

Trym Dyrkolbotn Hauglund

Machine-Vision for Multi-Pass Robotic Welding of Large Subsea Structures

A proof of concept applying a laser vision sensor in the extraction of 3-D information from weld grooves.

Master's thesis in Mechanical Engineering

Supervisor: Lars Tingelstad

June 2019

Trym Dyrkolbotn Hauglund

Machine-Vision for Multi-Pass Robotic Welding of Large Subsea Structures

A proof of concept applying a laser vision sensor in
the extraction of 3-D information from weld groves.

Master's thesis in Mechanical Engineering
Supervisor: Lars Tingelstad
June 2019

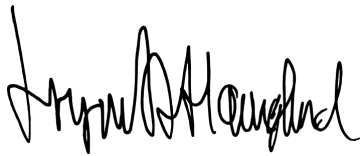
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

 **NTNU**
Norwegian University of
Science and Technology

Acknowledgments

First and foremost I would like to thank Kværner Verdal, SINTEF Manufacturing, my supervisor Lars Tingelstad and Professor Olav Egeland for the opportunity to participate on this project and for the important insight I've gotten from the visits at Kværner and through the meetings we've had, I have truly learned a great deal. I would also like to give a special thanks to my supervisor Lars Tingelstad for the patience and guidance that he have shown me through this project. I would very much like to thank Morten Lind at SINTEF for being a great source of help and for dedicating his time helping me out with my project. I would especially like to thank Espen Liavik, Jonas Dieker and Njål Munthe-Kaas for the collaboration we've had and I wish you the best of luck in your future endeavours. I would also like to thank the employees in the workshop at Valgrinda for helping me manufacture custom parts for my project, they were always was spot on!

And of course I would also like to thank the boys at the office; Dag, Didrik, Eirik, Kristoffer and Hans Kristian, for the conversations we have had which have helped me boost the mood during a hectic time. Lastly I would like to thank my friends and family for all the support I've gotten over the years, and I would like to give a special thanks to my girlfriend Marthe, which have always been there for me and always pushes me to be better, this would not have been possible without you.



Trym Dyrkolbotn Hauglund

Abstract

This master's thesis serves as an introduction to the research and theory considering the subject of laser triangulation sensors, robotics and computer vision. The primary goal of the thesis is to examine the use of structured light sensors in robotic welding of large multi-pass weld groves as well as to develop and test a laser triangulation sensor on steel weld groves. Previous research, as well as commercial actors in the field of robotic welding and laser triangulation, are presented and used as basis to evaluate the results from the test at the end. Theory and methods regarding the operation of laser vision sensors which includes the description of important system parameters like working distance, the principle of triangulation and depth of field is also discussed in this thesis. Methods for calibration of laser vision sensor is also presented, which includes the intrinsic and extrinsic calibration of image sensors, hand-eye calibration between the image sensor and robot end-effector and the estimation of the laser plane.

An experimental setup of a laser vision sensor mounted on an industrial KUKA robot is developed where the sensor consists of a Z-LASER Z18 line laser and an industrial AT C4-2040 image sensor that is able to detect laser points based on advanced internal processing. Potential factors which might have an effect on the accuracy of the sensor is also presented and discussed. Because the sensor is vision based, it suffers from the effects of environmental factors like specular reflection and ambient light which becomes apparent through the initial testing. Simple solutions to mitigate these effects are proposed and tested out at the end of the thesis, where a proof of concept is presented with the main objectives to reconstruct an accurate 3-D representation of a single-v weld groove. The final test results of the laser vision sensor is shown to have a deviation in depth measurement of 0.7 mm, compared to previously achieved results on similar projects where the reported mean deviation is < 0.5 mm.

Sammendrag

Denne masteroppgaven er en introduksjon til forskning og teori om temaer rundt laser trianguleringssensorer, robotisert sveising og datasyn. Det primære målet med oppgaven er å undersøke bruken av sensorikk som benytter seg av strukturerte lys for å inspisere overflater i robotisert sveising av flerlags sveis, som benyttes i produksjon av store offshore stål konstruksjoner. I tillegg blir også en laser trianguleringssensor utviklet og testet for å kunne identifisere ytelsesmål. Tidligere forskning samt kommersielle aktører innen robotisert sveising og lasersyn blir presentert og diskutert, hvor funn brukes som basis for å evaluere resultatene fra slutttesten. Teori og metoder for bruk av lasersyn sensorer blir også diskutert. Dette inkluderer beskrivelse av viktige systemparametere som arbeidsavstand, trianguleringsprinsipp og dybdeskarphet. I tillegg til grunnleggende teori innen robotikk og datasyn, så presenteres også metoder for kalibrering av lasersyn sensorer. Dette inkluderer intrinsisk og ekstrinsisk kalibrering av bildesensor, hånd-øye-kalibrering mellom bildesensor og robotens end-effektor, samt estimeringen av laserplan.

Et eksperimentelt oppsett av en lasersyn sensor som er montert på en industriell KUKA-robot blir presentert. Sensoren består av en Z-LASER Z18 linje laser og en industriell AT C4-2040 bildesensor som er i stand til å oppdage laserpunkt basert på avansert intern bildebehandling. Potensielle faktorer som kan ha en innvirkning på sensorens nøyaktighet presenteres og diskuteres. Fordi sensoren baserer seg på datasyn, så lider den av ekstern påvirkning fra arbeidsmiljøet slik som reflekterende overflater og lysforurensning. Enkle løsninger for å redusere slik påvirkning blir foreslått og testet i slutten av avhandlingen, hvor et bevis på konseptet blir presentert. Hovedmålet med slutttesten er å rekonstruere en nøyaktig 3-D-representasjon av en enkeltveis som kan brukes til å parameterisere sveisen. De endelige resultatene blir vist å ha et avvik i dybdemåling på 0.7 mm, sammenlignet med tidligere oppnådde resultater på lignende prosjekter hvor det rapporterte gjennomsnittlige avviket ligger på < 0.5 mm.

Contents

Acknowledgments	i
Abstract	ii
Sammendrag	iii
1. Introduction	1
1.1. Background	1
1.2. Objectives	3
1.3. Structure of the report	4
2. Robotic Welding using Machine Vision	5
2.1. Previous Research	5
2.2. Commercial Actors	8
2.2.1. Inrotech	8
2.2.2. SERVO-ROBOT	8
2.2.3. LTI Technologies	9
2.2.4. Automation Technology	10
3. Theoretical Fundamentals of Rigid Body Kinematics and Computer Vision	11
3.1. Notation	11
3.2. Rotations	12
3.2.1. The rotation matrix	13
3.2.2. Basic rotations	14
3.2.3. Composite rotation	14
3.2.4. Euler Angles	14
3.2.5. Angle-axis and quaternions	16
3.2.6. Deviation in rotation	17
3.2.7. Homogeneous transformation matrix and rigid transformations	18
3.3. Forward Kinematics	19
3.3.1. Denavit-Hartenberg convention	20

3.3.2.	Displacement of the end-effector	21
3.4.	Computer Vision	21
3.4.1.	The pinhole model	22
3.4.2.	Intrinsic and extrinsic camera parameters	23
3.4.3.	The relation between pixel coordinates, normalized image coordinates and Cartesian coordinates	24
3.4.4.	Euclidean transformation between world and camera frames	25
3.5.	Points, lines and planes in 3-D	25
3.5.1.	Points	25
3.5.2.	Lines	26
3.5.3.	Planes	27
3.6.	Plane Fitting	30
3.6.1.	Inliers and outliers	30
3.6.2.	RANSAC	32
4.	Laser Vision System	35
4.1.	System Parameters of an Image Acquisition System	35
4.1.1.	Field Of View	36
4.1.2.	Resolution	36
4.1.3.	Work distance	37
4.1.4.	Measurement principle	37
4.2.	Camera	38
4.2.1.	Area Of Interest (AOI)	39
4.2.2.	Multiple Slope Exposure	40
4.2.3.	Camera Modes	40
4.2.4.	Lens	44
4.3.	Laser	47
4.3.1.	Optics and fan angle	48
4.3.2.	Wave length and spectral response	49
5.	Calibration of Robotic Laser Vision Systems	51
5.1.	Camera Calibration	51
5.1.1.	Lens distortion	54
5.1.2.	Reprojection error	56
5.2.	Laser Plane Calibration	56
5.3.	Hand-Eye Calibration	59
5.4.	Laser Triangulation with Calibrated System	65
6.	Experimental setup	67
6.1.	System configuration	68
6.2.	Calibration Process	69
6.3.	Test Scanning of a Single-V Weld Groove	71

6.4. System Accuracy	72
6.4.1. Internal Factors	72
6.4.2. External Factors	73
6.4.3. Other factors	74
7. Results and Discussion	77
7.1. System calibration	77
7.2. Results from scanning of test piece	81
8. Conclusion and Future Work	86
8.1. Recommendations for future work	87
8.1.1. Python Code	88
A. Appendix	94
A.1. Technical specifications	94
A.2. Tables	95

List of Figures

1.1. Third jacket delivered to Johan Sverdrup from Kværner Verdal [25]	1
2.1. Heavy reflections on a polished surface [26]	7
2.2. Definitions of weld bead and weld groove geometry [26]	7
2.3. Inrotechs AMWT showcase video [21]	8
2.4. SERVO-ROBOT iCUBE showcase video [40]	9
2.5. LTI Gocator 2880 3D Laser Line scanner [30]	9
2.6. C5-series [43]	10
3.1. A rotation about the z-axis	12
3.2. Roll, pitch and yaw on a plane	15
3.3. KUKA KR10 R900 in Visual Components [6]	19
3.4. Pin hole camera [32]	22
3.5. The basic pinhole model, Zisserman [3]	22
3.6. The image plane [10]	24
3.7. Plane and noisy data points	30
3.8. Outliers vs Inliers	31
4.1. CAD model of the laser vision system	35
4.2. Field Of View	36
4.3. Triangulation geometry	37
4.4. AT C4-2040 CMOS sensor	38
4.5. Area of Interest	39
4.6. Multiple Slope Mode	40
4.7. Laser line search	41
4.8. Maximum Intensity Profile Mode	42
4.9. Threshold Mode	42
4.10. Center of Gravity Mode	43
4.11. The idealized lens model [18]	44
4.12. Depth of field	45
4.14. Z-LASER Z185 line laser	47
4.15. Gaussian and Powell laser line distribution	48
4.16. Fan Angle, q	49

4.17. Spectral Responce of thr AT C4-2040 [14]	49
5.1. Positive (barrel) and negative (pincushion) radial distortion	55
5.2. Reprojection error	56
5.4. Laser line in the image plane	58
5.5. 3-D plot of laser line in camera frame	59
5.6. Transformation and frames of hand-eye calibration	60
5.7. Illustration of two pairs of four different poses	62
5.8. Laser triangulation	65
6.1. Proposed test scenario	67
6.2. Laser vision sensor mounted on the KUKA KR16-2 end effector	68
6.3. Calibration in progress	70
6.4. Test Scene	71
6.5. Error caused by corrupted pixels	72
6.6. Light pollution and specular reflection	74
6.7. Diverging error in hand eye estimation	75
7.2. Plot of RANSAC and least squares plane	79
7.3. Error in translation and rotation	80
7.4. Estimated orientation between end effector and camera	81
7.5. Point Cloud	82
7.7. Multiple plane segmentation	84
8.1. Stereo vision scenario	88
A.1. System Architecture	97

Chapter 1.

Introduction

1.1. Background

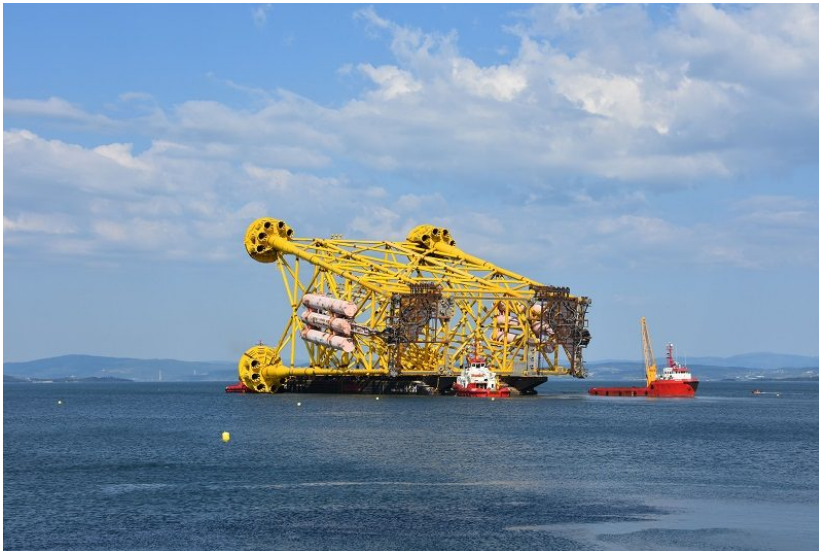


Figure 1.1.: Third jacket delivered to Johan Sverdrup from Kværner Verdal [25]

This project is done in cooperation with Kværner Verdal, one of Norway's largest offshore construction manufacturers that specializes in manufacturing of large steel jackets and topsides for offshore oil and gas installations on the Norwegian continental shelf. Jackets produced at Kværner Verdal are very large, usually at several tens of thousand tonnes, and have hundreds of pipe junctions where the braces and legs of the structure are joined together. Due to the size of the jackets, they are prefabricated in parts and assembled outdoors. Most of the assembly is

done horizontally, and as little work as possible are performed above the ground, this is done in order to decrease the risks and hazards as well as the increased complexity tied to carrying out work at heights. The prefabricated bracings are cut from a CAD-model and then welded to the legs at the pipe junctions. Most of the welding is currently done manually, where the largest weld grooves requires many layers consisting of multiple weld beads, and may take as much as a full man-year to complete. Because the welding is very labor intensive and puts a lot of strain on the welder, the welding of a single multi-layer weld groove is usually done in shifts. A general rule of thumb is that welders are only able to efficiently weld for about 20 min of every work hour, therefore the benefits of robotizing the welding process can be easily recognized.

In cooperation with SINTEF Manufacturing and NTNU, Kværner Verdals Autokons project aims to automate the welding processes of large multi-pass weld grooves using robotic arc welding. The robot cell at Kværner Verdal is delivered by igm Robotssysteme AG which is a leading company in welding automation from Switzerland. Currently Kværner Verdal is able to carry out welding of the pipe junctions with stub angle less than 80 degrees, but due to a tedious programming process there is still room for improvements. The weld grooves of the pipe junctions are often irregular-shaped due to the geometry of the joining between the different pipes, which is dependent on the the stub angle and the diameter-ratio between the pipes. This complicates the path planning process. Tests done by Kværner, using an iCAM laser scanner from igm, has shown that the result is far from perfect. To perform a good weld, the surface of the workpiece often requires to be machined and cleaned before welding. This can result in noisy data from the laser scanner due to the reflective properties of the machined steel surface.

As previously mentioned, the primary challenges with robotic welding of large subsea structures are related to the programming of welding paths and the execution of the robot trajectories using sensor feedback. Welding paths and robot trajectories can be generated with help of an offline programming system. However, this requires that the CAD model of the workpiece and the actual workpiece geometry are within the geometric tolerances of the welding process. This is not the case in manufacturing of large subsea constructions which might have large geometric discrepancies in the build geometry. Moreover, with a varying weld seam geometry several welding seams need to be applied to the weld groove, which leads to a non-uniform layout. As a consequence of this, the placement of the welding paths might need correction. In the utmost consequence, all remaining welding paths and robot trajectories need to be re-planned and corrected based on the build geometry.

The re-planning requires an iterative online strategy where a sensor system scans

and extract information from the build geometry, and uses the information to generate the remaining welding paths. This can be achieved in multiple ways, but one of the most effective methods is by using a sensor system attached to the robot, consisting of a structured light projector and high speed industrial cameras to build the 3-D model based on laser triangulation and computer vision methods.

In addition to the challenges related to the layout and execution of welding paths and robot trajectory locally in the welding seam, a primary challenge is the path planning of the robot due to the possibly very complex geometry of the work piece. This challenge requires 6-D path planning methods that can generate collision-free paths and trajectories while taking into account the process and quality requirements, the external environment of the robot as well as its geometry and working envelope.

The challenges related to geometric deviations can be analyzed at a global, local and micro level. In short, the geometric deviations at the global level relates to identification of the deviations in the relative positioning and orientation of work-piece in the robot system envelope. At the local level identification of deviations are related to the local geometry, that is the welding seam of the work piece, and to compare the physical setup towards the CAD model. At the micro level the challenges are tied to the identification of the welding seam geometry in proximity to the welding process, that is the measurement of welding bead and welding seam at the welding pistol. Welding automation at the micro level has been solved for several decades, and there exists commercial actors on the market. The general challenges for Norwegian industry, as well as the primary challenge of this project, are related to the global and local levels, which is to generate collision-free welding paths and robot trajectories for welding of large subsea constructions that satisfy the process and quality requirements of the final product.

1.2. Objectives

The primary focus of this master's thesis is on the development, calibration and performance evaluation of a machine vision system for use in the multi-pass robotic welding of large subsea structures.

The system should be developed in the robot lab, of the Robotics and Automation group at the Department of Mechanical and Industrial Engineering at Valgrinda NTNU Trondheim. It should consist of one or more laser triangulation sensors

mounted on an industrial robot which can be used to scan and build up a 3-D representation of the actual geometry of the work pieces that is to be welded. The 3-D representation should then be used to parameterize the weld groove for the programming of welding trajectories. Performance indicators should be defined, and the performance of the system should be evaluated based on said indicators.

1.3. Structure of the report

In the first chapter the status quo concerning the use of robotic welding in industrial processes is described and the objective of this thesis is defined. In the second chapter a state-of-the-art literature review where findings from research on the use of computer vision and structured light in robotic welding processes are presented. In chapter 3 the theoretical fundamentals of computer vision and robot kinematics, as well as methods for estimating planes are presented. In chapter 4 the fundamentals of laser vision systems is described, which includes the description of system parameters of an image acquisition system as well as some of the important functionalities of a modern laser image sensor. In the next section of the thesis, chapter 5, the description of methods for calibrating a laser vision sensor attached to a robot manipulator is presented. This includes calibration with respect to the intrinsic parameters of the camera, as well as the hand-eye transformation and laser plane. In chapter 6 the experimental setup is presented and the configuration of the laser vision sensor is defined. Chapter 6 also discusses some of the factors which might have contributed to errors in the system. In chapter 7 the results from the calibration as well as the results from the final test is presented and at the end of the thesis the conclusion and future work is presented.

Chapter 2.

Robotic Welding using Machine Vision

The purpose of this chapter is to give an overview of previous research done on the machine vision technologies that support robotic welding. The articles and papers presented in this chapter is only a small portion of all the research that has been done on the subject, as there are many challenges tied to the use of machine vision in a welding process. At the end of this chapter some findings from the commercial market is briefly presented to give an understanding of the technology that is relevant in similar processes today.

2.1. Previous Research

With the introduction of more sophisticated and cheaper sensor hardware and the given state of the offshore industry which demands faster real-time and more cost efficient solutions, manufacturers look to computer vision to solve the challenges related to automated welding of large structures. Vision-based sensors are often favored because of their high performance, real-time capability and adaptability. Conventional methods such as ultrasound inspection or magnetic inspection of weld bead are sub-optimal [26] because they are often too slow, requires highly skilled operators or have low degree of adaptability which makes them unattractive for robotic welding applications. With vision based sensors, online real-time inspection can be implemented to reduce cycle time and material waste. This can be achieved by inspecting each pass and terminate the welding process when defects are discovered [26]. This suggests that online inspection is necessary to achieve high quality welds.

Other methods for weld inspection including the use of structured light vision for seam tracking and 3D-geometry extraction are discussed in [45, 49, 26]. In [8] and [36] pure stereo vision without structured light is used to detect the weld joints, and in [48] a system that combines structured light and stereo vision is proposed. Other methods use structured light vision to self-regulate the welding process [11], and [28] introduces a combination of neuron-network, fuzzy control approach based on feedback from a structured light sensor that tries to simulate a human welder.

Measurement from vision sensor are not as precise as the measurement from using mechanical devices [36], but are in most cases well within the thresholds of the industrial welding processes [35] for steel construction. Mean deviation in measurements < 0.5 mm are reported in [36, 49, 26]. The self-tuning system proposed in [11] is able to achieve directional mean deviations as small as (0.04 mm, 0.07 mm) for horizontal and vertical direction respectively. The papers report that the main challenges using computer vision in surface inspection is tied to calibration of the sensors and specular reflection of light on the machined metal surface of weld joints [8, 49, 46]. An example of reflection from the laser caused by the smooth surface of the workpiece can be seen in Fig. 2.1. In addition, ambient light can also have a negative effect on the quality of the image from the sensors, an example of this is arc light from welding during online tracking, which makes the extraction of features from the image difficult [26]. To overcome this problem, the authors of [8] suggests using pure stereo vision, edge detection algorithms and filtering techniques for seam detection. In [48] a combination of stereo vision and structured light is employed to add redundancy information and improve precision and robustness of the sensor system. The paper also claims that the proposed system is less sensitive to illumination. In [45] a laser triangulation system that uses an AT-C4 CMOS camera and a laser line projector to align parts for welding. The manufacturer of the camera claims that the High Dynamic Range capabilities of the camera makes it possible to scan surfaces with in-homogeneous reflection properties.

To make an adaptive system that is able to plan weld seam trajectories on non-linear weld joints, case studies have been done to develop methods for collision free path planning for welding of brace-to-chord joints for offshore oil rigs [2]. In [29] path planning of weld seam in intersecting pipes are discussed at length. A methodology for estimating the bead layout of cross-sections in multi-pass welding groves are presented in [47]. The parameters of the weld bead layout are based on manual welds that have been carried out on steel test pieces with wall thickness larger than 40 mm The authors suggests that their the proposed methodology is applicable for adjustment of welding trajectory based on sensor feedback. One example that uses vision sensors in multi-pass robotic welding is shown in [26].

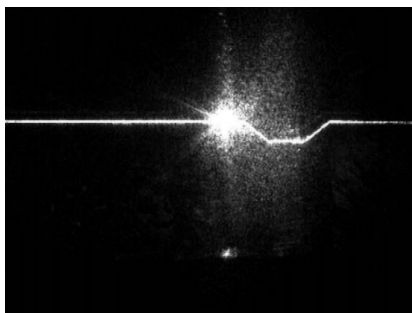


Figure 2.1.: Heavy reflections on a polished surface [26]

The authors aims to measure weld beads and detect defects without the the need for complicated 3D calibration and reconstruction techniques. The parameters needed to describe a single v-weld are illustrated in Fig. 2.2.

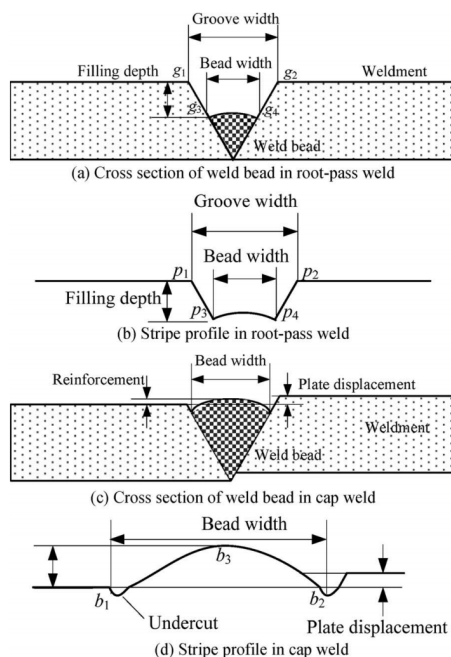


Figure 2.2.: Definitions of weld bead and weld groove geometry [26]

In [36], the authors suggests that a disadvantage of using structured light vision for bead inspection is that it restricts the the sensor to only perform search at the local level of the welding process. This means that it cannot be used for detection at the global level, that is the work envelope of the welding robot. Instead, the authors proposes an approach for path correction in pre-offline programmed

robotic welding using stereo vision.

2.2. Commercial Actors

In this section state-of-the-art technologies from some of the leading companies in vision based sensing and robotic welding are briefly presented.

2.2.1. Inrotech



Figure 2.3.: Inrotechs AMWT showcase video [21]

Inrotech [21] is a danish company located in Odense which specializes on welding automation in shipyard manufacturing and manufacturing of pipe sections for offshore windmill productions. The sensor technology used in their adaptive multipass welding system is laser line scanning sensors. The weld groove is scanned prior to the welding, and a 3D image of the weld groove is created which is used for path planning. The welding system is build upon traditional submerged arc welding where welding is done in the flat PA position where the sensor is shielded from the environment by a welding curtain. Inrotech claims that their sensor system can be used in all welding positions and is able to handle weld groove variations of about 30%, as well as having around 85% effective welding time.

2.2.2. SERVO-ROBOT

SERVO-ROBOT is a Canadian company located near Montreal. Combining seam tracking, seam finding and weld inspection makes a SERVO-ROBOT recognized

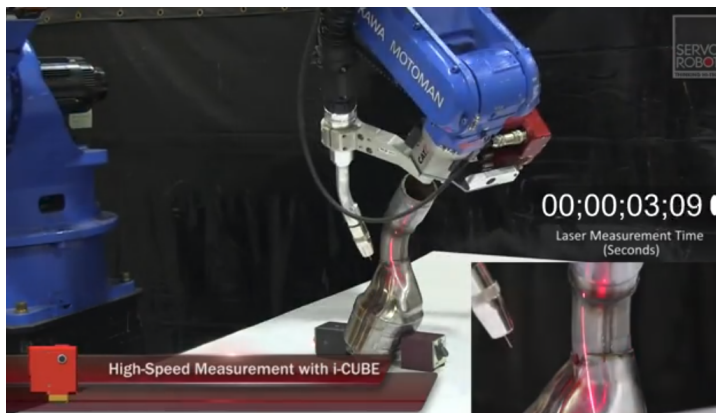


Figure 2.4.: SERVO-ROBOT iCUBE showcase video [40]

leader in 3D robot-vision and sensing system for robotic welding and automated manufacturing. Their i-CUBE seam finding technology [40] is able to locate and correct a displaced weld seam by projecting a laser line on the part and capture it in different orientations.

2.2.3. LTI Technologies



Figure 2.5.: LTI Gocator 2880 3D Laser Line scanner [30]

LMI Technologies specializes in 3D smart sensor technologies and has a long product line of advanced laser line sensors. Their Gocator 2000- and 3000 series is built for industrial usage and uses stereo vision and laser triangulation to rapidly scan large objects with complex geometries. The use of stereo vision means that the sensor delivers more information about the scene, and the sensor is able to com-

pensate for occlusion that occurs when the laser line is blocked from the cameras view. The manufacturer reports a measurement accuracy of $\pm 50\mu\text{m}$

2.2.4. Automation Technology

Automation Technology specializes in image sensors for industrial process automation and inspection tasks. Their advanced top of the line C5 sensors use internal hardware and external software technologies to extract laser line profiles and perform surface inspection at high speeds on a wide range of materials. Reported precision [44] on welding seam inspection steel welds is as low as $10\mu\text{m}$ at a welding speed of 250 mm/s.



Figure 2.6.: C5-series [43]

Chapter 3.

Theoretical Fundamentals of Rigid Body Kinematics and Computer Vision

The purpose of this chapter is to give an understanding of the fundamental theory behind the robot kinematics and computer vision. The chapter does not include the basics of vector calculus and linear algebra, which is assumed to be known. At the beginning of the chapter, basic robot kinematics is presented. This includes rigid body motions and forward kinematics. The kinematics are based on robotic literature Siciliano et.al. [41] and [9]. The theory on computer vision which is introduced is based on Zisserman et.al. [3] and [10]. This includes the description of the ideal pinhole model, image formation and camera to world transformations. At the end of the chapter a description of points, lines and planes in 3-D is presented as well as methods for plane estimation where the RANSAC algorithm and the least squares are presented.

3.1. Notation

To understand the notation used in this chapter it is useful to start with a short description. Points in 2D are written with lowercase letters and a 3-D point is written in uppercase letters

$$p = p(x, y) \quad \text{and} \quad P = P(X, Y, Z)$$

Vectors are written in bold lowercase letter and matrices are in bold uppercase letters, that is \mathbf{v} and \mathbf{M} respectively.

A point P can be represented by a 3-D column vector

$$\mathbf{p} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3.1)$$

The *homogeneous* representation of the same point is a 4-D vector

$$\tilde{\mathbf{p}} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

3.2. Rotations

Rotation in space are used to describe the orientation of an object with reference to some coordinate frame. A coordinate frame is defined by three orthogonal axes representing the 3 dimensions in Euclidean space, that is the x -axis, the y -axis and the z -axis. Coordinate frames are often used to describe rotation because they give an intuitive geometrical illustration of how one orientation relates to another, see Fig. 3.1.

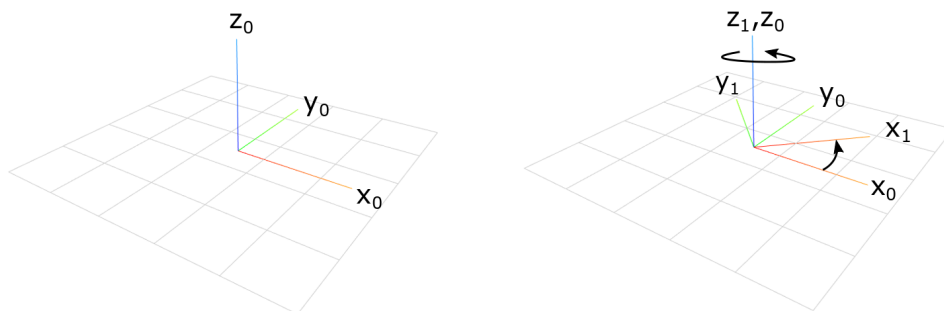


Figure 3.1.: A rotation about the z -axis

3.2.1. The rotation matrix

A rotation matrix is a 3×3 matrix that describes the rotation from one frame to another. This notation gives us information about how the axis of the frame at the end of the rotation is oriented relative to the frame at the start of the rotation.

Let frame a and frame b denote the start frame and the end frame respectively. Then a rotation from frame a to b can be written as

$$\mathbf{R}_b^a = \begin{bmatrix} \mathbf{x}_b^a & \mathbf{y}_b^a & \mathbf{z}_b^a \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (3.2)$$

where

$$\mathbf{x}_b^a = \begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \end{bmatrix} \quad \mathbf{y}_b^a = \begin{bmatrix} r_{12} \\ r_{22} \\ r_{32} \end{bmatrix} \quad \mathbf{z}_b^a = \begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \end{bmatrix} \quad (3.3)$$

represents the three axis of frame b relative frame a .

In a 3-D coordinate frame each axis is orthogonal to the two other axis, which means that the rotation matrix is an orthogonal matrix. The orthogonality gives the rotation matrix some unique and useful properties

$$\mathbf{R}^T \mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{I}, \text{ identity matrix} \quad (3.4)$$

which means that

$$\mathbf{R}^T = \mathbf{R}^{-1} \quad (3.5)$$

This can be useful when we know the rotation from one frame to another and we want to know the opposite rotation, that is

$$(\mathbf{R}_b^a)^T = (\mathbf{R}_b^a)^{-1} = \mathbf{R}_a^b \quad (3.6)$$

3.2.2. Basic rotations

Basic rotations in Euclidean space is the rotations about the x -, y - and z -axis.

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}, \text{ rotation } \theta \text{ about the } x\text{-axis} \quad (3.7)$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}, \text{ rotation } \theta \text{ about the } y\text{-axis} \quad (3.8)$$

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ rotation } \theta \text{ about the } z\text{-axis} \quad (3.9)$$

In Fig. 3.1 we see an example of a basic rotation about the z -axis.

3.2.3. Composite rotation

If we want to know the rotation from one frame to another via some intermediate frame we can use the rotations between each frame to describe the rotation between the desired frames. This is called a composite rotation.

$$\mathbf{R}_d^a = \mathbf{R}_b^a \mathbf{R}_c^b \mathbf{R}_d^c \quad (3.10)$$

and the inverse rotation is

$$\mathbf{R}_a^d = (\mathbf{R}_d^a)^{-1} = \mathbf{R}_c^d \mathbf{R}_d^c \mathbf{R}_a^b \quad (3.11)$$

3.2.4. Euler Angles

Euler angles are a simple and intuitive way of describing rotation. It was first introduced by Leonhard Euler and can describe any orientation of a rigid body by using a sequence of three angles in combination. The roll-pitch-yaw (RPY) convention often used to describe the movement of airplanes about its principal rotation axes, is an example of a common Euler angle convention. RPY rotation is given as a sequence of rotations about current axes, which implies that the current rotation axis is always affected by previous rotations in the sequence.

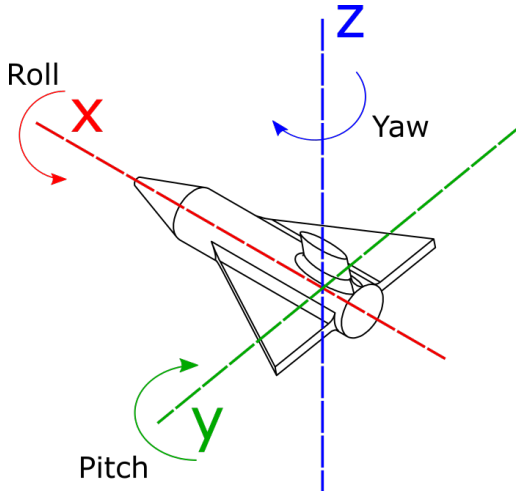


Figure 3.2.: Roll, pitch and yaw on a plane

The rotation sequence for RPY is

$$\mathbf{R}_{ZYX} = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi)$$

where the *yaw* is given by the angle ψ about the z -axis, the *pitch* is given by the angle θ about the current y -axis and the *roll* is given by the angle ϕ about the current rotated x -axis. The resulting rotation matrix is

$$\mathbf{R}_{ZYX} = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \quad (3.12)$$

Let the elements of the rotation matrix in Eq. (3.12) be given as $\mathbf{R} = r_{ij}$, then the solutions for the RPA angles are

$$\psi = \text{Atan2}(Er_{21}, Er_{11}) \quad (3.13)$$

$$\theta = \text{Atan2}(-r_{31}, E\sqrt{r_{32}^2 + r_{33}^2}) \quad (3.14)$$

$$\phi = \text{Atan2}(Er_{32}, Er_{33}) \quad (3.15)$$

where the variable E is

$$E = 1, \text{ when, } c_\theta > 0 \quad (3.16)$$

$$E = -1, \text{ when, } c_\theta < 0 \quad (3.17)$$

KUKA robot controllers describes orientation with a sequence of rotations similar

to the roll-pitch-yaw convention

$$\mathbf{R}_{KUKA} = \mathbf{R}_z(A)\mathbf{R}_y(B)\mathbf{R}_x(C) \quad (3.18)$$

where the RPY angles are given by A , B and C .

One disadvantage using Euler angles to describe orientations are that the convention suffers from *gimbal lock* which occurs when two or more rotation axes are aligned. In this situation there exists no unique solution to a given orientation. In robotics gimbal lock can cause singularities in the wrist joint which results in the robot trying compensate by doing rapid 180° joint movements to maintain the orientation. This should be avoided because it can have a serious effect on the robot end effector, tool and process.

3.2.5. Angle-axis and quaternions

Angle-axis representation is a more compact representation of rotation, where a single axis and a rotation about the axis is used to describe the spatial orientation of a coordinate frame

$$\mathbf{R} = (\theta, \mathbf{k}) \quad (3.19)$$

here \mathbf{k} is the rotation axis and θ is the rotation angle. This can be re-parameterized into four rational terms without any singularities called *the Euler parameters*

$$\eta = \cos \frac{\theta}{2} \quad (3.20)$$

$$\boldsymbol{\epsilon} = \mathbf{k} \sin \frac{\theta}{2} \quad (3.21)$$

Quaternions was first introduced by Hamilton in 1843 and are by many considered to be the golden standard in 3-D pose estimation and path interpolation. The quaternion is a complex number consisting of one real part and three imaginary parts which can be defined in terms of the angle-axis parameters (θ, \mathbf{k}) . The unit quaternion \mathbf{q} is defined as

$$\mathbf{q} = \eta + \boldsymbol{\epsilon} = \cos \frac{\theta}{2} + \mathbf{k} \sin \frac{\theta}{2} \quad (3.22)$$

where $\|\mathbf{q}\| = 1$. The rotation matrix can be described using unit quaternions

$$\mathbf{R} = \mathbf{I} + 2\eta\boldsymbol{\epsilon}^\times + 2\boldsymbol{\epsilon}^\times\boldsymbol{\epsilon}^\times \quad (3.23)$$

where ϵ^\times is the skew symmetric of ϵ . It is worth noting that the quaternion representation is not affected by gimbal lock because the rotation is described using the Euler parameters instead of an orthogonal matrix.

3.2.6. Deviation in rotation

In many applications concerning robotic path planning and computer vision, a method of quantifying and evaluate the *distance* between two rotations is necessary. In pose estimating the deviation between two rotations is the amount of rotation needed in order to achieve the desired rotation from the estimated rotation. This can be written as

$$\mathbf{R}_d = \mathbf{R}\mathbf{R}_e \quad (3.24)$$

where \mathbf{R}_d represents the desired rotation, \mathbf{R}_e is the error in rotation and \mathbf{R} is the actual estimated rotation. Its easy to see that the ideal situation occurs when $\mathbf{R}_e = \mathbf{I}$ which results in $\mathbf{R}_d = \mathbf{R}$. This leads to

$$\mathbf{R}_d\mathbf{R}^T = \mathbf{I} \quad (3.25)$$

which can be further derived into a *distance function*

$$d(\mathbf{R}_d, \mathbf{R}) = \|\mathbf{I} - \mathbf{R}_d\mathbf{R}^T\| \quad (3.26)$$

this is also referred to as a *metric*. There exists many different metrics, all with advantages and disadvantages. Huynh [15] and Kuffner [24] suggests that using quaternions in rotation estimating are the most computational and spatial efficient.

Let the actual rotation and the desired rotation be defined by their quaternions

$$\begin{aligned} \bar{\mathbf{q}} &= \eta - \epsilon \\ \mathbf{q}_d &= \eta_d + \epsilon_d \end{aligned}$$

where $\bar{\mathbf{q}}$ is the conjugate quaternion of the actual rotation, corresponding to the transpose rotation \mathbf{R}^T .

The inner product of the quaternions returns the angle θ_e between them, which is the magnitude of rotation needed to bring the actual rotation to the desired

rotation

$$\theta_e = \arccos(Q) \quad (3.27)$$

here Q is the quaternion inner product

$$Q = \mathbf{q}_d \cdot \bar{\mathbf{q}} \quad (3.28)$$

$$= \eta_d \eta - \boldsymbol{\epsilon}_d \cdot \boldsymbol{\epsilon} \quad (3.29)$$

Quaternions are based on the angle-axis representation, hence there exists exactly two unit quaternion representations of the same rotation. In order to account for multiple representations, the norm of the product should be used. The quaternion equivalent to Eq. (3.26) is then

$$d(\mathbf{q}_d, \mathbf{q}) = |1 - |Q|| \quad (3.30)$$

where $d(\mathbf{q}_d, \mathbf{q})$ is a distance metric that returns a scalar value between $[0, 1]$.

3.2.7. Homogeneous transformation matrix and rigid transformations

A transformation, that is rotation and translation, between two frames, a and b , can be described using the *homogeneous transformation matrix* \mathbf{T}_b^a .

$$\mathbf{T}_b^a = \begin{bmatrix} \mathbf{R}_b^a & \mathbf{t}_{ab}^a \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (3.31)$$

where \mathbf{t}_{ab}^a is the translation from frame a to frame b and \mathbf{R}_b^a is the rotation from frame a to frame b .

The inverse of the homogeneous transformation matrix is

$$(\mathbf{T}_b^a)^{-1} = \mathbf{T}_a^b = \begin{bmatrix} (\mathbf{R}_b^a)^T & -(\mathbf{R}_b^a)^T \mathbf{t}_{ab}^a \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_a^b & -\mathbf{R}_a^b \mathbf{t}_{ab}^a \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (3.32)$$

Composite transformations can be described the same way as composite rotations. Let the desired transformation from frame a to frame c be \mathbf{T}_c^a . If the transformation between an intermediate frame b is known, that is \mathbf{T}_b^a and \mathbf{T}_c^b , then the desired transformation is

$$\mathbf{T}_c^a = \mathbf{T}_b^a \mathbf{T}_c^b = \begin{bmatrix} \mathbf{R}_b^a \mathbf{R}_c^b & \mathbf{r}_{ab}^a + \mathbf{R}_b^a \mathbf{r}_{bc}^b \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_c^a & \mathbf{t}_{ac}^a \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (3.33)$$

Sometimes it can be useful to know the position of a point from another frame of reference than what has been given. If we know the transformation between the current frame and the desired frame, and we know the displacement between the point and the current frame, we can find the point given in the desired reference frame.

Then

$$\tilde{\mathbf{p}}^a = \mathbf{T}_b^a \tilde{\mathbf{p}}^b \quad (3.34)$$

and

$$\tilde{\mathbf{p}}^b = (\mathbf{T}_b^a)^{-1} \tilde{\mathbf{p}}^a = \mathbf{T}_a^b \tilde{\mathbf{p}}^a \quad (3.35)$$

3.3. Forward Kinematics



Figure 3.3.: KUKA KR10 R900 in Visual Components [6]

A typical robot can be seen in Fig. 3.3, it is usually made up of n number of links that are attached together in joints that can either rotate about or translate along the joint axis. The first link is defined at the base of the robot, and is denoted as link 0. The last link of the robot, link n , is called the end-effector

and can be viewed as the hand of the robot manipulator. When we know the configuration of the joint variables of the robot, which is an angle for rotational joints or a translation for a prismatic joint, it is possible to compute the orientation and position of the end-effector relative the base of the robot. This computation process is called forward kinematics.

3.3.1. Denavit-Hartenberg convention

The DH-convention is standard material in most robotic literature and it is used to describe the homogeneous transformation as a composite displacement of 4 homogeneous transformation matrices. The composite displacement can be described as follow:

1. A rotation θ about the current z-axis
2. A translation d along the current z-axis
3. A rotation α about the current x-axis.
4. A translation a along the current x-axis.

On compact form the composite transformation is

$$\mathbf{T}_i^{i-1} = \mathbf{Rot}_z(\theta_i)\mathbf{Trans}_z(d_i)\mathbf{Rot}_x(\alpha_i)\mathbf{Trans}_x(a_i) \quad (3.36)$$

where

$$\mathbf{Rot}_z(\theta_i)\mathbf{Trans}_z(d_i) = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.37)$$

and

$$\mathbf{Rot}_x(\alpha_i)\mathbf{Trans}_x(a_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.38)$$

Then Eq.(3.36) can then be derived as

$$\mathbf{T}_i^{i-1} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.39)$$

The parameters of each joint on the robot are typically presented on the following form

Link($i - 1$)	θ_i	d_i	α_i	a_i	Link i
0	θ_1	d_1	α_1	a_1	1
1	θ_2	d_2	α_2	a_2	2
2	θ_3	d_3	α_3	a_3	3
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$n - 1$	θ_n	d_n	α_n	a_n	n

3.3.2. Displacement of the end-effector

Using the DH-convention the end-effector displacement from the base of the robot can be found by calculating the transformation from the base to the end-effector which is given by

$$\mathbf{T}_n^0 = \mathbf{T}_1^0 \mathbf{T}_2^1 \mathbf{T}_3^2 \dots \mathbf{T}_n^{n-1} \quad (3.40)$$

where the transformation between each joint, that is joint ($i - 1$) and joint i is described in Eq.(3.39).

3.4. Computer Vision

In computer vision information is extracted from digital images and can be used in decision making or to build 3-D data representation of real world objects. This is an imitation of how humans and animals are able to extract and act on information acquired through sight. To be able to do this, mathematical models must be built to simulate the eyes, which can be modeled as a *camera model*. A camera model is a matrix with particular properties that describes the relationship between a 3-D point in space and its projection onto the image plane of the camera. The most common model is the ideal pinhole camera model.

3.4.1. The pinhole model

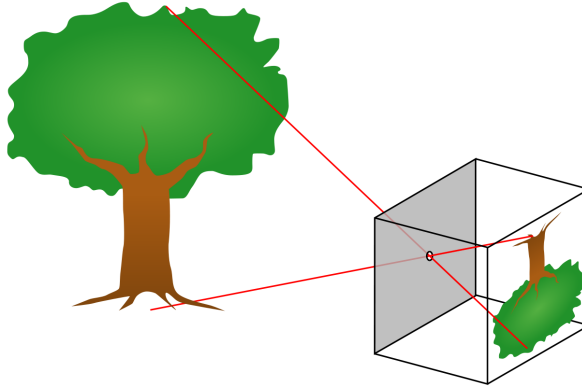


Figure 3.4.: Pin hole camera [32]

The pinhole camera is a simple idealized camera model with a single small aperture, which allows rays of light to pass through which results in a projected inverted image onto the image plane of the camera.

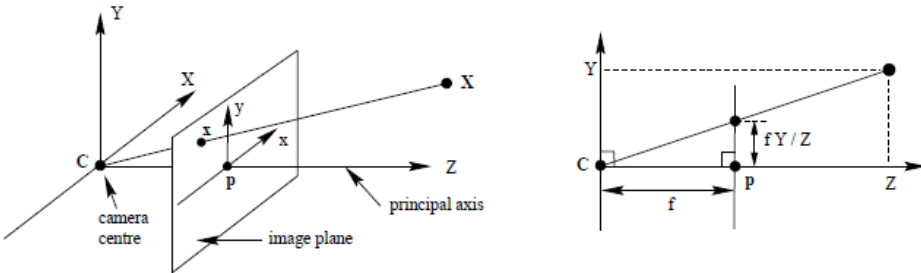


Figure 3.5.: The basic pinhole model, Zisserman [3]

A conventional representation of this model can be seen in Fig. 3.5. The *image plane*, also known as the *virtual image plane*, is positioned in front of the camera at a distance equal to the focal length f of the camera. By doing this we get the upright image instead of the inverted image. The z -axis of the camera intersects the image plane in its center p , which is called the *principal point*.

Let a 3-D world point $\mathbf{X} = [X, Y, Z]$ relative to the camera be projected onto the image plane in point $\mathbf{x} = [x_f, y_f, z_f]$ where the ray from \mathbf{X} to the origin of the camera intersect the image plane. From similarity triangles it is possible to find

the image coordinates

$$x_f = f \frac{X}{Z} \quad (3.41)$$

$$y_f = f \frac{Y}{Z} \quad (3.42)$$

$$z_f = f \frac{Z}{Z} = f \quad (3.43)$$

If we divide the image coordinate by the focal length of the camera we get the corresponding *normalized* image coordinate

$$\tilde{\mathbf{s}} = \frac{\mathbf{x}}{f} = \left(\frac{X}{Z}, \frac{Y}{Z}, 1 \right)^T = (s_x, s_y, 1)^T \quad (3.44)$$

3.4.2. Intrinsic and extrinsic camera parameters

Focal length, pixel dimension and image resolution is called *intrinsic* parameters. They are intrinsic because they are inherent to the camera, which means that different cameras have different intrinsic parameters.

The intrinsic parameters can be described using the camera calibration matrix \mathbf{K} :

$$\mathbf{K} = \begin{bmatrix} f/\rho_w & s & u_0 \\ 0 & f/\rho_h & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.45)$$

Here, f is the focal length, ρ_h and ρ_w are the pixel dimensions, u_0 and v_0 are the pixel coordinates of the center of the image.

Note that the *skew* parameter, s , which is non-zero when the image axes are not perpendicular, are assumed to be zero.

The inverse of the camera matrix K is

$$\mathbf{K}^{-1} = \frac{1}{f} \begin{bmatrix} \rho_w & 0 & -\rho_w u_0 \\ 0 & \rho_h & -\rho_h v_0 \\ 0 & 0 & f \end{bmatrix} \quad (3.46)$$

Translation and rotation of the camera in reference to some frame, usually the world frame or the base frame of the robot, are called *extrinsic* parameters since they do not depend on the camera itself. The extrinsic parameters of the camera

are as follows

$$\mathbf{T}_{camera}^{world} = \begin{bmatrix} \mathbf{R}_c^w & \mathbf{t}_{wc}^w \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (3.47)$$

3.4.3. The relation between pixel coordinates, normalized image coordinates and Cartesian coordinates

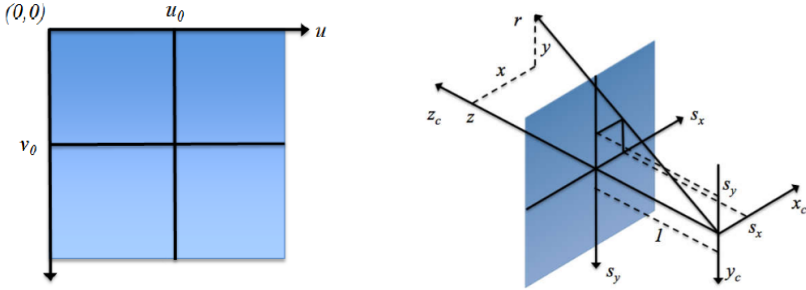


Figure 3.6.: The image plane [10]

The information extracted from images is given in pixel values, that is pixel coordinates and pixel intensity. If we want to know the position of a point in the world relative to the camera in metric values, we first need to transform the pixel coordinates.

Let the corresponding pixel coordinates of a world point in the camera image be

$$\mathbf{p} = [u, v]^T \quad (3.48)$$

The homogeneous form of the pixel coordinates is then

$$\tilde{\mathbf{p}} = [u, v, 1]^T \quad (3.49)$$

Pixel and normalized image coordinates are related by the camera calibration matrix \mathbf{K}

$$\tilde{\mathbf{p}} = \mathbf{K}\tilde{\mathbf{s}} \quad (3.50)$$

$$\tilde{\mathbf{s}} = \mathbf{K}^{-1}\tilde{\mathbf{p}} \quad (3.51)$$

The relationship between the normalized image coordinates and the actual displacement between the camera and point in the world is given by

$$z_c\tilde{\mathbf{s}} = \mathbf{r}_{cb}^c \quad (3.52)$$

where z_c is the z -coordinate of \mathbf{r}_{cb}^c which can be found if we know the transformation between the camera and the world frame.

3.4.4. Euclidean transformation between world and camera frames

Points in space will usually be expressed in terms of the world coordinate frame, where the camera frame and the world frame are related by a rotation and a translation. Let $\tilde{\mathbf{X}} = [X, Y, Z, 1]^T$ be the homogeneous representation of a point in the world. Then the corresponding homogeneous image coordinate is given by

$$\tilde{\mathbf{x}} = \mathbf{P}\tilde{\mathbf{X}} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]\tilde{\mathbf{X}} \quad (3.53)$$

where

$$\mathbf{P} = \begin{bmatrix} \alpha r_{11} + u_0 r_{31} & \alpha r_{12} + u_0 r_{32} & \alpha r_{13} + u_0 r_{33} & \alpha t_1 + u_0 t_3 \\ \beta r_{21} + v_0 r_{31} & \beta r_{22} + v_0 r_{32} & \beta r_{23} + v_0 r_{33} & \beta t_2 + v_0 t_3 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \quad (3.54)$$

has both the intrinsic and extrinsic parameters of the camera and is called the *perspective transformation matrix*.

3.5. Points, lines and planes in 3-D

3.5.1. Points

A point in space can be represented by a 3-D vector $\mathbf{p} = [X, Y, Z]^T$ where X , Y and Z represents the Cartesian coordinates of the point. Often it is useful to represent a point in homogeneous coordinates. Let \mathbf{p} be represented as a homogeneous vector

$$\tilde{\mathbf{x}} = [x_1, x_2, x_3, x_4]^T \quad (3.55)$$

where

$$\mathbf{p} = \frac{1}{x_4} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3.56)$$

The homogeneous representation is often written on the form

$$\tilde{\mathbf{x}} = x_4 \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \quad (3.57)$$

where it is easier to see that x_4 is a scaling factor, and that $\tilde{\mathbf{x}}$ represents the same point in space \mathbf{p} for all $x_4 \neq 0$.

3.5.2. Lines

A straight line is defined as an infinite set of colinear points. In two dimensions the line equation is well known

$$y = ax + b \quad (3.58)$$

where a is the slope of the line and b gives the interception between the line and the y -axis. In 3-D the line representation is somewhat more complicated. It is often easier to view the line as a *ray* with a initial starting point and an end point. Let the origin of a line be $\mathbf{p}_0 = [x_0, y_0, z_0]^T$ and the end point $\mathbf{p} = [x, y, z]^T$. The equation for a line in space can then be represented by three parametric equations

$$x = v_1 t + x_0 \quad (3.59)$$

$$y = v_2 t + y_0 \quad (3.60)$$

$$z = v_3 t + z_0 \quad (3.61)$$

which gives us the vector equation for a line

$$\mathbf{p} = t\hat{\mathbf{v}} + \mathbf{p}_0 \quad (3.62)$$

where \mathbf{p} is a function of the scalar t which gives an indication of how far along the line the end point is located and $\hat{\mathbf{v}}$ represents the unit direction vector of the line

$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{|\mathbf{v}|} = \frac{[v_1, v_2, v_3]^T}{\sqrt{v_1^2 + v_2^2 + v_3^2}} \quad (3.63)$$

3.5.3. Planes

A plane π in 3-D can be represented as the set of all points $\mathbf{p} \in \mathbb{R}^3$ that satisfy the plane equation

$$ax + by + cz + d = 0 \quad (3.64)$$

where the plane π is represented by its coefficients

$$\boldsymbol{\pi} = [a, b, c, d]^T \quad (3.65)$$

A more geometrical representation of the plane is

$$\boldsymbol{\pi} = \begin{bmatrix} \mathbf{n} \\ d \end{bmatrix} \quad (3.66)$$

where $\mathbf{n} = [a, b, c]^T$ represents the plane normal in Cartesian space, and $\frac{d}{|\mathbf{n}|}$ is the distance to the origin along the plane normal.

Plane defined by 3 points

Let \mathbf{p}_1 , \mathbf{p}_2 and \mathbf{p}_3 represent three known points in space. If the points lies in the the unknown plane π , it is possible to find the plane from the known points. Let two vectors in the plane be

$$\mathbf{v}_1 = \mathbf{p}_1 - \mathbf{p}_2 \quad (3.67)$$

$$\mathbf{v}_2 = \mathbf{p}_2 - \mathbf{p}_3 \quad (3.68)$$

Since the two vectors both are parallel to the plane, it follows that the cross product between the two vectors must be perpendicular to the plane and in the direction of the normal vector to the plane

$$\mathbf{n} = \mathbf{v}_1 \times \mathbf{v}_2 \quad (3.69)$$

The distance from the origin to the plane is then given by

$$\delta|\mathbf{n}| = \mathbf{p}_3 \cdot (\mathbf{p}_1 \times \mathbf{p}_2) \quad (3.70)$$

where $d = -\delta|\mathbf{n}|$. The plane is then

$$\boldsymbol{\pi} = \begin{bmatrix} \mathbf{v}_1 \times \mathbf{v}_2 \\ -\mathbf{p}_3 \cdot (\mathbf{p}_1 \times \mathbf{p}_2) \end{bmatrix} = \begin{bmatrix} \mathbf{n} \\ d \end{bmatrix} \quad (3.71)$$

Plane defined by a set of points

Let $\mathbf{A} \in \mathbb{R}^{n \times 4}$ be a matrix with n -row vectors representing n -homogeneous points

$$\mathbf{A} = \begin{bmatrix} \tilde{\mathbf{p}}_1^T \\ \tilde{\mathbf{p}}_2^T \\ \vdots \\ \tilde{\mathbf{p}}_n^T \end{bmatrix} \quad (3.72)$$

where $\tilde{\mathbf{p}}_i^T = [\mathbf{p}_i, 1] = [x_i, y_i, z_i, 1]$. To find the unknown plane $\boldsymbol{\pi}$ fitted to the points in \mathbf{A} , the plane must satisfy Eq. (3.76). This can be written as a homogeneous system of linear equations

$$\mathbf{A}\boldsymbol{\pi} = 0 \quad (3.73)$$

which can be solved using singular value decomposition

$$\mathbf{A} = \sum_{i=1}^4 \sigma_i \mathbf{u}_i \mathbf{v}_i^T = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \quad (3.74)$$

\mathbf{U} and \mathbf{V} are orthogonal matrices and $\boldsymbol{\Sigma}$ is a diagonal matrix containing the singular values of \mathbf{A} . The only non-trivial solution for $\boldsymbol{\pi}$ is the last column of \mathbf{V} , that is

$$\boldsymbol{\pi} = k\mathbf{v}_4 \quad (3.75)$$

where k is some constant. Its worth noting that if \mathbf{A} contains noisy points so that the points are approximately on the plane, the solution becomes a least square solution.

Condition for a point to be in a plane

Given a plane $\boldsymbol{\pi} = [a, b, c, d]^T$ and a point $\mathbf{p} = [x, y, z]^T$ represented on homogeneous form as $\tilde{\mathbf{p}}$, the condition for the point to be in the plane is then

$$\boldsymbol{\pi}^T \tilde{\mathbf{p}} = 0 \quad (3.76)$$

which is equivalent to Eq. (3.64).

Distance between a point and a plane

Given a plane $\boldsymbol{\pi} = [a, b, c, d]^T$ and the centroid to the plane $\boldsymbol{p}_0 = [x_0, y_0, z_0]^T$ such that

$$ax_0 + by_0 + cz_0 + d = 0 \quad (3.77)$$

where

$$x_0 = \frac{\sum x}{n}, \quad y_0 = \frac{\sum y}{n} \quad \text{and} \quad z_0 = \frac{\sum z}{n} \quad (3.78)$$

Then the distance between any given point $\boldsymbol{p} = [x, y, z]$ and the plane along the plane normal is

$$\delta = \frac{\boldsymbol{n}^T(\boldsymbol{p} - \boldsymbol{p}_0)}{|\boldsymbol{n}|} \quad (3.79)$$

Intersection between a line and a plane

Let a plane be defined as an infinite set of points \boldsymbol{p} for which

$$(\boldsymbol{p} - \boldsymbol{p}_0) \cdot \boldsymbol{n} = 0 \quad (3.80)$$

where \boldsymbol{n} is the plane normal and an arbitrary point \boldsymbol{p}_0 lying on the plane. From Eq. (3.62) the vector equation for a line can be written as

$$\boldsymbol{p} = d\hat{\boldsymbol{l}} + \boldsymbol{l}_0 \quad (3.81)$$

where d is a scalar, $\hat{\boldsymbol{l}}$ is the unit direction vector and \boldsymbol{l}_0 is some point on the line. If the line intersects the plane, then the set of plane points, \boldsymbol{p} , must contain at least one point of intersection which solves Eq. (3.80). The line equation can be substituted into Eq. (3.80)

$$(d\hat{\boldsymbol{l}} + \boldsymbol{l}_0 - \boldsymbol{p}_0) \cdot \boldsymbol{n} = 0 \quad (3.82)$$

which can be solved for d

$$d = \frac{(\boldsymbol{l}_0 - \boldsymbol{p}_0) \cdot \boldsymbol{n}}{\hat{\boldsymbol{l}} \cdot \boldsymbol{n}} \quad (3.83)$$

Note that this has two possible outcomes, namely

$$\begin{cases} \hat{\mathbf{l}} \cdot \mathbf{n} \neq 0 \\ \hat{\mathbf{l}} \cdot \mathbf{n} = 0 \end{cases} \quad (3.84)$$

If $\hat{\mathbf{l}} \cdot \mathbf{n} = 0$ there are two possibilities; either there exists no solution because the line never intersects the plane; or all points along the line lie in the plane, which implies that the line itself is contained in the plane. If $\hat{\mathbf{l}} \cdot \mathbf{n} \neq 0$ then d can be computed and the point of intersection can be found from Eq. (3.81)

3.6. Plane Fitting

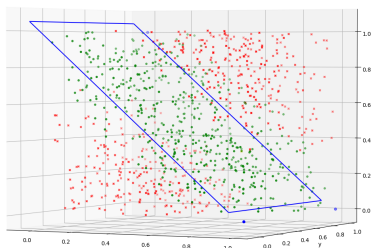


Figure 3.7.: Plane and noisy data points

If we want to fit a plane in space to a set of data points, we need to find the best plane model which fits the point data. This can be done in several different ways, each with different advantages and disadvantages. In this section two such methods will be discussed, namely least square fit and random sample consensus (RANSAC). Concepts like inliers and outliers are also shortly discussed.

3.6.1. Inliers and outliers

Most data sets and point clouds from the real world contain noisy data. Noisy data is often referred to as outliers because the data lies outside the general population of data. Noise can potentially make it difficult to find an accurate model which describes the general case of the data. In Fig. 3.7 the relationship

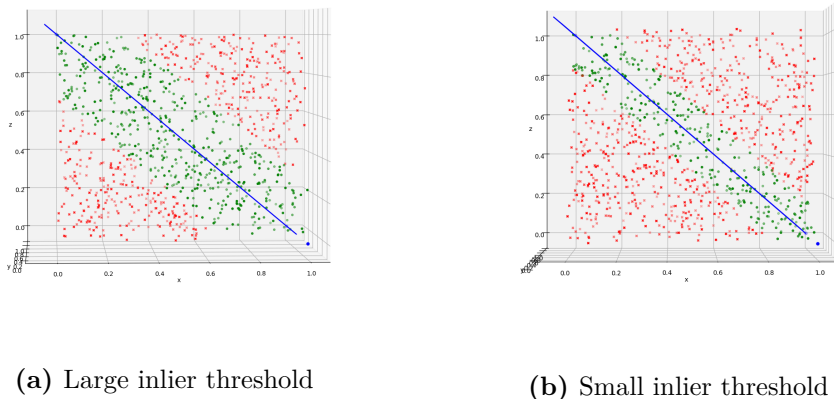


Figure 3.8.: Outliers vs Inliers

between the model estimate, a plane in this case, and the set of point data is illustrated. The green points lie within some predefined threshold of the plane and are defined as inliers to the plane, while the red points are defined as outliers because they lie outside the threshold.

Least Squares

Least squares is a mathematical method for finding a curve in 2-D or a plane in 3-D that minimizes the sum of the squared residuals. The residuals are the offset between the actual points and the estimated model. Because the squares of the residuals are used, LS is sensitive to outliers, which means that if the data set contains outliers with large offsets the outliers will have a disproportionate effect on the fitted model.

Let a plane be defined by its normal vector $\mathbf{n} = [a, b, c]$ and a distance d . A point $\mathbf{p} = [X, Y, Z]$ that lies in the plane is then defined by

$$\mathbf{n} \cdot \mathbf{p} + d = 0 \tag{3.85}$$

this can be written as the usual plane equation

$$aX + bY + cZ + d = 0 \tag{3.86}$$

Since a plane is 3 dimensional and the plane equation above uses 4 parameters the

solution space is over-determined. A workaround is to assign the z-component of the normal to the plane equal to 1, that is $c = 1$ so that

$$aX + bY + d = -Z \tag{3.87}$$

Taking n points Eq.(3.87) becomes

$$\begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ d \end{bmatrix} = - \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_n \end{bmatrix} \tag{3.88}$$

Let

$$\mathbf{A} = \begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} a \\ b \\ d \end{bmatrix}, \mathbf{b} = \begin{bmatrix} -z_0 \\ -z_1 \\ \vdots \\ -z_n \end{bmatrix} \tag{3.89}$$

Then Eq. (3.88) can be written as a system of equations

$$\mathbf{Ax} = \mathbf{b} \tag{3.90}$$

The least squares solution can be found by solving for \mathbf{x}

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \tag{3.91}$$

To evaluate the fitted model, the vertical offset, between the points in the data set and the fitted model is calculated as follows

$$\mathbf{e} = \mathbf{b} - \mathbf{Ax} \tag{3.92}$$

where the residual is equal to the Frobenius norm

$$\|e\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n abs(e_{ij})^2} \tag{3.93}$$

3.6.2. RANSAC

Random sample consensus or (RANSAC) was first introduced by Fischler and Bolles [12], and is an algorithm that can accurately estimate a model even when

subjected to noisy data. The RANSAC-algorithm takes n random samples from the data set and estimates a model from the random sampled points and test the model on the data set. The number of random samples is typically the smallest number of data points required to describe the model. The test first computes the distances between all the points and the plane, similar to Eq. (3.76) and then counts the amount of inliers to the estimated model, based on some threshold. This is repeated until a model that satisfy the desired ratio of inliers is found. A best fit model can then be estimated using least squares or singular value decomposition based on the inlier data set. The best fit model is kept until a more accurate model is found.

Disadvantages of RANSAC includes the requirement of problem specific settings to perform well and the settings often need to be based on experimentation. Such settings include inlier threshold, inlier/outlier-ratio and number of iterations. If the number of iterations is set too low the estimated model may be not optimal, or even fit the data. By increasing the iterations, the probability of having a good estimated model increases, but so does the computational cost.

The inlier tolerance threshold can in many cases be found empirically, though this process can often be tedious and problem specific. A more reasonable method for setting the inlier threshold is to estimate a model, compute the mean and standard deviation of the errors between the model and the points in the data set, then set the threshold to the mean error plus one or two standard deviations.

Let the error vector \mathbf{e} be defined as

$$\mathbf{e} = [d_1, d_2, \dots, d_i]^T \quad (3.94)$$

where

$$d_i = \frac{|A(x_i - x_0) + B(y_i - y_0) + C(z_i - z_0)|}{\sqrt{A^2 + B^2 + C^2}} \quad (3.95)$$

is the error in distance between the plane and point \mathbf{p}_i along the plane normal. The mean error μ and standard deviation σ is then

$$\mu = \frac{1}{N} \sum_{i=1}^N d_i \quad (3.96)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (d_i - \mu)^2} \quad (3.97)$$

The inlier threshold is then computed from

$$t = \mu \pm a\sigma \tag{3.98}$$

where a is a constant that can be found empirically with typical values $\in [0, 2]$

Chapter 4.

Laser Vision System

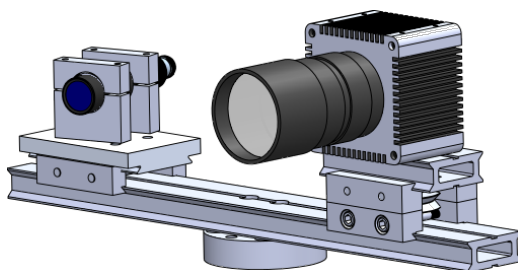


Figure 4.1.: CAD model of the laser vision system

There are a number of elements that needs to be considered in order to properly set up an image acquisition system in a laser-triangulation configuration. In this chapter the main parameters of the imaging system is briefly described and the main components of the triangulation setup is presented at the end of the chapter.

4.1. System Parameters of an Image Acquisition System

In this section the fundamental parameters of an image acquisition system as suggested by [22] are discussed. The parameters include working distance, field of view, resolution, image sensor size and depth of field.

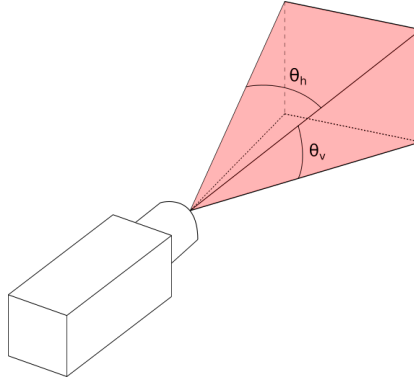


Figure 4.2.: Field Of View

4.1.1. Field Of View

The *Field Of View* also called *Angle Of View* of the camera can be illustrated by an open rectangular pyramid that captures the area contained in the camera scene, this is illustrated in Fig. 4.2. The FOV is a function of the cameras focal length, the image sensor size and the pixel dimensions. It can be measured horizontally, vertically or diagonally and is usually given in degrees or radians. Horizontal and Vertical FOV can be found from

$$\theta_h = 2 \cdot \arctan \left(\frac{W}{f \cdot 2\rho_w} \right) \quad (4.1)$$

$$\theta_v = 2 \cdot \arctan \left(\frac{H}{f \cdot 2\rho_h} \right) \quad (4.2)$$

where, W and H is the width and height of the sensor chip in millimeters, ρ_w and ρ_h is the pixel dimensions and f is the focal length of the camera. The base area of the open rectangular pyramid is the dimensions of the scene that the camera is able to capture. This area is dependent on the FOV as well as the displacement from the scene, where a larger FOV or displacement equals a larger area.

4.1.2. Resolution

The resolution of the image acquisition system is the smallest feature that the camera is able to distinguish. This is tied to the pixel size of the camera as well as the amount of pixels the camera is able to produce for a given area. In order

to make accurate measurements the smallest feature in the scene should be at least two pixels wide. Needless to say a camera with high resolution is capable of producing very detailed images, which in turn makes it more accurate.

4.1.3. Work distance

The working distance of the camera is the distance between camera lens and the observed object in the scene. When operating with lenses that have manually adjustable focus, the working distance needs to be more or less constant, otherwise the observed scene might be out of focus, resulting in less accurate measurements. How the work distance is chosen is affected by many factors and can have a large impact on the accuracy of the measurements done by the vision system.

4.1.4. Measurement principle

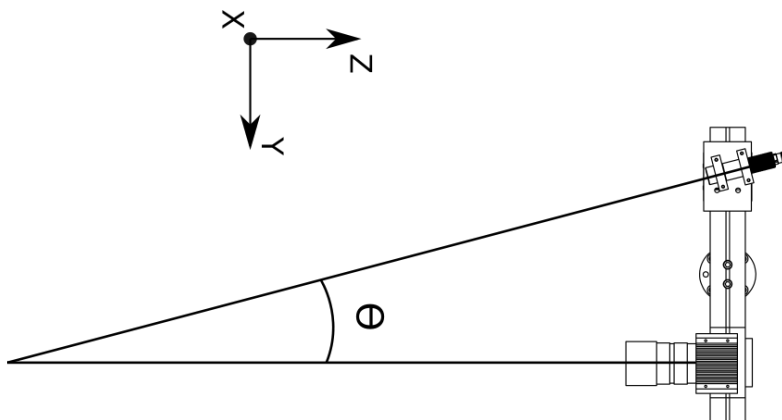


Figure 4.3.: Triangulation geometry

According to the measurement principle of a laser triangulation system, the laser projects a line onto the object from one direction, and the camera observes the object from another direction. The angle between the two directions defines the triangulation geometry of the system. The triangulation configuration can be

set up in a variety of different configurations. Some are more prone to *occlusion*, where parts of the laser line are blocked from the camera's view, while others offer increased height resolution. How the triangulation system is configured, is highly dependent on the use case. Consider Fig.4.3, in this particular configuration, often referred to as *reverse geometry*, the camera is perpendicular to the object surface and the laser line is projected onto the object at an angle Θ to the cameras z -axis which is called the *triangulation angle*. In this setup there is little occlusion of the laser line since the camera is viewing the line perpendicular to the surface.

With this triangulation setup the height resolution, ΔZ along the image ray can be approximated from the following equation

$$\Delta Z \approx \frac{\Delta X}{\tan(\Theta)} \quad (4.3)$$

where ΔX is the resolution along the projected laser line. It is easy to verify that when the triangulation angle is increased, the distance ΔZ is decreased.

4.2. Camera



Figure 4.4.: AT C4-2040 CMOS sensor

In order to detect the laser line and transform it into usable data, a light sensitive sensor is needed. Until recently the most commonly used sensor was the charge-

coupled device (CCD) sensor. The CCD sensor have a semiconductor chip that is divided into rectangular areas called pixels. The pixels are sensitive to light and reacts when hit by photons, producing a small electrical charge which can be read out. Pixel hit by more photons will produce a larger charge due to the increased light intensity. In order to determine the position of the light spot, the data from the CCD sensor must be post-processed. The light intensity profile is read out from the sensor and the position of the light spot can then be found with sub-pixel precision by calculating the pixel position that corresponds to some chosen intensity algorithm. Another important feature in post-processing is noise filtering, which makes the sensor more robust and less susceptible to noise. Recently the use CCD sensors have been declining due to the introduction of low cost complementary metal-oxide-semiconductor sensors (CMOS). The CMOS sensor captures light similarly to the CCD, via arrays of photon-detectors, but have lower power consumption and are cheaper to produce. There are some disadvantages using a CMOS sensor, that is it has lower performance in environment with poor lighting and higher sensitivity to noise. Still, it has similar performance to the CCD sensor when used together with laser light projectors, which combined with the low cost makes it attractive in laser sensor technologies.

4.2.1. Area Of Interest (AOI)

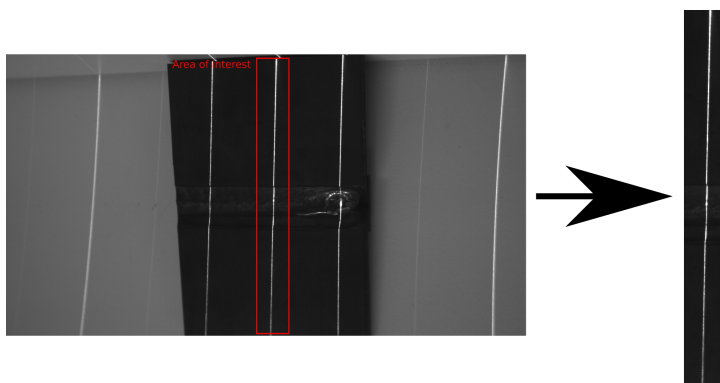


Figure 4.5.: Area of Interest

Modern laser vision sensors have AOI-search and AOI-tracking capabilities, which search for the laser line and continuously tracks it. Setting an AOI is crucial to get the most effective laser line acquisition possible, since this significantly lowers the amount of data transferred from the camera. [Figure 4.5](#) gives an example of how a potential area of interest is illustrated as a red box. The rightmost part of the image represents the cropped image that is processed instead of the full

image of the scene, this significantly reduces the amount of data and decreases processing time without any loss of information.

4.2.2. Multiple Slope Exposure

In some cases the scene might be differently lit, for example as a result of external light sources or in-homogeneous surface structure. The information in area with both high and low saturation might be equally important, which means that if the amount of light is restricted to counter the areas with high saturation, this might result in loss of important information in parts of the scene with low saturation. To counter this the dynamic range of the sensor can be increased through a process called *multiple-slope exposure* which restricts bright pixels up to a certain preset threshold and resets the value of those pixels. This way brighter areas of the scene is exposed at a shorter period while the darker areas are exposed for longer period, resulting in a high contrast image with both the dark and the bright areas visible for image analysis.

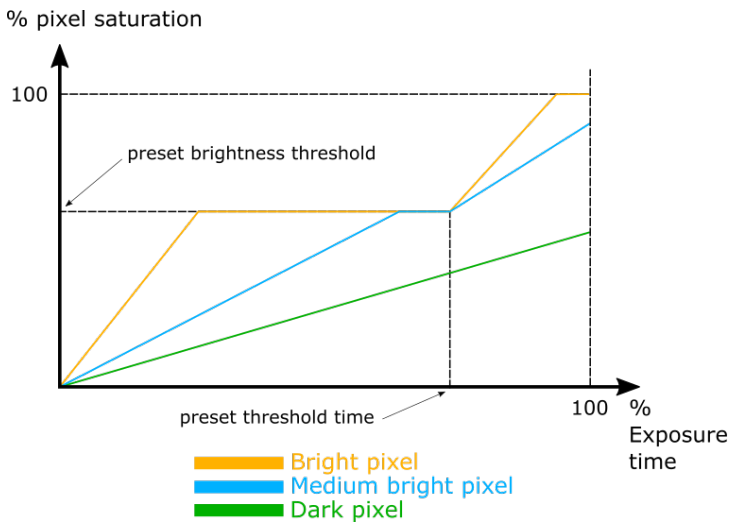


Figure 4.6.: Multiple Slope Mode

4.2.3. Camera Modes

Vision sensors that are specialized towards laser vision can typically be operated in a variety of modes including different 3-D profile modes as well as the default image

mode. The most common 3-D modes are Maximum Intensity (MAX), Threshold (TRSH) and Center of Gravity (COG). In these modes the pixel position of the laser line is computed based on the pixel intensity profile and the corresponding algorithm. In order to suppress signal noise, only pixels with intensity values higher than some intensity threshold AOI_{TRSH} are processed.

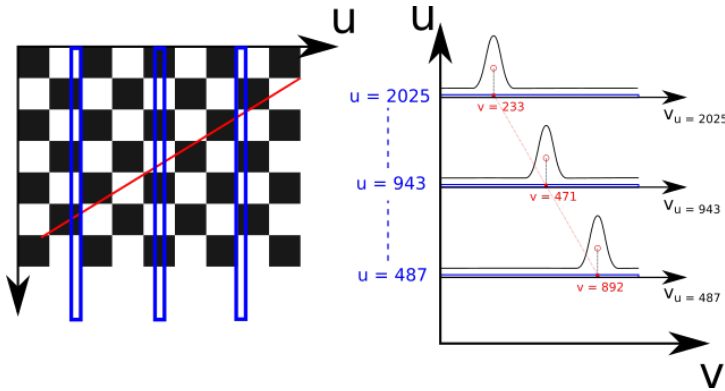


Figure 4.7.: Laser line search

Figure 4.7 gives an example of how the camera extracts the laser line from the image data. The image is divided into columns of pixels, that is, if the resolution of the image is 2048×1080 then the image is divided into 2048 columns of 1080 pixels. For each column the pixel position that corresponds to the given intensity algorithm is extracted. In this particular example it is the COG position that is extracted, where the pixel position of three of the imaged laser line points are

$$\begin{aligned} p_1^L &= (487, 897) \\ p_2^L &= (943, 471) \\ p_3^L &= (2025, 233) \end{aligned}$$

Maximum Intensity Mode (MAX)

The maximum intensity mode calculates the position of the pixel with the maximum intensity of the laser beam profile. The output in this mode includes the position as well as the maximum intensity value, P_{MAX} and I_{MAX} respectively. It is worth noting that if there is more than one local maximum, that is to say, when the intensity is saturated, the position of the first detected maximum is returned. This could lead to a position error and should be avoided.

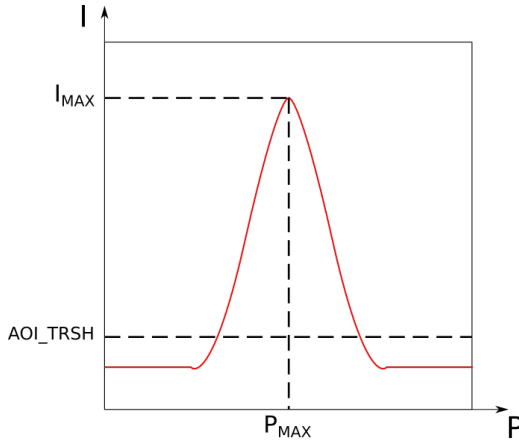


Figure 4.8.: Maximum Intensity Profile Mode

Threshold Mode (TRSH)

In this mode the position of the left (P_L) and the right (P_R) edge of the laser beam profile are detected for a given threshold intensity threshold AOI_{TRSH} , and the position value of the laser line is approximated from

$$P_{TRSH} = \frac{(P_L + P_R)}{2} \tag{4.4}$$

The output is either the left and right threshold position separately or the sub-pixel position ($P_L + P_R$) and the line width ($P_R - P_L$). The maximum intensity is also an optional output.

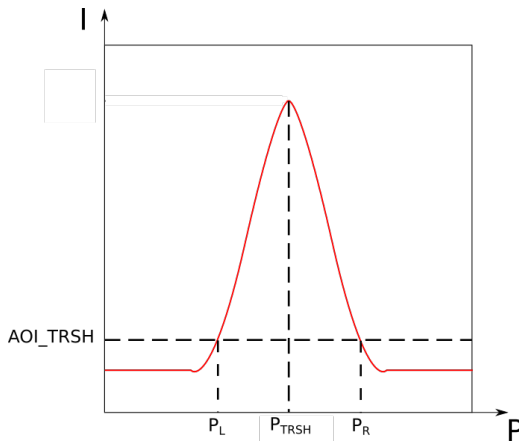


Figure 4.9.: Threshold Mode

Center of Gravity (COG)

The COG mode computes the pixel position from the center of gravity of the laser beam profile. It is considered to be more robust and resilient than the two other algorithms with precision down to 6 subpixels [17].

When the position value of the left edge (P_L) of the laser profile is given for an intensity threshold value AOI_{TRSH} , the sum of intensity values I_S and the sum of first order moment M_S can be computed from

$$I_S = \sum I_P \quad (4.5)$$

$$M_S = I_S \cdot P \quad (4.6)$$

where P is the position value along the laser beam profile.

The position value of the laser line, that is the center of gravity of the beam profile, is then obtained from

$$P_{COG} = P_L + \frac{M_S}{I_S} \quad (4.7)$$

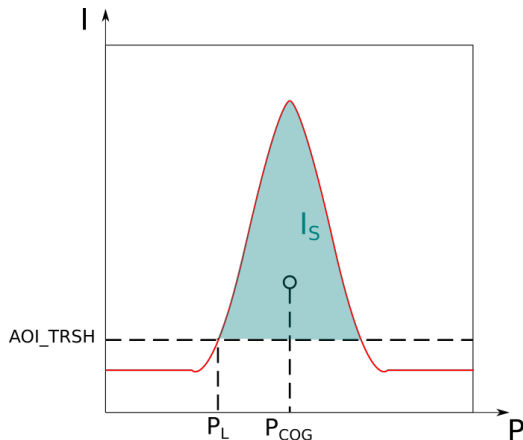


Figure 4.10.: Center of Gravity Mode

4.2.4. Lens

The main purpose of the lens is to collect the light reflected from the scene and focus it on the the light sensitive image sensor inside the camera. A single lens often has multiple layers of glass inside, which alters the image rays in different ways in order to correct the deviations caused by aberrations which is small defects in the lenses which causes the image rays to be offset from their true positions. Parameters such as lens radius, distance between the internal glass in the lens,

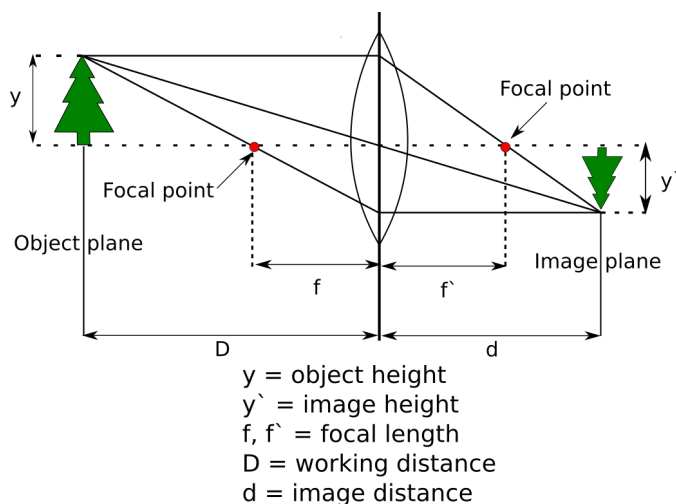


Figure 4.11.: The idealized lens model [18]

the distance between the lens and the object (working distance) and the distance between the lens and the image sensor all have an effect on the image.

Magnification

The ratio between the image and the actual real world object is called *magnification*. It is the focal length of the lens that defines its magnification. If the lens have fixed elements, the working distance and the magnification is also fixed. On the other hand if the elements of the lens are adjustable, the lens can be used at different working distances. This is more common in situations and products where versatility is appreciated, that is commercial available cameras and mobile phones. Industrial applications often requires custom sensor calibration and sensors with high repeatability and accuracy. Having adjustable lens elements makes the lens mechanically unstable which in turn can have an impact on the sensor output and can result in the need for total re-calibration of the sensor.

Figure 4.11 shows the relationship between an object and its projected image. Let β denote the magnification of the lens, then

$$\beta = \frac{d}{D} = \frac{y'}{y} \quad (4.8)$$

Depth Of Field

The depth of field of a lens is the "depth" at which the lens is able to keep objects in focus. A lens with large depth of field is therefore able to keep objects located at greater distances from each other in focus. The limits of the DOF can be calculated from

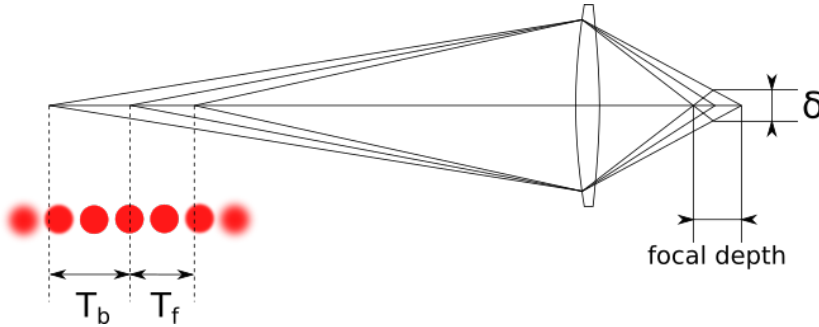


Figure 4.12.: Depth of field

$$T_b = \frac{\delta \cdot F \cdot L^2}{f^2 - \delta \cdot F \cdot L} \quad (4.9)$$

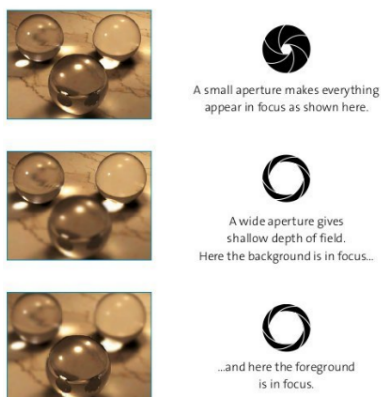
$$T_f = \frac{\delta \cdot F \cdot L^2}{f^2 + \delta \cdot F \cdot L} \quad (4.10)$$

$$DOF = T_b + T_f \quad (4.11)$$

where T_b and T_f is the backward and forward depth of field respectively, L is the distance to the object, δ is the *permissible circle of confusion* which is dependent on sensor format, f is the focal length and F is the F-number which is connected to the *lens aperture*.

Aperture

The aperture or iris tells us the amount of light that can pass through the lens. It is defined by a numerical value called a *F-number* which is the focal length of the lens divided by its effective aperture. A small F-number means that the aperture is large and vice versa. Having a small aperture makes more of the scene appear in focus which is equivalent to having a larger depth of field, this is illustrated in Fig. 4.13a. The trade-off having a small iris opening is that it makes the image darker since less light passes through the lens. This means that a large aperture is often needed in environments with poor lighting, which in turn makes the depth of field shallower. Lenses with adjustable iris normally uses increments, see Fig. 4.13b, where each increment represents a 50% reduction in the amount of light passing through the lens.



(a) Aperture and Depth of field [18]



(b) Aperture increments on a FujiFilm CF16HA-1 lens

Aberrations

With a perfect lens, all light rays from a single point in the scene will be focused on a single point on the image plane. Most lenses are only near-perfect, which means there are some aberrations. The most common are defect aberration and chromatic aberration. Defect aberrations are caused by small irregularities and defects on the surface of the lens which deflects the image rays and inhibits them to converge to a single point of focus, which causes the lens to not focus correctly. Without proper lens design and camera calibration to counter this, performing accurate measurements using computer vision would be impossible. The second

form of aberrations are chromatic aberration which might occur when white light enters the lens and disperses. This leads to blurred edges and color fringe patterns in the image. To mitigate the error caused by chromatic aberration, a monochromatic sensor, which is a sensor that only outputs gray scale values, could be used if the color of the object that is measured is of no importance. Furthermore a single color light source like a laser could also be used to further avoid chromatic aberration.

4.3. Laser



Figure 4.14.: Z-LASER Z185 line laser

A laser projector is capable of projecting high intensity light and focus it onto objects forming sharp features on the surface of the object. These features can be observed using a camera and analyzed to calculate the depth information of the surface. This information can be used to build a height map or point cloud that can be applied in reverse engineering or surface inspection. The the geometry of the features depends on the type of projection, but one of the most common projection is the line projection which forms a line on the object where the laser plane intersects with the object surface. The projection from a laser is easy to distinguish in a variety of lighting conditions making it the preferred choice in industrial applications that demands measurement with high speed and accuracy.

4.3.1. Optics and fan angle

Traditional cylindrical lenses produces a Gaussian distribution of light along the beam profile, which results in heavy illumination at the center of the projection and poorly illuminated edges. *Powell lenses* are typically preferred over Gaussian lenses because they uniformly distribute laser light along the projection, this is illustrated in see Fig. 4.15.

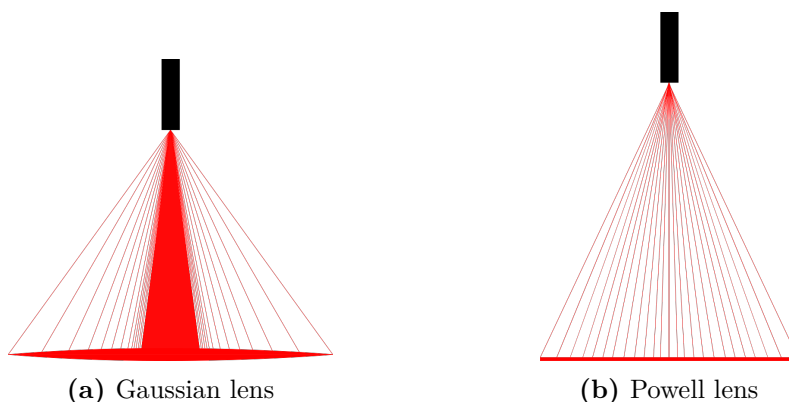


Figure 4.15.: Gaussian and Powell laser line distribution

Another factor which contributes to the beam profile is the *standoff distance* of the laser. The standoff is the distance at which the spot size of the laser beam is at its smallest, which is desirable because it provides higher resolution and accuracy. Naturally this distance should be set very close to or equal the actual work distance of the system. The standoff distance is defined at the center of the *measurement range* of the laser sensor, which is the laser-equivalent to the depth-of-field of a camera. Typically this range is within $\pm 50\text{mm}$ of the standoff distance. Laser lenses that have adjustable focus lets the laser vision system be used at a greater range of distances, where the trade-off having a larger standoff distance with a large field of view results in decreased resolution and accuracy.

The *fan angle* of the laser determines the size of the laser projection at a specific distance. When scanning large objects, a longer laser line is required in order to cover the desired part of the geometry. This might be solved by increasing the work distance, but as previously discussed this will have a negative effect on the accuracy of the system and should be avoided. Instead a laser with a larger fan angle at the same work distance is recommended. The fan angle can be computed

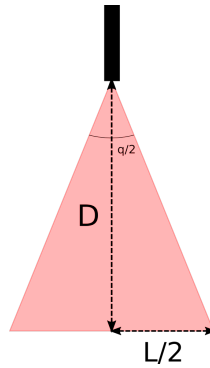


Figure 4.16.: Fan Angle, q

from

$$q = 2 \cdot \arccos \left(\frac{D}{\sqrt{D^2 + \frac{L^2}{4}}} \right) \quad (4.12)$$

where D is the working distance of the laser vision system, L is the length of the laser line and q is the fan angle.

4.3.2. Wave length and spectral response

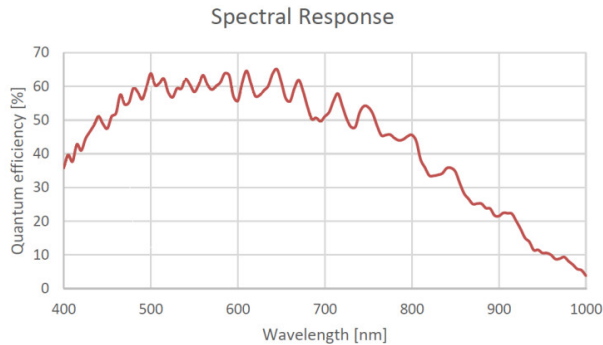


Figure 4.17.: Spectral Response of thr AT C4-2040 [14]

The wavelength of a laser gives an indication on the color of the light that is produced by the laser, where the two most common colored lasers are between 500 nm-600 nm (green) and 630 nm-700 nm (red). When lasers are used together with an image sensor in a laser vision system, the choice of laser wavelength should be consistent with the maximum spectral response of the image sensor. A camera

with high spectral response at a certain wavelength, will have a higher saturation capacity for that given wavelength which yields a better signal-to-noise ratio and a higher dynamic range. The spectral response is often measured as *quantum efficiency* which is the probability of how many electrons will be generated per 100 photons hitting the pixel of the image sensor.

Chapter 5.

Calibration of Robotic Laser Vision Systems

This chapter gives an introduction to the theory and methods for calibration of the laser triangulation system used in this thesis. The structure of the chapter is as follows. The first section is related to the calibration of laser vision system, that is the calibration of intrinsic and extrinsic parameters of the camera as well as the calibration of the laser plane. In the second section hand-eye calibration between the camera and end-effector of the robot is presented, and in the third section a method for laser triangulation using a calibrated system is presented.

5.1. Camera Calibration

Calibration of the camera is necessary in order to extract accurate metric information about the scene in the camera images. The method used in this thesis was first introduced by Zhang [50] and Bouguets [5] and proposes a flexible and robust technique for calibration of cameras that only requires the camera to observe a checkerboard pattern from a few different orientations. The relative motion between the camera and the scene is not needed, and can be estimated from the calibration which greatly simplifies the task.

Let the image point $\tilde{\mathbf{x}} = [u, v, 1]^T$ and the corresponding world point $\tilde{\mathbf{X}} = [X, Y, Z, 1]^T$ be related through the usual pin-hole camera model:

$$\lambda \tilde{\mathbf{x}} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}] \tilde{\mathbf{X}} = \mathbf{K}[r_1 \ r_2 \ r_3 \ t] \tilde{\mathbf{X}} \quad (5.1)$$

where λ is an arbitrary scale factor, \mathbf{K} is the intrinsic parameter matrix defined in Eq.(3.45), $[\mathbf{R} \mid \mathbf{t}]$ is the extrinsic parameter matrix of the camera where \mathbf{r}_i represents the column vectors of \mathbf{R}

Equation (5.1) can be simplified by assuming that the model plane is on $Z = 0$ of the world coordinate system

$$\tilde{\mathbf{X}} = [X, Y, 0, 1]^T = [X, Y, 1]^T$$

then

$$\lambda \tilde{\mathbf{x}} = \mathbf{K}[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}] \tilde{\mathbf{X}} \quad (5.2)$$

where the camera perspective transformation matrix is

$$\mathbf{P} = \mathbf{K}[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}] = [\mathbf{p}_1 \ \mathbf{p}_2 \ \mathbf{p}_3] \quad (5.3)$$

The perspective transformation matrix \mathbf{P} is also referred to as a homography because it maps points in the image plane to points in the world frame.

Since \mathbf{r}_1 and \mathbf{r}_2 are orthogonal we get the two basic constraints on the intrinsic parameters:

$$\mathbf{p}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{p}_2 = 0 \quad (5.4)$$

$$\mathbf{p}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{p}_1 = \mathbf{p}_2^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{p}_2 \quad (5.5)$$

Due to \mathbf{P} having 8 degrees of freedom and 6 extrinsic parameters, (5.4) and (5.5) are the only two intrinsic constraints that can be obtained.

Let

$$\mathbf{B} = \mathbf{K}^{-T} \mathbf{K}^{-1} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \quad (5.6)$$

The matrix \mathbf{B} is symmetric and is defined by a 6D vector:

$$\mathbf{b} = [b_{11}, b_{12}, b_{22}, b_{13}, b_{23}, b_{33}]^T \quad (5.7)$$

where

$$b_{11} = \frac{1}{\alpha^2} \quad (5.8)$$

$$b_{12} = b_{21} = 0 \quad (5.9)$$

$$b_{13} = b_{31} = \frac{-u_0\beta}{\alpha^2\beta} \quad (5.10)$$

$$b_{22} = \frac{1}{\beta^2} \quad (5.11)$$

$$b_{23} = b_{32} = -\frac{v_0}{\beta^2} \quad (5.12)$$

$$b_{33} = \frac{(-u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \quad (5.13)$$

Let the column vectors of \mathbf{P} be denoted as $\mathbf{p}_i = [p_{i1}, p_{i2}, p_{i3}]^T$. We can then define:

$$\mathbf{p}_i^T \mathbf{B} \mathbf{p}_j = \mathbf{a}_{ij}^T \mathbf{b} \quad (5.14)$$

where

$$\mathbf{a}_{ij} = \begin{bmatrix} p_{i1}p_{j1} \\ p_{i1}p_{j2} + p_{i2}p_{j1} \\ p_{i2}p_{j2} \\ p_{i3}p_{j1} + p_{i1}p_{j3} \\ p_{i3}p_{j2} + p_{i2}p_{j3} \\ p_{i3}p_{j3} \end{bmatrix} \quad (5.15)$$

The two intrinsic constraints Eq.(5.4) and Eq.(5.5) can then be rewritten as 2 homogeneous equations in \mathbf{b} :

$$\begin{bmatrix} \mathbf{a}_{12}^T \\ (\mathbf{a}_{11} - \mathbf{a}_{22})^T \end{bmatrix} \mathbf{b} = \mathbf{0} \quad (5.16)$$

Let the camera observe n images of the checkerboard pattern, from stacking Eq.(5.16) n -times we get:

$$\mathbf{A} \mathbf{b} = \mathbf{0} \quad (5.17)$$

where \mathbf{A} is a $2n \times 6$ matrix.

If $n \geq 3$ it is possible to find a non-trivial solution \mathbf{b} , defined up to a scale, using singular value decomposition [23].

$$\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} = \text{svd}(\mathbf{A}) \quad (5.18)$$

The columns of \mathbf{U} are the left singular vectors of \mathbf{A} , $\mathbf{\Sigma}$ is a diagonal matrix containing the singular values of \mathbf{A} and the columns of \mathbf{V} contains the right singular vectors of \mathbf{A} .

The solution \mathbf{b} to Eq.(5.17) can be computed where the solution is equal to the last column of \mathbf{V} .

If \mathbf{b} is found, the intrinsic parameters can be calculated

$$v_0 = \frac{b_{12}b_{13} - b_{11}b_{23}}{b_{11}b_{22} - (b_{12})^2} \quad (5.19)$$

$$\lambda = b_{33} - \frac{(b_{13})^2 + v_0(b_{12}b_{13} - b_{11}b_{23})}{b_{11}} \quad (5.20)$$

$$\alpha = \sqrt{\lambda/b_{11}} \quad (5.21)$$

$$\beta = \sqrt{\lambda b_{11}/(b_{11}b_{22} - (b_{12})^2)} \quad (5.22)$$

$$\gamma = \frac{-b_{12}\alpha^2\beta}{\lambda} \quad (5.23)$$

$$u_0 = \frac{\gamma v_0}{\beta} - \frac{b_{13}\alpha^2}{\lambda} \quad (5.24)$$

5.1.1. Lens distortion

The pin-hole camera model is an idealized model without lens [31] and does not account for lens distortion which most cameras are affected by. Lens distortion occurs during the initial projection of the world onto the image plane. There are

two main types of distortion, radial distortion which causes straight lines in the image to not appear straight, and tangential distortion which is caused by the lens and image plane not being parallel to each other.

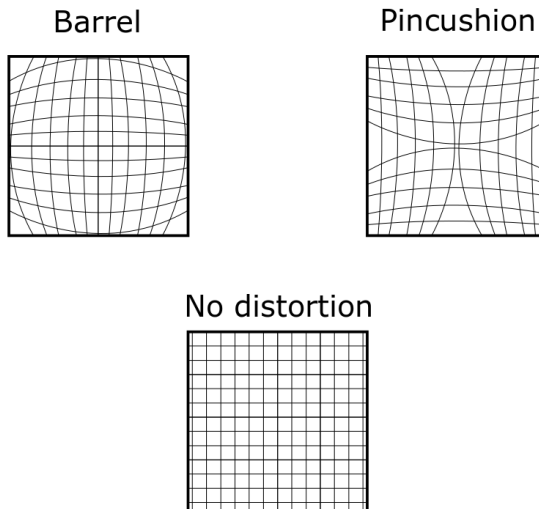


Figure 5.1.: Positive (barrel) and negative (pincushion) radial distortion

In distorted images the position of feature points cannot be used for accurate measurements, therefore in order to use the pin hole model we first need to undistort the distorted image. The image distortion coefficients are calculated during camera calibration and are usually given in the vector form

$$\mathbf{d}_{\text{coeff}} = [k_1, k_2, p_1, p_2, k_3] \quad (5.25)$$

where k_1 , k_2 and k_3 are the radial distortion coefficients and p_1 and p_2 are the tangential distortion coefficients.

Radial distortion is modeled as

$$\begin{aligned} x_{\text{distorted}} &= x(1 + k_1r^2 + k_2r^4 + k_3r^6) \\ y_{\text{distorted}} &= y(1 + k_1r^2 + k_2r^4 + k_3r^6) \end{aligned} \quad (5.26)$$

and tangential distortion is modeled as

$$\begin{aligned} x_{\text{distorted}} &= x + [2p_1xy + p_2(r^2 + 2x^2)] \\ y_{\text{distorted}} &= y + [p_1(r^2 + 2y^2) + 2p_2xy] \end{aligned} \quad (5.27)$$

where x and y is the undistorted normalized image coordinates and $r = x^2 + y^2$.

5.1.2. Reprojection error

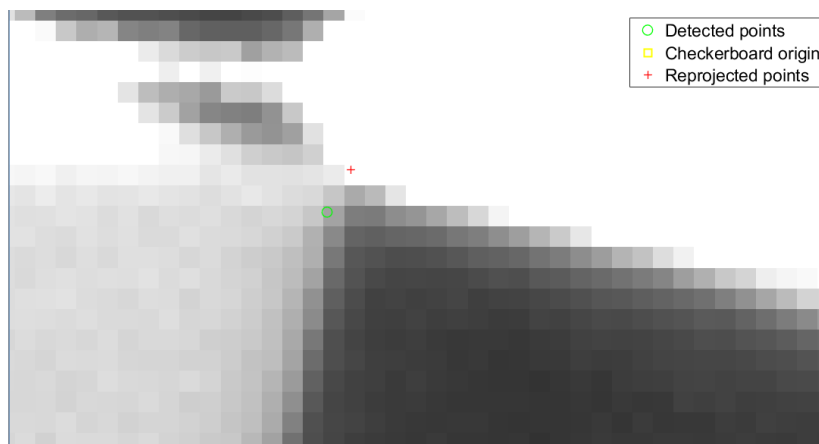


Figure 5.2.: Reprojection error

The reprojection error is defined as the euclidean distance between a projected feature point and the corresponding measured point in the image, this is illustrated in Fig. 5.2. An image with large amount of deviations can be filtered out based on its mean reprojection error and a more accurate estimation can be achieved. The overall mean reprojection error is often used as a measure on the accuracy of the calibration and is given as a numerical value in pixels.

5.2. Laser Plane Calibration

The laser plane calibration problem is to estimate the transformation between the laser plane and the camera frame. The method proposed in this section is based on Zhou and Zhang [51] which introduced a method of calibration a laser stripe plane concurrent to the intrinsic calibration of the camera.

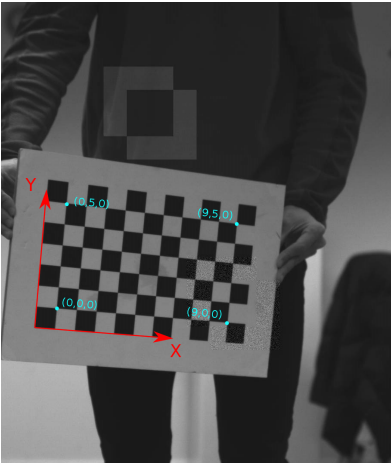
Let a planar object like a checkerboard be viewed from different angles to determine the intrinsic parameters of the camera. When the intrinsic parameters of the camera are known it is possible to calculate the translation and orientation between the camera frame and the local world frame Fig. 5.3a. Let the homogeneous transformation between the local calibration frame and the camera at pose

i be given from the extrinsic camera calibration

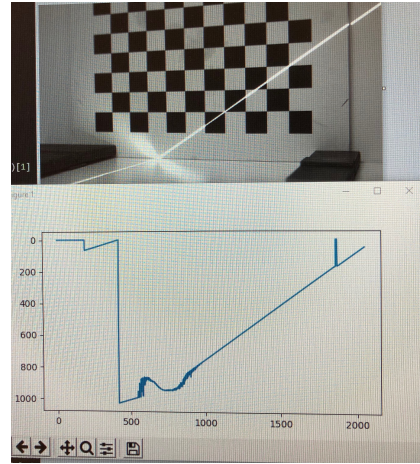
$$\mathbf{T}_i^c = \begin{bmatrix} \mathbf{R}_i^c & \mathbf{t}_i^c \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (5.28)$$

where \mathbf{R}_i^c is the relative orientation and \mathbf{t}_i^c is the relative translation of the camera frame.

Consider Fig. 5.3b, when the laser plane intersect the checkerboard, a bright line is formed across it. Using image processing techniques or cameras that are specialized towards laser vision, this line can be extracted as an array of image pixel points. The AT C4 camera used in this thesis have a built-in processing chip which have 3-D functions for detecting laser lines based on gray scale intensity of pixels along the laser line, see subsection 4.2.3.



(a) Local World Coordinate Frame



(b) Laser line COG image

The laser points on the calibration target are imaged in the camera image plane as pixel coordinates $\mathbf{p} = [u, v]^T$, see Fig. 5.4. In order to build a point cloud that can be used to estimate the laser plane, the pixel coordinates along the laser line are transformed according to Eq. (3.51)

$$\tilde{\mathbf{s}}_k = \mathbf{K}^{-1} \tilde{\mathbf{p}}_k \quad (5.29)$$

where $\tilde{\mathbf{p}}_k = [\mathbf{p}_k, 1]^T$ is the k -th pixel along the laser line and \mathbf{K} is the camera intrinsic matrix.

At each pose the planar calibration pattern have a plane normal equal to the third

column of the extrinsic rotation matrix \mathbf{R}_i^c

$$\mathbf{n}_i^c = \mathbf{z}_i^c \quad (5.30)$$

and a point in the plane equal to the origin of the calibration object

$$(\mathbf{p}_0)_i^c = \mathbf{p}_0 = \mathbf{t}_i^c \quad (5.31)$$

An image ray with origin $\mathbf{l}_0 = [0, 0, 0]$ and unit direction vector $\hat{\mathbf{l}}$, goes through the image plane of the camera in $\tilde{\mathbf{s}}_k$ and intersects the calibration target in point $\mathbf{P}_k = [X, Y, Z]$. According to Eq. (3.52), the actual 3-D displacement between the image plane of the camera and the laser points in the world can be computed from normalized image coordinates if the z -component of the actual displacement is known. Using Eq. (3.62) and Eq. (3.83), the displacement Z can be found from

$$Z = \frac{\mathbf{p}_0 \cdot \mathbf{n}_i^c}{\hat{\mathbf{l}} \cdot \mathbf{n}_i^c} \quad (5.32)$$

and the k -th 3-D laser point can be computed as

$$\mathbf{P}_k = Z \tilde{\mathbf{s}}_k \quad (5.33)$$

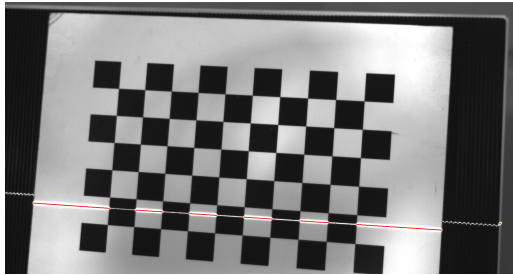


Figure 5.4.: Laser line in the image plane

This process of estimation the 3-D laser points are done for each image, resulting in a point cloud consisting of all laser points found by the camera. The point cloud are then used to estimate the laser plane normal \mathbf{n}^c using the RANSAC algorithm as well as a centroid \mathbf{p}_0^c which then defines the laser plane

$$\boldsymbol{\pi}^c = \begin{bmatrix} \mathbf{n}^c \\ \mathbf{p}_0^c \end{bmatrix} \quad (5.34)$$

where the plane is defined in the camera reference frame.

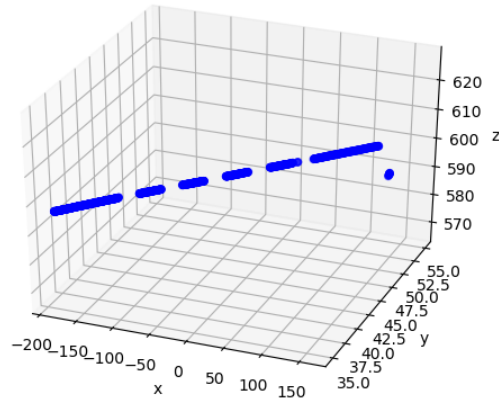


Figure 5.5.: 3-D plot of laser line in camera frame

5.3. Hand-Eye Calibration

When a laser vision system is used to estimate positions of points in the scene, these points will be given in the camera reference frame. If the laser vision system is attached to a robot manipulator and used to inspect the scene by moving the robot end-effector around, the points must be related to the relative motion between the end effector and robot base. This is achieved when the relation between the laser vision system and the robot end effector is known. The process of estimating this relation is often referred to as *hand-eye* or *eye-in-hand* calibration.

There exists many different methods of performing hand-eye calibration [7, 42, 34, 48]. Many of these are very fast and powerful, but are considered to be out of the scope of this paper. Instead the method proposed in this paper is based on a method for hand eye calibration first introduced by Park and Martin [33], where a system of equations $\mathbf{AX} = \mathbf{XB}$ is solved with respect to the unknown transformation \mathbf{X} .

Consider Fig. 5.6, let frame O represent the world origin, E is the end-effector frame, C is the camera frame and W is the work-space frame. The transformation \mathbf{T}_W^O describes the relative position and orientation between the world origin and the work space, \mathbf{T}_W^C describes the transformation between the work-space and the camera, \mathbf{T}_C^E is the transformation between the end-effector and the camera and \mathbf{T}_E^O is the transformation of the end effector relative the robot base.

The desired transformation is the transformation between end-effector and camera

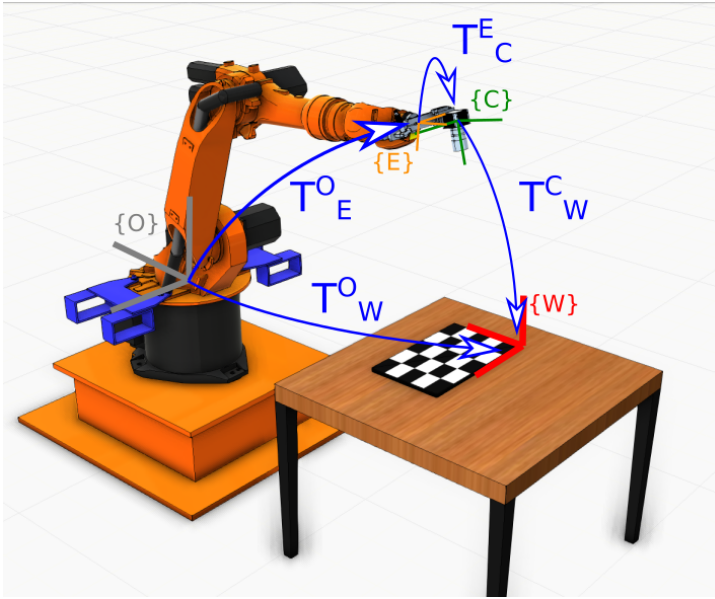


Figure 5.6.: Transformation and frames of hand-eye calibration

\mathbf{T}_C^E , that is

$$\mathbf{X} = \mathbf{T}_C^E \quad (5.35)$$

The base to end-effector transformation \mathbf{T}_E^O can be computed using forward kinematics when the joint vector of the robot is known. The transformation between the camera and the work space \mathbf{T}_W^C can be estimated from the camera calibration. From Fig. 5.6 it can be verified that

$$\mathbf{T}_W^O = \mathbf{T}_E^O \mathbf{T}_C^E \mathbf{T}_W^C \quad (5.36)$$

For convenience it is often helpful to assume that the world frame is located at the base of the robot

$$\mathbf{T}_W^O = \begin{bmatrix} 1 & 0 & 0 & b_x \\ 0 & 1 & 0 & b_y \\ 0 & 0 & 1 & b_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.37)$$

where $[b_x, b_y, b_z]$ is the position of the work space origin relative the robot base frame. This way we can relate the 3-D points estimated from the laser vision sensor when we know the transformation between the camera and the robot base.

To find the unknown hand-eye transformation \mathbf{X} the robot is moved to pairs of different poses. Let the two different poses be denoted \mathbf{q}_{1a} and \mathbf{q}_{1b} , where \mathbf{q} is the robot joint vector. The transformation between the end effector and the robot base at each pose is denoted as

$$\begin{aligned} &(\mathbf{T}_E^O)_{1a} \\ &(\mathbf{T}_E^O)_{1b} \end{aligned}$$

and the transformation between the workpiece and the camera at each pose is

$$\begin{aligned} &(\mathbf{T}_W^C)_{1a} \\ &(\mathbf{T}_W^C)_{1b} \end{aligned}$$

The position and orientation of the work space origin with respect to the base of the robot never change for each different pose, that is

$$(\mathbf{T}_E^O)_{1a} \mathbf{X} (\mathbf{T}_W^C)_{1a} = (\mathbf{T}_E^O)_{1b} \mathbf{X} (\mathbf{T}_W^C)_{1b} \quad (5.38)$$

Let

$$\mathbf{A}_1 = (\mathbf{T}_E^O)_{1b}^{-1} (\mathbf{T}_E^O)_{1a} \quad \text{and} \quad \mathbf{B}_1 = (\mathbf{T}_W^C)_{1b} (\mathbf{T}_W^C)_{1a}^{-1} \quad (5.39)$$

then Eq.(5.38) can be rewritten into

$$\mathbf{A}_1 \mathbf{X} = \mathbf{X} \mathbf{B}_1 \quad (5.40)$$

In order to accurately estimate \mathbf{X} at least 2 pairs ($n \geq 2$) of different poses are required. This will give n sets of Eq.(5.42)

$$\begin{aligned} \mathbf{A}_1 \mathbf{X} &= \mathbf{X} \mathbf{B}_1 \\ \mathbf{A}_2 \mathbf{X} &= \mathbf{X} \mathbf{B}_2 \\ &\vdots \\ \mathbf{A}_n \mathbf{X} &= \mathbf{X} \mathbf{B}_n \end{aligned} \quad (5.41)$$

If we define

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_n \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \\ \mathbf{B}_n \end{bmatrix}$$

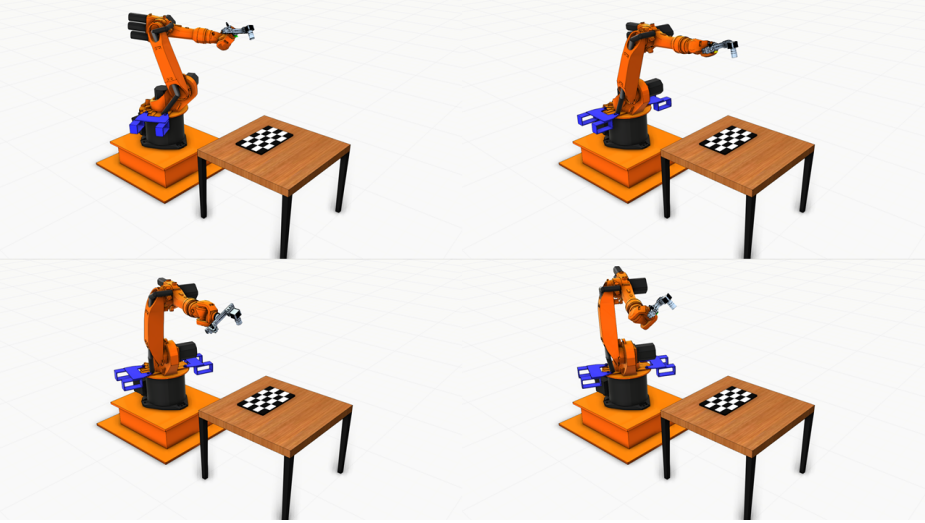


Figure 5.7.: Illustration of two pairs of four different poses

then the basic hand-eye equation can be written as

$$\mathbf{AX} = \mathbf{XB} \quad (5.42)$$

Equation (5.42) can be expanded into

$$\begin{bmatrix} \mathbf{R}_a & \mathbf{t}_a \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_x & \mathbf{t}_x \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_x & \mathbf{t}_x \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_b & \mathbf{t}_b \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (5.43)$$

which can be written into two equations

$$\mathbf{R}_a \mathbf{R}_x = \mathbf{R}_x \mathbf{R}_b \quad (5.44)$$

$$\mathbf{R}_a \mathbf{t}_x + \mathbf{t}_a = \mathbf{R}_x \mathbf{t}_b + \mathbf{t}_x \quad (5.45)$$

These equations can be solved by first finding \mathbf{R}_x from Eq.(5.44) and then compute \mathbf{t}_x from Eq.(5.45). It is worth noting that we can find an expression for \mathbf{R}_a from Eq.(5.44)

$$\mathbf{R}_a = \mathbf{R}_x \mathbf{R}_b \mathbf{R}_x^T \quad (5.46)$$

Let θ_a , \mathbf{k}_a , θ_b and \mathbf{k}_b be the angle-axis representations of \mathbf{R}_a and \mathbf{R}_b respectively. Then \mathbf{R}_a and \mathbf{R}_b can be written on the exponential form as

$$\mathbf{R}_a = \exp(\theta_a \mathbf{k}_a^\times) = \mathbf{I} + \sin \theta_a \mathbf{k}_a^\times + (1 - \cos \theta_a) \mathbf{k}_a^\times \mathbf{k}_a^\times \quad (5.47)$$

$$\mathbf{R}_b = \exp(\theta_b \mathbf{k}_b^\times) = \mathbf{I} + \sin \theta_b \mathbf{k}_b^\times + (1 - \cos \theta_b) \mathbf{k}_b^\times \mathbf{k}_b^\times \quad (5.48)$$

where \mathbf{k}^\times is the skew symmetric matrix form of $\mathbf{k} = [k_x, k_y, k_z]$

$$\mathbf{k}^\times = \begin{bmatrix} 0 & -k_z & k_y \\ k_z & 0 & -k_x \\ -k_y & k_x & 0 \end{bmatrix} \quad (5.49)$$

The matrix exponential function of a skew-symmetric matrix is defined as

$$\exp(\mathbf{k}^\times) = \mathbf{I} + \mathbf{k}^\times + \frac{1}{2!}(\mathbf{k}^\times)^2 + \frac{1}{3!}(\mathbf{k}^\times)^3 + \dots \quad (5.50)$$

Since $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ Eq.(3.4) we can verify that

$$\mathbf{R} \exp(\mathbf{k}^\times) \mathbf{R}^T = \mathbf{I} + \mathbf{R} \mathbf{k}^\times \mathbf{R}^T + \frac{1}{2!}(\mathbf{R} \mathbf{k}^\times \mathbf{R}^T)^2 + \frac{1}{3!}(\mathbf{R} \mathbf{k}^\times \mathbf{R}^T)^3 + \dots \quad (5.51)$$

which according to Eq.(5.50) gives the result

$$\mathbf{R} \exp(\mathbf{k}^\times) \mathbf{R}^T = \exp(\mathbf{R} \mathbf{k}^\times \mathbf{R}^T) \quad (5.52)$$

Using Eq.(5.47),Eq. (5.48) and Eq. (5.52), Eq.(5.46) can be written as

$$\mathbf{R}_a = \exp(\theta_a \mathbf{k}_a^\times) = \mathbf{R}_x(\exp(\theta_b \mathbf{k}_b^\times) \mathbf{R}_x^T = \exp[\mathbf{R}_x(\theta_b \mathbf{k}_b^\times) \mathbf{R}_x^T] \quad (5.53)$$

it follows that

$$\theta_a \mathbf{k}_a^\times = \mathbf{R}_x(\theta_b \mathbf{k}_b^\times) \mathbf{R}_x^T \quad (5.54)$$

Equation (5.54) can be written on vector form as

$$\theta_a \mathbf{k}_a = \mathbf{R}_x(\theta_b \mathbf{k}_b) \quad (5.55)$$

This equation is called an *orthogonal Procrustes problem*. The solution for the rotation matrix \mathbf{R}_x can be found from the minimization problem

$$\min_R L_r = \sum_{i=1}^n \|(\theta_a \mathbf{k}_a)_i - \mathbf{R}_x(\theta_b \mathbf{k}_b)_i\|^2 = \text{tr}(\mathbf{K}_a \mathbf{K}_a^T + \mathbf{K}_b \mathbf{K}_b^T - 2\mathbf{R} \mathbf{K}_b \mathbf{K}_a^T) \quad (5.56)$$

where

$$\mathbf{K}_a = ((\theta_a \mathbf{k}_a)_1, \dots, (\theta_a \mathbf{k}_a)_n) = \begin{bmatrix} \theta_{a1} \mathbf{k}_{a1} & \theta_{a2} \mathbf{k}_{a2} & \dots & \theta_{an} \mathbf{k}_{an} \end{bmatrix}$$

$$\mathbf{K}_b = ((\theta_b \mathbf{k}_b)_1, \dots, (\theta_b \mathbf{k}_b)_n) = \begin{bmatrix} \theta_{b1} \mathbf{k}_{b1} & \theta_{b2} \mathbf{k}_{b2} & \dots & \theta_{bn} \mathbf{k}_{bn} \end{bmatrix}$$

and $\text{tr}(\dots)$ is called the *trace-function* which returns the sum of the diagonal

elements of a matrix. Let such a matrix be

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

then the trace of \mathbf{A} is computed as

$$\text{tr}(\mathbf{A}) = a_{11} + a_{22} + a_{33}$$

Equation (5.56) can be solved by the maximization problem

$$\max_{\mathbf{R}} \text{tr}(\mathbf{R}\mathbf{H}) \quad (5.57)$$

where

$$\mathbf{H} = \mathbf{K}_b \mathbf{K}_a^T \quad (5.58)$$

Using the singular value decomposition of \mathbf{H} we can find the optimal solution \mathbf{R}

$$[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \text{svd}(\mathbf{H}) \quad (5.59)$$

$$\mathbf{R}_x = \mathbf{R} = \mathbf{V}\mathbf{S}\mathbf{U}^T \quad (5.60)$$

where

$$\mathbf{S} = \text{diag}[1, 1, \det(\mathbf{V}\mathbf{U}^T)] \quad (5.61)$$

is called the *Umeyama correction*, which ensures that \mathbf{R} is a rotation matrix.

When \mathbf{R}_x is found, the equation for the unknown translation \mathbf{t}_x can be solved

$$(\mathbf{R}_a - \mathbf{I})\mathbf{t}_x = \mathbf{R}_x \mathbf{t}_b - \mathbf{t}_a \quad (5.62)$$

The solution can be found as the usual least squares solution

$$\mathbf{t}_x = (\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T \mathbf{d} \quad (5.63)$$

where

$$\mathbf{C} = \begin{bmatrix} \mathbf{R}_{a1} - \mathbf{I} \\ \vdots \\ \mathbf{R}_{an} - \mathbf{I} \end{bmatrix} \quad \text{and} \quad \mathbf{d} = \begin{bmatrix} \mathbf{R}_x \mathbf{t}_{b1} - \mathbf{t}_{a1} \\ \vdots \\ \mathbf{R}_x \mathbf{t}_{bn} - \mathbf{t}_{an} \end{bmatrix} \quad (5.64)$$

5.4. Laser Triangulation with Calibrated System

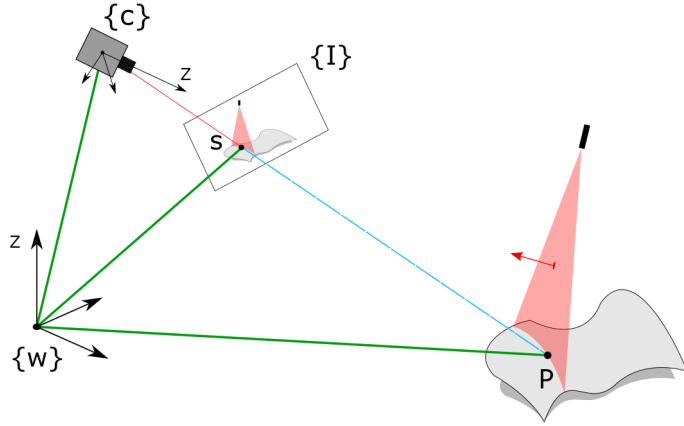


Figure 5.8.: Laser triangulation

In order to perform 3-D measurements with a laser vision system, a method of triangulation is needed. The method used in this thesis is influenced by the work done by Tingelstad and Egeland at the Department of Mechanical and Industrial Engineering NTNU [45]. The method is based on the assumption that calibration of the camera, laser and hand-eye calibration is already achieved.

When the parameters of the laser plane, the intrinsic and extrinsic parameters of the camera and the robot-camera transformations are known it is possible to reconstruct the 3-D point. Let an extracted image point of the laser line profile be $\tilde{\mathbf{p}} = [u, v, 1]^T$. The pixel coordinates are transformed into normalized image coordinates, using the intrinsic camera calibration matrix

$$\tilde{\mathbf{s}}^c = \mathbf{K}^{-1} \tilde{\mathbf{p}}$$

where the image coordinates are referenced in the camera frame $\tilde{\mathbf{s}}^c = [x/z, y/z, 1]^T$

The pinhole model does not take into account the lens distortion of the camera, and the normalized image coordinates needs to be undistorted before transforming them into another reference frame. This can be done using the OpenCV-library in Python if the distortion coefficients and the intrinsic matrix is known. The image coordinates referenced in the world frame can be found from the rigid

transformation between the world frame and the camera frame

$$\tilde{\mathbf{s}}^w = \mathbf{R}_c^w \tilde{\mathbf{s}}^c + \mathbf{t}_c^w \quad (5.65)$$

where $\tilde{\mathbf{s}}^c$ is now the undistorted image coordinates given in the camera frame. The normalized image coordinates can be used to find the direction of the line from the camera origin \mathbf{l}_0 , which is given in terms of the world frame, through the pixel in the image plane

$$\mathbf{l} = \tilde{\mathbf{s}}^w - \mathbf{l}_0 \quad (5.66)$$

where the line \mathbf{l} is also referred to as an *image ray*. The unit direction vector of \mathbf{l} is computed by normalizing \mathbf{l}

$$\hat{\mathbf{l}} = \frac{\mathbf{l}}{|\mathbf{l}|} \quad (5.67)$$

The distance from the camera center, along $\hat{\mathbf{l}}$ to the intersection point between the image ray and the laser plane can be computed from

$$d = \frac{(\mathbf{p}_0 - \mathbf{l}_0) \cdot \mathbf{n}}{\hat{\mathbf{l}} \cdot \mathbf{n}} \quad (5.68)$$

where \mathbf{p}_0 is a point in the laser plane given in world coordinates, and \mathbf{n} is the normal to the laser plane. The desired point $\mathbf{P} = [X, Y, Z]$ given in the world coordinate frame is then

$$\mathbf{P} = d\hat{\mathbf{l}} + \mathbf{l}_0 \quad (5.69)$$

Chapter 6.

Experimental setup

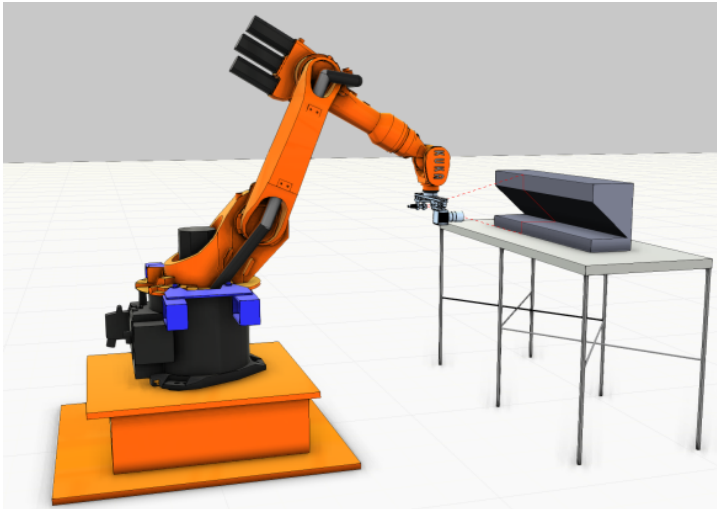


Figure 6.1.: Proposed test scenario

This chapter describes the experimental setup of the laser vision system with robot manipulator built in the lab at the Department of Mechanical and Industrial Engineering Valgrinda. Furthermore, the calibration process is also described and discussed, and at the end of the chapter some of the factors which had an impact on the accuracy of the system is presented. For a simplified schematic representation of the system architecture see [Appendix A.1](#).

6.1. System configuration

In order to perform scanning using a laser vision sensor, the scene and the sensor need to be moved relative each other. In most cases it is easier to move the sensor rather than the scene. To move the sensor, the system is mounted on the end effector of an industrial KUKA KR16-2 [1] robot in a hand-eye configuration, see Fig. 6.2. The mounting adapter is designed using CAD-software, based on machine drawings of the end effector flange, and made in the workshop at Valgrinda. The camera and laser are mounted on a Montech aluminum rail with clamp elements that can rotate and translate along the rail. This is done in order to be able to change the triangulation configuration if needed. The laser vision sensor is then mounted to the robot end effector via the mounting adapter.

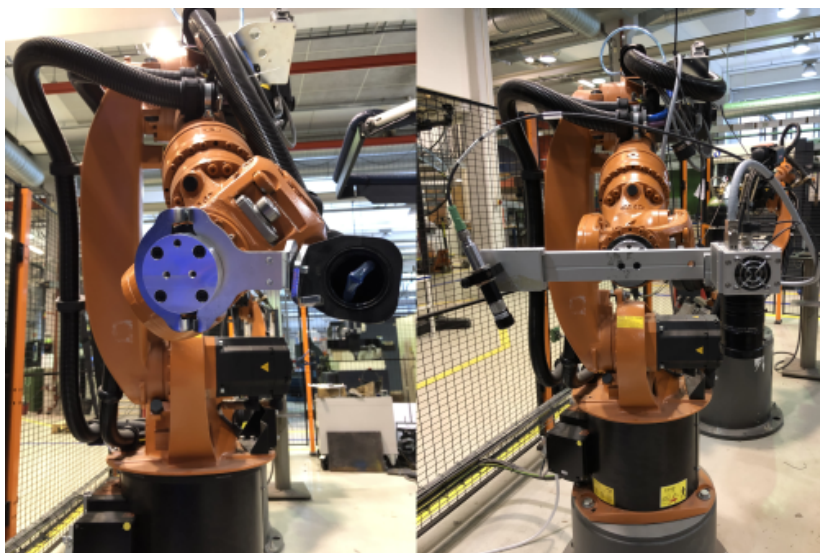


Figure 6.2.: Laser vision sensor mounted on the KUKA KR16-2 end effector

The system is calibrated at a work distance of approximately 500 mm, which is based on the assumption that this system is to be used on large multi-pass welding groves. The distance between the camera and the laser is measured with a tape measure to be approximately 325 mm, this gives the configuration a triangulation angle of approximately 49.5° .

The camera used in this project is the Automation Technology C4-2040 [14] image sensor, which is a fast CMOS sensor specialized towards laser vision. The camera has an integrated processor unit with internal functions for detecting laser lines

in the image plane up to 340 frames each second. The camera lens is a FUJINON CF16HA-1 [13] fixed focal lens with manual operated aperture and focus which is designed for industrial environments. The laser projector is a Z-LASER ZM18 [19] industrial diode line laser with Powell lens, red color light and 45° fan angle. The manufacturer of the laser recommend using max 80% of the total length of the laser line when performing measurements, this is the area where the lens ensures homogeneous distribution of light. At a work distance of 500 mm this is equivalent to a line length of approximately 331mm. The desired wavelength of the laser with respect to the maximum spectral response of the AT C4 camera is given in the specification of the camera, see Appendix 4.17. It shows that lasers with wavelength ranging between 580 nm - 670 nm yields the largest quantum efficiency. The ZM18 laser has a wavelength of 635 nm which is well within the acceptable range of the camera.

To control the robot, PyMoCo [27] is used which is a motion control framework in Python for industrial robots developed by Morten Lind at SINTEF Manufacturing, enabling real-time sensor based motion control at the application level. From PyMoCo the end effector frame can be computed with respect to the robot base and saved in numpy-arrays for later use or for real-time processing. The control and configuration of the AT-C4 camera requires Windows with a software development kit (SDK) and a standalone version of Python that is provided by Automation Technology. The parameters of the 3-D mode on the camera is configured using the Python SDK, with the help of calibration software developed by the manufacturer.

6.2. Calibration Process

The calibration target used in the calibration of the camera is a cardboard with a glued on 7×10 -checkerboard pattern with 20 mm squares, printed from a laser printer. During the camera calibration the chessboard is stationary while the camera is moved relative the target by the KUKA-robot. In order to obtain a good calibration the laser vision system requires between 15 to 20 different positions where the camera views the entire checkerboard. A position-check is done manually to ensure that both the calibration object is in focus and that the laser line can be viewed in the image, this is achieved by streaming images from the camera to Python. It is important that the laser line obstructs as little as possible of the square pattern in the image, otherwise it will interfere with the calibration of the camera. When the view of the target and the laser line is acceptable, an image of the calibration object as well as a pixel array of laser points positions

can be extracted from the camera. At the same time, the end effector position in reference to the robot base is registered and stored via PyMoCo. This process is repeated until enough positions are visited to ensure a good calibration. After the camera is calibrated, the laser plane can be estimated using RANSAC and a point cloud generated by triangulating the laser points with respect to the camera.

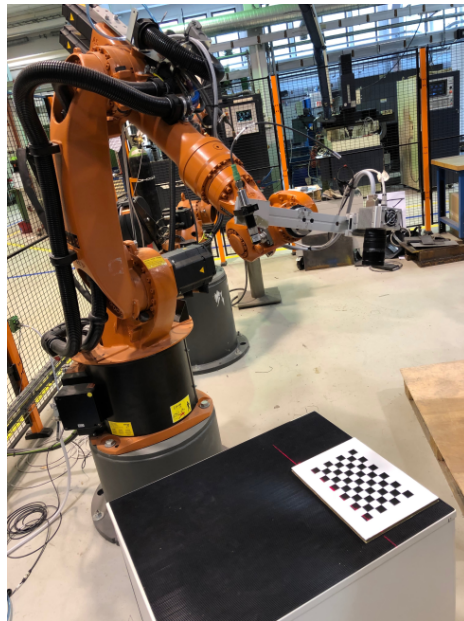


Figure 6.3.: Calibration in progress

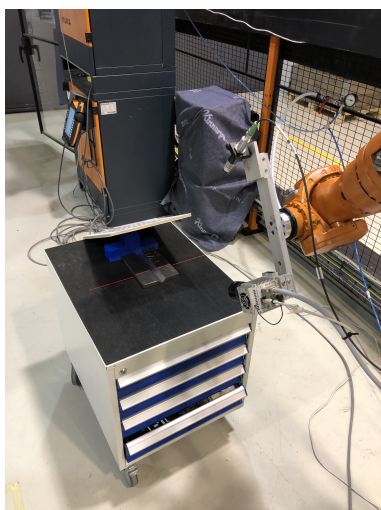
Before performing hand-eye calibration, pre-processing of the pose data is needed. Schmidt et.al [39, 38] presents a well suited method of selecting data for hand eye calibration, and the authors lists some important factors that needs to be considered when performing hand eye calibration. In this thesis the data from the camera- and end effector-poses are pre-processed where camera and end effector poses are permuted into different combinations with the inverse combinations excluded. This means that if the combination is (i, j) then its inverse (j, i) is not computed. This is done in order to mitigate singularities which might occur in the estimation process caused by inverse orientations. The method is called the *handshake principle* in Combinatorics and yields N unique combinations

$$N = \frac{n(n-1)}{2} \quad (6.1)$$

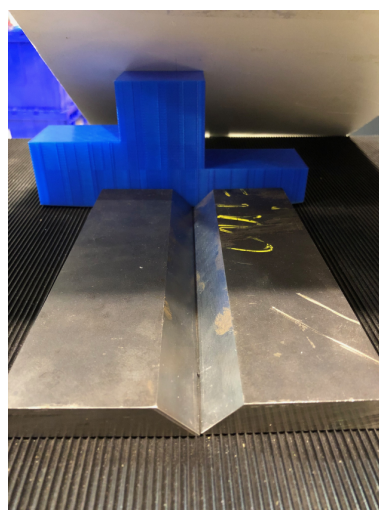
where n is the number of unique poses used in the estimation. As long as $N > 3$ the estimation should converge towards a solution.

6.3. Test Scanning of a Single-V Weld Grove

To analyze the performance of the system the laser vision sensor is tested on a single-v weld groove. Different configurations of the camera parameters was tried out, but the best suited configuration was found to having an AOI threshold of 120, aperture of F1.4 with an exposure time of $500 \mu\text{s}$. A total of 44 snap shots where taken of the weld joint surface where the camera is moved parallel to the ground with 5 mm increments between each snap shot. At each increment the laser line profile is saved as well as the position of the end effector, similar to the calibration process.



(a) System test



(b) V-weld test piece

Figure 6.4.: Test Scene

In order to mitigate the effect of the reflective surface of the weld groove and the overhead lighting from the lab, the camera views the scene at an angle and an improvised shade is applied, as seen in Fig. 6.4a. The laser points captured in the camera frame are related to the robot base by the hand eye transformation and the end effector to base transformations, thus building a point cloud.

6.4. System Accuracy

6.4.1. Internal Factors

The internal factors are defined as elements of the laser vision system that can have an impact on the accuracy the system. The re-projection errors tied to the intrinsic calibration of the camera, mechanical defects and defects in the camera lens and mounting rig can all be defined as intrinsic to the laser vision system. The compatibility of the components that are used in the laser triangulation configuration is also a contributing factor to the total accuracy of the system. This includes the sensor and resolution of the camera with respect to the use case, choice of lens, the compatibility between the laser and the camera, and the triangulation configuration itself.

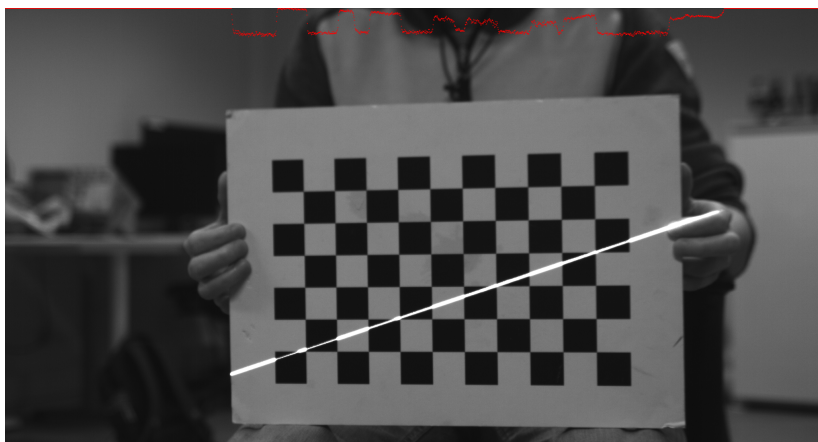


Figure 6.5.: Error caused by corrupted pixels

There have been reports [17] that the ideal working temperature of the AT-C4 camera is between 40°C - 45°C and that the integrated circuits of the camera are increasingly affected by noise with rising temperatures. This will have an effect on the image quality of the image sensor, and corrupted frames can occur if the image acquisition is started right after the camera has been turned on. This phenomena have indeed been present during the calibration of the camera which in some cases resulted in the need for re-calibration. A workaround have been implemented into the calibration where the camera image and the laser pixel coordinates are plotted on the same image in order to validate the pixel data. As seen in Fig. 6.5 the laser pixels from the COG-mode are plotted on the corresponding image as red points. If the laser points are properly extracted from the camera the points should lie

along the line, which in this instance is not the case because due to the data being corrupted.

6.4.2. External Factors

The external factors which affects the accuracy of the system, are environmental elements that are independent to the laser vision system. This includes the material and reflective properties of the surface that is scanned as well as light pollution caused by ambient lighting. The test piece used in this thesis can be viewed as an extreme test case because it has a very fine machined surface, which is not optimal for performing accurate measurements using structured light. In reality the weld groves on large offshore structures have a much more rough surface, which might make them better suited for measurements performed using structured light.

The effect of having a highly reflective surface combined with light pollution can be seen in Fig. 6.6, where the bottom part of the weld grove is lit up by the overhead lights in the robot cell. The reflective property, also known as *specular reflection*, of the scanned surface can be addressed by changing the inspection angle of the camera, or by applying a finish to the weld grove. When the light pollution is combined with the laser light it can saturate the image sensor and make it impossible to make accurate measurements. In order to prevent this from happening, ambient lighting should be kept at a minimum, and polarising filters and lens filters can be applied to counter reflection and to block out light at a certain wavelength. Modern image sensors also have adjustable parameters like exposure time and lens aperture which can limit the amount of light reaching the sensor, and thus preventing saturation. Care should be taken when restricting the amount of light reaching the sensor, so that the amount of data required to make a good measurement is still present. To further mitigate the effect of over-saturation, an image sensor with functions to increase the dynamic range of the sensor, also known as HDR, should be used. This allows for both strong and weak signals to be captured at the same time, which is very useful on objects with in-homogeneous surface structures or in situations where parts of the image-scene is saturated.

The error in robot positioning can also be viewed as an external factor since it is intrinsic to the robot and not the laser vision system. The manufacturers of KUKA KR16-2 is reports that the robot have a repeatability in positioning of approximately ± 0.05 mm. This is usually lower or on pair with the accuracy achieved with vision based sensors.

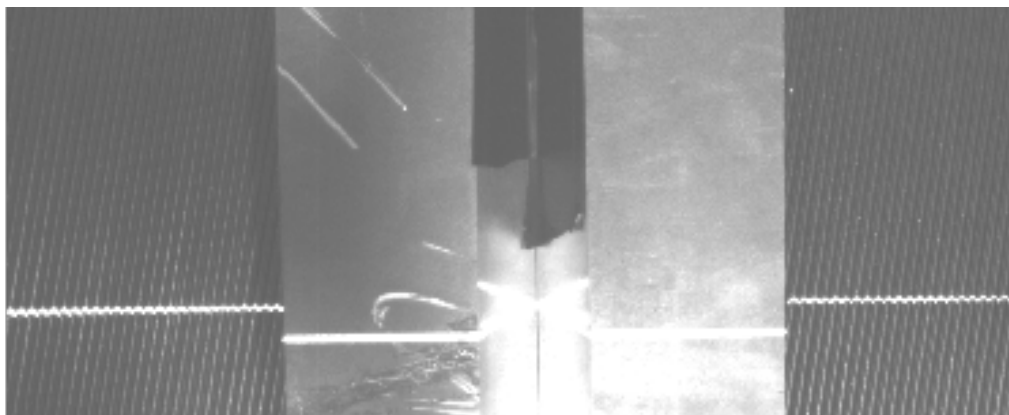


Figure 6.6.: Light pollution and specular reflection

A vision system requires the scene to be in focus in order to make accurate measurements. The scene's focus is tied to the depth of field of the camera which in turn is dependent on the work distance. Since the error in work distance and the depth of field both describes the allowed boundaries along the z -axis where the scene should be in focus, the accepted error in work distance can be computed as the depth of field. The work distance itself have a direct impact on the error of the measurements, which increases as the work distance increases. To achieve the best possible accuracy the work distance should therefore be as small as possible. In a laser vision sensor there will be a trade-off between having a minimum work distance and size of the area that is inspected. If the area is large, the work distance have to be set according to the field of view of the camera and the size of the laser projection. The size of the laser projection is also dependent on the fan angle of the laser, which means that if the work distance is set, while the area requires a larger laser projection to properly cover it, then a laser with a larger fan angle is required. Furthermore, the triangulation angle between the optical axis of the camera and the laser projection also have an impact on the work distance of the laser vision system, where a larger angle gives a shorter work distance and better accuracy.

6.4.3. Other factors

The calibration of the sensor systems have a significant effect on the total accuracy of the system and is often the main source explaining deviations in the final performance of the system.

During the hand eye estimation, the error in translation and rotation did not converge when data for pose 9 to 18 were input to the algorithm, as can be seen in Fig. 6.7. This might imply that some of the poses could be indexed wrongly during the process of storing the different poses, or when later re-uploading them in the calibration. This could mean that the object-to-camera pose is related to the wrong end-effector-to-base transformation causing the abnormality. Implementing better code, for calibration and for storing the poses is a potential solution. Instead of using MATLAB for estimating the intrinsic and extrinsic camera parameters, and Python for estimating the hand eye transformation, OpenCV could be used to both implement camera calibration and hand eye calibration in Python. This would remove the need for intermediary functions and help automate the process further.

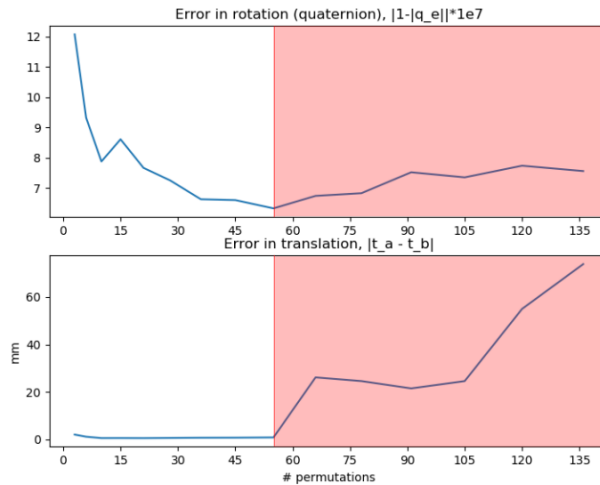


Figure 6.7.: Diverging error in hand eye estimation

Another potential source of the error is in the process of positioning the camera [38], which was done by manually jogging the robot end effector. If the distance especially in rotation, between the relative poses is not sufficiently large enough, large meaning $\Delta\theta \approx \pm 90^\circ$, this can result in a bad estimation because for angle close to zero the rotation axes are not well defined. Additionally, angles close to 180° can result in singularities in the hand-eye calibration and should be avoided. A solution would be to pre-plan the different orientations, by setting the range of the rotation magnitude so that the rotation is sufficiently large enough for the estimation to be good, but low enough so that singularities in the hand eye calibration don't occur. However, in order to accurately estimate the laser

plane the camera must detect both the calibration target and the laser line. This complicates the planning and are the main reason pre-planning was not developed in this thesis.

Chapter 7.

Results and Discussion

In this chapter the results from the calibration of the laser vision system, as well as the results from the scanning of the test piece is presented. The result from the calibration includes the intrinsic and extrinsic camera parameters, as well as the estimated hand eye transformation.

7.1. System calibration

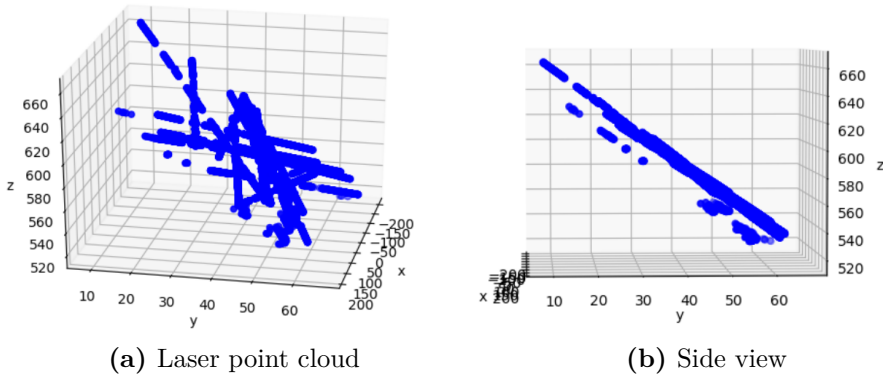
During the calibration process, the robot is moved to 20 different poses. At each pose an image of the calibration object and the laser line are both extracted from the camera via Python. The robot base to end effector transformation is also extracted using PyMoCo. The camera calibration is performed using the Camera Calibration Toolbox [5] in MATLAB where the calibration feature points were found on 18 of the 20 images. The estimated intrinsic parameters and distortion coefficients as well as the error in estimation are

Parameters	Estimation Error
$\alpha = 3026.4$	$\sigma_{\alpha} = 2.8419$
$\beta = 3029.8$	$\sigma_{\beta} = 2.7976$
$u_0 = 1039.8$	$\sigma_{u_0} = 0.8751$
$v_0 = 570.5$	$\sigma_{v_0} = 1.1340$
$k_1 = -0.0468$	$\sigma_{k_1} = 0.0028$
$k_2 = 0.3704$	$\sigma_{k_2} = 0.0324$

where all of the units are given in pixels and the estimation errors σ are given as standard error in pixels. The intrinsic parameters are estimated with an overall mean reprojection error

$$\mu = 0.1955 \quad (7.1)$$

which is given in pixels.



The laser plane is calibrated based on the intrinsic and extrinsic parameters found from the camera calibration as well as laser point images taken by the COG-mode of the camera. The extrinsic rotation and translation of the camera image plane relative to the calibration plane is given in 3-D vectors, where the rotation is given as the angle-axis representation. The laser points are triangulated based on input from the camera calibration matrix and the relative transformation between the image plane and the calibration target plane. A plot of the triangulated points can be seen in Fig. 7.1a. Using RANSAC the laser plane is estimated from the point cloud where the estimated plane coefficients are

$$A = -0.00665 \quad B = -0.09164 \quad C = -0.04003 \quad D = 271.33819$$

The plane normal are represented by A , B and C , while the parameter D represents the shortest distance along the plane normal between the laser plane and the origin of the camera. The measure of accuracy is the mean of the euclidean distances between each point along the plane normal

$$\mu = 0.17376 \text{ mm}$$

In order to evaluate the RANSAC estimation, a least square plane is also estimated from the same point cloud data. The estimated least square coefficients are

$$A = -0.00689 \quad B = -0.09170 \quad C = -0.03988 \quad D = 270.42456$$

with a mean error

$$\mu = 0.20620 \text{ mm}$$

The results shows that the difference in error ($\Delta\mu \approx 0.032 \text{ mm}$) is small. This is most likely due to the low amount of noise in the laser images, resulting in very low ratio of points deviating from the consensus along the plane normal, as seen in Fig.7.1b. In this case LS and RANSAC have similar performances as can be seen in Fig.7.2 where the green RANSAC plane and the blue LS plane are overlapping.

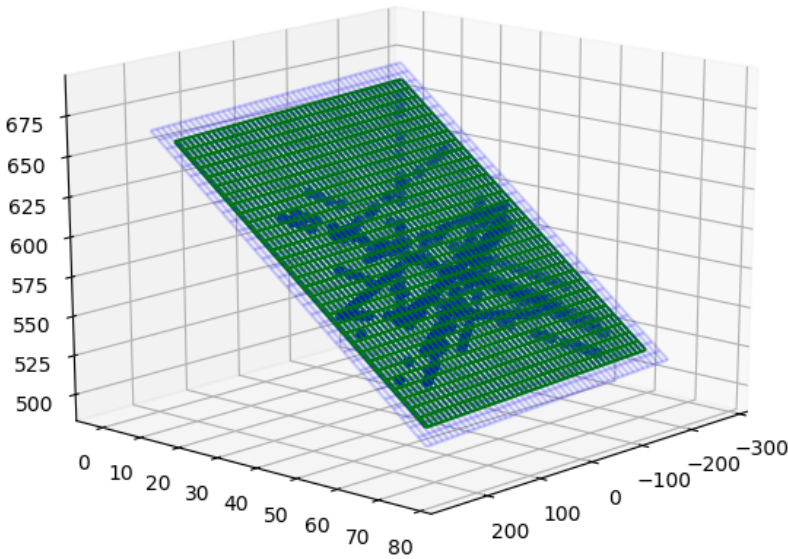


Figure 7.2.: Plot of RANSAC and least squares plane

The estimated hand eye transformation is

$$\mathbf{R}_X = \begin{bmatrix} -0.002155 & -0.003825 & 0.99999 \\ 0.000247 & -0.999993 & -0.003525 \\ 0.999998 & -0.000256 & 0.000715 \end{bmatrix}, \quad \mathbf{t}_X = \begin{bmatrix} 63.486 \\ 134.772 \\ 107.075 \end{bmatrix}$$

Due to erroneous data in poses from 9 to 18, the hand eye calibration where per-

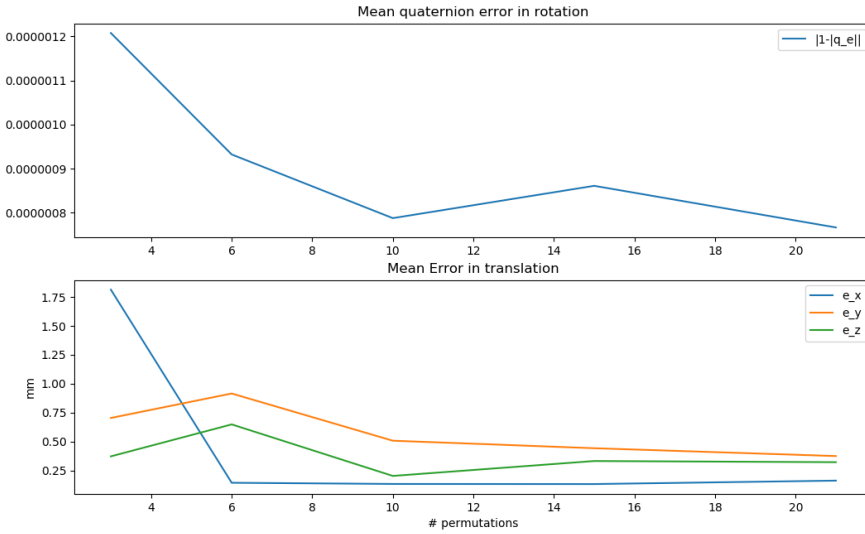


Figure 7.3.: Error in translation and rotation

formed using the first 8 poses. The solution are estimated from permutation computed based on the handshake principle previously described, which for 8 unique poses returns 21 different combinations. This should be enough combinations to achieve a good estimate for the hand eye transformation, which is indicated in Fig. 7.3 where the error converges toward sub-millimeter in translation

$$\mathbf{e}_t = [e_x, e_y, e_z] = [0.1610, 0.3738, 0.3207] \quad (7.2)$$

and a low quaternion error in rotation

$$|1 - |\mathbf{q}_e|| = 7.6641 \cdot 10^{-7} \quad (7.3)$$

The translation \mathbf{t}_X is in accordance with rough measurements between the end effector and the back of the camera on the actual setup

$$\mathbf{t}_{measured} = \begin{bmatrix} 22 \\ 138 \\ 102 \end{bmatrix} \quad (7.4)$$

The deviation in the x -direction and y -direction between the estimate and the measurement might be due to the actual origin of the camera frame being inside

of the camera, which makes it difficult to measure physically. Furthermore the origin might not be perfectly aligned at the physical center of the camera due to the calibration of the camera.

The estimated orientation \mathbf{R}_X is equivalent to a 180° counterclockwise rotation about the z -axis followed by a 90° clockwise rotation about the current y -axis

$$\mathbf{R}_X \approx \mathbf{R}_z(\pi)\mathbf{R}_y\left(-\frac{\pi}{2}\right) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (7.5)$$

The relative frame transformation is illustrated in Fig. 7.4

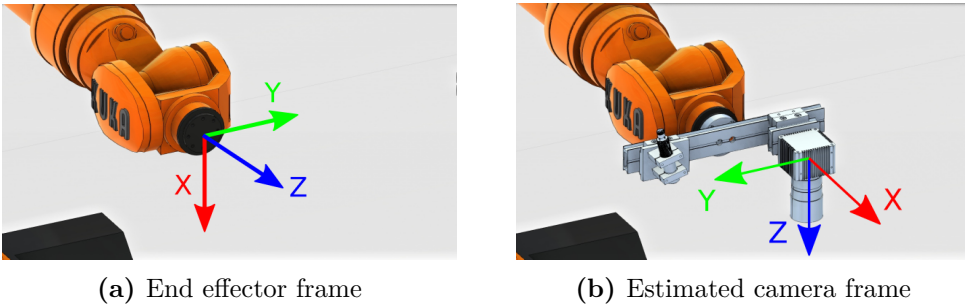


Figure 7.4.: Estimated orientation between end effector and camera

7.2. Results from scanning of test piece

The result of the test scan is a point cloud containing approximately 35200 points, which is roughly 800 points pr. snap taken. By inspecting the point cloud in Fig. 7.5 the effect of the overhead lighting can be seen as less points are detected on the front left side of the weld groove due to reflection and illumination, while more points are detected in the weld groove at the back of the image, where the shade is at its strongest, see Fig. 6.4b. This indicates the importance of having adequate lighting conditions in order to accurately extract information from highly reflective surfaces when using vision based sensing. An argument can be made that the configuration suffers from occlusion since the left side of the scan have almost no points visible in the point cloud. However, this is due to pre-processing of the laser points which is done with respect to the camera having a large field of view caused by the relatively large work distance. By cutting out part of the

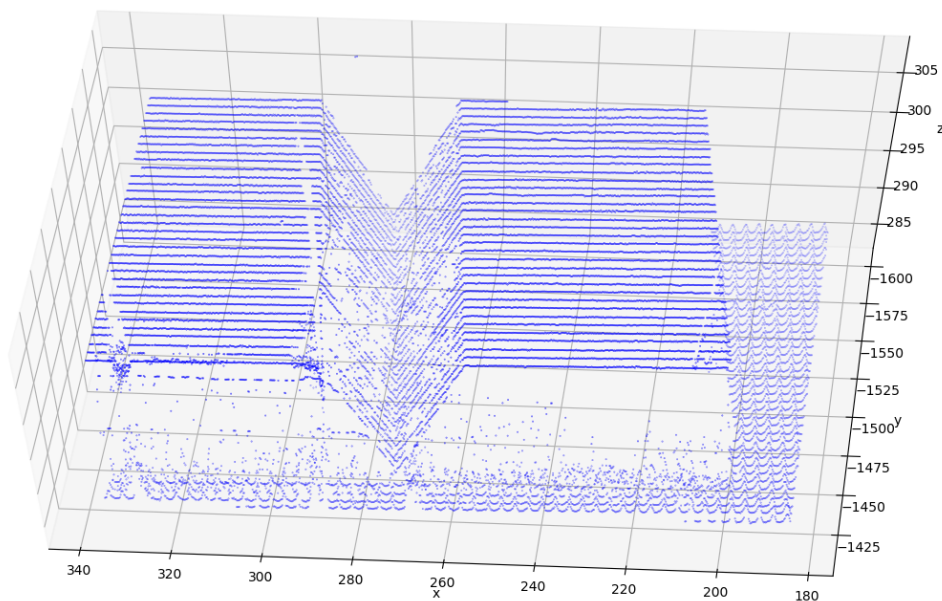
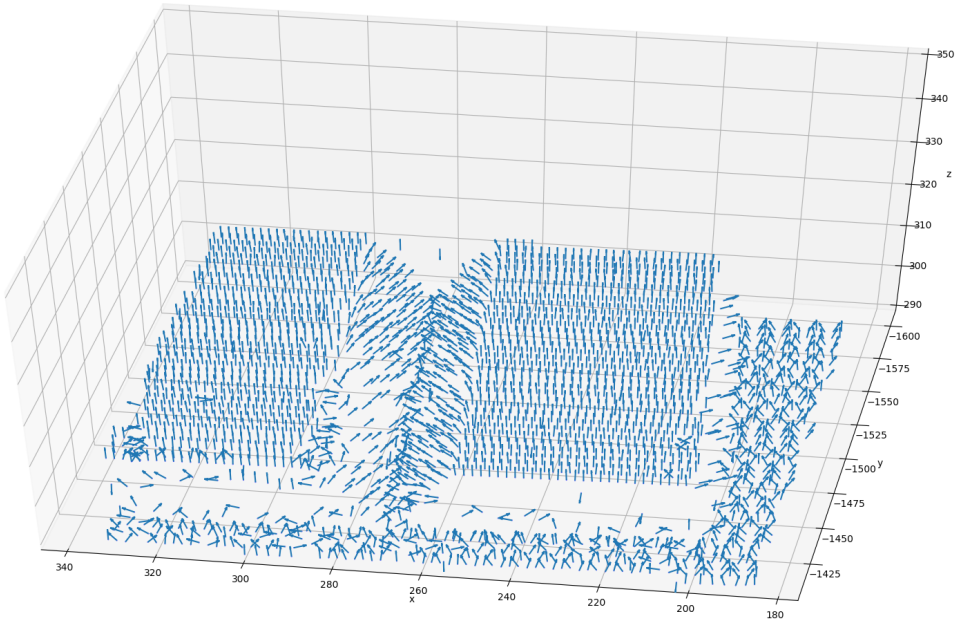


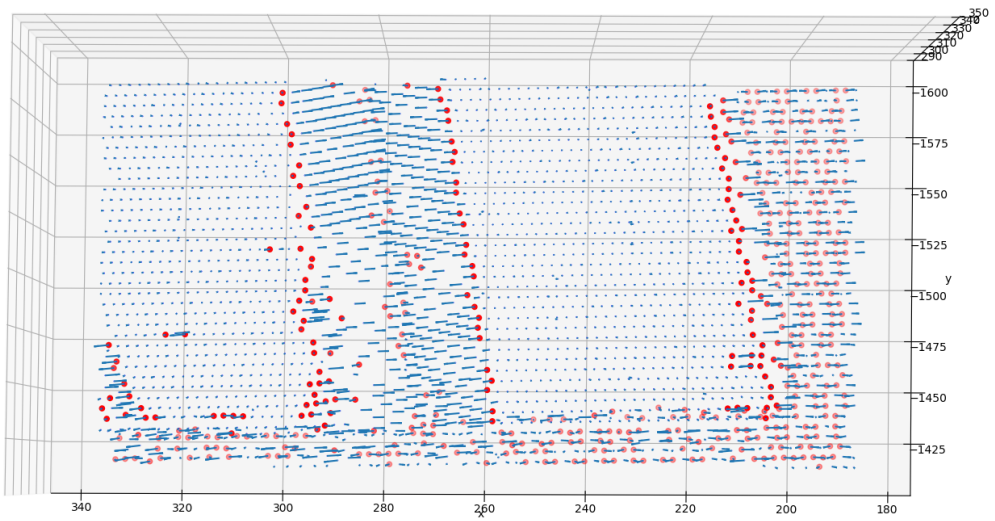
Figure 7.5.: Point Cloud

scene, unwanted information can be filtered away, thus increasing processing time. A caveat is that this should be done with caution when the system is used in real time scanning of larger complex-shaped surfaces, because it limits the information from the scene, and therefore might exclude important data. For the purposes of this test, which is a proof of concept performed on a relatively linear test piece with small dimensions, this can be allowed as long as enough information about the most critical part, that is the weld groove, is present.

The point cloud can be used to estimate information about the weld groove, which is used to parameterize the geometry for path planning. One way of doing this is to compute the cross product between each point and its neighbour, which will result in a vector perpendicular to the vector between the points. If the points lie in the same plane the cross product between neighbours should approximately be equal to the plane normal, this is illustrated in Fig. 7.6a where each normal represent the mean of a neighbourhood consisting of ten points. The edges of the weld groove can be detected as the points at which the normal between one neighbourhood largely deviates from the next neighbourhood normal, this is illustrated in Fig. 7.6b. This way the weld groove can be identified and parameterized similarly to what was shown in Fig. 2.2.



(a) Point-to-point normals



(b) Edge points

To validate the accuracy of the scanning, the thickness of the test piece is measured in the point cloud and compared to measured values of the actual test piece. The measurements of the actual test piece are performed manually at ten random selected points using a Mitutoyo caliper with standard uncertainty of 0.05 mm. The measurements are found in Appendix A.4 and the resulting mean is

$$\mu = 15.01 \text{ mm} \quad (7.6)$$

with a experimental standard deviation $s_X = 0.05/\sqrt{10} = 0.0158 \text{ mm}$.

The thickness of the point cloud is found by estimating the upper plane of the test piece and the bottom plane, that is the table that the test piece is resting on, see Fig. 7.7. These planes should be approximately parallel to each other, and at a relative distance that is equal to the thickness of the test piece. The plane estimation is achieved by applying RANSAC to the point cloud, estimate a plane, remove the inliers to that plane from the point cloud and then run the plane estimation again. This procedure will estimate the planes with the most and second most points, which in this case is the top of the table and the top surface of the test piece.

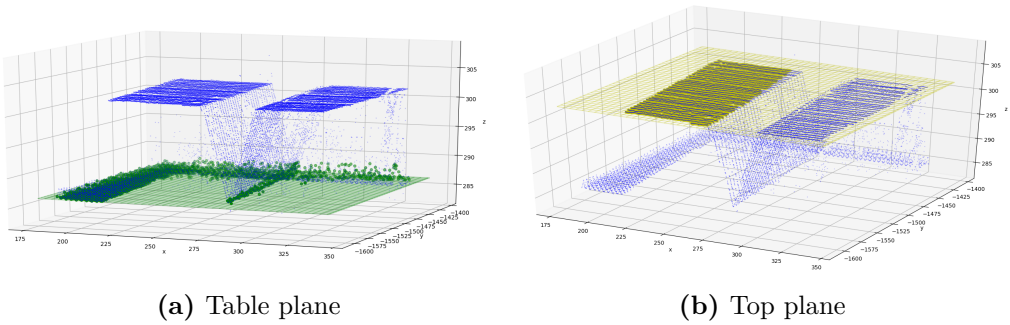


Figure 7.7.: Multiple plane segmentation

The planes are estimated to be

$$\boldsymbol{\pi}_{table} = [-0.0053, -0.0014, -1, 285.39] \quad (7.7)$$

$$\boldsymbol{\pi}_{top} = [-0.0045, -0.0016, -1, 301.10] \quad (7.8)$$

where it can be seen that both planes is approximately parallel to the xy -plane of the coordinate system. The distance between the two planes can be computed according to the point to plane distance Eq.(3.79), where the point \mathbf{p} is defined in the first plane and the centroid of second plane \mathbf{p}_0 is defined where the z -axis

of the coordinate system intersects the plane.

$$\mathbf{p} = [0, 0, 285.39] \quad (7.9)$$

$$\mathbf{p}_0 = [0, 0, 310.10] \quad (7.10)$$

$$\mathbf{n} = \mathbf{n}_{top} = [-0.0045, -0.0016, -1] \quad (7.11)$$

The plane to plane distance can then be computed as

$$\delta = \frac{\mathbf{n}^T(\mathbf{p} - \mathbf{p}_0)}{|\mathbf{n}|} = 15.71 \text{ mm} \quad (7.12)$$

Compared to the mean of the measurements done with the caliper this shows that the measurements from the laser vision sensor have a deviation of approximately 0.7 mm. A source of error in this estimation might be in the process of estimating the table plane. Since the table top has a rubbery v-groove surface, which is actually detected by the laser (Fig. 7.5), the table plane might be offset due to the unevenness. Another source of deviation is in the choice of working distance, which in this case is rather large given the dimensions of the test piece. The camera could be calibrated to a shorter distance, which would decrease the error in depth measurements. Nevertheless a sub-millimeter deviation might be considered to be adequate in welding processes, especially of large subsea structures with larger weld groove dimensions that typically operates with deviations on a millimeter-scale. Compared to similar processes, previously mentioned in this thesis, where deviations are reported to be < 0.5 mm, the sensor is close to this range and can possibly obtain similar results given a better calibration and configuration of the sensor.

Chapter 8.

Conclusion and Future Work

The focus of the thesis have been on building a laser vision sensor and identifying important system parameters of laser vision systems that is used in robotic welding of large subsea structures. A final test have been performed, where the system is first calibrated with respect to hand-eye, intrinsic calibration and laser plane calibration, and then used to inspect a weld groove in order to build a 3-D representation that can be used to parameterize the weld groove. The calibration, configuration and system design is of critical importance when performing measurements using structured light and vision based sensors. The main source of error in system calibration is usually tied to the process of acquiring and pre-process of data used in the calibration. The triangulation setup of the sensor is highly dependent on the use case, and knowledge about the scene and environment is key to implement the correct configuration.

The intrinsic parameters of the camera was estimated with a mean reprojection error of 0.1955 pixels based data from 18 pictures of a checkerboard. The laser plane was estimated using RANSAC with a resulting mean error of 0.17376 mm and the hand-eye calibration where calibrated with an error in translation $\mathbf{e}_t = [0.1610, 0.3738, 0.3207]$ and a quaternion error in rotation $|1 - |\mathbf{q}_e|| = 7.6641 \cdot 10^{-7}$. The final test shows that the system performs adequately, with deviation in measurements around 0.7 mm. However, this is still somewhat larger compared to previously achieved results from similar processes using visual sensors [36, 49, 26] where the reported mean deviation is as low as < 0.5 mm.

This is most likely due to the calibration of the system as well as the system being vulnerable to external factors like ambient lighting and specular reflection. Steps can be taken during the design and configuration of the system to mitigate errors from such factors, for example by changing the triangulation configuration, reduce

the working distance or augment the scene by applying shade to block out the ambient light. Given that the sensor is calibrated and configured correctly to the specific use case, achieving higher accuracy should be possible with the proposed setup.

8.1. Recommendations for future work

The proposed method of calibration and scanning is quite tedious and requires multiple systems in order to properly function, never mind the process of intermediate storing information which might lead to an erroneous calibration. A system where both the camera and robot is controlled by the same operating system and API is potentially a huge advantage because it would be possible to further automate the process and to achieve real-time performance. There exist libraries and software for GeniCam based cameras which operates on both Linux and Windows with the Python API, which should be further investigated. One of them are Common Vision Blox (CVB) [16], which is a powerful vision software for implementing imaging and machine vision applications developed by Stemmer Imaging. Unfortunately it is not open source and requires a license, furthermore as of today the Linux compatibility is limited and doesn't have the full range of tools that CVB offers. Another alternative would be Aravis [4] which is an open source library for video acquisition using cameras with the Genicam standard that is compatible with Python and Linux. Besides the camera software, the robot software could also be changed or altered to fit the needs of this project. It should be possible to run PyMoCo on Windows, however this would require some alterations to the software. Another solution, which also would need some alterations to be able to function with the Python API would be to use *KukaVarProxy* [20, 37] which is a software application installed in the KR C4 robot controller. *KukaVarProxy* is capable of, among other things, reading and writing variables that can be used to control the KUKA robot.

Future work would also include applying multiple image sensors, for example by applying stereo vision in the laser vision sensor, see Fig. 8.1. The advantage of using multiple cameras to view the same scene from different angles would potentially result in a more accurate sensor since the additional camera introduces redundancy in the data through the epipolar geometry of the stereo vision configuration. Also because using one image sensor have been shown to suffer from the effects of ambient light, specular reflection and occlusion, a second sensor would possibly mitigate many of these effects due to the changed viewing angle. Furthermore, the cameras could also be configured differently with respect to parameters

such as aperture, exposure time, threshold values and HDR thus giving the laser vision sensor a broader sensing range.

A robust method for parameterizing the scanned surface for path planning should also be investigated. A more elegant edge detection of the test piece than what was done in this thesis would be to perform multiple plane segmentation on the point cloud, and compute the line of intersection between the different planes. This method would also be more robust to edges where the points are not well defined. However when scanning complex geometries this may not be adequate or even possible. A more general approach should be investigated using other edge detection methods or by applying neural networks to increase the generality of the system.

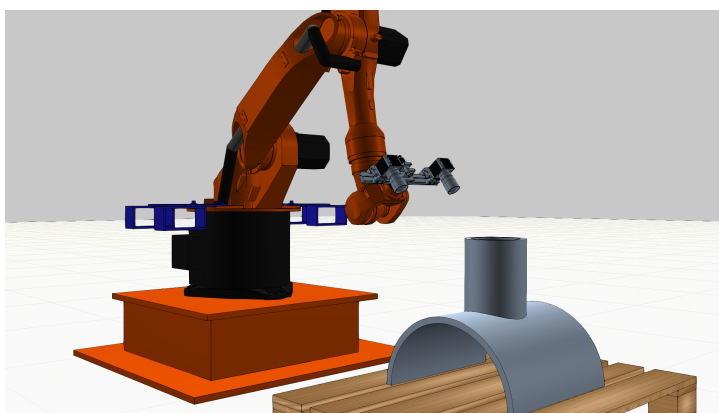


Figure 8.1.: Stereo vision scenario

8.1.1. Python Code

Some of the code snippets developed during this project are included as a .zip attachment to this thesis. A git repository has also been made and can be found at <https://github.com/trymdh/tpk4940Master>. As a disclaimer, the intention was to not include the code because it is still a work in progress, and the required dependencies are not well documented. However, since a lot of time and effort have been spent on developing this code it is provided as an attachment. As of this moment the camera functions are able to control and to configure the camera and it should be possible to use this as a basis for further development of the camera software. The repository also includes some utility functions as well as code for calibrating the laser vision sensor based on the methods described in this thesis.

References

- [1] KUKA AG. *KUKA KR16-2 Manual*. URL: <https://www.kuka.com/en-sg/services/downloads>. (accessed: 05.01.2019).
- [2] S. M. Ahmed, J. Yuan, Y. Wu, C. M. Chew, and C. K. Pang. “Collision-free path planning for multi-pass robotic welding”. In: *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*. Sept. 2015, pp. 1–4. DOI: [10.1109/ETFA.2015.7301594](https://doi.org/10.1109/ETFA.2015.7301594).
- [3] Alex M. Andrew. “MULTIPLE VIEW GEOMETRY IN COMPUTER VISION, by Richard Hartley and Andrew Zisserman, Cambridge University Press, Cambridge, 2000, xvi607pp., ISBN 0521623049 (hardback, 60.00).” eng. In: *Robotica* 19.2 (2001), pp. 233–236. ISSN: 0263-5747.
- [4] AravisProject. *Aravis GeniCam*. URL: <https://github.com/AravisProject/aravis>. (accessed: 10.05.2019).
- [5] Jean-Yves Bouguet. *Camera Calibration Toolbox for Matlab*. URL: http://www.vision.caltech.edu/bouguetj/calib_doc/. (accessed: 01.01.2019).
- [6] Visual Components. *Visual Components 4.0 Premium*. URL: <https://www.visualcomponents.com/>. (accessed: 01.01.2019).
- [7] K. Daniilidis and E. Bayro-Corrochano. “The dual quaternion approach to hand-eye calibration”. In: *Proceedings of 13th International Conference on Pattern Recognition*. Vol. 1. Aug. 1996, 318–322 vol.1. DOI: [10.1109/ICPR.1996.546041](https://doi.org/10.1109/ICPR.1996.546041).
- [8] M. Dinham and G. Fang. “Weld seam detection using computer vision for robotic Arc Welding”. In: *2012 IEEE International Conference on Automation Science and Engineering (CASE)*. Aug. 2012, pp. 771–776. DOI: [10.1109/CoASE.2012.6386339](https://doi.org/10.1109/CoASE.2012.6386339).
- [9] Professor Olav Egeland. *A note on robot kinematics*. Tech. rep. NTNU, Department of Mechanical and Industrial Engineering, Nov. 2017.
- [10] Professor Olav Egeland. *A note on vision*. Tech. rep. NTNU, Department of Mechanical and Industrial Engineering, Nov. 2017.

- [11] Z. Fang, D. Xu, and M. Tan. “A Vision-Based Self-Tuning Fuzzy Controller for Fillet Weld Seam Tracking”. In: *IEEE/ASME Transactions on Mechatronics* 16.3 (June 2011), pp. 540–550. ISSN: 1083-4435. DOI: [10.1109/TMECH.2010.2045766](https://doi.org/10.1109/TMECH.2010.2045766).
- [12] Martin A Fischler and Robert C Bolles. “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. In: *Communications of the ACM* 24.6 (1981), pp. 381–395.
- [13] Fujifilm. *CF16HA-1 Lens specifications*. URL: https://www.fujifilmusa.com/products/optical_devices/machine-vision/1-15-mp/cf16ha-1/index.html (visited on 02/03/2019).
- [14] Automation Technology GmbH. *AT C4-2040 Hardware Reference Manual*. URL: <https://www.stemmer-imaging.com/media/uploads/cameras/at/12/122200-Automation-Technology-User-Manual-C4-2040.pdf>. (accessed: 01.01.2019).
- [15] Du Q Huynh. “Metrics for 3D rotations: Comparison and analysis”. In: *Journal of Mathematical Imaging and Vision* 35.2 (2009), pp. 155–164.
- [16] Stemmer Imaging. *Common Vision Blox*. URL: <https://www.commonvisionblox.com/en/common-vision-blox-powerful-fast-modular/>. (accessed: 01.01.2019).
- [17] Stemmer Imaging. *Integrating an Automation Technology 3D camera*. URL: <https://www.stemmer-imaging.com/media/uploads/cameras/at/12/122206-Stemmer-Imaging-Integration-Guide-for-Automation-Technology-CX-Series-devices.pdf>. (accessed: 06.04.2019).
- [18] Stemmer Imaging. *Stemmer Imaging and Vision Handbook 2016-2017*. Tech. rep. Jan. 2019.
- [19] Stemmer Imaging. *Z-Laser ZM18 series*. URL: <https://www.stemmer-imaging.com/media/uploads/websites/documents/products/illumination/Z-Laser/en-Z-Laser-ZM18-BZLAS3-201402.pdf>. (accessed: 10.01.2019).
- [20] IMTS. *KukavarProxy*. URL: <https://github.com/ImtsSrl/KUKAVARPROXY>. (accessed: 28.05.2019).
- [21] Inrotech. *Inrotech Adaptive Multipass Planning Technology*. URL: <http://www.inrotech.com/cases>. (accessed: 06.01.2019).
- [22] National Instruments. *Image Acquisition System Setup*. URL: <http://www.ni.com/product-documentation/4229/en/> (visited on 02/01/2019).

- [23] V. Klema and A. Laub. “The singular value decomposition: Its computation and some applications”. In: *IEEE Transactions on Automatic Control* 25.2 (Apr. 1980), pp. 164–176. ISSN: 0018-9286. DOI: [10.1109/TAC.1980.1102314](https://doi.org/10.1109/TAC.1980.1102314).
- [24] James J Kuffner. “Effective sampling and distance metrics for 3D rigid body path planning”. In: *ICRA*. Citeseer. 2004, pp. 3993–3998.
- [25] Kvaerner. *Another delivery of platform jackets for Johan Sverdrup*. URL: <https://www.kvaerner.com/toolsmenu/Media/Press-releases/2018/Another-delivery-of-platform-jackets-for-Johan-Sverdrup-/>. (accessed: 06.04.2019).
- [26] Y. Li, Y. F. Li, Q. L. Wang, D. Xu, and M. Tan. “Measurement and Defect Detection of the Weld Bead Based on Online Vision Inspection”. In: *IEEE Transactions on Instrumentation and Measurement* 59.7 (July 2010), pp. 1841–1849. ISSN: 0018-9456. DOI: [10.1109/TIM.2009.2028222](https://doi.org/10.1109/TIM.2009.2028222).
- [27] Morten Lind, Johannes Schrimpf, and Morten Lind. “6.7 Real-Time Robot Trajectory Generation with Python”. In: *Sensor-based Real-time Control of Industrial Robots* (2013), p. 129.
- [28] Y. Liu, W. Zhang, and Y. Zhang. “Dynamic Neuro-Fuzzy-Based Human Intelligence Modeling and Control in GTAW”. In: *IEEE Transactions on Automation Science and Engineering* 12.1 (Jan. 2015), pp. 324–335. ISSN: 1545-5955. DOI: [10.1109/TASE.2013.2279157](https://doi.org/10.1109/TASE.2013.2279157).
- [29] Yan Liu, Jiang Liu, and Xincheng Tian. “An approach to the path planning of intersecting pipes weld seam with the welding robot based on non-ideal models”. In: *Robotics and Computer-Integrated Manufacturing* 55 (2019), pp. 96–108. ISSN: 0736-5845. DOI: <https://doi.org/10.1016/j.rcim.2018.07.010>. URL: <http://www.sciencedirect.com/science/article/pii/S0736584518300280>.
- [30] LMI. *LMI GOCATOR 2800*. URL: <https://www.stemmer-imaging.com/en/products/series/lmi-gocator-2800-series/>. (accessed: 06.01.2019).
- [31] Mathworks. *Camera calibration*. URL: <https://www.mathworks.com/help/vision/ug/camera-calibration.html>. (accessed: 04.04.2019).
- [32] Bob Mellish. *Pin Hole camera*. URL: https://en.wikipedia.org/wiki/Pinhole_camera_model#/media/File:Pinhole-camera.svg. (accessed: 10.01.2019).
- [33] F. C. Park and B. J. Martin. “Robot sensor calibration: solving $AX=XB$ on the Euclidean group”. In: *IEEE Transactions on Robotics and Automation* 10.5 (Oct. 1994), pp. 717–721. ISSN: 1042-296X. DOI: [10.1109/70.326576](https://doi.org/10.1109/70.326576).

- [34] S. Remy, M. Dhome, J. M. Lavest, and N. Daucher. “Hand-eye calibration”. In: *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97*. Vol. 2. Sept. 1997, 1057–1065 vol.2. DOI: [10.1109/IROS.1997.655141](https://doi.org/10.1109/IROS.1997.655141).
- [35] E Richter. “Dimensional tolerances of welded products”. In: *Welding International* 1.1 (1987), pp. 58–61. DOI: [10.1080/09507118709449026](https://doi.org/10.1080/09507118709449026). eprint: <https://doi.org/10.1080/09507118709449026>. URL: <https://doi.org/10.1080/09507118709449026>.
- [36] A. Ryberg, M. Ericsson, A. -. Christiansson, K. Eriksson, J. Nilsson, and M. Larsson. “Stereo vision for path correction in off-line programmed robot welding”. In: *2010 IEEE International Conference on Industrial Technology*. Mar. 2010, pp. 1700–1705. DOI: [10.1109/ICIT.2010.5472442](https://doi.org/10.1109/ICIT.2010.5472442).
- [37] F. Sanfilippo, L. I. Hatledal, H. Zhang, M. Fago, and K. Y. Pettersen. “JOpenShowVar: An open-source cross-platform communication interface to Kuka robots”. In: *2014 IEEE International Conference on Information and Automation (ICIA)*. July 2014, pp. 1154–1159. DOI: [10.1109/ICInfA.2014.6932823](https://doi.org/10.1109/ICInfA.2014.6932823).
- [38] Jochen Schmidt and Heinrich Niemann. “Data selection for hand-eye calibration: a vector quantization approach”. In: *The International Journal of Robotics Research* 27.9 (2008), pp. 1027–1053.
- [39] Jochen Schmidt, Florian Vogt, and Heinrich Niemann. “Vector Quantization Based Data Selection for Hand-Eye Calibration.” In: *VMV*. 2004, pp. 21–28.
- [40] SERVO-ROBOT. *i-CUBE arc-seam-finding*. URL: <https://servo-robot.com/arc-seam-finding/>. (accessed: 06.01.2019).
- [41] Bruno Siciliano. *Robotics : Modelling, Planning and Control*. eng. 2009.
- [42] K. H. Strobl and G. Hirzinger. “Optimal Hand-Eye Calibration”. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Oct. 2006, pp. 4647–4653. DOI: [10.1109/IROS.2006.282250](https://doi.org/10.1109/IROS.2006.282250).
- [43] Automation Technology. *C5 Series*. URL: <https://www.automationtechnology.de/cms/en/c5-series-compact-3d-sensors-with-ultra-hd-resolution/>. (accessed: 06.01.2019).
- [44] Automation Technology. *C5 Series Specification*. URL: https://www.automationtechnology.de/cms/wp-content/uploads/2017/06/C5-Series_web.pdf. (accessed: 06.01.2019).

- [45] Lars Tingelstad and Olav Egeland. “Robotic Assembly of Aircraft Engine Components Using a Closed-loop Alignment Process”. In: *Procedia CIRP* 23 (2014). 5th CATS 2014 - CIRP Conference on Assembly Technologies and Systems, pp. 110–115. ISSN: 2212-8271. DOI: <https://doi.org/10.1016/j.procir.2014.10.070>. URL: <http://www.sciencedirect.com/science/article/pii/S2212827114011275>.
- [46] Kenneth E Torrance and Ephraim M Sparrow. “Theory for off-specular reflection from roughened surfaces”. In: *Josa* 57.9 (1967), pp. 1105–1114.
- [47] Y. Wu, J. Z. M. Go, S. M. Ahmed, W. Lu, C. Chew, and C. K. Pang. “Automated bead layout methodology for robotic multi-pass welding”. In: *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*. Sept. 2015, pp. 1–4. DOI: [10.1109/ETFA.2015.7301590](https://doi.org/10.1109/ETFA.2015.7301590).
- [48] De Xu, Min Tan, Xiaoguang Zhao, and Zhiguo Tu. “Seam tracking and visual control for robotic arc welding based on structured light stereovision”. In: *International Journal of Automation and Computing* 1.1 (Oct. 2004), pp. 63–75. ISSN: 1751-8520. DOI: [10.1007/s11633-004-0063-0](https://doi.org/10.1007/s11633-004-0063-0). URL: <https://doi.org/10.1007/s11633-004-0063-0>.
- [49] Chenghao Zhang, Haichao Li, Zeshi Jin, and Hongming Gao. “Seam sensing of multi-layer and multi-pass welding based on grid structured laser”. In: *The International Journal of Advanced Manufacturing Technology* 91.1 (July 2017), pp. 1103–1110. ISSN: 1433-3015. DOI: [10.1007/s00170-016-9733-7](https://doi.org/10.1007/s00170-016-9733-7). URL: <https://doi.org/10.1007/s00170-016-9733-7>.
- [50] Zhengyou Zhang. “A Flexible New Technique for Camera Calibration”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (Dec. 2000). MSR-TR-98-71, Updated March 25, 1999, pp. 1330–1334. URL: <https://www.microsoft.com/en-us/research/publication/a-flexible-new-technique-for-camera-calibration/>.
- [51] Fuqiang Zhou and Guangjun Zhang. “Complete calibration of a structured light stripe vision sensor through planar target of unknown orientations”. In: *Image and Vision Computing* 23.1 (2005), pp. 59–67. ISSN: 0262-8856. DOI: <https://doi.org/10.1016/j.imavis.2004.07.006>. URL: <http://www.sciencedirect.com/science/article/pii/S026288560400188X>.

Appendix A.

Appendix

A.1. Technical specifications

Parameters	Specifications
Sensitivity	16000 LSB / μJ / cm^2 @ 550nm
Max. Spectral Response	0.290 A / (Wnm) @ 665nm
Resolution	2048 \times 1088 pixels
Pixel Size, ρ	5.5 μm
Sensor Size, $W \times H$	11.264mm \times 5.98mm, diag: 14.638mm
Sensor Format	2/3 inches
Sensor ADC Resolution	10 bit
Sensor Dynamic Range	90 dB with HDR
Max. Internal Full-Frame Rate	340 fps
Max. External Full-Frame Rate	50 fps

Table A.1.: AT C4-2050 specifications [18]

A.2. Tables

Pose nr.	v_1	v_2	v_3	σ_{v_1}	σ_{v_2}	σ_{v_3}
1	-0.0485	0.1148	0.0658	0.0007213	0.00060727	8.1658e-05
2	-0.0623	0.1386	0.0657	0.00071061	0.00062202	7.9519e-05
3	-0.0882	0.1801	0.0650	0.00074317	0.00057405	8.5522e-05
4	-0.0724	0.1604	0.0639	0.00069307	0.00059336	8.1253e-05
5	-0.1712	0.0426	0.1112	0.00065501	0.00058764	8.8327e-05
6	-0.2070	0.0988	0.1064	0.00074328	0.0006172	8.2053e-05
7	-0.2207	0.0689	-0.1599	0.00058936	0.00051624	9.0865e-05
8	-0.2207	0.0678	-0.1598	0.00066622	0.00053364	9.1294e-05
9	-0.3083	0.2367	-0.1740	0.00050952	0.00045176	0.00010749
10	0.2800	0.1941	0.2718	0.00043612	0.00043872	0.00010179
11	0.2819	0.1928	0.2723	0.00050531	0.0004417	9.7382e-05
12	0.3777	-0.2587	0.1138	0.0005002	0.00043207	9.6404e-05
13	0.2898	-0.2892	0.1704	0.00056899	0.00047737	9.5941e-05
14	0.2938	-0.2968	0.1698	0.00059774	0.00049893	9.9365e-05
15	0.2888	-0.2840	-0.0341	0.0006067	0.00050238	9.4322e-05
16	0.2953	-0.2380	-0.0072	0.00052885	0.00044849	9.0203e-05
17	0.2841	-0.2488	-0.0108	0.000573	0.00049138	9.2227e-05
18	0.2797	-0.0820	0.0393	0.00058087	0.00046159	7.7577e-05

Table A.2.: Rotation vectors and standard error, **NOTE:** $|v| = [v_1, v_2, v_3] \neq 1$

Pose nr.	x	y	z	σ_x	σ_y	σ_z
1	-117.7312	-89.0088	592.9758	0.1688	0.2211	0.5448
2	-121.3699	-48.8360	596.9720	0.1691	0.2225	0.5415
3	-75.5357	-66.2965	587.2141	0.1660	0.2182	0.5224
4	-112.0050	-47.5979	591.4106	0.1672	0.2203	0.5314
5	-148.2737	-88.4358	620.4198	0.1781	0.2309	0.5725
6	-136.0338	-26.5030	635.7404	0.1805	0.2370	0.5710
7	-141.4928	-56.6046	607.9349	0.1738	0.2264	0.5573
8	-54.0100	-63.1582	614.9710	0.1760	0.2280	0.5517
9	-149.0814	-12.2305	652.8577	0.1852	0.2437	0.5732
10	-151.1517	-74.6119	59.3849	0.1605	0.2114	0.5272
11	-96.0257	-70.2350	558.8427	0.1598	0.2103	0.5181
12	-58.6775	-74.6965	520.1076	0.1508	0.1961	0.5117
13	-113.4978	-64.3221	518.2079	0.1516	0.1945	0.5097
14	-130.2513	-66.2855	534.5669	0.1569	0.2004	0.5275
15	-138.7720	-57.2738	511.5158	0.1497	0.1917	0.5032
16	-89.3682	-49.0878	505.4651	0.1471	0.1900	0.4921
17	-97.2930	-60.6064	532.9273	0.1551	0.2003	0.5192
18	-45.0161	-60.7441	520.8133	0.1525	0.1964	0.4964

Table A.3.: Translation vectors and standard error given in millimeters

Measurements (mm), X

15.01
15.00
15.05
15.01
15.01
15.00
15.01
15.00
15.01
15.00

Table A.4.: Caliper measurements of test piece

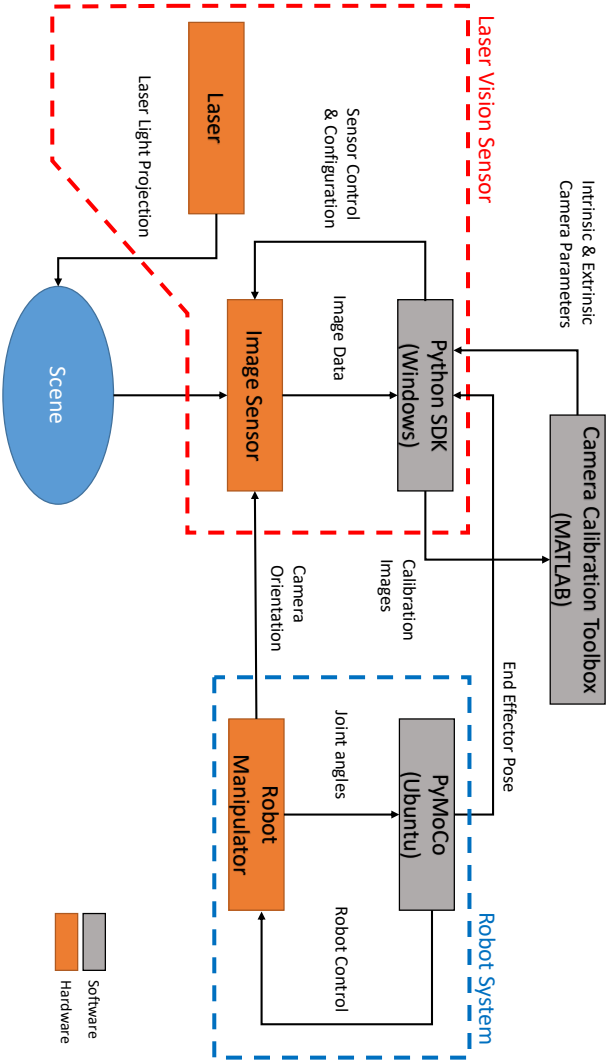


Figure A.1.: System Architecture

