



Fabian Oblinger, BSc

Visual Map building for Industrial Robot Applications

MASTER'S THESIS

to achieve the university degree of

Master of Science

Master's degree programme: Telematics

submitted to

Graz University of Technology

Supervisor

Dipl.-Ing. Dr.techn., Matthias R  ther

Institute for Computer Graphics and Vision

ABSTRACT

The paradigm shift from mass production to customized production has created the need for flexibility, cost-efficiency and transformability in the field of automation and especially in robotics. Mobile manipulators can perform a variety of tasks, can react to changes in the environment and can easily be integrated into various applications. One of the most challenging tasks when using mobile manipulators is to move the robot exactly to the desired target object pose, especially if the relation between the robot basis and the work station is not known and if the kinematic model of the robot consists of small inaccuracies.

This master thesis focuses on the realization of a precise pick and place application with a mobile manipulator, equipped with a stereo system on its end-effector. I mainly concentrate on the reliable and precise movement of the robot tool center point (TCP) to an object that is defined by a CAD model. In order to achieve this, the extended visual servoing procedure is presented. It consists of the industrial marker to object map and a linear pose extrapolation in combination with the common concept of visual servoing. The marker to object map ensures a robust detection of the object pose by using a pre-defined relation between the object and several 3D markers. The linear target extrapolation is performed on basis of multiple observations of this industrial marker to object map during the TCP movement towards the object. The extrapolation helps to overcome small deviations in the kinematic robot model and in the camera detections. All robot movements towards the target object are purely based on stereo system observations of the industrial marker to object map. No pose teaching of the manipulator is required. Thus, the movement to the object is independent of the current robot coordinate system (RCS) with respect to the work station.

A pick-and-place experiment of a pre-defined object in an industrial environment confirmed the successful operation of the extended visual servoing concept. The object handling from one workstation to another was performed with an absolute accuracy of about $\pm 0,3$ mm with respect to the object's X- and Y-axis. A change of the RCS between the pick and place operations at the two work stations had no effect on the precision of the robot/object interaction.

Keywords: mobile manipulator, stereo system, industrial marker to object map, visual servoing, linear extrapolation

KURZFASSUNG

Der Wechsel von Massenproduktion zur kundenspezifischen Produktion führt im Bereich der Automatisierung und vor allem in der Robotik zu höheren Anforderungen in Bezug auf die Flexibilität, Kosteneffizienz und Transformierbarkeit. Mobile Robotereinheiten können eine Vielzahl an Aufgaben erfüllen, auf Änderungen im Umfeld reagieren und einfach in verschiedene Anwendungen eingebunden werden. Die größte Schwierigkeit bei der Verwendung von mobilen Robotern liegt in der Ausführung von exakten Bewegungen zu einer spezifizierten Zielposition, wenn der exakte Bezug zwischen Roboterbasis und Arbeitsstation nicht bekannt ist oder die Modellierung der Roboterkinematik geringe Abweichungen beinhaltet.

Diese Masterarbeit konzentriert sich auf die Umsetzung einer präzisen Pick-and-Place Applikation mithilfe eines mobilen Manipulators, dessen Endeffektor mit einem Stereosystem ausgestattet ist. Dabei wird speziell die exakte und wiederholbare Bewegung des Roboter-Werkzeugkoordinatensystems (TCP – Tool Center Point) zu einem mittels CAD Modell definierten Bauteils betrachtet. Diesbezüglich wird eine Erweiterung des Visual Servoing eingeführt, welche zusätzlich zum Grundkonzept des Visual Servoing noch eine visuell erstellte industrielle Marker/Objekt Karte und eine lineare Pose Extrapolation beinhaltet. Die industrielle Marker/Objekt Karte dient zur robusten Pose Detektion des Objektes durch Verwendung des vordefinierten Bezuges zwischen dem Objekt und mehreren 3D Markern. Die lineare Extrapolation basiert auf mehreren Detektionen der visuellen Marker zu Objekt Karte während sich der TCP in Richtung Zielposition bewegt. Die Extrapolation hilft kleine Abweichungen in dem Modell der Roboterkinematik und kleine Detektionsfehler auszugleichen. Alle Bewegungen des Roboters in Richtung des Objektes bedienen sich des Stereosystems und der visuellen Karte. Dem Roboter müssen keine Positionen vordefiniert werden. Dadurch ist die Bewegung des TCP zum Objekt unabhängig von dem aktuellen Bezug zwischen dem Roboterkoordinatensystem und der Arbeitsstation.

Ein Pick-and-Place Experiment eines vordefinierten Objektes in einem industriellen Umfeld bestätigte die erfolgreiche Implementation des erweiterten Visual Servoing Konzeptes. Die Aufnahme des Objektes von einer Station und die Ablage in eine andere Station wurde mit einer absoluten Genauigkeit von etwa $\pm 0,3$ mm in Bezug zu der X- und Y-Achse des Objektkoordinatensystems durchgeführt. Eine Änderung des Roboterkoordinatensystems zwischen der Aufnahme und dem Ablegen des Objektes hatte keinen Effekt auf die Präzision der Roboter/Objekt Interaktion.

Schlüsselwörter: mobiler Manipulator, Stereosystem, industrielle Marker/Objekt Karte, Visual Servoing, lineare Extrapolation

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

ACKNOWLEDGEMENTS

At this point, I would like to offer my special gratitude to my research supervisor Matthias R ther for his professional support and his constructive ideas and suggestions during the development of this thesis. Furthermore, I want to thank him for his interesting lectures which introduced me to the topic of Computer Vision in combination with Robotics.

Great thanks also goes to Andreas Wimmer, Rainer Reisinger and Alois Wiesinger for giving me the opportunity to realize my master's thesis in cooperation with the companies FILL GesmbH and Weitblick Systems GmbH. Moreover, they provided me with valuable equipment and sophisticated technical support.

Assistance offered by my work colleagues was greatly appreciated. The long discussions about the usage of different Computer Vision algorithms and the correct evaluation of measurement results certainly improved my work. It is a great pleasure and honor to work in a team with all of you.

Moreover, I want to thank Claudia Huber and Elisabeth Zeilinger for proof-reading this document and for the valuable comments on this thesis.

Finally, I would like to express my profound gratitude for the great support provided by my friends and family and for the continuous encouragement throughout my years of study. A special thanks goes to Stefanie Denkmayr for encouraging me in stressful situations and in times of slow progress. I also want to thank Christoph Leitner, who sparked my interest to specialize on Computer Vision at the TU Graz and who accompanied and assisted me in almost every lecture of my bachelor and master studies.

Thank all of you very much!

Graz, April 2017

Fabian Oblinger

CONTENTS

1.	Introduction/Motivation	1
2.	Related Work	6
3.	Background.....	13
3.1.	Position and Orientation Representation	13
3.1.1.	Position and Displacement.....	13
3.1.2.	Orientation and Rotation	13
3.1.3.	Homogenous Transformations	15
3.2.	Camera System	16
3.2.1.	Basics to Camera and Lens Selection.....	16
3.2.2.	Monocular System – Background.....	17
3.2.3.	Monocular System – Calibration Method.....	19
3.2.4.	Region Characterization using Geometric Moments.....	20
3.2.5.	Binocular System – Background	22
3.2.6.	Binocular System – Calibration Method.....	24
3.2.7.	Binocular System – Depth Resolution	25
3.3.	Robot Kinematics.....	27
3.3.1.	Forward Kinematics	28
3.3.2.	Inverse Kinematics.....	29
3.4.	Machine and Process Capability	30
4.	Methodology	31
4.1.	Extended Visual Servoing Concept.....	32
4.2.	Industrial Marker to Object Map	35
4.3.	Object Detection – Shape-Based 3D Matching	37
4.3.1.	Basic Concept of Shape Based 3D Matching (SBM)	37
4.3.2.	SBM Refinement using a Stereo System (SBM-SOT)	41
4.3.3.	SBM Refinement using a Stereo System in Combination with Ellipse Fitting (SBM-ECT)	42
4.4.	3D Marker Detection	44
4.4.1.	3D ARToolKit Marker – Monocular (3D ARTM-M)	44
4.4.2.	3D ARToolKit Marker – Binocular (3D ARTM-B)	46
4.4.3.	3D QR Code Marker (3D QRCM)	48

4.4.4.	3D Fiducial Propeller Marker (3D FPM).....	51
4.5.	Robot Control.....	54
4.5.1.	Hand-Eye Calibration	54
4.5.2.	Position-Based Cartesian Robot Control	56
4.5.3.	Robot Extrapolation.....	59
5.	Experiments/Results	62
5.1.	Equipment.....	63
5.1.1.	Industrial Manipulator – Universal Robot UR10.....	63
5.1.2.	Camera System	63
5.2.	Monocular Camera System – Accuracy	65
5.3.	Binocular Stereo System – Accuracy	69
5.4.	3D Object Detection – Method Evaluation.....	73
5.4.1.	SBM Accuracy	74
5.4.2.	SBM-SOT Accuracy	76
5.4.3.	SBM-ECT Accuracy	78
5.4.4.	Erroneous 3D Pose Detections	79
5.5.	3D Marker Detection – Method Evaluation	83
5.5.1.	ARToolKit Marker – Monocular (3D ARTM-M)	83
5.5.2.	ARToolKit Marker – Binocular (3D ARTM-B).....	86
5.5.3.	3D QR Code Marker (3D QR CM).....	88
5.5.4.	3D Fiducial Propeller Marker (3D FPM).....	90
5.5.5.	Fiducial Marker – Detection-Height Variations.....	92
5.6.	Industrial Marker to Object Map	95
5.7.	Position-Based Cartesian Robot Control.....	98
5.7.1.	Universal Robot UR10 – Accuracy Measurements.....	98
5.7.2.	Hand-Eye Calibration	101
5.7.3.	Marker Detection Accuracy with respect to the RCS	102
5.7.4.	Position Based Visual Servoing.....	104
5.8.	Extended Visual Servoing – Extrapolation.....	110
5.8.1.	Varying Number and Height of Support Points	111
5.8.2.	Absolute Accuracy of Robot/Object Interaction	114
5.9.	Time Performance	117
5.10.	Pick and Place Experiment	118

6.	Discussion	121
7.	Conclusion and Future Work.....	128
8.	Bibliography	I
9.	List of Figures	VI
10.	List of Tables	XIII

1. Introduction/Motivation

Since the beginning of the industrialization, there were three major technological leaps that have led to paradigm shifts in industrial production (also known as “industrial revolutions”). The first revolution (industry 1.0) started with the mass production of goods. Water and steam powered mechanical machines were fabricated to enhance the output [5]. Replacing water and steam as a power source for machines with electricity led to the second paradigm shift (industry 2.0). It was the beginning of series production and globalization. The third industrial revolution began by improving the automated production with the use of electronics and IT. The successive automatization of production processes and the replacement of the human worker with machines in the field of series production are characteristic for industry 3.0 [21]. The ongoing digitalization of previously analogue data and the combination of “smart” (future oriented technologies) objects and internet technologies seem to result in a new fundamental paradigm shift in industrial production and represent the beginning of industry 4.0.

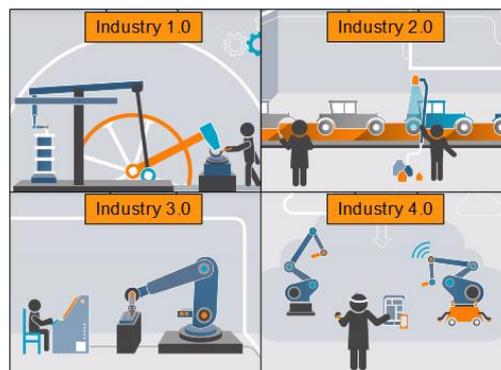


Figure 1: Visualization of the four industrial revolutions [21]

Industry 4.0 (also known as digital revolution) not only describes the development of industrial technologies, but also the change of the production and working environment. It can be divided into four basic principles [42]:

- Connection: Machines, equipment, sensors, logistics and humans should be connected and are able to communicate directly with each other.
- Information transparency: The data delivered by sensors extend the information system of a virtual factory model to get a virtual image of the real world.
- Technical assistance: Assistance systems support the humans with aggregated, visualized and understandable information. This helps to make fast and informed decisions in case of an error situation. Furthermore, humans are assisted in exhausting, uncomfortable or dangerous working procedures.
- Decentralized decisions: The system should be capable of making its own decisions and perform tasks autonomously. The task should only be passed on to a higher instance in case of conflicts or errors.

One of the main aspects of industry 4.0 is that the human represents the center of the machines. He or she intuitively assigns tasks, delivers information that is not covered by sensor systems and controls and monitors the mainly autonomous manufacturing process. Therefore, workers are included in the manufacturing process and can contribute to the system in case it stops due to an undefined situation or an error. Simultaneously, the new technologies offer the opportunity for humans to work hand in hand with robots. Instead of placing the robot behind safety technology and separating the workspace of humans and robots, a collaborative work between those two parties is desired. People are integrated into cyber-physical structures, so that their skills and talents can be used in combination with the support of robots [5].

To be competitive with other companies, one has to economically produce a high variety of products at a low batch size. The mass production of products without the actual order of the customer wastes resources and material and is, therefore, not state of the art. Industry 4.0 is able to react to trends and needs of the customer, the market and the society with great flexibility and speed. New production systems have to consider the need for strong product individualization and flexible process adaptations.

The paradigm shift from mass production to customized production has created the need for flexibility, cost-efficiency and transformability in the field of automation and especially in robotics. Robot arms are capable of moving at great speed with great accuracy to perform repetitive tasks. According to [58], robotics has achieved its greatest success to date in the world of industrial manufacturing.

Yet, there is one fundamental disadvantage: the lack of mobility. The majority of today’s robots operate in environments that are contrived to fit the requirements of the robot. Once the working environment has been created, all relevant positions and operations of the robot must be taught. It is important for a precise interaction that the relation between the robot base and the target machine remains unchanged. Applications with changing work environments cannot be controlled accurately and represent a problem for robots due to their lack of perception of changes [32].

Therefore, it is advised to introduce autonomous industrial mobile manipulation (AIMM) systems to replace the static robot. AIMMs are represented by robot arms that are mounted on a mobile platform to increase the flexibility of the robot’s working range.

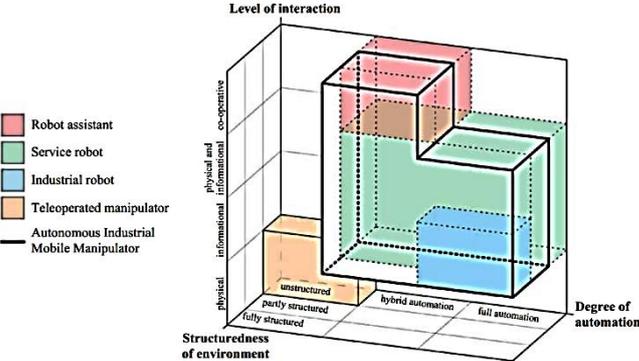


Figure 2: Classification of the AIMM domain with respect to the other robotic systems [34]

Figure 2 visualizes the classification of the AIMM based on the level of interaction, degree of automation and structure of environment. It shows that AIMM robots have the efficiency of industrial robots and the flexibility of service robots and robot assistants [34].

Mobile manipulators can perform a variety of tasks, can react to changes in the environment and can be easily integrated into various applications. According to [34], an AIMM can be characterized by the following properties:

- It is an independent system that can operate fully automatically.
- It can work in industrial environments in cooperation with humans.
- It can gather information in order to react to changes.

The main tasks for AIMMs are the transport of parts between workstations and storages and the loading and unloading of components into feeders and machines. Furthermore, the AIMM can be used to deliver parts, screws or tools in small load carriers from a storage shelter to a worker on an assembly station. Mobile manipulators can replace workers in simple and repetitive tasks (like commissioning of parts) to free them up for more creative assignments that are difficult to automate. The introduction of AIMMs will open up new possibilities in the area of manufacturing for small and medium sized companies. The combination of robot/human interaction enables an optimized work utilization, where the robot performs simple repetitive tasks, whereas the human performs work which requires specific knowledge and skills (Figure 3). This concept embodies the factory of the future [34].

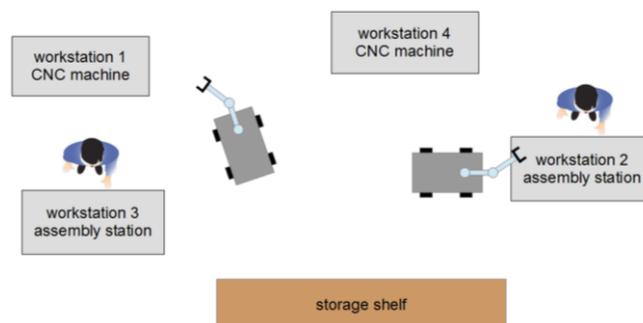


Figure 3: Concept of AIMM usage – autonomous industrial mobile manipulators supply all working stations (assembly or automated manufacturing station) with parts from the storage shelter

Especially small companies are interested in the automatization of part handling to keep up with the flexibility and product variation of large companies. The high complexity with respect to planning, designing, adaption, initial setup and maintenance of today's automatic part handling applications requires the work of specialists and is therefore associated with high costs. Consequently most small sized companies cannot effort to introduce AIMMs [34].

To reduce the complexity and the costs, I will develop an autonomous part handling system that can easily be initialized and adapted by the user. The user should be able to change the logistic process without investing much time or effort. I concentrate on:

- Reduction of the configuration effort and complexity (robot teaching)
- Enhancement of self-adaption in case of environmental changes
- Enhancement of the absolute robot/object interaction accuracy (+/- 1mm)

The robot should be capable to move between different work stations and perform tasks that are not profitable for a static mounted robot. Furthermore, the manipulator should be able to pick and place objects with great precision, even if the relation between the mobile platform and the work station changes (Figure 4).



Figure 4: Significant change of the mobile platform pose at the workstation. Nevertheless, the robot should perform accurate pick and place operations

This thesis leaves the mobile platform aside and focuses on the reliable 3D object detection and on how to correctly control a robot to a target pose. The path-planning and movement between the work stations is performed by a black box that sends a signal as soon as the platform has reached its destination. Furthermore, I solely concentrate on pick and place operations of parts that have a static relationship with the work cell, for instance a part in a clamping device of a CNC (computerized numerical response) machine. Figure 5 illustrates the basic workflow of my approach for a pick and place handling at a workstation called W1. Two of the aspects that make this thesis particularly interesting are the introduction of the industrial marker to object map and the extended visual servoing process.

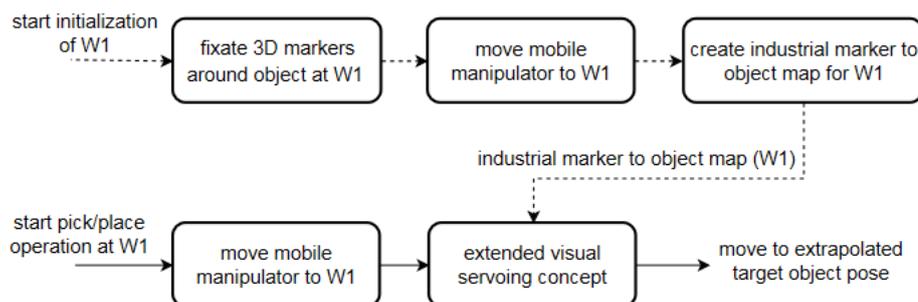


Figure 5: Simplified workflow for pick and place operations described in this thesis

The industrial marker to object map is built in an initial step and helps to define a reliable object pose with respect to several 3D markers (Figure 6). This relation ensures a robust detection of the object pose by simply taking the combination of a 3D marker detection algorithm and the pre-defined industrial marker to object map. The extended visual servoing concept uses the marker detection and the marker to object map to feed the robot controller with accurate object poses to ensure a correct and accurate definition of the robot's target pose. During the approach of the object, the visual servoing process continues with the detection of the object pose with respect to the base coordinate system of the robot. Corrections are applied until all 3D markers are out of sight. If no further detections can be performed, an extrapolation is applied to estimate the correction of the final movement. This helps to overcome inaccuracies of the robot's kinematic model.

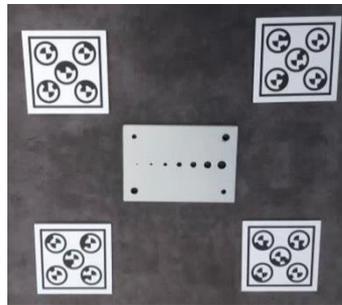


Figure 6: Setup for the definition of the industrial marker to object map. In an initial step, numerous object detections are put into relation to the four 3D markers which surround the object

This thesis is organized as follows: In section 2, I mention related work on autonomous industrial mobile robots and how I improve these operations by using an extension of visual servoing and the industrial marker to object map. With regard to these topics, I comment on some of the already existing object detection algorithms ('Deformable Planar Surface-Based 3D Matching' [30], Descriptor-Based Matching [43], Surface-Based 3D Matching [17] and Shape-Based 3D Matching [64]). Moreover, I explain the already existing 3D marker of ARToolKit [1] and the 2D QR Code and 2D Fiducial Propeller marker [56]. Afterwards in section 3, I provide the reader with background information on topics of computer vision and robotics that are related to this work. In section 4, I discuss the complete workflow for the methodology of this thesis and how to use an extended version of position based visual servoing and the industrial marker to object map, to implement an accurate movement to the target poses. Relevant topics like the creation of the industrial marker to object map, object and marker detection, Hand-Eye calibration, iterative robot control and linear extrapolation of the robot movement are explained. In section 5, I apply these methods and evaluate their quality and performance. The experiments have been conducted bottom-up, beginning with the camera calibration and ending with the measurement of the absolute accuracy of the robot's end position. The set of experiments conducted are summarized in section 6. Finally, a conclusion of the thesis and an outlook on future work is presented in section 7.

2. Related Work

Many investors have increased their interest in the topic of mobile manipulation technology. The development of mobile manipulation products strides forward as companies seek to automate the transport of parts between several working stations. According to [34], the majority of the research in this direction has focused on the improvement of single technologies and functionalities like motion control and coordinated movements. This resulted in mainly academic applications which were operated in laboratories and were not tested in real-world industrial environments.

The majority of the academic projects focuses on service robotics to perform low accuracy operations like grabbing a bottle or a can with a vacuum or a high tolerance gripper. These applications are not capable of performing tasks in an industrial environment. Examples for academic projects are the RB-1 (Robotnik Automation), the X_WAM (Robotnik Automation), TUM-Rosie (Technical University Munich), the Care-O-Bot (Fraunhofer IPA) or the PR2 (Willow Garage) [34]. Some of these projects are visualized in Figure 7.

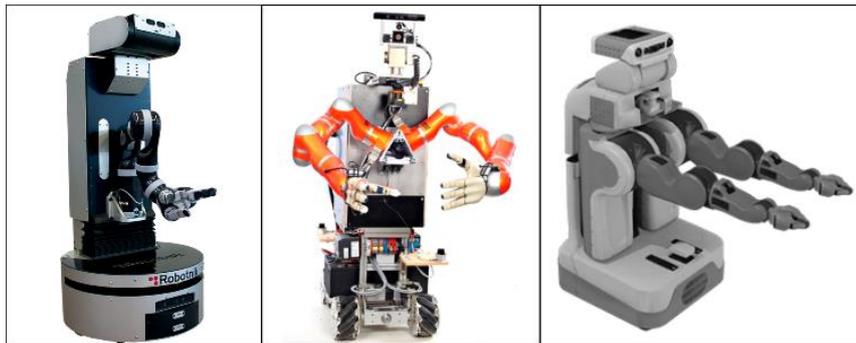


Figure 7: Autonomous academic mobile manipulators – RB-1 [54] (left), TUM-Rosie [63] (middle) and PR2 [67] (right)

Besides the academic research, some project groups concentrate on the development of AIMM systems that are capable of dealing with industrial environments (Figure 8). For instance, it is the goal of project STAMINA (sustainable and reliable robotics for part handling in manufacturing automation) to develop an autonomous mobile industrial robot to handle logistic tasks like bin-picking, kitting and de-palletizing [3]. Three static cameras are mounted on the mobile platform over the manipulator and perform detections in front of and on both sides of the robot. As the manipulator on the platform moves into the workspace towards the object, no further detection and correction of the object pose can be applied due to occlusion. The field of view of all cameras can only be changed by moving the platform. The basic concept is explained in [31], but it does not contain any specification of the system's accuracy.

Another project for autonomous manipulators is called TAPAS (Robotics-enabled logistics and assistive services for the transformable factory of the future) [62]. This project focuses on the automatization of logistic tasks by using mobile manipulators. An exam-

ple of this project is called “Little Helper” (from the Department of Production at Aalborg University). A monocular camera is mounted on the end-effector of the robot to observe the object pose. The 3D pose is determined by using a combination of 2D detection and a pre-defined grasping height which can be taught by a user. Details to the algorithm are explained in [2]. The accuracy of the pose estimated via the vision system lies within +/- 10 mm. It can be improved by using a haptic calibration. This is very time consuming and has to be performed each time the mobile platform reaches a working station. Furthermore, the robot has to be equipped with a precise tactile sensor and an accurate moving mobile platform (omnidirectional moving platform). If the platform does not place the end-effector close to the tactile reference point, the calibration will fail.



Figure 8: Autonomous industrial mobile manipulators – project STAMINA [3] (left), project TAP-AS [62] (middle) and the project of DLR [40] (right)

There exists another possible method for determining the relation between the mobile platform and the working station. A calibration plate is mounted on each work station. The manipulator calibrates itself to the work station by detecting the calibration plate pose as explained in [33]. Depending on the accuracy and speed of the calibration, the following results as depicted in Table 1 can be accomplished.

	High-precision	High-speed
Linear displacement	+/- 0.1 mm	+/- 1.0 mm
Angular displacement	+/- 0.1°	+/- 1.0°
Calibration time	60 seconds	10 seconds

Table 1: Accuracy results of the mobile manipulator “Little Helper” when using an additional high precision calibration plate to determine the relationship between the work cell and the robot [33]

The accuracies of Table 1 only represent the detection repeatability of the object with respect to the mobile manipulator. The documentation of this project ([33]) does not mention any measurements regarding the absolute accuracy of the end-effector position when grasping the object.

The Institute of Robotics and Mechatronics of the German Aerospace Center (DLR) is currently developing a mobile manipulator [40]. It uses a stereo system that is mounted on the mobile platform to detect objects and build the relation to the target working

space. This project concentrates on handling small load carriers for screws or small parts to manage the logistics between several storage shelves. The object detection is performed by fitting a model into a point cloud. The shelf localization and identification is accomplished by detecting QR codes that are mounted on the storage shelf. The 3D detection of the QR codes replaces the robot to work cell calibration mentioned for the “Little Helper”. The time for the calculation of the robot to work cell relation is significantly reduced. Neither the website of the project nor [18] contain any information about the overall accuracy of the system. The grasping process of the video [15] shows that the end-effector of the robot arm slightly pushes the box until it aligns with the gripper. Therefore, I assume that the tolerance lies above ± 3 mm.

In this work, I want to extend the previously mentioned concept of the mobile manipulator developed by the Institute of Robotics and Mechatronics of the DLR. As started in section 1, I will concentrate only on controlling the robot arm on top of the mobile platform. I will assume that the movement of the platform from one workstation to another is performed by a black box that delivers a signal when the platform reaches its target destination.

The main goal is to create a user friendly application that performs precise pick and place tasks. An average worker without special education should be capable of performing the setup and operating the application. The accuracy of the object interactions should be performed within a tolerance of ± 1 mm, even if low cost resources are used. Despite the fact that the previously mentioned project groups made no absolute accuracy specifications of the object interaction, the demonstration videos show that it does not lie within ± 1 mm. To my knowledge, there is no vision based approach of a mobile manipulator that can handle pick and place operations of parts in an industrial environment with an absolute accuracy smaller than ± 1 mm.

In order to achieve such a precise interaction with the environment, even when using low cost resources, the industrial robot has to be robustly controlled towards the target. This task can be realized by using the concept of visual servoing (Figure 9).

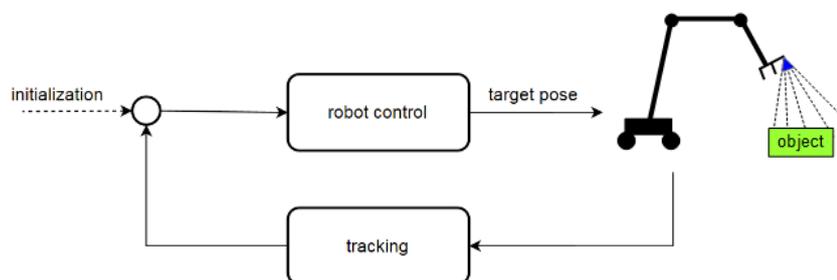


Figure 9: Basic visual servoing concept (adapted from [39])

Visual servoing consists of robot control and a tracking system of the object pose. During the controlling towards the target, the tracking system always has to observe the target pose. Camera configurations as used in the TAPAS project are therefore not suited for visual servoing.

According to [10], visual servoing topics, starting from simple pick-and-place tasks to advanced manipulation of objects, have been studied for more than three decades and can be separated into two different approaches [9]:

- Image-based visual servo control (IBVS)
- Position-based visual servo control (PBVS)

IBVS uses a set of 2D measurements that are immediately available in the image to estimate the desired movement of the robot. This method performs tracking and controlling tasks by reducing the image distance error between the currently measured and the given desired image features in the image plane (shown in Figure 10). Hence, the servoing process is performed directly on the basis of image features. The manipulator control is typically defined in either joint coordinates or in task space coordinates. As a consequence, it is necessary to relate changes of the image features to changes in the position of the robot, which is done by introducing an image Jacobian [39].

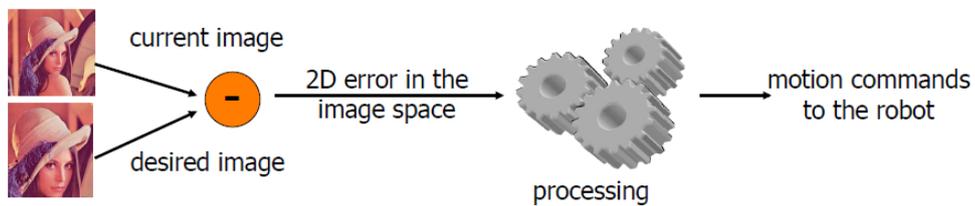


Figure 10: Image based visual servoing - basic concept [12]

In PBVS, extracted image features and a geometric model of the target are used to estimate the 3D pose of the object with respect to the camera, which is known as 3D localization problem. Using the estimated 3D pose, one can minimize the difference between the 3D poses of the robot end-effector and the object [10]. The main advantage of the PBVS is that the task can be described in terms of a Cartesian pose as it is common in robotics. Therefore, the guidance of the manipulator can be performed by feeding the common robot controller with the calculated offset poses. On the other hand, the robust and accurate estimation of the 3D object pose to calculate the offset pose is a challenging task. PBVS strongly depends on an exact calibration and an accurate object model. These requirements have to be fulfilled to ensure a proper robot/object interaction [10].

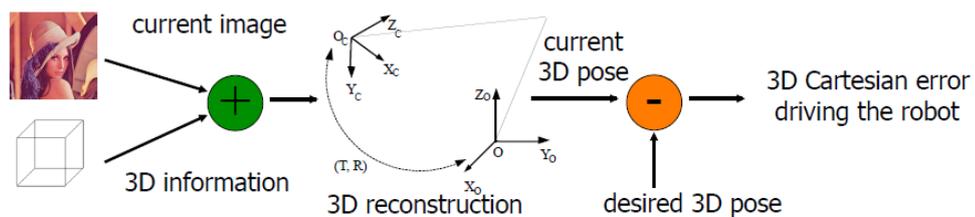


Figure 11: Position based visual servoing - basic concept [12]

Due to the simple mathematics of the robot controller, the PBVS is expected to be more promising for my thesis. Furthermore, there are already several algorithms available to determine a robust and accurate 3D object pose.

One of these object detection algorithms is the ‘Deformable Planar Surface-Based 3D Matching’ [30]. This method is based on an edge-direction template matching algorithm and applies this matching to a characteristic planar part (Figure 12). This technique assumes that spatially coherent structures stay the same even after perspective distortion. With an accurate camera calibration and a metric size model (computer aided design (CAD) data), the 3D pose can be determined by decomposing the homography that maps the points of the model to the image (singular value decomposition (SVD)). This method can be successfully applied only if the 3D object contains a unique but planar part that significantly differs from other structures in the expected scene [51].

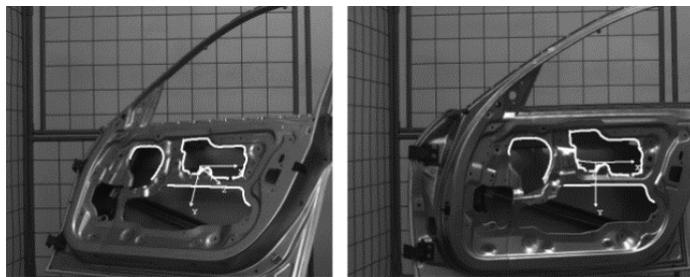


Figure 12: Deformable Planar Surface-Based 3D Matching – example (adapted from [30])

Another 3D pose estimation algorithm exploits distinctive points of an object (Figure 13). The ‘Descriptor-Based 3D Matching’ is used for objects that are characterized by an arbitrary but fixed texture [50]. The feature points are extracted by a detector and then classified according to their location and their local neighborhood (descriptor). The process of extraction and classification can be efficiently trained using randomized trees as proposed in [43] to ensure a robust and reliable matching of interest points. Having an accurate calibration and the point matches, the 3D pose can be estimated by applying the SVD to the homography that maps the interest points of the model to the image, similar to the ‘Deformable Planar Surface-Based 3D Matching’.

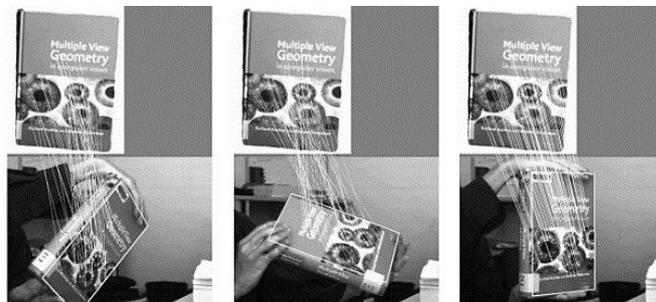


Figure 13: Descriptor-Based 3D Matching – example (adapted from [43])

The ‘Surface-Based 3D Matching’ algorithm is used to quickly locate free-form 3D objects in a 3D scene (point cloud). The model that is used for the object fitting can be

obtained either from a CAD model or from an acquired reference 3D scene. This method uses a global representation of the object and is independent from local surface information. Therefore, the detection does not depend on the quality and resolution of the acquired data. Even a sparse scene or model data leads to an efficient, stable and accurate result [64].

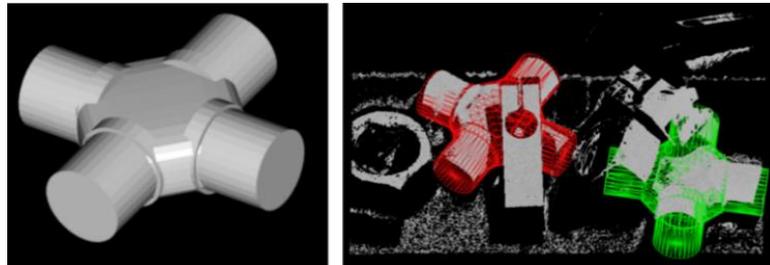


Figure 14: Surface-Based 3D Matching – example (adapted from [64])

The ‘Shape-Based 3D Matching’ uses the geometry information of a 3D CAD model to automatically generate a hierarchical model for 3D object detection [51]. The hierarchical model is based on 2D views of the object which can be used to find the object in an image. The main disadvantage of view-based methods is that the accuracy of the pose is limited to the density of the sampled views. Therefore, the pose of each found object candidate has to be refined by minimizing a geometric distance measure in the image. This approach does not depend on texture or on reflectance information and is able to handle true perspective, noise, occlusions and contrast changes [17].

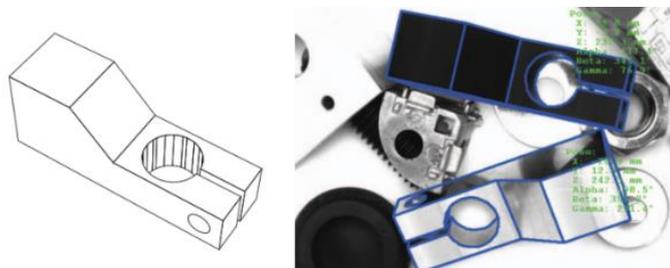


Figure 15: Shape-Based 3D Matching – example (adapted from [17])

Each of the four introduced object detection methods strongly depend on an accurate creation of the model and furthermore on a scene that is well illuminated and in focus. Shadows, blurry edges or small deviations between the object and model geometry influence the 3D estimation and lead to an inaccurate pose. As mentioned before in section 1, the main focus lies on the part handling between machines with a static object to work station relation. The fixed relation between the part and the work station enables the creation of an industrial marker to object map.

The industrial marker to object map describes the relation between optimal object detections and the detections of robust 3D markers which are placed around the object. A good marker is distinctive, can be detected with subpixel accuracy and is robust with respect to illumination changes and reflections. All object poses that are used during

the visual servoing process are based on the combination of 3D marker detection and the relations of the pre-built industrial marker to object map.

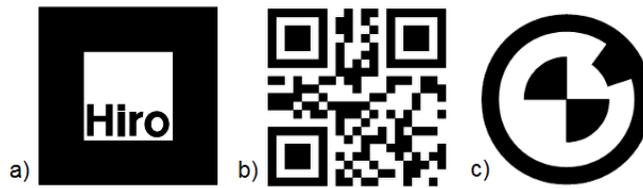


Figure 16: Marker types – a) 3D ARToolKit square marker [1]. b) 2D QR Code marker. c) 2D Fiducial Propeller marker (adapted from [56])

One available 3D marker is the templet square marker of the software library ARToolKit (Figure 16 a)). It consists of a square black border of known size that is used to estimate the 3D pose by image analysis. The sub-image in the interior of the square region contains a pattern that is used for identification via template matching [1].

Another reliable and robust marker type is the QR Code (Figure 16 b)). QR Codes can store a lot of information which is contained in both vertical and horizontal direction. The precise localization in 2D is accomplished through square position detection patterns located at three corners of the symbol. All three detection patterns must be visible to ensure a reliable result [44]. Combining several QR Codes to a single marker enables the extraction of multiple 3D information and therefore the usage of multiple QR codes as 3D markers, as shown in Figure 17 (details in section 4.4.3).

A different approach to creating a 3D marker is represented by using 2D Fiducial Propeller markers (Figure 16 c)) [56]. This marker type provides robust correspondences across views and subpixel precise localization. The subpixel precise center coordinates are calculated by using a saddle-point optimization of the propeller pattern that is placed at the marker center [11]. The rotationally invariant black and white circular code around the marker center is used for identification. Similar to the QR Code marker, several Fiducial Propeller markers have to be combined to form a 3D marker as shown in Figure 17 (details in section 4.4.4).

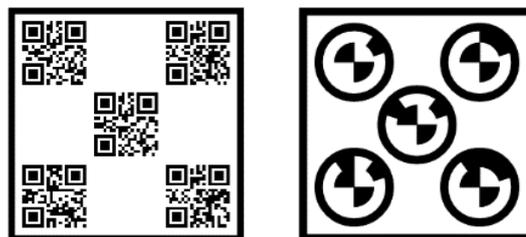


Figure 17: Multiple 2D QR Codes and 2D Fiducial Propeller markers combined as 3D markers

3. Background

This section provides background information on the methodology applied in this thesis. After discussing the representation of position and orientation, I will elaborate on details about monocular and binocular camera systems. The mathematical definition of the camera model as well as the mathematical background to the calibration process will be described. Furthermore, I will briefly inform the reader about the basics of forward and inverse kinematics to give an overview on how to transform joint angles to a robot pose and vice versa. Finally, I will examine the determination of the machine and process capability.

3.1. Position and Orientation Representation

Each body is uniquely defined in Euclidean space by six coordinates. To define the pose of a body, I make use of coordinate frames, which consist of an origin \mathbf{O}_i and a triad of mutually orthogonal basis vectors $[\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i, \hat{\mathbf{z}}_i]$. The pose of the body is always expressed as the pose of one coordinate system to another [66].

3.1.1. Position and Displacement

According to [66], the 3×1 vector of Equation (1) can be used to define the position of a coordinate frame i relative to a coordinate frame j .

$${}^j\mathbf{P}_i = \begin{bmatrix} {}^j p_i^x \\ {}^j p_i^y \\ {}^j p_i^z \end{bmatrix} \quad (1)$$

${}^j\mathbf{P}_i$ represents the Cartesian coordinates of the origin \mathbf{O}_i in frame j . A translation of a rigid body is a displacement in which all straight lines remain parallel to their initial orientations. Any displacement can be represented by using the notation of a position.

3.1.2. Orientation and Rotation

The representation of the basis vectors $[\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i, \hat{\mathbf{z}}_i]$ in terms of the basis vectors $[\hat{\mathbf{x}}_j, \hat{\mathbf{y}}_j, \hat{\mathbf{z}}_j]$ is used to describe the orientation of the frame i relative to frame j . jR_i is defined by the dot products of the basis vectors of the frames i and j (Equation (2)).

$${}^jR_i = \begin{bmatrix} \hat{\mathbf{x}}_i \cdot \hat{\mathbf{x}}_j & \hat{\mathbf{y}}_i \cdot \hat{\mathbf{x}}_j & \hat{\mathbf{z}}_i \cdot \hat{\mathbf{x}}_j \\ \hat{\mathbf{x}}_i \cdot \hat{\mathbf{y}}_j & \hat{\mathbf{y}}_i \cdot \hat{\mathbf{y}}_j & \hat{\mathbf{z}}_i \cdot \hat{\mathbf{y}}_j \\ \hat{\mathbf{x}}_i \cdot \hat{\mathbf{z}}_j & \hat{\mathbf{y}}_i \cdot \hat{\mathbf{z}}_j & \hat{\mathbf{z}}_i \cdot \hat{\mathbf{z}}_j \end{bmatrix} \quad (2)$$

Equation (3) describes the standard rotation of frame i around the three principal axis $\hat{\mathbf{x}}_j, \hat{\mathbf{y}}_j$ and $\hat{\mathbf{z}}_j$ [14].

$$\begin{aligned}
R_x(\alpha) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\alpha) & -s(\alpha) \\ 0 & s(\alpha) & c(\alpha) \end{bmatrix} \\
R_y(\beta) &= \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} = \begin{bmatrix} c(\beta) & 0 & s(\beta) \\ 0 & 1 & 0 \\ -s(\beta) & 0 & c(\beta) \end{bmatrix} \\
R_z(\gamma) &= \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c(\gamma) & -s(\gamma) & 0 \\ s(\gamma) & c(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{3}$$

These rotations can be combined through simple matrix multiplication. The resulting rotation matrix contains nine elements, although only three parameters are required for defining the orientation in space. In this work, each rotation matrix is built rotating around the Z-axis first, then the Y-axis and finally the X-axis (Equation (4)) [66].

$$\begin{aligned}
{}^jR_i &= R_x(\alpha)R_y(\beta)R_z(\gamma) = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \\
{}^jR_i &= \begin{bmatrix} c(\beta)c(\gamma) & -c(\beta)s(\gamma) & s(\beta) \\ c(\alpha)s(\gamma) + c(\gamma)s(\alpha)s(\beta) & c(\alpha)c(\gamma) - s(\alpha)s(\beta)s(\gamma) & -c(\beta)s(\alpha) \\ s(\alpha)s(\gamma) - c(\alpha)c(\gamma)s(\beta) & s(\alpha)c(\gamma) + c(\alpha)s(\beta)s(\gamma) & c(\alpha)c(\beta) \end{bmatrix}
\end{aligned} \tag{4}$$

This order of rotation matrix multiplication is denoted as Yaw-Pitch-Roll convention. Vice versa, the rotation matrix jR_i can be decomposed to extract the Euler angles α , β and γ . According to [19], there are three different cases to consider:

Case 1 ($\beta \in (-\pi/2, \pi/2)$):

$$\begin{aligned}
\alpha &= \text{atan2}(-R_{23}, R_{33}) \\
\beta &= \text{atan2}\left(R_{13}, \sqrt{(R_{23})^2 + (R_{33})^2}\right) \\
\gamma &= \text{atan2}(-R_{12}, R_{11})
\end{aligned} \tag{5}$$

Case 2 ($\beta = \pi/2$):

$$\begin{aligned}
\beta &= \pi/2 \\
\alpha + \gamma &= \text{atan2}(R_{10}, R_{11})
\end{aligned} \tag{6}$$

Case 3 ($\beta = -\pi/2$):

$$\begin{aligned}
\beta &= -\pi/2 \\
\alpha - \gamma &= \text{atan2}(R_{10}, R_{11})
\end{aligned} \tag{7}$$

The phenomenon of case 2 and 3 is called Gimbal lock. Due to the linking of α and γ , there is an infinite number of solutions. To find a single solution, it is convenient to set $\alpha = 0$ and compute γ as described in Equation (7). The vector $[\alpha \ \beta \ \gamma]$ can be used for a minimal representation of the orientation of coordinate frame i relative to frame j .

3.1.3. Homogenous Transformations

Homogenous transformation matrices are used to combine position vectors and rotation matrices in a compact notation [66]. If the previously addressed position ${}^j\mathbf{P}_i$ and rotation jR_i are known, any vector ${}^i\mathbf{Q}$ expressed relative to the coordinate frame i can be expressed relative to the frame j by Equation (8).

$$\begin{aligned} {}^j\mathbf{Q} &= {}^jR_i {}^i\mathbf{Q} + {}^j\mathbf{P}_i \\ \begin{bmatrix} {}^j\mathbf{Q} \\ 1 \end{bmatrix} &= \begin{bmatrix} {}^jR_i & {}^j\mathbf{P}_i \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} {}^i\mathbf{Q} \\ 1 \end{bmatrix} = {}^jT_i \begin{bmatrix} {}^i\mathbf{Q} \\ 1 \end{bmatrix} \end{aligned} \quad (8)$$

jT_i represents a 4×4 homogenous transformation matrix. The inverse ${}^jT_i^{-1} = {}^iT_j$ transforms vectors from coordinate frame j to coordinate frame i . Similar to the 3×3 rotation matrices, homogenous 4×4 transformation matrices can be composed by simple matrix multiplication (Equation (9)) [66].

$${}^kT_i = {}^kT_j {}^jT_i \quad (9)$$

3.2. Camera System

One of the main components for visual servoing is the camera system. It is used to gather information in the image scene to enable the servoing of the robot to the desired target. In this work, the camera system is a stereo system that extracts 3D information of the environment. The accuracy of the extracted data depends on the used camera equipment and the camera properties, which are determined by performing a calibration.

In this section, I will give a brief overview on the basics of the camera and lens selection depending on the system requirements. Furthermore, I will explain the background of the mathematical formulation and the calibration of a monocular and a binocular camera system. Additionally, I will introduce a method for a monocular system to characterize regions by the means of image moments.

3.2.1. Basics to Camera and Lens Selection

To enable accurate measurements with a camera system, the camera and the lens have to be chosen according to the following system requirements:

- Field of view (FOV)
- Camera distance (CD)
- Smallest observable feature (SOF)
- Subpixel Accuracy (SPA)
- Sampling factor (SF)

The required number of active pixels (AP), which is also known as resolution, can be determined as shown in Equation (10). The factor SF is a consequence of the sampling theorem that requires at least 2 pixels to resolve a feature, like an edge. The higher this sampling is, the more robust the measurements are going to be [53].

$$AP = SF \cdot \frac{FOV}{SOF} \cdot SPA \quad (10)$$

The number of active pixels is used to choose a camera with a specified active area (AA) and pixel size (PS). These values are further used to determine the required lens. The lens is used to project light rays on the image sensor. The focal length f of the lens has to be determined, so that all rays originating from a distinct point in the scene intersect in one point on the image plane (Equation (11)). Only then a sharp acquisition can be made [57].

$$f \cdot FOV = AA \cdot CD \rightarrow f = \frac{AA \cdot CD}{FOV} \quad (11)$$

The iris of the chosen lens is used to limit the amount of light. The less lens area is used for an acquisition, the fewer lens errors are incorporated (Figure 18).

As a consequence, the sharpness gets increased. The value of CD can vary a certain amount in keeping the object sharp, if the inaccuracies are limited to a blur spot (BS) of the size of 1 pixel. The deviation of CD without influencing the sharpness of the object is called depth of field (DoF), which can be calculated as shown in Equation (12) [57].

$$DoF = \frac{CD}{1 \pm BS \cdot I \cdot \frac{CD - f}{f^2}} \quad (12)$$

I is denoted as the amount of iris aperture. The value of DoF describes a far and near point of the object with respect to the camera for a sharp acquisition.



Figure 18: Influence of a changing amount of iris aperture on the blur spot of the image [57]

3.2.2. Monocular System – Background

The most specialized and simplest camera model is the basic pinhole camera which is described in [26]. Using this model, a point in 3D space $\mathbf{X} = (X, Y, Z)^T$ is mapped to the image plane, where the line between X and the center of projection meets the image plane. The plane is located at $Z = f$ where f is denoted as focal length.

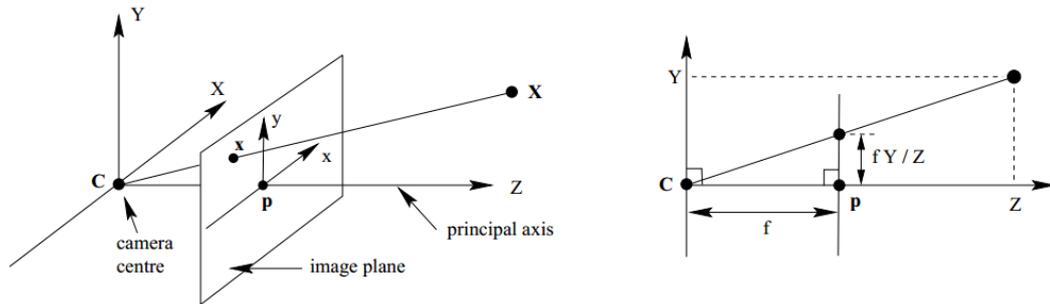


Figure 19: Pinhole camera – basic concept of mapping a 3D point X to the image plane [26]

By using similar triangles the image coordinates $(x, y)^T$ can be computed (Equation (13)).

$$(X, Y, Z)^T \mapsto (fX/Z, fY/Z)^T = (x, y)^T = \mathbf{x}^T \quad (13)$$

The central projection can be written as matrix multiplication using homogenous coordinates (Equation (14)). \mathbf{X} is now denoted as a homogenous 4-vector and the image point \mathbf{x} as a homogenous 3-vector. The projection matrix P is a 3×4 matrix.

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (14)$$

$$\mathbf{x} = P\mathbf{X}$$

The origin of the image plane coordinate system is usually located at one of the image corners and not at the principal point $P = (p_x, p_y)^T$. Therefore, the general mapping is extended to Equation (15).

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (15)$$

By introducing the camera matrix K and writing X as ${}^{cam}X$, Equation (15) may be expressed as Equation (16). X is written ${}^{cam}X$, to emphasize that this point is expressed in the camera coordinate system (CCS). I is the identity matrix.

$$x = K[I|0]{}^{cam}X \quad (16)$$

In general, 3D points will be expressed in relation to a different Euclidean coordinate frame, known as world coordinate system (WCS) (shown in Figure 20).

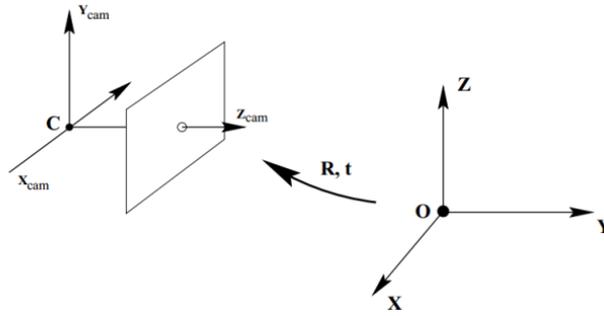


Figure 20: The Euclidean transformation using the rotation R and translation t to describe the relation between the CCS and the WCS [26]

This relation can be described by a translation and a rotation (Equation (17)). \tilde{C} represents the coordinate center of the camera in the WCS and R is the rotation matrix which defines the orientation of the CCS.

$${}^{cam}X = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} X \quad (17)$$

Combining Equation (16) and (17) leads to Equation (18). P denotes the general pin-hole camera matrix and has 9 degrees of freedom (3 for K , 3 for R and 3 for \tilde{C}).

$$x = KR[I|-\tilde{C}]X = PX \quad (18)$$

X now defines a point in WCS. The parameters contained in K are called intrinsic camera parameters, whereas R and \tilde{C} represent the extrinsic parameters. Usually Equation (18) is simplified by substituting $-R\tilde{C}$ with t (Equation (19)) [26].

$$\begin{aligned} {}^{cam}X &= RX + t \\ P &= K[R|t] \end{aligned} \quad (19)$$

In case of a charge-coupled device (CCD) camera, there is the possibility of having non-square pixels. Therefore, the general calibration matrix of a CCD camera is defined by Equation (20). This equation also consists of a parameter s which represents a skew parameter. This value is non-zero for pixel layouts with skewed axes or cameras with desynchronized pixel readout.

$$K = \begin{bmatrix} \alpha_x & s & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (20)$$

The new camera projection matrix $P = KR[I] - \tilde{C}$ is called finite projective camera matrix and has 11 degrees of freedom. Using this camera matrix, a 2D image point can be mapped to a line in 3D space, which connects the camera center C and the point P^+x (Equation (21)).

$$X(\lambda) = P^+x + \lambda C \quad (21)$$

P^+ is the pseudo inverse of the projection matrix P for which $PP^+ = I$ holds true [26].

3.2.3. Monocular System – Calibration Method

According to [27], the procedure of an accurate camera calibration can be separated into 3 steps. The first step is the initialization of the model to propose a starting point for the subsequent iterative search. One can define nominal values for the focal length, aspect ratio and the image center, or one can use the Direct Linear Transformation (DLT) to get an initial estimate of the projection matrix P , addressed in section 3.2.2. P is determined by taking advantage of corresponding 2D image points x_i and 3D scene points X_i . The task is to find a camera matrix P that fulfills Equation (22) for all i .

$$x_i = P X_i \quad (22)$$

Equation (22) involves homogenous vectors. Thus, x_i and $P X_i$ are not equal. They share the same direction, but may differ in magnitude which leads to Equation (23).

$$x_i \neq P X_i \quad x_i \times P X_i = 0 \quad (23)$$

This equation can be written as a linear system and solved by using the singular value decomposition SVD, if at least 6 point correspondences were found. After the calculation of P , the QR decomposition can be applied to determine the intrinsic and extrinsic camera parameters. Procedure details can be found in [26].

In the next step, a minimization of the weighted sum of square differences between the observations and the model is performed to estimate the parameters of the forward camera model (3D to 2D mapping). Assuming the usage of N circular points and K images, a vector can be formed that contains the observed image coordinates of the ellipse center n in frame k called $e_o(n, k)$. Furthermore, a corresponding vector can be built by using the forward camera model denoted by $e_d(n, k)$. According to [27], the parameters θ (intrinsic and extrinsic) are obtained by minimizing the objective function $J(\theta)$ (Equation (24)).

$$J(\boldsymbol{\theta}) = \mathbf{y}^T(\boldsymbol{\theta}) C_e^{-1} \mathbf{y}(\boldsymbol{\theta})$$

$$\mathbf{y}(\boldsymbol{\theta}) = [(\mathbf{e}_0(1,1) - \mathbf{e}_d(1,1))^T, (\mathbf{e}_0(2,1) - \mathbf{e}_d(2,1))^T, \dots, (\mathbf{e}_0(N,K) - \mathbf{e}_d(N,K))^T]^T \quad (24)$$

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

C_e is the covariance matrix of the observation error which depends on the sources of the measurement errors. According to [27], these errors are a result of:

- Insufficient projection model (not all distortion components are compensated)
- Illumination changes (substantial effect due to chromatic aberration)
- Camera electronics (horizontal shift, phase locked loop)
- Calibration target (unprecise manufacturing)

The optimization of Equation (24) can be solved by using a numerical technique, such as Levenberg-Marquardt.

The last step includes the creation of a backward camera model (2D to 3D mapping) to ensure a more consistent result in both directions. This is necessary because the distortion model is not the same for the forward and backward model (intrinsic parameters remain unchanged) [27].

3.2.4. Region Characterization using Geometric Moments

Consider the task of monocular object detection or recognition. The main goal is to describe an object in a manner that is independent of scale, position and orientation. This can be achieved by the introduction of moments that are derived from the moment theory used in statistics and mechanics. According to [52], moments describe numeric quantities at some distance from a reference point or axis. The two-dimensional moment of order $(p + q)$ for a discretized image $f(x, y)$ is denoted as shown in Equation (25).

$$m_{pq} = \sum_{j \in R} \sum_{i \in C} x_i^p y_j^q f(i, j) \quad (25)$$

R and C represent all rows and columns of the extracted region. The properties of low-order moments represent well known geometric properties. The zero order moment m_{00} represents the total object area for a silhouette image of a segmented object (Equation (26)).

$$m_{pq} = \sum_{j \in R} \sum_{i \in C} f(i, j) \quad (26)$$

The two first order moments m_{10} and m_{01} describe the coordinates of the center of mass (Equation (27)).

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (27)$$

\bar{x} and \bar{y} can be used to define a unique location of an object. If the center of mass of an object coincides with the origin of the extracted region, such that $\bar{x} = 0$ and $\bar{y} = 0$, then

the computed moments are referred to as central moments. According to [38], central moments of the order $(p + q)$ can be expressed as depicted in Equation (28).

$$\mu_{pq} = \sum_{j \in R} \sum_{i \in C} (x_i - \bar{x})^p (y_j - \bar{y})^q f(i, j) \quad (28)$$

Equation (29) holds true for central moments.

$$\mu_{00} = m_{00} \quad \mu_{10} = 0 \quad \mu_{01} = 0 \quad (29)$$

The second order moments μ_{20} , μ_{02} and μ_{11} can be compared to the moments of inertia and may be used to determine the orientation and the principal axis of an image feature. The principal axes describe two orthogonal axes around which the object can be rotated with minimal (axis a) and maximal (axis b) inertia. Their calculation can be performed by using the inertial tensor J (Equation (30)).

$$J = \begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix} \quad (30)$$

The direction of the axis is represented by the eigenvectors and the length by the eigenvalues of J [46]. The calculation of the eigenvalues is shown in Equation (31). Note that if $\mu_{11} = 0$, then $\lambda_1 = \mu_{20}$ and $\lambda_2 = \mu_{02}$.

$$\lambda_{1,2} = \frac{1}{2} \left(\mu_{20} + \mu_{02} \pm \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \right) \quad (31)$$

According to [52], the orientation θ between the x axis and the semi-major axis a is calculated as shown in Equation (32). θ lies in the range of $-\pi/4 \leq \theta \leq \pi/4$.

$$\theta = \frac{1}{2} \arctan \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \quad (32)$$

Equation (31) and (32) define an image ellipse. This ellipse has the same principal axes direction and moments of inertia as the extracted region (Figure 21).

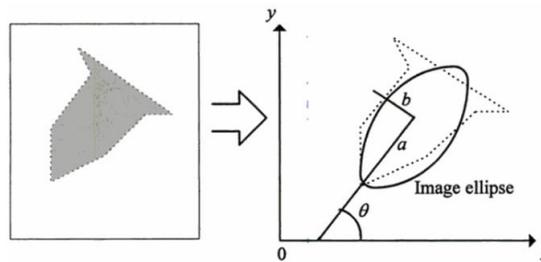


Figure 21: Image ellipse calculation (right) of an extracted region (left) via second order moments for object approximation [46]

The length of the semi-major axis a and semi-minor axis b of the ellipse is calculated as given in Equation (33) [46].

$$a = 2 \sqrt{\frac{\lambda_1}{\mu_{00}}} \quad b = 2 \sqrt{\frac{\lambda_2}{\mu_{00}}} \quad (33)$$

3.2.5. Binocular System – Background

Despite the fact that many 3D detections could be performed with a monocular camera system, a binocular stereo system delivers more reliable and accurate 3D pose measurements. A stereo system is based on triangulation methods, whereas a monocular system uses minimization methods to fit a known geometric model optimally to the acquired image. These minimizations can diverge severely from the actual data and strongly depend on illumination, focus and object to model correspondence.

According to [26], the basic concept of stereo vision can be described by the epipolar geometry. The epipolar geometry describes the intrinsic projective geometry between two views and depends only on the internal parameters of the cameras and their relative pose to each other.

A 3D point X , that is observed by two cameras with their optical centers at C and C' , is imaged as x in the first and x' in the second view. Figure 22 shows that X , C , C' , x and x' lie in the same plane π . Even the back-projected rays from x and x' that intersect in X are coplanar to π . This property is most significant for the search of point correspondences [26].

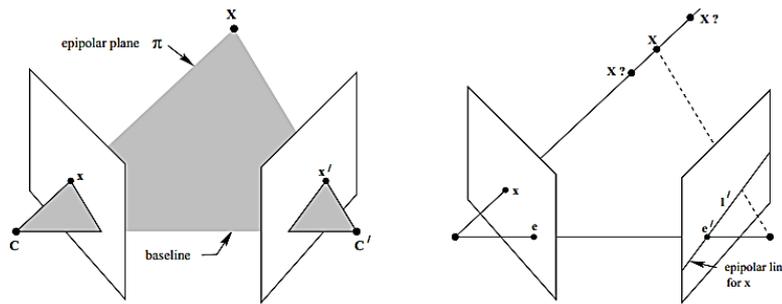


Figure 22: Epipolar geometry - point correspondence geometry (adapted from [26])

The line joining the two camera centers C and C' is called baseline and intersects the two image planes at the epipoles e and e' . Any plane π that contains the baseline and therefore e and e' is called epipolar plane. The epipolar plane π intersects the image planes in the epipolar lines l and l' . If the point x' is unknown, one can take advantage of the fact that the ray corresponding to x' lies in the plane π and consequently lies on the line l' . l' is the back-projected ray of x in the second image and is called the epipolar line of x . The search for x' , the corresponding point to x , can be reduced to a search on the line l' . The mapping of x to l' can be represented by the fundamental matrix F , which is a projective mapping from points to lines [26]. F holds Equation (34) for any point pair in the two images.

$$\mathbf{x}'^T F \mathbf{x} = 0 \quad (34)$$

Equation (35) contains the calculation of the epipoles and all epipolar lines.

$$\begin{aligned} \mathbf{l}' &= F \mathbf{x} & \mathbf{l} &= F^T \mathbf{x}' \\ F \mathbf{e} &= 0 & F^T \mathbf{e}' &= 0 \end{aligned} \quad (35)$$

According to [26], F can be described in terms of the two camera projection matrices P, P' . Introducing P^+ as the pseudo-inverse of P , F can be algebraically determined using Equation (36).

$$F = [e']_x P' P^+ \quad (36)$$

F can be used to apply a rectification to both stereo images, in order to match the epipolar lines. Rectification is the process of resampling pairs of stereo images to produce projections in which the epipolar lines run parallel to the X-axis (epipoles are at infinity). Consequently, disparities between images are in X-direction only.

The calculation of a rectification mapping is quite simple [26]:

- Use the fundamental matrix F to determine the epipoles e and e' .
- Choose a projective transformation T' to map the epipole e' to infinity.
- Determine the projective transformation T by minimizing the sum of squared distances of Equation (37).
- Resample the first image according to T and the second image according to T' .

$$\sum_i d(Tx_i, T'x'_i)^2 \quad (37)$$

Now, the determination of the 3D coordinates of a known image point x is performed. As mentioned before, the correspondence search in the second image is simplified by taking advantage of the epipolar geometry to a search along the line l' . Due to some errors in the measured image coordinates and probably because of some inaccuracies in the camera projection matrices, the back-projected rays of x and x' will not intersect in general. Thus, the image points do not satisfy the epipolar constraint of Equation (34) (illustrated in Figure 23).

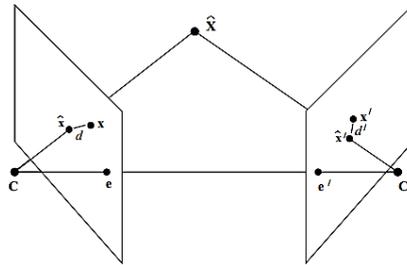


Figure 23: Epipolar constraint - deviations due to errors in the measured image coordinates [26]

The task is to find two image points \hat{x} and \hat{x}' that minimize the cost function of Equation (38).

$$C(x, x') = d(x, \hat{x})^2 + d(x', \hat{x}')^2 \text{ subject to } x'^T F x = 0 \quad (38)$$

The Euclidean distance between points is represented by $d(*,*)$. The cost function of Equation (38) is equivalent to minimizing the reprojection error of \hat{X} . The image points \hat{x} and \hat{x}' are mapped to the image by projection matrices P and P' , that are consistent with F . Once \hat{x} and \hat{x}' are found, the 3D point can easily be determined by any triangulation method, since the rays of both image points will meet precisely [26].

3.2.6. Binocular System – Calibration Method

As pointed out by [6], the calibration of a stereo camera is very similar to the calibration of the single camera. Each calibration image has to be taken pairwise (calibration plate at one pose has to be observed by cameras as displayed in Figure 24).

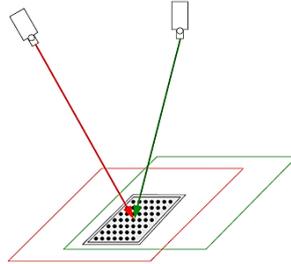


Figure 24: Stereo calibration - image acquisition of the calibration plate (adapted from [51])

First, each camera has to be calibrated separately to determine the intrinsic and extrinsic parameters of each camera with respect to the calibration target using the calibration method addressed in section 3.2.3. The results are the parameters for a single intrinsic matrix K and multiple extrinsic parameters R_i and t_i for each pose of the calibration plate. It is common for multi-camera setups to assume that K is a robust estimate since the calculation of K uses all calibration plate positions. R_i and t_i on the other hand are determined for each input, whereby their accuracy depends on how well the pattern was detected in a single image.

Commonly, the reprojection error is used to evaluate the camera calibration. A low reprojection error indicates an accurate projection matrix. The main problem using the reprojection error for a stereo calibration is that extrinsic parameters and therefore the world coordinate system are defined for the calibration plane i_1 . The 3D location of the point Q on grid i_2 is not available, and so the reprojection error of Q cannot be computed (visualized in Figure 25 a)).

Hence, a qualitative error measurement is introduced which uses the rectification explained in section 3.2.5. The epipolar constraint defines that a point Q that is projected onto the rectified versions of the left and right image should lie on the same scanline (see Figure 25 b) and c)). Otherwise, the calibration is inaccurate. This fact is independent of the 3D location of Q . Thus, all points from all calibration plate locations that are observed in both images can be taken to evaluate the calibration error [6].

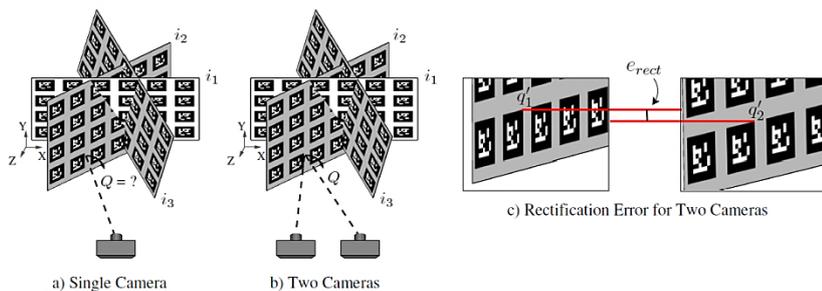


Figure 25: Stereo calibration - evaluating the extrinsic parameters using calibration grid i_1 [6]

Equation (39) describes a measure of the rectification error for two cameras c_1 and c_2 and a calibration plate view i . The unknown 3D point Q^k of the k^{th} detected grid point corresponds to $q_1^k = (u_1^k, v_1^k)$ on image plane c_1 and to $q_2^k = (u_2^k, v_2^k)$ on image plane c_2 . $q_c^k[1]$ is denoted to refer to v_c^k for $c \in \{1,2\}$. $T_i^{c_1}$ is denoted as the rectifying transformation for camera c_1 using calibration i . $T_i^{c_2}$ is defined similarly for c_2 [6].

$$e_{rect}^{c_1}[i] = \frac{1}{N} \sum_{j=1}^N \left(\frac{1}{M_j} \sum_{k=1}^{M_j} |(T_i^{c_1} q_1^k)[1] - (T_i^{c_2} q_2^k)[1]| \right) \quad (39)$$

N is the total number of the different calibration plate views and M_j is the number of calibration points for the plate position j . More accurate stereo calibrations can be determined by choosing the single calibration target that yields the lowest average rectification error. Thus, out of all calibration grids S , the one grid i that minimizes Equation (40) is chosen [6].

$$\min_{i \in S} \sum_{c \in \{c_1, c_2\}} e_{rect}^c[i] \quad (40)$$

The optimal extrinsic parameter can be used to build the relationship ${}^{c_1}T_{c_2}$ between camera c_1 and c_2 . ${}^{c_1}T_{c_2}$ describes the matrix that transforms a 3D point in the coordinate system of c_2 to the coordinate system of c_1 .

3.2.7. Binocular System – Depth Resolution

According to [22], the calculation of a depth value with a stereo system can be performed by using triangulation (Equation (41)). B is the base line between the cameras, f the focal length and d the difference in X coordinates of two corresponding pixels in rectified images. d is also referred to as disparity.

$$z = \frac{Bf}{d} \quad (41)$$

The deviation of the depth measurement Δz in a stereo system can be written in terms of the disparity error Δd (Equation (42)). A more detailed derivation of Equation (42) can be found in [8].

$$\Delta z = \frac{Bf}{d} - \frac{Bf}{d + \Delta d} = \frac{z^2 \Delta d}{Bf + z \Delta d} \approx \frac{z^2}{Bf} \Delta d \quad (42)$$

Δz can be separated into the correspondence error Δd and into a geometric resolution $z^2/(Bf)$. The geometric resolution includes errors that are a result of the geometric setup of the stereo system. It includes the baseline, focal length and depth. To achieve a good distance resolution, B and f of the setup should be large, whereas z should be as small as possible. The correspondence error describes the inaccuracies due to incorrect matches or the lack of subpixel accuracy.

Figure 26 shows the development of the distance resolution as a function of the distance for different configurations. The disparity error is estimated to be $1 \mu\text{m}$.

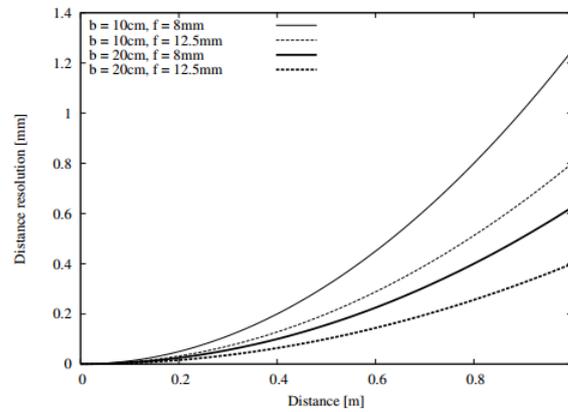


Figure 26: Distance resolutions as a function of distance for four different configurations of focal length and base lines [51]

The ratio between baseline and height should be $B = z$ maximum. [51] addresses this topic and states that the optimal ratio depends on the surface characteristics of the object. If the height differences on an object are large, the ratio should be smaller than $B = z$. Typically, one chooses a ratio between $B = z/3$ and $B = z/4$.

3.3. Robot Kinematics

Robotic mechanisms are systems of rigid bodies connected by joints. The pose of these rigid bodies are described by the robot kinematics. The notation of the system and the attachment of the frames follow the Denavit-Hartenberg convention which is visualized in Figure 27 and described by the following steps [66]:

1. Each joint i is located and labeled with $\hat{z}_0, \dots, \hat{z}_{n-1}$.
2. The base frame is established anywhere on the \hat{z}_0 axis.
3. The origin O_i is at the intersection of \hat{z}_i and the common normal of \hat{z}_{i-1} to \hat{z}_i .
4. The \hat{x}_i axis is located along the common normal between \hat{z}_{i-1} and \hat{z}_i through O_i .
5. a_i denotes the distance along \hat{x}_i from O_i to the intersection of \hat{x}_i and \hat{z}_{i-1} axes.
6. d_i is the distance along \hat{z}_{i-1} from O_{i-1} to the intersection of \hat{x}_i and \hat{z}_{i-1} axes.
7. α_i denotes the angle from \hat{z}_{i-1} to \hat{z}_i about \hat{x}_i .
8. θ_i denotes the angle from \hat{x}_{i-1} to \hat{x}_i about \hat{z}_{i-1} .

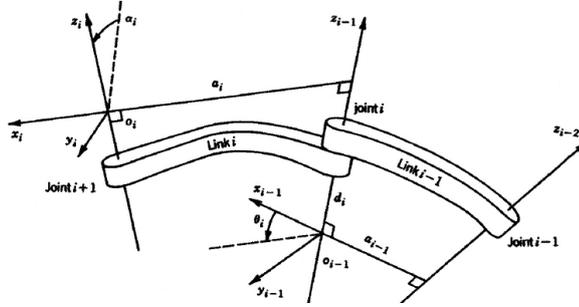


Figure 27: Denavit-Hartenberg convention for a serial chain manipulator [59]

Using this convention, a coordinate frame i can be described relative to the coordinate frame $i - 1$ by the homogenous transformation matrix of Equation (43).

$$\begin{aligned}
 {}^{i-1}T_i &= Rot(\hat{z}_{i-1}, \theta_i) Trans(\hat{z}_{i-1}, d_i) Trans(\hat{x}_i, a_i) Rot(\hat{x}_i, \alpha_i) = \\
 &= \begin{bmatrix} c(\theta_i) & -s(\theta_i) & 0 & 0 \\ s(\theta_i) & c(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c(\alpha_i) & -s(\alpha_i) & 0 \\ 0 & s(\alpha_i) & c(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} c(\theta_i) & -s(\theta_i)c(\alpha_i) & s(\theta_i)s(\alpha_i) & a_i c(\theta_i) \\ s(\theta_i) & c(\theta_i)c(\alpha_i) & -c(\theta_i)s(\alpha_i) & a_i s(\theta_i) \\ 0 & s(\alpha_i) & c(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (43)
 \end{aligned}$$

The kinematics of the robot can be divided into forward and inverse kinematics. The relationship between these two kinematic directions can be illustrated as follows in Figure 28.

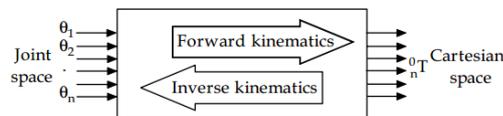


Figure 28: Relationship between forward and inverse kinematics [41]

3.3.1. Forward Kinematics

According to [66], forward kinematics describes the problem of calculating the pose and orientation of the end-effector of the manipulator relative its base (0T_N). The positions of all joints and the values of the geometric link parameters are given. This problem is solved by calculating the homogenous transformation between the tool and station frames. Assuming that the manipulator has N joints and the transformations ${}^{i-1}T_i$ between these joints are known, 0T_N can be described as shown in Equation (44).

$${}^0T_N = {}^0T_1 {}^1T_2 \dots {}^{N-2}T_{N-1} {}^{N-1}T_N \quad (44)$$

A small example of a two-linked planar manipulator is illustrated in Figure 29.

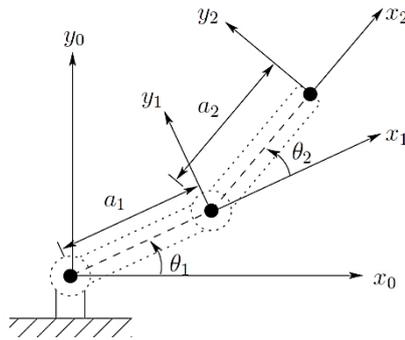


Figure 29: Two-link planar manipulator [59]

The link parameters are defined by Table 2.

Link	a_i	α_i	d_i	θ_i
1	a_1	0	0	θ_1
2	a_2	0	0	θ_2

Table 2: Link parameters of two-link planar manipulator of Figure 29

Using Table 2, one can build the homogenous transformation matrix for each link according to Equation (43) and combine them as depicted in Equation (44) to get the relation 0T_2 between the base and the robot end-effector (Equation (45)).

$${}^0T_2 = \begin{bmatrix} c(\theta_1 + \theta_2) & -s(\theta_1 + \theta_2) & 0 & a_1 c(\theta_1) + a_2 c(\theta_1 + \theta_2) \\ s(\theta_1 + \theta_2) & c(\theta_1 + \theta_2) & 0 & a_1 s(\theta_1) + a_2 s(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (45)$$

The first two entries of the last column of 0T_2 are the x and y coordinates of the origin with respect to the base frame [59].

3.3.2. Inverse Kinematics

Vice versa to the forward kinematics, the inverse kinematics describes the problem to find the joint values to a given position and orientation of the end-effector relative to the base. It is needed to control the manipulators to a given pose. The tasks that are performed by robots are given in Cartesian space which consists of a position vector and an orientation matrix. The actuators of the robot, however, operate in joint space which is represented by joint angles [41].

Considering the example shown in Figure 29, the inverse kinematic can be used to determine the joint angles θ_1 and θ_2 . The calculations to this are illustrated in Equation (46) and (47).

$$\begin{aligned}
 x^2 + y^2 &= a_1^2 + a_2^2 - 2a_1a_2\cos(\pi - \theta_2) \\
 \cos(\theta_2) &= \frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1a_2} := D \\
 \sin(\theta_2) &= \pm\sqrt{1 - D^2} \\
 \theta_2 &= \tan^{-1}\left(\frac{\pm\sqrt{1 - D^2}}{D}\right)
 \end{aligned} \tag{46}$$

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{a_2\sin(\theta_2)}{a_1 + a_2\cos(\theta_2)}\right) \tag{47}$$

The inverse kinematics is more difficult to solve than the forward kinematics. It requires the solution of non-linear sets of equations and delivers one, multiple or no solution. Even in the case of the two-link planar manipulator, the inverse kinematic delivers two results (Figure 30). Each solution has to be checked to see if they bring the end-effector to the desired position. A detailed derivation and explanation of the inverse kinematics for serial six-degree-of-freedom manipulators is shown in [48].

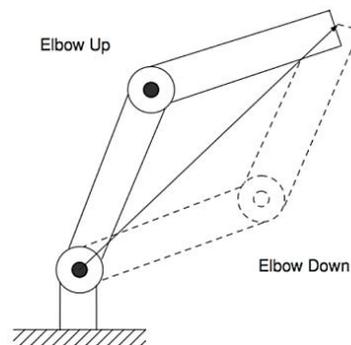


Figure 30: Inverse kinematics – two solutions for the same pose [59]

3.4. Machine and Process Capability

The measurements in this work are evaluated by using the machine and process capability indices C_m and C_{mk} [4]. These values are formed by using the mean (*mean*) and the standard deviation (*Std.Dev*) of the recorded data of a short term study. The data is assumed to be normally distributed (Equation (48)).

$$mean = \frac{1}{n} \cdot \sum_{i=1}^n x_i \quad Std.Dev = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^n (x_i - mean)^2} \quad (48)$$

The C_m index is used to describe the machine capability. The higher the value of C_m , the better the machine. It is a relationship between the spread of measurements (*Std.Dev*) and a given tolerance width T . If C_m is equal to one, the spread is equal to the tolerance width. The normal requirement of a machine is that the C_m is larger or equal to 1,67. The calculation of C_m is shown in Equation (49) [4].

$$C_m = \frac{T}{6 \cdot Std.Dev} \geq 1,67 \quad (49)$$

The C_m index does not consider the possibility that the spread is off-center [4]. It does not take into account where the spread is positioned in relation to the lower specification limits (LSL) and upper specification limits (USL). The C_{mk} on the other hand describes the capability corrected for position. If the C_{mk} is very high, the measurements have a small spread in relation to the given tolerance and are also well centered within that width. The C_{mk} index is determined as shown in Equation (50).

$$C_{mk} = \min\left(\frac{USL - mean}{3 \cdot Std.Dev}, \frac{mean - LSL}{3 \cdot Std.Dev}\right) \geq 1,67 \quad (50)$$

The relation between the C_m and C_{mk} index is visualized in Figure 31.

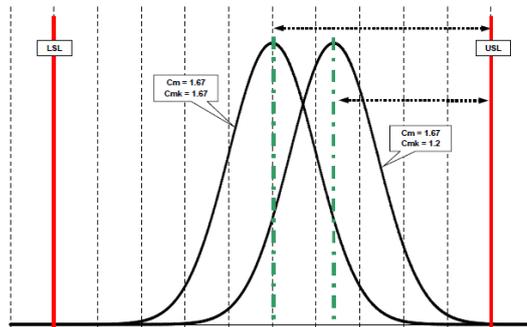


Figure 31: Machine and process capability – comparison between the C_m and C_{mk} index (adapted from [4])

4. Methodology

In this thesis, I concentrate on a 6 axis manipulator that is mounted on a mobile platform and performs simple pick and place operations of a known object. The robot consists of small deviations in the kinematic chain, which leads to inaccurate movements that need to be compensated. To ensure a proper movement and an accurate handling of the object, the robot is equipped with a stereo vision. Providing a robot with the ability of sight leads to a system that is aware of the environment's state. This system can react to environmental changes in case the robot base moves. The robot does not need to know the target coordinates a priori. Thus, the process of robot teaching is eliminated. Deficiencies in assemblies without knowledge of the exact location of the robot/object can be compensated by visual servoing. Visual servoing is a closed-loop control of a robot system. It uses vision as the underlying sensor and processes the visual feedback to guide the manipulator step by step to its target position.

In this work, the concept of visual servoing becomes enhanced by the usage of the industrial marker to object map to robustly detect the object. Moreover, a linear extrapolation is added to overcome errors of the robot kinematics. The overall workflow to realize a pick/place handling at a workstation called W1 is visualized in Figure 32.

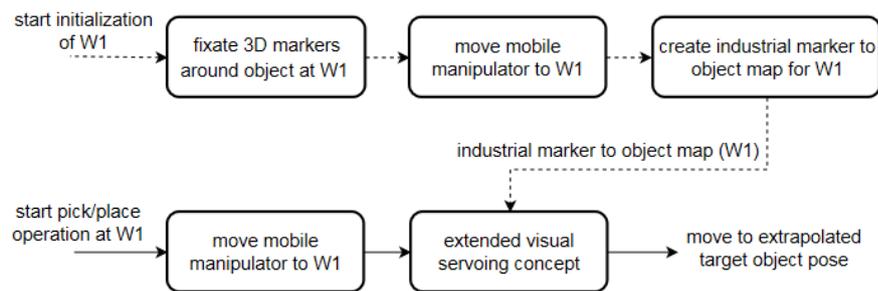


Figure 32: Overall workflow to realize a pick/place handling at a workstation called W1

Before the extended visual servoing concept can be applied, the workstation has to be initialized. At this initialization process, the industrial marker to object map is created. To achieve this, several 3D markers are manually fixated around the static target object. Then, the mobile manipulator has to be moved to the workstation to conduct several observations of the 3D markers and the object from different poses. The calculated relations between the markers and the object represent the map. These relations are then further used as a basis for the extended visual servoing concept, when the mobile manipulator performs pick and place tasks at W1.

In the beginning of this section, I will give details on the extended visual servoing concept and why this concept was developed. Furthermore, I will explain the creation of a reliable industrial marker to object map. Then, the object and marker detection algorithms, which are suitable to create the industrial marker to object map, will be evaluated and outlined in detail. Afterwards, I will concentrate on the controlling of the robot using 3D object information with respect to the camera. Finally, the extrapolation of the robot target pose in order to compensate kinematic errors will be discussed.

4.1. Extended Visual Servoing Concept

The extended visual servoing concept is based on the concept of visual servoing, which consists of two main procedures (shown in Figure 33): tracking and controlling.

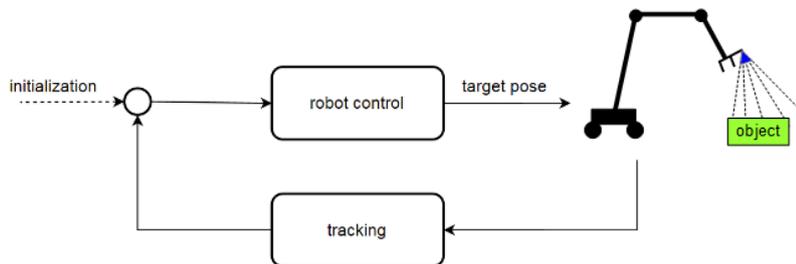


Figure 33: Basic visual servoing concept (adapted from [39])

The tracking process provides a continuous estimation of the object pose during the robot motion. This data serves as a basis to generate a control sequence that moves the robot in direction of the target. Furthermore, the system may require an initialization which commonly includes providing data for the object detection or information about an initial robot position where object observations can be started [39].

The extension of the visual servoing concept can be split up into three steps:

- Defining the robot control and tracking process
- Extending the tracking process by an industrial marker to object map
- Extending the robot control process by an extrapolation based on gathered data

In the first step, the robot control and the tracking processes are defined (Figure 34).

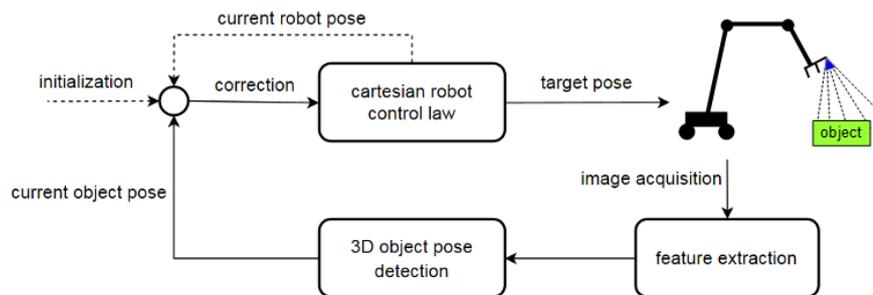


Figure 34: Extended visual servoing concept. Step 1 – define details of the robot control and tracking process (adapted from [39])

The robot control process is specified to be a Cartesian robot controller, which is fed with correction values that are calculated as a difference between the current object and robot pose. This correction is then compensated by driving the manipulator to the new target pose. The tracking process consists of a feature extraction that gathers important information of the current scene and delivers it to the subsequent 3D object detection. Depending on the given initial and extracted information, the 3D object pose can be estimated using the methods mentioned in section 2.

Each of the addressed 3D object detection algorithms is based on optimization methods and can therefore result in a local solution which deviates from the desired global solution. These algorithms strongly depend on optimal illumination as well as a unique and focused representation of the object. If those conditions are not held, the estimation of the pose can severely deviate from the actual values. As a consequence, I extend the 3D object detection with the industrial marker to object map.

At first, the workstation W1 is initialized by fixating M 3D markers around the static object pose. Then, the mobile manipulator is moved to W1 to carry out N observations $\{j \in \mathbb{N} \mid 0 < j \leq N\}$ of the M 3D markers and the object. The mean of the N detections is used to create the industrial marker to object map. Each observation is conducted at a different camera pose. The initialization process is illustrated in Figure 35.

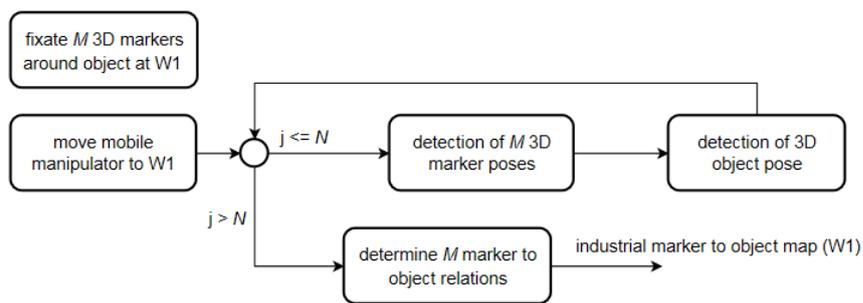


Figure 35: Extended visual servoing concept. Step 2 – creation of the industrial marker to object map at a workstation called W1

The determined poses of the M 3D markers and the static object are combined to build the relation between each marker and the object. These M relations define the industrial marker to object map.

After the initial setup, the object target pose at W1 can now be determined by combining 3D marker poses and the industrial marker to object map. This map enables a more accurate and robust localization of the object. Instead of one pose estimation, it is now possible to use optimal pre-defined relations to build a robust object pose by simply detecting 3D markers (Figure 36). Further details on the creation and the usage of the industrial marker to object map are explained in section 4.2.

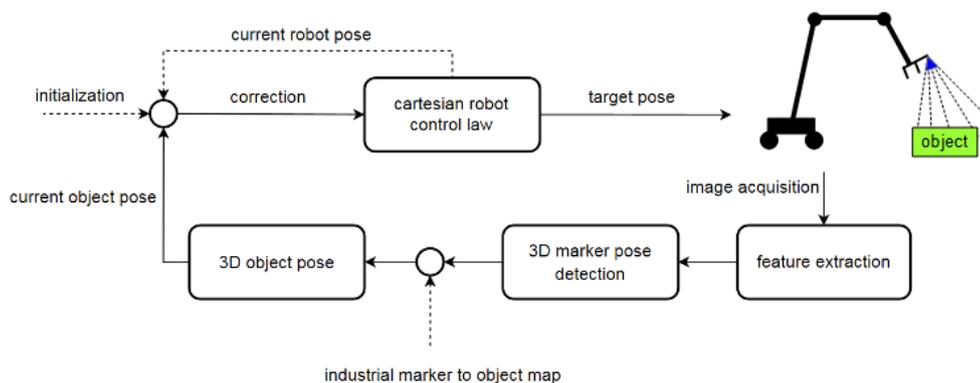


Figure 36: Extended visual servoing concept. Step 2 – 3D object detection via an industrial marker to object map (adapted from [39])

The final extension of the visual servoing process is an extrapolation algorithm for the robot control (Figure 37). This step is introduced to compensate inaccurate robot movements due to errors of the robot's kinematic chain and also to eliminate small deviations of the 3D marker detection.

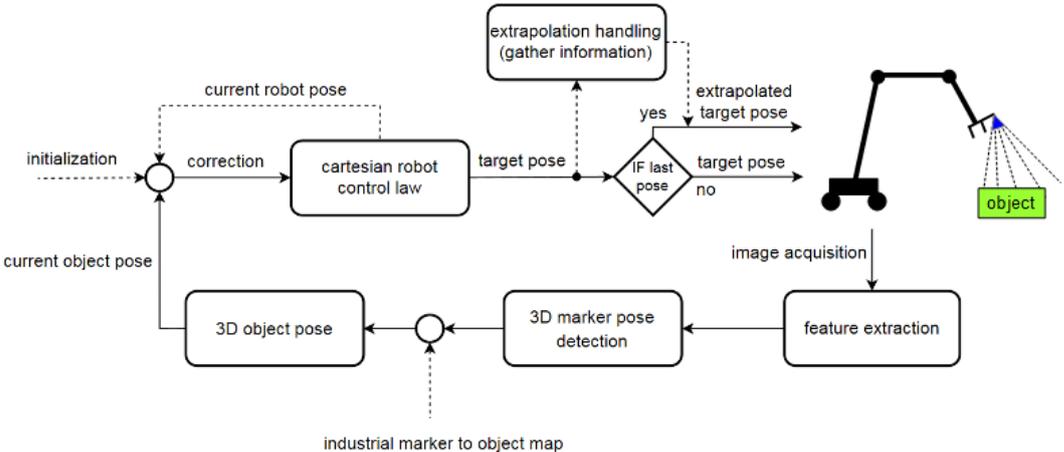


Figure 37: Extended visual servoing concept. Step 3 – add an extrapolation handling to the detailed robot control process (adapted from [39])

While the robot approaches the object, the extrapolation handling records the target poses of the robot control. The final step of the robot movement is predicted on the basis of all gathered information during the object approach. More information to the concept of the robot control can be found in section 4.5.

4.2. Industrial Marker to Object Map

As mentioned before, 3D object detection is not robust with respect to illumination or small deviations between the CAD and the real object model. Thus, I perform a supervised detection of a static object from different viewing angles and put the detections with respect to the static 3D markers defined in section 4.4. This information can later be used to extract the 3D pose of the object by simply detecting one or multiple 3D markers (Figure 38).

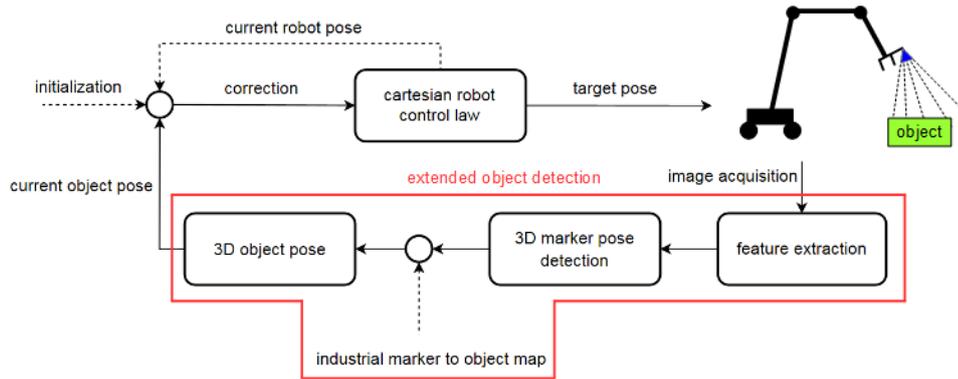


Figure 38: Extended visual servoing concept. Step 2 – extended object detection by using the industrial marker to object map (adapted from [39])

The relation between the 3D object and marker is defined in an initial step, which is visualized in Figure 35. In this step, N acquisitions of the scene are taken from different camera positions with the object and M markers in the field of view (Figure 39). Like before, $({}^cP_o, {}^cT_o)/({}^cP_m, {}^cT_m)$ represent the 3D pose P and the corresponding transformation matrix T from the camera center to the object/marker coordinate system.

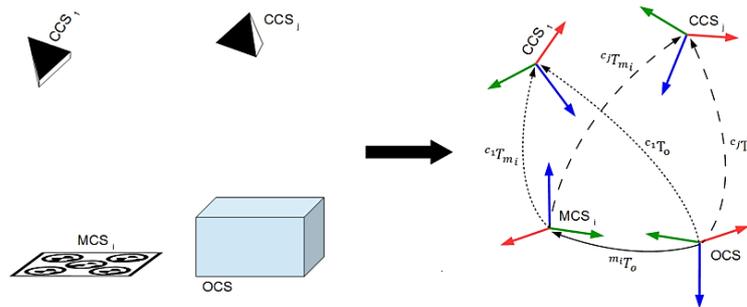


Figure 39: Industrial marker to object map – simultaneous observation of marker and object at different camera positions (left); observation in terms of coordinate systems (right)

The main task is to determine the transformation ${}^{m_i}T_o$ between the marker coordinate system $MCS_i \{i \in \mathbb{N} \mid 0 < i \leq M\}$ and the object coordinate system OCS . The camera coordinate system is denoted as $CCS_j \{j \in \mathbb{N} \mid 0 < j \leq N\}$. The determined transformation cT_o can be evaluated by projecting the object contour into the image. cT_o is valid only if the projected contours match the object in the image. The same evaluation

has to be performed by comparing the projected 3D marker center ${}^{m_{ij}}T_o$ with the actual 2D marker center in the image. The calculation of ${}^{m_{ij}}T_o$ is defined by Equation (51).

$${}^{m_{ij}}T_o = {}^{c_j}T_{m_i}^{-1} {}^{c_j}T_o \quad (51)$$

To increase the robustness, one can form the mean of ${}^{c_j}T_o$ and ${}^{m_{ij}}T_o$ from multiple acquisitions at the same camera position j . After gathering multiple ${}^{m_{ij}}T_o$ from N different camera positions, one can build the mean of all ${}^{m_{ij}}T_o$ to get a robust marker to object transformation ${}^{m_i}T_o$.

$${}^{m_i}T_o = \frac{1}{N} \sum_{j=1}^N {}^{m_{ij}}T_o \quad (52)$$

The built relations between the markers and the object can be used to determine the object pose without detecting the object itself. The object pose can be extracted by detecting at least one of the M 3D markers. Using more than one marker enables a better compensation of inaccuracies of the 3D marker center (Figure 40).

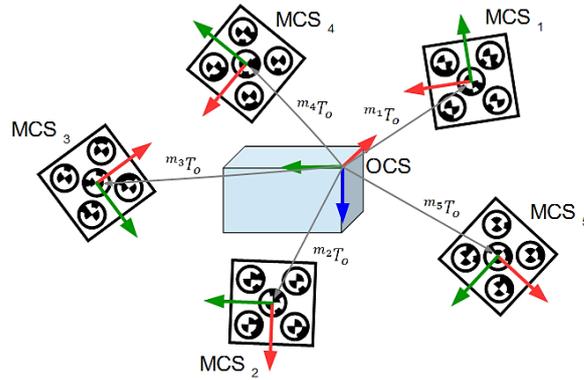


Figure 40: Industrial marker to object map –relations ${}^{m_i}T_o$ between multiple markers and the object

Considering Figure 40, the object pose can be determined by detecting M 3D marker poses and using Equation (53).

$${}^{c}T_o = \frac{1}{M} \sum_{i=1}^M {}^{c}T_{m_i} {}^{m_i}T_o \quad (53)$$

The markers can be placed with an arbitrary position and orientation around the object. The main restriction of this method is that the markers and the object are in the same field of view to define their relation in the pre-processing step.

4.3. Object Detection – Shape-Based 3D Matching

The creation of the industrial marker to object map requires an object detection algorithm that delivers precise results when providing optimal environmental conditions. The ‘Surface-Based Matching’, addressed in section 2, is excluded from the detection methods due to the requirement of a complete 3D surface scan and the associated inaccuracy because of the 3D matching of the model in a point cloud. Furthermore, I want to handle objects without texture and structures that can uniquely identify the part. Therefore, ‘Shape-Based 3D Matching’ (SBM) represents the only 3D object detection method mentioned in section 2 that can fulfill those requirements.

In the next chapters I will explain details to the concept of SBM and how to improve the accuracy by refining the detection result with a stereo system and with the usage of local features.

4.3.1. Basic Concept of Shape Based 3D Matching (SBM)

As described in section 2, the SBM, developed by [64], can be used to determine and identify a 3D object in a single camera image. Using a 3D CAD model, a hierarchical model is automatically generated, which enables the algorithm to robustly handle objects with no texture, as well as shiny or reflective surfaces. A prerequisite for an accurate and robust extraction of 3D information from the image is a precise geometric camera calibration. In the following, I will describe how to create a hierarchical model to detect the metallic clamp shown in Figure 41 by using the concept of [64].

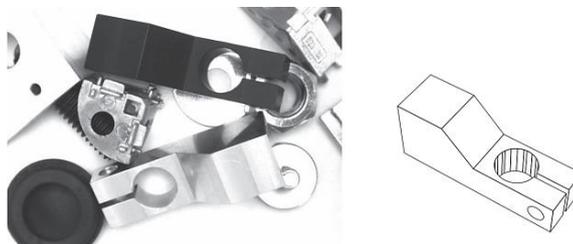


Figure 41: Shape-Based Matching - image of two different colored clamps (left) and the CAD model of the clamp (right) [64]

One of the main components that are used in this algorithm is the similarity measure. It is installed to find a transformed model image (translated, rotated, scaled) within a search image. This method uses brute force template matching between edges and gradients of a created 2D model image and the search image. It is invariant to contrast changes and can handle occlusions.

The creation of the hierarchical model, which is the basis component of the detection algorithm, can be separated into the view generation and the model image generation. The view generation includes the automatic creation of different views to build the hierarchical model. The object is placed at the center of a virtual sphere which defines a spherical coordinate system. A virtual camera is placed around the 3D object within a predefined pose range and projects the object at each pose into the image plane of the

virtual camera. The cameras always point to the center of the sphere. The blue section, visualized in Figure 42, shows a user defined pose range, which is limited by a longitude and latitude angle and a minimum and maximum distance to the center. The sampling of the view within the pose range is automatically determined during the generation process with the aim to maximize robustness and speed. Furthermore, an image pyramid is used to generate multiple layers of the 2D models. The algorithm starts at the lowest image pyramid (Figure 42 – Level 1) and applies an oversampling of views (high similarity between neighboring views). Then the similarity between all views is computed. The oversampled views are merged till the highest similarity measure is smaller than a threshold T_{merge} . The remaining views represent the aspects. The smaller T_{merge} is chosen, the more views are merged and consequently the smaller the final hierarchical model will be. The remaining aspects are stored in the lowest level of the hierarchical model. The next level of the aspect graph is created by continuing the merging while relaxing the similarity constraint. The threshold remains unchanged. Due to the reducing image resolution, smaller dissimilarities are eliminated and therefore the similarity constraint automatically relaxes.

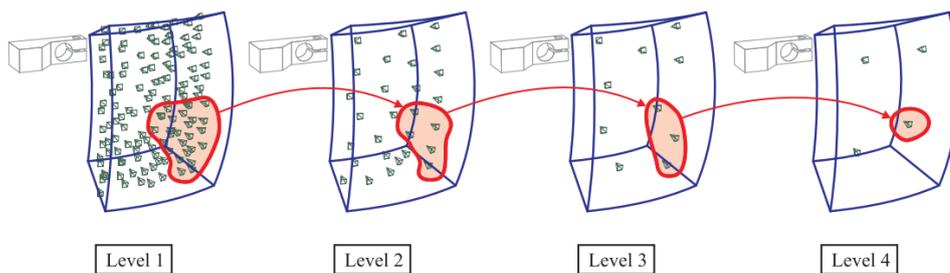


Figure 42: Shape-Based Matching – building a hierarchical model of different views around a virtual object defined by a 3D CAD model [64]

For each single aspect, the child views, that the aspect represents, are stored in a tree structure. The child views are those views on the next lower pyramid level that have been merged to obtain the aspect (Figure 44).

In the next step, the model image generation can be started. For each view on each pyramid level a model image is created by projecting the object into the image plane by using the pose of the current view. Then, the image pyramid of this model image is derived and edges are extracted on each pyramid level. In this way, the scale-space effects in the pyramid can be approximated. In the following step, the CAD model is projected into a 3-channel image, whereby each channel of each planar model face represents a component of its 3D normal vector. The angle in 3D space between the normal vectors of two neighboring faces can therefore be measured in the edge amplitude of the 3-channel image (Figure 43).

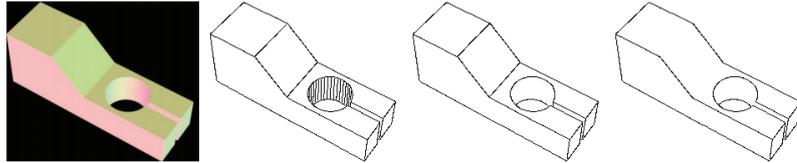


Figure 43: Shape-Based Matching: 3-channel model image as RGB color image. The three visualizations on the right show the extracted edges with increasing threshold T_{amp} [64]

Introducing a threshold T_{amp} for the amplitude, one can suppress edges that are in fact invisible, like the edges of a curved surface. The silhouette of the projected object should not be suppressed. The 2D model is now created from the model image on the associated image pyramid level. Each 2D model and its corresponding pose are stored in the tree structure visualized in Figure 44. The 2D models of neighboring pyramid levels are connected.

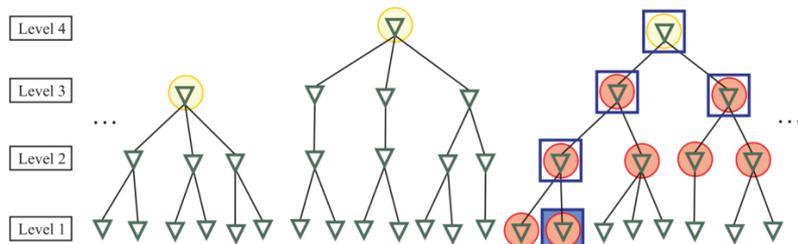


Figure 44: Shape-Based Matching – visualization of the tree structure of the hierarchical model and the process of the hierarchical search [64]

The creation of the hierarchical model is finished (Figure 44). It can now be used to detect the 3D object in a single camera image and to determine the 3D pose of the object. The search starts at the highest pyramid level by computing the similarity measure between the 2D models of the views and the current image pyramid level. The 2D model of the first parent node of each view has to be rotated and scaled to cover the full 360° and the scale range of the merged views to obtain the current view (yellow circular areas in Figure 44). Those matches with a similarity measure greater than T_{score} are stored in a list of match candidates (blue squares in Figure 44). In the next lower pyramid level the found matches are refined by computing the similarity measure between the child and the current image pyramid level (red circles in Figure 44). The scale, translation and rotation range can be limited to a close neighborhood of the parent match. The search is completed as soon as all candidates have been tracked down to the lowest pyramid level (blue filled square in Figure 44). The determined 3D pose is computed based on the 2D matching pose and the 3D pose of the corresponding view. The accuracy of this pose is limited to the sampling of the views. Hence, a refinement of the 3D pose has to be performed. The 3D object is projected into the search image using the initial start pose. Then, the 3D pose is refined by using the Levenberg-Marquardt algorithm that offers a robust iterative nonlinear optimization. The squared distances of the projected CAD model edges and the corresponding image edge points are minimized over 6 pose parameters. Finally, an 6×1 object pose

${}^c\mathbf{P}_o$ with respect to the camera coordinate system is extracted. The corresponding transformation matrix from the camera coordinate system to the object is denoted as cT_o .

$${}^c\mathbf{P}_o = [X, Y, Z, \alpha, \beta, \gamma]^T$$

$${}^cT_o = \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & X \\ r_{21} & r_{22} & r_{23} & Y \\ r_{31} & r_{32} & r_{33} & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (54)$$

α , β and γ describe the rotation of the object and are denoted as Euler angles in the Yaw-Pitch-Roll convention. Details to α , β , γ and the calculation of the 3×3 rotation matrix R are given in section 3.1.2. The algorithm supports full perspective. Therefore, when placing the object in a corner of the image, a homography is applied to the image model before the matching to compensate the perspective distortion. Details to the calculation of the homography can be found in [64].

Another important issue for a correct behavior of the method is the limitation of the pose range to avoid the creation of degenerated views, such as a side view of a planar object. This view would be found for each pair of parallel edges in the image.

According to the results of [64], the pose of the object can be observed with a mean error of about 0.1-0.3% of the object distance in position and 0.3-0.5° in rotation. The result strongly depends on the correspondence between the CAD model and the real object. Small deviations between these two results in strong degeneracies in the final object pose (Figure 45).

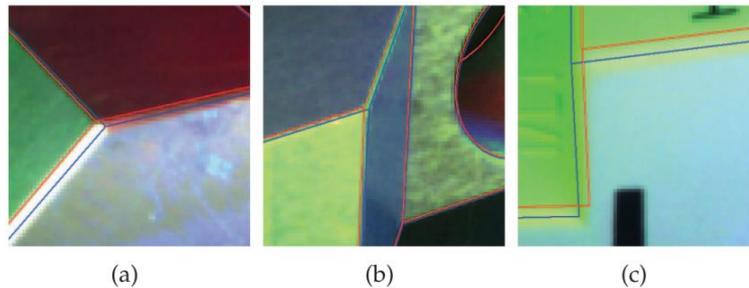


Figure 45: Shape-Based Matching – inaccurate object detection due to round edges, inaccurate CAD model and bad illumination [64]

For instance, if the object has no “sharp” edges or tolerances are not considered, the CAD model has to be adapted or the search has to be stopped at a higher pyramid level where the differences between the model and the image are small enough. Furthermore, inaccuracies of the camera parameters lead to wrong projections of the model to the image plane.

4.3.2. SBM Refinement using a Stereo System (SBM-SOT)

The SBM-SOT (Shape Based 3D Matching – Shape Origin Triangulation) uses a stereo system in combination with the SBM method [64] to ensure a more robust and accurate estimation of the object pose. The method of SBM as explained in 4.3.1 is applied to both camera images of the stereo system to get the 3D object origin pose ${}^{c_1}\mathbf{A}_o$ for camera 1 and ${}^{c_2}\mathbf{B}_o$ for camera 2. The projection of ${}^{c_1}\mathbf{A}_o$ and ${}^{c_2}\mathbf{B}_o$ to the corresponding image plane by using the projection matrix P_c for $c \in \{1,2\}$ delivers the 2D object center coordinates $\mathbf{a}_1 = (u_1, v_1)$ and $\mathbf{b}_2 = (u'_2, v'_2)$ (Equation (55)).

$$\mathbf{a}_1 = P_1 {}^{c_1}\mathbf{A}_o \quad \mathbf{b}_2 = P_2 {}^{c_2}\mathbf{B}_o \quad (55)$$

The 2D points \mathbf{a} and \mathbf{b} can now be used to calculate a 3D position ${}^c\mathbf{P}_o = [X, Y, Z]^T$ by performing a triangulation with the binocular stereo system described in section 3.2.5. Each stereo triangulation ${}^c\mathbf{P}_o$ is referenced to the first camera coordinate system (${}^c\mathbf{P}_o = {}^{c_1}\mathbf{P}_o$).

Figure 46 visualizes the acquisitions of a stereo system with the projected 3D center pose ${}^{c_1}\mathbf{A}_o$ and ${}^{c_2}\mathbf{B}_o$. Depending on the setup of the binocular system (baseline, focal length) and its calibration quality, the 3D object position should exceed the accuracy of the monocular detection.

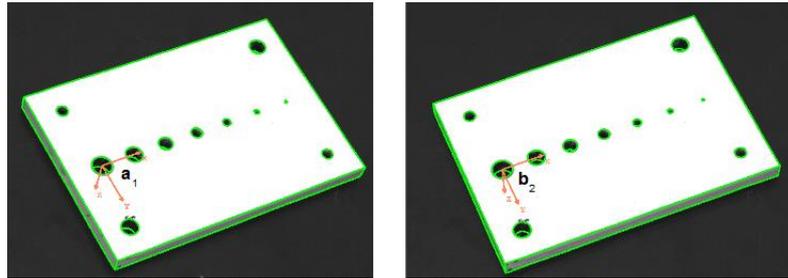


Figure 46: Shape-Based Matching - visualized coordinate system of the object shape center in the left and right image of a stereo system

In the next step, I add the object orientation to the 3D position (${}^c\mathbf{P}_o = [X, Y, Z, \alpha, \beta, \gamma]^T$) by using ${}^{c_1}\mathbf{A}_o$ and ${}^{c_2}\mathbf{B}_o$. I transform ${}^{c_2}\mathbf{B}_o$ to the coordinate system of c_1 by using the known transformation between the two camera centers ${}^{c_1}T_{c_2}$ (calculation shown in section 3.2.6). The obtained pose ${}^{c_1}\mathbf{B}_o$ and ${}^{c_1}\mathbf{A}_o$ are then combined to build the mean value of α , β and γ , which represent the orientation of ${}^c\mathbf{P}_o$.

$${}^c\mathbf{P}_{oi} = \frac{{}^{c_1}\mathbf{A}_{oi} + {}^{c_2}\mathbf{B}_{oi}}{2} \quad \text{for } i \in \{\alpha, \beta, \gamma\} \quad (56)$$

α , β and γ describe Euler angles in the Yaw-Pitch-Roll convention. The major limitation of this approach is that the center point of the object derives from its actual pose in case of an inaccurate SBM fitting. Small changes of the 2D center position result in not negligible deviations of the stereo triangulation along the Z-axis.

4.3.3. SBM Refinement using a Stereo System in Combination with Ellipse Fitting (SBM-ECT)

SBM-ECT (Shape Based 3D Matching – Ellipse Center Triangulation) works similarly to the principle of the SBM-SOT method explained in 4.3.3. The main difference is that I do not use the projected object origin of ${}^{c_1}A_o$ and ${}^{c_2}B_o$ for the triangulation, but a distinctive feature near this center, like a circle or an ellipse. Considering the object of Figure 46, one can extract the edge of the hole and determine the center of the circle/ellipse with subpixel accuracy.

Using the proposed method of [20], the best fitting ellipse can be estimated by using the least-squares criterion for the conic representation of an ellipse (Equation (57)). The coefficients of the ellipse are a, b, c, d, e, f and the coordinates of the points lying on it are x, y . At least 5 points are needed to estimate the parameters [25].

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad (57)$$

Additionally to Equation (57), a specific constraint is introduced to force the conic to be an ellipse. According to [20], this is achieved by optimizing Equation (57) with subject to the equality constraint in Equation (58).

$$4ac - b^2 = 1 \quad (58)$$

After extracting the ellipse center of the image plane in c_1 and c_2 , one can perform the triangulation on the basis of subpixel accurate points to determine ${}^cP_o = [X, Y, Z]^T$. However, the feature does not have to be a circle/ellipse. Any feature can be used, as long as it uniquely defines a subpixel accurate 2D position.

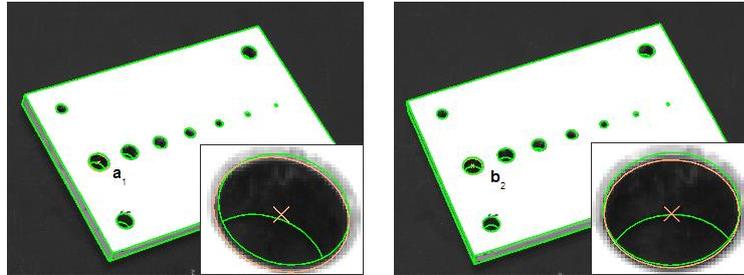


Figure 47: Shape-Based Matching – visualization of ellipse center after estimating the best fitting ellipse using least squares optimization

In this section I add the object orientation to the 3D position (${}^cP_o = [X, Y, Z, \alpha, \beta, \gamma]^T$) by using a weighted combination of ${}^{c_1}A_o$ and ${}^{c_2}B_o$. Therefore, ${}^{c_2}B_o$ is transformed to c_1 (${}^{c_1}B_o$) by using ${}^{c_1}T_{c_2}$ and ${}^{c_1}A_o$ to c_2 (${}^{c_2}A_o$) by using ${}^{c_2}T_{c_1} = {}^{c_1}T_{c_2}^{-1}$. The transformed points are projected to the corresponding image plane in order to get \mathbf{a}_2 and \mathbf{b}_2 . Furthermore, I project cP_o to the image plane of c_1 and c_2 and obtain \mathbf{p}_1 and \mathbf{p}_2 .

The differences $\mathbf{p}_1 - \mathbf{a}_1$ and $\mathbf{p}_2 - \mathbf{a}_2$ are used to measure the quality of the 3D pose estimation of ${}^{c_1}A_o$ and further of its influence on the orientation for cP_o (Equation (59)). Similarly, the weight of ${}^{c_1}B_o$ is calculated by using $\mathbf{p}_1 - \mathbf{b}_1$ and $\mathbf{p}_2 - \mathbf{b}_2$.

$$\begin{aligned}
dist_a &= (\|\mathbf{p}_1 - \mathbf{a}_1\|)^2 + (\|\mathbf{p}_2 - \mathbf{a}_2\|)^2 \\
dist_b &= (\|\mathbf{p}_1 - \mathbf{b}_1\|)^2 + (\|\mathbf{p}_2 - \mathbf{b}_2\|)^2 \\
w_i &= \frac{1}{dist_i} \quad for \ i \in \{a, b\} \\
{}^c\mathbf{P}_o &= \frac{{}^{c_1}\mathbf{A}_{oi}w_a + {}^{c_1}\mathbf{B}_{oi}w_b}{w_a + w_b} \quad for \ i \in \{\alpha, \beta, \gamma\}
\end{aligned} \tag{59}$$

The larger the distance between the positions, the smaller the influence of the corresponding object pose to the orientation. The main limitation of this method is that the distinctive feature has to be visible all the time and is not allowed to be occluded. Furthermore, noise or interfering contours have a greater negative effect on the accuracy of the feature extraction than on the result of the normal SBM.

4.4. 3D Marker Detection

Additionally to the object detection, I need to find a robust and precise 3D pose of the 3D markers to build the industrial marker to object map. As shown in Figure 48, the marker detection is not used only for the creation of the map, but also directly for the extended visual servoing process to define an accurate object pose by combining the 3D marker detection with the industrial marker to object map. The detection of the marker has to be robust with respect to illumination and other environmental changes (shading, focus etc.).

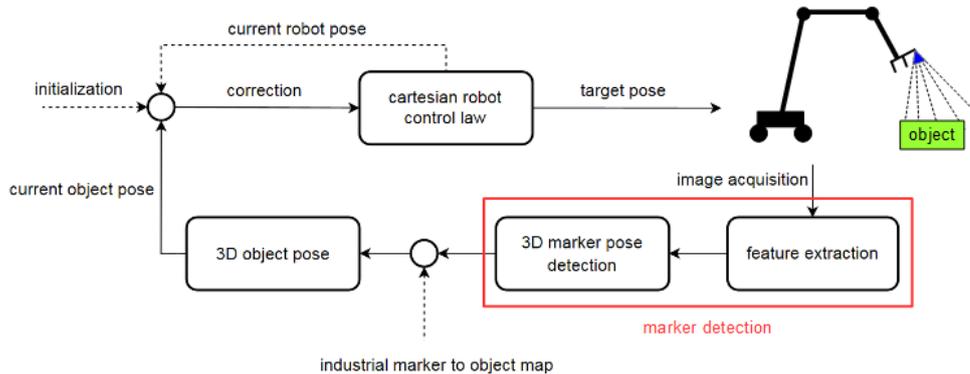


Figure 48: Extended visual servoing concept. Step 2 – 3D marker detection for the use of an industrial marker to object map (adapted from [39])

In the following chapters, the concept and the detection of the 3D ARToolKit marker, the 3D QR Code marker and the 3D Fiducial Propeller marker are explicated. The 3D ARToolKit marker detection will be explained for both monocular and binocular camera systems to get a general overview about the different detection methodologies and their accuracy. The other marker detections will be performed by using a binocular camera system.

4.4.1. 3D ARToolKit Marker – Monocular (3D ARTM-M)

The open source library ARToolKit [1] includes the definition and implementation of a 3D marker. The marker consists of a square black border of known size and a sub-image in the interior of the square which is used for the identification (Figure 49).

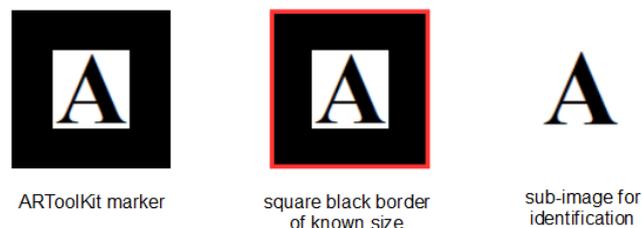


Figure 49: 3D ARToolKit marker – visualization of the square black border of known size and the sub-image for identification (adapted from [1])

The first step of the detection is searching for square shapes in acquired images. Therefore, the image is preprocessed by applying thresholding and contour extraction

of found regions. A line fitting algorithm is used to approximate the contour of each region with 4 lines. The fitted lines are evaluated by the distance error between the line and the data points. If the error is smaller than a predefined threshold, the lines are used to calculate corner points. In the next, step the region that is enclosed by the lines and the corner points is compared to the region of a minimal bounding rectangle of the initial data points. The properties of the bounding rectangle and the found shape should be similar [44].

If these conditions are fulfilled, the four corner points are used to apply a perspective warping to rectify the image pattern. The homography that maps between the quadrilateral and the square/rectangle can be determined by knowing 4 point correspondences between image coordinates (x', y') and rectified marker coordinates (x, y) .

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (60)$$

Details to the calculation of the projective transformation matrix can be found in [26]. The normalized pattern is now used for identification via template matching [36]. The pattern is compared to all trained patterns. A confidence of 1 means equality, whereas deviations lead to smaller values.

If a square marker has been found and identified to be a 3D ARToolKit marker, the calculation of its 3D pose can be performed. This geometrical calculation is based on the 4 vertices and determines both the position of the black square and the orientation of the image pattern with respect to the camera [36].

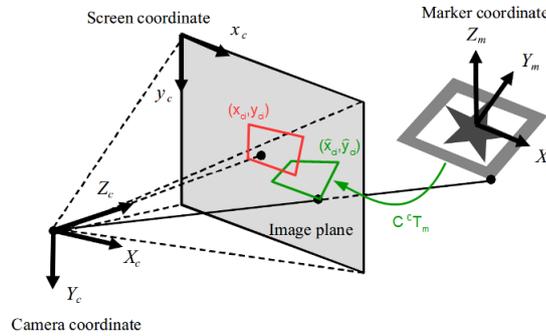


Figure 50: 3D ARToolKit marker – definition of camera and marker coordinate systems and the visualization of the ideal marker points (\hat{x}_c, \hat{y}_c) as well as the undistorted marker points (x_c, y_c) (adapted from [44])

The relation between the camera coordinates $[X_c, Y_c, Z_c, 1]^T$ and the known marker coordinates $[X_m, Y_m, Z_m, 1]^T$ is described in Equation (61) and visualized in Figure 50. Details to the calculation can be found in [36].

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = {}^cT_m \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad (61)$$

Once the marker is transformed into the camera coordinate system, it can be projected to the image plane by using Equation (62). C are the intrinsic camera parameters which are determined by the camera calibration described in 3.2.2. h is a scalar value.

$$\begin{bmatrix} h\hat{x}_c \\ h\hat{y}_c \\ h \end{bmatrix} = \begin{bmatrix} sf_x & 0 & p_x & 0 \\ 0 & sf_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = C \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = C {}^cT_m \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad (62)$$

\hat{x}_c and \hat{y}_c are the projected coordinates for a marker point $[X_m, Y_m, Z_m, 1]^T$ with the relation cT_m to the camera. In the next step, the optimal ${}^c\hat{T}_m$, which minimizes the distance d_{err} between each on cT_m based marker points (\hat{x}_c, \hat{y}_c) and the undistorted observed marker point (x_c, y_c) , is determined (Equation (63)) [36].

$$d_{err} = \frac{1}{4} \sum_{i=1,2,3,4} \left\{ (x_{c_i} - \hat{x}_{c_i})^2 + (y_{c_i} - \hat{y}_{c_i})^2 \right\} \quad (63)$$

$${}^c\hat{T}_m = \arg \min_{{}^cT_m} d_{err}$$

The resulting homogenous 4×4 transformation matrix ${}^c\hat{T}_m$ is then converted to the 6×1 pose ${}^cP_m = [X, Y, Z, \alpha, \beta, \gamma]^T$. The rotation of the matrix ${}^c\hat{T}_m$ is decomposed to $[\alpha, \beta, \gamma]$ in a manner of a Yaw-Pitch-Roll Euler convention, shown in section 3.1.2. Therefore, P_{cm} is defined as depicted in Equation (64).

$${}^cP_m = [{}^c\hat{T}_{m_{14}}, {}^c\hat{T}_{m_{24}}, {}^c\hat{T}_{m_{34}}, \alpha, \beta, \gamma]^T \quad (64)$$

4.4.2. 3D ARToolKit Marker – Binocular (3D ARTM-B)

The detection of the 3D ARToolKit marker is now extended to a binocular stereo system. All processing steps up to the estimation of the 3D pose do not change. Instead of estimating the 3D pose on the basis of the marker size, I use the extracted corner points of the marker. Before applying the triangulation method mentioned in section 3.2.5, I conduct a refinement of the corner points to get subpixel accuracy. According to [7], the most common way for subpixel refinement is based upon on a mathematical observation that concludes that the dot product of a vector and an orthogonal vector is zero. This is the case at corner locations (shown in Figure 51).

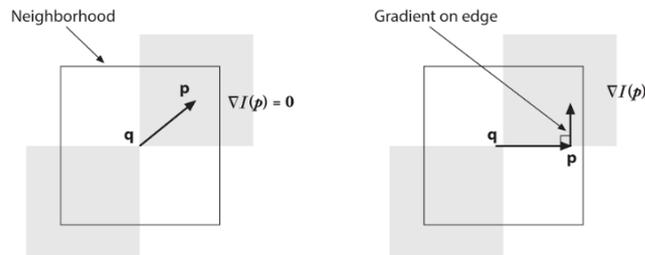


Figure 51: 3D ARToolKit marker – basic concept of subpixel refinement of corner points [7]

In the first step, vectors starting at a point q and ending at p are examined. If p lies on an edge, the vector $q - p$ is orthogonal to the gradient at p . If p is in a flat region, the

gradient is zero. Therefore, the dot product between the vector $\mathbf{q} - \mathbf{p}$ and the gradient at \mathbf{p} is zero, if \mathbf{p} lies on an edge or in a flat region. The corresponding mathematical formulation looks as the following in Equation (65) [7].

$$\langle \nabla I(\mathbf{p}), \mathbf{q} - \mathbf{p} \rangle = 0 \quad (65)$$

By assembling a lot of such pairs, one can build a system of equations and solve it. The solution will yield a subpixel location for \mathbf{q} . Using the refined corner points, the triangulation process delivers $N = 4$ points \mathbf{P} in 3-dimensional space (Figure 57).

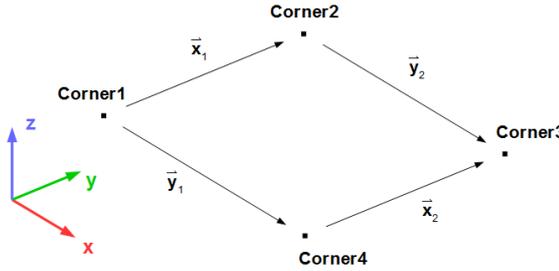


Figure 52: 3D ARTolKit marker – triangulation of all 4 corner points to get 3D position and build the orientation vector for RX, RY and RZ

After the triangulation, a redundant 3D center position \mathbf{t} of the marker can be defined by taking the mean of the N 3D corner positions (Equation (66)).

$$\mathbf{t} = \frac{1}{N} \sum_{i=1}^N \mathbf{P}_i \quad (66)$$

The points \mathbf{P} are further used to determine the unit vectors between the corner points, which represent the axis of the marker coordinate system (Equation (67)).

$$\begin{aligned} \vec{x}_1 &= \frac{\mathbf{Corner2} - \mathbf{Corner1}}{\|\mathbf{Corner2} - \mathbf{Corner1}\|} & \vec{x}_2 &= \frac{\mathbf{Corner3} - \mathbf{Corner4}}{\|\mathbf{Corner3} - \mathbf{Corner4}\|} \\ \vec{y}_1 &= \frac{\mathbf{Corner4} - \mathbf{Corner1}}{\|\mathbf{Corner4} - \mathbf{Corner1}\|} & \vec{y}_2 &= \frac{\mathbf{Corner3} - \mathbf{Corner2}}{\|\mathbf{Corner3} - \mathbf{Corner2}\|} \end{aligned} \quad (67)$$

The mean of \vec{x}_1 and \vec{x}_2 is considered to be the marker's \vec{x} vector and the mean of \vec{y}_1 and \vec{y}_2 is the marker's \vec{y} vector. The cross-product of \vec{x} and \vec{y} delivers the \vec{z} vector (Equation (68))

$$\vec{z} = \frac{\vec{x} \times \vec{y}}{\|\vec{x} \times \vec{y}\|} \quad (68)$$

Using the 3D position and the vectors of the marker, one can build a homogenous 4x4 matrix T which describes the pose of the 3D ARTolKit marker (Equation (69)).

$${}^cT_m = \begin{bmatrix} x_1 & y_1 & z_1 & t_x \\ x_2 & y_2 & z_2 & t_y \\ x_3 & y_3 & z_3 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (69)$$

Similarly to section 4.4.1, cT_m is then converted to the 6x1 pose cP_m .

4.4.3. 3D QR Code Marker (3D QRCM)

In this section I consider the concept on how to detect a 3D QR Code marker by using a binocular stereo system. This marker consists of 5 individual QR Codes that are surrounded by a black square (Figure 53). The QR Codes in the corner positions of the square have the same identification for each 3D QR Code marker. The only code that changes is the QR code in the center position. It is used to differentiate between several 3D QR Codes in a scene. The main task is to find corresponding 3D QR Code markers in both the first and the second image of the stereo system and to calculate the 3D position of each QR Code in the 3D marker.

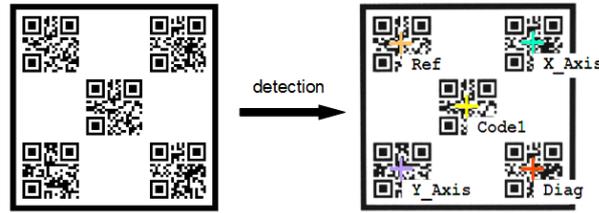


Figure 53: 3D QR Code marker – visualization of the marker concept. The marker has 4 consistent markers (Ref, X_Axis, Y_Axis, Diag) and 1 identification marker (CodeX)

In the first step, I try to find rectangular regions in the acquired image. The characterization of the shape is performed by evaluating the size, the convexity and the rectangularity of a selected region.

The area of the region A_1 is not allowed to be smaller or bigger than predefined values A_{min} and A_{max} [49]. The area of the object can be simply calculated by using Equation (26) of section 3.2.4.

To evaluate the convexity, one compares the area of the region with the area of the convex hull of this region (Equation (70)). C is equal to 1, if the region is convex and does not contain concavities. Otherwise, the value of C is smaller than 1 [49].

$$C = \frac{A_1}{A_{convex}} \quad (70)$$

The rectangularity is evaluated by using the geometric image moments described in 3.2.4. According to [55], considering the three central moments of second order μ_{20} , μ_{02} and μ_{11} , one can build the inertial tensor as well as derive an estimate of a rectangle's measurements \hat{a} and \hat{b} and its orientation $\hat{\theta}$ with its eigenvalues and eigenvectors. Using Equation (32) and (33), [55] calculates the measurements as shown in Equation (71).

$$\hat{a} = a\sqrt{3} \quad \hat{b} = b\sqrt{3} \quad \hat{\theta} = \theta \quad (71)$$

The determined rectangle is then used to clip the region, whereby A_1 is denoted as the complete region, A_2 as the clipped region and A_3 as the rectangle.

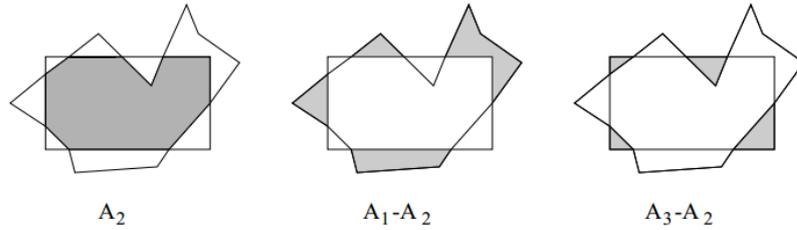


Figure 54: 3D QR Code marker - clipping of the region and the fitted rectangle to get a measure for rectangularity [55]

The discrepancy between the region and the rectangle consists of the region outside the rectangle ($A_1 - A_2$) and the area inside the rectangle that is not filled ($A_3 - A_2$). Combining these two parts, one can build a normalized error measurement for rectangularity R_D , which has its peak at 1 (optimal rectangle).

$$R_D = 1 - \frac{(A_1 - A_2) + (A_3 - A_2)}{A_3} \quad (72)$$

After a square with the predefined properties (area, convexity and rectangularity) has been found, I check if the region consists of the five individual QR codes mentioned previously. To identify and locate a QR code, [24] applies contour detection to extract the three distinct position detection patterns (shown in Figure 55).

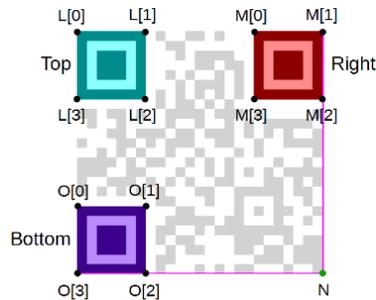


Figure 55: 3D QR Code marker - visualization of the three QR Code localization patterns (adapted from [24])

Having detected three patterns, the orientation of the markers and the positions with respect to each other are calculated by using a triangle that is formed by the centers of the three contours. This information helps to classify the markers into 'Top', 'Right' and 'Bottom' marker. The vertex that does not involve the largest side of the triangle is obviously the 'Top' marker. The other two points are identified by considering the slope of the line they form and the position of the marker 'Top' with respect to this line.

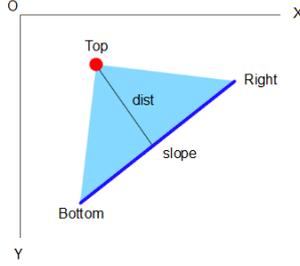


Figure 56: 3D QR Code marker - triangle formed by detection patterns to classify into 'Top', 'Right' and 'Bottom' pattern (adapted from [24])

As soon as the corner points of all identification markers are known, one can determine the fourth corner point of the QR Code by calculating the intersection N with two lines formed by $M[1]$ and $M[2]$ and by $O[3]$ and $O[2]$. Then, one can apply a perspective warping to restore the QR Code to a readable position [24]. The warping transforms a general quadrilateral defined by the 4 corner points to a square. Details to the calculation of the projective transformation matrix can be found in [26]. After the transformation of the marker to a square, the data can be extracted from the QR Code by using a capable library such as OpenCV or ZBar.

If the second camera image of the binocular system also contains a 3D QR Code marker with the same identification code in the center, the triangulation of 2D point correspondences, as mentioned in section 3.2.5, is applied to each QR Code position. Therefore, a triangulation process of one 3D QR Code marker delivers $N = 5$ points P in 3-dimensional space (shown in image Figure 57).

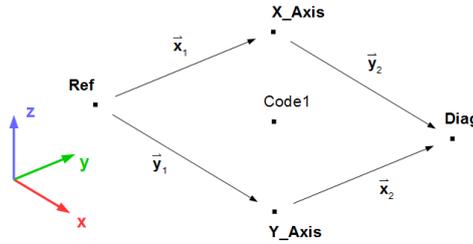


Figure 57: 3D QR Code marker – triangulation of all 5 QR code centers to get 3D position and build the orientation vector for RX, RY and RZ

After the triangulation, a redundant 3D center position t of the marker can be defined by taking the mean of the N 3D QR code positions (Equation (73)).

$$t = \frac{1}{N} \sum_{i=1}^N P_i \quad (73)$$

The homogenous 4×4 matrix cT_m and the 6×1 pose cP_m of the marker are determined as described in section 4.4.2. The main difference between this section and section 4.4.2 is that QR code centers **Ref**, **X_Axis**, **Y_Axis** and **Diag** are used to calculate \bar{x} , \bar{y} and \bar{z} , instead of the corner points of the markers.

4.4.4. 3D Fiducial Propeller Marker (3D FPM)

The next marker type that can be used to determine a 3D pose with a binocular stereo system is the 3D Fiducial Propeller marker. The concept of this marker is similar to the 3D QR Code marker explained in 4.4.3. The 3D Fiducial Propeller marker consists of 5 individual sub-markers which are surrounded by a black square (Figure 58). The 2D Fiducial Propeller sub-markers were designed as described in [56].

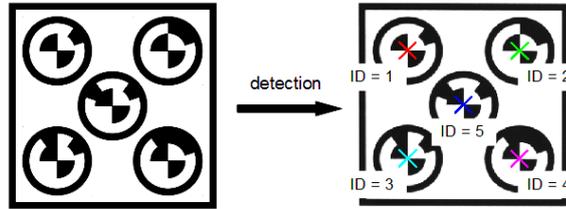


Figure 58: 3D Fiducial Propeller marker – visualization of the marker concept. The marker has 4 consistent markers (ID = 1,2,3,4) and 1 identification marker (ID ≥ 5)

Four of the sub-markers are consistent for all 3D markers (ID=1, 2, 3, 4). Only the identification marker in the center of the 3D marker changes (ID ≥ 5) to define each 3D marker uniquely. The detection of the 3D Fiducial Propeller marker also starts with the extraction of the black square and uses the same method of shape characterization as the 3D QR Code marker (evaluation of size, convexity and rectangularity).

After the extraction of rectangular shapes, I check if each of the rectangles contains the 5 Fiducial Propeller markers (ID = 1, 2, 3, 4 and identification). Hence, I firstly try to find 5 circles in the rectangle by evaluating the area size and the circularity of the inner regions.

The area A is determined by using the zero order moment, as described in section 3.2.4, Equation (26), and by comparing it to a predefined A_{min} and A_{max} [49].

The circularity is evaluated by using the area A of the region and the maximum distance r_{max} between the center of mass and all contour points of the region. The center of mass is calculated by applying Equation (27) of section 3.2.4. The formulation of the circularity C is shown in Equation (74) [49].

$$C = \min\left(1, \frac{A}{r_{max}^2 * \pi}\right) \quad (74)$$

In case of a perfect circle, C is equal to 1. Otherwise the value decreases.

In the next step, I check if each circle contains exactly one saddle point. To get a sub-pixel accurate detection, the input image is approximated by a quadratic polynomial in x and y . The polynomial is created by convolving the input image with derivatives of Gaussians up to second order. Details to this approach can be found in [60]. The saddle point detection is performed by building the Hesse matrix H and evaluating its eigenvalues. H is the matrix of second derivatives. For instance, I_{xy} is the partial derivate of an approximated polynomial $I(x, y)$ in x and then in y direction [13].

$$H = \begin{bmatrix} I_{xx} = \frac{\partial^2 I(x,y)}{\partial x^2} & I_{xy} = \frac{\partial^2 I(x,y)}{\partial x \partial y} \\ I_{yx} = \frac{\partial^2 I(x,y)}{\partial y \partial x} & I_{yy} = \frac{\partial^2 I(x,y)}{\partial y^2} \end{bmatrix} \quad (75)$$

If the eigenvalues are both positive and negative, the stationary point is a saddle point. This refers to the fact that a saddle point has minimal image intensity along one direction and maximal image intensity along another direction. If only one saddle point exists in the selected region, the marker is further processed by performing identification.

The identification of the marker is conducted by evaluating the binary code between the propeller pattern and the outer border. To read the binary code, I applied the procedure described in [56]. Therefore, I first warp the circular pattern to a rectangle by sampling the points of the marker around its center. Due to perspective distortions, I have to sample points on an ellipse. In general, points on an ellipse are defined by Equation (76).

$$x(t) = a \cos(t) \quad y(t) = b \sin(t) \quad (76)$$

The major and minor axis of the ellipse are a and b . The current angle is described by t . In case that the ellipse is rotated around an angle φ , each point on the ellipse can be defined by Equation (77). An ellipse of this kind is visualized in Figure 59.

$$\begin{aligned} x(t) &= a \cos(t) \cos(\varphi) - b \sin(t) \sin(\varphi) \\ y(t) &= b \sin(t) \cos(\varphi) + a \cos(t) \sin(\varphi) \end{aligned} \quad (77)$$

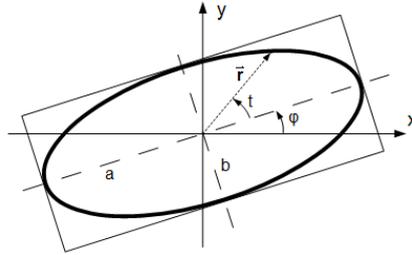


Figure 59: 3D Fiducial Propeller marker - sampling of points on an ellipse

Introducing a scale factor $\{s \in \mathbb{R} | 0 \leq s \leq 1\}$, which is applied on $x(t)$ and $y(t)$, enables a sampling of every point that lies on the vector \vec{r} . By varying s and t one can sample every point in- and outside the ellipse. Thereby, it is possible to create a mapping function along the X-axis as well as along the Y-axis. The mapping of every pixel location can be expressed through Equation (78) [56].

$$g(s, t) = f(h(s, t)) \quad (78)$$

$g(s, t)$ represents the remapped image of the source image $f(\cdot)$, according to a mapping function $h(s, t)$ that operates on (s, t) . The mapping operation of a circular marker to a rectangular image based on (s, t) is visualized in Figure 60.



Figure 60: 3D Fiducial Propeller marker – the variation of s and t can be used to create a mapping for the x and y direction. The remapping of the marker results in a rectangular image that can be used to extract the marker identification (adapted from [56]).

The rectangular image of the marker can be used to extract a binary code. The code is evaluated according to the detection probabilities. The probability is calculated by comparing the area of the one filled/empty bin of the code with the ideal area of one code bin. Figure 60 displays the probabilities for center, binary code and outer circle. The ID of the marker is obtained by checking the code in a pre-defined lookup table. The marker is rejected if the code probability is too low or if the extracted code pattern is not found in the lookup table [56].

After extracting a subpixel accurate 2D location and the ID of the 3D Fiducial Propeller marker, I search for a corresponding marker in the second camera image of the binocular stereo system. If a match is found, the calculation of the homogenous 4×4 matrix cT_m and the 6×1 pose cP_m of the marker is performed, as explained in section 4.4.3.

4.5. Robot Control

The calculated 3D object pose cP_o is used to control the end-effector of a 6-axis robot. The main goal is to enable an interaction between the tool center point (TCP) of the robot and the object. This process is known as visual servoing and is shown in Figure 61. Every Cartesian robot control is able to deliver the current 3D robot pose and to receive 3D target poses for the movement. Therefore, the robot control can be considered as a black box that performs all necessary operations to ensure a correct movement. The calculations of the forward and inverse kinematics mentioned in section 3.3 are automatically performed by the robot controller. No information about the robot's geometric properties and its axis configuration have to be known.

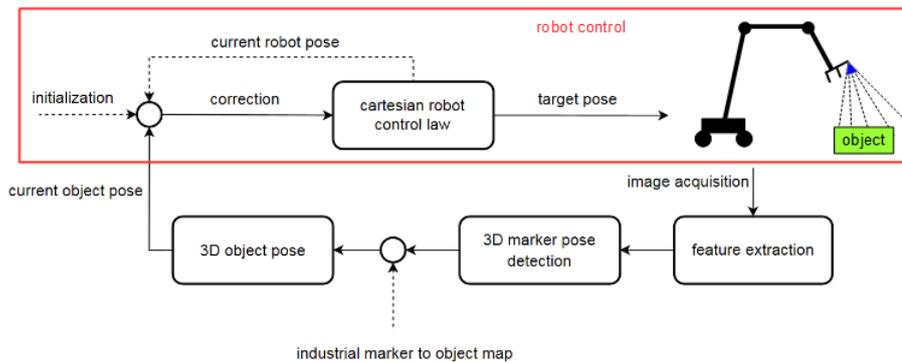


Figure 61: Extended visual servoing concept. Step 2 – robot control with 3D target positions of the camera system (adapted from [39])

The binocular camera system that feeds the robot control with information is mounted on the end-effector of the robot. This system is called Hand-Eye system because the robotic hand is controlled by mechanical eyes. The relationship of the TCP and the camera system can be determined by performing a Hand-Eye calibration (section 4.5.1). If the camera is stationary mounted in the environment, one has to robustly track the robot end-effector and the object to enable visual servoing. Furthermore, each time the mobile platform moves, the relation between the robot base coordinate system and the camera coordinate system has to be determined anew. Otherwise, controlling on basis of the camera coordinates is not possible.

The main disadvantage of an end-effector mounted camera system is that the camera's field of view changes with the robot movement. Consequently, there is no guarantee that the features I want to inspect are in focus or even visible.

4.5.1. Hand-Eye Calibration

The Hand-Eye calibration is performed with a stationary calibration plate that is observed from different robot positions. The basic idea of this calibration method is to use the extracted information of the calibration plate (3D pose relative to the camera) and the current 3D robot pose to estimate two unknown transformations [28]. The first unknown transformation is between the camera coordinate system (CCS) and the robot

flange coordinate system (FCS), the second one between the robot coordinate system (RCS) and the calibration plate/object coordinate system (OCS). Figure 62 visualizes the concept of the Hand-Eye calibration for two different robot end-effector poses.

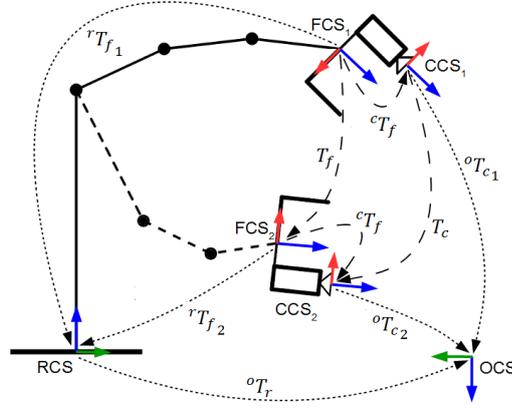


Figure 62: Hand-Eye and Robot-World calibration – declaration of transformation matrices (adapted from [28])

The two transformations from CCS to OCS, illustrated in Figure 62, are denoted as ${}^oT_{c1}$ and ${}^oT_{c2}$. The calculation of the relative movement between the camera positions is shown in Equation (79).

$$T_c = \begin{pmatrix} R_{T_c} & \mathbf{t}_{T_c} \\ \mathbf{0}^T & 1 \end{pmatrix} = {}^oT_{c2}^{-1} {}^oT_{c1} \quad (79)$$

$R_{T_c} \in SO(3)$ represents a 3×3 rotation matrix and $\mathbf{t}_{T_c} \in \mathbb{R}^3$ a translation vector. The movement of the end-effector can be described as

$$T_f = \begin{pmatrix} R_{T_f} & \mathbf{t}_{T_f} \\ \mathbf{0}^T & 1 \end{pmatrix} = {}^rT_{f2}^{-1} {}^rT_{f1} \quad (80)$$

${}^rT_{f1}, {}^rT_{f2} \in \mathbb{R}^{4 \times 4}$ are the transformations from the FCS to the RCS. Using the determined transformations T_c and T_f , the first unknown transformation ${}^cT_f \in \mathbb{R}^{4 \times 4}$ between the FCS and the CCS can be described in the following kinematic loop:

$$T_c {}^cT_f = {}^cT_f T_f \quad (81)$$

Equation (81) can be uniquely solved by at least two relative motions with non-parallel axes [61]. The accuracy of the result increases by executing several motions to get a set of matrices $T_{c_i}, T_{f_i}, i = 1 \dots n$. The overdetermined system is solved by minimizing Equation (82).

$$\min_{T_{fc}} \sum_{i=1}^n \|T_{c_i} {}^cT_f - {}^cT_f T_{f_i}\|^2 \quad (82)$$

The transformation between the RCS and the OCS is denoted as ${}^oT_r \in \mathbb{R}^{4 \times 4}$. According to [68], by using this transformation, one can build another closed kinematic loop (Equation (83)).

$${}^oT_{c_i} {}^cT_f = {}^oT_r {}^rT_{f_i}, \quad i = 1 \dots m \quad (83)$$

Instead of the relative movements T_c and T_f , I use absolute camera and robot poses ${}^oT_{c_i}$, ${}^rT_{f_i}$. cT_f , as determined in Equation (82), is now used for the calculation of oT_r in the means of simple matrix multiplication.

$${}^oT_r = {}^oT_{c_i} {}^cT_f {}^rT_{f_i}^{-1}, \quad i = 1 \dots m \quad (84)$$

[28] and [68] display a method for refining the result of cT_f and oT_r . This method uses cT_f , oT_r as initialization and minimizes Equation (83).

$$\min_{{}^cT_f, {}^oT_r} \sum_{i=1}^n \left\| {}^oT_{c_i} {}^cT_f - {}^oT_r {}^rT_{f_i} \right\|^2 \quad (85)$$

It is also possible to start with Equation (83) and solve both transformations cT_f and oT_r simultaneously [61] [68]. In this case, Equation (83) is decomposed to a rotational matrix equation and a translational vector equation.

$$\begin{aligned} R_{{}^oT_{c_i}} R_{{}^cT_f} &= R_{{}^oT_r} R_{{}^rT_{f_i}} \\ R_{{}^oT_{c_i}} \mathbf{t}_{{}^cT_f} + \mathbf{t}_{{}^oT_{c_i}} &= R_{{}^oT_r} \mathbf{t}_{{}^rT_{f_i}} + \mathbf{t}_{{}^oT_r} \end{aligned} \quad (86)$$

Detailed mathematical background and explanation to this method are presented in [16] and [61]. If only cT_f has to be determined, Equation (82) is sufficient for the Hand-Eye calibration. The minimization over the extended closed loop (Equation (85)) is merely used if oT_r is required or if the results of Equation (82) are not accurate enough. The quality of the Hand-Eye calibration strongly depends on the accuracy of the known transformations between FCS and RCS and between CCS and OCS. To check the result, one can calculate the pose error of the complete chain transformation, which should result in the identity matrix (Equation (87)).

$$I = {}^oT_r^{-1} {}^oT_{c_i} {}^cT_f {}^rT_{f_i}^{-1} \quad (87)$$

4.5.2. Position-Based Cartesian Robot Control

In this section, I assume that I have successfully performed a Hand-Eye calibration to determine cT_f . Furthermore, I have a robust estimate of cT_o using the object detection via the industrial marker to object map discussed in 4.2. The task is to move the TCS of the robot to the observed OCS by utilizing a position-based Cartesian robot control. The transformations and coordinate systems are visualized in Figure 63.

The position-based Cartesian robot control handles transformations referring to the FCS. It is used to retrieve the current robot pose rT_f or to drive the robot to a new target pose ${}^rT_{f_{new}}$. The calculation of rT_f with respect to the current robot joint angles and the calculation of the joint angles for a given target pose ${}^rT_{f_{new}}$ are explained in section 3.3.

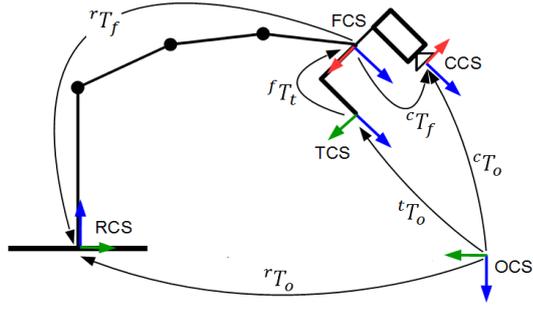


Figure 63: Notation of the coordinate systems and transformations between the coordinate systems (adapted from [28])

The Hand-Eye calibration is defined with respect to the FCS. Therefore, it is possible to use different robot end-effectors (changing TCS) without redefining the Hand-Eye relation cT_f . Using cT_o , cT_f and rT_f , one can determine the relation between the OCS and the RCS (rT_o).

$${}^rT_o = {}^rT_f {}^cT_f^{-1} {}^cT_o \quad (88)$$

The main goal is to move the robot into a position, where the TCS coincides with the OCS. To achieve that, tT_o has to be equal to the identity matrix. Thus, the new target pose of the robot (with respect to the robot flange) is represented by Equation (89).

$${}^rT_{f_{new}} = {}^rT_o {}^tT_o^{-1} {}^fT_t^{-1} = {}^rT_o I {}^fT_t^{-1} = {}^rT_o {}^fT_t^{-1} \quad (89)$$

The process of the movement is visualized in Figure 64. Firstly, the robot is controlled to an initial pose. Then the pose is evaluated to determine whether the target is reached. If the offset to the target is larger than a predefined threshold (POT – pose offset threshold), a flange correction is applied.

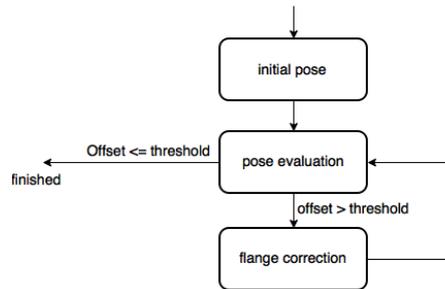


Figure 64: Iterative robot control towards the desired target pose

When performing pick-and-place operations, the robot usually moves to an offset pose, before it moves to the object. The offset is either denoted as ${}^oT_{o_2}$ (offset with respect to OCS) or ${}^tT_{t_2}$ (offset with respect to TCS). Both ${}^oT_{o_2}$ and ${}^tT_{t_2}$ are visualized in Figure 65.

Figure 65 (left) visualizes the definition of the initial pose, which approximately places the robot to the desired offset location. This movement is calculated with respect to the RCS. As a consequence, the new target pose (*initial pose*) is defined as follows:

$${}^rT_{f_{new}} = {}^rT_o {}^oT_{o_2} {}^tT_{o_2}^{-1} {}^fT_t^{-1} = {}^rT_o {}^oT_{o_2} I {}^fT_t^{-1} = {}^rT_o {}^oT_{o_2} {}^fT_t^{-1} \quad (90)$$

After the movement, an evaluation is performed to check if the robot has reached its end position. For that, I use the concept illustrated in Figure 65 (right), which excludes the kinematic chain of the robot to remove the influence of possible errors in the robot's geometric relations. According to 4.3.1, the object detection can vary up to 0.3-0.5°. As a consequence, applying the offset transformation with respect to the OCS would result in severe deviations in the final robot offset pose. The robot target pose would change with each measurement.

Thus, in this approach the offset ${}^tT_{t_2}$ is added with respect to the TCS. The calculation of the remaining movement t_2T_o (pose evaluation) is shown in Equation (91).

$${}^t_2T_o = {}^tT_{t_2}^{-1} {}^fT_t^{-1} {}^cT_f^{-1} {}^cT_o \quad (91)$$

If the target pose is reached perfectly, t_2T_o is equal to the identity matrix I . If t_2T_o exceeds a predefined threshold (if it deviates from I), a correction is applied to the flange position. t_2T_o is mapped to the FCS with $T({}^t_2T_o)_f$ and added to the current robot pose rT_f (flange correction).

$${}^rT_{f_{new}} = {}^rT_f T({}^t_2T_o)_f \quad (92)$$

The evaluation of the pose and the correction are performed until t_2T_o is almost equal to the identity matrix and TCS_2 almost coincides with the OCS.

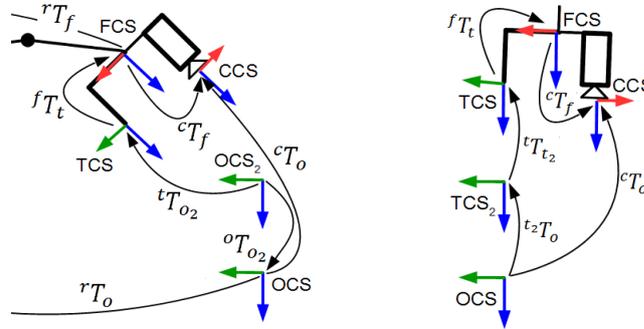


Figure 65: Position-based Cartesian robot control: initial pose definition with kinematic chain (left) and refinement concept without the kinematic chain (right) (adapted from [28])

Deviations between the initial and the target pose can occur due to inaccuracies in the object detection or the Hand-Eye calibration. In case of an imprecise Hand-Eye calibration, the correction movements do not coincide with the movements of the FCS. Then, several correction steps have to be applied until the offset lies within the threshold.

4.5.3. Robot Extrapolation

I assume that the robot has a small kinematic error and that therefore its actual movement derives from the desired one. According to [23], this error is influenced by the following factors:

geometric errors	non-geometric errors
Zero position error (80-90%)	Elasticity of joints (3-8%)
Length and angular error (5-10%)	Drive errors (1-2%)
Temperature (0-10%)	Stochastic error (1-2%)

Table 3: Geometric and non-geometric errors of the robot's kinematic chain [23]

As long as the markers for object detection are visible, the errors can be compensated by visual servoing. At a certain object offset height, all markers are out of the field of view and no detections can be performed. The errors for the final movement from this point cannot be compensated by visual servoing.

Thus, I introduce the third step of our extended visual servoing concept, which now consists of a target pose extrapolation based on previously gathered data (Figure 66).

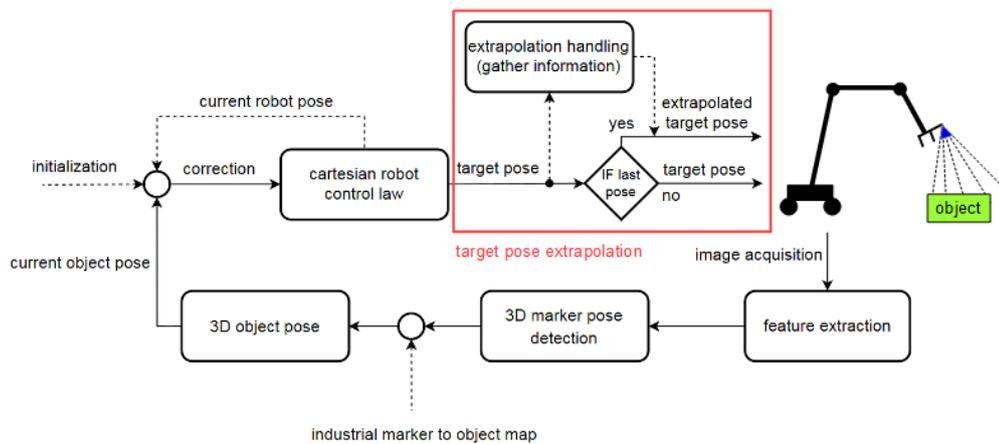


Figure 66: Extended visual servoing concept. Step 3 – robot control via target pose extrapolation (adapted from [39])

The extrapolation information is gathered by moving the robot to several object offset positions (support points). The robot is controlled to each support point, using the method described in section 4.5.2. At first, a coarse location that approximately coincides with the actual target is approached, then an iterative refinement is applied to get to the actual target pose.

To determine the behavior of the TCS, one can either extrapolate the target pose by using the poses of the FCS with respect to the RCS or the OCS with respect to the RCS. The accuracy of the FCS poses are limited to the incremental steps of the robot and should therefore not be used. If the robot is capable of moving only with 0.3 mm increments, the FCS can derive up to 0.3 mm of the desired pose. All poses of the FCS are afflicted with this tolerance. On the other hand, the OCS with respect to the RCS,

takes the actual FCS and the actual OCS into account and hence delivers more accurate extrapolation points. The procedure of extrapolation according to the OCS is visualized in Figure 67.

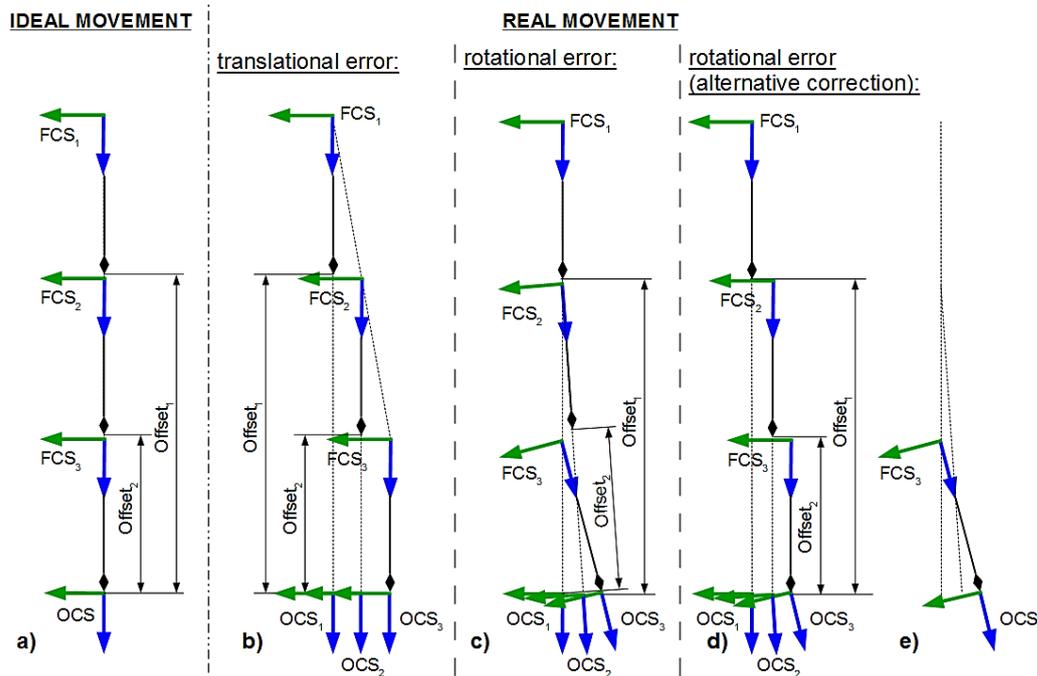


Figure 67: Target pose extrapolation – extrapolation procedure based on the target coordinates of the OCS with respect to the RCS

Figure 67 a) shows the ideal movement that is expected from a precisely calibrated robot and an optimal detection of the offset pose. When moving towards the object by incrementally reducing the object offset, the pose of the OCS should not change at all. As mentioned before, in reality there are some deviations in the movement of the robot that are not accounted by the robot controller.

Figure 67 b) shows the correction of a purely translational error. Due to the error of the robot's kinematic model, it seems that the object moves, although its pose stays the same the whole time. Based on the observed movement, a function is fitted to predict the final pose OCS_3 by extrapolation.

This approach can also be applied in case of a purely rotational error, which is shown in Figure 67 c). The pose is corrected by predicting the object's translational and rotational movement. The orientation of the object can vary between $0.3-0.5^\circ$ as pointed out by the results of the object detection in section 4.3.1. As a consequence, I do not correct the orientation at each step, but improve only the position of the FCS (Figure 67 d)). A correction of an inaccurate object orientation would result in high deviations of the FCS position at high offset values. Thus, the orientation information of each support point is gathered, but not changed until the last step, where a robust rotation correction is applied with respect to the TCS (Figure 67 e)).

As soon as the last support point is reached, the collected pose data is fitted by a linear function. I assume that if I perform a linear movement with the robot, the error will also

behave linear. The line through the data is estimated by using linear regression (Figure 68).

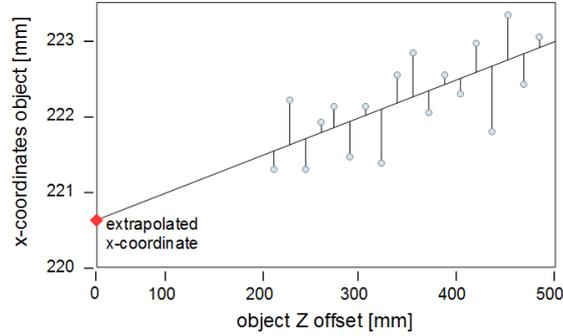


Figure 68: Linear regression of gathered data

Denoting the slope of the line with k and the increment with d , y in relation to x can be drafted in the form of Equation (93).

$$y = kx + d \quad (93)$$

According to [45], when estimating a line, not all points will fall exactly on this line. As a result, there will be an error ε that accounts the failure of the model to fit the data. Equation (94) is also known as regression model.

$$\varepsilon_i = y_i - kx_i - d \quad (94)$$

Having multiple points, I choose the parameters k and d that minimize the error ε for all points. The least squares formulation of this problem is shown in Equation (95).

$$S(k, d) = \sum_{i=1}^n (y_i - kx_i - d)^2 \quad (95)$$

Solving Equation (95) results in parameters for k and d that describe the best fitting lines with the means of least squares distances [45]. The line that fits the data best is visualized in Figure 68. Extrapolating this line to an object offset value of zero leads to an estimate for the target pose. This extrapolation has to be performed for the target $X, Y, Z, \alpha, \beta, \gamma$ values.

The compensation holds true, if the kinematic deviations cause a linear error during linear movements. Otherwise the prediction will fail. The complexity of the regression model can be increased to achieve a better fitting result in case the error is not linear. As a consequence, deviations that correspond to more general functions can be estimated, too. Depending on the data input, the complexity has to be chosen at a degree where overfitting and oscillations are avoided.

5. Experiments/Results

In this section, the extended visual servoing method and the industrial marker to object map mentioned in section 4 are tested in a real application (Figure 69). The task is to realize a pick-and-place handling of a known object with a robot that is able to interact with humans. The region that has to be observed by the camera system is limited to a 450 mm x 450 mm plane. The robot/object interactions should be executed with an absolute accuracy smaller than ± 1 mm. Consequently, I perform a bottom-up evaluation of the accuracies of each system that was used to realize the extended visual servoing. I begin with the accuracy of a single camera and end with the accuracy evaluation of the final robot TCP pose with respect to the object target pose.

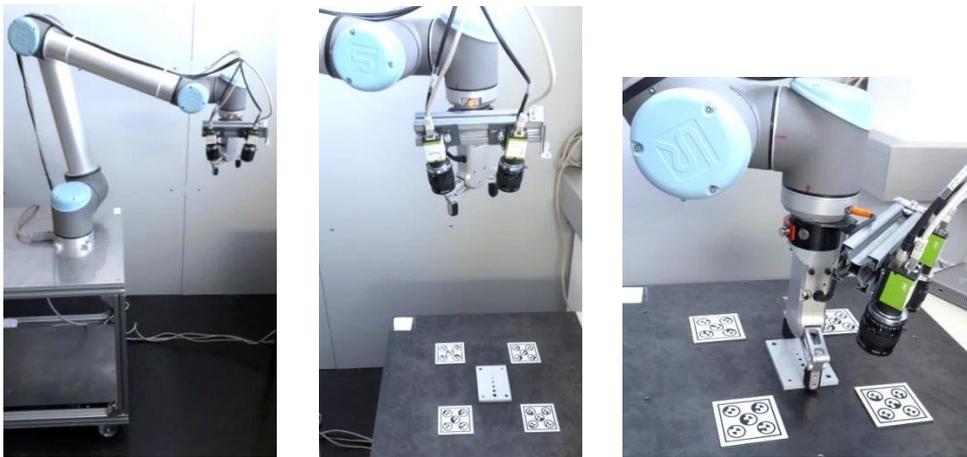


Figure 69: Visualization of the conducted experiment – the mobile platform is moved to a work station. The industrial manipulator grasps the object precisely by using the industrial marker to object map and the extended visual servoing method

In the first section, I describe the equipment that was used to perform the experiment. I proceed with the accuracy of the monocular and binocular camera calibration. Afterwards, I concentrate on the accuracy and limitations of each object and marker detection method that were stated in sections 4.3 and 4.4. Following, I elaborate on the creation and the accuracy of the industrial marker to object map. In the next step, I discuss some details about the robot system in use and its precision. Then, I evaluate the Hand-Eye calibration and measurements of the end-effector mounted binocular camera system with respect to the robot base (RCS). After the evaluation of the robot's kinematic error, I test the extrapolation method and determine the absolute accuracy of the robot pose with respect to the target pose.

5.1. Equipment

To realize the extended visual servoing process and the industrial marker to object map, an industrial manipulator and a binocular camera system are required. In this section, I explain which equipment was used and why it was chosen for the realization of the mobile manipulator.

5.1.1. Industrial Manipulator – Universal Robot UR10

The robot used for the experiment was a UR10 from Universal Robots (Figure 70). It is a collaborative robot that can be used to interact with a human worker and stops in case of a collision. The properties of the robot are depicted in Table 4.

Repeatability	$\pm 0,1$ mm
Absolut accuracy	No specifications from the manufacturer
Collaboration operation	15 advanced adjustable safety functions
Payload	10 kg
Reach	1300 mm
Degree of freedom	6 rotating joints

Table 4: Universal Robot UR10 – specifications [65]

As mentioned in Table 4, there are no specifications concerning the absolute accuracy of the robot. The manufacturer has not conducted measurements regarding this topic. Therefore, I manufactured a reference plate and determined the accuracy myself (details in section 5.7.1). The measurements resulted in a position tolerance of about ± 1 mm and an orientation tolerance of about $\pm 1,5^\circ$. This robot was chosen because of its collaborative operation and its reach of 1300 mm. The main disadvantage is its poor absolute accuracy, which I will try to overcome with the use of a camera system and the concept of visual servoing.



Figure 70: Visualization of the Universal Robot UR10 [65]

5.1.2. Camera System

The camera system that is used for the detections is a stereo system which is mounted on the end-effector of the industrial manipulator (UR10). The camera properties were

chosen to enable a detection accuracy that is significantly smaller than the absolute accuracy of the UR10 (± 1 mm).



Figure 71: End-effector of the industrial robot with stereo system and gripper

Considering the previously explained pick-and-place task of this experiment, the requirements are defined as follows in Table 5.

Field of view (FOV)	450 mm x 450 mm
Camera distance (CD)	450 mm (chosen)
Smallest observable feature (SOF)	0,25 mm ($\ll 1$ mm)
Subpixel Accuracy (SPA)	0,25 (assumption)
Sampling factor (SF)	4 (chosen)

Table 5: Requirements for the camera observations

The required number of active pixels (AP), which is also known as resolution, can be determined as shown in Equation (11).

$$AP = SF \cdot \frac{FOV}{SOF} \cdot SPA = 4 \cdot \frac{450}{0,25} \cdot \frac{1}{4} = 1800$$

The resulting number of active pixels is used to define the camera. The camera must have at least 1800×1800 pixels (3,2 megapixel). I chose the 5 megapixel Jai GO camera of the type GO-5000-PGE. The properties of the camera are illustrated in Table 6.

Sensor	1" CMOS global shutter
Frame rate	22 frames/sec
Active area	12,8 mm (h) x 10,2 mm (v), 16,36 mm (diagonal)
Cell size	5,0 μm (h) x 5,0 μm (v)
Active pixels	2560 (h) x 2048 (v) (~ 5 megapixel)

Table 6: Jai GO-5000-PGE - camera properties [35]

The smallest dimension of the camera resolution is larger than the previously determined minimal number of active pixels ($2048 > 1800$). Using the camera properties of Table 6, one can calculate the required focal length according to Equation (11).

$$f = \frac{ActiveArea \cdot CD}{FOV}$$

$$f_1 = \frac{12,8 \cdot 450}{450} = 16,36 \text{ mm} \quad f_2 = \frac{10,2 \cdot 450}{450} = 10,2 \text{ mm}$$

I choose the focal length of our cameras to be 12,5 mm. In the next step, I can determine the field of depth using Equation (12).

$$DoF = \frac{CD}{1 \pm BS \cdot I \cdot \frac{CD - f}{f^2}} = \frac{450}{1 \pm 5 \cdot 10^{-3} \cdot 5,6 \cdot \frac{450 - 12,5}{12,5^2}}$$

$$DoF_{near} = 417,28 \text{ mm} \quad DoF_{far} = 488,28 \text{ mm}$$

The camera is mounted on the end-effector of the robot and moves towards the object during the extended visual servoing approach. Thus, the image is only in focus when moving in a range between DoF_{far} and DoF_{near} .

The distance CD is used to determine the required baseline of the stereo system as explained in section 3.2.7.

$$B = z/4 = 450/4 = 112,5 \text{ mm}$$

5.2. Monocular Camera System – Accuracy

To perform accurate measurements with a camera system, the parameters of the camera have to be known precisely. Therefore, I apply the calibration procedure, previously mentioned in section 3.2.3. The calibration is carried out by using a precise calibration plate (Figure 72). According to the distributor IDS Imaging Development Systems GmbH, the calibration plate is manufactured with a tolerance of $\pm 0,03$ mm between two points. The calibration plate was moved to 60 different locations in which both cameras of the stereo system were able to observe the target. The first 20 acquisitions of the plate were conducted while the plate was moved and rotated in a plane. For the other 40 positions the plate was additionally tilted.

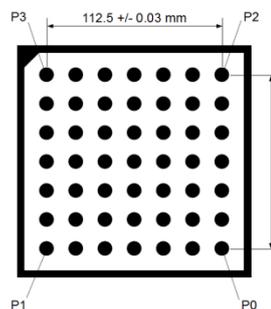


Figure 72: Monocular calibration – definition of the calibration plate and measurement points

The calibration process commented in section 3.2.3 is available in the machine vision software HALCON [49]. A successful calibration process returns a calibration error which corresponds to the average pixel distance between the optimal image coordinates and the projected calibration plate points. According to [51], an error up to 0.1

pixels indicates a good result. The development of the parameters of camera c_1 and camera c_2 during the calibration is shown in Figure 73 and in Figure 74. There is a significant change in the parameter values as the tilted calibration plate images are added to the calibration procedure.

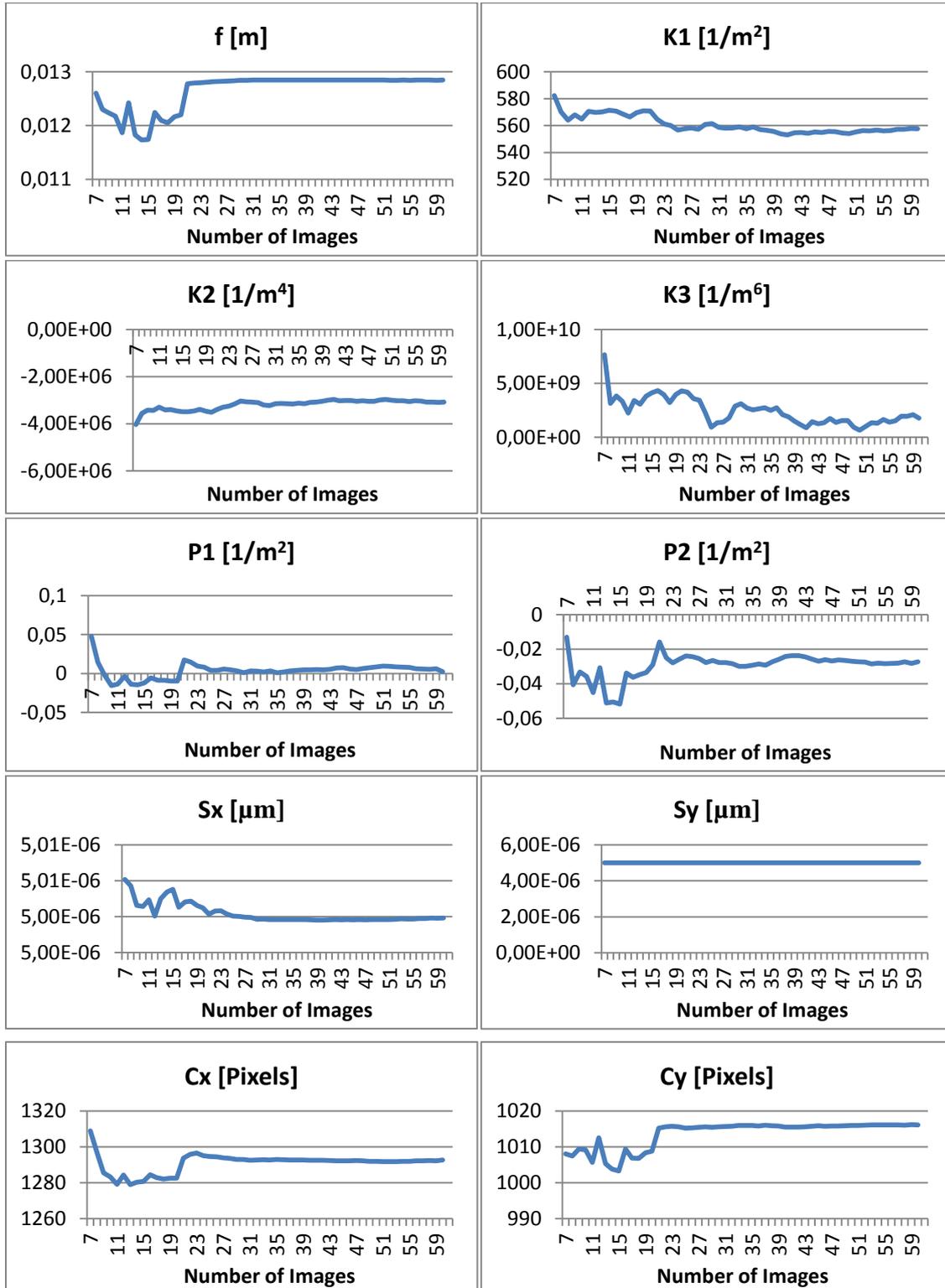


Figure 73: Monocular calibration of c_1 – development of camera parameters with growing number of calibration images

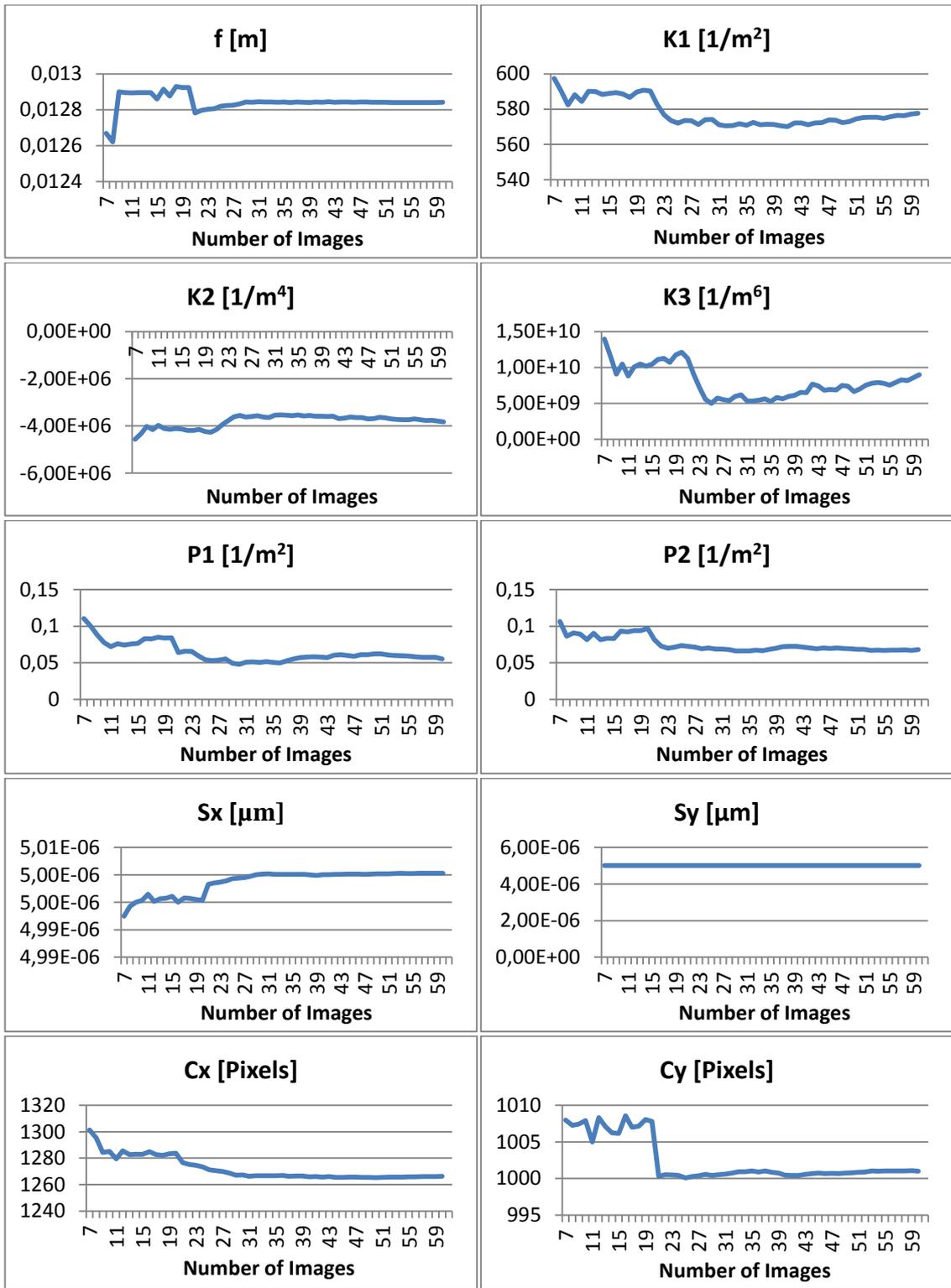


Figure 74: Monocular calibration of c_2 – development of camera parameters with growing number of calibration images

The calibration procedure optimizes the following parameters [49]:

- Focal length f [mm]
- Radial distortion parameters $K1$ [$1/m^2$], $K2$ [$1/m^4$] and $K3$ [$1/m^6$]
- Tangential distortion parameters $P1$ [$1/m^2$] and $P2$ [$1/m^2$]
- Horizontal and vertical distance of two cells Sx [μm] and Sy [μm]
- Column and row of optical center Cx [Pixels] and Cy [Pixels]

The resulting calibration error for both cameras is depicted in Table 7.

Mean calibration error of c_1	0,0968 Pixels
Mean calibration error of c_2	0,0968 Pixels

Table 7: Monocular calibration error of the left and right camera

The mean calibration error err_{calib} has to be multiplied to the size that is covered with one pixel to determine the mean measurement uncertainty err_{meas} of the monocular camera. The calculation is visualized in Equation (96).

$$err_{meas} = \frac{FOV}{ActiveArea} \cdot err_{calib} = \frac{450}{2048} \cdot 0,0968 = 0,021 \text{ mm} \quad (96)$$

To check the calibration results, I performed distance measurements between the points P1, P2, P3 and P4 on the calibration plate (illustrated in Figure 72). The results are presented in Table 8. err_{meas} describes only the mean distance error of a detected point from its actual point. As a consequence, when measuring a distance between two points, the tolerance is afflicted by $2 \cdot err_{meas}$.

	P0/P1 [mm]	P2/P3 [mm]	P0/P2 [mm]	P1/P3 [mm]
c_1	112,457	112,505	112,51	112,537
c_2	112,497	112,471	112,493	112,543

Table 8: Monocular 3D measurements in the calibration plane of the calibration plate corner points P1, P2, P3 and P4

The tolerance of the measurement is a combination of the manufacturer tolerance and the mean measurement uncertainty of the calibration determined by Equation (96). The tolerance of the distance measurement is therefore defined by Equation (97).

$$err = 0,03 + 2 \cdot err_{meas} = 0,03 + 2 \cdot 0,021 = 0,072 \text{ mm} \quad (97)$$

$$tolerance = \pm err = \pm 0,072 \text{ mm}$$

The maximal deviation of the conducted measurements from the nominal value of 112,5 mm is 0,043 mm and lies within the tolerance of $\pm 0,072$ mm.

5.3. Binocular Stereo System – Accuracy

After the calibration of c_1 and c_2 , I performed the binocular calibration as explained in section 3.2.6 to determine the transformation ${}^{c_1}T_{c_2}$ between the cameras. This pose in combination with the camera calibration of section 5.2 is later used to calculate the 3D position of distinctive feature points via triangulation.

This calibration method is also available in the machine vision software HALCON [49]. A successful calibration process returns the rectification error which has been mentioned in section 3.2.6. According to [51], this error has to be smaller than 0,1 pixels. I used the calibration images as stated in section 5.2. As before, the development of the calibration parameters changes significantly, as the tilted calibration plate images are added to the calibration procedure. The development of the transformation ${}^{c_1}T_{c_2}$ with a growing number of calibration images is shown in Figure 75.

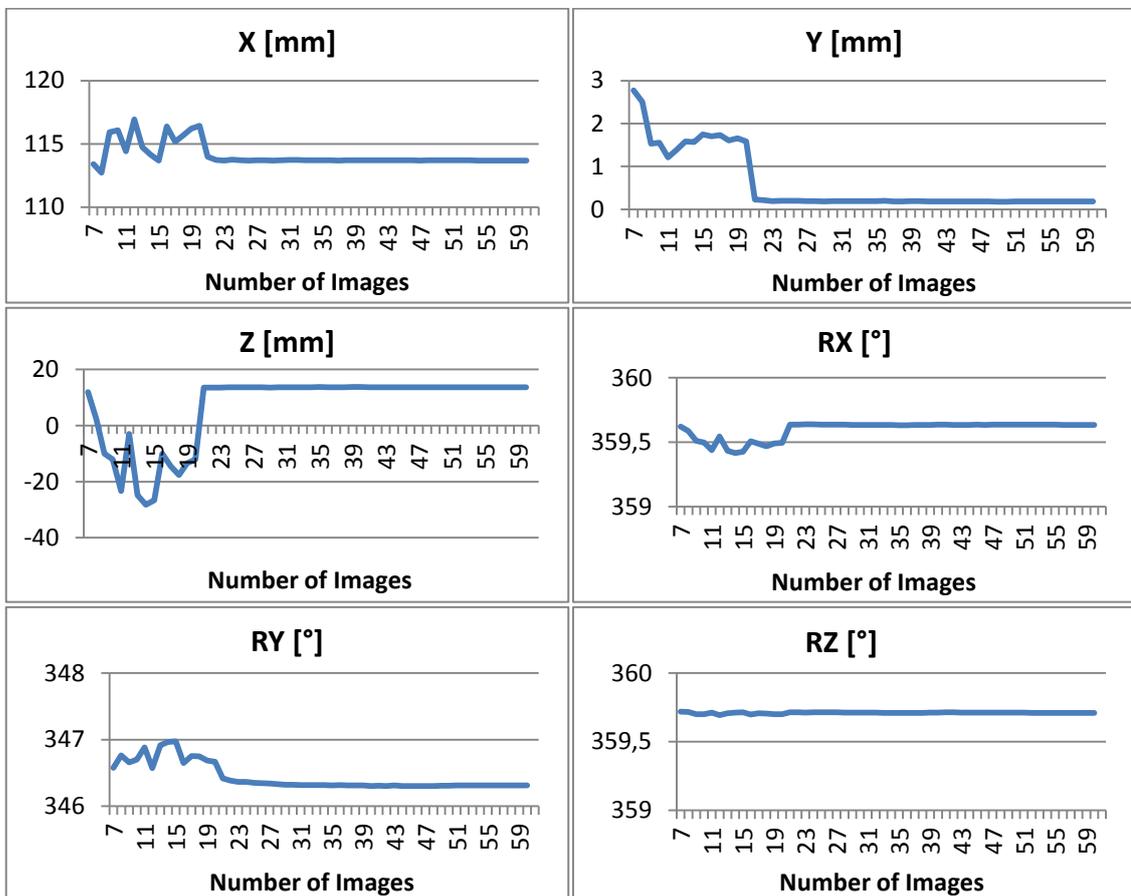


Figure 75: Binocular calibration – development of the pose parameters with growing number of calibration images

The resulting pose of the binocular calibration is demonstrated in Table 9.

${}^{c_1}T_{c_2}$	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
Mean	113,68	0,18	13,70	359,63	346,31	359,71

Table 9: Resulting parameters of ${}^{c_1}T_{c_2}$ with 60 different calibration images

Figure 76 depicts the rectification of two corresponding calibration images with the epipolar lines of the points of the last calibration plate column.

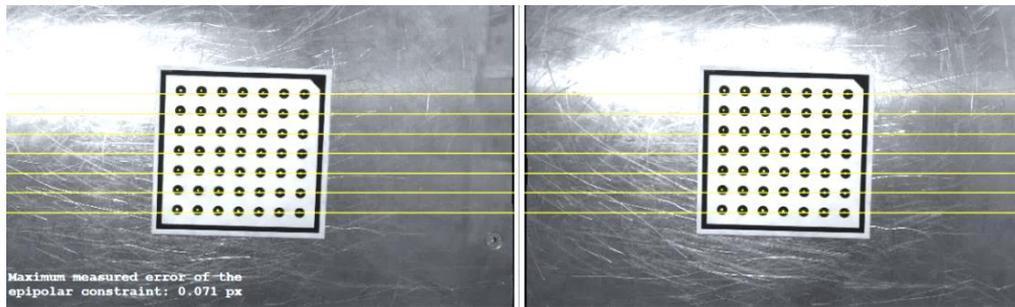


Figure 76: Binocular calibration – evaluation of the epipolar constraint

The calibration resulted in an epipolar constraint error of 0,071 pixels, which is smaller than the required 0,1 pixels. To check the calibration results, I performed distance measurements in 3D space between the points P1, P2, P3 and P4 on the calibration plate shown in Figure 72. The repeatability for 70 measurements is given in Table 10.

[mm]	X [mm]	Y [mm]	Z [mm]
range_{rep}	0,004	0,004	0,015
tol_{rep}	$\pm 0,005$	$\pm 0,004$	$\pm 0,02$

Table 10: Binocular calibration – tolerance of detection repeatability for 70 distance measurements of a static calibration plate

After the evaluation of the repeatability, 70 measurements were performed with the calibration plate randomly placed in a plane. These distance measurements between the corner points P0/P1, P2/P3, P0/P2 and P1/P3 are visualized in Figure 77.

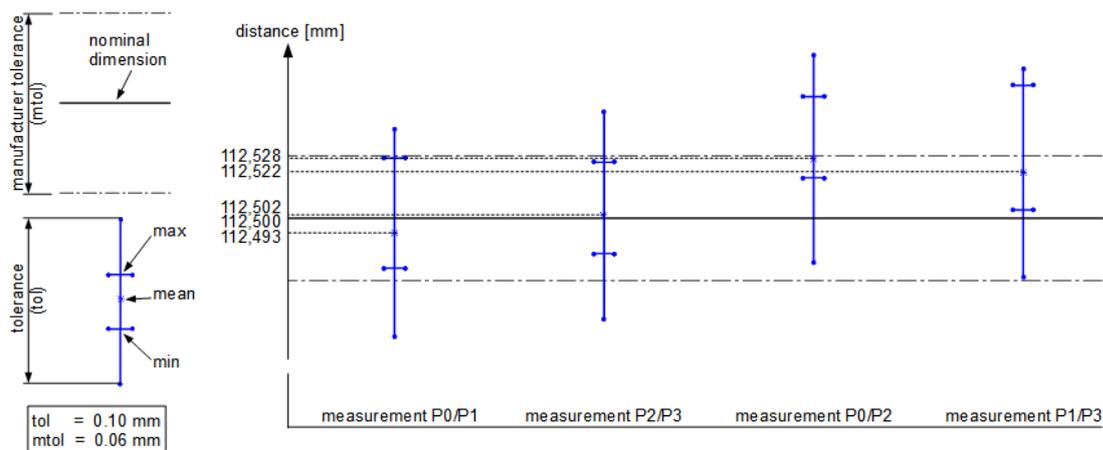


Figure 77: Binocular calibration – distance measurements between the corner points of the calibration plate at 70 random positions in a plane

Figure 77 shows the mean, maximal and minimal values that were taken for the distance measurements between P0/P1, P2/P3, P0/P2 and P1/P3. These values are

compared to the nominal distances and the manufacturer tolerance. Furthermore, the figure visualizes the tolerance tol ($\pm 0,05$ mm) which represents the determined measurement variation about the mean distance based on the C_m value of the process and machine capability (section 3.4). The mean values of all detections are within the given manufacturer tolerance $mtol$ ($\pm 0,03$ mm). Due to the fact that the actual distance values are not known, I consider a best and a worst case scenario for calculating the absolute measurement error of the stereo system. The worst case scenario is visualized in Figure 78.

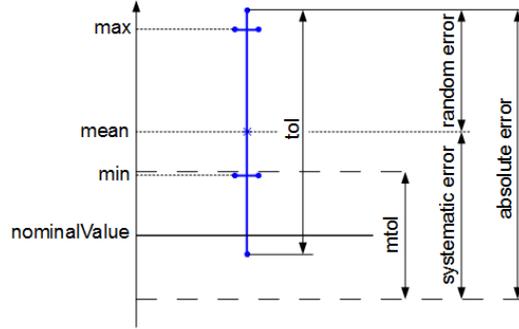


Figure 78: Worst case scenario of an absolute stereo system measurement error. The systematic error is assumed to be the maximum possible value

The worst case absolute error is composed of the worst case systematic error, which is represented by the maximum deviation between the mean and nominal value, and the random error, which is defined by the measurement deviation ($tol/2$). The absolute error is used to characterize the measurement tolerance. Since this measurement error describes the deviations with respect to a plane, I denote the tolerance as tol_{XY} . The resulting tolerance tol_{XY} is divided by the factor 2, to get a point and not a distance tolerance. The worst case accuracy is defined in Equation (98).

$$err_w = \left(\max(|nominalValue - mean|) + \frac{mtol}{2} + \frac{tol}{2} \right) \cdot \frac{1}{2}$$

$$err_w = \left(|112,5 - 112,528| + \frac{0,06}{2} + \frac{0,1}{2} \right) \cdot \frac{1}{2} = 0,054 \text{ mm} \quad (98)$$

$$tol_{XY_w} = \pm err_w = \pm 0,054 \text{ mm}$$

For the best case scenario, I assume that the mean measurement is equal to the actual manufactured value. Hence, the deviation is defined as follows in Equation (99).

$$tol_{XY_b} = \pm \left(\frac{tol}{2} \right) \cdot \frac{1}{2} = \pm \left(\frac{0,1}{2} \right) \cdot \frac{1}{2} = \pm 0,025 \text{ mm} \quad (99)$$

The actual accuracy of the stereo system distance measurements in a plane is somewhere between tol_{XY_b} and tol_{XY_w} .

In the next step, all points of the calibration plate for each of the 70 measurements were triangulated to get numerous 3D points. Due to the fact that the calibration plate was moved in a plane, the triangulated 3D points should also form a plane. Figure 79

shows the deviations of the 3D points from an optimally fitted plane and the tolerance based on the machine capability mentioned in section 3.4. The deviations from this plane are denoted as tol_Z .

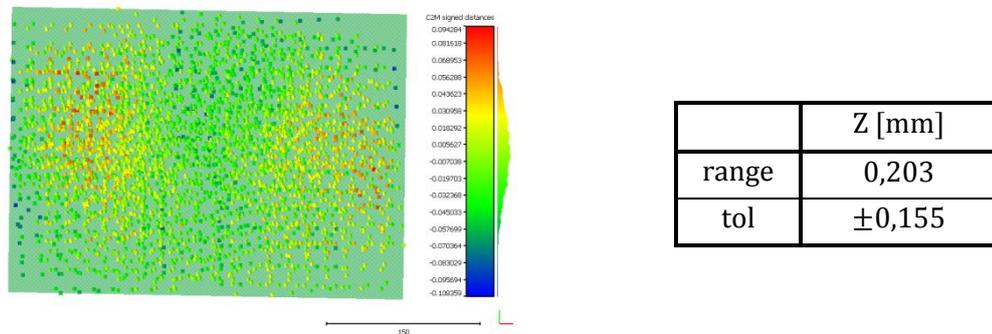


Figure 79: binocular calibration – Z deviation of the calibration plate at multiple positions in a plane

In conclusion, the overall worst case detection error of the stereo system is calculated as a combination of tol_{XY} and tol_Z (Equation (100)).

$$tol_{Stereo_w} = \pm \sqrt{tol_{XY}^2 + tol_Z^2} \quad (100)$$

$$tol_{Stereo_w} = \pm \sqrt{0,054^2 + 0,155^2} \approx \pm 0,17 \text{ mm}$$

The best case detection error of the stereo system is defined in Equation (101).

$$tol_{Stereo_b} = \pm \sqrt{tol_{XY}^2 + tol_Z^2} \quad (101)$$

$$tol_{Stereo_b} = \pm \sqrt{0,025^2 + 0,155^2} \approx \pm 0,16 \text{ mm}$$

The actual detection error of the binocular stereo system lies somewhere between $tol_{Stereo_b} = \pm 0,16 \text{ mm}$ and $tol_{Stereo_w} = \pm 0,17 \text{ mm}$.

5.4. 3D Object Detection – Method Evaluation

The calibrated camera system is now used to determine the 3D pose of a known object by using the methods described in section 4.3.1 (SBM), 4.3.2 (SBM-SCT) and 4.3.3 (SBM-ECT). The most accurate object detection will be applied in section 5.6 for the creation of the industrial marker to object map.

The object is asymmetric to ensure a unique definition of the orientation and is predefined by its CAD data (visualized in Figure 80 (left)). Furthermore, it has a rectangular outer contour to test the behavior of the object detection algorithm in case of simple contours. Rectangular shapes are quite common in an industrial image scene. Therefore, it is more difficult for the algorithm to identify the object. The coordinate system of the object was placed at the center of the hole with diameter $\varnothing 7,00$ mm.

To evaluate the accuracy of the object 3D pose detections I used a measurement plate that enables a defined positioning of the object with dowel pins (Figure 80 (right)). The measurement plate was utilized for inspecting the detection repeatability and for performing distance measurements between the positions P0, P1, P2 and P3 on the plate.

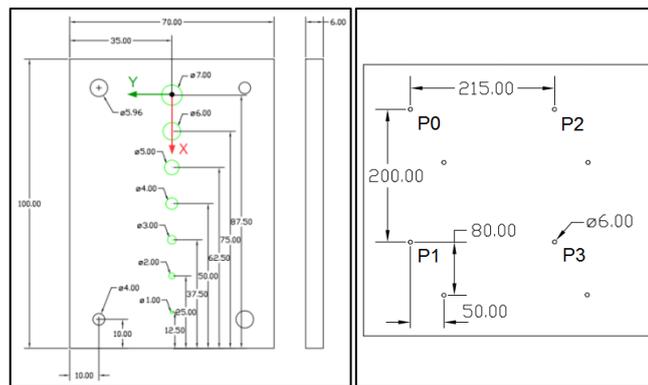


Figure 80: 3D object detection – rectangular object for detection (left) and measurement plate to evaluate the accuracy of the detection methods (right)

The four positions on the measurement plate are manufactured with a distance tolerance of $\pm 0,3$ mm to each other. The positioning of the object on the measurement plate with the use of dowel pins is visualized in Figure 81.

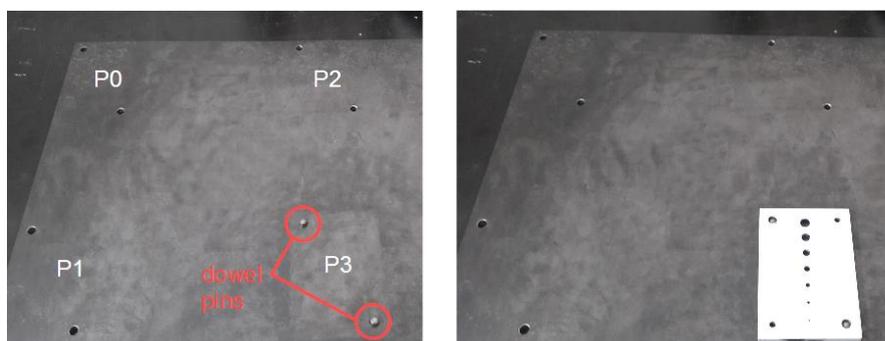


Figure 81: visualization of object positioning on measurement plate by fixing the object with dowel pins

5.4.1. SBM Accuracy

Firstly, the SBM algorithm as described in section 4.3.1 was tested. The object was placed and detected 70 times at each of the four positions P0, P1, P2 and P3 on the measurement plate. The repeatability of these measurements is shown in Table 11.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
$range_{rep}$	0,01	0,06	0,13	0,17	0,05	0,02
tol_{rep}	$\pm 0,02$	$\pm 0,1$	$\pm 0,16$	$\pm 0,215$	$\pm 0,055$	$\pm 0,02$

Table 11: SBM – tolerance of static object detection repeatability for 70 measurements

Secondly, the distances between P0/P1, P2/P3, P0/P2, P1/P3, P0/P3 and P1/P2 were observed (Figure 82). The evaluation of these measurements is performed similarly to the experiment conducted in section 5.3, Figure 77.

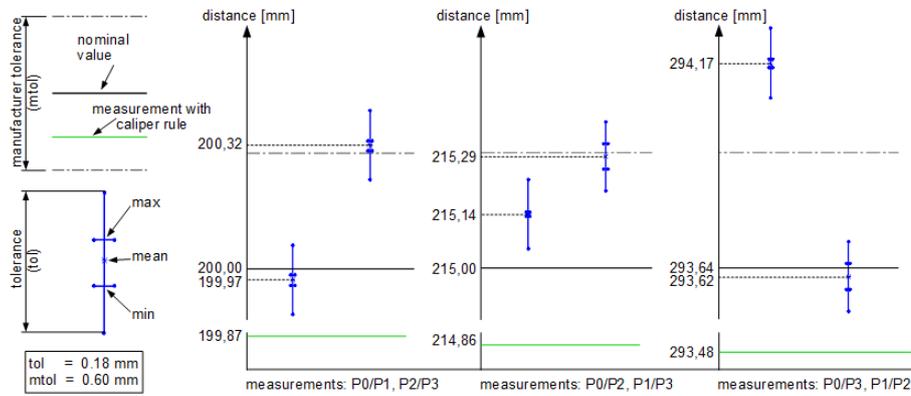


Figure 82: SBM – evaluation of the distance measurements between P0, P1, P2 and P3

The mean, maximal and minimal measurement and the variance tol ($\pm 0,09$ mm) are visualized and compared to the nominal distances and the manufacturer tolerance. The green line represents the distance between the points P0, P1, P2 and P3 that was measured with a caliper rule. This value is assumed to be the new actual distance value ($caliperValue$) with a tolerance of $caliperTol = \pm 0,1$ mm. Most of the measurements are within the given manufacturer tolerance of $mtol = \pm 0,3$ mm, but are far off the newly defined tolerance determined by the caliper rule.

The calculation of the worst and best case absolute measurement tolerance in a plane (tol_{XY}) of the SBM method is performed similarly to section 5.3. The main difference is that $mtol$ and $nominalValue$ are replaced by $caliperTol$ and $caliperValue$. Thus, the worst case absolute error is defined as presented in Equation (102).

$$err_w = \left(\max(|caliperValue - mean|) + \frac{caliperTol}{2} + \frac{tol}{2} \right) \cdot \frac{1}{2}$$

$$err_w = \left(|293,48 - 294,17| + \frac{0,1}{2} + \frac{0,18}{2} \right) \cdot \frac{1}{2} \approx 0,42 \text{ mm} \quad (102)$$

$$tol_{XY_w} = \pm err_w = \pm 0,42 \text{ mm}$$

The best case absolute measurement error is determined by assuming that the resulting mean distance is equal to the actual value. Consequently, the tolerance is defined as shown in Equation (103).

$$\text{tol}_{XY_b} = \pm \left(\frac{\text{tol}}{2} \right) \cdot \frac{1}{2} = \pm \left(\frac{0,18}{2} \right) \cdot \frac{1}{2} = 0,045 \text{ mm} \quad (103)$$

The actual static object detection tolerance using the SBM method in a plane is therefore somewhere between the tol_{XY_b} and tol_{XY_w} .

In the next step, the object was placed and detected at 70 randomly chosen positions in a plane. Due to the fact that the object was always put in the same plane, the calculated 3D points should also form a plane. Considering an optimal detection, the determined orientation angles α and β should stay constant. Figure 83 and Figure 84 show the deviations of tol_Z , tol_α and tol_β from an optimally fitted plane and the tolerance values of Z, α and β based on the machine capability mentioned in section 3.4. The visualization of tol_α and tol_β in Figure 84 was built by replacing the Z coordinates of each 3D point with the corresponding values of α and β .

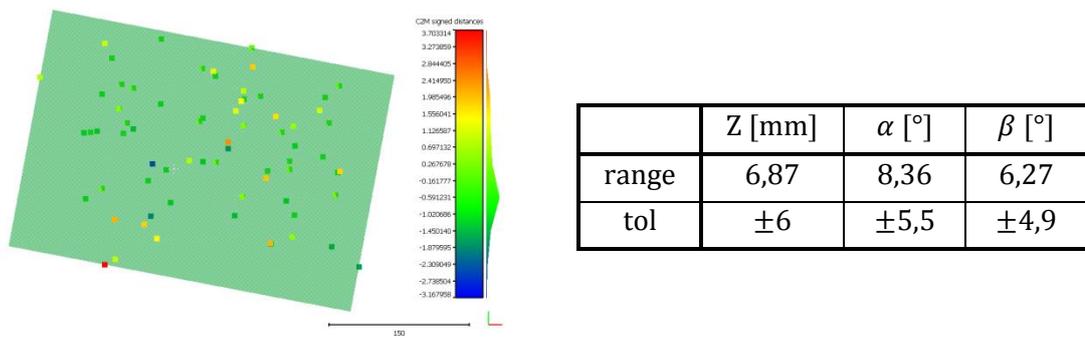


Figure 83: SBM – Z deviation of 70 different object positions in a plane and resulting tolerance of Z, α and β

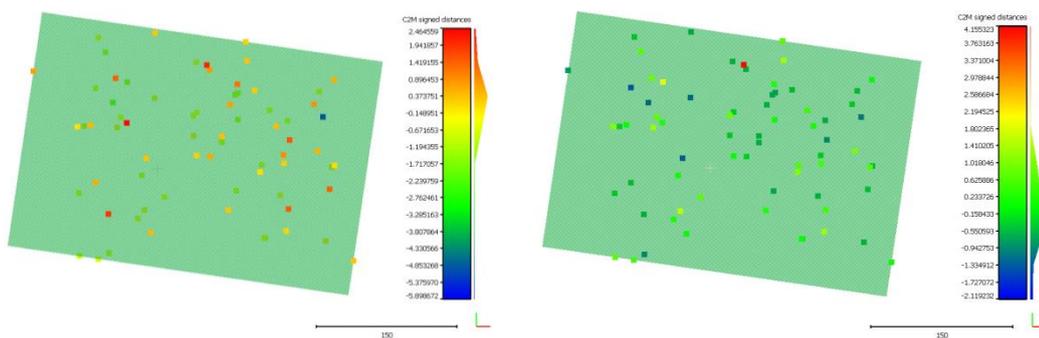


Figure 84: SBM – α and β deviation of 70 different object positions in a plane

The values of tol_Z , tol_α and tol_β are very high. The distribution of the colored points illustrates that this is a result of some outliers that cause a major increase in the standard deviation. Figure 83 and Figure 84 show that the distribution of Z, α and β has no visible patterns and that the outliers are randomly distributed. Causes for the deviations of Z, α and β to the optimally fitted plane are discussed in section 6.

A combination of tol_{XY} and tol_Z as introduced in section 5.3, Equation (100), makes no sense, considering that tol_{XY} was put up for a static object position, whereas tol_Z was determined for 70 randomly placed objects in a plane.

5.4.2. SBM-SOT Accuracy

In a further action, the object detection was performed with a stereo system by triangulating the object origin points of the matched shape model (SBM-SOT). All measurements in this section are executed and evaluated in the same way as the experiments conducted in section 5.4.1. Thus, the repeatability was determined with 70 detections at each of the four positions P0, P1, P2 and P3 on the measurement plate (Table 12).

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
$range_{rep}$	0,02	0,03	0,06	0,11	0,03	0,01
tol_{rep}	$\pm 0,02$	$\pm 0,04$	$\pm 0,05$	$\pm 0,12$	$\pm 0,04$	$\pm 0,02$

Table 12: SBM-SOT – tolerance of static object detection repeatability for 70 measurements

The distance measurements between the four measurement plate positions using the SBM-SOT are depicted in Figure 85.

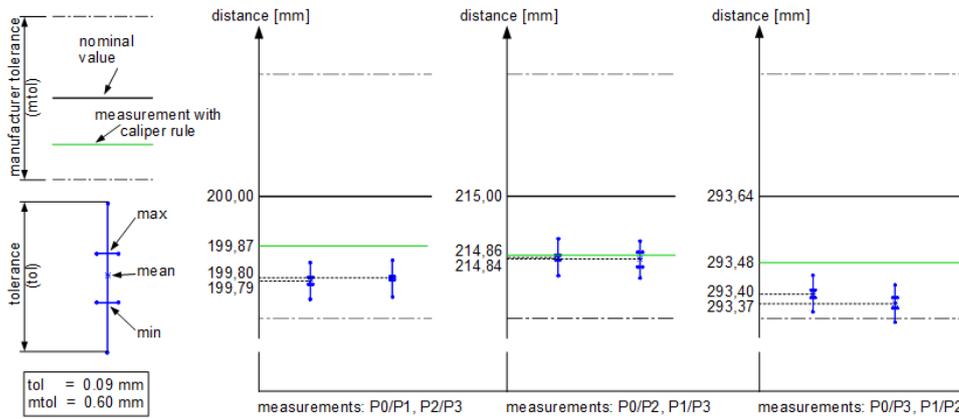


Figure 85: SBM-SOT – evaluation of the distance measurements between P0, P1, P2 and P3

The tolerance has decreased to $tol = \pm 0,045$ mm. Furthermore, the distances between P0/P1, P2/P3, P0/P2, P1/P3, P0/P3 and P1/P2 are now close to the tolerance defined by the caliper rule. The worst and best case absolute errors are presented in Equation (104) and Equation (105). The derivation is described in section 5.4.1.

$$err_w = \left(\max(|caliperValue - mean|) + \frac{caliperTol}{2} + \frac{tol}{2} \right) \cdot \frac{1}{2}$$

$$err_w = \left(|293,48 - 293,37| + \frac{0,1}{2} + \frac{0,09}{2} \right) \cdot \frac{1}{2} \approx 0,1 \text{ mm} \quad (104)$$

$$tol_{XY_w} = \pm err_w = \pm 0,1 \text{ mm}$$

$$\text{tol}_{XY_b} = \pm \left(\frac{\text{tol}}{2}\right) \cdot \frac{1}{2} = \pm \left(\frac{0,09}{2}\right) \cdot \frac{1}{2} \approx \pm 0,023 \text{ mm} \quad (105)$$

As a result, the actual tolerance of the static SBM-SOT detection in a plane is somewhere between tol_{XY_b} and tol_{XY_w} .

In the next step, tol_Z , tol_α and tol_β were determined as presented in section 5.4.1. Figure 86 and Figure 87 illustrate the visualization of the Z, α and β deviation from an optimally fitted plane.

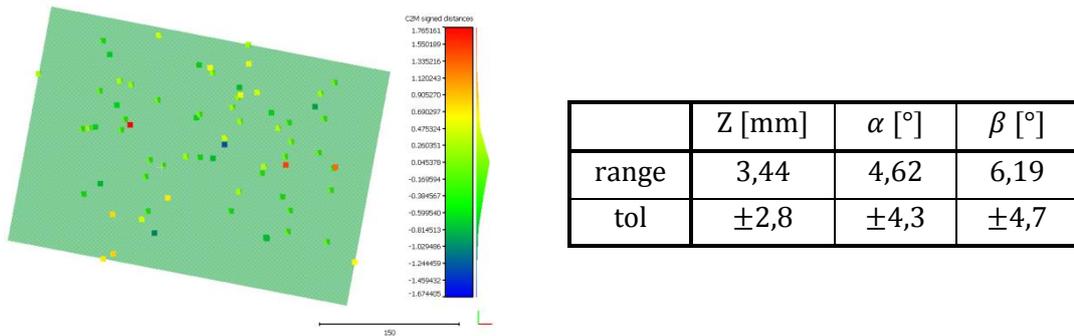


Figure 86: SBM-SOT – Z deviation of 70 different object positions in a plane and resulting tolerance of Z, α and β

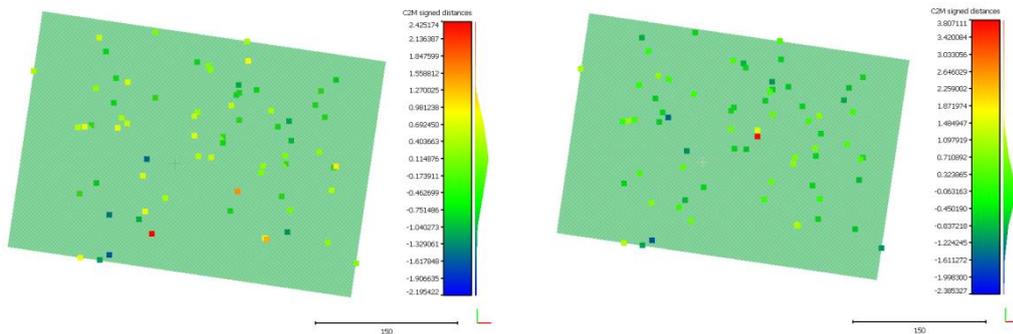


Figure 87: SBM-SOT – α and β deviation of 70 different object positions in a plane

The tolerances tol_Z , tol_α and tol_β are still very high, but were reduced significantly in contrast to the results determined in section 5.4.1. The distribution of the points depicted in Figure 86 and Figure 87 leads to the conclusion that the measurements of the SBM-SOT are also strongly influenced by outliers. These outliers are randomly distributed and have no visible patterns. Accounts for possible inaccurate detections will follow in section 6.

As mentioned before, there is no point in combining tol_{XY} and tol_Z .

5.4.3. SBM-ECT Accuracy

Finally, the accuracy of the SBM-ECT method (section 4.3.3) was evaluated by performing the previously mentioned set of experiments (section 5.4.1, section 5.4.2). The results of the repeatability are shown in Table 13.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
$range_{rep}$	0,06	0,07	0,22	0,09	0,03	0,02
tol_{rep}	$\pm 0,07$	$\pm 0,07$	$\pm 0,22$	$\pm 0,11$	$\pm 0,04$	$\pm 0,03$

Table 13: SBM-ECT – tolerance of static object detection repeatability for 70 measurements

Afterwards, the distances between the four measurement plate positions are evaluated. The results are visualized in Figure 88.

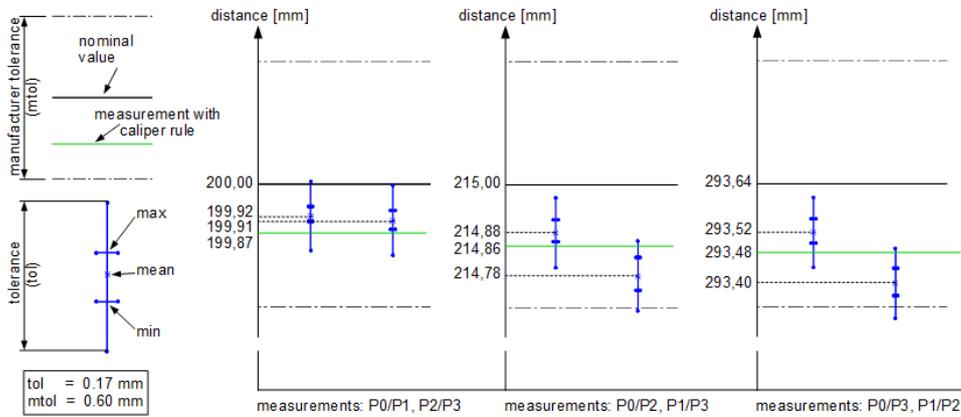


Figure 88: SBM-ECT – distance evaluation of measurement plate

The distance measurements led to a tolerance of $tol = \pm 0,085$ mm. Similarly to the SBM-SOT method, the determined distances between P0/P1, P2/P3, P0/P2, P1/P3, P0/P3 and P1/P2 are very close to the tolerance defined by the caliper rule. Based on the explanation in section 5.4.1, the worst and best case absolute errors of the SBM-ECT are defined as shown in Equation (106) and Equation (107).

$$err_w = \left(\max(|caliperValue - mean|) + \frac{caliperTol}{2} + \frac{tol}{2} \right) \cdot \frac{1}{2}$$

$$err_w = \left(|293,48 - 293,4| + \frac{0,1}{2} + \frac{0,17}{2} \right) \cdot \frac{1}{2} \approx 0,11 \text{ mm} \quad (106)$$

$$tol_{XY_w} = \pm err_w = \pm 0,11 \text{ mm}$$

$$tol_{XY_b} = \pm \left(\frac{tol}{2} \right) \cdot \frac{1}{2} = \pm \left(\frac{0,17}{2} \right) \cdot \frac{1}{2} \approx \pm 0,043 \text{ mm} \quad (107)$$

The actual tolerance of the static SBM-ECT detections in a plane is interpreted to be somewhere between tol_{XY_b} and tol_{XY_w} .

The calculation of tol_Z , tol_α and tol_β was executed and evaluated as explained in section 5.4.1. The deviations from an optimally fitted plane with respect to the SBM-ECT method are illustrated in Figure 89 and Figure 90.

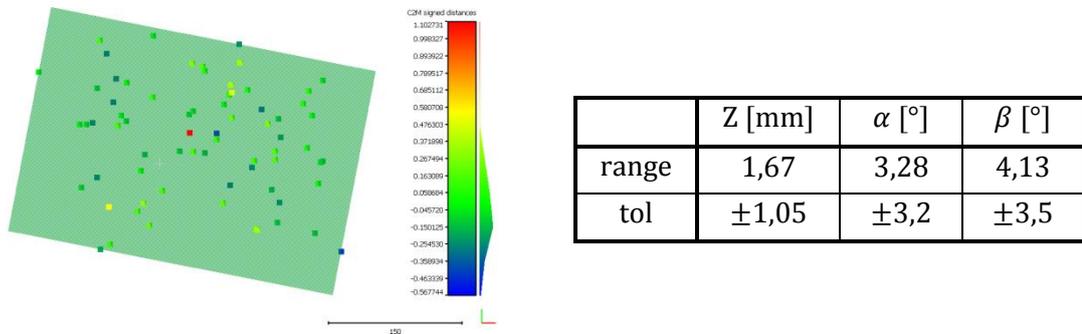


Figure 89: SBM-ECT – Z deviation of 70 different object positions in a plane and resulting tolerance of Z, α and β

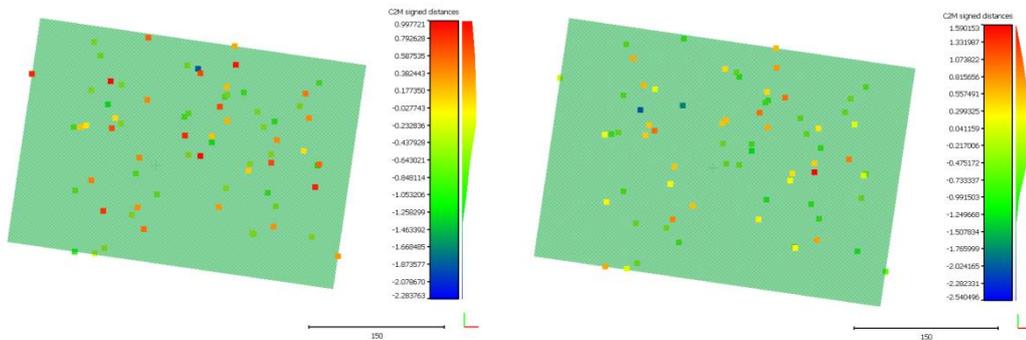


Figure 90: SBM-ECT – α and β deviation of 70 different object positions in a plane

The tolerances tol_Z , tol_α and tol_β are still high, but smaller than those of the SBM and SBM-SOT method. The distribution of the colored points in Figure 89 and Figure 90 illustrates, that the SBM-ECT is also afflicted with outliers. The reasons for the inaccurate detections are discussed in section 6.

As stated already, a combination of tol_{XY} and tol_Z is pointless, since tol_{XY} was determined for a static object position, whereas tol_Z was determined for 70 randomly placed objects in a plane.

5.4.4. Erroneous 3D Pose Detections

Sections 5.4.1 (SBM), 5.4.2 (SBM-SOT) and 5.4.3 (SBM-ECT) show that the results of the object detections are sometimes afflicted with severe outliers. In order to determine the source of the errors, I conducted a number of experiments that are reported in this section. Each of the following test cases was carried out with the same object in the same location.

At first, I performed a valuable object detection to get a reference for later detections. The projection of the 3D CAD model to the acquired image results in an almost perfect correlation (Figure 91). This detection was performed by using a good camera calibra-

tion and by illuminating the object from all sides. The determined object poses of the three detection methods just explained are shown in Table 14.

Method	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
SBM	-33,51	56,44	420,49	343,05	356,52	155,41
SBM – SOT	-33,11	56,29	419,48	343,14	356,51	155,44
SBM – ECT	-33,36	56,45	419,85	343,10	356,52	155,43

Table 14: Pose results of an optimal object detection using an accurate camera calibration and a homogenous illumination of the object

The results are very similar to each other. The main difference between the methods is the Z value. In general, the results of the SBM-SOT and SBM-ECT methods are assumed to be more accurate due to the usage of the stereo system.

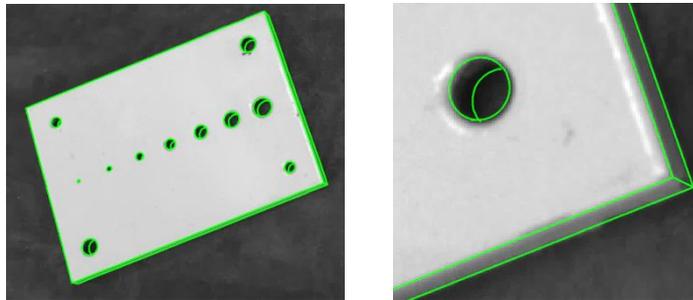


Figure 91: visualization of the 3D CAD model projection of a good object detection

In the first error scenario, I performed the detections of the objects with an inaccurate calibration. I slightly changed the calibration file to simulate a small deviation in the focal length of the camera system. The focal length was reduced by 0,1 mm. Figure 92 illustrates that the projection of the 3D CAD model correlates almost exactly with the object in the acquisition image. The object poses depicted in Table 15, on the other hand, changed significantly in contrast to the reference pose visualized in Table 14.

Method	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
SBM	-32,76	56,43	417,51	343,06	356,50	155,41
SBM – SOT	-32,98	56,30	418,14	343,03	356,46	155,41
SBM – ECT	-33,25	56,49	418,46	343,04	356,47	155,41

Table 15: inaccurate object detection due to an inaccurate camera calibration

This test case shows that small deviations of the camera calibration have some major effects on the position of the SBM algorithm. The deviations of the resulting position for the SBM-SOT and SBM-ECT method are smaller, but still not negligible. The detected orientation of the object did not change at all.

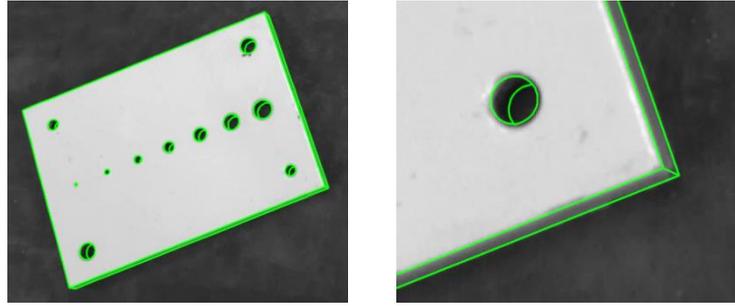


Figure 92: inaccurate object detection due to an inaccurate camera calibration

Furthermore, I ran tests on the object detection using different illumination scenarios. Changing the illumination led to severe inaccuracies of the fitted shape model (Figure 93). The pose values are shown in Table 16.

Method	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
SBM	-36,88	57,30	427,21	342,98	355,69	155,33
SBM – SOT	-32,88	55,79	416,22	343,07	356,04	155,34
SBM – ECT	-33,35	56,23	419,69	343,142	356,31	155,42

Table 16: inaccurate object detection due to an inaccurate camera calibration

The detected object poses with SBM and SBM-SOT depicted in Table 16 deviate severely from the detections in Table 14. Figure 93 illustrates that some of the object contours in the scene were not found due to suboptimal illumination. Therefore, the projected 3D model does not correlate with the actual object in the scene. The results of the SBM-ECT method are still very accurate on grounds of the independence of the pose calculation from the fitted shape model.

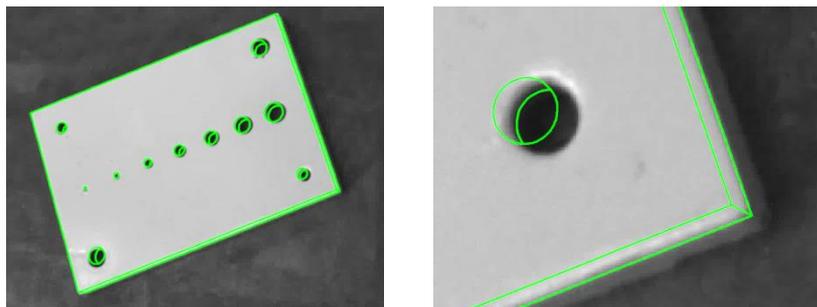


Figure 93: inaccurate object detection due to an inaccurate camera calibration

While performing these tests, I further noticed small differences between the resulting poses of the SBM-SOT and SBM-ECT method (as represented in Table 14). An evaluation clarified that this deviation is a result of an inaccurate ellipse fitting. Figure 94 shows a close acquisition of the hole at the origin of the CAD model that is used as a distinctive feature for the SBM-ECT method. The ellipse fitting procedure of this hole delivers different results depending on the illumination due to the chamfer.

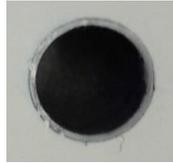


Figure 94: Acquisition of the hole that represents the object origin of the shape model

Figure 95 a) and b) show the result of the SBM-SOT and SBM-ECT. The fitting of the shape model using SBM-SOT results in a better description of the object origin point. Despite the fact that the chamfer is not included in the shape model, the fitting is correct. Due to bad illumination, shadowing occurs and one half of the chamfer is dark. The ellipse fitting procedure of the SBM-ECT associates the shadowed side of the chamfer as a hole and includes it in the determination of the center of the hole. A comparison of these two methods is provided by Figure 95 c).

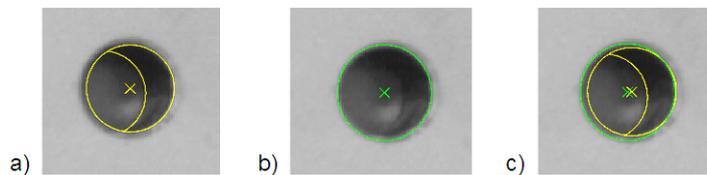


Figure 95: a) resulting origin of shape model fitting (SBM-SOT); b) resulting hole center using ellipse fitting (SBM-ECT); c) comparison between SBM-SOT and SBM-ECT

The difference of the two detected center points results in the deviation of the SBM-SOT and SBM-ECT method. The ellipse fitting algorithm also reaches its limits in case of noise close to the distinctive feature. Figure 96 shows irregularities near the distinctive feature that cause a false estimation of the edges of the hole.

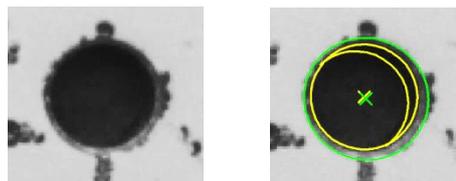


Figure 96: comparison between the origin detection of the SBM fitting and the ellipse fitting of the hole in the origin with irregularities in the image

The projection of the 3D CAD model according to the result of the SBM-SOT method delivered a good correlation with the actual object and therefore a good object origin estimation. The SBM-ECT, on the other hand, resulted in an ellipse which included the irregularities. Such being the case, it can be inferred that the estimation of the hole center is wrong.

5.5. 3D Marker Detection – Method Evaluation

The accuracy of the 3D markers mentioned in section 4.4.1 (3D ARTM-M), 4.4.2 (3D ARTM-B), 4.4.3 (3D QRCM) and 4.4.4 (3D FPM) is determined in this part of my experiments. The most accurate marker is then further used in section 5.6, to build the industrial marker to object map mentioned in section 4.2.

To evaluate the accuracy of the 3D markers, four 3D markers of each marker type were printed on an aluminium composite plate with a defined distance to each other. According to the manufacturer, the markers were printed with an absolute accuracy of $\pm 0,1$ mm to each other. The aluminium composite plate was then used to evaluate the repeatability and to perform distance measurements between the four marker positions at various locations of the plate in a plane.

5.5.1. ARToolKit Marker – Monocular (3D ARTM-M)

First, the accuracy of the 3D ARToolKit marker with monocular camera detection is evaluated. The printed measurement plate with the marker identification patterns A, B, F and G is visualized in Figure 97.

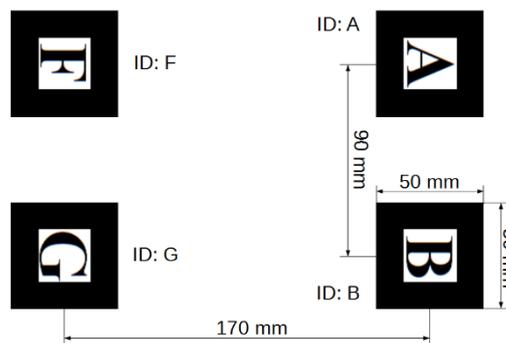


Figure 97: 3D ARTM-M – printed measurement plate with four markers with the identification patterns A, B, F and G

Equivalent to the previously mentioned object detection, the marker measurement plate was placed in the field of view of the camera to perform 70 static detections of each marker. The repeatability of these measurements is shown in Table 17.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
$range_{rep}$	0,04	0,06	0,37	0,34	0,31	0,04
tol_{rep}	$\pm 0,04$	$\pm 0,06$	$\pm 0,36$	$\pm 0,32$	$\pm 0,3$	$\pm 0,05$

Table 17: 3D ARTM-M – tolerance of static marker detection repeatability for 70 measurements

Afterwards, the marker plate was randomly placed and detected at 70 different positions in a plane. The gathered distance measurements between the markers on the plate are represented in Figure 98. It shows the mean, maximal and minimal values that were determined at the distance measurements between A/B, F/G, A/F and B/G. These measurements are compared to the nominal distances and the manufacturer

tolerance. Furthermore, Figure 98 illustrates the tolerance tol ($\pm 3,6$ mm) which represents the measurement variation about the mean value based on the C_m value of the process and machine capability (section 3.4). It is clearly visible that the mean values are not within the range of manufacturer tolerance $mtol$ and that tol exceeds the possible measurement tolerance of a stereo system shown in section 5.3.

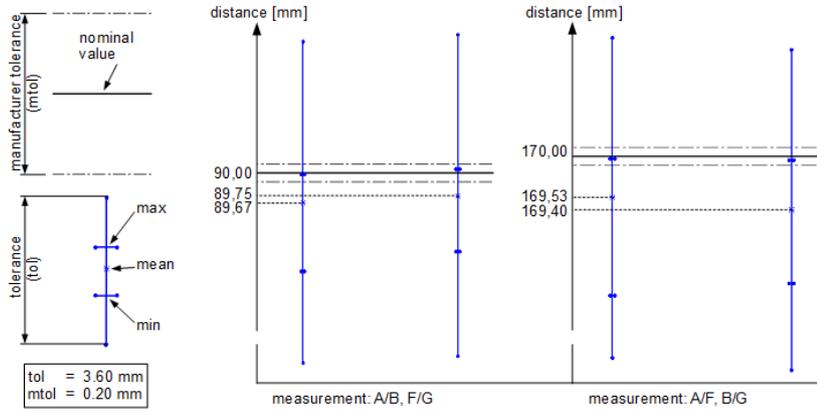


Figure 98: 3D ARTM-M – distance evaluation of measurement plate

The calculation of the measurement tolerance in a plane tol_{XY} is performed equally to the determination of the absolute error for the binocular camera system conducted in section 5.3. The resulting tol_{XY} is divided by the factor 2 to get a point and not a distance tolerance. I consider a worst and best case absolute measurement scenario in a plane. The worst case absolute measurement error of the monocular 3D ARToolKit marker detection in the plane is defined in Equation (108).

$$err_w = \left(\max(|actualValue - mean|) + \frac{mtol}{2} + \frac{tol}{2} \right) \cdot \frac{1}{2}$$

$$err_w = \left(|170 - 169,4| + \frac{0,2}{2} + \frac{3,6}{2} \right) \cdot \frac{1}{2} \approx 1,3 \text{ mm} \quad (108)$$

$$tol_{XY_w} = \pm err_w = \pm 1,3 \text{ mm}$$

The best case absolute measurement error is calculated by assuming that the mean distance is equal to the actual value. Hence, the tolerance can be defined as presented in Equation (109).

$$tol_{XY_b} = \pm \left(\frac{tol}{2} \right) \cdot \frac{1}{2} = \pm \left(\frac{3,6}{2} \right) \cdot \frac{1}{2} = \pm 0,9 \text{ mm} \quad (109)$$

The actual tolerance of the monocular 3D ARToolKit marker distance measurements in a plane is somewhere between tol_{XY_b} and tol_{XY_w} .

Since the distance measurements were conducted in a plane, the triangulated 3D marker points should also form a plane and the orientation of α and β should stay constant. Figure 99 and Figure 100 show the deviations tol_z , tol_α and tol_β from an optimally fitted plane and the tolerance of Z, α and β based on the machine capability. The

visualization of tol_α and tol_β in Figure 100 was built by replacing the Z coordinates of each 3D point with the corresponding values of α and β .

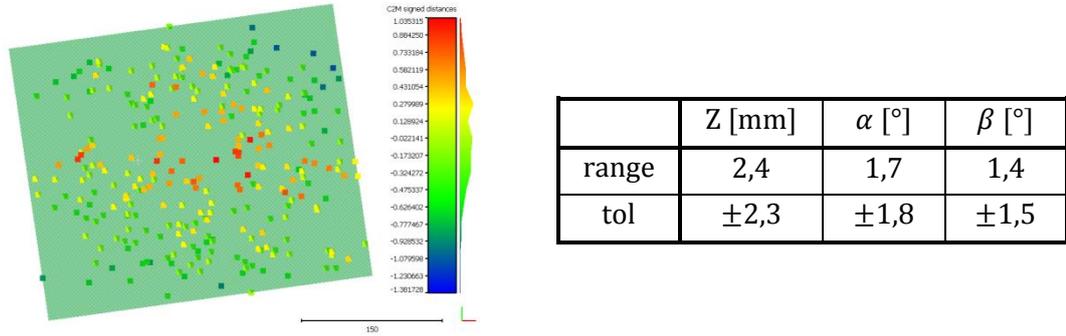


Figure 99: 3D ARTM-M – Z deviation of 70 different object positions in a plane and resulting tolerance of Z, α and β

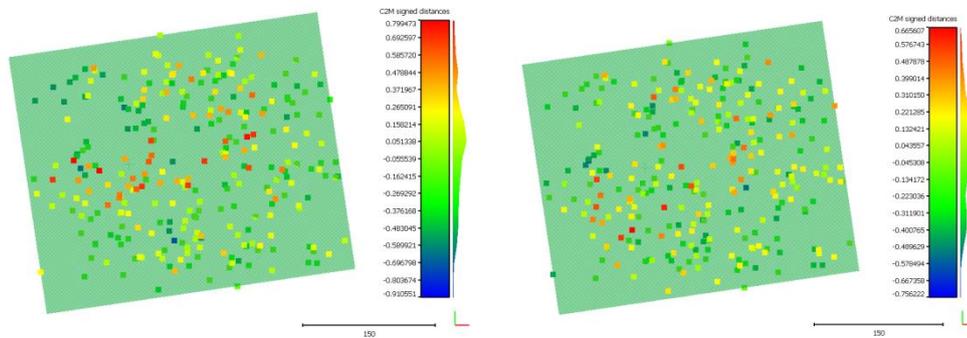


Figure 100: 3D ARTM-M – α and β deviation of 70 different object positions in a plane

The tolerances tol_Z , tol_α and tol_β are very high. Figure 99 shows that the deviations of Z are not randomly distributed. The 3D points in the middle of the plane are detected to be closer to the camera (red points) than the points near the border region (blue/green points). Furthermore, the color representation of the deviations in Figure 100 illustrates that the distribution of α and β also has visible patterns. The orientation changes with the distance of the points to the center of the field of view.

In conclusion, the overall worst case 3D position accuracy of the monocular detection of the 3D ARTToolkit marker is defined by combining the tolerance tol_{XY_w} and tol_Z (Equation (110)).

$$tol_{ARTM-M_w} = \pm \sqrt{tol_{XY_w}^2 + tol_Z^2} = \pm \sqrt{1,3^2 + 2,3^2} \approx \pm 2,6 \text{ mm} \quad (110)$$

The best case detection error is presented in Equation (111).

$$tol_{ARTM-M_b} = \pm \sqrt{tol_{XY_b}^2 + tol_Z^2} = \pm \sqrt{0,9^2 + 2,3^2} \approx \pm 2,5 \text{ mm} \quad (111)$$

The actual error of the monocular 3D ARTToolkit marker detection is somewhere between $tol_{ARTM-M_b} = \pm 2,5 \text{ mm}$ and $tol_{ARTM-M_w} = \pm 2,6 \text{ mm}$.

5.5.2. ARToolKit Marker – Binocular (3D ARTM-B)

The next step included the replacing of the monocular camera system with a binocular camera system to detect the 3D ARToolKit markers depicted in Figure 97. To evaluate the difference between the monocular and binocular system, all measurements performed in section 5.5.1 were repeated. The resulting repeatability is shown in Table 18.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
$range_{rep}$	0,01	0,01	0,06	0,2	0,2	0,05
tol_{rep}	$\pm 0,02$	$\pm 0,02$	$\pm 0,06$	$\pm 0,2$	$\pm 0,2$	$\pm 0,06$

Table 18: 3D ARTM-B – tolerance of static marker detection repeatability for 70 measurements

As described in section 5.5.1, the marker plate was randomly placed and detected at 70 different positions in a plane. The gathered distance measurements between the markers on the plate are compared with the nominal distance and the manufacturer tolerance of $\pm 0,10$ mm (Figure 101).

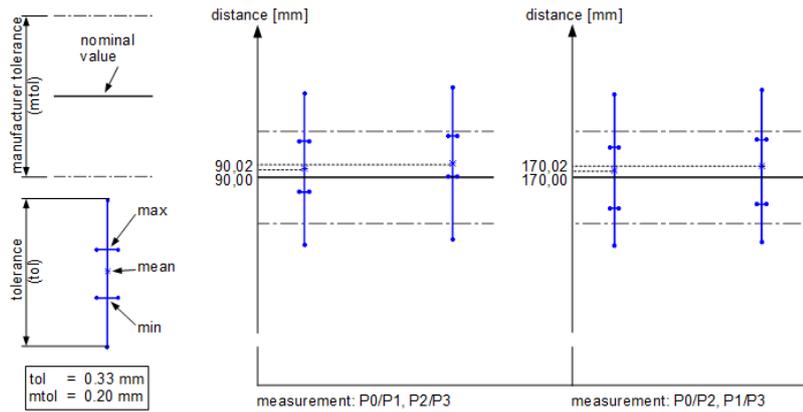


Figure 101: 3D ARTM-B – distance evaluation of measurement plate

The surveyed mean distance values are close to the expected nominal values. Also, compared to the previous section, the tolerance tol ($\pm 0,33$ mm) could significantly be reduced. All conducted distance measurements are within the manufacturer tolerance. The calculation of the tolerance in a plane tol_{XY} is performed equally to section 5.5.1. The worst and best case absolute measurement errors of the binocular 3D ARToolKit marker detection in the plane are defined in Equation (112) and Equation (113).

$$err_w = \left(\max(|actualValue - mean|) + \frac{mtol}{2} + \frac{tol}{2} \right) \cdot \frac{1}{2}$$

$$err_w = \left(|170 - 170,02| + \frac{0,2}{2} + \frac{0,33}{2} \right) \cdot \frac{1}{2} \approx 0,14 \text{ mm} \quad (112)$$

$$tol_{XY_w} = \pm err_w = \pm 0,14 \text{ mm}$$

$$tol_{XY_b} = \pm \left(\frac{tol}{2} \right) \cdot \frac{1}{2} = \pm \left(\frac{0,33}{2} \right) \cdot \frac{1}{2} \approx \pm 0,08 \text{ mm} \quad (113)$$

As a result, the actual tolerance of the binocular 3D ARToolKit marker is defined to be somewhere between tol_{XY_b} and tol_{XY_w} .

Equivalent to section 5.5.1, the distance measurements were also used to determine the deviations tol_Z , tol_α and tol_β as illustrated in Figure 102 and Figure 103.

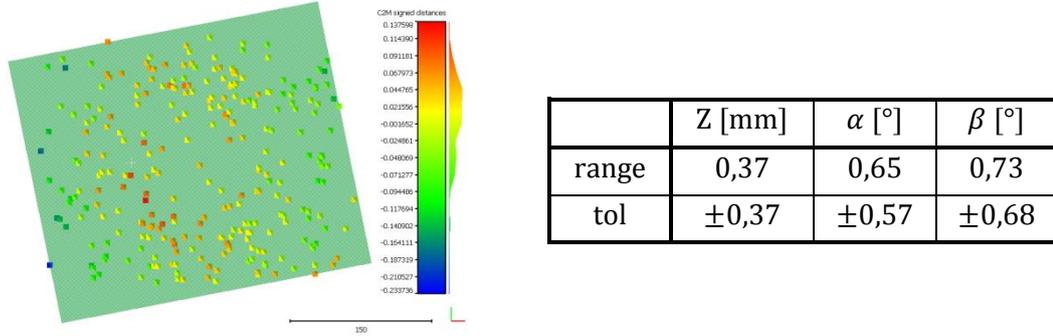


Figure 102: 3D ARTM-B – Z deviation of 70 different object positions in a plane and resulting tolerance of Z, α and β

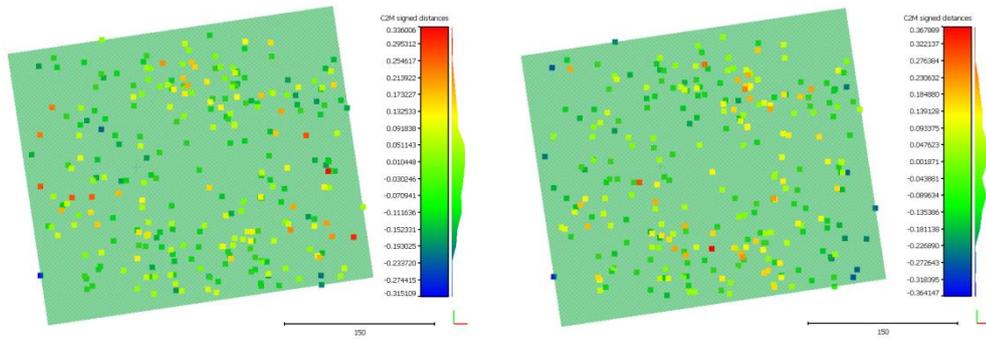


Figure 103: 3D ARTM-B – α and β deviation of 70 different object positions in a plane

The Z deviations are not randomly distributed (Figure 102). Similarly to the deviations presented in section 5.5.1, the points in the middle appear closer to the camera system (red points) than the ones at the border region. Figure 103 visualizes that the distribution of the measurement outliers of α and β are also not randomly distributed. The orientations at the border region seem to be lower than those close to the image center. Compared to the results of section 5.5.1, the deviations decreased significantly due to the use of a binocular camera system. The worst and best case 3D position accuracy is defined by Equation (114) and Equation (115).

$$tol_{ARTM-B_w} = \pm \sqrt{tol_{XY_w}^2 + tol_Z^2} = \pm \sqrt{0,14^2 + 0,37^2} \approx \pm 0,40 \text{ mm} \quad (114)$$

$$tol_{ARTM-B_b} = \pm \sqrt{tol_{XY_b}^2 + tol_Z^2} = \pm \sqrt{0,08^2 + 0,37^2} \approx \pm 0,38 \text{ mm} \quad (115)$$

The actual error of the binocular 3D ARToolKit marker detection is somewhere between $tol_{ARTM-B_b} = \pm 0,38 \text{ mm}$ and $tol_{ARTM-B_w} = \pm 0,40 \text{ mm}$.

5.5.3. 3D QR Code Marker (3D QR Code)

The experiment is proceeded with an evaluation of the detection accuracy of the 3D QR Code marker with a binocular system. The measurement plate with the four 3D markers and the identification patterns Code1, Code2, Code3 and Code4 that were used for the evaluation is visualized in Figure 104.

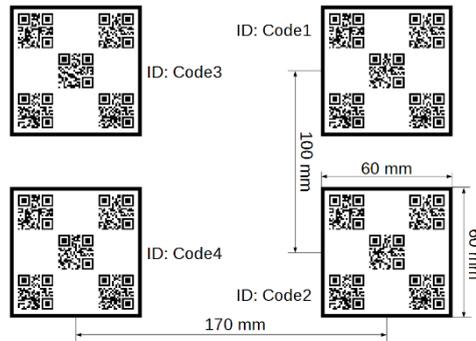


Figure 104: 3D QR Code – printed measurement plate with four markers with the identification patterns Code1, Code2, Code3 and Code4

Equivalent to section 5.5.2, all experiments of section 5.5.1 were repeated in this segment to get results that are comparable to the previous segments. The repeatability of the 3D QR Code detection is shown in Table 19.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
$range_{rep}$	0,004	0,007	0,02	0,023	0,031	0,007
tol_{rep}	$\pm 0,005$	$\pm 0,01$	$\pm 0,02$	$\pm 0,025$	$\pm 0,035$	$\pm 0,01$

Table 19: 3D QR Code – tolerance of static marker detection repeatability for 70 measurements

The distance measurements between the 3D markers are depicted in Figure 105.

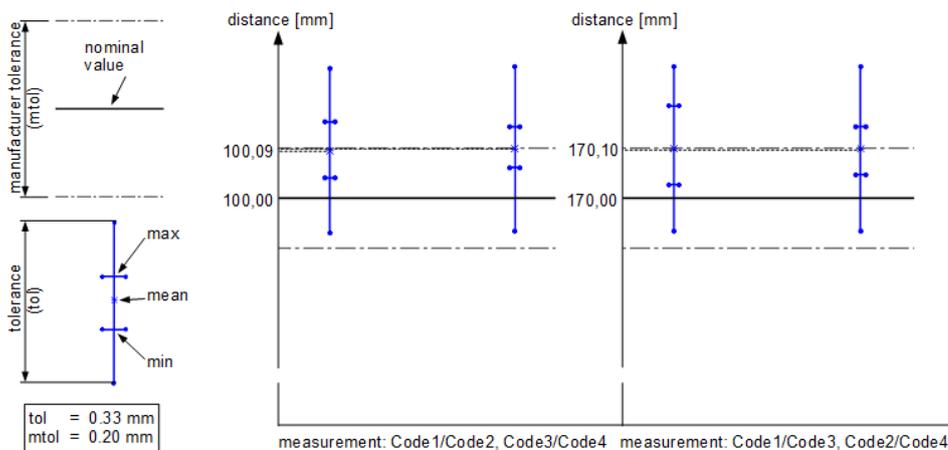


Figure 105: 3D QR Code – distance evaluation of measurement plate

All of the measured mean distance values are almost outside the given manufacturer tolerance. The value of tol ($\pm 0,33$ mm) is equal to the determined distance measure-

ment tolerance defined in section 5.5.2. As in the previous section, I consider a best and worst case measurement scenario for the definition of the absolute error. The calculations are presented in Equation (116) and Equation (117).

$$\text{err}_w = \left(\max(|\text{actualValue} - \text{mean}|) + \frac{\text{mtol}}{2} + \frac{\text{tol}}{2} \right) \cdot \frac{1}{2}$$

$$\text{err}_w = \left(|170 - 170,10| + \frac{0,2}{2} + \frac{0,33}{2} \right) \cdot \frac{1}{2} \approx 0,18 \text{ mm} \quad (116)$$

$$\text{tol}_{XY_w} = \pm \text{err}_w = \pm 0,18 \text{ mm}$$

$$\text{tol}_{XY_b} = \pm \left(\frac{\text{tol}}{2} \right) \cdot \frac{1}{2} = \pm \left(\frac{0,33}{2} \right) \cdot \frac{1}{2} \approx \pm 0,08 \text{ mm} \quad (117)$$

The actual tolerance of the binocular 3D QR Code marker distance measurements in a plane is somewhere between tol_{XY_b} and tol_{XY_w} .

The deviations of tol_Z , tol_α and tol_β were then formed by putting the distance measurements in relation to an optimally fitted plane. The results are illustrated in Figure 106 and Figure 107.

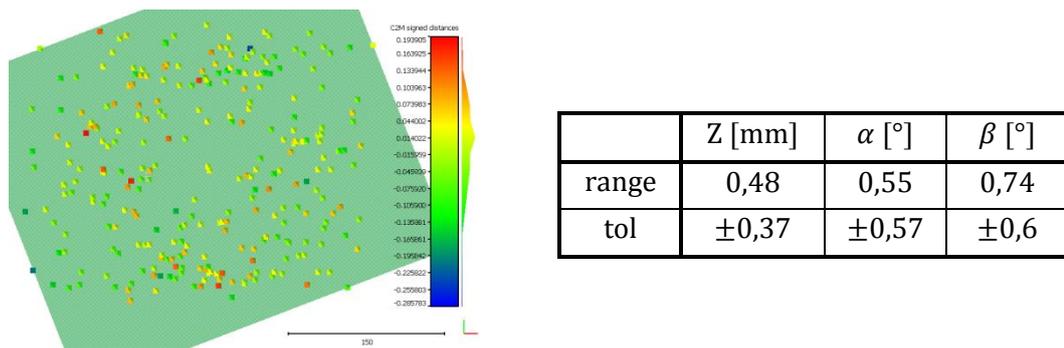


Figure 106: 3D QRCM – Z deviation of 70 different object positions in a plane and resulting tolerance of Z, α and β

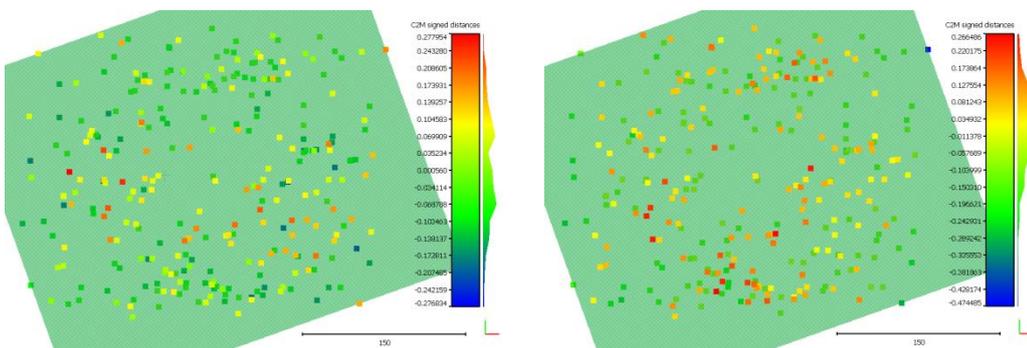


Figure 107: 3D QRCM – α and β deviation of 70 different object positions in a plane

The Z, α and β deviations are very alike to the results of section 5.5.2. The values of the points close to the image center differ from the measurements at the border region of the camera's field of view.

The overall worst case 3D position accuracy of the binocular detection of the 3D QR Code marker is defined by Equation (118).

$$\text{tol}_{\text{QRCM}_w} = \pm\sqrt{\text{tol}_{XY_w}^2 + \text{tol}_Z^2} = \pm\sqrt{0,18^2 + 0,37^2} \approx \pm 0,41 \text{ mm} \quad (118)$$

The best case detection error is shown in Equation (119).

$$\text{tol}_{\text{QRCM}_b} = \pm\sqrt{\text{tol}_{XY_b}^2 + \text{tol}_Z^2} = \pm\sqrt{0,08^2 + 0,37^2} \approx \pm 0,38 \text{ mm} \quad (119)$$

The actual error of the binocular 3D QR Code marker detection is almost equal to the accuracy of the 3D ARTM-B and lies somewhere between $\text{tol}_{\text{QRCM}_b} = \pm 0,38 \text{ mm}$ and $\text{tol}_{\text{QRCM}_w} = \pm 0,41 \text{ mm}$.

5.5.4. 3D Fiducial Propeller Marker (3D FPM)

As a last point, the accuracy of the 3D Fiducial Propeller marker is evaluated. The measurement plate with the four 3D markers with the identification 5, 6, 7 and 8 that was used for the evaluation, is visualized in Figure 108.

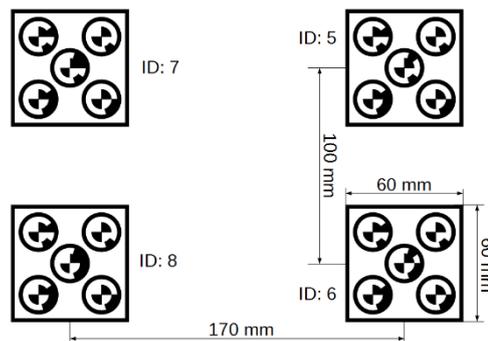


Figure 108: marker detection (fiducial marker) – definition of measurement plate

The plate was placed in the field of view of the camera to perform 70 static measurements of each marker. The repeatability of these measurements is shown in Table 20.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
range_{rep}	0,01	0,02	0,06	0,16	0,20	0,06
tol_{rep}	$\pm 0,015$	$\pm 0,03$	$\pm 0,06$	$\pm 0,13$	$\pm 0,1$	$\pm 0,04$

Table 20: 3D FPM – tolerance of static marker detection repeatability for 70 measurements

Following the previous marker detection experiments, the marker plate was then randomly placed and detected at 70 different positions in a plane. The gathered distance measurements between the markers on the plate are compared with the nominal distance and the manufacturer tolerance of $\pm 0,10 \text{ mm}$ (Figure 109).

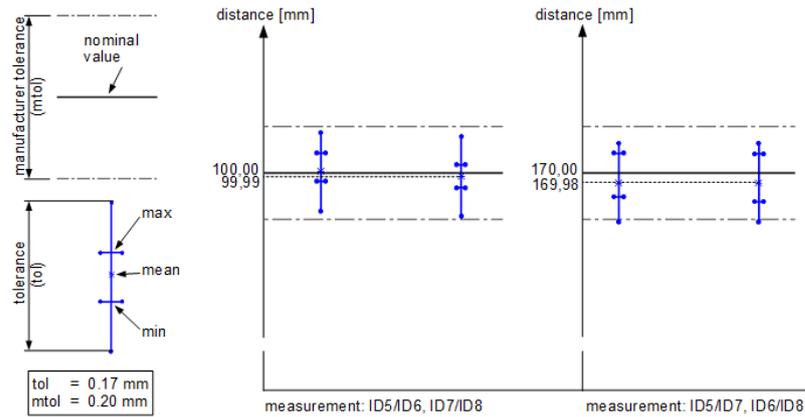


Figure 109: 3D FPM – distance evaluation of measurement plate

The measured mean distance values are almost equal to the given nominal values. Due to the very low tolerance tol ($\pm 0,17$ mm) of the measurements, all conducted distance measurements and all possible measurements (incl. tolerance) are within the manufacturer tolerance. The calculation of the measurement tolerance in a plane tol_{XY} is performed (for more details see section 5.5.1). The best and worst case measurement scenario of the absolute error are presented in Equation (120) and Equation (121).

$$err_w = \left(\max(|actualValue - mean|) + \frac{mtol}{2} + \frac{tol}{2} \right) \cdot \frac{1}{2} \quad (120)$$

$$err_w = \left(|170 - 169,98| + \frac{0,2}{2} + \frac{0,17}{2} \right) \cdot \frac{1}{2} \approx 0,10 \text{ mm}$$

$$tol_{XY_w} = \pm err_w = \pm 0,10 \text{ mm}$$

$$tol_{XY_b} = \pm \left(\frac{tol}{2} \right) \cdot \frac{1}{2} = \pm \left(\frac{0,17}{2} \right) \cdot \frac{1}{2} \approx \pm 0,04 \text{ mm} \quad (121)$$

The actual tolerance of the binocular 3D Fiducial Propeller marker distance measurements in a plane is somewhere between tol_{XY_b} and tol_{XY_w} .

Finally, the tolerances of Z , α and β were determined as explained in section 5.5.1 and visualized in Figure 110 and Figure 111.

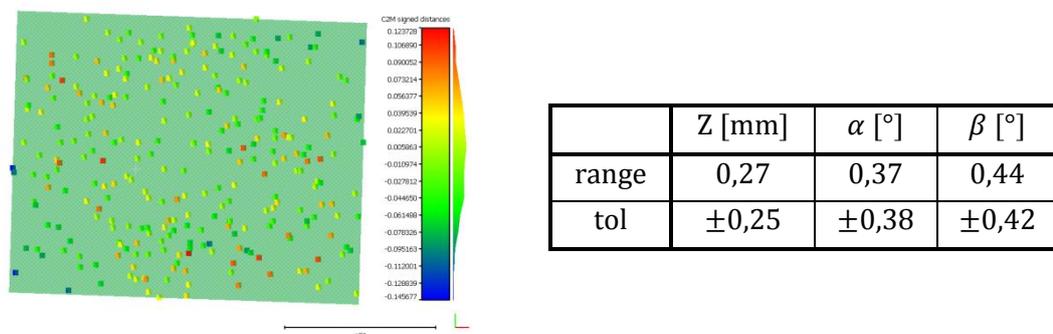


Figure 110: 3D FPM – Z deviation of 70 different object positions in a plane and resulting tolerance of Z, α and β

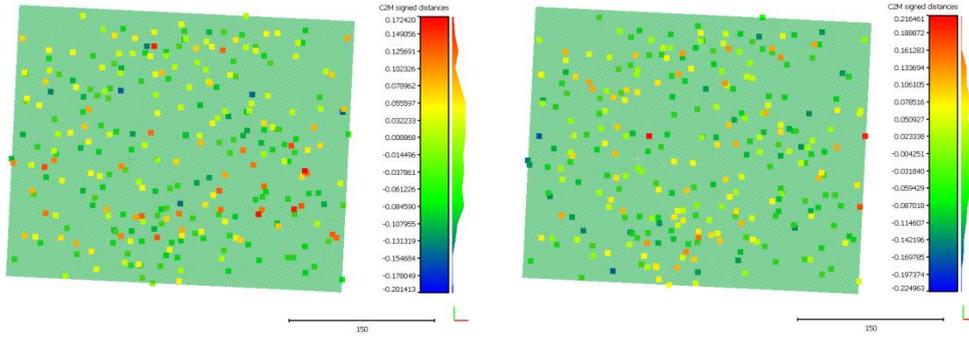


Figure 111: 3D FPM – α and β deviation of 70 different object positions in a plane

Figure 110 shows that the Z values are not randomly distributed. Despite the low deviation, the pattern as given for each of the 3D marker detection methods is still visible. The points close to the sensor border seem to be further away from the camera than the points close to the center of the camera. The color representation of the deviations in Figure 111, on the other hand, shows that the distribution of α and β now has no visible patterns.

The overall worst case 3D position accuracy of the binocular 3D Fiducial Propeller marker detection is defined by Equation (122).

$$\text{tol}_{\text{FPM}_w} = \pm \sqrt{\text{tol}_{XY_w}^2 + \text{tol}_Z^2} \pm \sqrt{0,10^2 + 0,25^2} \approx \pm 0,27 \text{ mm} \quad (122)$$

The best case detection error is described by Equation (123).

$$\text{tol}_{\text{FPM}_b} = \pm \sqrt{\text{tol}_{XY_b}^2 + \text{tol}_Z^2} = \pm \sqrt{0,04^2 + 0,25^2} \approx \pm 0,25 \text{ mm} \quad (123)$$

The actual error of the binocular 3D Fiducial Propeller marker detection is somewhere between $\text{tol}_{\text{FPM}_b} = \pm 0,25 \text{ mm}$ and $\text{tol}_{\text{FPM}_w} = \pm 0,27 \text{ mm}$.

5.5.5. Fiducial Marker – Detection-Height Variations

This section focuses on the testing of the accuracy of the chosen Fiducial Propeller marker at different detection heights. The marker is placed in a distance of 300 mm, 450 mm and 600 mm from the camera (shown in Figure 112).

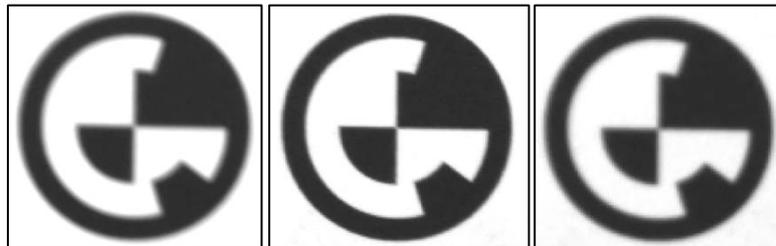


Figure 112: 3D FPM – Acquisition of the Fiducial Propeller marker at different heights (left: 300 mm, middle: 450 mm, right: 600 mm)

Table 21 shows the results of 70 static marker measurements at different heights.

Height	tol_{rep_X}	tol_{rep_Y}	tol_{rep_Z}	$tol_{rep_α}$	$tol_{rep_β}$	$tol_{rep_γ}$
300 mm	$\pm 0,013$ mm	$\pm 0,01$ mm	$\pm 0,03$ mm	$\pm 0,12^\circ$	$\pm 0,13^\circ$	$\pm 0,05^\circ$
450 mm	$\pm 0,015$ mm	$\pm 0,03$ mm	$\pm 0,03$ mm	$\pm 0,13^\circ$	$\pm 0,1^\circ$	$\pm 0,04^\circ$
600 mm	$\pm 0,015$ mm	$\pm 0,02$ mm	$\pm 0,06$ mm	$\pm 0,16^\circ$	$\pm 0,19^\circ$	$\pm 0,05^\circ$

Table 21: 3D FPM – tolerance of static marker detection repeatability for 70 measurements

Furthermore, the marker plate was randomly placed and detected at 70 different positions in a plane. At a height of 300 mm, distance measurements between ID5/ID6, ID7/ID8, ID5/ID7 and ID6/ID8 on the Fiducial Propeller measurement plate (Figure 108) were possible only close to the sensor border. Therefore, the accuracy evaluation of the measurements at different heights is performed with respect to the Z, α and β deviation. Figure 113 shows the Z deviation of the measurements at a height of 300 mm and 600 mm. The results for the height of 450 mm are visualized in section 5.5.4.

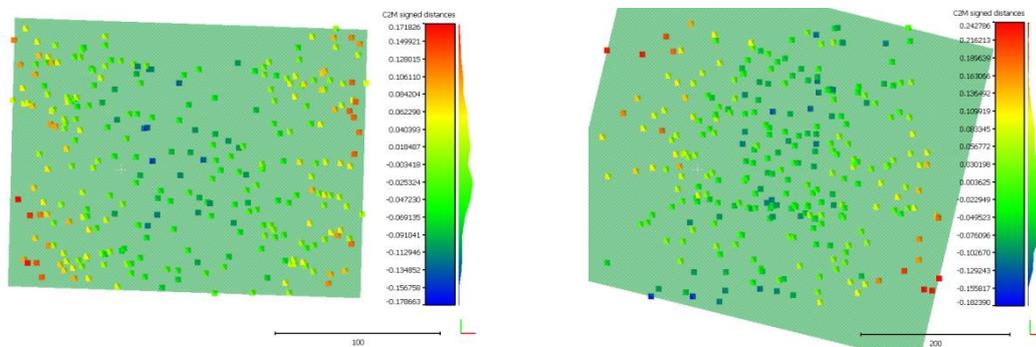


Figure 113: 3D FPM – Z deviation of 70 randomly distributed 3D FPM position measurements at a height of 300 mm (left) and 600 mm (right)

The Z deviations of the measurements illustrated in Figure 110 and Figure 113 display a pattern, although the pattern of Figure 110 is not as significant as the one in Figure 113. In all three measurements, the measurement points close to the middle of the images have different heights than those at the border region. The deviations of the α and β values (Figure 111, Figure 114 and Figure 115) show a random distribution.

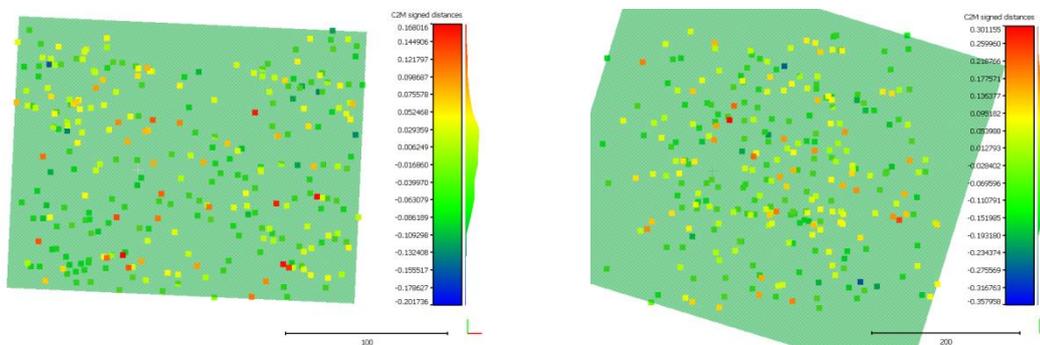


Figure 114: 3D FPM – α deviation of 70 randomly distributed 3D FPM position measurements at a height of 300 mm (left) and 600 mm (right)

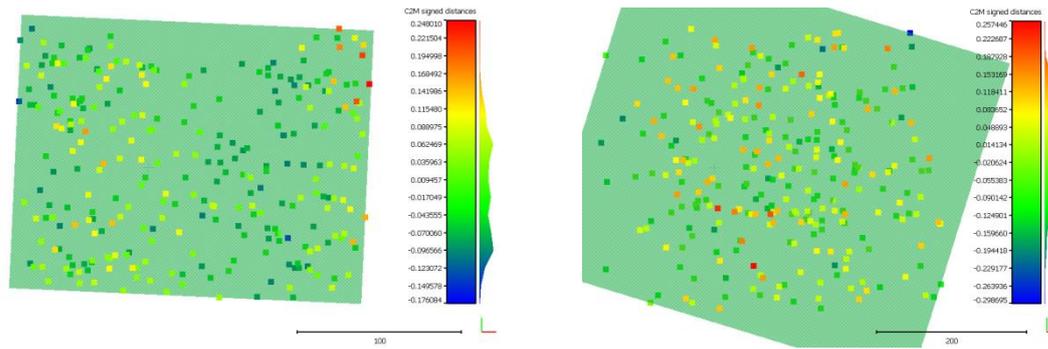


Figure 115: 3D FPM – β deviation of 70 randomly distributed 3D FPM position measurements at a height of 300 mm (left) and 600 mm (right)

The resulting tolerance and measurement range of the conducted measurements shown above are listed in Table 22 and Table 23.

Height [mm]	tol _Z [mm]	tol _α [°]	tol _β [°]
300	±0,34	±0,43	±0,42
450	±0,25	±0,38	±0,42
600	±0,41	±0,51	±0,53

Table 22: 3D FPM – resulting tolerance of Z, α and β for 70 randomly distributed 3D FPM position measurements in a plane at different heights

Height [mm]	tol _Z [mm]	tol _α [°]	tol _β [°]
300	0,35	0,37	0,44
450	0,27	0,37	0,44
600	0,43	0,66	0,56

Table 23: 3D FPM – resulting range of Z, α and β for 70 randomly distributed 3D FPM position measurements in a plane at different heights

Using the resulting measurement range of Z given in Table 23 and the formulation of Equation (42), one can determine the subpixel accuracy of the marker detection at each height (Table 24).

$$\Delta z \approx \frac{z^2}{Bf} \Delta d = \frac{z^2}{Bf} \text{CellSize} \cdot 2 \cdot \text{subPix}$$

$$\text{subPix} = \frac{\Delta z B f}{2 \cdot \text{CellSize} \cdot z^2}$$

Height [mm]	tol _Z [mm]	subPix [Pixels]
300	0,35	0,55
450	0,27	0,19
600	0,43	0,17

Table 24: 3D FPM – Subpixel accuracy for 3D FPM detections at different heights

5.6. Industrial Marker to Object Map

The object detection algorithm SBM-ECT (explained in section 4.3.3 and evaluated in section 5.4.3) and the 3D Fiducial Propeller marker (explained in section 4.4.4 and evaluated in section 5.5.4) are utilized in this segment to create the industrial marker to object map described in section 4.2. The main goal of this section is to create a relation between the 3D markers and the object, which can be used to reduce the tolerance of the 3D object detections that were evaluated in section 5.4.

Four 3D Fiducial Propeller markers with the ID 5, 6, 7 and 8 were placed and fixated near the object (see Figure 116). The object is mounted on the plate with dowel pins and the relation between the markers and the object stays constant. If the relation changes in case of a marker or object displacement, the industrial marker to object map has to be recreated. This marker to object map is denoted as M2O-Map.

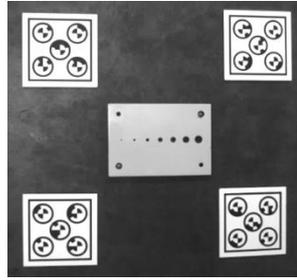


Figure 116: M2O-Map – The pose of the object in the middle is defined by the arrangement of four 3D Fiducial Propeller markers

The relationship between the markers and the object was built from the detections of 10 different camera poses. At each pose, 35 marker detections and 35 object detections were performed. The mean of the 35 marker detections is denoted as the transformation ${}^cT_{m_{ij}}$ of the marker coordinate system MCS_i $\{i \in \mathbb{N} \mid 5 \leq i \leq 8\}$ to the camera coordinate system CCS_j $\{j \in \mathbb{N} \mid 0 < j \leq 10\}$. The mean of the 35 object poses is represented by the transformation ${}^{m_i}T_o$ of the object coordinate system OCS to the marker coordinate system MCS_i . The transformation ${}^{m_{ij}}T_o$ can be determined by using Equation (51). In the next step, the detections of the 10 camera positions are combined to build a robust transformation ${}^{m_i}T_o$ by applying Equation (52).

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
ID5 \rightarrow ${}^{m_5}T_o$	94,22	62,44	-3,52	0,16	-0,05	90,38
ID6 \rightarrow ${}^{m_6}T_o$	56,08	84,23	-3,55	-0,09	-0,17	0,26
ID7 \rightarrow ${}^{m_7}T_o$	87,94	132,84	-3,64	-0,26	0,09	269,22
ID8 \rightarrow ${}^{m_8}T_o$	132,48	86,08	-3,68	0,04	0,28	179,85

Figure 117: M2O-Map – Transformations ${}^{m_i}T_o$ from OCS to MCS_i

The defined relation ${}^{m_i}T_o$ is then used in combination with the marker detection ${}^cT_{m_i}$ to determine the object pose cT_o . Each combination of ${}^cT_o = {}^cT_{m_i} {}^{m_i}T_o$ for all MCS_i $\{i \in$

$\mathbb{N} \mid 5 \leq i \leq 8$ has to deliver the same result. Otherwise, the relations between the markers and the object are not correctly defined. The results of 70 static object measurements for each marker by using the M2O-Map are illustrated in Table 25.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
ID5 \rightarrow ${}^cT_{m_5}^{m_5}T_o$	52,72	19,22	454,19	1,25	352,69	181,16
ID6 \rightarrow ${}^cT_{m_6}^{m_6}T_o$	52,73	19,22	454,28	1,24	352,71	181,13
ID7 \rightarrow ${}^cT_{m_7}^{m_7}T_o$	52,74	19,21	454,41	1,23	352,65	181,14
ID8 \rightarrow ${}^cT_{m_8}^{m_8}T_o$	52,76	19,23	454,29	1,26	352,68	181,14

Table 25: M2O-Map – Transformations from *OCS* to *CCS* for each marker

The results are very similar to each other. The main difference occurs in Z direction with a maximum difference of 0,22 mm. The combination of the four poses results in the mean pose shown in Table 26 (calculation presented in Equation (53)). The mean target pose cT_o defined by the M2O-Map is only slightly different from the mean object pose measurement conducted with the SBM-ECT method.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
cT_o	52,74	19,22	454,29	1,25	352,68	181,14
SBM – ECT	52,77	19,00	454,53	0,95	352,46	181,14

Table 26: M2O-Map – Mean transformation cT_o of all markers

The repeatability of the mean pose for 70 static detections is depicted in Table 27.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
$range_{rep}$	0,02	0,02	0,11	0,04	0,04	0,01
tol_{rep}	$\pm 0,02$	$\pm 0,02$	$\pm 0,11$	$\pm 0,05$	$\pm 0,04$	$\pm 0,01$

Table 27: M2O-Map – Repeatability of 70 detections of the transformation from *OCS* to *CCS*

The projection of the determined mean object pose to the corresponding image delivers a good result. Figure 118 shows the mean object pose at different camera poses.

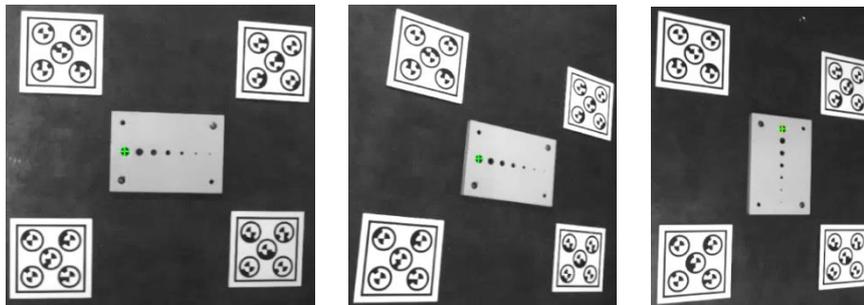


Figure 118: M2O-Map – Projected 3D position of three object detections via the M2O-Map at different camera positions

In addition to the M2O-Map as just mentioned, I introduce another marker to object map which is purely built with 3D Fiducial Propeller markers (Figure 119). The markers with ID 10, 11, 12, 14 were used to describe the pose of the marker with ID 13, which is considered to be the object. This map is denoted as M2M-Map.

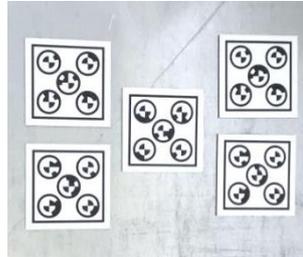


Figure 119: M2M-Map – This map is purely built with 3D Fiducial Propeller markers. The marker in the middle with ID 13 is considered to be the object.

The results of 70 static object measurements for each marker by using the M2M-Map are shown in Table 28.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
ID10 \rightarrow ${}^cT_{m_{10}}{}^{m_{10}}T_o$	11,52	26,80	455,07,	1,18	352,45	181,62
ID11 \rightarrow ${}^cT_{m_{11}}{}^{m_{11}}T_o$	11,53	26,82	455,10	1,18	352,37	181,63
ID12 \rightarrow ${}^cT_{m_{12}}{}^{m_{12}}T_o$	11,55	26,80	455,11	1,18	352,42	181,64
ID14 \rightarrow ${}^cT_{m_{14}}{}^{m_{14}}T_o$	11,54	26,82	455,03	1,14	352,47	181,63

Table 28: M2M-Map – Transformations from *OCS* to *CCS* for each marker

The results of the M2O-Map and the M2M-Map are very similar to each other. The main deviation of the M2M-Map occurs in Z direction with a maximum difference of 0,08 mm. The combination of the four poses results in the mean pose shown in Table 29 (for calculation see Equation (53)). The mean target pose cT_o defined by the M2M-Map highly corresponds with the mean target pose measurement ${}^cT_{m_{13}}$.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
cT_o	11,53	26,81	455,08	1,17	352,44	181,63
${}^cT_{m_{13}}$	11,52	26,81	455,16	1,18	352,44	181,65

Table 29: M2M-Map – Mean transformation cT_o of all markers and mean measurement of ${}^cT_{m_{13}}$

The repeatability of the mean pose for 70 static detections is depicted in Table 30.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
$range_{rep}$	0,01	0,01	0,05	0,03	0,04	0,01
tol_{rep}	$\pm 0,01$	$\pm 0,01$	$\pm 0,05$	$\pm 0,04$	$\pm 0,04$	$\pm 0,01$

Table 30: M2M-Map – Repeatability of 70 detections of the transformation from *OCS* to *CCS*

5.7. Position-Based Cartesian Robot Control

The stereo system is mounted on the end-effector of the Universal Robot UR10 as shown in Figure 71 to guide the robot to the desired target pose. The main focus is now to determine the behavior of the robot when using visual servoing to approach a target pose.

Due to the fact that Universal Robot does not give any information about the robot's absolute accuracy, I first concentrate on the determination of a rough approximation of the robot's absolute precision. According to the manufacturer, the robot is only suited for operations like teach-in tasks that solely require a good repeatability. Afterwards, I present a number of results to the Hand-Eye calibration explained in section 4.5.1, to determine the transformation between the end-effector of the robot and the camera coordinate system. Further on, I perform measurements as the robot approaches the target pose, which is detected via the industrial marker to object map.

5.7.1. Universal Robot UR10 – Accuracy Measurements

To determine the absolute accuracy of the UR10, I use the UR10 calibration plate visualized in Figure 120 [47]. Each position of the plate is afflicted with a tolerance of $\pm 0,1$ mm. Each hole on the calibration plate is perfectly sized to fit the end-effector of the UR10. Due to a manufacturer error, the red colored hole shown in Figure 120 was too small to fit the end-effector and was therefore excluded from the measurements.

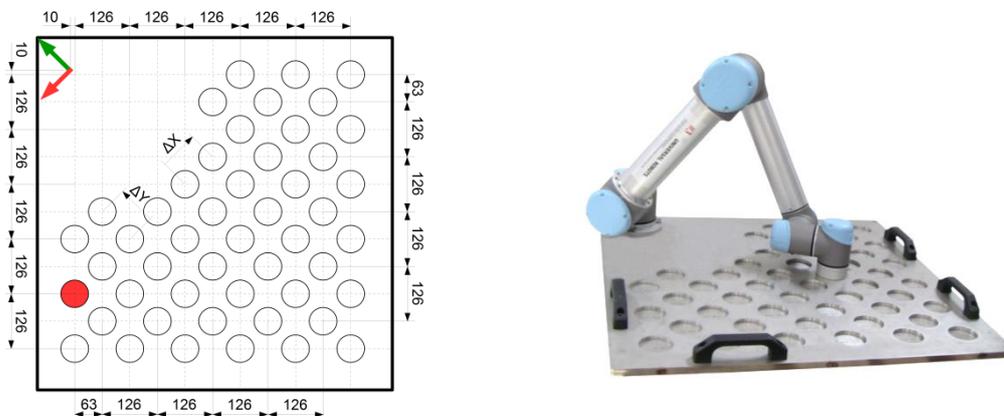


Figure 120: visualization of the UR-10 calibration plate dimensions (adapted from [47])

The end-effector of the robot was put into every hole of the calibration plate. The resulting robot poses which were returned by the robot control were saved and used to evaluate the overall position accuracy. The following accuracies give only a coarse overview about the overall precision of the robot and are not comparable to the actual absolute accuracy of the robot.

Initially, the distances ΔX and ΔY between all holes were evaluated. The nominal distance is 89,095 mm. The mean and min/max values of the actual distances are shown in Table 31.

	X – Distance Measurements	Y – Distance Measurements
Mean	89,125 mm	89,033 mm
Std. Dev.	0,25 mm	0,15 mm
Min	88,523 mm	88,705 mm
Max	89,630 mm	89,303 mm
Range	1,107 mm	0,598 mm

Table 31: X and Y distance measurements between two neighboring calibration plate positions

The ΔX and ΔY measurements of Table 31 are visualized in Figure 121. The color represents the height of the distance measurement, whereas the brightness of the color is proportional to the established distance value of ΔX and ΔY . A high value of ΔX and ΔY is equal to a bright color point. The colors were scaled between the minimum and maximum measurement shown in Table 31. The distance ΔX seems to increase slightly with growing X value, while the ΔY appears to decrease with a growing X value.

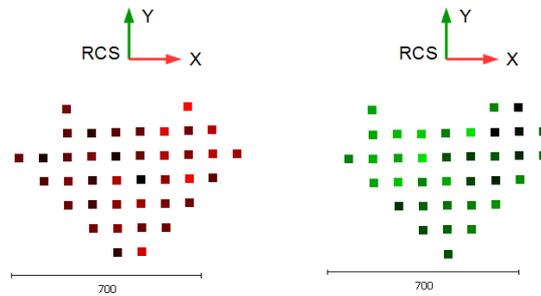


Figure 121: ΔX (left) and ΔY (left) distance measurement deviations between two neighboring calibration plate positions

In the next step, the actual distance from each hole to the robot base is taken and compared with the known nominal distance. The results are shown in Table 32.

	Mean [mm]	Min [mm]	Max [mm]	Range [mm]
Distance – Deviation	0,053	-0,492	0,318	0,810

Table 32: Distance deviation of nominal and actual distances from a defined position on the calibration plate to the origin of the robot

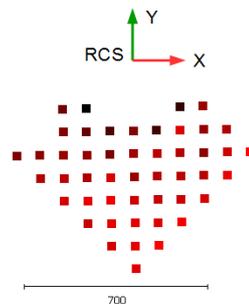


Figure 122: Distance deviation of nominal and actual distance from a defined position on the calibration plate to the origin of the robot

Figure 122 visualizes the results of Table 32. The brightness of the color values is proportional to the distance measurement. A high value of the distance is equal to a bright color point. The colors were scaled between the minimum and maximum value.

After that, the orientation of the robot was evaluated by moving the end-effector of the robot to 50 randomly chosen positions in a plane. The values of RX, RY and RZ should stay constant for each position. The results are shown in Table 33.

	<i>RX</i>	<i>RY</i>	<i>RZ</i>
Std. Dev [°]	0,18	0,37	0,29
Min [°]	-0,22	178,89	-1,94
Max [°]	0,58	180,34	-0,94
Range [°]	0,8	1,45	1

Table 33: RX, RY and RZ deviation of the robot pose randomly moved in a plane

The measurements of Table 33 are visualized in Figure 123. The brightness of the color values is proportional to the value of the angle. A high angle is equal to a bright color point. The colors were scaled between the minimum and maximum value of Table 33.

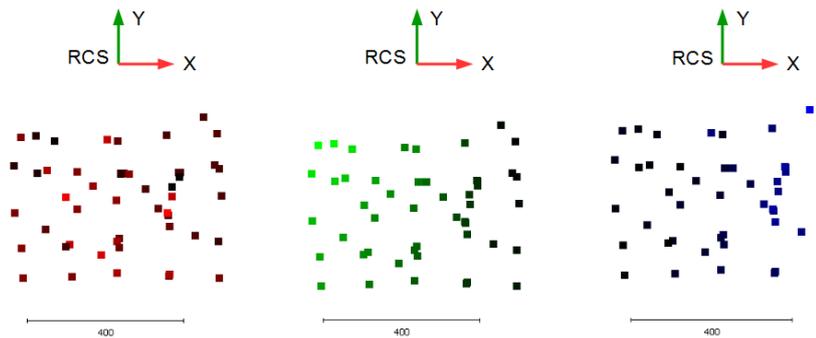


Figure 123: RX (left), RY (middle) and RZ (right) deviation of the robot pose of 50 randomly chosen positions in a plane

The RY and RZ values seem to change along the X values, whereas the RX values seem to change randomly. Therefore, the RY and RZ values were sorted according to the X axis and observed in detail in the diagrams shown in Figure 124.

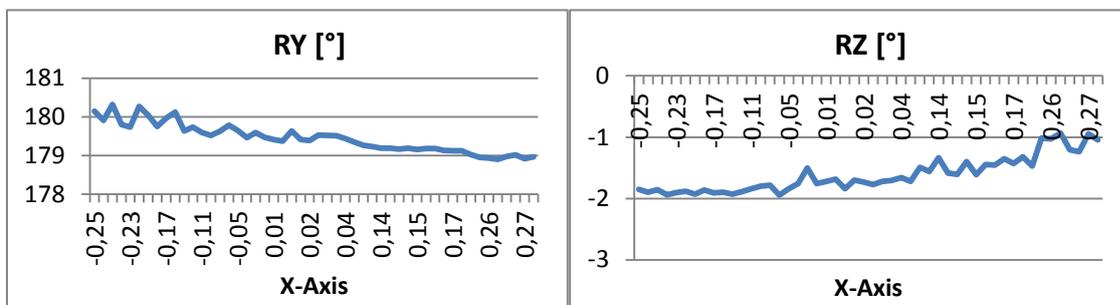


Figure 124: RY and RZ deviation values sorted according to the X positions of the robot which was randomly moved in a plane

5.7.2. Hand-Eye Calibration

The Hand-Eye calibration was performed by using the extended closed-loop Hand-Eye calibration presented in Equation (85) in section 4.5.1. The algorithm is available in the machine vision software HALCON [49]. The calibration procedure was performed with 60 different acquisitions of the HALCON calibration plate (visualized in Figure 72) at 60 different positions of the robot end-effector. The resulting transformation from flange coordinate system to the camera coordinate system cT_f is listed in Table 34.

cT_f	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
Mean	68,69	124,6	-57,47	11,26	352,58	1,87

Table 34: Resulting parameters of cT_f with 60 different Hand-Eye calibration images

The development of the parameters with a growing number of Hand-Eye calibration images is depicted in Figure 125.

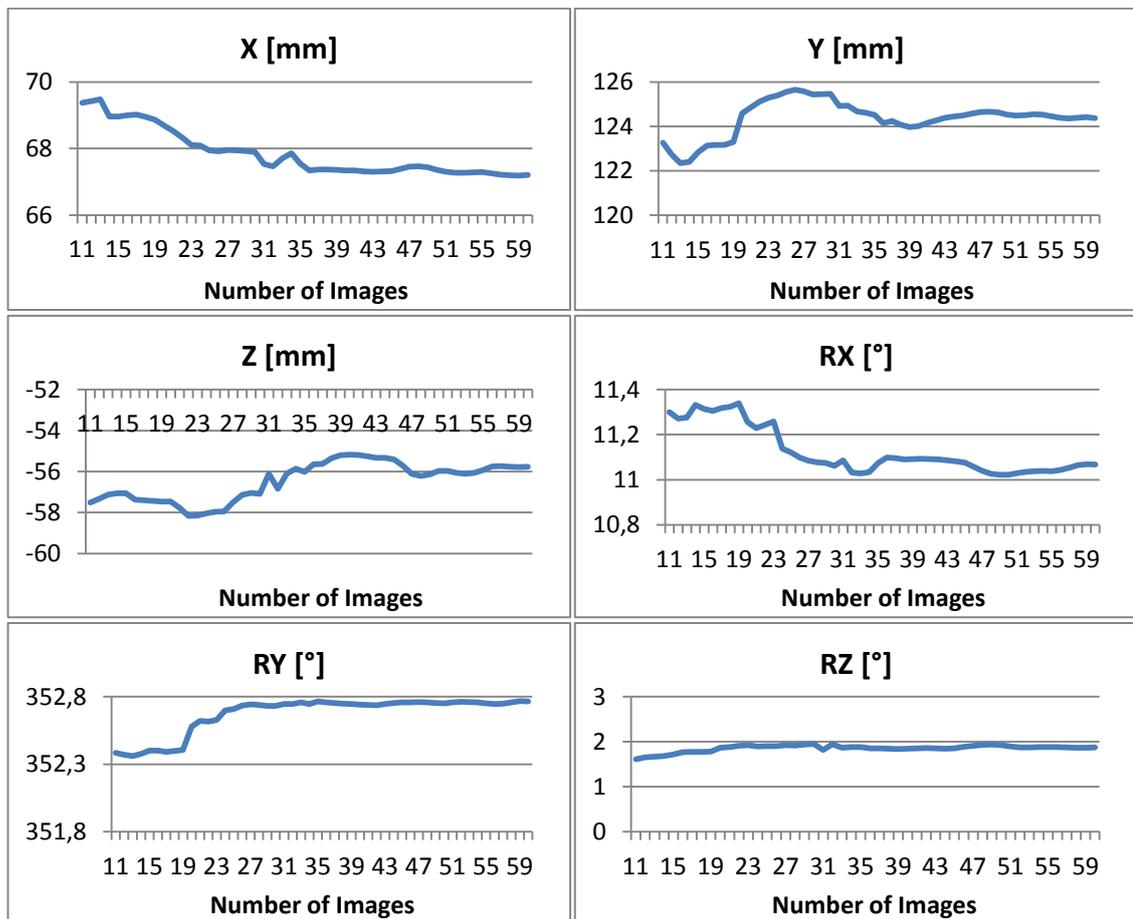


Figure 125: Development of the parameters of cT_f with growing number of Hand-Eye calibration images

Figure 125 shows that the fluctuation of the parameters for the Hand-Eye transformation is much higher than the parameter development of the transformation between the two cameras of the stereo system illustrated in Figure 75.

The accuracy of cT_f was evaluated by performing a chain transformation for each of the 60 calibration poses according to Equation (87) shown in section 4.5.1. The mean and maximum deviations of the transformations from the identity matrix are listed in Table 35.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
Mean	0,91	0,51	1,71	0,14	0,13	0,13
Max	3,03	1,43	4,97	0,55	0,45	0,58

Table 35: Mean and maximal error of a complete chain transformation for each of the 60 different robot positions

5.7.3. Marker Detection Accuracy with respect to the RCS

The previously determined Hand-Eye relation is now used to evaluate the transformation of the camera system (CCS) to the base coordinate system of the robot (RCS). Thus, several detections of the four 3D Fiducial Propeller markers on the measurement plate were conducted at different TCP poses and transformed to the RCS. During the experiment, the pose of the marker measurement plate remained constant.

The robot TCP was moved to 100 different points that span a grid as depicted in Figure 126. The coordinates visualized in Figure 126 represent the position of the FCS at the grid corner points. The orientation was kept constant for all measurement points. A detection of the marker poses and transformation to the RCS was performed at each of the 100 grid positions. Using an optimal kinematic robot model and an optimal Hand-Eye calibration, all of the 100 transformations of the found 3D FPM should result in one 3D pose for each marker in the RCS.

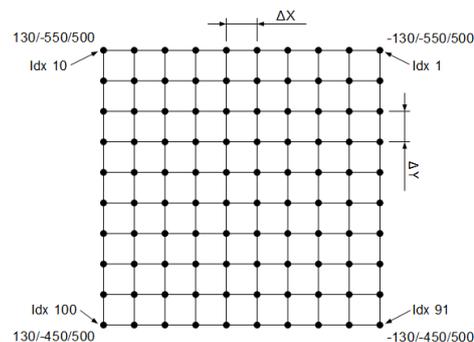


Figure 126: Visualization of the 100 different robot positions at which the static Fiducial Propeller measurement plate was detected

The actual result of the 100 detections and transformations to the RCS are illustrated in Figure 127 (left). The right image of Figure 127 visualizes the resulting 3D positions of the marker with ID5.

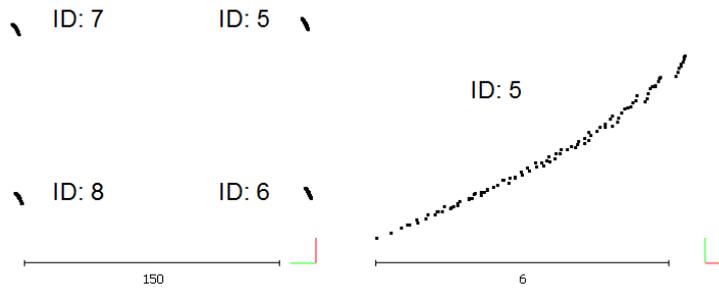


Figure 127: Detection and transformation to the RCS of four 3D FPM at 100 different robot positions (left); detailed visualization of the resulting 3D position of the marker with ID5 (right)

Figure 127 shows a significant change of the marker pose parameters in the RCS as the TCP moves. The development of the pose parameters of the ID5 marker is visualized with respect to the grid position in Figure 128.

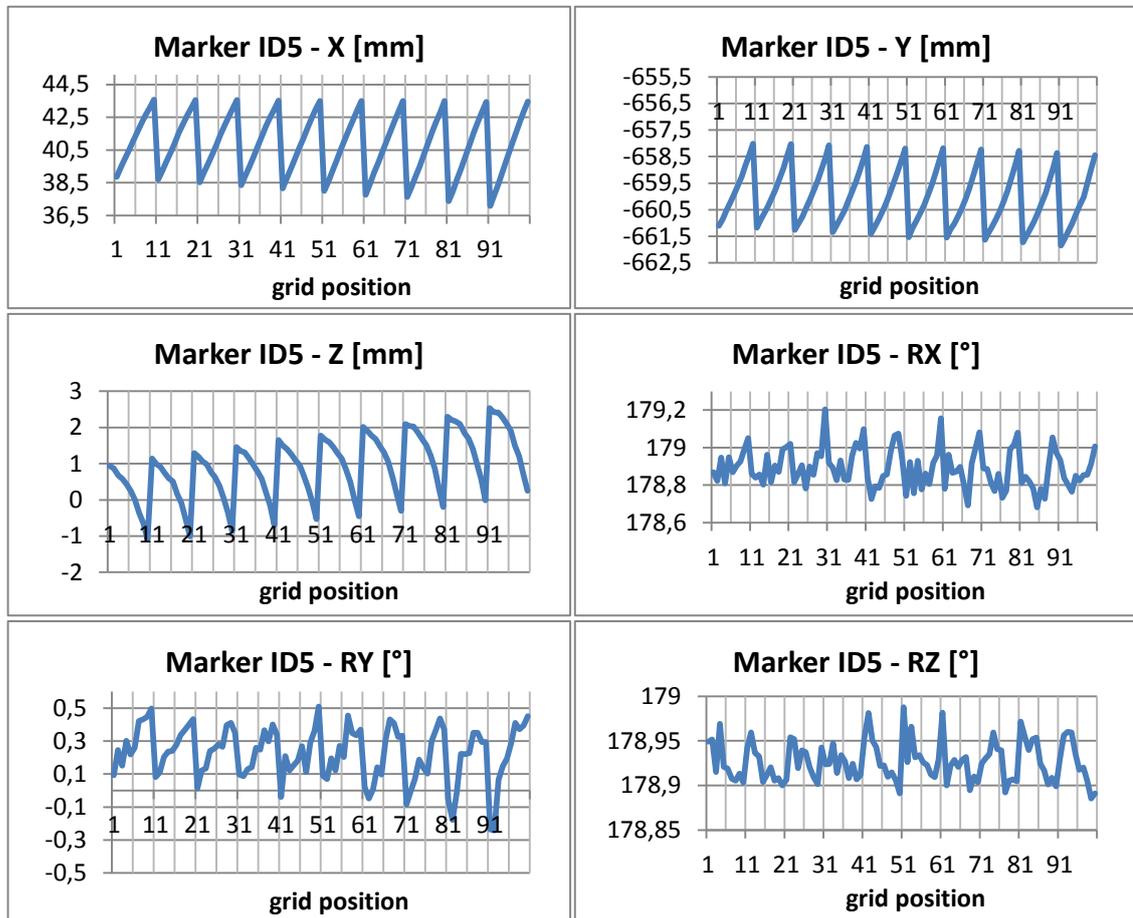


Figure 128: Development of the 3D pose of the marker with ID5 that was transformed to the RCS for each of the 100 robot positions

All observed parameters of the ID5 marker show a significant variance between the detection at the first and at the last grid column pose. The change of X, Y and Z is almost linear. The results of 100 marker pose transformations to the RCS with respect to the ID5 marker are also presented in Table 36.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
mean	40,83	-660,01	0,93	178,89	0,23	178,93
Std. Dev	1,80	1,06	0,86	0,10	0,16	0,02
Min	37,08	-661,85	-1,09	178,68	-0,24	178,89
Max	43,58	-658,02	2,54	179,20	0,51	178,99
R	6,50	3,83	3,63	0,52	0,75	0,10

Table 36: Resulting pose of the 100 transformations of the marker with ID5 to the RCS at different robot positions

The behavior of the markers with ID6, ID7 and ID8 is congruent to the behavior of the marker with ID5.

5.7.4. Position Based Visual Servoing

The results of section 5.7.3 lead to the conclusion that static 3D pose detections in the RCS, conducted with an end-effector mounted camera system, change their pose parameters as the robot moves. Thus, I apply the position based visual servoing algorithm explained in section 4.5.2 to iteratively compensate the change of the target object pose.

In the first experiment, the robot was iteratively controlled to an offset pose with respect to the object defined by the M2O-Map. The remaining offset between the object and the TCP, denoted as ${}^tT_{t_2}$ (visualized in Figure 65), was defined to be 350 mm along the Z-axis of the tool. Figure 129 visualizes some of the conducted experiments. It illustrates the randomly chosen initial start position and the final target offset position of the visual servoing. Furthermore, it displays the image acquisition at the final offset pose with the projection of the TCP (red cross) and the detected object pose (green cross). The projection of the TCP almost coincides with the projection of the detected object pose.

During these experiments I have also assessed an adequate value for the pose offset threshold (POT) defined in section 4.5.2. If the difference between the TCP pose and the detected target pose is smaller than the pre-defined POT, the movement of the robot is stopped and the target is assumed to be reached. The evaluation showed that the robot controller does not accept movement commands that are smaller than the values depicted in Table 37. As a consequence, the smallest possible increments of the robot movement are chosen to define the POT. This threshold is used for all further experiments that consist of position based visual servoing.

	X, Y, Z [mm]	α, β, γ [°]
Threshold	0,15	0,17

Table 37: Defined pose offset threshold (POT) due to minimal robot increments

The POT causes small differences between the TCP and the target object pose, which are illustrated in Figure 129 and Figure 130 (Detail of Projected TCP Pose).

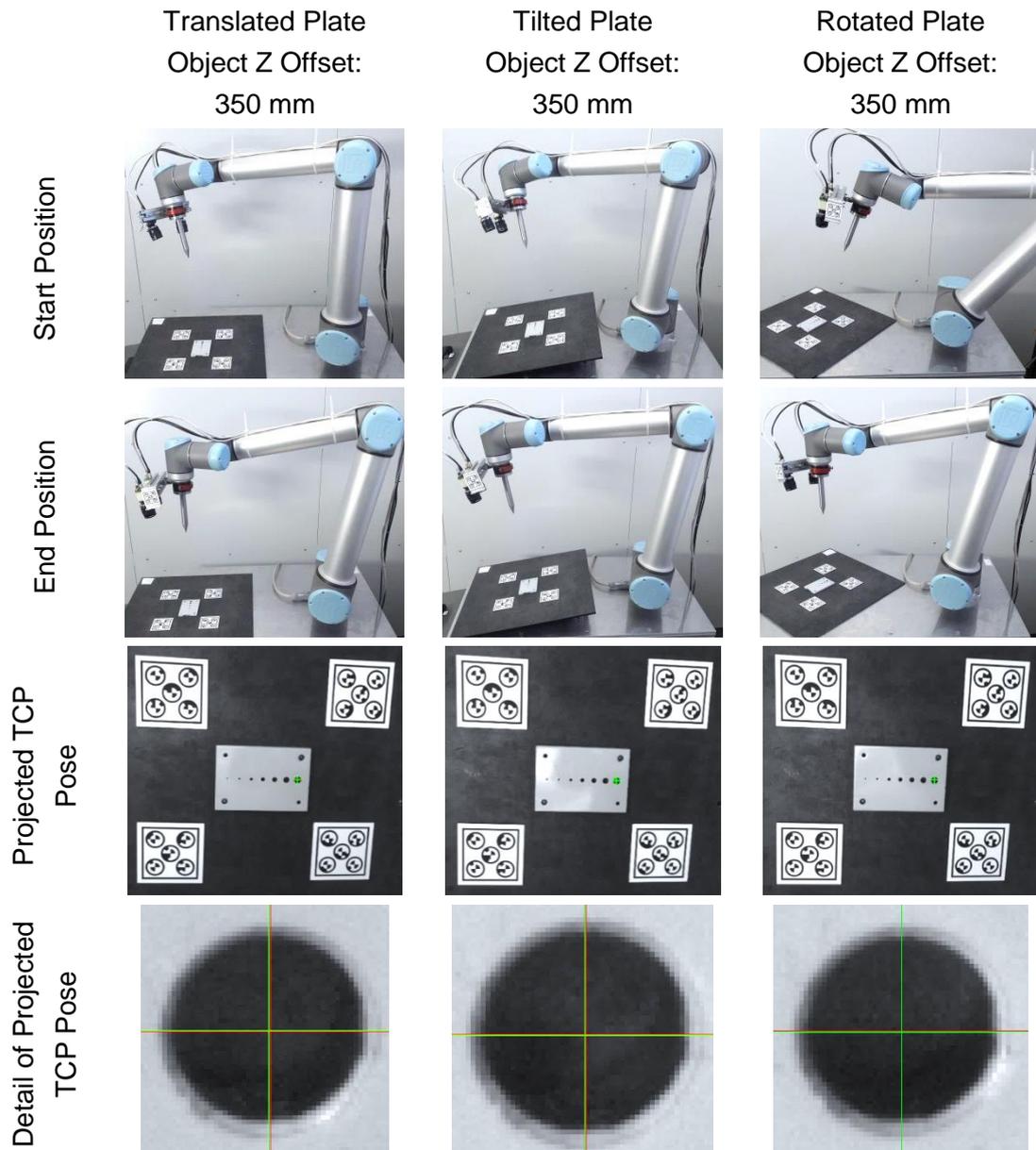


Figure 129: Visualization of the position based visual servoing approach to different translated, rotated and tilted M2O-Map target poses

In the next experiment, I used the M2O-Map and the position based visual servoing to move the robot towards the object along the TCP Z-axis. 15 support points were equally distributed between the start object Z offset of $Z = 350$ mm and the end object Z offset of $Z = 200$ mm. The final target pose of the robot and the projection of the TCP with respect to the actual target pose are illustrated in Figure 130. The actual object target pose is represented by a cross drawn on a label that was affixed to the object. The exact location of the cross was defined by using a caliper rule. This experiment was conducted 10 times to evaluate the repeatability of the algorithm.

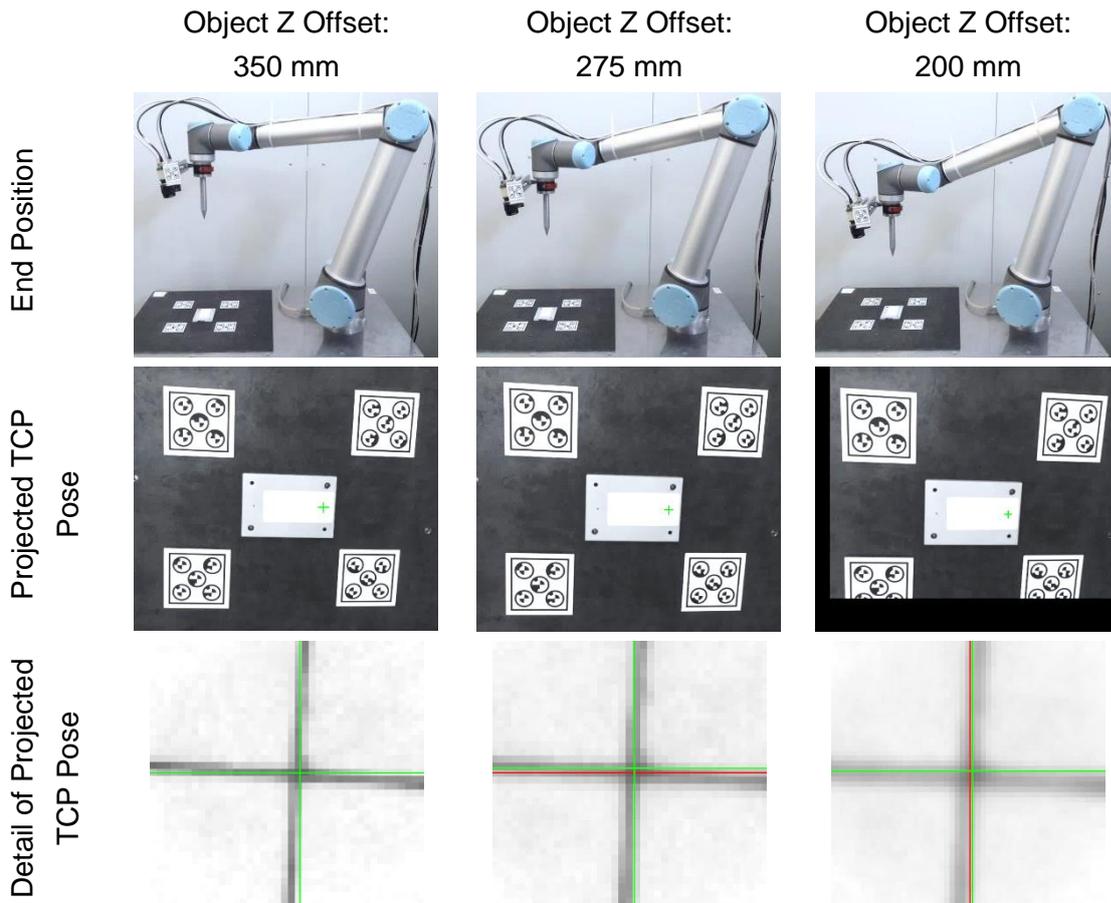


Figure 130: Visualization of the position based visual servoing algorithm when moving towards the target pose defined by the M2O-Map

The difference between the projected TCP and the detected object pose is caused by the previously mentioned POT. Two of the 3D FPM are out of sight at an object Z offset of 200 mm. Thus, the target pose is only built by the mean of two instead of four markers. The 3D object poses detected at each of the 15 support points and transformed to the RCS are visualized in Figure 131.

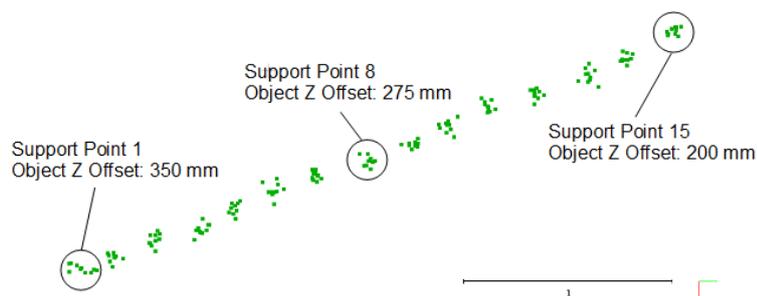


Figure 131: 10 object position detections at 15 offset positions represented in the RCS

The development of the object pose with respect to the change of the object Z offset is depicted in Figure 132.

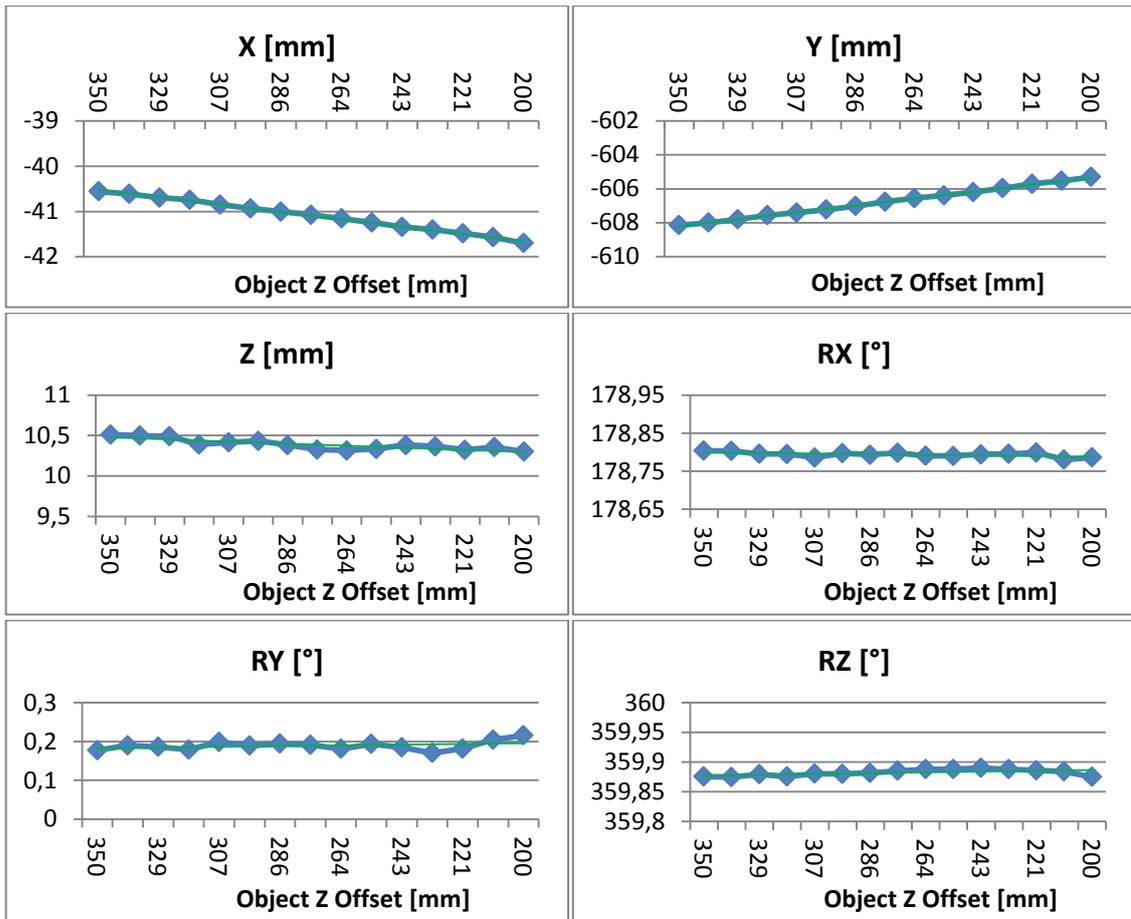


Figure 132: Development of the object pose in the RCS with respect to the object Z offset

All pose parameters show linear behavior with respect to the object Z offset. The orientation starts to fluctuate slightly at a Z offset of about 235 mm (only two markers remain in the camera's field of view). The repeatability of the pose detection in the RCS at an object Z offset of 200 mm is presented in Table 38.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
$range_{rep}$	0,05	0,07	0,10	0,04	0,06	0,02
tol_{rep}	$\pm 0,09$	$\pm 0,11$	$\pm 0,16$	$\pm 0,05$	$\pm 0,09$	$\pm 0,025$

Table 38: RCS object pose repeatability measurement at the object Z offset of 200 mm

The repeatability of the TCP pose in the RCS at an object Z offset 200 mm is shown in Table 39.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
$range_{rep}$	0,18	0,18	0,26	0,14	0,24	0,04
tol_{rep}	$\pm 0,32$	$\pm 0,26$	$\pm 0,36$	$\pm 0,19$	$\pm 0,38$	$\pm 0,06$

Table 39: RCS TCP pose repeatability measurement at the object Z offset of 200 mm

The development of the repeatability at all support points is illustrated in Figure 133. The tolerances of the object detections for all object Z offset poses lie very close to each other. The tendency of the deviations decreases with the Z offset value and is visualized as a line. The closer the robot is placed to the object, the smaller are the measurement deviations.

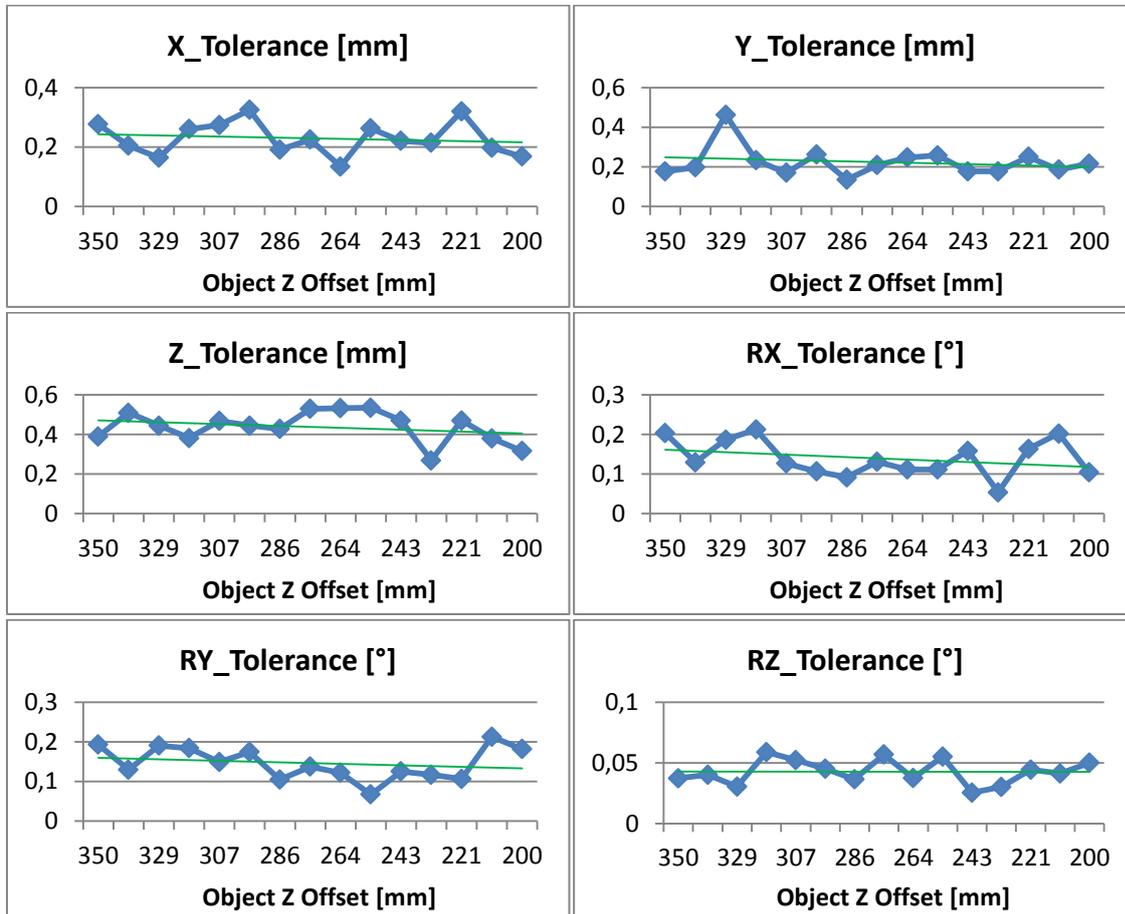


Figure 133: Development of the object pose tolerance in the RCS with respect to the object Z offset

The position based visual servoing along the TCP Z-axis was also tested with the M2M-Map, defined in section 5.6 (Figure 119). This map is purely built with 3D Fiducial Propeller markers and is more accurate than the M2O-Map.

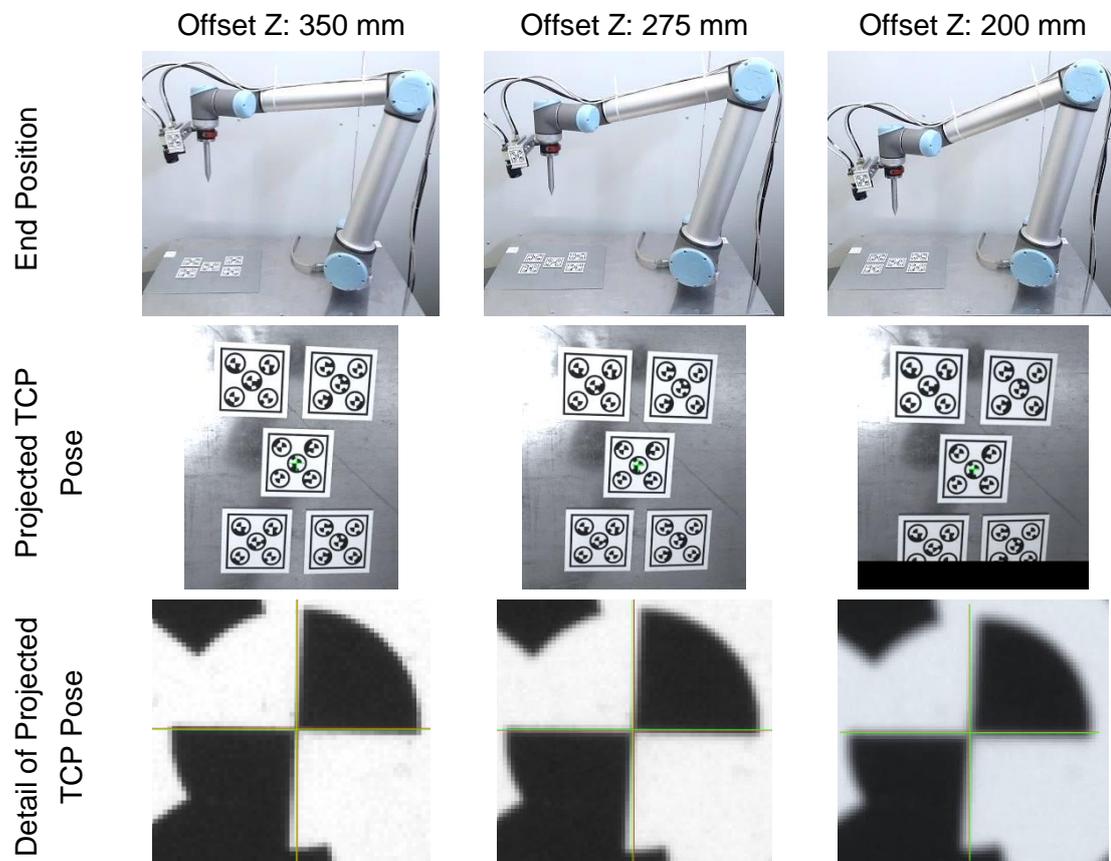


Figure 134: Visualization of the position based visual servoing approach when moving towards the target object pose defined by the M2M-Map

5.8. Extended Visual Servoing – Extrapolation

The previous sections showed that the object pose in the RCS changes linearly as the robot moves towards the target object pose along the TCP Z-axis. Therefore, in this section, I extended the experiments of section 5.7.4 with a linear extrapolation. The extrapolation was based on the object pose detections (M2O-Map), with respect to the RCS, at the uniformly distributed support points. Details can be found in section 4.5.3. In the first experiment, 60 support points were used to gather extrapolation data between a start object Z offset of 450 mm and an end object Z offset of 150 mm. An object Z offset of 450 mm was equal to a camera to object distance of about 580 mm. An object Z offset of 150 mm was equal to a camera to object distance of about 280 mm. The results of the extrapolation are shown in Figure 135. The final point (right side) of each diagram represents the extrapolated value.

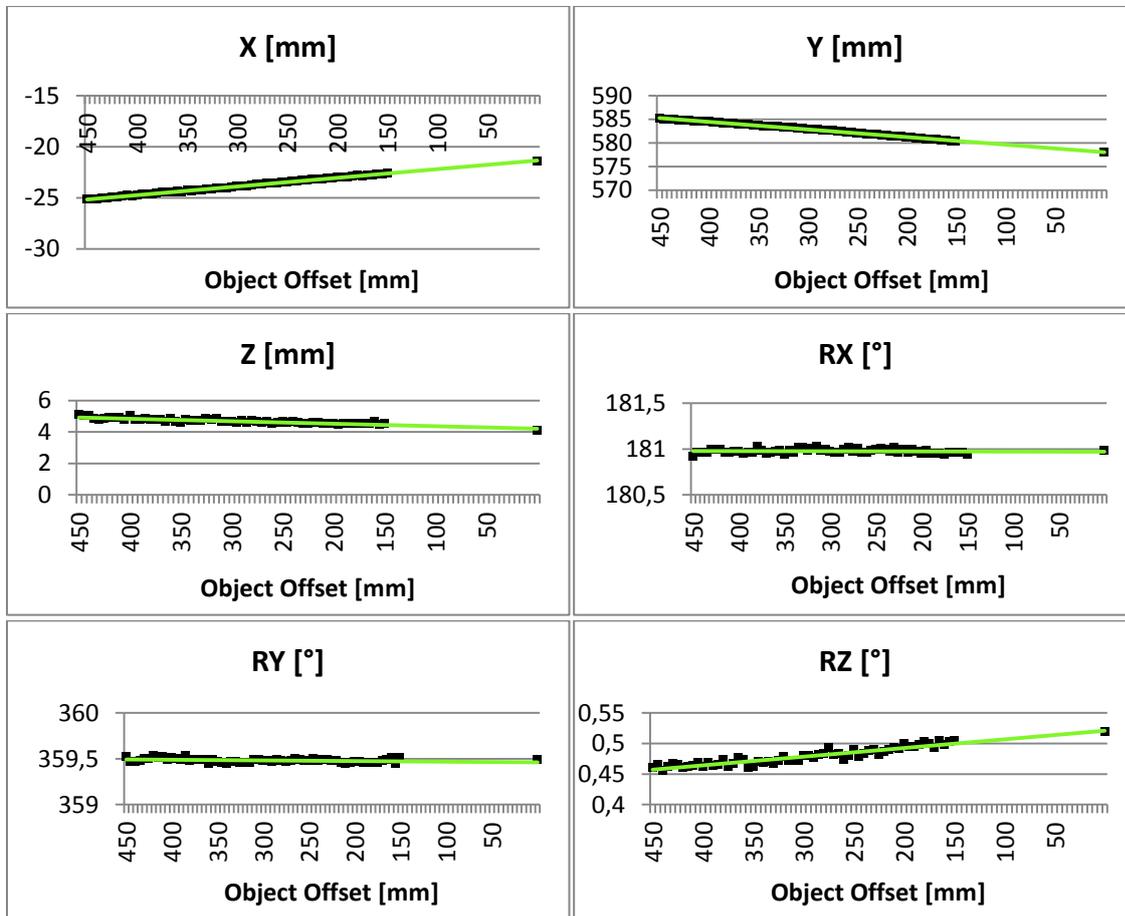


Figure 135: Visualization of the linear extrapolation of the object detections (with respect to the RCS) performed at 60 uniformly distributed support points

Afterwards, the robot was manually controlled to the actual object target pose to get reference values for X, Y, Z, RX, RY and RZ. Table 40 presents the manually determined target pose (Actual Pose) and the target object pose calculated via the extrapolation (Target Pose).

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
Actual Pose	-21,28	577,83	4,24	181,10	395,41	0,53
Target Pose	-21,30	577,91	4,28	180,94	359,49	0,54

Table 40: Comparison between the actual and the extrapolated object target pose

Figure 136 shows how the TCP is controlled to the extrapolated object target pose. The tip of TCP tool coincides almost exactly with the actual object target pose. As previously mentioned in section 5.7.4, the actual target pose was marked by a cross on a label (affixed to the object). The exact location of the cross was defined with a caliper rule.



Figure 136: TCP pose of the robot at the extrapolated target pose of the M2O-Map

The same experiment was performed with the M2M-Map. As before, the tip of the TCP tool coincides almost exactly with the defined target pose (Figure 137).



Figure 137: TCP pose of the robot at the extrapolated target pose of the M2M-Map

5.8.1. Varying Number and Height of Support Points

The key target of this section is to determine an optimal parameter set which delivers a result close to the actual pose values shown in Table 40. The number of support points and the biggest distance between the first support point and the object should be minimal. Thus, the extrapolation depicted in Figure 135 was evaluated for a different set of parameters. Initially, the number of extrapolation points was varied for the movement range of 450 mm to 150 mm. The results are shown in Table 41.

	X [mm]	Y [mm]	Z [mm]	RX [°]	RY [°]	RZ [°]
60 Points 450 mm – 150 mm	-21,30	577,91	4,28	180,94	359,49	0,54
30 Points 450 mm – 150 mm	-21,29	577,91	4,28	180,94	359,49	0,53
15 Points 450 mm – 150 mm	-21,32	577,91	4,28	180,94	359,48	0,53
7 Points 450 mm – 150 mm	-21,37	578,03	4,24	180,99	359,43	0,51

Table 41: Changing number of extrapolation points for the movement range of 450 mm to 150 mm

Then, the movement range was reduced to object Z offset values between 450 mm and 300 mm. The results for different numbers of support points are illustrated in Table 42.

	X [mm]	Y [mm]	Z [mm]	RX [°]	RY [°]	RZ [°]
30 Points 450 mm – 300 mm	-21,44	578,67	3,97	181,06	359,37	0,50
15 Points 450 mm – 300 mm	-21,50	578,60	3,80	181,05	359,32	0,52
7 Points 450 mm – 300 mm	-21,57	578,73	3,74	181,14	359,38	0,49

Table 42: Changing number of extrapolation points for the movement range of 450 mm to 300 mm

In the next step, the support points were uniformly distributed between object Z offset values of 300 mm and 150 mm. The results of this experiment are depicted in Table 43.

	X [mm]	Y [mm]	Z [mm]	RX [°]	RY [°]	RZ [°]
30 Points 300 mm – 150 mm	-21,28	577,72	4,37	180,88	359,47	0,53
15 Points 300 mm – 150 mm	-21,29	577,72	4,31	180,86	359,52	0,53
7 Points 300 mm – 150 mm	-21,26	577,71	4,53	180,88	359,48	0,52

Table 43: Changing number of extrapolation points for the movement range of 300 mm to 150 mm

Finally, the support points were uniformly distributed between object Z offset values of 350 mm and 200 mm. The results for different numbers of support points are depicted in Table 44.

	X [mm]	Y [mm]	Z [mm]	RX [°]	RY [°]	RZ [°]
30 Points 350 mm – 200 mm	-21,23	577,86	4,26	180,92	359,51	0,54
15 Points 350 mm – 200 mm	-21,25	577,82	4,27	181,02	359,37	0,50
7 Points 350 mm – 200 mm	-21,24	577,79	4,29	181,09	359,47	0,54

Table 44: Changing number of extrapolation points for the movement range of 350 mm to 200 mm

Furthermore, I conducted several experiments with 60 support points (uniformly distributed between an object Z offset of 450 mm and 150 mm) at randomly chosen object locations. The extrapolation of the X, Y and Z values always resulted in a final TCP pose close to the actual target. In contrast to the position, the development of the orientation sometimes resulted in a behavior shown in Figure 138. The orientation changed significantly as only two of the four 3D Fiducial Propeller markers remained in the field of view (offset height of about 235 mm). This behavior was also noticed in section 5.7.4 (Figure 132).

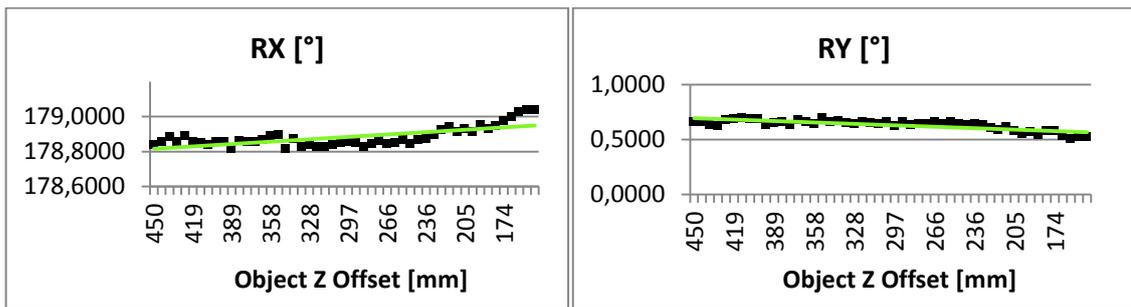


Figure 138: Non-linear development of the parameters RX and RY for a certain test case

In Table 45, the actual values of RX and RY are compared to different extrapolation experiments. The variation of the parameter set led to significant differences in the extrapolation of RX and RY.

	RX [°]	RY [°]
Actual Values	178,96	0,45
50 Points 450 mm – 150 mm	179,11	0,40
15 Points 450 mm – 350 mm	178,90	0,54
15 Points 350 mm – 250 mm	178,84	0,50
15 Points 250 mm – 150 mm	179,42	0,19

Table 45: Actual and extrapolated values of RX and RY for a non-linear development

In the next experiment, I tested the repeatability of the extrapolated TCP pose with different numbers of support points. In order to do this the robot was placed at 10 randomly chosen starting positions and was controlled to the object. The support points for the extrapolation were uniformly distributed between the start object Z offset of 350 mm and the end object Z offset of 200 mm. Table 46 illustrates the repeatability of the TCP pose with respect to 7 support points.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
range_{rep}	0,30	0,22	0,41	0,27	0,09	0,05
tol_{rep}	$\pm 0,47$	$\pm 0,31$	$\pm 0,67$	$\pm 0,37$	$\pm 0,16$	$\pm 0,06$

Table 46: Repeatability of the extrapolated TCP pose for 10 movements – 7 support points

Table 47 shows the repeatability of the TCP pose with respect to 15 extrapolation points.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
range_{rep}	0,18	0,18	0,26	0,14	0,12	0,04
tol_{rep}	$\pm 0,32$	$\pm 0,26$	$\pm 0,36$	$\pm 0,19$	$\pm 0,16$	$\pm 0,06$

Table 47: Repeatability of the extrapolated TCP pose for 10 movements – 15 support points

The repeatability of the TCP pose with respect to 30 support points is depicted in Table 48.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
range_{rep}	0,16	0,12	0,17	0,13	0,07	0,05
tol_{rep}	$\pm 0,26$	$\pm 0,21$	$\pm 0,23$	$\pm 0,17$	$\pm 0,10$	$\pm 0,08$

Table 48: Repeatability of the extrapolated TCP pose for 10 movements – 30 support points

5.8.2. Absolute Accuracy of Robot/Object Interaction

In this section, I evaluate the absolute accuracy of the robot/object interaction when approaching the object defined by the M2O-Map. The resulting deviations are a combination of the inaccuracies of all sub-systems, for example, the camera, the object detection via M2O-Map and the Hand-Eye calibration. Similar to the experiments mentioned in this section, the actual target pose was determined by manually moving the robot to the desired location.

The object was placed at 30 randomly distributed locations. The TCP of the robot was moved via extended visual servoing to the estimated target pose and compared to the manually determined actual pose. The deviations with respect to the RCS are shown in Table 49.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
Mean	-0,11	0,23	-	-0,02	-0,07	-
Std. Dev	0,15	0,18	-	0,06	0,07	-
Min	-0,51	-0,66	-	-0,11	-0,21	-
Max	0,26	0,14	-	0,13	0,07	-

Table 49: Deviation of 30 extrapolated TCP poses with respect to the manually determined actual object target pose

The comparison between the extrapolated and the actual pose was only possible with respect to X, Y, RX and RY. The absolute deviations of the 30 positions in the XY plane are shown in Table 50. The Z deviation was not measured in this experiment, due to the fact that the object target pose of the M2O-Map is located at a hole. Furthermore, RZ was also not evaluated due to the usage of a tool that is rotationally invariant around RZ.

	Mean [mm]	Min [mm]	Max [mm]
Position XY	0,30	0,005	0,71
Orientation XY	0,10	0,001	0,21

Table 50: Absolute XY position deviation between the actual and the extrapolated target pose

Subsequently, I estimated the deviations along the Z-axis. In order to do this, I determined the relation between the XY plane and the Z deviation of the repeatability measurements illustrated in Table 47 (Equation (124)).

$$f_{Z/XY} = \frac{\text{range}_{repz}}{\sqrt{\text{range}_{rep_X}^2 + \text{range}_{rep_Y}^2}} \cdot s = \frac{0,26}{\sqrt{0,18^2 + 0,18^2}} \cdot 1,5 \approx 1,5 \quad (124)$$

The calculated relation was additionally multiplied with a factor $s = 1,5$ to include a safety factor in case that the actual Z deviations are higher. The factor $f_{Z/XY}$ was then multiplied to each of the 30 absolute XY measurements to estimate the absolute Z deviation. The results are depicted in Table 51.

	Mean	Std. Dev	Min	Max
Z [mm]	0,45	0,27	0,008	1,06

Table 51: Z deviation as a result of the multiplication of the XY deviation of the different absolute object positions with the scale factor $f_{Z/XY}$

The combination of the absolute X, Y and Z differences between the actual and the extrapolated target pose is visualized in Table 52.

	Mean [mm]	Min [mm]	Max [mm]
Position XYZ	0,54	0,01	1,27
Orientation XYZ	0,10	0,001	0,21

Table 52: Absolute position and orientation deviation between the actual and the extrapolated target pose

Since the repeatability deviation of RZ is very small compared to RX and RY, I assumed that the deviation of the absolute RZ value is negligible. Thus, the XYZ orientation deviation of Table 52 is equal to the values depicted in Table 50.

The absolute deviation seems to increase as the object is placed closer to the robot base. This behavior is illustrated in Figure 139. The measured deviation values are sorted according to the Y-axis of the RCS.

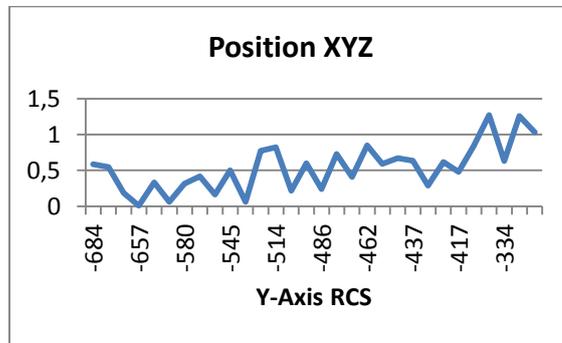


Figure 139: Distance deviation of the actual and the extrapolated target pose sorted according to the Y-axis of the RCS

The results of Figure 139 are also visualized in Figure 140. Here, the color represents the height of the deviation, whereas the brightness of the color is proportional to the size of the difference between the actual and the extrapolated pose. A high difference is equal to a bright color point. The color values were scaled between the minimum and maximum deviation.

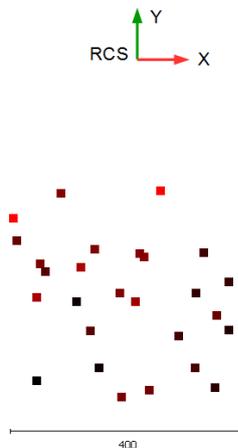


Figure 140: Distance deviation of the actual and the extrapolated target pose

5.9. Time Performance

The mean time performance of 20 extended visual servoing experiments with 15 support points is evaluated in this section. The start and the end object Z offset of the movement are defined to be 350 mm and 200 mm. The robot was placed at 20 randomly chosen starting poses and controlled towards the static object defined by the M2O-Map.

The first movement (position and orientation are corrected) takes about 10 seconds. In average, three refinement iterations are conducted until the difference between the TCP and the first support point is smaller than the pose offset threshold (POT), defined in section 5.7.4. All further detections and movements that are performed to get from one support point to another (only position is corrected) take ~3,20 seconds. On average only one correction iteration has to be conducted at a support point. Those 3,20 seconds can be split up into the following steps:

- ~0,30 seconds: image acquisition (both cameras)
- ~0,90 seconds: image distortion removal (both cameras)
- ~0,70 seconds: 3D marker detection (both cameras)
- ~0,02 seconds: extract object pose from M2O-Map and transform to RCS
- ~1,30 seconds: robot movement to the next support point

The final movement of the robot from the last support point to the extrapolated object target pose takes about 3 seconds. Thus, the mean time to move the robot to an object with the concept of the extended visual servoing can be calculated as shown below:

$$time_{init} + NrSupportPoints * time_{supportPoint} + time_{final} = 10 + 15 \cdot 3,2 + 3 = 61 \text{ seconds}$$

Each robot movement time contains a waiting time of 0,5 seconds where no actions are performed. This waiting process is carried out after each movement, due to vibrations after the robot stops. If the waiting time was set to zero, the vibrations would cause high deviations with respect to the 3D camera detections.

5.10. Pick and Place Experiment

Finally, a pick and place experiment was performed to determine the functionality of the extended visual servoing in combination with the industrial marker to object map. The task was to pick the object defined by the M2M-Map and place it precisely at the target location of the M2O-Map. Changing the pose of the RCS between the pick and place operation should have no effect on the accuracy. The end-effector of the robot was equipped with a gripper that can grasp objects with a maximum size of 101 mm. The target object (introduced in section 5.4) has a size of 100 mm. Therefore, the robot/object interaction accuracy has to be smaller than $\pm 0,5$ mm with respect to the object's X-axis to avoid a collision. Figure 141 shows the TCP, located at the object pick/place position with an opened gripper.

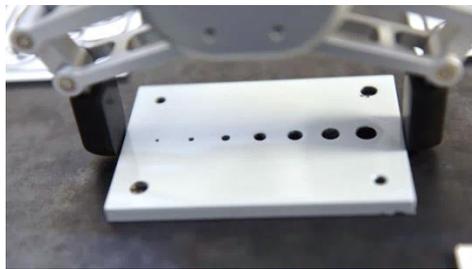


Figure 141: TCP at object pick/place position with opened gripper

The pick and place operations are performed with respect to the M2O-Map and the M2M-Map, defined in section 5.6. The object should be grasped from the M2M-Map and placed on the dowel pins of the M2O-Map. The M2M-Map has no defined position for the object, unlike the M2O-Map. As a consequence, the object cannot be placed manually, but with an initial pick and place step. The object was grasped at the M2O-Map and placed at the M2M-Map by using the extended visual servoing. Due to this initial step, the object pose coincides with the target pose defined by the M2M-Map.

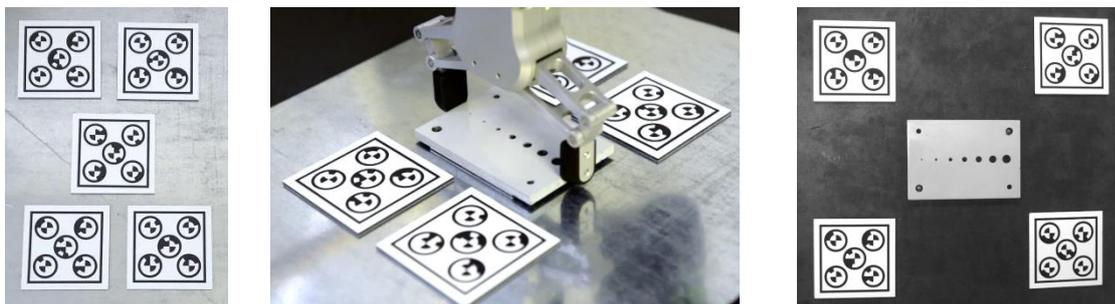


Figure 142: M2M-Map (left), M2M-Map with object placed at the M2M-Map target position using extended visual servoing (middle), M2O-Map (right)

The mobile platform with the industrial manipulator was moved from a randomly chosen starting position to the first location to perform the pick operation (Figure 143). The robot base of the current grasping position at the M2M-Map does not coincide with the robot pose of the previous initial step.



Figure 143: Initial starting pose of mobile platform (left) and mobile platform at the location for the object picking task (right)

The extended visual servoing was performed with 15 support points that were uniformly distributed between the start object Z offset of 350 mm and the end object Z offset of 200 mm. The extrapolated target object pose is visualized in Figure 144. The pick process was performed very accurately. As mentioned before, the gripper was only allowed to have a maximal deviation of $\pm 0,5$ mm with respect to the object's X-axis. In case of a higher deviation, a collision with the object would have occurred.



Figure 144: Final pick position of the robot TCP with respect to the object

After the object was picked, the base coordinate system of the robot (RCS) was changed by moving the mobile platform to another position (Figure 145). The change of the robot basis shows that the movement of the robot towards the object does not depend on a defined RCS. The movement is purely controlled by the camera observations.



Figure 145: Change of the RCS between the pick and place operation by moving the mobile platform

The placement of the object was performed with the same extrapolation parameter set as used in the pick operation. The extrapolated target object pose is visualized in Figure 146. The object was placed on two dowel pins with a clearance fit, which proves that the robot/object interaction was carried out with an absolute accuracy smaller than 1 mm. To successfully place the object on the dowel pins, the TCP has to perform a movement with a tolerance of $\pm 0,3$ mm with respect to the X- and Y-axis of the robot.

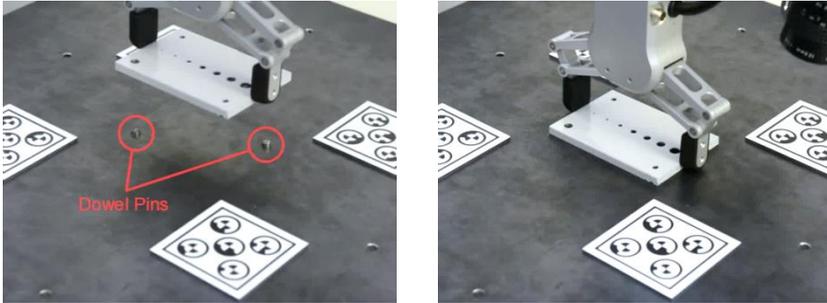


Figure 146: Precise placement of object on two dowel pins at target location

This experiment was conducted at least 20 times and always resulted in an effective robot/object interaction.

6. Discussion

The first experiment, discussed in this part, is the monocular camera calibration. The average mean pixel error of the two monocular calibrations mentioned in section 5.2 (shown in Table 7) are very close to the limit of 0,1 pixels, defined by HALCON [49]. This is a result of the manufacturer tolerance and some damages on the calibration plate (shown in Figure 147). Furthermore, the acquisition of the calibration images was performed in a real industrial environment. The illumination was not homogenous and some of the tilted calibration plate acquisitions were out of focus.

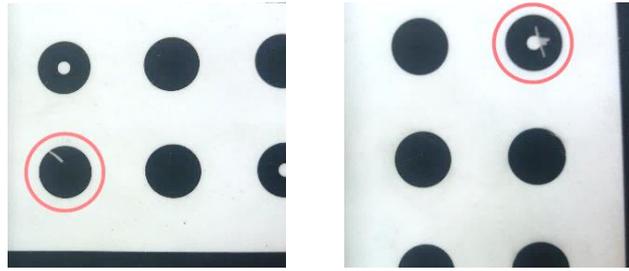


Figure 147: Monocular calibration – visualization of calibration plate defects

All conducted monocular distance measurements in the calibration plane (shown in Table 8) lie within the tolerance of $\pm 0,072$ mm defined by Equation (97). Hence, the monocular calibration was successfully performed and validated.

The extension of the monocular to a binocular calibration with the same calibration images, as explained in section 5.3, led to a rectification error that was also smaller than 0,1 pixels. The measurements with the binocular camera system are very accurate. The stereo system is able to conduct 3D detections with a tolerance of about $\pm 0,17$ mm. The main deviation is caused by inaccuracies along the camera's Z-axis. Measurements along the X- and Y-axis are afflicted only with $\pm 0,05$ mm. As a consequence of the low deviations, the calibration process of the camera system is assumed to be successfully completed.

The evaluation of the object detection algorithms performed in section 5.4 shows that all methods are strongly influenced by outliers. The incorrect object pose determinations are mainly caused by bad illumination of the scene.

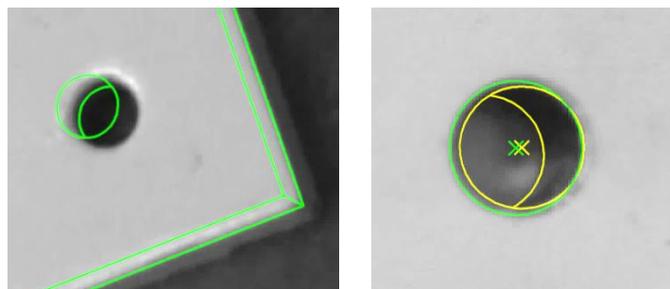


Figure 148: Object detection inaccuracies due to bad illumination. The left image shows deviations of the SBM and SBM-SOT. The right image visualizes an inaccurate SBM-ECT.

Figure 148 (left) illustrates a major target pose deviation of the SBM and SBM-SOT, due to insufficient illumination of the object's side surface. Figure 148 (right) depicts an incorrect result of the SBM-ECT due to bad illumination of the chamfer located at the target hole (reference feature). All experiments of section 5.4 lead to the conclusion that SBM-SOT and SBM-ECT deliver more robust and accurate results than SBM. The tolerance and measurement range with respect to the Z-axis is shown in Table 53. The Z-measurements are afflicted by the previously mentioned outliers.

	SBM	SBM – SOT	SBM – ECT
range _z	6,9 mm	3,4 mm	1,67 mm
tol _z	±6 mm	±2,8 mm	±1,05 mm

Table 53: Accuracy evaluation of 3 different 3D object pose detection methods with respect to the measurement range and deviation along the Z-axis

SBM is more inaccurate, due to the illumination problems and due to its high dependency on a precise correspondence between the CAD and the actual object model. In our case, the object does not exactly correspond with the CAD model. Chamfers and manufacturer tolerances are not considered in the CAD model. Despite the fact that the SBM-SOT and SBM-ECT detections deliver acceptable results, there are also some limitations. The pro and contra arguments are listed below.

SBM-SOT	SBM-ECT
+ No distinctive feature point required	+ Very accurate in case of a good fitting
+ Noise and irregularities to a certain degree do not affect the result	+ Very accurate even if SBM is inaccurate
- SBM has to find a good match that exactly correlates with the actual object	- A distinctive feature is required
	- Small disturbances of the distinctive feature result in severe deviations

Table 54: Pro and contra list of SBM-SOT and SBM-ECT

Using only optimal object detections with homogenous illumination and good calibration data, the resulting poses of the SBM-SOT and SBM-ECT are almost identical. The mean pose deviation between the two methods is about $\pm 0,15$ mm. The detection with supervision is required for both methods to achieve accurate results and to sort out all outliers. Because of its superior tolerance with respect to Z, α and β , SBM-ECT with human supervision is used to build the industrial marker to object map.

In the next step, the precision of the 3D marker detection experiments of section 5.5 are compared to each other. The determined worst case detection accuracies are summed up in Table 55. The difference between the accuracies of the four marker detection methods is significant, especially between the monocular and binocular pose extraction.

	3D ARTM – M	3D ARTM – B	3D QRCM	3D FPM
tol_w	$\pm 2,6$ mm	$\pm 0,40$ mm	$\pm 0,41$ mm	$\pm 0,27$ mm

Table 55: Accuracy evaluation of 4 different 3D marker types

The measurements of the monocular 3D ARToolKit Marker detection (3D ARTM-M, section 5.5.1) deviate severely from the actual data. The accuracy of the resulting marker pose strongly depends on the optimal correspondence between the dimensions of the actual and the defined square marker size. Due to manufacturing tolerances, the dimensions are not completely identical. Furthermore, the farther away the marker is placed from the camera, the more inaccuracies occur because of the decreasing resolution. According to [29], the monocular ARToolKit marker detection favors simplicity and speed over accuracy.

The upgrade of the monocular to a binocular camera system led to a significant improvement. The remaining inaccuracies of the binocular 3D ARToolKit marker (3D ARTM-B, section 5.5.2) detection are the result of an imprecise extraction of the marker's corner points. The extracted 3D points close to the camera center deviate from the 3D points close to the sensor's border region. The points at the border region are probably more inaccurate due to a lower resolution and due to a higher distance between the 3D points and the camera center. Moreover, this deviation can also be the result of an imprecise camera distortion parameter modelling of the camera calibration.

The 3D QR Code marker (3D QRCM, section 5.5.3) detection accuracy is almost equal to the results of the 3D ARTM-B. The QR Code is not constructed to extract precise positions and uses no advanced subpixel refinement. These facts are the cause for pose deviations.

The best results were achieved with the 3D Fiducial Propeller marker (3D FPM, section 5.5.4). The exact 2D marker positions for the triangulation are extracted by using a subpixel refinement with respect to the saddle point of the propeller pattern. Therefore, this marker is chosen for the creation of the industrial marker to object map. Additionally, the 3D Fiducial Propeller marker detection was tested at different detection heights (distance between camera and object). The best measurement results were achieved at a height of 450 mm. Deviations at other heights are created by:

- Inaccurate modelling of the distortion parameters. The distortion parameters were optimized for the calibration height at 450 mm.
- Marker features are out of focus. This results in inaccurate feature detection and matching (shown in Figure 112).
- The quality of the extracted image points reduces with the image resolution.
- The deviation in Z increases as a quadratic function along with the distance between the feature points and the camera center (Equation (42)).

The main effects on the accuracy are probably caused by the change of the focus and the detection height.

Then, the SBM-ECT and 3D-FPM method were combined to build the industrial marker to object map M2O-Map. The M2O-Map consists of four 3D-FPMs and was created under human supervision to manually reject detection outliers. The marker to object relation of each 3D-FPM should describe the same object target pose. The deviations between the four resulting object poses of the four 3D-FPMs in combination with the M2O-Map are visualized in Table 56.

	X [mm]	Y [mm]	Z [mm]	α [°]	β [°]	γ [°]
Deviation	0,03	0,05	0,17	0,07	0,1	0,03

Table 56: Maximal deviation of the four object poses that were determined from each of the four 3D-FPMs in combination with the M2O-Map

The deviations between the four object poses are caused by the inaccuracies of the 3D-FPM and SBM-ECT detection methods. Additionally to the M2O-Map, the M2M-Map was built. Instead of an object, the target pose is now described by another 3D-FPM. The M2M-Map is more accurate with respect to the repeatability and the deviation between the four extracted target poses. This result is reasonable. The creation of the M2M-Map has to be more precise, as the extraction of the 3D-FPM pose is more accurate than the extraction of the object pose using SBM-ECT.

In the next step, the precision of the industrial manipulator UR10 of Universal Robots was evaluated (section 5.7.1). These experiments were conducted to get a coarse overview on the absolute accuracy of the robot (no specifications of the manufacturer). The error between the actual robot position and the position provided by the robot controller changes almost linearly with respect to the absolute distance between the TCP pose and the robot coordinate system (RCS). Also the orientation about RY and RZ changes almost linearly, but regarding to the X axis of the RCS. In conclusion, the robot pose provided by the robot controller seems to have an absolute deviation of more than 1 mm in position and about 1,5° in orientation from the actual pose. As mentioned before, the size of the error depends on the current position of the TCP.

The inaccuracies of the kinematic model of the UR10 affect the result of the Hand-Eye calibration shown in section 5.7.2. The remaining errors of the chain transformations presented in Table 35 are very high. As a consequence, about 60 calibration images had to be taken until the development of the Hand-Eye pose parameters seemed to converge to a constant value. Nevertheless, the transformation between the camera and the TCP cannot be estimated exactly with this robot.

The errors of the UR10 and the inaccurate Hand-Eye calibration have a great effect on the experiment conducted in section 5.7.3. The detection of the static 3D-FPM measurement plate at 100 different TCP positions resulted in severe pose deviations. In case of optimal conditions, the transformation of each of the 100 3D-FPM observations would result in the same four poses in the RCS. But instead, the poses in the RCS drift

off as the robot moves its TCP (depicted in Figure 149). Especially the erroneous development of the TCP RX and RY orientation causes large deviations at big detection heights, as it severely changes the observed 3D marker position. This experiment confirms that a compensation of the robot's inaccuracies via a visual servoing algorithm and an extrapolation seems inevitable.

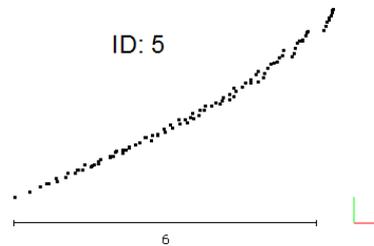


Figure 149: Drift of the 3D marker position that was transformed to RCS as the robot TCP moved to different locations in the shape of a grid

In section 5.7.4, the implementation of the position based visual servoing approach was tested. The projection of the detected target and TCP pose always resulted in a good match. The small deviation between the projections is a result of the pose offset threshold (POT) defined in Table 37. The projections coincide almost perfectly, even if the robot moves towards the object along its TCP Z-axis (Figure 130). This experiment leads to the conclusion that the erroneous movement of the TCP is successfully compensated and that the target pose changes linearly as the robot moves towards the object.

Furthermore, the deviations of the object detections transformed to the RCS for 10 different TCP movements towards the same object (shown in Table 38) are smaller than the static repeatability of the SBM-ECT in the CCS (depicted in Table 13). Therefore, one can conclude that the introduction of the industrial marker to object map M2O-Map increased the object detection precision. Moreover, it is further proof that the concept of visual servoing was implemented correctly. The TCP of the robot reaches almost the same target pose for each movement that is conducted with randomly chosen starting positions (Table 39). The repeatability measure of the TCP pose in the RCS is additionally afflicted with the POT. Therefore, the deviations of the TCP pose in the RCS (Table 39) are higher than the deviations of the object detections in the RCS (Table 38). The position based visual servoing approach in combination with the M2M-Map also results in an almost perfect match of the TCP and object pose projection.

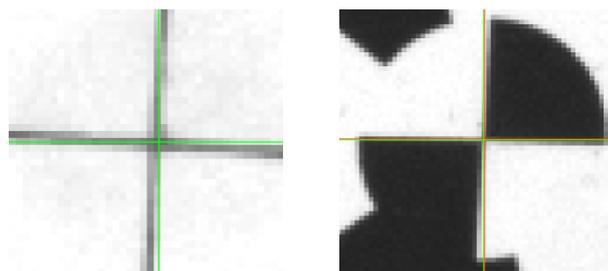


Figure 150: Projection of the current TCP and detected object pose to the image plane

The experiments of section 5.8 showed that the linear extrapolation of the object detections (transformed to the RCS) with 50 support points leads to a very good estimation of the final TCP target pose. The tip of the tool coincides almost exactly with the manually determined actual target pose (depicted in Figure 151).



Figure 151: Extrapolated TCP target pose of the M2O-Map (left) and the M2M-Map (right) with the use of 50 support points uniformly distributed between the object Z offset of 450 mm and 150 mm.

The effects on the extrapolated result when changing the extrapolation parameter set (number of support points, start and end object Z offsets) were tested with respect to the M2O-Map. The M2O-Map represents the map that will actually be used in an industrial environment (3D FPM in combination with an object defined by a CAD model) and is therefore defined as a reference for all evaluations concerning the absolute accuracy. The experiments of section 5.8.1 led to the following statements:

- The more markers are observed during the movement, the better the extrapolated result is going to be, especially with respect to the orientation. The mean of many poses is more robust.
- If the object Z offset is too big (450 mm), the deviations of the extrapolated target position increase. The acquired images are out of focus and the stereo system Z deviations get worse as the distance between the object and the camera increases.
- If the object Z offset is too small (150 mm), the deviations of the extrapolated target orientation increase. The orientation estimation is more robust since many markers are observed. Below a height of about 235 mm, two of the markers are out of the field of view.
- The higher the number of support points is, the better the repeatability and therefore the overall accuracy will be. The extrapolation becomes more robust as the support point number is increased, which is of advantage in case of outliers or wrong detections.

The camera distance to the object at an object Z offset of 450 mm is identical to 580 mm. The camera distance at the object Z offset of 150 mm is equal to 280 mm. Both heights are far off the optimal camera to object distance of 450 mm, which was defined to be the height where all acquisitions are in focus and where the calibration is performed. Due to the previously mentioned experiments and statements to the parameter set, 15 support points were used and uniformly distributed between the start object Z

offset of 350 mm (camera height of 480 mm) and the end object Z offset of 200 mm (camera height 330 mm). Thus, the range of the support points is close to the optimal image acquisition height. Furthermore, only a few of the support points lie below 235 mm (where only two markers can be observed). The reduction of 50 to 15 extrapolation points speeds up the application and still ensures a good repeatability.

This parameter set was then further used to determine the absolute TCP/object interaction accuracy in section 5.8.2. The comparison of 30 extrapolated and manually determined actual TCP target poses led to the conclusion that the accuracy decreases as the distance between the robot base and the object is reduced. The deviation of 30 measurements between the extrapolated and the actual target pose is depicted in Table 50 and Table 52.

Only three out of 30 deviations consisted of a higher deviation than the desired maximum of 1 mm. The mean deviation is 0,53 mm. If the work space of the pick and place handling is restricted to an area that is at least 420 mm apart of the origin of the RCS, the TCP/object interactions can be performed with a maximal deviation of 0,85 mm and a mean deviation of 0,47 mm. The absolute accuracy significantly improves with a growing distance between the OCS and the RCS.

The time performance of this application was evaluated in section 5.9 and is very low. The movement of the robot towards the desired target position was performed with 15 support points, uniformly distributed between an object Z offset of 350 mm and 200 mm. The movement currently takes about 61 seconds. Especially the image acquisition and the removal of the image distortion consume a lot of time.

Finally, a pick and place experiment was conducted to evaluate the application in an actual example of industrial usage. The experiment was carried out several times for different object poses and always resulted in a successful operation. Due to the small difference between the object and gripper size, the application had to perform a TCP/object interaction with an accuracy that lies within $\pm 0,5$ mm along the X-axis of the object. In fact, the placement of the object with respect to the M2O-Map was performed with an accuracy that had to be close to $\pm 0,3$ mm along the X- and Y-axis. Otherwise it would not have been possible to place the object on the dowel pins with clearance fit (Figure 152). The $\pm 0,3$ mm tolerance is a combination of the clearance fit of the dowel pins and the elasticity of the robot gripper itself.

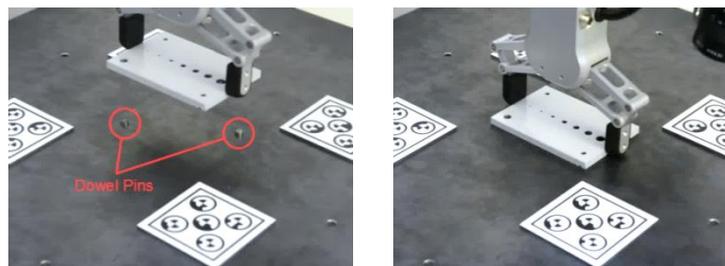


Figure 152: Precise placement of object on two dowel pins at the desired target location

7. Conclusion and Future Work

In this master thesis, the extended visual servoing procedure has been presented. It consists of the industrial marker to object map and a linear TCP pose extrapolation in combination with the common concept of visual servoing. This procedure is used to control the TCP of an industrial manipulator, mounted on a mobile platform, exactly to a fixated target object (for instance, an object in a clamping device). All robot movements towards the pre-defined target object are purely based on the observations of the industrial marker to object map.

The calibration of the end-effector mounted stereo system was performed with a high precision calibration plate (manufacturer tolerance of $\pm 0,03$ mm) and resulted in a 3D measurement accuracy of about $\pm 0,17$ mm. Based on this stereo system, the 3D object detection algorithm Shape-Based 3D Matching (SBM) and some extensions of this method (Shape Based 3D Matching – Shape Origin Triangulation (SBM-SOT), Shape Based 3D Matching – Ellipse Center Triangulation (SBM-ECT)) were evaluated. Furthermore, the detection accuracy of the already existing 3D ARToolKit marker, the self-made 3D QR Code marker and the 3D Fiducial Propeller marker were tested. Based on the evaluation of the measurements, the SBM-ECT and the 3D Fiducial Propeller marker turned out to be the best choice for the creation of the industrial marker to object map (M2O-Map).

Accuracy measurements of the Universal Robot UR10 in combination with the hand mounted stereo system turned out to be very inaccurate. As a consequence, the evaluation of the Hand-Eye calibration also confirmed the presence of severe errors. The chain transformation of all 60 calibration poses resulted in a mean error of about 1,7 mm with respect to the Z-axis of the robot coordinate system (RCS). These experiments led to the conclusion that a compensation of the robot's inaccuracies via a visual servoing algorithm and a linear TCP extrapolation is inevitable.

The basic visual servoing procedure performed well in combination with the M2O-Map. The experiments resulted in an object detection repeatability of $\pm 0,16$ mm and a TCP pose repeatability of $\pm 0,36$ mm along the RCS Z-axis (maximal deviation along Z). Furthermore, it is notable that the pose of the object detection changes almost linearly as the robot TCP approaches the desired target object.

As a consequence, the usage of a linear extrapolation of robot movement, based on information gathered during the object approach, led to a good compensation of the kinematic deviations. The absolute accuracy of the extended visual servoing concept turned out to be smaller than 1 mm, in case that the object is placed at least 420 mm away from the origin of the RCS. A final pick-and-place experiment in an industrial environment confirmed this assumption. The object handling from one workstation to another was performed with an absolute accuracy of about $\pm 0,3$ mm. A change of the RCS with respect to the work station had no effect on the precision of the robot/object interaction. In this thesis, every TCP movement towards the object was performed

based on camera observations of the industrial marker to object map. No pose teaching of the manipulator was required to reach the desired target pose.

All of these experiments show that the extended visual servoing concept is very promising for mobile robot applications in an industrial environment. However, currently the application takes about 61 seconds to reach the desired target pose which is certainly too slow for industrial usage. Therefore, in a next step one could try to improve this application and reduce the amount of time required to interact with objects. One possible approach is to speed up the removal of image distortions which requires about 0,9 seconds for the two stereo system acquisitions. Furthermore, the inaccurate Universal Robot UR10 could be replaced with a more precise robot to increase the accuracy of the absolute movements. This exchange would lead to a reduction in the extrapolation support points and therefore to a significant improvement of the process time. In addition, the currently required waiting time of 0,5 seconds after each robot movement (due to vibrations) could probably be eliminated.

Moreover, the current state of the application does not allow the object to change its position with respect to the markers without redefining the industrial marker to object map first. Thus, as an extension of this thesis, it would be interesting to add a robot/object interaction for objects with a varying relation to the 3D markers. For example, objects that are fed in a working cell with a conveyor belt have an undefined pose and could be grasped by a robot with the new extension. The 3D markers might be used to define the plane of operation of the conveyor belt to get robust values with respect to the Z coordinates. The precise location along the X- and Y-axis could then be determined by applying a simple 2D detection in this plane. This extension would significantly increase the areas of application of the extended visual servoing concept and the industrial marker to object map.

8. Bibliography

- [1] ARToolKit, "About the Traditional Template Square Marker," [Online]. Available: https://artoolkit.org/documentation/doku.php?id=3_Marker_Training:marker_about. [Accessed 11 01 2017].
- [2] R. S. Andersen, C. Schou, J. S. Damgaard, O. Madsen and T. B. Moeslund, "Human Assisted Computer Vision on Industrial Mobile Robots," in *Proceedings of 1st AAU Workshop on Human-Centered Robotics*, 2013.
- [3] BA Systems, "STAMINA - Mobile robot for part handling in the automotive industry," [Online]. Available: <https://www.basystemes.com/en/rd-projects/stamina/>. [Accessed 19 02 2017].
- [4] Bosch GmbH, "Machine and Process Capability," Quality Management in the Bosch Group, Stuttgart, 2004.
- [5] A. Botthof, "Zukunft der Arbeit in Industrie 4.0," Springer Verlag, 2015.
- [6] D. Bradley and W. Heidrich, "Binocular Camera Calibration Using Rectification Error," in *Computer and Robot Vision (CRV)*, Ontario, Canada, 2010.
- [7] G. Bradski and A. Kaehler, "Learning OpenCV 3: Computer Vision with the OpenCV Library," O'Reilly Media, Inc, USA, Sebastopol, United States, 2017.
- [8] C. Chang and S. Chatterjee, "Quantization Error Analysis in Stereo Vision," in *Twenty-Sixth Asilomar Conference on Signals, Systems and Computers*, 1992.
- [9] F. Chaumette and S. Hutchinson, "Visual Servo Control Part I: Basic Approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82-90, 2007.
- [10] P. I. Corke, *Visual Control of Robots: High-Performance Visual Servoing*, New York, NY: Research Studies Press, 1996.
- [11] S. Daftry, M. Maurer, A. Wendel and H. Bischof, "Flexible and User-Centric Camera Calibration using Planar Fiducial Markers," in *British Machine Vision Conference*, 2013.
- [12] A. De Luca, "Robotics 2 - Visual servoing," Università di Roma, [Online]. Available: http://www.diag.uniroma1.it/~deluca/rob2_en/17_VisualServoing.pdf. [Accessed 11 01 2007].
- [13] O. Demirkaya, M. H. Asyali and P. Sahoo, *Image Processing with MATLAB: Applications in Medicin and Biology*, CRC Press, 2009.

- [14] J. Diebel, "Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors," in *Stanford University*, Stanford, California, 2006.
- [15] DLR - Institute of Robotics and Mechatronics, "Autonomous Pick and Place Operations in Industrial Production," German Aerospace Center, 19 04 2016. [Online]. Available: <https://www.youtube.com/watch?v=fozaEzrl4Aw>. [Accessed 14 02 2017].
- [16] F. Dornaika and R. Horaud, "Simultaneous Robot-World and Hand-Eye Calibration," in *IEEE Transactions on Robotics and Automation*, 1998.
- [17] B. Dorst, M. Ulrich, N. Navab and S. Ilic, "Model Globally, Match Locally: Efficient and Robust 3D Object Recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, 2010.
- [18] A. Dömel, S. Kriegel, M. Brucker and M. Suppa, "Autonomous Pick and Place Operations in Industrial Production," in *International Conference on Ubiquitous Robots and Ambient Intelligence*, Korea, 2015.
- [19] D. Eberly, "Euler Angle Formulas," 10 03 2014. [Online]. Available: <https://www.geometrictools.com/Documentation/EulerAngles.pdf>. [Accessed 31 01 2017].
- [20] A. W. Fitzgibbon, M. Pilu and R. B. Fisher, "Direct Least Squares Fitting of Ellipses," Department of Artificial Intelligence, The University of Edinburgh, Edinburgh, Scotland, 1996.
- [21] J. Friedrich, P. Schwarzkopf, F. Seus and A. Veerkamp-Walz, "Work 4.0 - Humans at its heart," VDMA, [Online]. Available: <https://humans-machines-progress.com/reportage/work-4-0-humans-at-its-heart/>. [Accessed 15 02 2017].
- [22] D. Gallup, J.-M. Frahm, P. Mordohai and M. Pollefeys, "Variable Baseline/Resolution Stereo," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, 2008.
- [23] H. Gattringer, *Starr-elastische Robotersysteme*, Heidelberg, Berlin: Springer-Verlag, 2011.
- [24] Y. Gu and W. Zhang, "QR Code Recognition Based On Image Processing," in *International Conference on Information Science and Technology*, Jiangsu, China, 2011.
- [25] R. Halir and J. Flusser, "Numerically Stable Direct Least Squares Fitting of Ellipses," Department of Software Engineering, Charles University, Prague, Czech Republic, 1998.
- [26] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision (Second Edition)*, Cambridge University Press, 2004.
- [27] J. Heikkilä, "Geometric Camera Calibration Using Circular Control Points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1066-1077, 2000.

- [28] J. Heller, D. Henrion and T. Pajdla, "Hand-Eye and Robot-World Calibration by Global Polynomial Optimization," in *IEEE International Conference on Robotics and Automation*, 2014.
- [29] M. Hirzer, "Marker Detection for Augmented Reality Applications," Inst. for Computer Graphics and Vision, Graz University of Technology, Graz, Austria, 2008.
- [30] A. Hofhauser, C. Steger and N. Navab, "Perspective Planar Shape Matching," in *Image Processing: Machine Vision Applications II*, San Jose, CA, USA, 2009.
- [31] D. Holz, A. Topalidou-Kyniazopoulou, J. Stückler and S. Behnke, "Real-Time Object Detection, Localization and Verification for Fast Robotic Depalletizing," in *International Conference on Intelligent Robots and Systems*, Hamburg, Germany, 2015.
- [32] S. Hutchinson, G. Hager and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, pp. 651-670, 1996.
- [33] M. Hvilshoj, S. Bogh, O. Madsen and M. Kristiansen, "Calibration Techniques for Industrial Mobile Manipulators: Theoretical configurations and Best practices," in *41st International Conference on Robotics*, 2010.
- [34] M. Hvilshoj, S. Bogh, O. S. Nielsen and O. Masen, "Autonomous industrial mobile manipulation (AIMM): past, present and future," *Industrial Robot: An International Journal*, vol. 39, pp. 120-135, 2012.
- [35] JAI, "GO-5000M-PGE / GO-5000C-PGE," [Online]. Available: <http://www.jai.com/en/products/go-5000-pge>. [Accessed 26 02 2017].
- [36] H. Kato, "Inside ARToolKit," ARToolKit, [Online]. Available: <https://artoolkit.org/documentation>. [Accessed 14 02 2017].
- [37] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmentedreality conferencing system," in *2nd IEEE and ACM International Workshop on Augmented Reality*, 1999.
- [38] J. Kilian, "Simple Image Analysis By Moments," Durham University, Durham, United Kingdom, 2001.
- [39] D. Kragic and H. I. Christensen, "Survey on Visual Servoing for Manipulation," Department of Numerical Analysis and Computing Science, Stockholm, 2002.
- [40] S. Kriegel, "Autonomous Industrial Mobile Manipulator," DLR - Institute of Robotics and Mechatronics, [Online]. Available: http://www.dlr.de/rmc/rm/en/desktopdefault.aspx/tabid-11409/19997_read-46794/. [Accessed 19 02 2017].
- [41] S. Kucuk and Z. Bingul, "Robot Kinematics: Forward and Inverse Kinematics," INTECH Open Access Publisher, 2006.

- [42] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld and Hoffmann Michael, "Industrie 4.0," Springer Fachmedien, Wiesbaden, 2014.
- [43] V. Lepetit and P. Fua, "Keypoint Recognition Using Randomized Trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(9), pp. 1465-1479, October 2006.
- [44] Y. Matsumoto, K. Sakamoto, S. Nomura, T. Hirotsu, K. Shiwa and M. Hirakawa, "Activity Replay System of Life Review Therapy Using Mixed Reality Technology," *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. Vol I, no. Hong Kong, 2009.
- [45] D. C. Montgomery, E. A. Peck and G. G. Vining, *Introduction to Linear Regression Analysis*, Wiley, 2015.
- [46] R. Mukundan and K. R. Ramakrishnan, "Moment Functions in Image Analysis: Theory and Applications," World Scientific, London, United Kingdom, 1998.
- [47] E. Muniz, *Universal Robots - Kinematic Calibration Manual for CB3*, Universal Robots, 2009-2016.
- [48] R. M. Murray, Z. Li and S. S. Sastry, "A Mathematical Introduction to Robotic Manipulation," CRC Press, Berkeley, 1994.
- [49] MVTec Software GmbH, "Halcon/HDevelop 13.0 - Reference Manual (en)," [Online]. Available: <http://www.mvtec.com/fileadmin/Redaktion/mvtec.com/documentation/halcon/halcon-13.0-reference-hdevelop.pdf>. [Accessed 16 02 2017].
- [50] MVTec Software GmbH, "Solution Guide II-B Matching," München, Germany, 2010-2016.
- [51] MVTec Software GmbH, "Solution Guide III-C 3D Vision," München, Germany, 2003-2016.
- [52] R. J. Prokop and A. P. Reeves, "A Survey of Moment-Based Techniques For Unoccluded Object Representation and Recognition," Cornell University, Ithaca, New York, 1992.
- [53] P. L. Reu, W. Sweatt, T. Miller and D. Fleming, "Camera system resolution and its influence on digital image correlation," Albuquerque, NM, 2014.
- [54] Robotnik Automation S.L.L, "Mobile manipulator RB-1," [Online]. Available: <http://www.robotnik.eu/manipulators/rb-one/>. [Accessed 19 02 2017].
- [55] P. L. Rosin, "Measuring Rectangularity," Brunel University, United Kingdom, 1999.

- [56] M. Rumpler, S. Daftry, A. Tscharf, R. Pretenthaler, C. Hoppe, G. Mayer and H. Bischof, "Automated End-to-End Workflow for Precise and Geo-accurate Reconstructions using Fiducial Markers," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 135-142, 2014.
- [57] M. R ther, *Camera Hardware and Technology*, Institut f r Maschinelles Sehen und Darstellen, TU Graz: Robot Vision, 2015.
- [58] R. Siegwart, I. R. Nourbarkhsh and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, Massachusetts Institute of Technology, 2011.
- [59] M. W. Spong, S. Hutchinson and M. Vidyasagar, "Robot Dynamics and Control," 2004.
- [60] C. Steger, "Unbiased Extraction of Curvilinear Structures from 2D and 3D Images," Technische Universit t M nchen, M nchen, 1998.
- [61] K. H. Strobl and G. Hirzinger, "Optimal Hand-Eye Calibration," in *International Conference on Intelligent Robots and Systems*, 2006.
- [62] TAPAS, "TAPAS - Robotics-enabled Logistics and Assistive Services for the Transformable Factory of the Future," [Online]. Available: <http://www.tapas-project.eu/index.html>. [Accessed 19 02 2017].
- [63] Technische Universit t M nchen, "TUM-Rosie," [Online]. Available: <http://ias.cs.tum.edu/robots/tum-rosie>. [Accessed 19 02 2017].
- [64] M. Ulrich, C. Wiedemann and C. Steger, "Combining Scale-Space and Similarity-Based Aspect Graphs for Fast 3D Object Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1902-1914, October 2012.
- [65] Universal Robots, "Universal Robots UR10 - Technical Details," [Online]. Available: https://www.universal-robots.com/media/1514624/101081_199917_ur10_technical_details_web_a4_art03_rls_de.pdf. [Accessed 26 02 2017].
- [66] K. Waldron and J. Schmiedeler, "Handbook of Robotics," USA, 2007.
- [67] Willow Garage, Inc, "Robot for Research and Innovation," [Online]. Available: <http://www.willowgarage.com/pages/pr2/overview>. [Accessed 19 02 2017].
- [68] H. Zhuang, Z. S. Roth and R. Sudhakar, "Simultaneous Robot/World and Tool/Flange Calibration by Solving Homogenous Transformations Equations of the Form $AX=YB$," in *IEEE Transactions on Robots and Automation*, 1994.

9. List of Figures

Figure 1: Visualization of the four industrial revolutions [21]	1
Figure 2: Classification of the AIMM domain with respect to the other robotic systems [34].....	2
Figure 3: Concept of AIMM usage – autonomous industrial mobile manipulators supply all working stations (assembly or automated manufacturing station) with parts from the storage shelter	3
Figure 4: Significant change of the mobile platform pose at the workstation. Nevertheless, the robot should perform accurate pick and place operations	4
Figure 5: Simplified workflow for pick and place operations described in this thesis	4
Figure 6: Setup for the definition of the industrial marker to object map. In an initial step, numerous object detections are put into relation to the four 3D markers which surround the object.....	5
Figure 7: Autonomous academic mobile manipulators – RB-1 [54] (left), TUM-Rosie [63] (middle) and PR2 [67] (right)	6
Figure 8: Autonomous industrial mobile manipulators – project STAMINA [3] (left), project TAPAS [62] (middle) and the project of DLR [40] (right).....	7
Figure 9: Basic visual servoing concept (adapted from [39]).....	8
Figure 10: Image based visual servoing - basic concept [12].....	9
Figure 11: Position based visual servoing - basic concept [12].....	9
Figure 12: Deformable Planar Surface-Based 3D Matching – example (adapted from [30]).....	10
Figure 13: Descriptor-Based 3D Matching – example (adapted from [43]).....	10
Figure 14: Surface-Based 3D Matching – example (adapted from [64]).....	11
Figure 15: Shape-Based 3D Matching – example (adapted from [17]).....	11
Figure 16: Marker types – a) 3D ARToolKit square marker [1]. b) 2D QR Code marker. c) 2D Fiducial Propeller marker (adapted from [56])	12
Figure 17: Multiple 2D QR Codes and 2D Fiducial Propeller markers combined as 3D markers.....	12
Figure 18: Influence of a changing amount of iris aperture on the blur spot of the image [57]	17
Figure 19: Pinhole camera – basic concept of mapping a 3D point X to the image plane [26]	17
Figure 20: The Euclidean transformation using the rotation R and translation t to describe the relation between the CCS and the WCS [26].....	18
Figure 21: Image ellipse calculation (right) of an extracted region (left) via second order moments for object approximation [46].....	21
Figure 22: Epipolar geometry - point correspondence geometry (adapted from [26])...22	22
Figure 23: Epipolar constraint - deviations due to errors in the measured image coordinates [26].....	23

Figure 24: Stereo calibration - image acquisition of the calibration plate (adapted from [51]).....	24
Figure 25: Stereo calibration - evaluating the extrinsic parameters using calibration grid $i1$ [6].....	24
Figure 26: Distance resolutions as a function of distance for four different configurations of focal length and base lines [51]	26
Figure 27: Denavit-Hartenberg convention for a serial chain manipulator [59].....	27
Figure 28: Relationship between forward and inverse kinematics [41].....	27
Figure 29: Two-link planar manipulator [59].....	28
Figure 30: Inverse kinematics – two solutions for the same pose [59]	29
Figure 31: Machine and process capability – comparison between the C_m and C_{mk} index (adapted from [4])	30
Figure 32: Overall workflow to realize a pick/place handling at a workstation called W1	31
Figure 33: Basic visual servoing concept (adapted from [39]).....	32
Figure 34: Extended visual servoing concept. Step 1 – define details of the robot control and tracking process (adapted from [39]).....	32
Figure 35: Extended visual servoing concept. Step 2 – creation of the industrial marker to object map at a workstation called W1	33
Figure 36: Extended visual servoing concept. Step 2 – 3D object detection via an industrial marker to object map (adapted from [39]).....	33
Figure 37: Extended visual servoing concept. Step 2 – add an extrapolation handling to the detailed robot control process (adapted from [39])	34
Figure 38: Extended visual servoing concept. Step 2 – creation of industrial marker to object detection for extended object detection.....	35
Figure 39: Industrial marker to object map – simultaneous observation of marker and object at different camera positions (left); observation in terms of coordinate systems (right)	35
Figure 40: Industrial marker to object map –relations mT_{oi} between multiple markers and the object.....	36
Figure 41: Shape-Based Matching - image of two different colored clamps (left) and the CAD model of the clamp (right) [64].....	37
Figure 42: Shape-Based Matching – building a hierarchical model of different views around a virtual object defined by a 3D CAD model [64].....	38
Figure 43: Shape-Based Matching: 3-channel model image as RGB color image. The three visualizations on the right show the extracted edges with increasing threshold T_{amp} [64]	39
Figure 44: Shape-Based Matching – visualization of the tree structure of the hierarchical model and the process of the hierarchical search [64].....	39
Figure 45: Shape-Based Matching – inaccurate object detection due to round edges, inaccurate CAD model and bad illumination [64].....	40

Figure 46: Shape-Based Matching - visualized coordinate system of the object shape center in the left and right image of a stereo system	41
Figure 47: Shape-Based Matching – visualization of ellipse center after estimating the best fitting ellipse using least squares optimization.....	42
Figure 48: Extended visual servoing concept. Step 2 – 3D marker detection for the use of an industrial marker to object map (adapted from [39])	44
Figure 49: 3D ARToolKit marker – visualization of the square black border of known size and the sub-image for identification (adapted from [1])	44
Figure 50: 3D ARToolKit marker – definition of camera and marker coordinate systems and the visualization of the ideal marker points (x_c, y_c) as well as the undistorted marker points (x_c, y_c) (adapted from [44])	45
Figure 51: 3D ARTolKit marker – basic concept of subpixel refinement of corner points [7]	46
Figure 52: 3D ARTolKit marker – triangulation of all 4 corner points to get 3D position and build the orientation vector for RX, RY and RZ.....	47
Figure 53: 3D QR Code marker – visualization of the marker concept. The marker has 4 consistent markers (Ref, X_Axis, Y_Axis, Diag) and 1 identification marker (CodeX)	48
Figure 54: 3D QR Code marker - clipping of the region and the fitted rectangle to get a measure for rectangularity [55]	49
Figure 55: 3D QR Code marker - visualization of the three QR Code localization patterns (adapted from [24])	49
Figure 56: 3D QR Code marker - triangle formed by detection patterns to classify into ‘Top’, ‘Right’ and ‘Bottom’ pattern (adapted from [24])	50
Figure 57: 3D QR Code marker – triangulation of all 5 QR code centers to get 3D position and build the orientation vector for RX, RY and RZ.....	50
Figure 58: 3D Fiducial Propeller marker – visualization of the marker concept. The marker has 4 consistent markers (ID = 1,2,3,4) and 1 identification marker (ID \geq 5)	51
Figure 59: 3D Fiducial Propeller marker - sampling of points on an ellipse	52
Figure 60: 3D Fiducial Propeller marker – the variation of s and t can be used to create a mapping for the x and y direction. The remapping of the marker results in a rectangular image that can be used to extract the marker identification (adapted from [56]).	53
Figure 61: Extended visual servoing concept. Step 2 – robot control with 3D target positions of the camera system (adapted from [39])	54
Figure 62: Hand-Eye and Robot-World calibration – declaration of transformation matrices (adapted from [28])	55
Figure 63: Notation of the coordinate systems and transformations between the coordinate systems (adapted from [28])	57
Figure 64: Iterative robot control towards the desired target pose	57

Figure 65: Position-based Cartesian robot control: initial pose definition with kinematic chain (left) and refinement concept without the kinematic chain (right) (adapted from [28])	58
Figure 66: Extended visual servoing concept. Step 3 – robot control via target pose extrapolation (adapted from [39]).....	59
Figure 67: Target pose extrapolation – extrapolation procedure based on the target coordinates of the OCS with respect to the RCS	60
Figure 68: Linear regression of gathered data	61
Figure 69: Visualization of the conducted experiment – the mobile platform is moved to a work station. The industrial manipulator grasps the object precisely by using the industrial marker to object map and the extended visual servoing method.....	62
Figure 70: Visualization of the Universal Robot UR10 [65]	63
Figure 71: End-effector of the industrial robot with stereo system and gripper.....	64
Figure 72: Monocular calibration – definition of the calibration plate and measurement points.....	65
Figure 73: Monocular calibration of c_1 – development of camera parameters with growing number of calibration images	66
Figure 74: Monocular calibration of c_2 – development of camera parameters with growing number of calibration images	67
Figure 75: Binocular calibration – development of the pose parameters with growing number of calibration images	69
Figure 76: Binocular calibration – evaluation of the epipolar constraint.....	70
Figure 77: Binocular calibration – distance measurements between the corner points of the calibration plate at 70 random positions in a plane	70
Figure 78: Worst case scenario of an absolute stereo system measurement error. The systematic error is assumed to be the maximum possible value	71
Figure 79: binocular calibration – Z deviation of the calibration plate at multiple positions in a plane	72
Figure 80: 3D object detection – rectangular object for detection (left) and measurement plate to evaluate the accuracy of the detection methods (right)	73
Figure 81: visualization of object positioning on measurement plate by fixing the object with dowel pins	73
Figure 82: SBM – evaluation of the distance measurements between P0, P1, P2 and P3	74
Figure 83: SBM – Z deviation of 70 different object positions in a plane and resulting tolerance of Z, α and β	75
Figure 84: SBM – α and β deviation of 70 different object positions in a plane.....	75
Figure 85: SBM-SOT – evaluation of the distance measurements between P0, P1, P2 and P3	76

Figure 86: SBM-SOT – Z deviation of 70 different object positions in a plane and resulting tolerance of Z, α and β	77
Figure 87: SBM-SOT – α and β deviation of 70 different object positions in a plane ...	77
Figure 88: SBM-ECT – distance evaluation of measurement plate	78
Figure 89: SBM-ECT – Z deviation of 70 different object positions in a plane and resulting tolerance of Z, α and β	79
Figure 90: SBM-ECT – α and β deviation of 70 different object positions in a plane....	79
Figure 91: visualization of the 3D CAD model projection of a good object detection....	80
Figure 92: inaccurate object detection due to an inaccurate camera calibration	81
Figure 93: inaccurate object detection due to an inaccurate camera calibration	81
Figure 94: Acquisition of the hole that represents the object origin of the shape model	82
Figure 95: a) resulting origin of shape model fitting (SBM-SOT); b) resulting hole center using ellipse fitting (SBM-ECT); c) comparison between SBM-SOT and SBM-ECT	82
Figure 96: comparison between the origin detection of the SBM fitting and the ellipse fitting of the hole in the origin with irregularities in the image	82
Figure 97: 3D ARTM-M – printed measurement plate with four markers with the identification patterns A, B, F and G	83
Figure 98: 3D ARTM-M – distance evaluation of measurement plate	84
Figure 99: 3D ARTM-M – Z deviation of 70 different object positions in a plane and resulting tolerance of Z, α and β	85
Figure 100: 3D ARTM-M – α and β deviation of 70 different object positions in a plane.....	85
Figure 101: 3D ARTM-B – distance evaluation of measurement plate.....	86
Figure 102: 3D ARTM-B – Z deviation of 70 different object positions in a plane and resulting tolerance of Z, α and β	87
Figure 103: 3D ARTM-B – α and β deviation of 70 different object positions in a plane.....	87
Figure 104: 3D QRCM – printed measurement plate with four markers with the identification patterns Code1, Code2, Code3 and Code4.....	88
Figure 105: 3D QRCM – distance evaluation of measurement plate.....	88
Figure 106: 3D QRCM – Z deviation of 70 different object positions in a plane and resulting tolerance of Z, α and β	89
Figure 107: 3D QRCM – α and β deviation of 70 different object positions in a plane.....	89
Figure 108: marker detection (fiducial marker) – definition of measurement plate	90
Figure 109: 3D FPM – distance evaluation of measurement plate.....	91
Figure 110: 3D FPM – Z deviation of 70 different object positions in a plane and resulting tolerance of Z, α and β	91
Figure 111: 3D FPM – α and β deviation of 70 different object positions in a plane.....	92

Figure 112: 3D FPM – Acquisition of the Fiducial Propeller marker at different heights (left: 300 mm, middle: 450 mm, right: 600 mm)	92
Figure 113: 3D FPM – Z deviation of 70 randomly distributed 3D FPM position measurements at a height of 300 mm (left) and 600 mm (right)	93
Figure 114: 3D FPM – α deviation of 70 randomly distributed 3D FPM position measurements at a height of 300 mm (left) and 600 mm (right)	93
Figure 115: 3D FPM – β deviation of 70 randomly distributed 3D FPM position measurements at a height of 300 mm (left) and 600 mm (right)	94
Figure 116: M2O-Map – The pose of the object in the middle is defined by the arrangement of four 3D Fiducial Propeller markers.....	95
Figure 117: M2O-Map – Transformations $miTo$ from OCS to $MCSi$	95
Figure 118: M2O-Map – Projected 3D position of three object detections via the M2O-Map at different camera positions	96
Figure 119: M2M-Map – This map is purely built with 3D Fiducial Propeller markers. The marker in the middle with ID 13 is considered to be the object.....	97
Figure 120: visualization of the UR-10 calibration plate dimensions (adapted from [47])	98
Figure 121: ΔX (left) and ΔY (left) distance measurement deviations between two neighboring calibration plate positions	99
Figure 122: Distance deviation of nominal and actual distance from a defined position on the calibration plate to the origin of the robot	99
Figure 123: RX (left), RY (middle) and RZ (right) deviation of the robot pose of 50 randomly chosen positions in a plane	100
Figure 124: RY and RZ deviation values sorted according to the X positions of the robot which was randomly moved in a plane	100
Figure 125: Development of the parameters of cTf with growing number of Hand-Eye calibration images.....	101
Figure 126: Visualization of the 100 different robot positions at which the static Fiducial Propeller measurement plate was detected	102
Figure 127: Detection and transformation to the RCS of four 3D FPM at 100 different robot positions (left); detailed visualization of the resulting 3D position of the marker with ID5 (right)	103
Figure 128: Development of the 3D pose of the marker with ID5 that was transformed to the RCS for each of the 100 robot positions.....	103
Figure 129: Visualization of the position based visual servoing approach to different translated, rotated and tilted M2O-Map target poses.....	105
Figure 130: Visualization of the position based visual servoing algorithm when moving towards the target pose defined by the M2O-Map	106
Figure 131: 10 object position detections at 15 offset positions represented in the RCS.....	106

Figure 132: Development of the object pose in the RCS with respect to the object Z offset	107
Figure 133: Development of the object pose tolerance in the RCS with respect to the object Z offset	108
Figure 134: Visualization of the position based visual servoing approach when moving towards the target object pose defined by the M2M-Map.....	109
Figure 135: Visualization of the linear extrapolation of the object detections (with respect to the RCS) performed at 60 uniformly distributed support points.....	110
Figure 136: TCP pose of the robot at the extrapolated target pose of the M2O-Map .	111
Figure 137: TCP pose of the robot at the extrapolated target pose of the M2M-Map .	111
Figure 138: Non-linear development of the parameters RX and RY for a certain test case	113
Figure 139: Distance deviation of the actual and the extrapolated target pose sorted according to the Y-axis of the RCS	116
Figure 140: Distance deviation of the actual and the extrapolated target pose	116
Figure 141: TCP at object pick/place position with opened gripper.....	118
Figure 142: M2M-Map (left), M2M-Map with object placed at the M2M-Map target position using extended visual servoing (middle), M2O-Map (right)	118
Figure 143: Initial starting pose of mobile platform (left) and mobile platform at the location for the object picking task (right)	119
Figure 144: Final pick position of the robot TCP with respect to the object	119
Figure 145: Change of the RCS between the pick and place operation by moving the mobile platform	119
Figure 146: Precise placement of object on two dowel pins at target location.....	120
Figure 147: Monocular calibration – visualization of calibration plate defects	121
Figure 148: Object detection inaccuracies due to bad illumination. The left image shows deviations of the SBM and SBM-SOT. The right image visualizes an inaccurate SBM-ECT.....	121
Figure 149: Drift of the 3D marker position that was transformed to RCS as the robot TCP moved to different locations in the shape of a grid	125
Figure 150: Projection of the current TCP and detected object pose to the image plane.....	125
Figure 151: Extrapolated TCP target pose of the M2O-Map (left) and the M2M-Map (right) with the use of 50 support points uniformly distributed between the object Z offset of 450 mm and 150 mm.	126
Figure 152: Precise placement of object on two dowel pins at the desired target location	127

10. List of Tables

Table 1: Accuracy results of the mobile manipulator “Little Helper” when using an additional high precision calibration plate to determine the relationship between the work cell and the robot [33].....	7
Table 2: Link parameters of two-link planar manipulator of Figure 29.....	28
Table 3: Geometric and non-geometric errors of the robot's kinematic chain [23].....	59
Table 4: Universal Robot UR10 – specifications [65].....	63
Table 5: Requirements for the camera observations	64
Table 6: Jai GO-5000-PGE - camera properties [35].....	64
Table 7: Monocular calibration error of the left and right camera	68
Table 8: Monocular 3D measurements in the calibration plane of the calibration plate corner points P1, P2, P3 and P4	68
Table 9: Resulting parameters of $c1Tc2$ with 60 different calibration images	69
Table 10: Binocular calibration – tolerance of detection repeatability for 70 distance measurements of a static calibration plate.....	70
Table 11: SBM – tolerance of static object detection repeatability for 70 measurements	74
Table 12: SBM-SOT – tolerance of static object detection repeatability for 70 measurements	76
Table 13: SBM-ECT – tolerance of static object detection repeatability for 70 measurements	78
Table 14: Pose results of an optimal object detection using an accurate camera calibration and a homogenous illumination of the object.....	80
Table 15: inaccurate object detection due to an inaccurate camera calibration	80
Table 16: inaccurate object detection due to an inaccurate camera calibration	81
Table 17: 3D ARTM-M – tolerance of static marker detection repeatability for 70 measurements	83
Table 18: 3D ARTM-B – tolerance of static marker detection repeatability for 70 measurements	86
Table 19: 3D QRCM – tolerance of static marker detection repeatability for 70 measurements	88
Table 20: 3D FPM – tolerance of static marker detection repeatability for 70 measurements	90
Table 21: 3D FPM – tolerance of static marker detection repeatability for 70 measurements	93
Table 22: 3D FPM – resulting tolerance of Z, α and β for 70 randomly distributed 3D FPM position measurements in a plane at different heights	94
Table 23: 3D FPM – resulting range of Z, α and β for 70 randomly distributed 3D FPM position measurements in a plane at different heights.....	94
Table 24: 3D FPM – Subpixel accuracy for 3D FPM detections at different heights	94
Table 25: M2O-Map – Transformations from OCS to CCS for each marker.....	96

Table 26: M2O-Map – Mean transformation cT_o of all markers.....	96
Table 27: M2O-Map – Repeatability of 70 detections of the transformation from <i>OCS</i> to <i>CCS</i>	96
Table 28: M2M-Map – Transformations from <i>OCS</i> to <i>CCS</i> for each marker.....	97
Table 29: M2M-Map – Mean transformation cT_o of all markers and mean measurement of cT_{m13}	97
Table 30: M2M-Map – Repeatability of 70 detections of the transformation from <i>OCS</i> to <i>CCS</i>	97
Table 31: X and Y distance measurements between two neighboring calibration plate positions	99
Table 32: Distance deviation of nominal and actual distances from a defined position on the calibration plate to the origin of the robot	99
Table 33: RX, RY and RZ deviation of the robot pose randomly moved in a plane	100
Table 34: Resulting parameters of cT_f with 60 different Hand-Eye calibration images	101
Table 35: Mean and maximal error of a complete chain transformation for each of the 60 different robot positions	102
Table 36: Resulting pose of the 100 transformations of the marker with ID5 to the RCS at different robot positions.....	104
Table 37: Defined pose offset threshold (POT) due to minimal robot increments.....	104
Table 38: RCS object pose repeatability measurement at the object Z offset of 200 mm	107
Table 39: RCS TCP pose repeatability measurement at the object Z offset of 200 mm.....	107
Table 40: Comparison between the actual and the extrapolated object target pose	111
Table 41: Changing number of extrapolation points for the movement range of 450 mm to 150 mm	112
Table 42: Changing number of extrapolation points for the movement range of 450 mm to 300 mm	112
Table 43: Changing number of extrapolation points for the movement range of 300 mm to 150 mm	112
Table 44: Changing number of extrapolation points for the movement range of 350 mm to 200 mm	113
Table 45: Actual and extrapolated values of RX and RY for a non-linear development.....	113
Table 46: Repeatability of the extrapolated TCP pose for 10 movements – 7 support points.....	114
Table 47: Repeatability of the extrapolated TCP pose for 10 movements – 15 support points.....	114
Table 48: Repeatability of the extrapolated TCP pose for 10 movements – 30 support points.....	114

Table 49: Deviation of 30 extrapolated TCP poses with respect to the manually determined actual object target pose.....	115
Table 50: Absolute XY position deviation between the actual and the extrapolated target pose	115
Table 51: Z deviation as a result of the multiplication of the XY deviation of the different absolute object positions with the scale factor $f_{Z/XY}$	115
Table 52: Absolute position and orientation deviation between the actual and the extrapolated target pose.....	116
Table 53: Accuracy evaluation of 3 different 3D object pose detection methods with respect to the measurement range and deviation along the Z-axis.....	122
Table 54: Pro and contra list of SBM-SOT and SBM-ECT	122
Table 55: Accuracy evaluation of 4 different 3D marker types	123
Table 56: Maximal deviation of the four object poses that were determined from each of the four 3D-FPMs in combination with the M2O-Map	124