Master Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering

Part localization for robotic manipulation

César Sinchiguano

Supervisor: Dr Gaël Pierre Écorchard.
May 2019

Acknowledgements

I would like to express my sincere gratitude to .....

Declaration

I hereby declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with methodical instructions for observing the ethical principles in the preparation of university theses. Prague, . May 2019

Abstract

The new generation of the collaborative robots allows the use of small robot arms working with human workers, e.g. the YuMi robot, a dual 7-DOF robot arms designed for precise manipulation of small objects. For the further acceptance of such a robot in the industry, some methods and sensors systems have to be developed to allow them to perform a task such as grasping a specific object. If the robot wants to grasp an object, it has to localize the object relative to itself. This is a task of object recognition in computer vision, the art of localizing predefined objects in image sensor data. This master thesis presents a pipeline for object recognition of a single isolated model in point cloud. The system uses point cloud data rendered from a 3D CAD model and describes its characteristics using local feature descriptors. These are then matched with the descriptors of the point cloud data from the scene to find the 6-DoF pose of the model in the robot coordinate frame. This initial pose estimation is then refined by a registration method such as ICP. A robot-camera calibration is performed also. The contributions of this thesis are as follows: The system uses FPFH (Fast Point Feature Histogram) for describing the local region and a hypothesize-and-test paradigm, e.g. RANSAC in the matching process. In contrast to several approaches those whose rely on Point Pair Features as feature descriptors and a geometry hashing, e.g. voting-scheme as the matching process.

Keywords: Object Detection, Pose Estimation, Robotics, Point Cloud Data

Supervisor: Dr Gaël Pierre Écorchard.
Czech Institute of Informatics, Robotics, and Cybernetics, Office
B-323,Jugoslávských partyzánů 3, 160 00
Prague 6

Abstrakt

Nová generace spolupracujících robotů umožňuje použití malých robotických ramen pracujících s lidskými pracovníky, např. robota YuMi, dvojitá robotická ramena 7-DOF určená pro přesnou manipulaci s malými předměty. Pro další přijetí takového robota v průmyslu musí být vyvinuty některé metody a systémy senzorů, které jim umožní provádět úkol, například uchopení určitého objektu. Pokud chce robot uchopit objekt, musí objekt umístit relativně vůči sobě. To je úkol rozpoznávání objektů v počítačovém vidění, což je umění lokalizace předdefinovaných objektů v datech obrazového snímače. Tato diplomová práce představuje potrubí pro rozpoznávání objektů jednoho izolovaného modelu v bodovém mračnu. Systém využívá data z bodového mračna vykreslená z 3D CAD modelu a popisuje jeho charakteristiky pomocí lokálních deskriptorů funkcí. Ty jsou pak porovnány s deskriptory dat z bodového mračna ze scény, aby se 6-DoF pozice modelu v souřadném rámci robota. Tento počáteční odhad pozice je pak vylepšen metodou registrace, jako je ICP. Provádí se také kalibrace robotické kamery. Příspěvky této práce jsou následující: Systém používá FPFH (Fast Point Feature Histogram) pro popis lokální oblasti a hypotézu - a paradigma testu, např. RANSAC v procesu párování. Na rozdíl od několika přístupů k těm, které se spoléhají na vlastnosti Point Pair jako deskriptory vlastností a geometrické hašování, např. hlasovací systém jako proces shody.

Klíčová slova: Detekce objektů, Odhad Pozice, Robotika, Bodová Data

Contents

Figures

Tables

Chapter 1
Introduction

Within this chapter, the reader receives an outline of the general context which surrounds this thesis. Starting with the motivation section and the ultimate goal to be accomplished, and a summary of the thesis' structure follows.

1.1 Motivation

For years, the industrial robot has undergone enormous development. Robot nowadays not only receives a command from the computer. But also has the ability to make decisions itself. Such abilities are well known in the world of the computer vision as recognizing and determining the 6D pose of a rigid body (3D translation and 3D rotation).
However, finding the object of interest or determining its pose in either 2D or 3D scenes is still a challenging task for computer vision. There are many researchers working on it with methods that go from state-of-the-art to deep learning ones where the object is usually represented with a CAD model or object's 3D reconstruction and typical task is the detection of this particular object in the scene captured with RBGD or depth camera. Detection considers determining the location of the object in the input image. This is typical in robotics and machine vision applications where the robot usually does a task like pick-and-place objects. However, localization and pose estimation is a much more challenging task due to the high dimensionality of the search in the workspace. In addition, the object of interest is usually sought in cluttered scenes under occlusion with the requirement of real-time performance which makes the entire task much more difficult.

1.2 Goal

We attempt to provide a system or pipeline for pose estimation of a rigid object in point cloud design for random picking of an isolated object by using depth images acquired from an RGB-D sensor. In addition, the development of a system that can help with the extrinsic calibration of a camera-robot

The goal is just to develop a suitable pipeline for localizing an isolated object where it can be suitable for future work such as a bin-picking system which is out of the scope for this master thesis.

1.3 Thesis structure

The thesis consists of 6 chapters, References and Appendix. The current chapter 1 briefly describes the motivation and the goal of thesis called "Part localization for Robotic Manipulation" which for convenience we refer as 6D pose estimation of a rigid body or pipeline pose estimation interchangeably. Chapter 2 gives a brief introduction to related work, Chapter 3 gives a theoretical background to camera calibration and a gentle description to the main tools used in this thesis such as openCV, open3D, ROS, and software where the CAD model is rendered. Chapter 4 presents the theory as well as every individual step in details of the implemented system, and chapter 5 describes the evaluation of the system. Chapter 6 concludes the thesis and showcase possible future works.

Chapter 2

Related work

Most of the literature tackle the problem of 3D Object Recognition(object detection and 3D pose estimation) by dividing into two broad categories as follow:

1. Global Feature-Based Methods
2. Local Feature-Based Methods

The global feature base methods process the object as a whole for recognition. They define a set of global features which describe the entire 3D object. On the other hand, the local feature based methods extract only local surfaces around specific keypoints. They can handle occlusion and clutter bettere when compared to the global feature-based methods.

2.1 Global Feature-Based Methods

The global feature-based methods define a set of global features which effectively and concisely describe the entire model. Examples of the global feature approach include shape distribution [5], and viewpoint feature histogram [4]. The global feature method ignores all details when it comes to the shape of the object and requires a priori segmentation of the object from the scene. Therefore, they are not suitable for recognition of a partially visible object from cluttered scenes.

2.2 Local Feature-Based Methods

The second class of method, the local feature based methods extract only local surfaces around specific keypoint. Yulan Guo et al. [26] presents a survey of local feature descriptors and cluster them into the three main groups which follow:

1. signature-based,
2. histogram-based, and
3. transform-based methods.

Yulan Guo et al. [26] in his survey claims that local features are much better than global features 2.1 for object recognition in occlusion and clutter scenes. This type of features has also proven to perform better in the area of 2D object recognition. That is why it has been extended to the area of 3D object recognition. Most articles such as [27] and [22] follow this pipeline and compare this with other local descriptors.

Chapter 3

Background

This chapter presents a briefly theoretical background as to mathematical tools and basics of computer vision. In addition, the API and tools used in this thesis. To dive deeply in any topic described ahead, a reference is given.

3.1 Mathematical Tools

3.1.1 Rigid Transformations

A rigid transformation also called Euclidean transformation is a geometric transformation of a Euclidean space that preserves the Euclidean distance between every pair of points. The rigid transformations include rotations, translations, reflections, or their combination. It can be shown that all rigid transformations can be expressed as follows.

$$g_(v) = R \cdot v + t, R \in \mathbb{R}^3 \tag{3.1}$$

A rigid transformation can be represented by using $4x4$ matrices by employing a homogenous coordinates as follows:

$$\begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P \\ 1 \end{pmatrix} = \begin{pmatrix} RP + t \\ 1 \end{pmatrix}$$

In the equation 3.1 the matrix, R, is referred to as a rotation matrix and has the following special properties.

- $R = (a\, b\, c)$, $a, b, c \in \mathbb{R}^3$
- $\|a\| = \|b\| = \|c\| = 1$ All columns are unit length
- $a \cdot b = b \cdot c = c \cdot a = 0$ The columns are mutually orthogonal

3.1.2 Rotation Matrices

The matrix R, a set of 3x3 matrices with the following properties, plus the operation of matrix multiplication forms a group called SO(3) which stands for special orthogonal group $\in \mathbb{R}^3$

$R = (a\, b\, c)$, $a, b, c \in \mathbb{R}^3$ is a rotation matrix for $\mathbb{R}^3$ iff

(a) : Astra Camera **(b) :** RealSense Camera

Figure 3.1: 2 RGB-D sensors

- $R^T \cdot R = I$
- $det(R) = 1$

Rotation Representations

- A rotation can be expressed as a $3X3$ matrix $R \in SO(3)$ where $R^T \cdot R = R \cdot R^T = I$ and $det(R) = 1$
- A rotation can also be expressed in terms of an angle θ and an axix $\hat{\omega} \in \mathbb{R}^3$ where $\|\hat{\omega}\| = 1$. It can relate to the matrix form via the Rodrigues formula.

$$R = exp(\theta J(\omega)) = I + \sin\theta J(\omega) + (1 - \cos\theta) J(\hat{\omega})^2$$

- And finally a rotation matrix expressed as a unit quaternion:

$$(u_0, u) = (\cos(\frac{\theta}{2}), \sin(\frac{\theta}{2})\hat{\omega})$$

3.2 Basics of 3D Computer Vision

3.2.1 RGB-D sensors

Nowadays novel camera systems like the Astra Orbbec and RealSense which provide both color and depth images have become readily available. Therefore, there are great expectations that such sensory devices will lead to a boost of new 3D perception-based applications in the fields of robotics. We are specifically interested in using RGB-D sensors for recognition and localization of an isolated part. In this thesis, both cameras are used. See Figure 3.1 in order to be acquainted with them.

Point Cloud

The receive measurement data from the input sensor get converted in a more generic data structure called point cloud, which is a set of vertices in a three-dimensional coordinate system usually defined by X, Y, and Z coordinates. The vertices are typically intended to represent the external surface of an object. Point clouds can be acquired from hardware sensors such as stereo cameras, 3D scanners, or time-of-flight cameras, or generated from a computer program synthetically. In this thesis, the point cloud is acquired from the sensory devices briefly described above in 3.2.1.

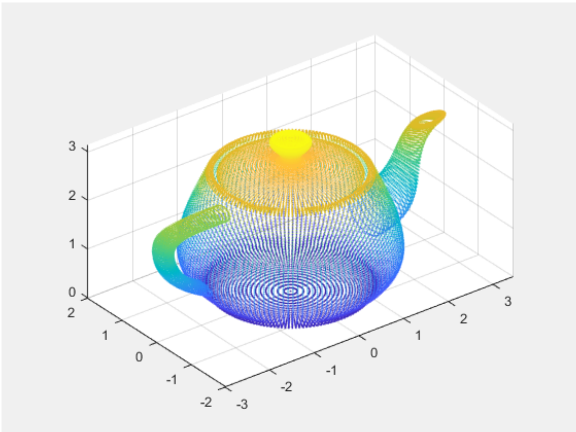

Figure 3.2: Overview of a point cloud (from MathWorks documentation)

3.2.2 Camera Pinhole Model

There are many lens models but Pinhole camera is used in this thesis. A pinhole camera is the simplest model that captures accurately the geometry of perspective projection. The image of the object is formed by the intersection of the light rays with the image plane. An illustration of the pinhole camera is seen in Figure 1. This mapping from the three dimensions onto two dimensions is called perspective projection. The camera projects point in the world frame $P_w = (X, Y, Z)^T \in \mathbf{R}^3$ through the pinhole to the point $p_c = (u, v)$ on the image plane.

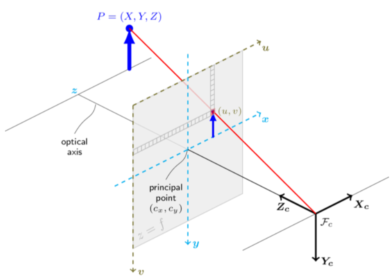

Figure 3.3: View of a Pinhole camera geometry (from Camera Calibration and 3D Reconstruction, openCV)

3.2.3 Parameters of camera model

We use $(u, v, 1)^T$ to represent a 2D point position in pixel coordinates or image plane. And $(x_w, y_w, z_w, 1)^T$ is used to represent a 3D point position in world coordinates. Note: they were expressed in augmented notation of homogeneous coordinates which is the most common notation in robotics and rigid body transforms. Referring to the pinhole camera model, a camera matrix is used to denote a projective mapping from world coordinates to Pixel coordinates(or image plane), the camera matrix is giving by Eq. 3.2.

$$z_c * \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K * \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} * \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{3.2}$$

3.2.4 Camera's Intrinsic Parameters

Images coordinates are measured in pixels, normally with the origin in the left upper corner. The focal plane in the pinhole camera model is embedded $\in R^3$ so we need to have a mapping that translates the points in the image plane into pixels, see Figure 3.4.

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

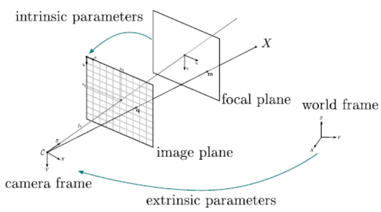

Figure 3.4: Overview of the transformation between the focal plane and the image plane

3.2.5 Camera's Extrinsic Parameters

The transformation between the world coordinate system and the camera coordinate system is achieved be a rotation and a translation. The translation is represented by a vector $t \in \mathbf{R}^3$ and the rotation by a 3x3 orthogonal matrix $\mathbf{R}$. So $\mathbf{R}$ represents a rotation matrix, and it must satisfy the following properties:

$$det(\mathbf{R}) = 1 \tag{3.4}$$

$$\mathbf{R}^T\mathbf{R} = I \tag{3.5}$$

Where I is the identity matrix. The matrix **R** and the vector **t** altogether are called camera's extrinsic parameters, see Figure.

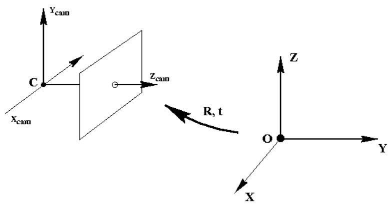

Figure 3.5: Overview of a world coordinate system and camera coordinate system

The transformation of a representation of point in the world coordinate system, $P_w = (X, Y, Z)^T$ into the camera coordinate system, $P_c = (X, Y, Z)^T$ can be done with the following equation.

$$P_c = \mathbf{R}.P_w + \mathbf{t} \tag{3.6}$$

The Equation 3.6 can also be written as:

$$P_c = [\mathbf{R}\ \mathbf{t}] \begin{bmatrix} P_w \\ \hline 1 \end{bmatrix} \tag{3.7}$$

3.3 Robotic Operating System

For this thesis The Robotic Operating System (ROS) is used as main platform. In addition, it is used for visualization purpose and debugging steps. ROS is a flexible framework for writing robot software. In addition, it is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platforms. It is based on the concepts of nodes, topics, messages and services. A node is an executable program that performs computation. Nodes need to communicate with each other to complete the whole task. The communicated data are called messages. ROS provides an easy way for passing messages and establishing communication links between nodes, which are running independently. They pass these messages to each other over a Topic, which is a simple string, Topics are asynchronous communication. As to, a synchronous communication, it is provided by services. Services act in a call-response manner where one node requests that another node execute a one-time computation and provide a response. For more details about ROS, the reader can refer to [6].

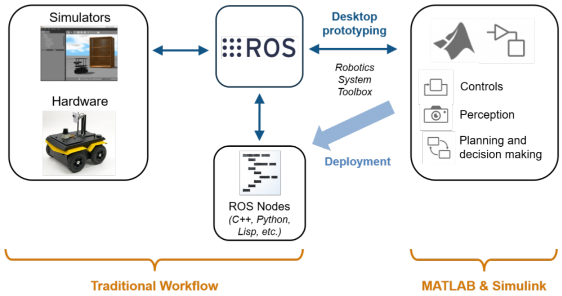

Figure 3.6: A ROS Overview

3.4 Open-source Libraries

3.4.1 PCL

The PCL[7] framework contains numerous state-of-the art algorithms including filtering, feature estimation, surface reconstruction, registration, model fitting and segmentation. These algorithms can be used, for example, to filter outliers from noisy data, align 3D point clouds together, segment relevant parts of a scene, extract keypoints and compute descriptors to recognize objects in the world based on their geometric appearance, and create surfaces from point clouds and visualize them.

For different processing steps, a Python bindings for the Point Cloud Library (PCL) is used. This is a reasonable python binding to the point cloud library. At present the following features of PCL, using PointXYZ point clouds, are available;

1. I/O and integration; saving and loading PCD (point cloud data) files
2. segmentation
3. sample consensus model fittting (RANSAC + others, cylinders, planes, common geometry)
4. smoothing (median least squares)
5. filtering (voxel grid downsampling, passthrough, statistical outlier removal)
6. exporting, importing and analysing pointclouds with numpy

3.4.2 Open3D

For the purpose of working with any ideal registration algorithm, the Open3D is used in this thesis which is an open-source library that supports rapid development of software that deals with 3D data. The Open3D frontend exposes

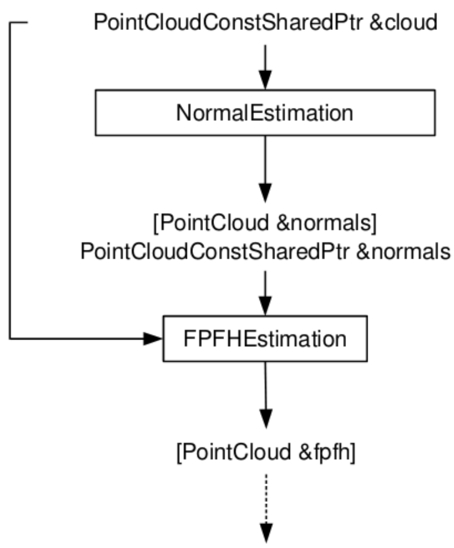

Figure 3.7: An example of the PCL implementation pipeline for Fast Point Feature Histogram (FPFH) [11] estimation.

a set of carefully selected data structures and algorithms in both C++ and Python. Open3D provides data structures for three kinds of representations: point clouds, meshes, and RGB-D images. For each representation, it offers a complete set of basic processing algorithms such as sampling, visualization, and data conversion. In addition, Open3D provides implementations of multiple state-of-the-art surface registration methods, including pairwise global registration, pairwise local refinement as the ICP registration [9], and multiway registration using pose graph optimization.

3.5 Software tools

For the purpose of rendering, convertion and manipulation of any 3D data(CAD model) several tools from the open source communities are used in this thesis such as CloudCompare, MeshLab and FreeCAD.

3.5.1 CloudCompare

CloudCompare is a 3D point cloud (and triangular mesh) processing software. It has been originally designed to perform comparison between two dense 3D points clouds (such as the ones acquired with a laser scanner) or between a point cloud and a triangular mesh. It relies on a specific octree structure dedicated to this task. Afterwards, it has been extended to a more generic point cloud processing software, including many advanced algorithms (registration, resampling, color/normal/scalar fields handling, statistics computation, sensor management, interactive or automatic segmentation, display enhancement, etc.)[12],.

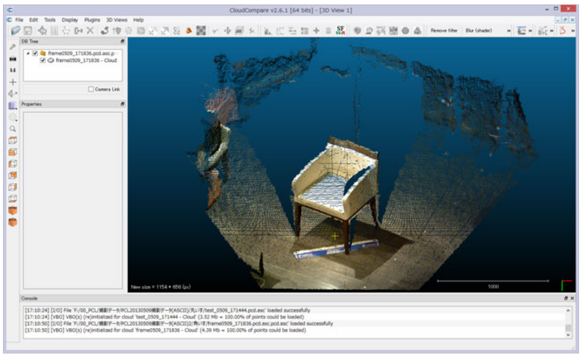

Figure 3.8: CloudCompare (view, edit and process).

3.5.2 MeshLab

Meshlab is an open source system for processing and editing 3D triangular meshes. It provides a set of tools for editing, cleaning, healing, inspecting, rendering, texturing and converting meshes. It offers features for processing raw data produced by 3D digitization tools/devices and for preparing models for 3D printing [13].

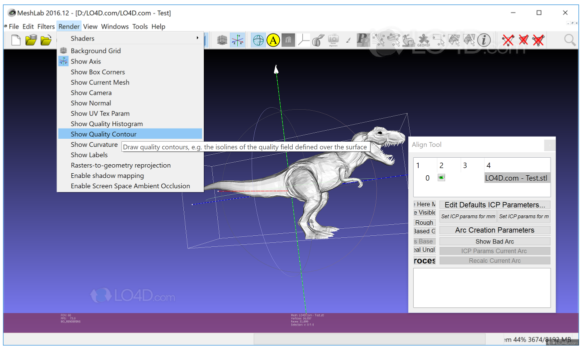

Figure 3.9: MeshLab (view, edit and process).

3.5.3 FreeCAD

FreeCAD is a 3D CAD/CAE parametric modeling application. It is primarily made for mechanical design, but also serves all other uses where you need to model 3D objects with precision and control over modeling history [14].

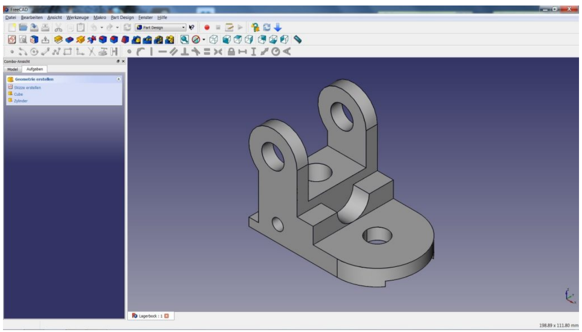

Figure 3.10: A view of the FreeCAD interface.

Chapter 4

Robot-Camera Calibration

This section presents the theory as well as each individual step for estimating the pose of the camera relative to the robot base frame, the problem to be solved is an extrinsic camera calibration (also known as a robot-camera calibration or Eye-to-hand calibration), but an intrinsic camera calibration needs to be solved a priori. The robot-camera calibration is a fundamental part of the subsequent use in the next chapter of this thesis. The extrinsic camera calibration methods generally require the position of the camera frame relative to a calibration target frame to be known. Therefore, the proposed method to solve the robot-camera calibration task for this thesis is based on tracking a calibration target (standard checkerboard calibration grid) attached to the end-effector of the robot with known forward kinematics.

4.1 Camera Calibration

Camera calibration is the process of estimating intrinsic and extrinsic parameters. The intrinsic parameters deal with the camera's internal characteristics, such as its focal distance, distortion, and image centre. The extrinsic parameters represent the position and orientation relative to the calibration target. In this thesis the camera calibration is treated separately and can be divided into two main stages:

- Sensor internal parameter calibration, like lens distortion, focal distance, optical center (image center) described above. In addiction to that, for RGB-D cameras, color and depth image offsets.
- Robot-camera calibration: the pose(position and orientation) of a camera coordinate system in a reference coordinate frame. In this thesis we also refer as to Eye-to-Hand calibration. The transformation from the camera coordinate system to the robot base coordinates system (also called world coordinates system interchangeably in this thesis) is shown in Figure4.1

Normally, it is sufficient to perform an internal camera parameter calibration only once for each device unless the lens or sensors itself will be changed or

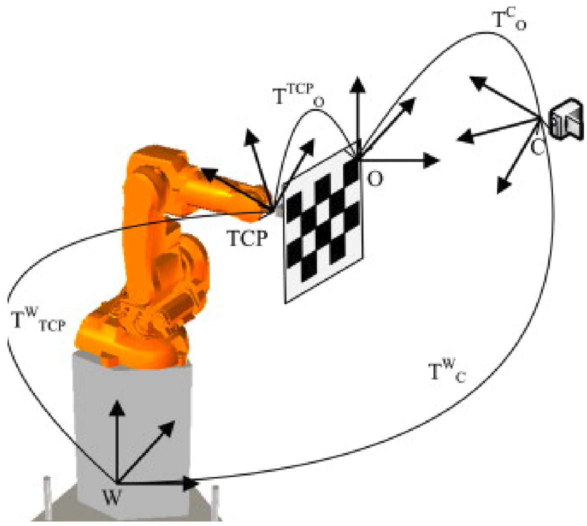

Figure 4.1: Overview of the camera pose estimation system. The system estimates the pose of the camera frame relative to the world frame(also known as robot base frame). Image from [17].

modified. Reliable calibration methods already exist, which are widely used [15] [16].

Robot-camera calibration, on the other hand, is more application specific and an important stage of any 6-DoF pose estimation system.

4.2 Sensor internal parameter calibration

4.2.1 Camera Model

The choice of camera model influences the final calibration results, so the first step is to select an appropriate camera model. In this thesis, the pinhole camera model 3.2.2 is used. It describes the mathematical relationship between the coordinates of a point in three-dimensional space and its projection onto the image plane of an ideal camera.

The MATLAB, Open CV and the *camera_calibration* ROS [**?**] packages are the most popular systems for camera calibration. They are already available for checkerboard detection based on the pinhole model and the method proposed by Zhang [15], All of them introduce the radial distortion and tangential distortion. In this thesis, the OpenCV and *camera_calibration* ROS packages are used for the purpose of comparison in this thesis. The technique proposed by Zhang only requires the camera to observe a calibration target shown at few (at least three) different orientations. the technique relates known points in the world to points in an image, in order to do so, one must first acquire a series of known world points. The most common method is to use known planar objects(checkerboard calibration grid) at different orientations with respect to the camera to develop an independent series of data points. The calibration object chosen in this thesis is a 6x9 checkerboard with the corner points as the known world points and it can be seen in Figure 4.2.

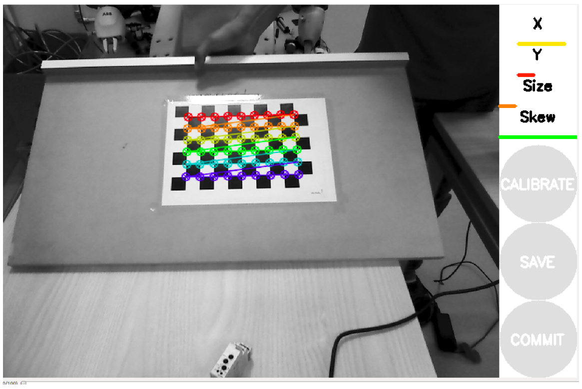

Figure 4.2: Overview of the intrinsic calibration based on industrial calibration ROS package with a 6X9 checkerboard calibration target

4.3 Eye-to-Hand Calibration

In order to know the pose of the camera coordinate system relative to the world coordinate system also known as robot base frame, extrinsic calibration (estimation of camera rotation and translation matrices) methods will be use. In this thesis, method for extrinsic camera calibration based on calibration target is used. It is assumed camera intrinsic parameters and distortion coefficients are known a prioria as describe in 4.2.1 and fixed during the entire sequence. Such system is shown in Figure 4.3.

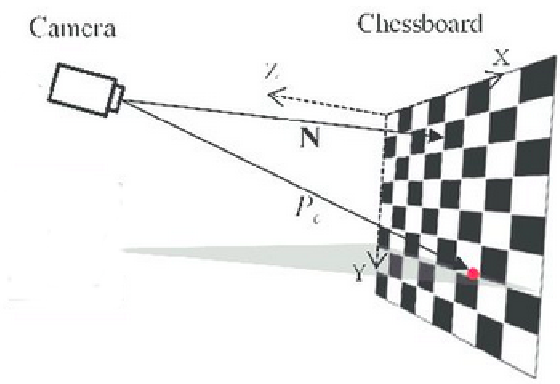

Figure 4.3: Overview of the camera pose estimation system. The system estimates the distance and orientation to the local coordinate system of the checkerboard

4.3.1 Calibration Targets

There are many types of camera calibration targets for use in imaging systems. In this thesis the planar targets are use since they can be easily printed with a standard printer and fixed to a surface. Planar targets can be subdivided as follow:

- Repeated pattern e.g. checkerboard patterns
- Non repeated pattern e.g augmented Reality (AR)

4.3.2 Checkerboard Patterns

Checkerboard calibration targets are one of the most frequently-used targets, where the calibration points are the corner points between squares. This pattern is simple to produce and allows for high accuracy because the corner points can be detected to subpixel precision. For example, the popular OpenCV library already contains algorithms to automatically locate plain checkerboards. Figure 4.4 shows an example of checkerboard calibration target.

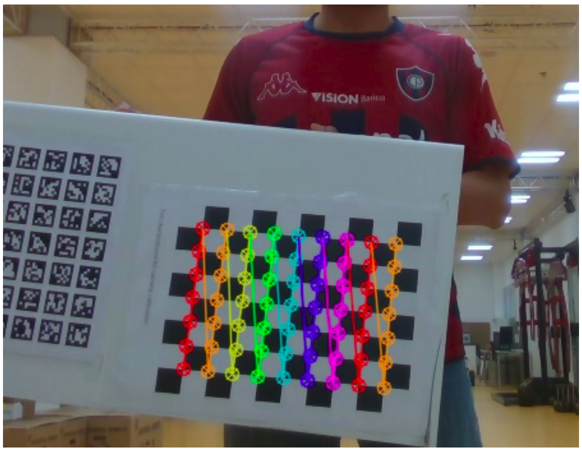

Figure 4.4: Overview of a 7X9 checkerboard calibration grid

4.3.3 Augmented Reality (AR)

AR markers also called Fiducial (individually identifiable) markers have become increasingly popular in recent years. Such markers can be used in a variety of settings such as camera calibration, where small markers are used, those who encode a unique code for identification purposes. There are a large number of markers available. One of the most common fiducial marker designs includes rectangular patterns with identification codes in the interior such as $ARTag$(2005), AprilTag and CALTag to name few of them. Refer to [21]

Figure 4.5: ARTag, AprilTag and CALTag markers example. Image from [21]

4.3.4 Selection

In this thesis, the checkerboard pattern is used. This pattern is simple to produce and allows for high accuracy because the corner points can be detected to subpixel precision [20].

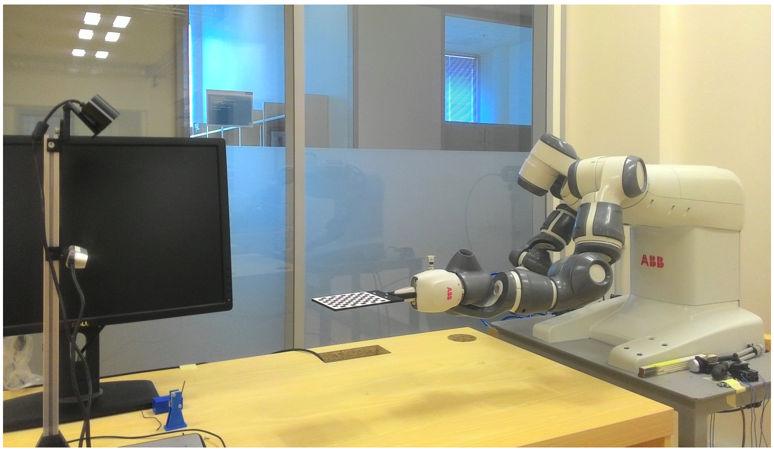

Figure 4.6: Overview of the setup, a ABB YUMI robot with a gripper holding the calibration plate. The camera is fixed around the robot workspace and pointed at the checkerboard

Bibliography

[1] Cheng-Hei Wu, Sin-Yi Jiang and Kai-Tai Song*, "CAD-Based Pose Estimation for Random Bin-Picking of Multiple Objects Using a RGB-D Camera" in: International Conference on Control, Automation and Systems (ICCAS 2015),Oct. 13-16,2015 in BEXCO, Busan, Korea,https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7364621.

[2] Ajmal S. Mian, Mohammed Bennamoun, and Robyn Owens, "Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes" in: IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 28, NO. 10, OCTOBER 2006,http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.123.6478&rep=rep1&type=pdf.

[3] Nitin J. Sanket and Daniel D. Lee, "6D Object Pose Estimation using RGBD Data and Fast-ICP" , https://nitinjsanket.github.io/project/ese650/p6/nitinsan_project6.pdf.

[4] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, John Hsu, "Fast 3D recognition and pose using the Viewpoint Feature Histogram" in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems , https://ieeexplore.ieee.org/document/5651280.

[5] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin, "Shape Distributions", http://graphics.stanford.edu/courses/cs468-08-fall/pdf/osada.pdf.

[6] Quigley, Morgan., Conley, Ken., Gerkey, Brian P.., Faust, Josh., Foote, Tully., Leibs, "ROS: an open-source Robot Operating System" in: Conference Paper; 2009. http://www.willowgarage.com/papers/ros-open-source-robot-operating-system

[7] Radu Bogdan Rusu and Steve Cousins, "3D is here: Point Cloud Library (PCL)" in: IEEE International Conference on Robotics and Automation (ICRA); 2011. http://pointclouds.org/assets/pdf/pcl_icra2011.pdf

[8] Qian-Yi Zhou and Jaesik Park and Vladlen Koltun, “Open3D: A Modern Library for 3D Data Processing” arXiv:1801.09847; 2018. http://www.open3d.org/

[9] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes”in: PAMI:1801.09847;1992.

[10] Klaas Klasing, Daniel Althoff, Dirk Wollherr and Martin Buss, “Comparison of Surface Normal Estimation Methods for Range Sensing Applications”

[11] Radu Bogdan Rusu, Nico Blodow, Michael Beetz, “Fast Point Feature Histograms (FPFH) for 3D Registration” in: IEEE International Conference on Robotics and Automation, Kobe International Conference Center Kobe, Japan, May 12-17, 2009.

[12] CloudCompare, https://www.danielgm.net/cc/

[13] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, G. Ranzuglia, "MeshLab: an Open-Source Mesh Processing Tool" in: Sixth Eurographics Italian Chapter Conference, page 129-136, 2008, http://www.meshlab.net/.

[14] FreeCAD, https://www.freecadweb.org/

[15] Zhengyou Zhang, "A Flexible New Technique for Camera Calibration" in: IEEE Transactions on Pattern Analysis and Machine Intelligence, Nov 2000 , https://ieeexplore.ieee.org/document/888718.

[16] Berthold K.P. Horn, "Tsai’s camera calibration method revisited", 2000 ,https://www.researchgate.net/publication/238495425_Tsai%27s_camera_calibration_method_revisited.

[17] Bernard Schmidt and Lihui Wang, "Automatic work objects calibration via a global–local camera system" in: obotics and Computer-Integrated Manufacturing, 2014, https://www.sciencedirect.com/science/article/pii/S0736584513001014.

[18] camera calibration ROS, http://wiki.ros.org/camera_calibration?distro=melodic.

[19] Y. M. Wang, Y. Li, and J. B. Zheng, "A Camera Calibration Technique Based on OpenCV", https://ieeexplore.ieee.org/document/888718.

[20] B. Atcheson, F. Heide, and W. Heidrich, "CALTag: High Precision Fiducial Markers for CameraCalibration" in: Vision, Modeling, and Visualization, 2010, https://pdfs.semanticscholar.org/2dba/e046717b058382a5a04f800405f92d040200.pdf.

[21] Artur Sagitov, Ksenia Shabalina, Leysan Sabirova, Hongbing Li, and Evgeni Magid, "ARTag, AprilTag and CALTag Fiducial

Marker Systems: Comparison in a Presence of Partial Marker Occlusion and Rotation", https://pdfs.semanticscholar.org/4b31/a78ff17b43b27afbddc63e0eede94bcec334.pdf?_ga=2.61993516.127748258.1556187821-1484823862.1553547725.

[22] ANIL K. JAIN and CHITRA DORAI, "3D object recognition: Representation and matching" in: Statistics and Computing(2000)10,167–182, https://epdf.tips/3d-object-recognition-representation-and-matching.html.

[23] Xian-Feng Han *, Jesse S. Jin, Ming-Jie Wang, Wei Jiang, Lei Gao, Liping Xiao, "A review of algorithms for filtering the 3D point cloud" in: Signal Processing Image Communication, May 2017, https://www.freecadweb.org/.

[24] Carlos Moreno and Ming Li, "A Comparative Study of Filtering Methods for Point Clouds in Real-Time Video Streaming" in: Proceedings of the World Congress on Engineering and Computer Science 2016 Vol I, October 19-21, 2016, San Francisco, USA, http://www.iaeng.org/publication/WCECS2016/WCECS2016_pp389-393.pdf.

[25] Rifat Kurban, Florenc Skuka, Hakki Bozpolat, "Plane Segmentation of Kinect Point Clouds using RANSAC" in: CIT 2015 The 7th International Conference on Information Technology, https://www.semanticscholar.org/paper/Plane-Segmentation-of-Kinect-Point-Clouds-using-Kurban-Skuka/5f8d9a32a21db5a7aa5bc68ce8ee8e486002ea06.

[26] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, and Jianwei Wan, "3D Object Recognition in Cluttered Sceneswith Local Surface Features: A Survey" in: IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 36, NO. 11, NOVEMBER 2014, https://www.semanticscholar.org/paper/3D-Object-Recognition-in-Cluttered-Scenes-with-A-Guo-Bennamoun/fa50e835690611b81929714602f06048ca1c2012.

[27] Radu Bogdan Rusu, Nico Blodow, Michael Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration" in: IEEE International Conference on Robotics and Automation, 2009, https://ieeexplore.ieee.org/document/5152473.

[28] Sebastian Thrun,Dirk Haehnel,Aleksandr V. Segal, "Generalized-ICP", http://www.robots.ox.ac.uk/~avsegal/resources/papers/Generalized_ICP.pdf.

Appendix A

List of Notation

Symbol	Meaning
$\mathbb{R}$	The real numbers
ICP	Iterative Closest Point
DOF	Degree(s) of Freedom.
CAD	Computer Aided Design.
FPFH	Fast Point Feature Histogram.
PCL	The Point Cloud Library is an open-source library of algorithms for point cloud processing tasks and 3D geometry processing.
Open3D	Open3D is an open-source library that supports rapid development of software that deals with 3D data.
RGB-D Camera	Specific type of depth sensing device that work in association with a RGB camera.
RANSAC	Random sample consensus. An iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers.
ROS	The Robot Operating System is a set of software libraries and tools that help you build robot applications.
ToF	Time-Of-Flight denotes a variety of methods that measure the time that it takes for an object, particle or wave to travel a distance through space.