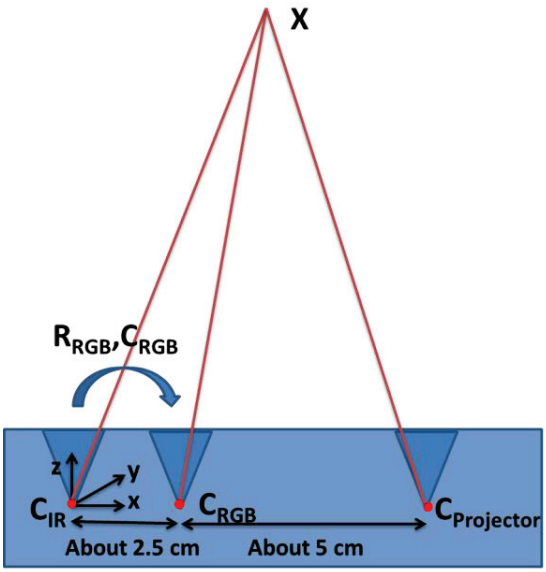


Figure 7.3: Geometrical model of Kinect.

7.4 Geometrical model of Kinect

In this thesis, we model Kinect as a multi-view system consisting of RGB, IR and Depth cameras. Geometrical model of RGB and IR cameras, which project a 3D point X into image point $[u,v]^T$, is given by [49]

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} s \\ t \\ 1 \end{bmatrix} \qquad (7.2)$$

$$\begin{bmatrix} s \\ t \\ 1 \end{bmatrix} = \left(1 + k_1 r^2 + k_2 r^4 + k_5 r^6\right) \begin{bmatrix} p \\ q \\ 0 \end{bmatrix} + \begin{bmatrix} 2k_3 pq + k_4(r^2 + 2p^2) \\ 2k_4 pq + k_3(r^2 + 2q^2) \\ 1 \end{bmatrix} \qquad (7.3)$$

$$r^2 = p^2 + q^2, \qquad \begin{bmatrix} pz \\ qz \\ z \end{bmatrix} = \mathbf{R}(X - C) \qquad (7.4)$$

With distortion parameters $k = [k_1, k_2, \cdots, k_5]$, camera intrinsic matrix $\mathbf{M}$, rotation $\mathbf{R}$ and camera center C [5].

The Depth camera of Kinect is associated to the geometry of the IR camera. It returns the inverse depth d along the z axis, Fig. 7.3, for every pixel $[u,v]^T$ of the IR cameras as

$$\begin{bmatrix} x \\ y \\ d \end{bmatrix} = \begin{bmatrix} u - u_0 \\ v - v_0 \\ \frac{1}{c_1}\frac{1}{z} - \frac{c_0}{c_1} \end{bmatrix} \qquad (7.5)$$

Where u,v are given by Equation (7.3), true depth z by Equation (7.4). $[u_0, v_0]^T$ is set to be $[3,3]^T$ according to experiments. X stands for 3D coordinates of a

3D point, and c_1, c_0 are parameters of the model. We associate the Kinect coordinate system with the IR camera and hence get $\mathbf{R}_{IR} = \mathbf{I}$ and $C_{IR} = \mathbf{0}$. A 3D point X_{IR} is constructed from the measurement $[x, y, d]$ in the depth image by

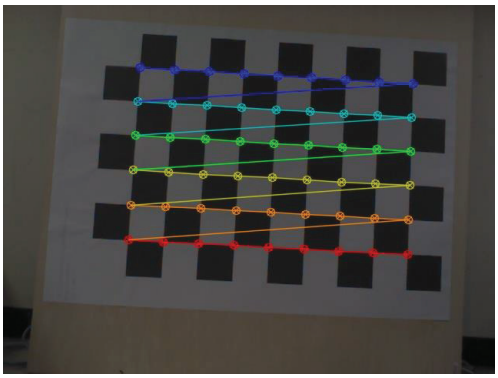
$$X_{IR} = \frac{1}{c_1 d + c_0} dis^{-1}\left(\mathbf{M}_{IR}^{-1} \begin{bmatrix} x + u_0 \\ y + v_0 \\ 1 \end{bmatrix}, k_{IR} \right) \tag{7.6}$$

And projected to the RGB images as

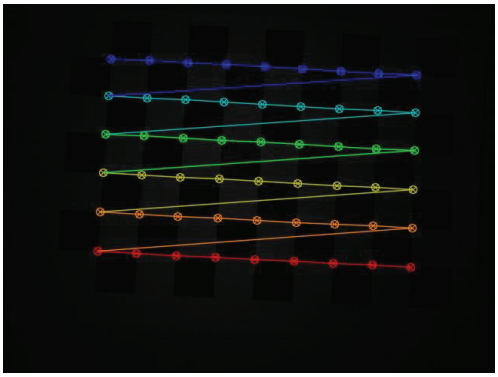
$$u_{RGB} = \mathbf{M}_{RGB} dis(\mathbf{R}_{RGB}(X_{IR} - C_{RGB}), k_{RGB}) \tag{7.7}$$

Where dis is the distortion function given by Equation (7.3), k_{IR}, k_{RGB} are respective distortion parameters of the IR and RGB cameras, $\mathbf{M}_{IR}$ is the IR camera intrinsic matrix and $\mathbf{M}_{RGB}, \mathbf{R}_{RGB}, C_{RGB}$ are the intrinsic matrix, the rotation matrix and the center, of the RGB camera, respectively.

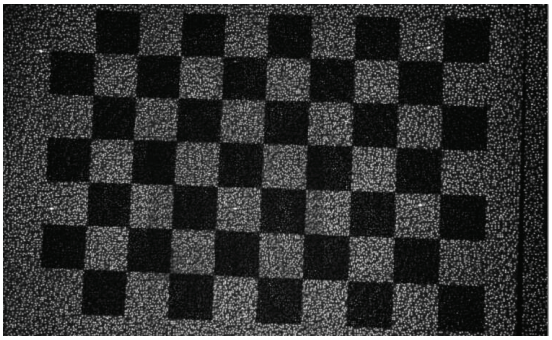
7.5 Calibrating one Kinect



(a) RGB image



(b) IR image (with IR camera blocked)



(c) IR image (with IR camera unblocked)

Figure 7.4: RGB and IR images of a calibration pattern

As our purpose is to calibrate a camera system consisting of non-overlapping

Table 7.1: Intrinsic parameters for Kinect1

IR camera	Focal length		Principal point	
	fx (pixel)	f_y (pixel)	x_0 (pixel)	y_0 (pixel)
	588.26	588.26	319.84	231.88
	Distortion coefficient			
	k_1	k_2	k_3	k_3
	-0.0916	0.2065	0.0011	0.0006
RGB camera	Focal length		Principal point	
	f_x (pixel)	f_y (pixel)	x_0 (pixel)	y_0 (pixel)
	519.73	519.33	326.18	240.55
	Distortion coefficient			
	k_1	k_2	k_3	k_3
	0.0290	-0.1198	0.0011	-0.0022

Kinects, we firstly calibrate single Kinect, including intrinsic parameters of IR and RGB cameras and relative pose between IR and RGB cameras. Fig. 7.4 (c) shows the IR image of the calibration board under the normal Kinect operation when it is illuminated by its IR projector. In the experiment, we found it is difficult to recognize all the corner points of the calibration board in such an IR image. Here, we used a little trick. A piece of tissue blocks in front of the IR camera, then a smoother IR image of the calibration board can be captured (see Fig. 7.4 (b)). And the corner points now can be detected robustly. We calibrate IR and RGB camera of a single Kinect by showing the same calibration target to them (Fig. 7.4). In this way, both cameras are calibrated w.r.t. the same 3D points and

Table 7.2: Intrinsic parameters for Kinect2

IR camera	Focal length		Principal point	
	fx (pixel)	f_y (pixel)	x_0 (pixel)	y_0 (pixel)
	585.68	583.66	323.26	241.49
	Distortion coefficient			
	k_1	k_2	k_3	k_3
	-0.0825	0.2008	0.0035	0.0006
RGB camera	Focal length		Principal point	
	f_x (pixel)	f_y (pixel)	x_0 (pixel)	y_0 (pixel)
	533.43	532.23	321.96	240.17
	Distortion coefficient			
	k_1	k_2	k_3	k_3
	0.0294	-0.0731	0.0022	-0.0003

hence the poses of the cameras w.r.t. calibration pattern can be chained to give their relative pose. Table 7.1, 7.2 shows intrinsic parameters of both Kinect1 and Kinect2

Parameters c0, c1 of the Depth camera are calibrated as follows. We get n measurements $X_{D_i} = [x_i, y_i, d_i]^T, i = 1, \cdots, n,$ of all the calibration points from Depth images. Moreover, in order to improve robustness against distance of object, the calibration pattern is moved from 0.5m to 1.0 m. For example, if we would like to use 10 Depth image, the calibration board will be placed in 10 different positions distribute from 0.5 m to 1.0 m. Note that, this distance range is chosen because in the later non-

overlapping calibration the distance from scene to Kinect will also be 0.5~1.0 m. The Cartesian coordinates X_{IR_i} of the same calibration points were measured in the IR Cartesian system by intersecting the rays projecting the points into IR images with the best plane fits to the reconstructed calibration points. Parameters c_0, c_1 were optimized to best fit X_{D_i} to X_{IR_i} using Equation (7.6).

7.6 3D point reconstruction from its 2D RGB image projection

A sketch for estimating the depth of laser spot is shown in Fig. 7.5. By taking RGB2 and IR2 as a stereo system, an epipolar line in IR image can be calculated according to the known laser spot q_{RGB2} and the calibrated fundamental matrix $\mathbf{F}_{RGB2-IR2}$. Assume there are N points ${q_{IR2}}^i (i \in [1, N])$ in the epipolar line. Corresponding IR image and Depth image are found to be shifted by $[u_0, v_0]$. Depth camera of Kinect2 is associated to the geometry of the IR2 [45]. For every point ${q_{IR2}}^i$, corresponding depth is $d_i = d\left(x_{{q_{IR2}}^i} - u_0, y_{{q_{IR2}}^i} - v_0\right)$. The corresponding 3D point is:

$$ {Q_{IR2}}^i = \frac{1}{c_1 d_i + c_0} \cdot {\mathbf{M}_{IR2}}^{-1} \cdot \left(dis^{-1}\left(\begin{bmatrix} {x_{qIR2}}^i \\ {y_{qIR2}}^i \\ 1 \end{bmatrix}, D_{IR2} \right) \right) \tag{7.8} $$

Where dis is the distortion processing according to the distortion parameters D_{IR} of IR2. $\mathbf{M}_{IR}$ is the projection matrix of IR2.

And the corresponding 3D point in the coordinate system of RGB camera is:

$$Q_{RGB2}{}^{i} = Q_{IR2}{}^{i} \cdot \mathbf{R}_2 + \mathbf{t}_2 \qquad (7.9)$$

where $(\mathbf{R}_2, \mathbf{t}_2)$ is the relative pose between RGB2 and IR2.

3D point $Q_{RGB2}{}^{i}$ can be projected to the RGB image as:

$$q_{RGB2}{}^{i} = dis(\mathbf{M}_{RGB2} \cdot Q_{RGB2}{}^{i}, D_{RGB2}) \qquad (7.10)$$

We search on the epipolar line, and choose the point $q_{IR2}{}^{j}$, whose corresponding point $q_{RGB2}{}^{j}$ is nearest to the observed laser spot q_{RGB2}.

$$q_{IR2}{}^{j} = \left(q_{IR2}{}^{i}\right)_{\min\left|q_{RGB2} - q_{RGB2}{}^{i}\right|} \qquad (7.11)$$

Where, the value $pixel\_error = \left|q_{RGB2} - q_{RGB2}{}^{j}\right|$ is thought to be the pixel error of point $q_{RGB2}{}^{j}$ found through this method.

By equations (7.8) (7.9) (7.10) (7.11), we can determine the point $q_{IR2}{}^{j}$, which is approximately thought to be the projection of the same laser point. Next we take the depth z^j of point $Q_{RGB2}{}^{j}$ as the depth of point q_{RGB2}. The average of $pixel\_error$ is about 0.4 pixel in our experiment. Therefore the error of depth determined in this way is acceptable.

As the depth of the laser spot Q in the coordinate of RGB2 (coordinate of Kinect2) has been determined, we can calculate its 3D coordinate:

$$Q_{RGB2} = z^j \cdot \mathbf{M}_{RGB2}{}^{-1} \cdot \left(dis^{-1}\left(\begin{bmatrix} x_{qRGB2} \\ y_{qRGB2} \\ 1 \end{bmatrix}, D_{RGB2} \right) \right) \qquad (7.12)$$

7.7 Calibrating two Kinects with non-overlapping views

7.7.1 Problem definition

A sketch is shown in Fig. 7.5 to give the problem definition. The two non-overlapping Kinects to be calibrated are denoted as Kinect1 and Kinect2. Kinect1 has two cameras, RGB1 and IR1, and three images, RGB image1, IR image1 and Depth image1. Kinect2 also has two cameras, RGB2 and IR2, and three images, RGB image2, IR

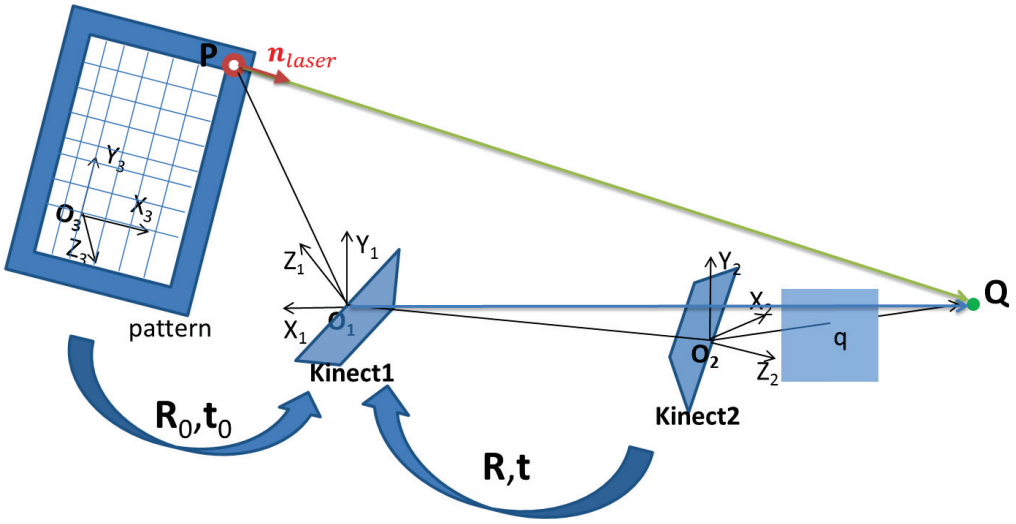


Figure 7.5: A sketch for calibrating non-overlapping Kinects with laser pointer

image2 and Depth image2. Note that, internal parameters of every Kinect are calibrated previously, using the method proposed by Jan Smisek [45]. Here, internal parameters of one Kinect consist of intrinsic matrix and distortion parameters of IR and RGB cameras, parameters of Kinect's depth model and relative pose between IR camera and RGB

camera of the same Kinect. What we want to do is to estimate the external parameters (relative pose), $(\mathbf{R},\mathbf{t})$, between the two Kinects.

We define the camera coordinate system of RGB1 as the coordinate system of Kinect1 and define the camera coordinate system of RGB2 as the coordinate system of Kinect2. Assume that there is a ray of light, $\overrightarrow{PQ}$, starting from point, P, and intersecting with scene at point , Q ; the points P and Q, can be observed by Kinect1 and Kinect2, respectively. Note that the starting point P and direction vector of the ray, $\mathbf{n}_{laser}$, represented at the coordinate system of the calibration board, $O_3 - X_3Y_3Z_3$, can be represented at the coordinate system of Kinect1, $O_1 - X_1Y_1Z_1$, via the transformation $(\mathbf{R}_0,\mathbf{t}_0)$. According to the depth cues, the length of $\overrightarrow{O_2Q}$ along with its direction can be obtained at the coordinate system of Kinect2, $O_2 - X_2Y_2Z_2$. Then, the coordinate of point Q at the coordinate system of Kinect1, $O_1 - X_1Y_1Z_1$, can be computed via the transformation $(\mathbf{R},\mathbf{t})$. Since the length of $\overrightarrow{O_1P}$ along with its direction can also be obtained at the coordinate system of Kinect1, $O_1 - X_1Y_1Z_1$, vector $\overrightarrow{PQ}$ is determined. Since $\overrightarrow{PQ}$ and laser ray $\mathbf{n}_{laser}$ exist at the same line, the *collinear* constrain of $\overrightarrow{PQ}$ and $\mathbf{n}_{laser}$ is obtained.

Note that if the depth cue of point Q is unknown, only the direction of $\overrightarrow{O_2Q}$ can be determined. By computing the coordinate of point P at the coordinate system of Kinect2, $O_2 - X_2Y_2Z_2$, via the transformation $(\mathbf{R},\mathbf{t})$, the *coplanar* constraint for $\mathbf{n}_{laser}$, $\overrightarrow{O_2P}$, and $\overrightarrow{O_2Q}$ is obtained.

Obviously, a collinear constraint is much more rigid than a coplanar constraint. Therefore, it is expected that more accurate result is obtained by exploiting the Kinects' depth cues. The expectation is verified by the experiment results later. This also discriminates the proposed method from the non-overlapping RGB camera calibration methods in essence.

7.7.2 Process

Next, we explain how to estimate the parameters, including the start point P, direction vector of the ray $\mathbf{n}_{laser}$ and the 3D point Q, and how to establish an equation according to the constraint, in detail. A block diagram of our calibration method is presented in Fig. 7.6.

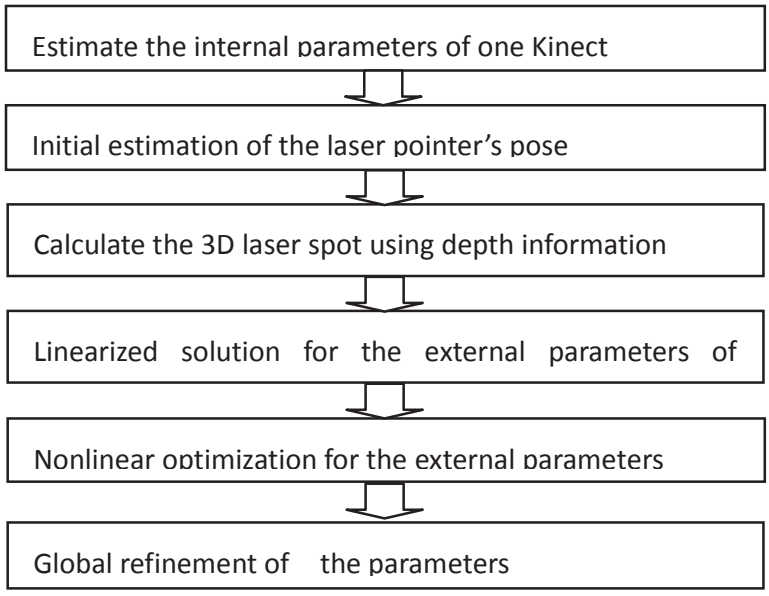


Figure 7.6: A sketch for calibrating non-overlapping Kinects with laser pointer

7.7.3 Linear estimation of relative pose

The collinear constraint for the vector $\overrightarrow{PQ}$ and the vector $\mathbf{n}_{laser}$ can be written in the form of Equation (7.13): (at the coordinate system of Kinect1)

$$\overrightarrow{PQ}^{1} = \gamma \cdot \mathbf{n}_{laser}{}^{1} \tag{7.13}$$

Here, the coefficient γ is an arbitrary constant. As shown in Fig. 7.5:

$$\overrightarrow{PQ}^{1} = Q^{1} - P^{1} = \mathbf{R} \cdot Q^{2} + \mathbf{t} - P^{1} \tag{7.14}$$

$$\mathbf{n}_{laser}{}^{1} = \mathbf{R}_{0} \cdot \mathbf{n}_{laser}{}^{3} \tag{7.15}$$

$$P^{1} = \mathbf{R}_{0} \cdot P^{3} + \mathbf{t}_{0} \tag{7.16}$$

If we define the vectors with x, y and z coordinate.

$$\overrightarrow{PQ}^{1} = \left[PQ_{x}{}^{1}, PQ_{y}{}^{1}, PQ_{z}{}^{1}\right]^{T}, \mathbf{n}_{laser}{}^{1} = \left[n_{x}{}^{1}, n_{y}{}^{1}, n_{z}{}^{1}\right]^{T}$$

Then we get these relationships from Equation (7.13):

$$\begin{aligned} \frac{PQ_{x}{}^{1}}{PQ_{y}{}^{1}} &= \frac{n_{x}{}^{1}}{n_{y}{}^{1}} \\ \frac{PQ_{x}{}^{1}}{PQ_{z}{}^{1}} &= \frac{n_{x}{}^{1}}{n_{z}{}^{1}} \end{aligned} \tag{7.17}$$

Define,

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}; \ \mathbf{t} = \left[t_{x}, t_{y}, t_{z}\right]^{T};$$

$$Q^{2} = \left[Q_{x}{}^{2}, Q_{y}{}^{2}, Q_{z}{}^{2}\right]^{T}; \ P^{1} = \left[P_{x}{}^{1}, P_{y}{}^{1}, P_{z}{}^{1}\right]^{T}.$$

According to Equation (7.14), Equation (7.17) can be rewritten as:

$$
\begin{aligned}
&(r_{11}\cdot {Q_x}^2 + r_{12}\cdot {Q_y}^2 + r_{13}\cdot {Q_z}^2 + t_x - {P_x}^1) - \\
&\frac{{n_x}^1}{{n_y}^1}\cdot(r_{21}\cdot {Q_x}^2 + r_{22}\cdot {Q_y}^2 + r_{23}\cdot {Q_z}^2 + t_y - {P_y}^1) = 0 \\
&(r_{11}\cdot {Q_x}^2 + r_{12}\cdot {Q_y}^2 + r_{13}\cdot {Q_z}^2 + t_x - {P_x}^1) - \\
&\frac{{n_x}^1}{{n_z}^1}\cdot(r_{31}\cdot {Q_x}^2 + r_{32}\cdot {Q_y}^2 + r_{33}\cdot {Q_z}^2 + t_z - {P_z}^1) = 0
\end{aligned}
\tag{7.18}
$$

The relative pose $(\mathbf{R}_0, \mathbf{t}_0)$ between Kinect1 and chessboard can be easily calculated using the image of the pattern [7, 12]. The laser pointer's pose $\left(P^3, {\mathbf{n}_{laser}}^3\right)$ has been estimated in Chapter 4. And 3D point Q^2 has been calculated in Section 7.6. So when a group of images (RGB1 image, IR2 image, RGB2 image and Depth2 image) are captured by Kinect1 and Kinect2, one group of data $\left({\mathbf{n}_{laser}}^1, P^1, Q^2\right)$ can be calculated using Equations (7.12) (7.15) (7.16). In Equation (7.18), only the relative pose $(\mathbf{R}, \mathbf{t})$ is unknown.

Equation (7.18) is basic linear equation for relative pose of two Kinects. During the calibration process, we change the pose of pattern to get a series of data, $\left({\mathbf{n}_{laser}}^1, P^1, Q^2\right)_{(i)}$. At least 3 groups of data are required for a direct solution according to D. Nister [47]. During the calibration we typically obtain the initial solution through an over-constrained linear system of equations.

7.7.4 Nonlinear estimation of relative pose

During the linearity estimation of relative pose, we didn't consider the constraint of rotation matrix strictly. Next, we consider the collinear constraint together with the

constraint of rotation matrix and represent the rotation matrix $\mathbf{R}$ with a quaternion. $\mathbf{R} = (a; b, c, d)$.

These constraint conditions can be obtained:

$$\begin{cases} \overrightarrow{PQ}^1 \bullet \mathbf{n}_{laser}{}^1 = \left|\overrightarrow{PQ}^1\right| \\ a^2 + b^2 + c^2 + d^2 = 1 \end{cases} \tag{7.19}$$

Considering equation (7.14), an error function can be derived from equation (7.19):

$$f(\mathbf{R},\mathbf{t}) = \sum_{i=1}^{N} \begin{pmatrix} \left(\left(\mathbf{R}\cdot Q^2 + \mathbf{t} - P^1\right)\bullet \mathbf{n}_{laser}{}^1 - \left|\mathbf{R}\cdot Q^2 + \mathbf{t} - P^1\right|\right)^2 \\ + \lambda\cdot\left(a^2 + b^2 + c^2 + d^2 - 1\right)^2 \end{pmatrix} \tag{7.20}$$

Here, N is the total number of groups of observations. As described in Section 7.7.3, by changing the chessboard's position and orientation, we obtain a series of data $\left(\mathbf{n}_{laser}{}^1, P^1, Q^2\right)_{(i)}$. Only relative pose $(\mathbf{R},\mathbf{t})$ is unknown in equation (7.20).

Once the initial $(\mathbf{R}', \mathbf{t}')$ has been found via the linearity estimation mentioned in Section 7.7.3, nonlinear optimization can be used to estimate the relative pose. The error function is minimized using Levenberg-Marquardt algorithm.

7.7.5 Parameter refinement

As explained in Chapter 4, in the process of determining the pose (P^{board}, $\mathbf{n}^{board}$) of the laser pointer, the manual operation may give rise to a measurement error. Finally, we refine the pose of the laser pointer and the relative pose between the two Kinects, ($\mathbf{R}$, $\mathbf{t}$, P^{board}, $\mathbf{n}^{board}$), taking the calibration result estimated in Chapter 4 and Section 7.7.4 as

initial values. Concretely, we minimize the following objective function by the Levenberg-Marquardt method.

$$f\left(\mathbf{R},\mathbf{t},P^{board},\mathbf{n}_{laser}{}^{board}\right)= \sum_{i=1}^{N}\left(\begin{array}{l}\left(\left(\mathbf{R}\cdot Q^2+\mathbf{t}-\left(\mathbf{R}_1\cdot P^{board}+\mathbf{t}_1\right)\right)\bullet\left(\mathbf{R}_1\cdot\mathbf{n}_{laser}{}^{board}\right)\right)^2 \\ -\left|\mathbf{R}\cdot Q^2+\mathbf{t}-\left(\mathbf{R}_1\cdot P^{board}+\mathbf{t}_1\right)\right| \\ +\lambda\cdot\left(a^2+b^2+c^2+d^2-1\right)^2\end{array}\right) \tag{7.21}$$

7.8 Experiments and results

In this section, using synthetic data, we first compare our collinear-constraint-based method with the coplanar-constraint-based method and mirror-based method, which ignore the depth cues. Then, we apply our collinear-constraint-based method in a real-world Kinects system. Note that the coplanar-constraint-based method and mirror-based method correspond to RGB cameras mentioned in the problem definition of Section7.7.1, which means calibrating the relative pose of RGB-D cameras as that of RGB cameras.

7.8.1 Simulations

During the simulation experiment, the Kinect's internal parameters and image resolution are set the same as real ones. A synthetic Pattern1 with 9×6 corners is created (Side length of Pattern1's gridding is 0.026m.). Laser pointer is set to be perpendicular to the Pattern1 and locate nearly at the center of the Pattern1. Concretely, the starting point of laser ray is $(0.117m,0.065m,0m)$ in the coordinate system of the

Pattern1. The extrinsic parameters' ground truth between two Kinects is set as: translation $\mathbf{t} = (0.1m, 0.1m, -0.1m)$ and rotation $\mathbf{R}$ expressed with roll, pitch, yaw $(159.61^{o}, 9.32^{o}, 173.55^{o})$. A scene (wall) is created 0.6 m in front of Kinect2. The Pattern1 is set 0.2m in front of Kinect1. Translation $\mathbf{t}_0$ is fixed as $\mathbf{t}_0 = (-0.104, -0.065, 0.2)m$. During the simulation, the orientation of the Pattern1 relative to Kinect1 keeps changing. Orientation $\mathbf{R}_0$ is created like this: Initially, the Pattern1 is facing directly to Kinect1. Then, an anticlockwise rotational angle $\theta_{p(i)}$ is set around unit vector $\mathbf{n}_{p(i)}$ from the initial orientation. The unit vector $\mathbf{n}_{p(i)}$ is randomly chosen in a unit sphere. Rotational angle $\theta_{p(i)}$ is randomly (uniformly) chosen from $(0.1, \pi/2)$ to make sure the Pattern1 can be observed by Kinect1. For every pose $(\mathbf{R}_0, \mathbf{t}_0)_{(i)}$ of the Pattern1, Pattern1's corners are projected to RGB1 image of Kinect1. The point of intersection of laser ray with the wall is calculated and also projected to RGB2 image of Kinect2. Then, the position values of image points in RGB1 and laser spot in RGB2 are added with a White Gaussian Noise (AWGN) with a standard deviation σ. Finally, the coplanar-constraint-based method and our collinear-constraint-based method are respectively applied on this noisy data to estimate $\mathbf{R}, \mathbf{t}$. Note that, for our collinear constraint method, depth cues can be used to calculate 3D information of laser spot. The depth of laser spot is noised by adding the ground truth with a White Gaussian Noise with a standard deviation $\tau = 1.3mm$. According to the research of J.

Smisek [45], the standard deviation of the residuals of the plane fit to the measurement of a planar target is about 1.3 mm.

To compare with mirror-based-method proposed by R. Rodrigues [15], a synthetic mirror is created in front of Kinect1. A Pattern2 with 9×6 corners is created (Side length of Pattern2's gridding is 0.075m). The Pattern2 is presumed to stick on the wall. The Kinect1 can view the virtual image of Pattern2 via the mirror. The mirror is parameterized with $(\mathbf{n}_m, d_m)$, where d_m is the distance from Kinect1's optical center to mirror plane and $\mathbf{n}_m$ is the surface normal vector of mirror plane at the coordinate system of Kinect1. During the simulation, the pose of mirror relative to Kinect1 keeps changing. $d_{m(i)}$ is randomly (uniformly) chosen from $(0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5)$ m. $\mathbf{n}_{m(i)}$ is generated by anticlockwise rotating vector $[0,0,1]^T$ with angle $\theta_{m(i)}$ around unit vector $\mathbf{n}_{0(i)}$. The unit vector $\mathbf{n}_{0(i)}$ is randomly chosen in a unit sphere. Rotation angle $\theta_{m(i)}$ is randomly (uniformly) chosen from $(0.1, \pi/6)$. Corner points of mirrored Pattern2 are created according to the mirror's pose $(\mathbf{n}_m, \theta_m)_{(i)}$ via a symmetry transformation [15]. Then, corners of mirrored Pattern2 are projected to RGB1 image of Kinect1. The position values of image points of RGB1 Image and RGB2 Image are added with a White Gaussian Noise (AWGN) with a standard deviation σ. Finally, the mirror-based algorithm [15], for which the code

implementation is made publicly available by the authors, is applied on the noisy data to estimate $\mathbf{R}, \mathbf{t}$.

The calibration accuracy is given by the norm $\|dT\|$ of the translation error vector (also $\|dT\|/\|T\|$ expressing as percentage of the magnitude of the camera baseline) and the angular error dR whose expression is given by [13]:

$$dR = d(\hat{\mathbf{R}}, \mathbf{R}^*) = \arccos\left(\frac{trace(\hat{\mathbf{R}}^T \mathbf{R}^*) - 1}{2}\right) \tag{7.22}$$

Where rotation $\mathbf{R}^*$ is the ground truth and $\hat{\mathbf{R}}$ is the estimated rotation. Fig.7.7 shows the performances of the algorithms. Note that, for convenience, the mirror-based algorithm, the laser-based coplanar algorithm and the laser-based collinear algorithm are denoted as Mirror-Method, Laser-Method and Laser-Depth-Method, respectively.

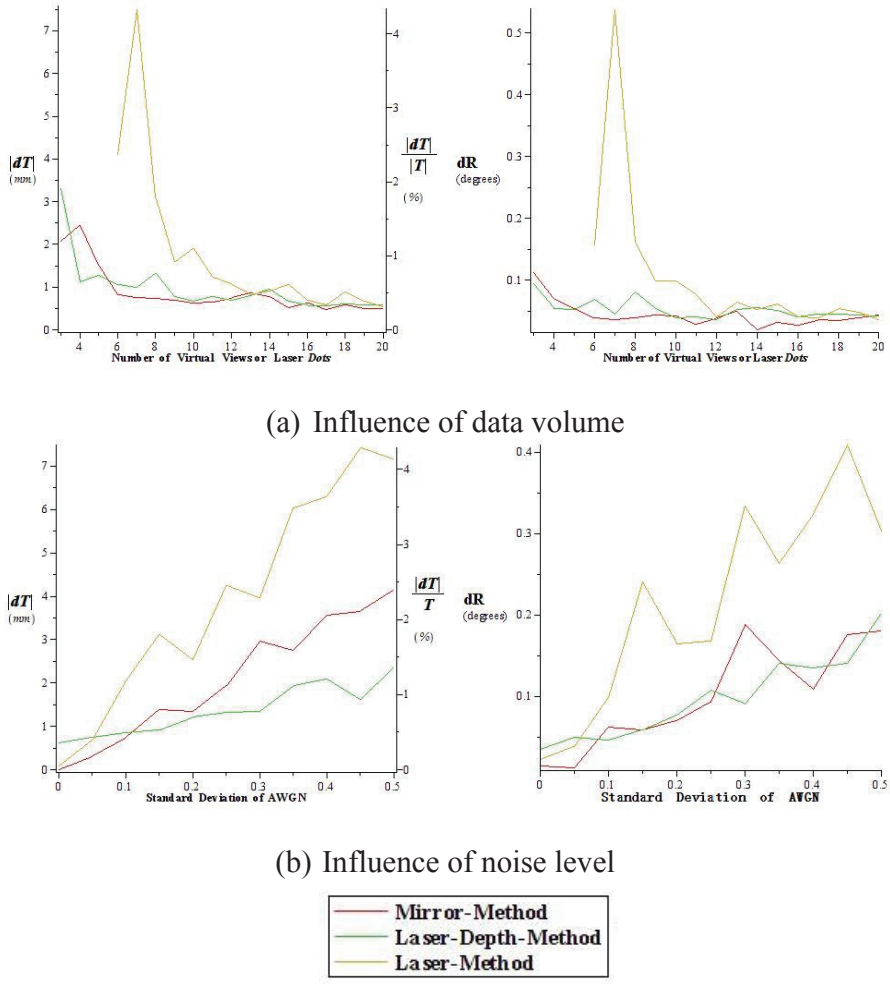


Figure 7.7: Evaluate accuracy against data volume and noise level for Mirror-Method, Laser-Method and Laser-Depth-Method.

To test the sensitivity to the amount of data for different algorithms, the standard deviation of AWGN is fixed as 0.1 pixels [13]. Data volume is changed from 3 to 20 (Virtual patterns' number and laser dots' number). For every data volume, the test is carried out 50 times, and the average accuracy is shown in Fig.7.7 (a). For Mirror-Method and Laser-Depth-Method, the problem can be solved with little data (N>=3), due

to the linearity of constraints. Since the constraint of Laser-Method is nonlinear, more data is necessary (N>=6). As shown in Fig.7.7 (a), the accuracy of Mirror-Method and Laser-Depth-Method is close. Besides, according to the simulation, the accuracy can be improved by increasing data volume. Especially for the Laser-Method, accuracy is influenced significantly by the data volume due to the nonlinearity.

Next, we test the sensitivity to the noise level for the three algorithms. Data volume is fixed to 10. Noise standard deviation σ is changed from 0.0 to 0.5. For every standard deviation, the test is carried out 50 times, and the average accuracy is shown in Fig. 7.7 (b). The Laser-Method is still restrained by data volume. Besides, when noise is small ($\sigma < 0.1$), the Mirror-Method outperforms, because for the Laser-depth-Method a fixed error has been added to the depth of laser spot (White Gaussian Noise with a standard deviation $\tau = 1.3mm$). However, with the increasing of noise ($\sigma > 0.1$), our proposed Laser-Depth-Method outperforms especially in the term of translation accuracy. It comes to a conclusion that, our proposed Laser-Depth-Method is more robust to pixel noise by using depth cues.

7.8.2 Real-world experiments

We applied our method to calibrate relative pose of two Kinects mounted on motor chair as shown in Fig. 7.1. The resolution of Depth, RGB and IR images were set 640×480 . A pattern with 54 corner points was used. The grid side length of pattern is 0.026m. 48 spots are used in every experiment. It's repeated for 10 times. Calibration results are shown in Table 7.3. Since the ground truth of the relative pose between the

Kinects is unknown in the real-world experiments, we evaluate the experimental results based on the statistical indices of relative pose ($\mathbf{R},\mathbf{t}$). The results are consistent with each other. For the about 150mm displacement between the two Kinects, the standard deviation is less than 7mm.

On the other hand, we evaluate the calibration result by calculating the reprojection error. With the calibrated relative pose ($\mathbf{R},\mathbf{t}$), the pose of the laser pointer is determined in the coordinate system of Kinect2. Depth of scene is also determined in Section 7.6. Then the 3D coordinate of laser spot $\hat{Q}$ can be calculated. Comparing to its measured value Q, the 3D distance $d = \left|\hat{Q} - Q\right|$ is defined as reprojection error. As shown in Table 7.3, the average reprojection error is about 1.82 mm.

Moreover, we compare our method with the coplanar constraint method mentioned in the simulation. The same experiment data except depth information are used to calibrate relative pose of the two Kinects using the coplanar constraint. The performance of two methods is compared by reprojection error, as shown in Fig. 7.8. It's clear that the calibration result has been improved by using the depth information in our collinear method. It's also consistent with the conclusion of simulation.

As shown in the simulation, the limitation of our method is requiring high accuracy of depth information. Therefore, the scene used to project the laser spot should be in the range of best depth accuracy (0.5-1.5 m for Kinect) [45, 48]. On the other hand, there may not be corresponding depth information at every pixel where the laser would

Table 7.3: Result of real-world experiments

	Relative Pose							Average Reprojection Error (m)
	Relative position (m)			Relative orientation (Quaternion form)				
Average	0.04159	0.04675	0.14410	0.02078	0.01326	0.99585	0.07756	0.00182
Standard Deviation	0.00608	0.01075	0.00659	0.00388	0.00080	0.00106	0.01343	0.00048

be detected in the 2nd Kinect image. An approximate estimate has been carried out. The error between real point without depth information and approximate pixel with depth information is about 0.4 pixels on average. Therefore, the scene used to project the laser spot had better to be a plan (e.g. wall), so that depth deviation between two points apart with 0.4 pixels can be ignored.

7.9 Conclusion and contribution

In this chapter, we propose a novel method of calibrating non-overlapping RGB-D cameras using one chessboard fixed with a laser pointer by exploiting the depth cues. The relative pose between the RGB-D cameras is estimated based on the collinear constraint. In comparison with the approach of using only RGB cues, which results in the looser coplanar constraint, the proposed method achieves a higher accuracy, as shown in

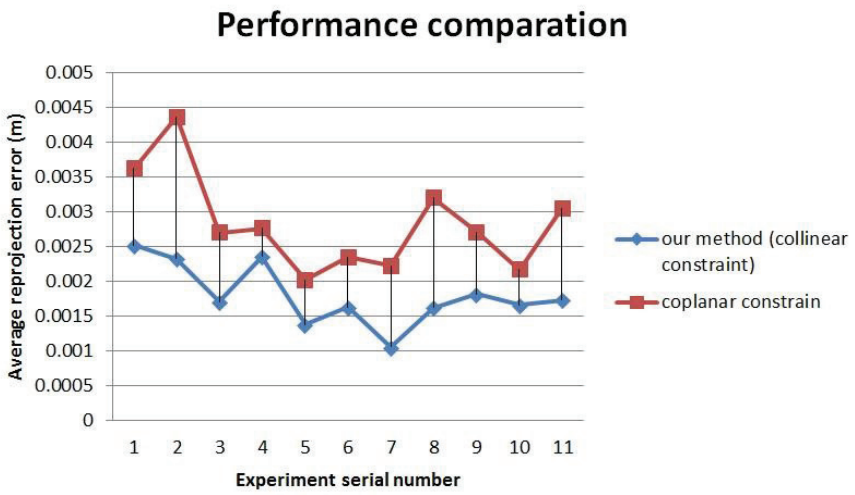


Figure 7.8: Compare two calibration methods

both simulation and real-world experiment results. To the point that depth cues are exploited along with RGB cues, this is the first work on the calibration of non-overlapping RGB-D cameras as far as we know.

The contribution of this paper is as follows:

- A novel method of calibrating two Kinects with non-overlapping view by a laser pointer is proposed. The start point and direction vector of laser ray at the coordinate system of first Kinect is estimated by mount a laser pointer on a chessboard. 3D position of the laser spot located on RGB image of the second Kinect is estimated according to depth cues and epipolar geometry. Thus, a collinear constraint for start point and laser spot can be used to calibrate the relative pose of two non-overlapping Kinects.
- The proposed method is flexible and especially suitable to calibrate two Kinect with non-overlapping. Calibration can be carried out as long as the

laser can get through from one Kinect's field of view to another's. In contrast with the conventional method of using a mirror, in the proposed method humans perform the operation more easily.

- In the calibration process of non-overlapping Kinects, the depth cues are exploited for the pose calibration of Kinects along with the RGB cues. This discriminates the proposed method from the existing non-overlapping RGB camera calibration methods. To this point, this work is the first research on the calibration of a pair of non-overlapping RGB-D cameras as far as we know.

Another feature of the proposed method is that we use laser pointer instead of a mirror; this increases the flexibility of calibration operation work in a real world environment.

CHAPTER 8

Conclusions and Contributions

8.1 Conclusion

In this thesis, we focus on the problem that, when we would like to improve the accuracy of an advanced driving assistance system or apply multi-camera system in robot, it will be necessary to calibrate the relative pose between non-overlapping cameras. The multi-camera system may consist of non-overlapping RGB cameras or RGB-D cameras. Some novel methods have been proposed to solve these problems in this thesis. The contributions are as following.

1. A novel technic is proposed to estimate the pose of a laser pointer, which is fixed on a calibration board. It is based on the principals that, according to laser's unidirectional property and ideal central perspective camera model, laser ray should be pointing to the optical center of camera when it is viewed by the camera. It is a practical and convenient method to calibrate the pose of a laser pointer. Moreover, it is very effective and can be extensive used in many fields such as camera calibration.
2. A novel approach for calibrating the relative pose of two non-overlapping cameras with a laser pointer and one calibration pattern has been proposed. A laser pointer is fixed at a calibration board so that its pose at the coordinate system of the calibration board can be obtained. While one of the cameras observes the calibration board, the intersection of the ray of the laser pointer

with scenes is observed by the other. Thus, the view field of both cameras is connected, and the relative pose between the cameras can be estimated through solving a coplanar constraint. Besides, 3D position of the laser spot located on RGB image of the second Kinect is estimated according to depth cues and epipolar geometry. We show the effectiveness of this method on both synthetic data and an in-vehicle camera system. Compared to the conventional calibration methods of non-overlapping cameras, our proposed method is simple, practical, and especially suitable to the calibration of non-overlapping in-vehicle cameras at a factory or a garage, where the existing methods are difficult to be applied. For a camera network consist of more than two non-overlapping cameras, relative transformation between arbitrary two non-overlapping cameras can be estimated with our proposed method.

3. A novel approach to calibrate inward and outward-facing in-vehicle cameras using a laser pointer and two calibration patterns is proposed. Since inward cameras, which observe the driver's behavior, and outward cameras, which observe the driver's field of view, do not have overlapping views, the conventional calibration methods for stereo cameras cannot be applied directly. A method is presented in which the two cameras can be connected by a laser emitted from a laser pointer. The laser pointer is mounted on a calibration board so that the laser ray’s pose within the calibration board’s coordinate system can be calculated. Two points lying on two calibration

boards respectively can be connected by the laser ray. A collinear method is presented based on this idea. We compare the performance of the proposed collinear algorithm with both coplanar algorithm (proposed in Chapter 5) and the conventional mirror-based method through simulations and experiments. In comparison with the conventional mirror-based method, the collinear constraint algorithm achieved higher accuracy in terms of the estimation of the displacement between the two cameras. The technique was also found to be robust in a range of different lighting conditions. Finally the laser-based collinear method is applied to the non-overlapping calibration of an in-vehicle camera system. In contrast to other methods, the proposed method is simple, practical, and especially well suited to the calibration of the relative pose between inward and outward-facing in-vehicle cameras in a factory or garage.

4. A novel method for calibrating non-overlapping RGB-D cameras is proposed using one calibration pattern and a laser pointer. Compared to the two algorithms previously mentioned, the depth cues have been exploited. Two 3D points, respectively located in the field of views of the two RGB-D cameras, are connected by a laser ray. The relative pose of the two RGB-D cameras can be estimated through this collinear constraint. In comparison with the approach of using only RGB cues, which results in the looser coplanar constraint, the proposed method achieves a higher accuracy, as

shown in both simulation and real-world experiment results. The proposed method is flexible and especially suitable to calibrate two Kinect with non-overlapping. Calibration can be carried out as long as the laser can get through from one Kinect's field of view to another's. In contrast with the conventional method of using a mirror, in the proposed method humans performs the operation more easily. In the calibration process of non-overlapping Kinects, the depth cues are exploited for the pose calibration of Kinects along with the RGB cues. This discriminates the proposed method from the existing non-overlapping RGB camera calibration methods. To this point, this work is the first research on the calibration of a pair of non-overlapping RGB-D cameras as far as we know.

In future work we will carry out further real-world experiments with ground truth. In-vehicle camera systems using calibration techniques such as ours will undoubtedly be used to monitor driver behavior in the future.

Another feature of the proposed method is that we use laser pointer together with a mirror; this will further increases the flexibility of calibration operation work in a real world environment.

8.2 Publications

The content of this thesis is based on the material published in the following papers:

Journals

1) *Wuhe Zou,Shigang Li. Calibration of Non-overlapping In-vehicle Camera with Laser Pointer. IEEE Transactions on Intelligent Transportation Systems, Volume:PP, Issue:99, pp. 1~12, 2014*

2) *鄒武合，李仕剛．レーザポインターを用いた視野の重ならない車載カメラの相対姿勢の算出．IEICE, Vol.J97-D,No.11,pp.-,Nov. 2014.*

International Conferences

1) *Wuhe Zou, Shigang Li. Calibrating non-overlapping RGB-D camera. International Conference on Pattern Recognition (ICPR) (pp.4200-4205) , 2014.08*

2) *Wuhe Zou, Shigang Li. Calibrating relative pose of non-overlapping in-vehicle cameras with laser pointer. 21st World Congress on Intelligent Transport Systems (ITSWC) , 2014.09*

REFERENCES

[1] A. Broggi, M. Bertozzi and A. Fascioli, “Self-calibration of a stereo system for automotive applications”, Proc. of IEEE conference on Robotics and Automation, pp.3698-3703, 2001.

[2] M.M. Trivedi, T. Gandhi and J. McCall. Looking-in and looking-out of a vehicle: computer-vision-based enhanced vehicle safety. IEEE Transactions on Intelligent Transportation Systems, Vol.8, No.1, pp.108-120, 2007.

[3] L. Fletcher, A. Zelinsky, Driver Inattention Detection based on Eye Gaze-Road Event Correlation. Int. J. Robotics Res., 28, pp. 774 – 801, 2009.

[4] S. Li and Y. Hai, Estimating camera pose from H-pattern of parking lot, IEEE International Conference on Robotics and Automation, pp.3954-3959, 2010.

[5] Hartley, A. Zisserman: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge, 2000.

[6] A. Zisserman et. al.: Metric Calibration of a Stereo Rig. Proc. WRVS, 1995.

[7] Z. Zhang. A flexible new technique for camera calibration. IEEE Transcations on Pattern Analysis and Machine Intelli-gence, pages 1330–1334, 2000.

[8] Y. Caspi and M. Irani. Alignment of non-overlapping se-quences. ICCV, Vancouver, BC, pages II: 76–83, 2001.

[9] S. Esquivel, F. Woelk, and R. Koch. Calibration of a multi-camera rig from non-overlapping views. In Lecture Notes in Computer Science 4713 (DAGM), 2007.

[10] S. Sinha, M. Pollefeys, and L. McMillan. Camera network calibration from dynamic silhouettes. IEEE CVPR, 2004.

[11] R. Kumar, A. Ilie, J.-M. Frahm, and M. Pollefeys. Simple calibration of non-overlapping cameras with a mirror. IEEE CVPR, 2008.

[12] Open source computer vision library. http://www.intel.com/research/mrl/research/opencv/.

[13] P. Lébraly, E. Royer, O. Ait-Aider, and M. Dhome. Calibration of Non-overlapping Cameras Application to Vision-Based Robotics. BMVC, page 1-12. British Machine Vision Association, (2010)

[14] J. A. Hesch, A. I. Mourikis, and S. I. Roumeliotis, "Extrinsic camera calibration using multiple reflections," pp. 311–325, 2010.

[15] R. Rodrigues, J. Barreto, and U. Nunes, "Camera Pose Estimation Using Images of Planar Mirror Reflections," Computer Vision–ECCV 2010, pp. 382–395, 2010.

[16] A. Agrawal, "Extrinsic camera calibration without a direct view using spherical mirror", Proc. of International Conference on Computer Vision, pp.2368-2375, 2013.

[17] P. Sturm, S. Ramalingam, J.-P. Tardif, S. Gasparini, and J. Barreto, Camera Models and Fundamental Concepts Used in Geometric Computer Vision. Now Publishers Inc, 2011.

[18] J. Ponce, K. McHenry, T. Papadopoulo, M. Teillaud, and B. Triggs, “On the absolute quadratic complex and its application to autocalibration,” Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, pp. 780–787 vol. 1, Jun. 2005.

[19] R. Pless, “Using many cameras as one,” presented at the Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, 2003, vol. 2.

[20] P. Sturm and T. Bonfort, “How to Compute the Pose of an Object without a Direct View?” Lecture Notes in Computer Science, vol. 3852, p. 21, 2006.

[21] J. Hesch, A. Mourikis, and S. Roumeliotis, “Determining the camera to robot-body transformation from planar mirror reflections,” Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008), Nice, France, 2008.

[22] J. Courchay, A. Dalalyan, and R. Keriven, “A global camera network calibration method with linear programming,” presented at the 3DPVT - Symposium in 3D Data Processing, Visualization and Transmission, 2010.

[23] A. Zaharescu, R. Horaud, R. Ronfard, and L. Lefort, “Multiple Camera Calibration Using Robust Perspective Factorization,” presented at the 3D Data Processing, Visualization, and Transmission, Third International Symposium on, 2006, pp. 504–511.

[24] T. Svoboda, D. Martinec, and T. Pajla, “A convenient multicamera self-calibration for virtual environments,” Presence-Teleoperators and Virtual Environments, vol. 14, no. 4, pp. 407–422, Aug. 2005.

[25] J. Barreto and K. Daniilidis, “Wide area multiple camera calibration and estimation of radial distortion,” presented at the OMNIVIS'2004 - Int. Workshop in Omnidirectional vision, camera networks, and non-conventional cameras, 2004.

[26] M. Agrawal and L. Davis. Complete camera calibration using spheres: Dual space approach. IEEE ICCV, pages 782–789, 2003.

[27] P. T. Baker and Y. Aloimonos. Calibration of a multi-camera network. in r. pless, j. santos-victor, and y. yagi, editors.Om-nivis 2003: Omnidirectional Vision and Camera Networks, 2003.

[28] X. Cao and H. Foroosha. Camera calibration and light source orientation from solar shadows.Computer Vision and Image Understanding, 105:60–72, 2006.

[29] C. Colombo, A. Bimbo, and F. Pernici. Metric 3d reconstruc-tion and texture acquisition of surfaces of revolution from a single uncalibrated view. IEEE Trans. Pattern Anal. Mach. Intell., 1:99–114, 2005.

[30] O. Faugeras and Q. Luong. The geometry of multiple images.MIT Press,Cambridge, 2001.

[31] P. Gurdjos, A. Crouzil, and R. Payrissat. Another way of looking at plane-based calibration: the centre circle con-straint.Proc. ECCV, pages 252–266, 2002.

[32] P. Gurdjos and P. Sturm. Methods and geometry for plane-based self-calibration.IEEE CVPR, pages 491–496, 2003.

[33] G. Jiang, H. Tsui, L. Quan, and A. Zisserman. Single axis geometry by fitting conies.IEEE Trans. Pattern Anal. Mach. Intell., 10, 2003.

[34] I. Kitahara, H. Saito, S. Akimichia, T. Onno, Y. Ohta, and T. Kanade. In Computer Vision and Pattern Recognition, Technical Sketches, 2001.

[35] A. Klaus, J. Bauer, K. Karner, P. Elbischger, R. Perko, and H. Bischof. Camera calibration from a single night sky im-age. IEEE CVPR, pages 151–157, 2004.

[36] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: establishing a common coordinate frame. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000.

[37] D. Liebowitz and A. Zisserman. Combining scene and auto-calibration constraints. IEEE ICCV, pages 293–300, 1999.

[38] E. Malis and R. Cipolla. Camera self-calibration from un-known planar structures enforcing the multi-view constraints between collineations.IEEE Trans. Pattern Anal. Mach. In-tell., 9:1268–1272, 2002.

[39] R.Y.Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. IEEE J. Robotics Autom, 3:323–344, 1987.

[40] H. Zhang, G. Zhang, and K.-Y. Wong. Camera calibration with spheres: linear approaches.IEEE ICIP, 2:1150–1153, 2005.

[41] Y. Ming and Q. Ruan, “Activity recognition form RGB-D camera with 3D local spatio-temporal features”, Proc. Of IEEE International Conference on Multimedia and Expo, pp.344-349, 2012.

[42] C. Kerl, J. Sturm and D. Cremers, “Dense visual SLAM for RGB-D camera”, Proc. Of IEEE International Conference on Intelligent Robots and Systems, pp.2100-2106, 2013.

[43] M. Fiala and A. Ufkes, "Visual Odometry Using 3-Dimensional Video Input," in Computer and Robot Vision (CRV), 2011 Canadian Conference on, 2011, pp. 86-93.

[44] D. Li, S. Fidler and R. Urtasum, “Holistic scene understanding for 3D object detection with RGBD cameraa”, Proc of ICCV, pp.1417-1424, 2013.

[45] J. Smisek , M. Jancosek and T. Pajdla "3-D with Kinect", Proc. IEEE ICCV Workshops, pp.1154 -1160 2011

[46] D. Herrera C., J. Kannala and J. Hekkila, “Joint depth and color camera calibration with distortion correction”, IEEE Transactions on PAMI, Vol.34, No.10, pp.2058-2064, 2012.

[47] D. Nister, "A Minimal Solution to the Generalised 3-Point Pose Problem," in Proceedings of the IEEE Computer Society Coriference on Computer Vision and Pattern Recognition, 2004, pp. 560-567.

[48] Khoshelham K, Elberink SO, “Accuracy and resolution of kinect depth data for indoor mapping applications,” Sensors 2012;12(2):1437–54.

[49] J. Y. Bouguet. Camera calibration toolbox. http://www.vision.caltech.edu/bouguetj/calib_doc/ ,2010.

[50] N. Burrus. Kinect calibration. http://nicolas.burrus.name/index.php/Research/KinectCalibration.

[51] M.Havlena,A.Torii,andT.Pajdla. Efficient structure from motion by graph optimization. InECCV, 2010.

[52] K. Konolige and P. Mihelich. Technical description of Kinect calibration. http://www.ros.org/wiki/kinect_calibration/technical , 2011.

[53] J. Smisek and T. Pajdla. 3D Camera Calibration .MSc.the-sis, Czech Techical University in Prague, 2011.

[54] N. Snavely, S. Seitz, and R. Szeliski. Modeling the world from internet photo collections. International Journal of Computer Vision , 2007.

[55] Willow Garage. ROS - Kinect calibration: Code complete. http://www.ros.org/news/2010/12/kinect-calibration-code-complete.html,2010.