University of Padova
Department of Information Engineering
Master Degree in

Control Systems Engineering

Master Thesis

Automatic multi-camera hand-eye calibration for robotic workcells

Supervisor: Prof. Stefano Ghidoni
Co-supervisor: Matteo Terreran, Ph.D
Daniele Evangelista, Ph.D

Candidate: Davide Allegro
ID n. 1236499

Academic Year 2021/2022
Date 11/04/2022

Abstract

Human-robot collaboration (HRC) is an increasingly successful research field, widely investigated for several industrial tasks. Collaborative robots can physically interact with humans in a shared environment and simultaneously guarantee an high human safety during all the working process. This can be achieved through a vision system equipped by a single or a multi camera system which can provide to the manipulator essential information about the surrounding workspace and human behavior, ensuring the collision avoidance with objects and human operators. However, in order to guarantee human safety and an excellent working system where the robot arm is aware about the surrounding environment and it can monitor operator motions, a reliable Hand-Eye calibration is needed. An additional improvement for a really safe human-robot collaboration scenario can be provided by a multi-camera hand-eye calibration. This process guarantees an improved human safety and give the robot a greater ability for collision avoidance, thanks to the presence of more sensors which ensures a constant and more reliable vision of the robot arm and its whole workspace.

This thesis is mainly focused on the development of an automatic multi-camera calibration method for robotic workcells, which guarantees ah high human safety and ensure a really accurate working system. In particular, the proposed method has two main properties. It is automatic, since it exploits the robot arm with a planar target attached on its end-effector to accomplish the image acquisition phase necessary for the calibration, which is generally realized with manual procedures. This approach allows to remove as much as possible the inaccurate human intervention and to speed up the whole calibration process. The second main feature is that our approach enables the calibration of a multi-camera system suitable for robotic workcells that are larger than those commonly considered in the literature.

Our multi-camera hand-eye calibration method was tested through several experiments with the Franka Emika Panda robot arm and with different sensors: Microsoft Kinect V2, Intel RealSense depth camera D455 and Intel RealSense LiDAR camera L515, in order to prove its flexibility and to test which are the hardware devices which allow to achieve the highest calibration accuracy. However, really accurate results are generally achieved through our method even in large robotic workcell where cameras are placed at a distance $d = 3\,\mathrm{m}$ from the robot arm, achieving a reprojection error even lower than 1 pixel with respect to other state-of-art methods which can not even guarantee a proper calibration at these distances. Moreover our method is compared against other single- and multi-camera calibration techniques and it was proved that the proposed calibration process achieves highest accuracy with respect to other methods found in literature, which are mainly focused on the calibration between a single camera and the robot arm.

Contents

1 Introduction **1**
- 1.1 Hand-eye calibration . . . 3
- 1.2 Automatic multi-camera hand-eye calibration . . . 5
- 1.3 Thesis outline . . . 6

2 Related work **9**
- 2.1 Calibration objects . . . 9
- 2.2 Eye-on-base calibration . . . 11

3 Camera models and calibration **15**
- 3.1 Pinhole camera model . . . 16
- 3.2 Intrinsic and extrinsic parameters . . . 19
- 3.3 Camera with lenses . . . 22
- 3.4 Camera calibration . . . 26

4 Proposed eye-on-base calibration method **31**
- 4.1 Single camera eye-on-base calibration . . . 32
- 4.2 Multi-camera eye-on-base calibration . . . 36
- 4.3 Filter function . . . 42

5 System setup **45**
- 5.1 Camera . . . 45
- 5.2 Robot arm . . . 47

6 Experimental results **49**
- 6.1 Simulated experiments . . . 50
  - 6.1.1 Analysis with different planar patterns . . . 51
  - 6.1.2 Analysis of the calibration method according to the distance between the camera and the robot arm . . . 55
- 6.2 Experiments in the Robot Vision Lab testbed . . . 60
  - 6.2.1 Analysis of the single and multi camera calibration with different sensors 61
  - 6.2.2 Analysis of the single-camera calibration with other state-of-art methods 63
  - 6.2.3 Analysis of the multi-camera calibration with other state-of-art methods 65
- 6.3 Image super resolution for large robotic workcells . . . 67

7 Conclusions **69**

Bibliography **71**

List of Figures

1.1 *Typical Human-robot collaboration workcell, where an operator collaborates, in a shared workspace, with a cobot equipped by a vision sensor [8].* 1
1.2 *Hand-Eye Calibration setups, collected by [33]. At the left the typical eye-in-hand calibration setup. At the right the eye-on-base calibration setup with a fixed external camera.* 4
1.3 *Example of workcell setup adopted for the calibration in this thesis. Image acquired by the virtual environment Gazebo.* 6

2.1 *Examples of planar calibration patterns.* 10
2.2 *Sphere center detection in a camera network by [50].* 10
2.3 *Ideal workcell setup proposed in [32].* 12
2.4 *Workcell setup for camera calibration adopted in [45].* 13
2.5 *Multi camera calibration setup proposed in [59].* 13

3.1 *Comparison between the two most typical camera models.* 15
3.2 *Pinhole camera model structure, where a 3D object is projected on the image plane, at the opposite side of the camera.* 16
3.3 *Perspective geometry of pinhole camera model.* 17
3.4 *Perspective geometry of a 3D point projected on Y and X axis.* 18
3.5 *Projection of point $p = (x, y)$ on the virtual image plane $\mathcal{V}$.* 18
3.6 *Perspective projection of Pinhole camera model, considering a principal point p^* at coordinates (c_x, c_y) of image plane.* 20
3.7 *Perspective projection of Pinhole camera model, considering a principal point p^* at coordinates (u_0, v_0) of image plane and a camera at R,t of world coordinates.* 21
3.8 *Extrinsic parameters transform the 3D point from the world coordinate to camera coordinates. Intrinsic parameters maps the 3D point from the camera coordinates to the image plane.* 22
3.9 *Perspective projection of thin lens model.* 23
3.10 *Lens can gather more light rays coming from the objects, but only a set of points at a specific distance are in focus, the others are out of focus and then creates on the image plane a circle of confusion.* 23
3.11 *Camera FoV is strictly related to the size of film and focal length of the lens.* . . 24
3.12 *Comparison among the original image at the left, its barrel distortion and at the right the pincushion distortion.* 25

3.13 *Tangential distortion of the image plane where the main issue is referred to the skewness of the image plane.* . . . 25
3.14 *Checkerboard with 4x7 corners and with known geometry. The distance between each corner on the x axis is equal to s_x, the distance on the y axis is s_y.* . . . 27
3.15 *Example of correspondences (m_i, M_i) between the image plane and the target plane in the 3D world for a checkerboard.* . . . 28

4.1 *Setup examples of a multi camera system and a robot arm for eye-on-base calibration for robotic workcells.* . . . 31
4.2 *The whole proposed eye-on-base calibration process. It is divided in three subprocesses, one dedicated to the image acquisition, one related to the robot motion and the other dedicated to the optimization process of the final calibration.* . . . 32
4.3 *Setup with a single camera and the robot arm with a checkerboard mounted on its end-effector, where are denoted the transformations between all reference frames.* . . . 33
4.4 *The whole process for multi-camera eye-on-base calibration. It is divided in three subprocesses, one dedicated to the image acquisition by all cameras, one related to the robot motion and the other dedicated to the optimization process for the robotic workcell calibration.* . . . 36
4.5 *State machine implemented in order to handle the image acquisition phase.* . . . 37
4.6 *Setup with a camera network of three cameras and the robot arm with a checkerboard mounted on its end-effector, where are denoted the transformations between all reference frames.* . . . 39
4.7 *Example of false positive, namely wrong checkerboard corners extraction.* . . . 42
4.8 *Wrong corners detection due to the high distance between camera and checkerboard. The control points are almost aligned even if they are wrongly detected.* 43

5.1 *Cameras tested on the thesis experiments.* . . . 46
5.2 *Franka Emika Panda robor arm adopted for thesis experiments.* . . . 47
5.3 *Custom 3D printed mount, composed by two elements necessary to attach checkerboard on the Panda robot arm.* . . . 48
5.4 *Custom 3D printed mount adopted for to attach the checkerboard to the robot arm.* . . . 48

6.1 *Example of two different workcell setups of our robotic workcell in Robot Vision Lab and in the simulator Gazebo.* . . . 49
6.2 *Robotic workcell setup designed for the experiments on simulator Gazebo.* . . . 50
6.3 *Planar patterns analyzed.* . . . 51
6.4 *Robotic workcell setup adopted to perform the experiments with different planar patterns.* . . . 52
6.5 *Simulated environment adopted with the three calibration patterns analyzed.* . . . 52
6.6 *Average results obtained with the three different planar patterns describe above.* 54
6.7 *Schematic robotic workcell arrangement for experiments related to the distance.* 55

6.8 *Workcell arrangement of the first experiment at the left and arrangement of the last experiments at the right.* 56
6.9 *Deviations of estimated transformation matrices from their real values, achieved achieved by the single and multi camera calibration on the first four graphs. Reprojection error related to the single and multi camera calibration depending on the distance d between the camera and the robot arm on the last two graphs.* 58
6.10 *Reprojected corners on the same image adopting the transformation matrices determined through the single and multi camera calibration at distance* $d = 3\,m$. 59
6.11 *Transformation accuracy achieved by the single and multi calibration methods, computing the deviation mean between the estimated transformation matrix* $T_{C_i}^{C_j}$ *and their real values at each distance d.* 59
6.12 *Robotic workcell designed in Robot Vision Lab; in particular in this image there are three depth sensors D455 with a robot arm in the center of the workcell with the chessboard attached on the end-effector.* 60
6.13 *Analysis of the calibration accuracy achieved by each sensor according to the distance through the single and the multi camera calibration method.* 62
6.14 *Checkerboard images captured by the Kinect and LiDAR camera from the same distance* $d = 1.2\,m$. 63
6.15 *Robotic workcell setup adopted to perform the comparison among different methods of single camera hand-eye calibration method.* 64
6.16 *Two adopted setups for multi-camera calibration where the camera networks are arranged in the same setup and sizes.* 65
6.17 *Calibration accuracy in terms of re-projection error obtained by using orginal images and resized images, according to the distance d.* 68

Chapter 1

Introduction

The fast growing demand of robot applications in industrial environment have required the development of several well-performing robots equipped by advanced control and vision systems. Such robots have the main aim of helping human to accomplish different tasks in collaborative scenarios. The term **Human–Robot Collaboration** (HRC) refers to the study of collaborative processes where human and robot work together to achieve a common goal, with the main purpose of reducing the workloads of human operators and speeding up the working process. This research field is getting increasingly wide, involving most of the industrial production areas such as automotive industries [1, 2], manufacturing industries [3] and mining industries [4], but even outside the industrial environment such as agricultural production [5] and medical and surgical applications, as described in [6, 7].
Industrial applications of Human-Robot Collaboration involve **collaborative robots**, also known as cobots, which can physically interact with humans in a shared workspace. This kind of robots are adopted to carry out simple and ripetitive tasks such as pick and place, as well as more complex works such as assembling mechanical objects, welding small pieces together and drilling screws, in order to release human workers from tricky tasks.

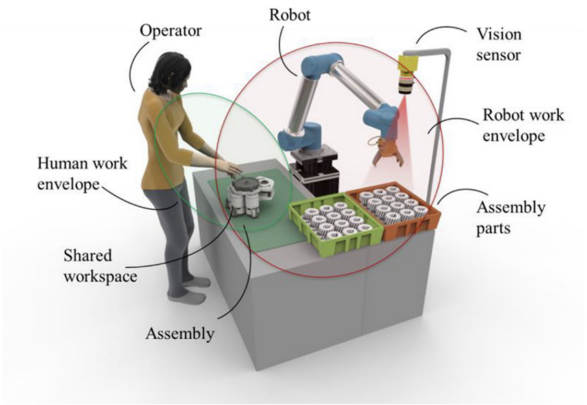


Figure 1.1: *Typical Human-robot collaboration workcell, where an operator collaborates, in a shared workspace, with a cobot equipped by a vision sensor [8].*

Cobots differ from classical industrial robots[1] for their capability of work alongside humans. This is allowed by advanced sensor systems which acquire data from the surrounding environment and from the operator, with two different methods: image based approach and non-image based approach.

- *Image-based approach*: the information acquisition is computed through the analysis and the elaboration of images, which can be captured by specific sensors such as RGB-D camera, which is a specific type of depth sensing devices that are combined with a RGB camera, that are able to augment the conventional image with depth information [9], or a stereo camera, which is a type of camera with two or more lenses with a separate image sensor, allowing the camera to capture three-dimensional images[2] [10].

- *Non image-based approach*: they were used generally to acquire data and recognize gestures and actions from an operator closed to the robot through wearable devices. Some of these system are special gloves used for gesture recognition [11] and sensors relying on a wristband as Myo [12].

All these methods are necessary to analyze the robot workspace and the operator behavior, to allow the robot to recognize some human gestures, identifying unexpected user motion and localize objects and persons in the workspace.
In last years, cobots have been widely deployed in several industrial areas such as manufacturing [13] and pharmaceutical industries [14] and even for different medical tasks [15, 16] thanks to the improvement of their properties:

- *safety*, since collaborative robots are able to safely work alongside human workers to complete a task. They are designed to minimize the risk of accidents and injuries in the workspace with other humans, thanks to the presence of specific sensors adopted for forces limitations, current overload prevention and blocking systems in case of unexpected contact;

- *flexibility*, since cobots can be easily programmed. Moreover, thanks to their small size, they can be moved more easily across the factory without changing the production environment and then they can be adapted from one production to another quickly and cheaply;

- *low-cost*, since the simplicity of their programming reduces the time for their integration and moreover they don't need any dedicated safety cell, meanly any fences or other industrial safety equipment are required. These features contribute to decrease significantly the cost of their deployment.

Safety is often enhanced adopting camera systems which allow to visually and precisely guide the robot to its goal, as it is described in [17, 18]. In particular it is interesting the case study proposed in [19], where a deep learning algorithm analyzes images captured by a camera to recognize the human operator that collaborates with the robot. Then, the human working ability is analyzed and is used by the robot to adapt its operation speed according

[1] Industrial robots don't the ability to work alongside humans. They work fast in pre-defined patterns and lack awareness about humans.
[2] This process is known as stereo photography.

to the worker skills or deliver parts to be manipulated according to the handedness of the human worker.
Most recent depth sensors, thanks to an increasingly hardware improvement, allow to enhance the performance of HRC for many applications, especially for industrial tasks. In particular camera systems allow the collaborative robots to supervise the surrounding workspace, recognize human actions, avoid unexpected motion with consequently human injuries with the main purpose of ensuring high human safety. Moreover, in order to guarantee a safe and reliable operating system, cameras have to precisely localize the robot within the workcell, in order to establish accurately where is placed an object or a person with respect the robot.

1.1 Hand-eye calibration

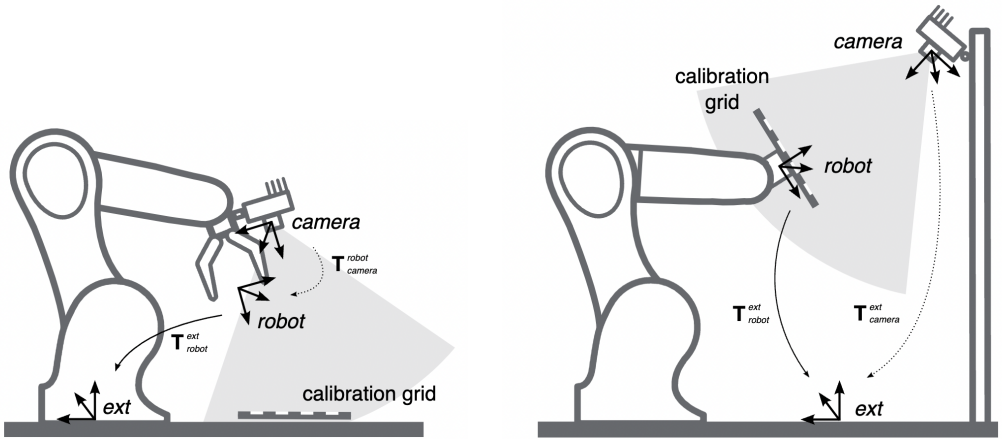
In order to accurately establish the camera position with respect to a fixed world coordinate system (e.g. the robot's base), a robust camera calibration is strongly required. **Camera Calibration** is a primary task in computer vision and it is defined as the process of determining the geometric parameters of the camera model, namely the function parameters which maps the 3-dimensional world onto a 2-dimensional plane and defines the camera 3D geometric coordinate with respect to a world reference frame. In particular, this process is generally divided in two procedures, the extrinsic [20] and the intrinsic calibration [21]. The extrinsic parameters are necessary to the transformation from world coordinate frame to the camera coordinate frame. Instead the intrinsic parameters map a 3D point with respect to the camera frame into the 2D point on the image plane. The whole process can be considered as an optimization procedure where the discrepancy between the observed image features and their theoretical positions is minimized according to camera's parameters [22], which are described accurately in section 3.2. In order to accomplish the optimization, identifying an accurate relation between 3D points in real world and their correspondent 2D projected points on the image plane is required. These correspondences can be obtained capturing more images from different perspective of a calibration pattern, whose 3D geometry is well known, such as a planar checkerboard, as explained in [23].
When the vision system is composed by more cameras, a multiple camera calibration is needed. The term **Multi-Camera Calibration** refers to the the calibration of more than one sensor simultaneously as it is proposed in [24], which deals with an accurate global calibration of multiple cameras with non-overlapping fields of view. However, in a camera network setup with overlapping fields of view, it can be exploited calibration pattern images seen at the same moment from multiple cameras as it is proposed in [25]. In both scenarios, the main purpose is determining the accurate locations of each camera with respect to the others, and with respect to a single world reference frame. In particular, in many camera calibration approaches, a manual procedure is required, in order to move the checkerboard in several positions and perspectives in front of the cameras and then to capture the related images, as it is described in [25, 26, 27, 28].
A very good option to automatize this calibration process is the usage of a robot arm which can move the checkerboard in different perspectives in front of the camera, in order to avoid

human intervention that can lead to inaccuracy errors. This solution can be accomplished in a robotic workcell, as it is illustrated in Figure 1.2b, that shows a camera system and a robot arm, where a calibration between the robot and the camera is required. The calibration procedure which is focused on determining the transformation between a robot base and a camera or between the end-effector and the sensor, is generally known as **Hand-Eye Calibration**.
There are two different setups for Hand-Eye calibration:

- **Eye-in-Hand calibration**, where a camera is mounted on the robot arm, as shown in Figure 1.2a. It is the hand-eye calibration approach most used, as it is proposed in [29, 30, 31]. A calibration pattern is placed on the robot's workspace and then moving the robot arm in different poses, the camera can capture one image of the pattern for each pose. Hence the calibration process determines the camera's position on the robot frame as well as the position of the calibration pattern in the robot's workspace. In particular this approach refers to the concurrence of the eye's reference frame, so the camera's frame to the hand frame, namely the end-effector reference system.

- **Eye-on-Base calibration**, where it is used a fixed external camera around the robot environment, as shown in Figure 1.2b, and described in [32]. In order to do this, a calibration pattern is attached on the robot arm's end-effector and it is moved to different poses in front of the camera. By capturing an image for each pose, the calibration can be computed and then the camera's pose and the pose of the calibration pattern can be estimated with respect to the robot's end-effector. In this event, the eye reference frame (namely, the camera frame) is fixed with respect to the reference frame of the robot's base.

In both calibration setups the aim is the identification of the transformation between the *Hand*, namely the robot base or the end-effector and the *Eye* which refers to the camera.



(a) *Camera on the robot arm.* (b) *External camera from the robot arm.*

Figure 1.2: *Hand-Eye Calibration setups, collected by [33]. At the left the typical eye-in-hand calibration setup. At the right the eye-on-base calibration setup with a fixed external camera.*

In this thesis, as it is described in Section 1.2, we focused on the Eye-on-base calibration approach. In particular, fixed external cameras and a robot arm with a checkerboard attached to its end-effector is the calibration setup adopted for thesis experiments. The usage of a camera network allows us to implement a robust robotic workcell calibration, that can be used for people tracking in human-robot collaboration tasks, avoiding persons occlusion issues, where an operator can not be observed and detected by a single camera, since he is partially covered by the robot arm. The ability to never lose the human tracking is a main requirement for human-robot collaboration tasks, necessary to guarantee the human safety avoiding unexpected collisions with the robot arm and to ensure the people tracking correctness. The robot arm allows to replace the classical manual procedure for images acquisition and automatize the whole procedure, making it more precise. Moreover an accurate hand eye calibration ensures an accurate visually guidance of the robot, allowing an excellent working systems, where the manipulator can properly help humans in their tasks.

1.2 Automatic multi-camera hand-eye calibration

As described in section 1, in Human-Robot Collaboration tasks the robot should be capable of recognizing its position with respect to the operator is working with, for the whole time in which the collaboration takes place, guaranteeing the human safety. This is made possible by a camera network accurately calibrated with respect to the robot arm. A common solution previously adopted in our research laboratory IAS-Lab[3], is proposed in OpenPTrack[4] [25]. It involves the use of a camera network in order to detect when users are inside the robot workspace. The adopted images acquisition procedure is completely accomplished by a long manual process, where a human intervention is required in order to move the calibration pattern in front of cameras in several perspectives. The main drawbacks of this process is the time necessary to compute the whole images acquisition and the inaccuracy of the calibration due to the human intervention.
This procedure was the main starting point for this thesis. In particular, according to the issues and considerations just described, the thesis objective was to automatize and speed up such procedure in order to improve precision and ease of use. More precisely, we focus on proposing an **automatic hand-eye calibration for robotic workcells**, which can be adopted for Human-Robot collaboration tasks.
In this thesis we proposed a novel calibration method to achieve an accurate extrinsic[5] calibration of a multi-camera setup with respect to a robot arm positioned in the center of a workcell, as shown in Figure 1.3. Regarding the intrinsic[6] calibration, it was considered already computed, because accurate camera intrinsic parameters were already provided by the cameras factory.
Since we exploited the robot arm to automatize the image acquisition process and make

[3]IAS-Lab stands for Intelligent Autonomous Systems Laboratory and it is one of the 28 laboratories of the Department of Information Engineering of the University of Padua.

[4]http://openptrack.org/.

[5]It is focused on determining the geometric transformation between the cameras coordinates frame and the robot's base frame.

[6]They are the camera parameters which relate the camera's coordinate system to the image plane of the camera.

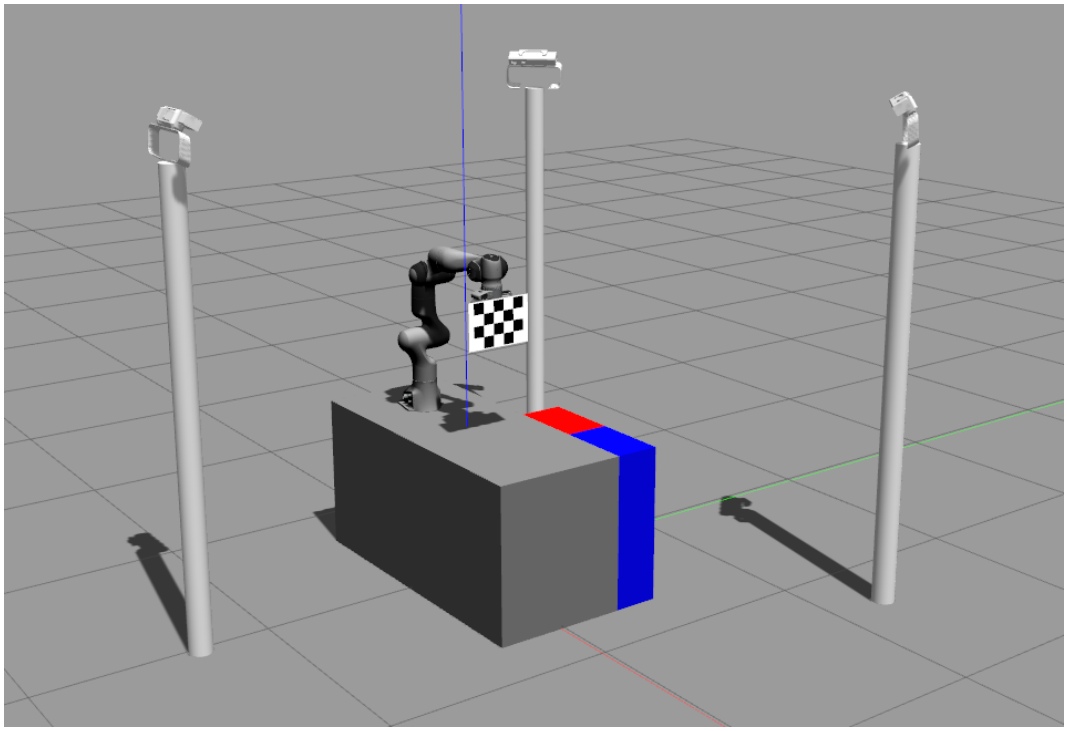


Figure 1.3: *Example of workcell setup adopted for the calibration in this thesis. Image acquired by the virtual environment Gazebo.*

it repeatable, in order to compute the hand-eye calibration an A4 size printed calibration target was adopted. First of all, this was done to facilitate the small robot arm movements in the image capture phase and avoid collisions. Furthermore, it allowed to simplify the attachment to the robot, accomplished by a custom 3D-printed mount. However, the main drawback of this implementation was the difficulty of the planar target detection, due to its small size and the its high distance from the camera.

1.3 Thesis outline

The remainder of the thesis is organized as follows: in Chapter 2 some related works, which concerns with different calibration objects adopted for camera calibration. Moreover most recent papers, which are focused on hand-eye calibration in a single and multi camera setups, are introduced. In Chapter 3 the theory behind the camera calibration is reported, with an initial description of the pinhole camera model and then with an introduction of the mathematics related to intrinsic and extrinsic camera calibration. Hence, in Chapter 4 the proposed method for the camera calibration is accurately described. Especially, in the first section it is analyzed the approach for a single camera calibration and then it is reported the calibration technique for a multi-camera setup. In Chapter 5 the setup adopted for the experiments is described. More precisely, the hardware specifications of the cameras and the robot arm adopted for the whole thesis work, are presented. Then, in Chapter 6 all the experiments and achieved results are accurately reported and discussed, reviewing the

difference with other state of art calibration methods and evaluating the performance of our proposed calibration technique. The results are reported in two small subsection, the first one which concerns with simulated experiments and the second one which is focused on the real experiments. Then, in the final Chapter 7 a short review of the proposed method is reported, by highlighting the main contributions to the state of art. Finally some future improvements and developments are presented.

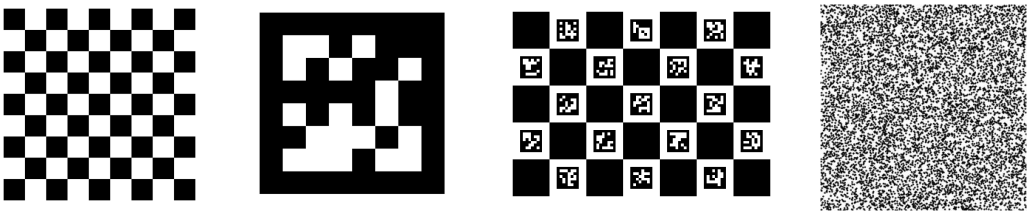
Chapter 2

Related work

Cameras calibration is generally divided into two phases: intrinsic and extrinsic calibration as further explained in Chapter 3. These procedures can be carried out by adopting different calibration targets, which observed from different perspectives by the camera allows its calibration.
However a key role is played by the approach adopted for the images acquisition. A manual procedure was often used to move the calibration objects such as in [27, 28]. Even though with the recent spread of human-robot collaboration setups, almost completely automatized procedures by robot arms were presented, in order to avoid the human intervention, as it is proposed in [34, 35, 36].
In the following two subsections, firstly some papers that analyze mostly adopted objects for the cameras calibration are illustrated and secondly works focused on procedures employed for hand-eye calibration for robotic workcells are analyzed.

2.1 Calibration objects

The extrinsic and intrinsic camera calibrations are performed by observing a particular object whose geometry in 3-D space is known with very good precision. Therefore several studies related to the camera calibration process were focused on testing the reliability of different kinds of patterns.
Planar targets are the most-commonly objects used for camera calibration. One of these, is the classical squared checkerboard shown in Figure 2.1a, as described in several papers such as in [37, 38, 39]. It was one of the first planar patterns used for camera calibration as described by Zhang's method in [40] and it is still adopted for its geometric simplicity and for its detection accuracy. Therefore it is used both for extrinsic calibration and intrinsic parameters estimation, as in [37].
In [41] and [42], ArUco and AprilTag markers were proposed respectively, for extrinsic camera calibration. They guarantee an accurate camera calibration and a great adaptability to camera models, moreover they allow to estimate the camera pose in tag coordinate system as long as there is one complete tag in the field of view. However, AprilTag markers shown in Figure 2.1b appeared robust for camera calibration, especially when they are placed relatively closed to the camera as described in [43].



(a) *Chessboard* (b) *AprilTag marker* (c) *ChArUco board* (d) *Random speckle pattern*

Figure 2.1: *Examples of planar calibration patterns.*

A special checkerboard called ChArUco is also adopted for camera extrinsic [44] and intrinsic calibration [45]. This planar board, shown in Figure 2.1c, tries to combine the benefits of the markers and the simplicity and accuracy of the checkerboard. ArUco markers are used to interpolate the position of the checkerboard corners, so that it has the versatility of marker boards, since it enables a camera calibration without necessarily being entirely within the camera's field of view [46], as long as there is one complete marker in the camera's field of view. Moreover, the checkerboard corners allow to achieve accurate calibration, in terms of subpixel accuracy [38].
Further planar targets were tested in literature such as polygonal boards [47], ring calibration pattern [48] and random-speckle pattern shown in Figure 2.1d. As it is accurately explained in [49], the random-speckle pattern has the main advantage of being characterized by many effective control points defined on the simulated speckle pattern, instead of the classical checkerboards which count at most some tens of control points. However one main limitation is that the calibration accuracy may significantly decrease when very large resolution difference exists between the simulated speckle pattern and the captured calibration images.
Nevertheless, even 3D objects with a depth variation are used for camera calibration, especially to calibrate depth sensor. Spheres is a proposed 3D object adopted for the calibration in [50, 51]. Its uniform and well known geometry allowed to overcome the problem of limited features scene shared by sensors, since the sphere center's location can be robustly estimated from any viewpoint, as it is clearly illustrated in Figure 2.2.

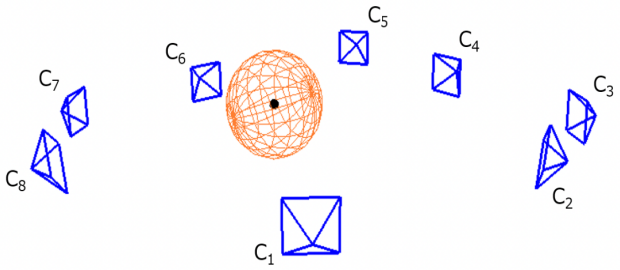


Figure 2.2: *Sphere center detection in a camera network by [50].*

The 3D locations of the center of a moving sphere are estimated from the color and depth images and they are used as visual correspondences across different camera views for estimate the relative poses among sensors. Although the sphere center is not directly observed by any sensor, the spherical constraint implies that the observation of a reasonably-sized surface patch from any perspective can be used to estimate the center location.
However, the use of a specific target over another one, is closely related to its robustness and its detection accuracy, but also to the environment in which it is tested. According to the hardware setup adopted and the aim of this thesis work, a detailed analysis was computed employing planar patterns that can be easily attached to the robot end-effector. They are listed and described more accurately in section 6 related to the experiments.

2.2 Eye-on-base calibration

As mentioned in section 1, the calibration procedure proposed in OpenPTrack [25], has been previously adopted in our research laboratory IASLab, for the multi-camera calibration in a RGB-D camera networks. The authors presented a calibration that through the use of a checkerboard, jointly optimize and estimate the pose of all sensors at once. When all cameras have been calibrated, the user had to place the checkerboard on the ground, in order to guarantee that the position of the ground plane with respect to each camera was calibrated. The main drawback of this procedure is the human intervention requirement, necessary to move the checkerboard in several poses for images acquisition. This leads to a slow-down of the whole calibration process as well as its accuracy reduction. Moreover a big checkerboard is required to ensure acceptable calibration performance, which yields the human task more complex and uncomfortable, as it is difficult to move an 80x80cm checkerboard and keep it still during the acquisition of each image.
The robot arm's presence in the multi-camera calibration for robotic workcells, allowed us to exploit the robot arm with a planar target attached to the end-effector to automatize the image acquisition process and replace the inaccurate human intervention.
In literature several robot hand-eye calibration were proposed, where the main purpose is the estimation of the relative pose between a camera frame and robot base frame. Some of these systems are accurately described in [15, 32, 52, 53, 54, 55]. In [32] authors proposed an automatic single camera eye-to-hand calibration for a small workcell, such as the one shown in Figure 2.3. They used a standard checkerboard calibration grid attached on the manipulator's tool in order to automatize and then speed up camera-to-robot calibrations.
In particular, they proposed a single camera calibration procedure, whose optimization process has to be computed once for each sensor. It consists in estimating a 3D affine transformation matrix between the camera and the robot end-effector, using the corresponding 3D corner points of the calibration checkerboard. Then, adopting the robot kinematics, the 3D points detected in the camera color image can be transformed from the camera coordinate system to the robot's base coordinate system. Additionaly they tested their optimization process according to the number of positions of the checkerboard used for each calibration procedure. According to experiment results shown in [32], in order to achieve good calibration accuracy, the calibration pattern has to cover at each pose the most of the camera's field

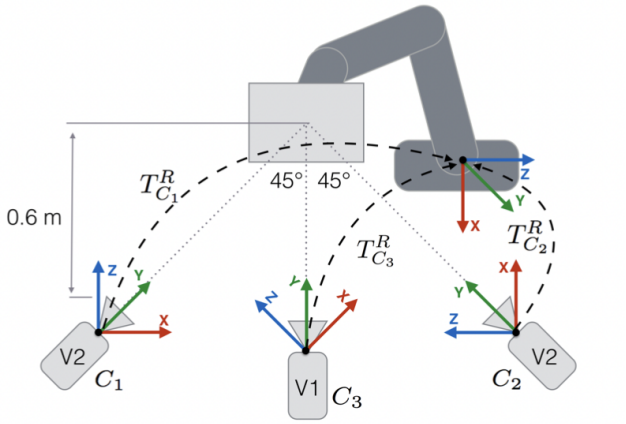


Figure 2.3: *Ideal workcell setup proposed in [32].*

of view. Therefore, since the robot arm has a limited action range, the cameras required to be placed closely to the manipulator, as it is illustrated in Figure 2.3, where the cameras are placed in front of the checkerboard at a distance of 60 cm. Although at this distance an accurate calibration can be achieved, it do not allow a robotic workcell calibration for human-robot collaboration tasks.

In [52], it is described a new technique for computing 3D position and orientation of a camera relative to the last joint of a robot manipulator in eye-on-hand configuration, but it can be easily inverted in a eye-on-base system.

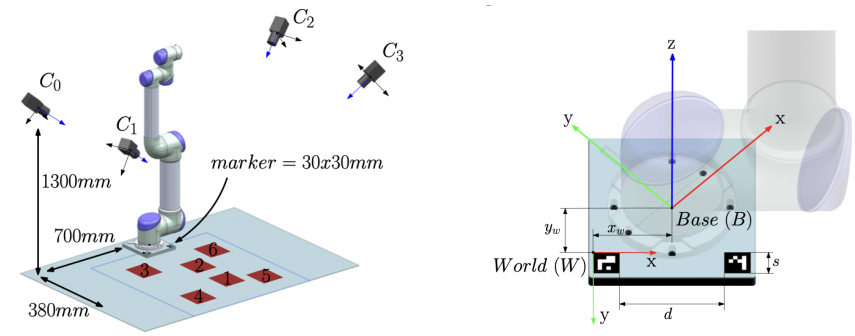
In [53] it is introduced a novel hand-eye calibration method which allows to determine the hand-eye transformation using known relative movements of the robot and the data acquired by the camera, by taking into account the unity constraint of dual quaternions necessary to represent rigid motions in space.

In the article [54] it is proposed an online hand-eye calibration method which allows to reduce the human supervision compared with classical calibration methods. Its derivation includes a new linear formulation of classical hand-eye calibration which is linear and numerically efficient.

Frank Park in its paper [55] proposed a method for solving the linear system $AX = XB$ for hand-eye optimization. The main innovation of this algorithm is the usage of the knowledge of Lie group theory [56] to solve the classical equations of hand-eye calibration.

Further research papers introduced extended optimization approaches with a simultaneous multi-camera calibration for a robot workcell. In [45] is presented an automated calibration technique to enable a vision-based robot control with a multi-camera setup depicted in Figure 3.1a. In order to find the 3D poses of cameras, the Perspective-n-Point[1] (PnP) problem is solved with ArUco markers. Hence, in order to estimate the relative pose between the robot base and camera, the predefined 3D geometric transformation from World frame to Base frame is adopted, as reported in Figure 3.1b.

[1]The problem of estimating the pose of a calibrated camera given a set of n 3D points in the world and their corresponding 2D projections in the image. Possible solutions to this problem are formulated in [57, 58].



(a) *Camera network setup* (b) *Robot base with markers*

Figure 2.4: *Workcell setup for camera calibration adopted in [45].*

However, since the ArUco markers have to be rigidly attached to the small robot base as shown in Figure 3.1b, they has to be fairly small, in the range of 2-3 cm^2. These markers speed up certainly the extrinsic calibration procedure, even though their small size increase the possibility to fail theirs detection, as the distance among cameras and robot arm enlarges.

Furthermore, in [59] a generalization of the Zhang's single camera calibration algorithm [40] for the case of N cameras with a robot arm and a checkerboard is proposed. The calibration procedure starts with a single camera calibration using Zhang's algorithm, in order to get an initial estimate of each camera's intrinsic parameters and the rigid transformation between camera and world reference frame. Then, exploiting the camera's overlapping field of view and the checkerboard simultaneously detected by all sensors, the estimation of each camera's position and orientation relative to a reference camera is computed. It could be done, by minimizing the reprojection error[2] using the Levenberg-Marquardt algorithm [60]. As it is mentioned, the big squared checkerboard is attached to the manipulator's tool, as shown in Figure 2.5, and it is moved in several perspectives for capturing the images. Thus, the robot arm is mainly adopted for controlling the checkerboard movements and solve the repeatability issue of manual calibration.

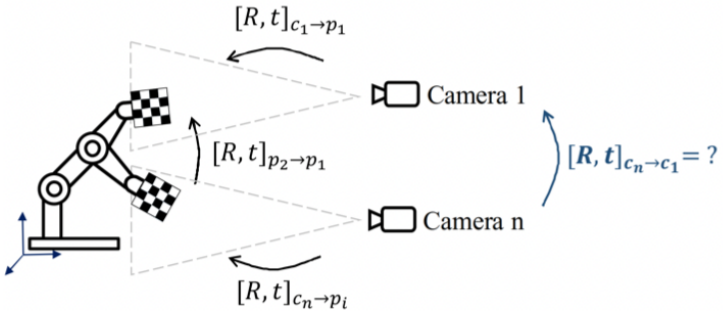


Figure 2.5: *Multi camera calibration setup proposed in [59].*

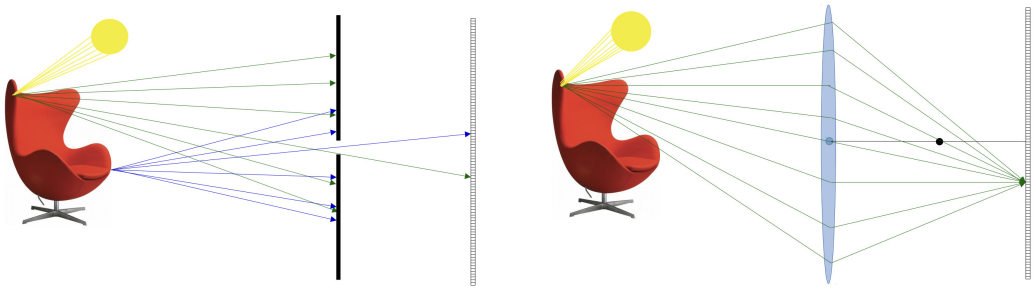
[2]The reprojection error corresponds to the Euclidean distance between detected image feature points and reprojected world points, obtained through the principles of projective geometry.

Since the checkerboard has to fill at least 20% of the image frame to guarantee acceptable calibration accuracy, the main requirement of the proposed calibration process is the closeness between camera and robot arm, in the range of $60 - 80\,\text{cm}$.
All the previous described calibration procedures, concern with multi-camera setup in a small workcell, namely the sensors are closely positioned to the robot arm. This setup allowed to overcome the distance issue and to avoid to fail markers or corners detection, however at the same time it makes difficult the implementation of such methods in a larger cell in which there may also be a human.
According to described related works and especially to the aim of developing a camera calibration process for human-robot collaboration tasks, in this thesis we illustrate a robust calibration procedure which can be applied in a robotic workcell with big-medium sizes, while maintaining very good calibration accuracy, even though a smaller planar pattern is adopted. According to this thesis aim the main contribution to the state-of-art is overcoming the issue of having a checkerboard that does not cover a large part of camera's field of view and achieving really accurate calibration results, even in a network of cameras 4-5 meters away from the robot.
In particular in section 4 we describes more precisley the approach adopted for cameras calibration and how the robot arm contribute to the whole optimization process.

Chapter 3

Camera models and calibration

Before defining the camera calibration procedure, firstly it is necessary to understand the image formation process and the camera model structure. The image formation consists in the radiometric and geometric process by which a 3D object is mapped into a 2D image plane [61]. In particular, vision starts with the detection of the light. The light is issued by a source, such as the sun, by rays and it strikes objects which are in its direction of emission. When the light rays strike an object, most of the light is absorbed and the remainder, which is not absorbed, is reflected by the object and it is perceived as its color by appropriate photoreceptors in our eyes [62]. Thus, allowing the object detection and the formation of its image in our eyes, in particular in our retina, which is a photosensitive surface composed by the photoreceptor cells: *cones* and *rods* [63]. This process can be accomplished by a camera as well, whose structure is able to capture the scattered light from an illuminated object and then to collect and focus the light on a dedicated film to create the image. Typically a thin lens[1] or a camera's aperture[2] and the image sensor/film[3] are the main elements necessary to create a **camera model**, as it is schematically shown in Figure 3.1.



(a) *Pinhole camera model allows the transition of a small amount of light rays coming from an illuminated object to strike the image sensor.*

(b) *Thin lens camera model gathers a bigger amount of light coming from an illuminated object and focuses it on the image sensor.*

Figure 3.1: *Comparison between the two most typical camera models.*

[1] A lens is a transmitting optical device which focuses or disperses a light beam by means of refraction.
[2] An aperture is a hole or an opening through which light travels.
[3] Device that allows the camera to convert photons (light) into electrical signals that can be interpreted by the device [64].

Pinhole camera model and **camera with thin lens** are the camera models illustrated in Figure 3.1, they are both structured in order to be able to absorb light coming from an objects and then project it on a film, commonly known as image plane.
In the following subsections it is described accurately the structure of the pinhole camera model and its perspective projections 3.1, the intrinsic and extrinsic parameters 3.2, camera lens distortion 3.3 and then the camera calibration process 3.4.

3.1 Pinhole camera model

The **pinhole** is the simplest camera model structure, which consists in a simple lightproof box without any lens and with a single tiny hole in the center of one of its sides, and a light-sensitive film paper positioned inside the box on the side facing this pinhole. In this camera model only a small amount of light rays, coming from an illuminated object, can pass through the pinhole to project the inverted image on the opposite box side, where a sensing film is placed (Figure 3.2). Moreover, in order to get a fairly sharp image, each point on the surface of an object needs to be projected to one single point on the film. This can only happen if the pinhole is really reduced to a point (which is physically impossible), because exactly one light ray would pass through the pinhole from any scene point to the image plane. Hence, the camera's aperture defines how much light can pass through it, then the size of the pinhole plays a key role in image formation. The smaller the pinhole, the sharper the image. However a small hole means that the amount of light passing through the hole and striking the surface of the film decreases. Therefore a small pinhole requires a longer exposure time[4], which, for real pinhole cameras, increases the risk of producing a blurred image if the camera is not perfectly still.

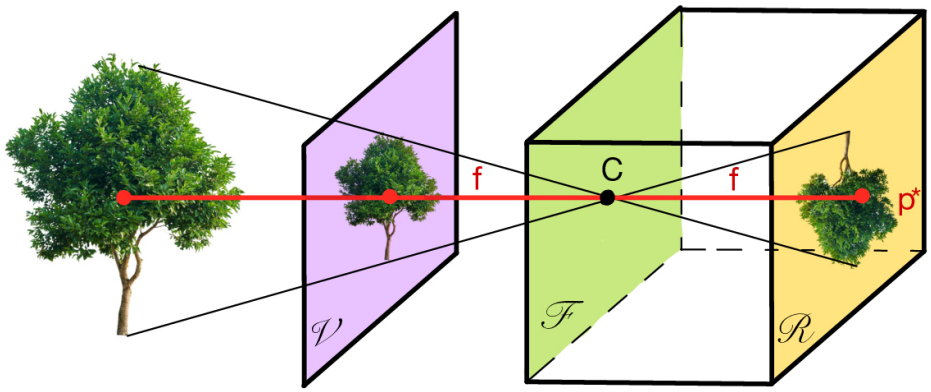


Figure 3.2: *Pinhole camera model structure, where a 3D object is projected on the image plane, at the opposite side of the camera.*

Although, this camera model presents these limitations, which are partly overcome by the thin lens addition on the camera model (Section 3.3), the pinhole perspective projection is mathematically convenient to analyze the image formation process because it provides an acceptable approximation of the mapping between 3D points and their projection onto 2D

[4]Exposure time is the length of time that the film inside the camera is exposed to light.

coordinates. In Figure 3.2 is illustrated the structure of the pinhole camera model and the following elements can be highlighted:

- the **pinhole** C, is called center of projection and it defines the aperture of the camera;
- the **image plane** $\mathcal{R}$ is the yellow plane in which is projected the inverted image;
- the **optical axis** coincides to the red line passing through the center C and perpendicular to the image plane, whose intersection is the **principal point** p^*;
- the **focal plane** $\mathcal{F}$ is the green plane which contains C, parallel to the image plane;
- the **focal distance** f refers to the distance between the center C and $\mathcal{R}$;
- the **virtual image plane** $\mathcal{V}$ is the purple plane located in front of the camera at the focal distance f which virtually contains the upright image of the scene;

These elements and the camera structure just described, are necessary to define the mathematical relationship between a 3D real object and its correspondent projection onto the 2D image plane. Perspective projection, as shown in Figure 3.2, illustrates accurately how the light coming from a 3D objects and passing through the tiny hole, creates inverted images on the image plane $\mathcal{R}$. However it is sometimes convenient to take into account the correspondent upright image on the virtual image plane $\mathcal{V}$, associated to the plane in front of the pinhole, at the distance f, in order to consider the positive coordinates on z axis.
In Figure 3.3 it is described how the 3D scene points $M = (X, Y, Z)$ is mapped to the correspondent point $m = (x, y, z)$, according to the coordinate system (C, x, y, z), where the origin C corresponds to the pinhole. In particular, since m lies on the image plane $\mathcal{V}$ we have $z = f$, then $m = (x, y, f)$.

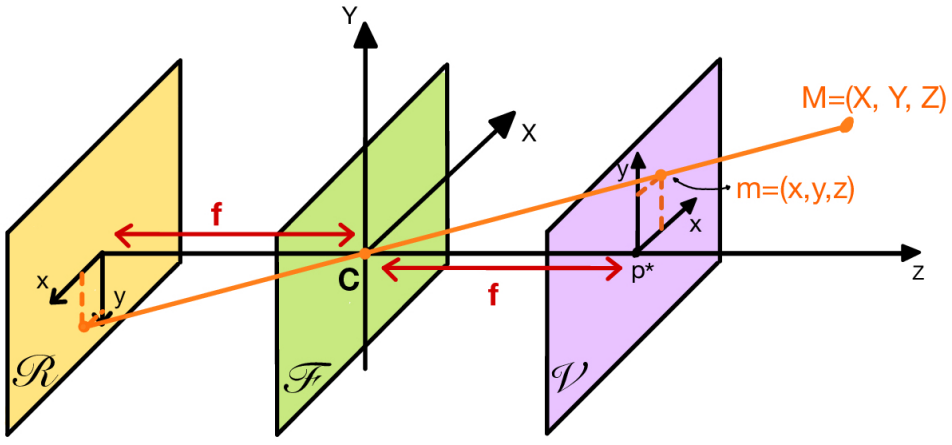


Figure 3.3: *Perspective geometry of pinhole camera model.*

Moreover, since C, M and m are collinear, we can write $\overrightarrow{Cm} = \lambda \overrightarrow{CM}$, for some values λ. So, it can be written the following geometric relationship:

$$\begin{cases} x = \lambda X \\ y = \lambda Y \\ f = \lambda Z \end{cases} \iff \lambda = x/X = y/Y = f/Z \tag{3.1}$$

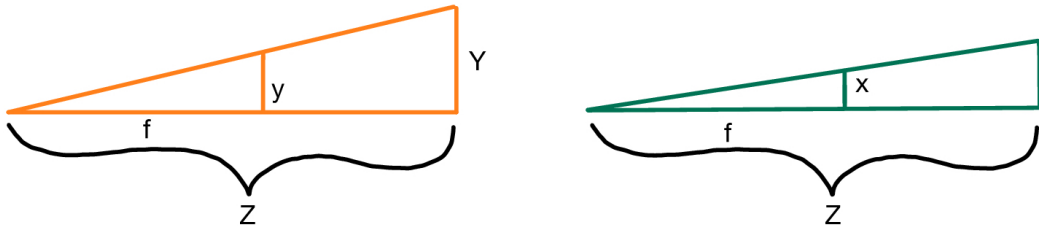


Figure 3.4: *Perspective geometry of a 3D point projected on Y and X axis.*

therefore, according to equation (3.1) and the triangle trigonometry, as it is shown in Figure 3.4, the coordinates (x, y) of the image point m are the following:

$$\begin{cases} x = fX/Z \\ y = fY/Z \end{cases} \tag{3.2}$$

The equations 3.1 and 3.2, which describe the perspective projection from $M = (X, Y, Z)$ to $m = (x, y)$, define the transformation from world coordinate system to image coordinate system that can be shown by the following map:

$$M = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow m = \begin{bmatrix} f\frac{X}{Z} \\ f\frac{Y}{Z} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.3}$$

where $m = (x, y)$ is given with respect to the 2D coordinate frame (O, x, y) as in Figure 3.5.

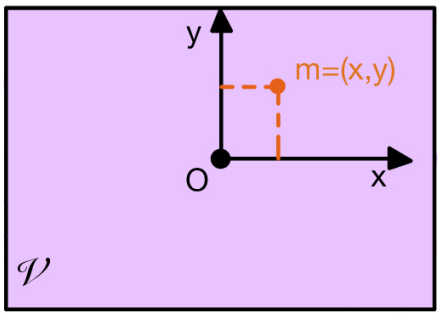


Figure 3.5: *Projection of point $p = (x, y)$ on the virtual image plane $\mathcal{V}$.*

The equation that compute this mapping from the 3D point $M = (X, Y, Z)$ to a 2D point $m = (x, y)$ is called **perspective projection transformation**.
In order to analyze this transformation and to define it as linear mapping , it is convenient to transform the 3D point M from cartesian coordinates $(X, Y, Z) \in \mathbb{R}^3$ to homogeneous coordinates $\tilde{M} = (X, Y, Z, W) \in \mathbb{R}^4$ as explained in [65]. The transformation from cartesian coordinates to homogeneous coordinates are done by adding an extra coordinate W[5] so that

[5]For simplicity, it is often equal to 1.

the dimensionality will be increased by 1.

	Cartesian coords	Homogeneous coords
3D	$M = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$	$\tilde{M} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$
2D	$m = \begin{bmatrix} x \\ y \end{bmatrix}$	$\tilde{m} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

Hence, according to the table 3.1 the mapping reported in equation (3.3) can be written as a linear mapping, exploiting the representation in homogeneous coordinates, from the 2D image point $\tilde{m} = (x, y, 1)$ to $\tilde{M} = (X, Y, Z, 1)$, as follow:

$$Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \overset{(3.2)}{=} Z \begin{bmatrix} fX/Z \\ fY/Z \\ 1 \end{bmatrix} = \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.4}$$

where it is assumed the principal point p^* at the origin of image reference frame (O, x, y) and the camera at the center of world coordinates. This linear mapping can be written compactly with the following equation:

$$\tilde{m} \simeq P\tilde{M} \qquad \text{with} \quad P = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.5}$$

where the matrix P is the $3x4$ matrix called **camera projection matrix**. It defines the mapping $\mathbb{R}^3 \to \mathbb{R}^2$, namely the transformation from a world 3D point $\tilde{M}$ to the correspondent 2D point $\tilde{m}$ with respect to the image reference frame.

3.2 Intrinsic and extrinsic parameters

The perspective equation (3.4) derived in section 3.1 is valid only assuming that the principal point p^* is placed at the origin of image plane (O, x, y) and the camera coordinates frame correspond to the world coordinates frame. However, the world and the camera are generally related by a set of parameters, such as the size of the pixels, the position of the image center, and the position and orientation of the camera with respect to a fixed world frame as well as the focal distance of the camera. In this section these parameters are identified and accurately described. In particular it is analyzed the distinction between the **intrinsic parameters** necessary to relate the camera's coordinate system to the image coordinate system and the **extrinsic parameters** which determine the position and the orientation of the camera coordinates system with respect to the world coordinates system. Hence, the principal point p^* is generally positioned in pixel coordinates (c_x, c_y) on the image plane as shown in Figure 3.6 and moreover coordinates (x, y) of the image point m are usually expressed in pixel units (u, v).

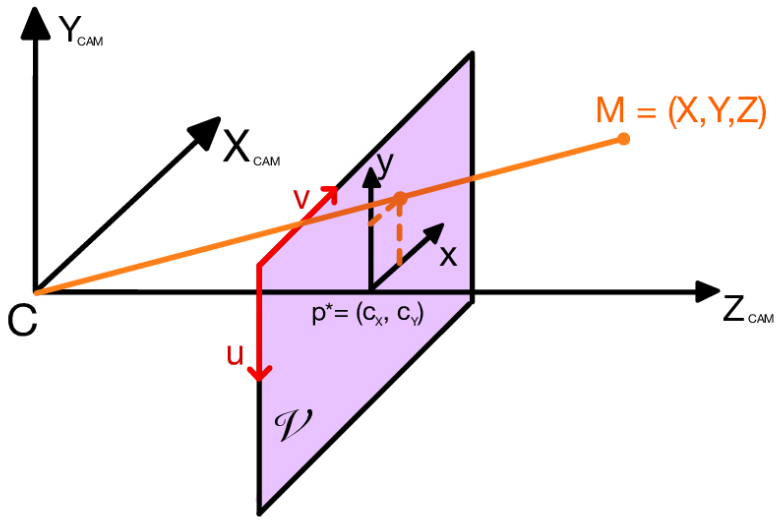


Figure 3.6: *Perspective projection of Pinhole camera model, considering a principal point* p^* *at coordinates* (c_x, c_y) *of image plane.*

Therefore the coordinates (x, y) of m defined in equation 3.2 can be rewritten in pixel coordinates (u, v) as follow:

$$\begin{cases} x = f\frac{X}{Z} \\ y = f\frac{Y}{Z} \end{cases} \rightarrow \begin{cases} u = k_u f\frac{X}{Z} + c_x \\ v = k_v f\frac{Y}{Z} + c_y \end{cases} \tag{3.6}$$

where k_u and k_v are the inverse of the pixel dimension in the u and v direction respectively $[pix/m]$, (c_x, c_y) are the coordinates in pixels of the principal point p^* and f is the focal distance. The parameters (k_u, k_v, c_x, c_y, f) are defined as **intrinsic parameters** of a camera and they are necessary to project a 3D point with respect to camera reference frame to a 2D point with respect the image reference frame. A further notation is $\alpha_u = k_u f$ and $\alpha_v = k_v f$ which are the focal lengths expressed in pixels. According to the pixel coordinates of m of the equation (3.6), the linear mapping described in equation (3.4) can be rewritten as follow:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \simeq \begin{bmatrix} k_u f & 0 & c_x & 0 \\ 0 & k_v f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{K}\,[I|\mathbf{0}]\,\mathbf{X}_{\text{cam}} \tag{3.7}$$

where $\mathbf{K}$ is the **camera calibration matrix** and $\mathbf{X}_{\text{cam}}$ denotes the 3D points coordinates with respect to the camera coordinates frame.
However, in general, world 3D points are expressed with respect to a world coordinate frame different from the camera coordinate frame. The two coordinates systems are related through the translation vector $\mathbf{t}$ and a rotation matrix $3x3$ R.

The transformation between the two reference frames can be described by the following equation:

$$\mathbf{X}_{\text{cam}} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_{\text{cam}} = G\mathbf{X}_{\text{world}} = \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_{\text{world}} \quad \text{with} \quad R = \begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \mathbf{r}_3^T \end{bmatrix} \quad t = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \tag{3.8}$$

where G denotes the rototranslation which transforms the 3D point $\mathbf{X}_{\text{world}}$ in world coordinates frame to the 3D point $\mathbf{X}_{\text{cam}}$ in camera coordinates frame, as illustrated in Figure 3.7. In particular, when the rotation matrix R is written as the product of three elementary rotations around each axis, the row vectors $\mathbf{r}_i$ for $i = 1, 2, 3$ can be written in terms of the corresponding three angles. Thus, the 3 elementary rotations related to R and the 3 translation components t_1, t_2, t_3 on x, y, z axis respectively, are defined as the **extrinsic parameters**. These 6 parameters define the camera orientation and position with respect to a world coordinate system.

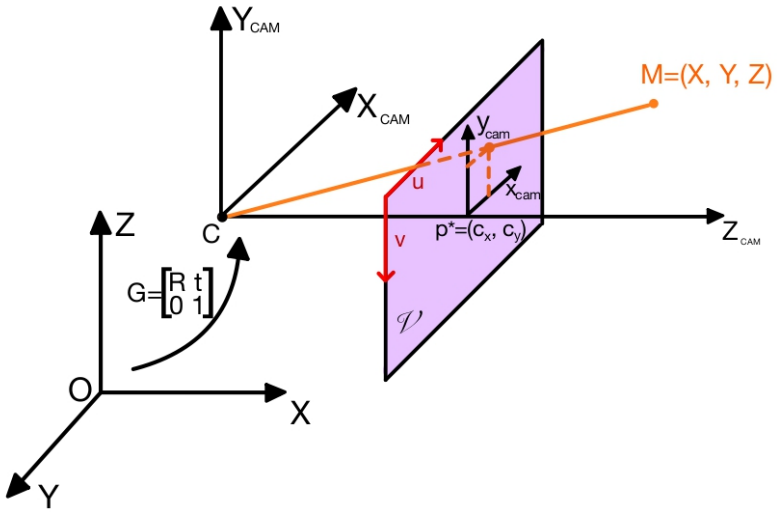


Figure 3.7: *Perspective projection of Pinhole camera model, considering a principal point p^* at coordinates (u_0, v_0) of image plane and a camera at R,t of world coordinates.*

Hence, considering a general scenario, in which a principal point is placed at (c_x, c_y) on the image plane and the camera is transformed from the world reference frame by a rotation R and translation $\mathbf{t} = [t_1, t_2, t_3]^T$, we can derive the following camera projection matrix:

$$P = K[I|\mathbf{0}]G = K[R|\mathbf{t}] = \begin{bmatrix} \alpha_u \mathbf{r}_1^T + c_x \mathbf{r}_3^T & \alpha_u t_1 + c_x t_3 \\ \alpha_v \mathbf{r}_2^T + c_y \mathbf{r}_3^T & \alpha_v t_2 + c_y t_3 \\ \mathbf{r}_3^T & t_3 \end{bmatrix} \tag{3.9}$$

Thus, the linear mapping reported in the equation (3.5) is rewritten as follow:

$$m \simeq K[I|\mathbf{0}]M_C = K[I|\mathbf{0}]GM = K[R|\mathbf{t}]M_C \tag{3.10}$$

where G corresponds to the matrix of **extrinsic parameters** and K is the matrix of **intrinsic parameters**.

It can be observed that the world points are transformed to camera coordinates using the extrinsics parameters and points in the camera coordinates are mapped to 2D points in the image plane using the intrinsics parameters, as it is schematically reported in Figure 3.8.

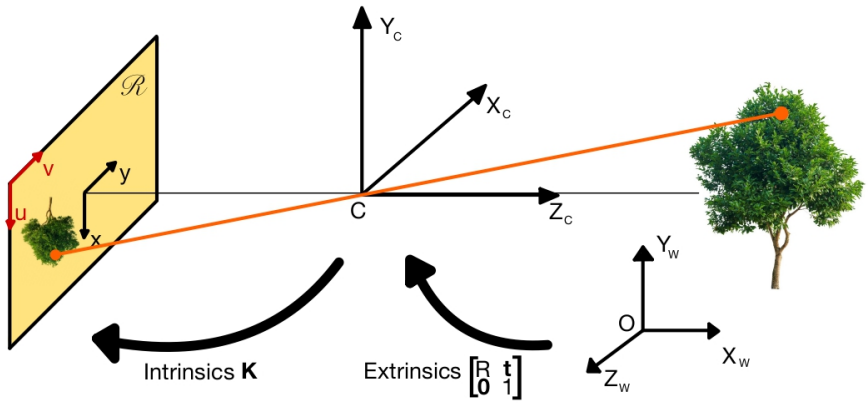


Figure 3.8: *Extrinsic parameters transform the 3D point from the world coordinate to camera coordinates. Intrinsic parameters maps the 3D point from the camera coordinates to the image plane.*

3.3 Camera with lenses

Most real cameras are equipped by lenses. This equipment allows the camera to gather more light than the simple pinhole camera model. As described in section 3.1 the pinhole camera model is ideally structured in order to let pass only a single light ray coming from each point of the 3D world scene and focus it on the correspondent 2D point on the image plane, however, due to the size of the real pinhole, in general each point in the image plane is illuminated by a cone of light rays. For this reason, the larger the hole, the wider the cone and then the brighter the image, but a large pinhole produces blurry pictures. Instead shrinking the tiny hole gives sharper images but reduces the amount of light reaching on the image plane. The addition of the **lens** in the camera model allows to gather more light rays coming from a point in the world rather than one light ray as in pinhole model, overcoming the crispness and brightness issues of pinhole camera model. As it is illustrated in Figure 3.9, by adding a thin lens all rays of light that are emitted by an object point are refracted by the lens such that they converge to a single point on the image plane, overcoming the problem of low brightness. However this property does not hold for each 3D point at different distance from the lens, for instance for a point P which is closer than M to the lens, the corresponding projection on the image plane is blurred or out of focus (see Figure 3.10). This lens property, called **depth of field**, defines the range of distance from the camera on which an object is in focus. Another interesting property of the lens is that all the light rays which travel parallel to the optical axis pass through a common point called **focal point**, shown in Figure 3.9. The distance between the focal point and the center of the lens is known as **focal length** and it strongly depends on the curvature[6] of the lens [66].

[6]Curvature represents how curved a surface is or how much it has deviated from being straight/planar. It is simply the reciprocal of the radius of the curve.

Hence, considering the previous lens properties, thin lens model can be illustrated as follow:

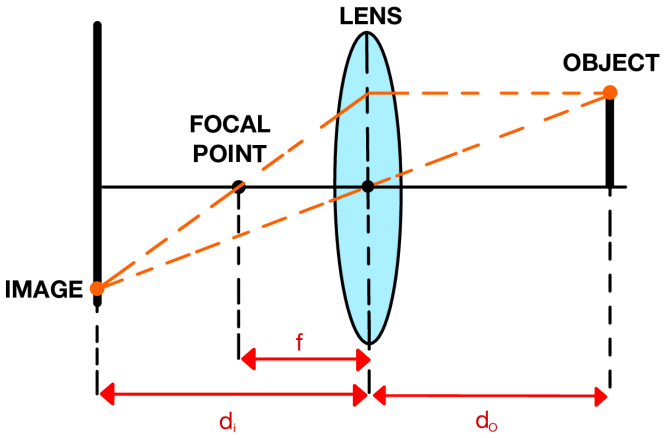


Figure 3.9: *Perspective projection of thin lens model.*

In particular f denotes the **focal length**[7] f of the lens, which is strictly related to the lens characteristics, d_i is the distance between the lens center and the image plane and d_O is the distance from the object to the lens. From the thin lens model structure can be derived the following this lens equation:

$$\frac{1}{d_O} + \frac{1}{d_i} = \frac{1}{f} \tag{3.11}$$

which determines at what distance d_O from the lens, an object is in focus. In particular an object that does not satisfy this constraint is not in focus and creates the phenomenon called *circle of confusion*, defined as the optical spot where a point of light grows to a circle that can be seen in the final image [67]. This phenomenon is illustrated in Figure 3.10.

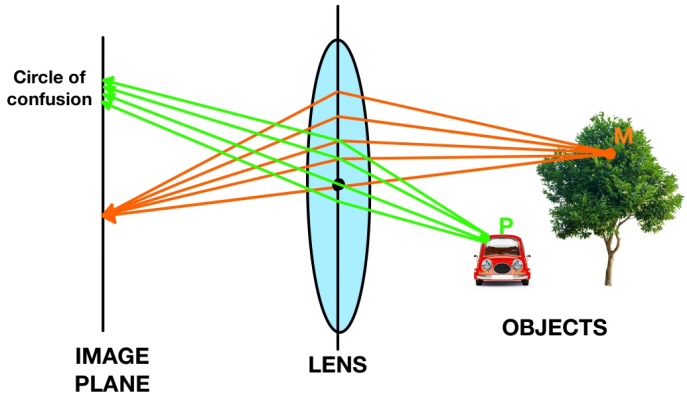


Figure 3.10: *Lens can gather more light rays coming from the objects, but only a set of points at a specific distance are in focus, the others are out of focus and then creates on the image plane a circle of confusion.*

[7] It is computed by $f = \frac{R}{2(n-1)}$ where R is the lens spherical surface and n is the refraction index [63].

This issue is caused by a cone of light rays coming from a lens not coming to a perfect focus when it is projected onto the image plane.
A further property of the lens is its **field of view** ϕ, or FoV, which is defined as the maximum area of a sample that a camera can image (see Figure 3.11).

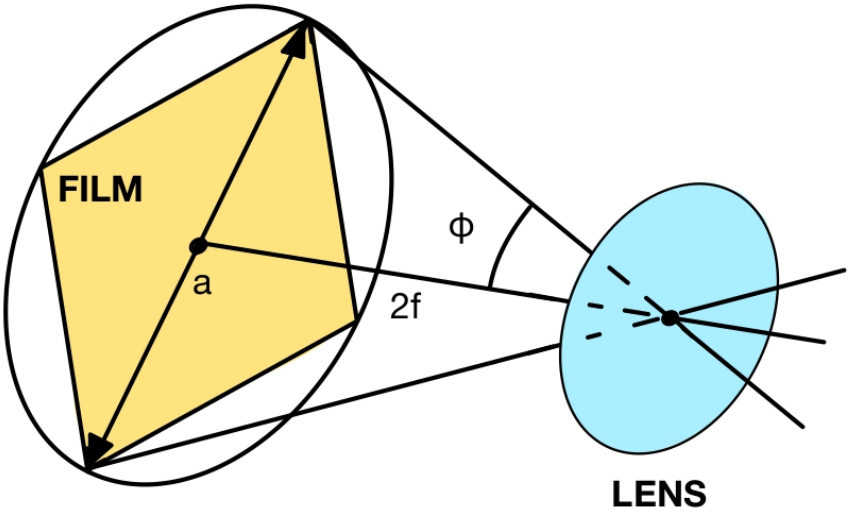


Figure 3.11: *Camera FoV is strictly related to the size of film and focal length of the lens.*

As it can be observed from the equation 3.12, the FoV is strictly related to the focal length f of the camera and the sensor size a [68]. It can be computed as follow:

$$\phi = \arctan\left(\frac{a}{2f}\right) \tag{3.12}$$

In particular, assuming that the focal length of the lens remains constant, the larger the sensor the larger the field of view. The sensor size is determined by both the number of pixels on the sensor, and the size of the pixels.

Considering the properties previously analyzed, it is assumed that the cameras do not have any distortions, namely the straight lines in the world remain straight line in the image. However, many lenses suffer from optical distortions, caused by the optical design of lenses. The most common lens distortions which occur, especially on wide-angle lenses[8] [69] is the **radial distortion**. This lens distortion is caused as a result of the shape of lens and happens when light rays bend more near the edges of a lens than they do at its optical center. The radial distortions can be classified in two main aberrations: pincushion distortion and barrel distortion. In particular, **pincushion distortion** consists in a projected distance increment beyond the expected locations as one moves farther away from the centre of the image plane and displays the coordinates in an image towards the image center. Instead the **barrel distortion** is the projected distance reduction from the expected locations as one moves farther away from the centre of the image plane and the coordinates in an image are placed away from the image center. In Figure 3.12 are illustrated images affected by these two radial distortions.

[8] A wide angle lens is any lens with a short focal length and a wide field of view.

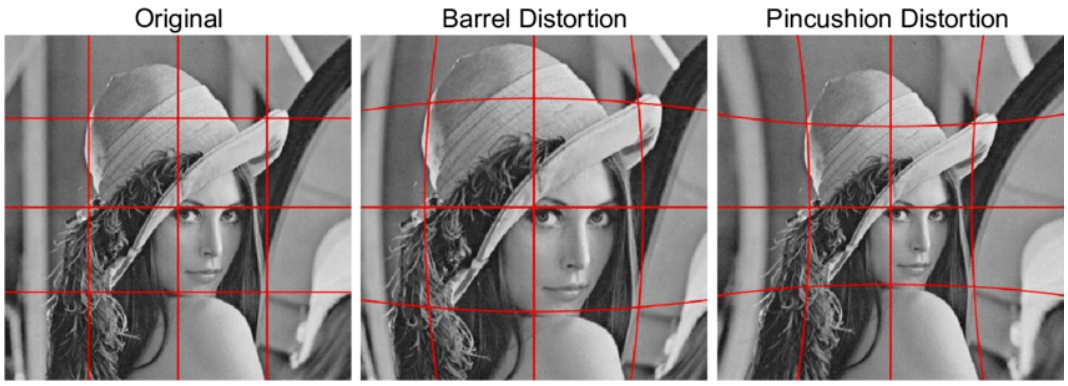


Figure 3.12: *Comparison among the original image at the left, its barrel distortion and at the right the pincushion distortion.*

A further lens distortion that can affect the lens model is the **tangential distortion**, which arises from the assembly process of the camera as a whole, namely it occurs when the lens and the image plane are not parallel. The effect of this lens distortion is illustrated in Figure 3.13.

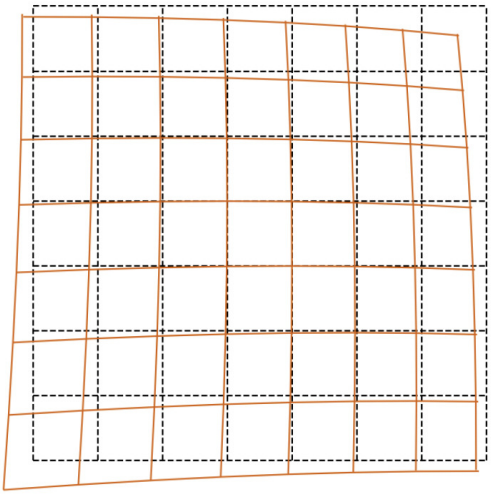


Figure 3.13: *Tangential distortion of the image plane where the main issue is referred to the skewness of the image plane.*

In particular, the image coordinates, when radial distortions and tangential distortions occur, can be relocated according to the following equations:

$$\begin{cases} x_{\text{correlated}} = x + x \cdot (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 xy + p_2(r^2 + 2x^2) \\ y_{\text{correlated}} = y + y \cdot (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2y^2 + 2p_2 xy) \end{cases} \tag{3.13}$$

where x, y denote the undistorted pixel locations. k_1, k_2 and k_3 are the radial distortion coefficients of the lens, p_1 and p_2 are the tangential distortion coefficients and $r^2 = x^2 + y^2$.

3.4 Camera calibration

As it is introduced in section 1 the **camera calibration** is the process of estimating the intrinsic and extrinsic parameters of the camera. Specifically, it is necessary referring to the following projective mapping from world coordinates to pixel coordinates:

$$\lambda \tilde{m} = K[I|\mathbf{0}]G\tilde{M} = K[R|\mathbf{t}]\tilde{M} \tag{3.14}$$

where λ is an arbitrary scale factor, $\tilde{m}$ is the 2D point on the image plane in pixel homogeneous coordinates , K is the matrix of intrinsic parameters, G is the roto-translation matrix composed by extrinsic parameters and $\tilde{M}$ is the 3D world point in homogeneous coordinates. In particular, camera calibration consists in the optimization process required to determine the matrices K, R and $\mathbf{t}$.
Most used approaches for camera calibration are:

- **Direct linear transformation** (DLT) method which exploits a set of control points of an object whose space coordinates are already known. It essentially compute the linear mapping between the 2D image space coordinates and the 3D object space coordinates, assuming that the camera does not suffer by any distortion such as for the pinhole camera model [70, 71, 72].

- **Zhang's method** [40]. It consists in a calibration technique only requiring the camera to observe a planar pattern shown at a few (at least two) different orientations. In particular in the first step, it adopts the homography[9] to identify the map between a model point M and its image m and then it takes into account lens distortion, adopting nonlinear optimization technique based on the maximum likelihood criterion.

- **Bouguet's method** which uses lines and planes in a consistant framework allowing to identify intrinsic and extrinsic camera parameters, and then extract closed form solutions for them, by exploiting geometrical constraints existing in the scene [73]. It is also released as Camera Calibration Toolbox for Matlab [74].

They are three different camera calibration methods which share the same basic procedure. In particular, they exploits a pattern whose space 3D geometry is already known (for instance checkerboard, aruco board[10] [75], etc.). They require to capture more images of that calibration target oriented in different perspective and to extract all corners/control points. Then it is implemented an optimization process, such that the squared distance between the observed image control points and their theoretical positions is minimized, with respect to the intrinsic and extrinsic parameters of the camera.
To get started to analyze the basic procedure of camera calibration process, it is assumed that the camera does not suffer by any distortion. Considering the first part of the camera matrix equation $\lambda\tilde{m} = K[I|\mathbf{0}]G\tilde{M} = PM$, a set of correspondences between 2D control points $\tilde{m}_i = [x_i, y_i]^T$ $(i = 0, 1, \ldots, N)$ on the image plane and the related 3D points M_i are required, in order to compute the camera matrix $3x4$ P. For this reason a calibration pattern with known geometric features is necessary, such as the checkerboard in Figure 3.14.

[9] It is a matrix which maps the points residing on a plane from world coordinates to the corresponding image coordinates.
[10] It is a particular planar board composed by several markers arranged in a grid.

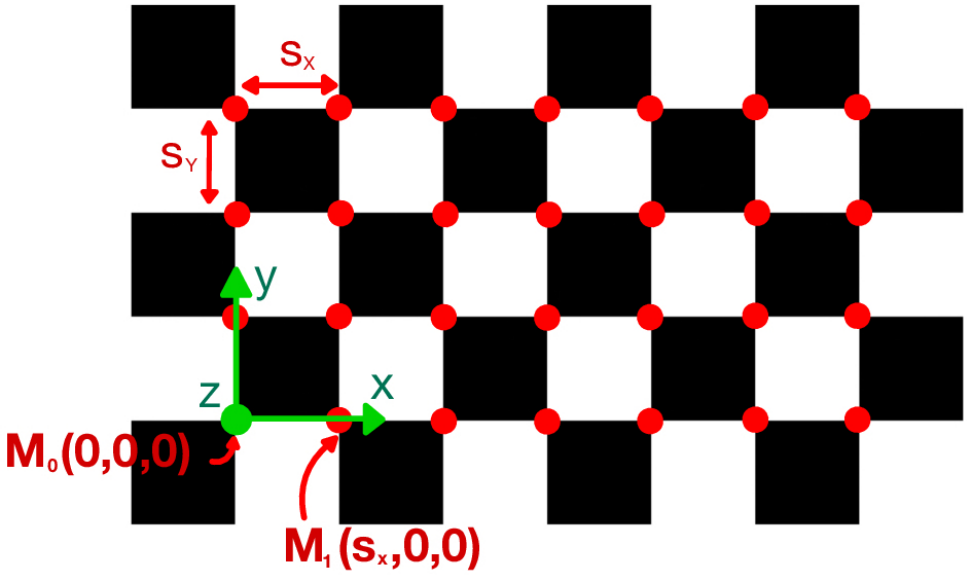


Figure 3.14: *Checkerboard with 4x7 corners and with known geometry. The distance between each corner on the x axis is equal to s_x, the distance on the y axis is s_y.*

Such pattern allows to list the N 3D corners position as $M_0(0,0,0)$, $M_1(s_x,0,0)$, etc., w.r.t. the checkerboard reference frame whose origin is placed in M_0 and then the third coordinate is $z = 0$. Moreover detecting these points in the captured image allows to determine the correspondent 2D points $m_1, \dots, m_N$ on the image. Hence, it can be set up a linear system of N correspondences as follow:

$$\begin{cases} \lambda m_1 = PM_1 \\ \dots \\ \lambda m_N = PM_N \end{cases} \tag{3.15}$$

In particular, since each correspondence of the linear system provide two equations, one for the x coordinate and the other for the y coordinate, we need $N \geq 6$ correspondences in order to determine the 11 camera parameters (5 intrinsic and 6 extrinsic, see Section 3.2). Furthermore, in order to build the optimization problem to determine the matrix P, it is convenient to rewrite that matrix P in its vectorized version p_{stack} as follow:

$$P = \left[\begin{array}{c|c} \mathbf{p}_1^T & p_{14} \\ \mathbf{p}_2^T & p_{24} \\ \mathbf{p}_3^T & p_{34} \end{array}\right] = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \longrightarrow p_{\text{stack}} = \left[\begin{array}{c} p_{11} \\ \vdots \\ p_{14} \\ \hline p_{21} \\ \vdots \\ p_{24} \\ \hline p_{31} \\ \vdots \\ p_{34} \end{array}\right] \in \mathbb{R}^{12\times 1} \tag{3.16}$$

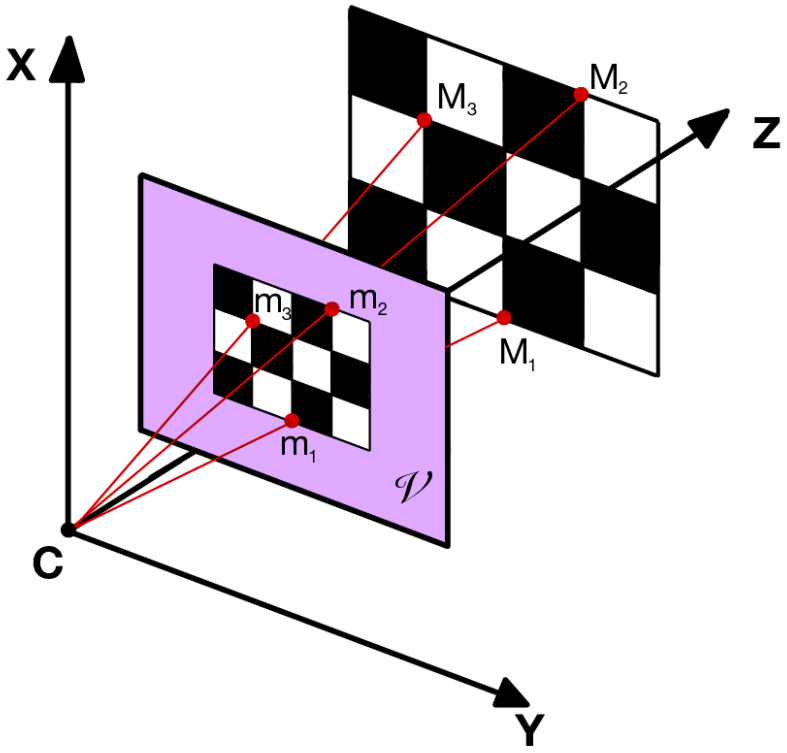


Figure 3.15: *Example of correspondences* (m_i, M_i) *between the image plane and the target plane in the 3D world for a checkerboard.*

Hence, given the correspondences $(m_i, M_i)_{i=0,\dots,N}$, as they are illustrated in Figure 3.15, it can be formalized a linear problem as $Ap_{\text{stack}} = 0$ where the matrix A contains the information about the correspondences and the unknown vector p_{stack} has to be determined. In particular, we can derive from the linear system (3.15) the following equations:

$$\begin{cases} \lambda x_i = \mathbf{p}_1^T M_i + p_{14} \\ \lambda y_i = \mathbf{p}_2^T M_i + p_{24} \\ \lambda = \mathbf{p}_3^T M_i + p_{34} \end{cases} \implies \begin{cases} x_i = \frac{\mathbf{p}_1^T M_i + p_{14}}{\mathbf{p}_3^T M_i + p_{34}} \\ y_i = \frac{\mathbf{p}_2^T M_i + p_{24}}{\mathbf{p}_3^T M_i + p_{34}} \end{cases} \qquad \forall i = 0, \dots, N \tag{3.17}$$

where the last equations are defined for each correspondences $(m_i, M_i)_{i=0,\dots,N}$. The system (3.17) can be rewritten as:

$$\begin{cases} \mathbf{p}_3^T M_i x_i + p_{34} x_i = \mathbf{p}_1^T M_i + p_{14} \\ \mathbf{p}_3^T M_i y_i + p_{34} y_i = \mathbf{p}_2^T M_i + p_{24} \end{cases} \implies \begin{cases} \mathbf{p}_3^T M_i x_i + p_{34} x_i - \mathbf{p}_1^T M_i - p_{14} = 0 \\ \mathbf{p}_3^T M_i y_i + p_{34} y_i - \mathbf{p}_2^T M_i - p_{24} = 0 \end{cases} \tag{3.18}$$

From equation's system (3.18), by swapping the transpose operation of the two vectors, we obtain:

$$\begin{cases} M_i^T \mathbf{p}_3 x_i + p_{34} x_i - M_i^T \mathbf{p}_1 - p_{14} = 0 \\ M_i^T \mathbf{p}_3 y_i + p_{34} y_i - M_i^T \mathbf{p}_2 - p_{24} = 0 \end{cases} \tag{3.19}$$

Then we can derive the matrix form representation of the linear problem as follow:

$$\left[\begin{array}{cc|cc|cc} -M_i^T & -1 & 0 & 0 & M_i^T x_i & x_i \\ 0 & 0 & -M_i^T & -1 & M_i^T y_i & y_i \end{array}\right] \left[\begin{array}{c} \mathbf{p}_1 \\ p_{14} \\ \hline \mathbf{p}_2 \\ p_{24} \\ \hline \mathbf{p}_3 \\ p_{34} \end{array}\right] = A_i p_{\text{stack}} = 0 \tag{3.20}$$

by repeating the $A_i \in \mathbb{R}^{2\times 12}$ matrix for N times, one for each correspondences $(m_i, M_i)_{i=0,\dots,N}$ and by stacking them we obtain the final matrix $A \in \mathbb{R}^{2N\times 12}$ as follow:

$$\left[\begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_N \end{array}\right] \left[\begin{array}{c} \mathbf{p}_1 \\ p_{14} \\ \hline \mathbf{p}_2 \\ p_{24} \\ \hline \mathbf{p}_3 \\ p_{34} \end{array}\right] = A p_{\text{stack}} = 0 \tag{3.21}$$

Hence, although theoretically the $rank(A)$ is 11, in practice, since the correspondences $(m_i, M_i)_{i=0,\dots,N}$ are provided by noisy measurements, we have $rank(A) = 12$, so the optimization problem can not be solved exactly. Then the optimization problem can be rewritten in the following form:

$$p^* = \arg\min_p ||A p_{\text{stack}}||^2 \qquad \text{such that} \quad ||p_{\text{stack}}|| = 1 \tag{3.22}$$

The solution of this optimization problem is given by the singular value decomposition of $A = USV^T$ corresponding to the smallest singular value of A [76].
After re-transforming the vector p into the initial matrix $P = K[R|\mathbf{t}]$, the extrinsic and intrinsic parameters can be determined, in particular since the SVD-solved of P is known up to scale, the true values of the camera matrix are some scalar multiple of P as follow:

$$\alpha P = [A|b] = \begin{bmatrix} \mathbf{a}_1^T & b_1 \\ \mathbf{a}_2^T & b_2 \\ \mathbf{a}_3^T & b_3 \end{bmatrix} = \begin{bmatrix} f_x \mathbf{r}_1^T - f_x \cot\theta \mathbf{r}_2^T + c_x \mathbf{r}_3^T & f_x t_x - f_x \cot\theta t_y + c_x t_z \\ \frac{f_y}{\sin\theta} \mathbf{r}_2^T + c_y \mathbf{r}_3^T & \frac{f_y}{\sin\theta} t_y + c_y t_z \\ \mathbf{r}_3^T & t_z \end{bmatrix} \tag{3.23}$$

where f_x and f_y are the focal lengths in x and y coordinates respectively, $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ are three rows of the rotation matrix R, t_x, t_y, t_z are the three components of translation vector $\mathbf{t}$, α is an unknown scale factor and θ is the angle between the two image axes which is not perfectly equal to $\frac{\pi}{2}$ and it is given by camera coordinate system skewness, due to some manufacturing error. In particular, exploiting the property of rotation matrix such that its rows have unit length and they are perpendicular to each other and considering $\theta \simeq \frac{\pi}{2} \Longrightarrow \sin\theta > 0$ we can derive the following intrinsic parameters:

$$\begin{cases} \alpha = \pm\frac{1}{\varepsilon} \\ c_x = \rho^2(\mathbf{a}_1 \cdot \mathbf{a}_3) \\ c_y = \rho^2(\mathbf{a}_2 \cdot \mathbf{a}_3) \\ \theta = \cos^{-1}\left(-\frac{(\mathbf{a}_1 \times \mathbf{a}_3)\cdot(\mathbf{a}_2 \times \mathbf{a}_3)}{||\mathbf{a}_1 \times \mathbf{a}_3|| \cdot ||\mathbf{a}_2 \times \mathbf{a}_3||}\right) \\ f_x = \rho^2 ||\mathbf{a}_1 \times \mathbf{a}_3|| \sin\theta \\ f_y = \rho^2 ||\mathbf{a}_2 \times \mathbf{a}_3|| \sin\theta \end{cases} \tag{3.24}$$

and the extrinsic parameters:

$$\begin{cases} \mathbf{r}_1 = \frac{\mathbf{a}_2 \times \mathbf{a}_3}{||\mathbf{a}_2 \times \mathbf{a}_3||} \\ \mathbf{r}_2 = \mathbf{r}_3 \times \mathbf{r}_1 \\ \mathbf{r}_3 = \rho \mathbf{a}_3 \\ T = \rho K^{-1} \mathbf{b} \end{cases} \tag{3.25}$$

In particular as described in Zhang's paper [40] the solution of the equation (3.22) is obtained through minimizing an algebraic distance which is not physically meaningful. Hence it can be refined through maximum likelihood inference. Since the correspondences are always affected by noise, M images of the planar pattern are given and there are N control points on the calibration pattern. Assuming that the image points are corrupted by independent and identically distributed noise, the maximum likelihood estimate can be obtained by minimizing the following functional:

$$\sum_{j=0}^{M} \sum_{i=0}^{N} ||m_{ji} - \hat{f}(K, R_j, \mathbf{t}_j, M_{ji})||^2 \tag{3.26}$$

where $\hat{f}(K, R_j, \mathbf{t}_j, M_j)$ is the projection of 3D points M_i of image j according to the equation (3.15). Hence, minimizing (3.26) we reformulate the problem as a nonlinear minimization problem, which is solved with the Levenberg-Marquardt Algorithm [60].
In general scenario, where a camera is equipped by at least one lens, the camera calibration is necessary to determine as well the radial distortion parameters of the lens. As proposed by the Zhang's method, the coefficients k_1 and k_2 of the radial distortion, can be estimated after the determination of the other parameters. Then considering the image coordinates defined by the equation (3.13), and taking into account only k_1 and k_2 for simplicity, we have:

$$\begin{cases} x_{\text{correlated}} = x + x \cdot (k_1(x^2 + y^2)^2 + k_2(x^2 + y^2)^4) \\ y_{\text{correlated}} = x + y \cdot (k_1(x^2 + y^2)^2 + k_2(x^2 + y^2)^4) \end{cases} \tag{3.27}$$

that can be rewritten in matrix form as follow:

$$\begin{bmatrix} x(x^2 + y^2) & x(x^2 + y^2)^2 \\ y(x^2 + y^2) & y(x^2 + y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} x_{\text{correlated}} - x \\ y_{\text{correlated}} - y \end{bmatrix} \tag{3.28}$$

Hence, given M images with N points, all $2MN$ equations can be stacked together in order to obtain the equation $D\mathbf{k} = d$ where $\mathbf{k} = [k_1, k_2]^T$. Then the linear least-squares solution is given by

$$\mathbf{k} = (D^T D)^{-1} D^T d \tag{3.29}$$

In alternative, the functional (3.26) can be extended to the following form and then minimized in order to estimate the complete set of parameters:

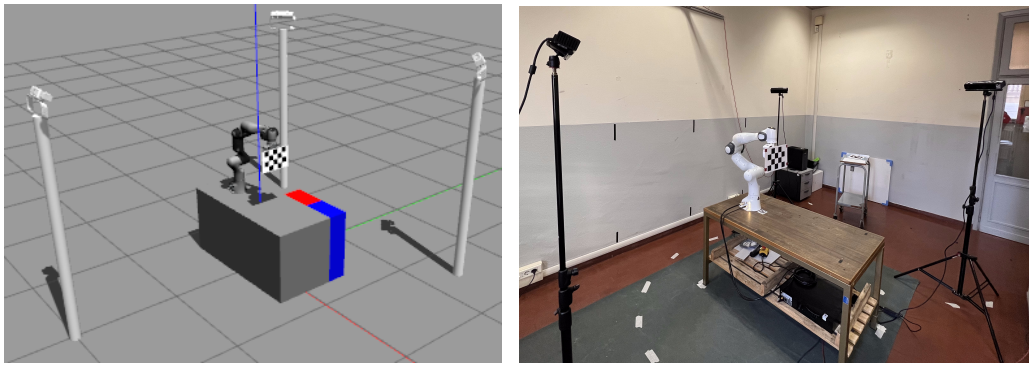
$$\sum_{j=0}^{M} \sum_{i=0}^{N} ||m_{ji} - \hat{f}(K, k_1, k_2, R_j, \mathbf{t}_j, M_j)||^2 \tag{3.30}$$

where $\hat{f}(K, k_1, k_2, R_j, \mathbf{t}_j, M_j)$ is the projection of 3D points M_i of image j considering the lens radial distortions.

Chapter 4

Proposed eye-on-base calibration method

As introduced in section 1.2, the main purpose of this thesis was developing a novel multi-sensor eye-on-base calibration method to be adopted in a robotic workcell for human-robot collaboration tasks as shown in Figure 4.1.



(a) *Virtual robotic workcell setup designed in a simulator.*

(b) *Real robotic workcell setup designed in our research laboratory Robot Vision Lab.*

Figure 4.1: *Setup examples of a multi camera system and a robot arm for eye-on-base calibration for robotic workcells.*

For this purpose, in this thesis we propose a calibration process for a robotic workcell with a multi-camera system that is fast to be accomplished, especially in a real scenario where it can happen that a camera is unintentionally touched or moved from its position and a new calibration is required and it is necessary not to stop the working system for a long time. At the same time, it is a calibration process as general as possible, namely that it can be implemented with different setup and with different hardware system, in order to be suitable for each possible environment. Moreover, as it will be described more precisely in experiments and results section 6, it is also really accurate, in order to ensure human operator safety and to allow an excellent collaboration working system even for more complex tasks. This section is focused on the accurate description of the proposed eye-on-base calibration

method and the analysis of mathematical theory which is behind such optimization process. In particular in the subsection 4.1 is described the first step of the calibration technique which deals with the single camera eye-on-base calibration process. In the subsection 4.2 the final multi-camera eye-on-base calibration process is accurately analyzed. Moreover in the subsection 4.3 an helpful function is described which helps to guarantee the correctness and the accuracy of the checkerboard detection.

4.1 Single camera eye-on-base calibration

The first step for this thesis was to accomplish a single camera eye-on-base calibration process which can be implemented one time for each sensor of the camera network in the robotic workcell. In particular this proposed calibration approach consists in a set of small processes that work together to achieve the final calibration of the camera with respect to the robot arm. The structure of the whole calibration framework is shown in Figure 4.2.

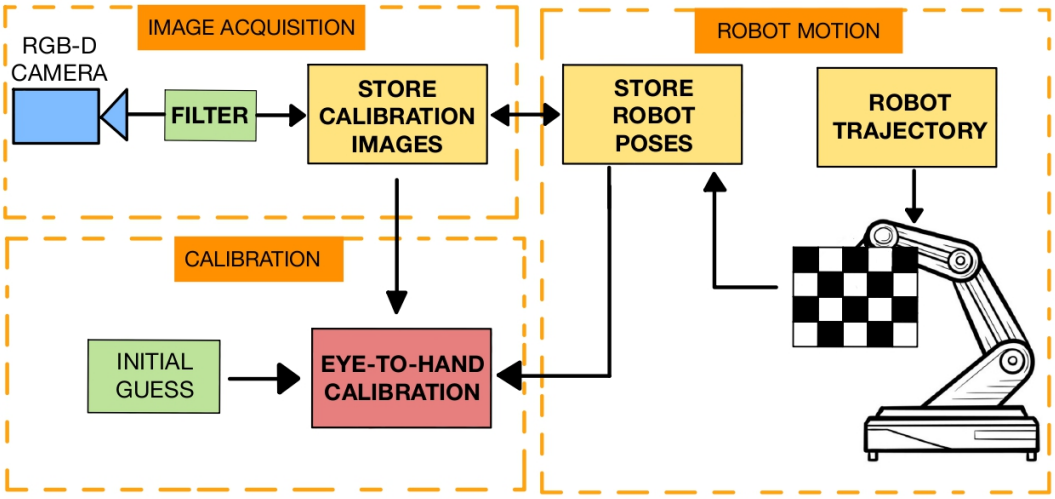


Figure 4.2: *The whole proposed eye-on-base calibration process. It is divided in three subprocesses, one dedicated to the image acquisition, one related to the robot motion and the other dedicated to the optimization process of the final calibration.*

As shown in Figure 4.2, the whole calibration process is split in three different modules which works together. It can be outlined the **robot motion phase**, which controls the correct execution of a predefined trajectory[1] by the robot arm, in order to move the robot in front of the camera. This phase works in parallel with **image acquisition phase**, where the camera is programmed to capture several checkerboard images, one for each pose of the robot arm. Moreover, during the images capture, the checkerboard is required to be clearly visible and correctly detectable, to guarantee more accuracy on the calibration. In order to accomplish this step and to store all the calibration images correctly, it is implemented a *filter* which guarantees to store, only the checkerboard images where control points are

[1]It was previously stored one trajectory for each camera to be adopted for the calibration of each of these. The trajectory is stored in order to move the checkerboard through the robot arm in front of the sensor on several perspectives.

visible and they can be accurately extracted (see section 4.3). Each time that an image is filtered and then stored, even the corresponding robot pose is stored. Hence, when the robot trajectory is concluded, all the robot poses and images stores are given as input to the final optimization process.
According to the considerations made in the introduction 1, the cameras intrinsic parameters are already accurately calibrated through a dedicated process, then in this thesis we focus only on the extrinsic calibration. Hence, the **calibration phase** is the final step of the whole process, which is the optimization necessary to determine the extrinsic parameters, namely the rototranslation from the robot base frame to the camera frame and the transformation between the checkerboard and the end-effector. This phase exploits the checkerboard images and the correspondent poses, in order to accomplish the calibration. Moreover it requires an *initial guess*[2] of the rototranslation matrices that have to be estimated as starting point for the iterative algorithm.
The calibration phase of our proposed method solves the eye-on-base calibration as a **non-linear optimization problem**. It generally consists in a search iterative algorithm which computes the solution to a least squares problem in each iteration through the Levenberg-Marquardt algorithm [60]. The eye-on-base calibration problem can be mathematically formalized in the following equation:

$$AX = ZB \tag{4.1}$$

where A and B are two relative poses transformations that are already known or provided, instead the matrices X and Z are the unknown transformations that have to be estimated. This relation can be derived, by observing the robotic workcell setup shown in Figure 4.3.

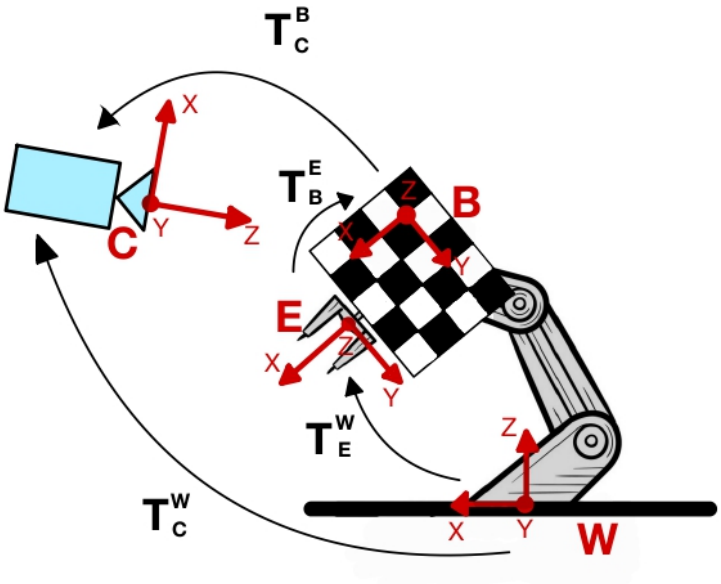


Figure 4.3: *Setup with a single camera and the robot arm with a checkerboard mounted on its end-effector, where are denoted the transformations between all reference frames.*

[2]It is a starting value for most optimization problems which use search algorithms, since those algorithms are mainly deterministic and iterative.

In particular, the reference system C denotes the camera frame, W identifies the world reference frame which corresponds to the robot base, B is the reference frame of the planar board and E denotes the end-effector reference frame. The outlined transformations between the reference frames are:

- $\mathbf{T}_C^W$ which denotes the rototranslation from the world reference frame W to the camera frame C;
- $\mathbf{T}_E^W$ describes the relative pose from the world's frame W to the end-effector reference frame E;
- $\mathbf{T}_B^E$ is the transformations from the end-effector's frame E to the checkerboard reference frame B;
- $\mathbf{T}_C^B$ describes the relative pose from checkerboard reference frame B to the camera's frame C.

In our proposed method, the optimization process is necessary to determine the two unknown transformations $\mathbf{T}_C^W$ and $\mathbf{T}_B^E$. All the additional transformations contribute to the formulation of the entire optimization problem, but many of them are constant and therefore not optimized during the entire calibration process, for example, the transformation matrix $\mathbf{T}_E^W$. This matrix is already given by the kinematics of the robot arm adopted for the experiments. This relative pose is provided with an high accuracy, in our specific case its deviation with respect to the real transformations $\mathbf{T}_E^W$ is less than 1 mm as described by the official data sheet of the manipulator, namely a Franka Emika panda manipulator used in our experiments [77].
According to the previous remarks and taking into account the set of N 3D control points $P_i^B = [X_i, Y_i, Z_i, 1]$ with $i = 0, \ldots, N$ that are the 3D corners with respect to the checkerboard reference frame denoted by B, the eye-on-base optimization problem described in the equation (4.1) can be generally rewritten in the following form:

$$AX = ZB \Longrightarrow T_E^W T_B^E P_i^B = T_C^W T_B^C P_i^B \qquad \text{with} \quad i = 0, \ldots, N \tag{4.2}$$

where $T_B^C = (T_C^B)^{-1}$, the matrices $A = T_E^W$, $B = T_B^C$ are already provided and the unknown matrices $X = T_B^E$, $Z = T_C^W$ have to be estimated. In particular in order to accomplish the optimization process, the problem is usually reformulated as a minimization problem in order to determine the unknown matrices by minimizing the squared distance between the reprojected 3D control points through two distinctive transformation chains: $T_E^W T_B^E P_i^B$ and $T_C^W T_B^C P_i^B$.
Hence the generic starting point is the minimization of the sum of squared difference between AX and ZB over M positions of the robot arm's trajectory:

$$\sum_{j=0}^{M-1} ||A_j X - Z B_j||^2 \tag{4.3}$$

This functional that has to be minimized can be reformulated by pre-multiplying both terms by T_W^C and expliciting the transformation matrices as follow:

$$\sum_{j=0}^{M-1} \sum_{i=0}^{N-1} \left|\left| T_W^C (T_E^W)_j T_B^E P_{ij}^B - (T_B^C)_j P_{ij}^B \right|\right|^2 \tag{4.4}$$

In our specific case, where we aim to minimize this functional in pixel terms through the re-projection of the 3D points P_{ij} on the image plane, we can derive the following final optimization problem related to a 2D distance minimization where the transformation T_B^C is not taken into account:

$$\underset{T_C^W, T_B^E}{\operatorname{argmin}} \sum_{j=0}^{M-1} \sum_{i=0}^{N-1} \left|\left| K\,[I|\mathbf{0}]\, T_W^C (T_E^W)_j T_B^E P_{ij}^B - p_{ij} \right|\right|^2 \tag{4.5}$$

where the matrix K corresponds to the matrix of intrinsic parameters that in our works is already provided, p_{ij} is the i^{th} detected corner on the j^{th} image of the checkerboard through specific functions based on Harris corner-detector [78, 79] and $K\,[I|\mathbf{0}]\, T_W^C (T_E^W)_j T_B^E P_{ij}^B$ is the projection of points P_{ij} in image j.
The proposed method for eye-on-base calibration it was solved as an optimization problem, namely minimizing the 2D distance between the reprojected 3D points P_{ij} on the image plane and the correspondent detected 2D corners. It can be summarized by the following minimization problem:

$$\underset{T_C^W, T_B^E}{\operatorname{argmin}} \sum_{j=0}^{M-1} \sum_{i=0}^{N-1} \left|\left| p_{ij}^P - p_{ij}^D \right|\right|^2 = \underset{T_C^W, T_B^E}{\operatorname{argmin}} \sum_{j=0}^{M-1} \sum_{i=0}^{N-1} \left|\left| \begin{pmatrix} u_x \\ u_y \end{pmatrix}_{ij}^P - \begin{pmatrix} u_x \\ u_y \end{pmatrix}_{ij}^D \right|\right|^2 \tag{4.6}$$

where P denotes the projected 2D points on the image plane, so $p_{ij}^P = K\,[I|\mathbf{0}]\, T_W^C (T_E^W)_j T_B^E P_{ij}^B$ and D refers to the detected corners, namely $p_{ij}^D = p_{ij}$.
The algorithm in pseudo-code which describes the just proposed calibration process, according to the whole calibration framework of the Figure 4.2 is shown in the algorithm 1.

Algorithm 1 Single camera eye-on-base calibration

$K \leftarrow$matrix of intrinsic parameters;
$X_0, Z_0 \leftarrow$initial guess of T_B^E and T_W^C;
$M \leftarrow$ number of predefined trajectory poses;
for $i = 0 : M - 1$ **do**
  Move the robot arm to pose i;
  Detect checkerboard & extract corners;
  Filter(extracted corners);
  if $filter == true$ **then**
    Save image i;
    Save pose i;
  end if
end for
Robot trajectory concluded;
Given saved calibration images, correspondent poses and the initial guess;
Calibration phase $\rightarrow \underset{T_C^W, T_B^E}{\operatorname{argmin}} \sum_j \sum_i \left|\left| p_j - d_j \right|\right|^2$

Filter(. . . section 4.3 . . .);

4.2 Multi-camera eye-on-base calibration

As it is introduced in previous sections the main aim of this thesis was to propose a multi-camera eye-on-base calibration method.
In section 4.1 is proposed the single camera calibration which can be adopted to calibrate a robotic workcell equipped by a camera network, but implementing a single calibration for each camera with respect to the robot arm, making the whole process very time-consuming. In this section, a novel method for multi-camera eye-on-base calibration is described. It computes the whole calibration of a robotic workcell, namely determines the extrinsic parameters of the camera network with respect to the robot arm through a single optimization problem.
The whole calibration framework can be summarized in Figure 4.4, where it is highlighted that the multi-camera calibration provides the calibration of the whole camera network and the manipulator through a single optimization process.

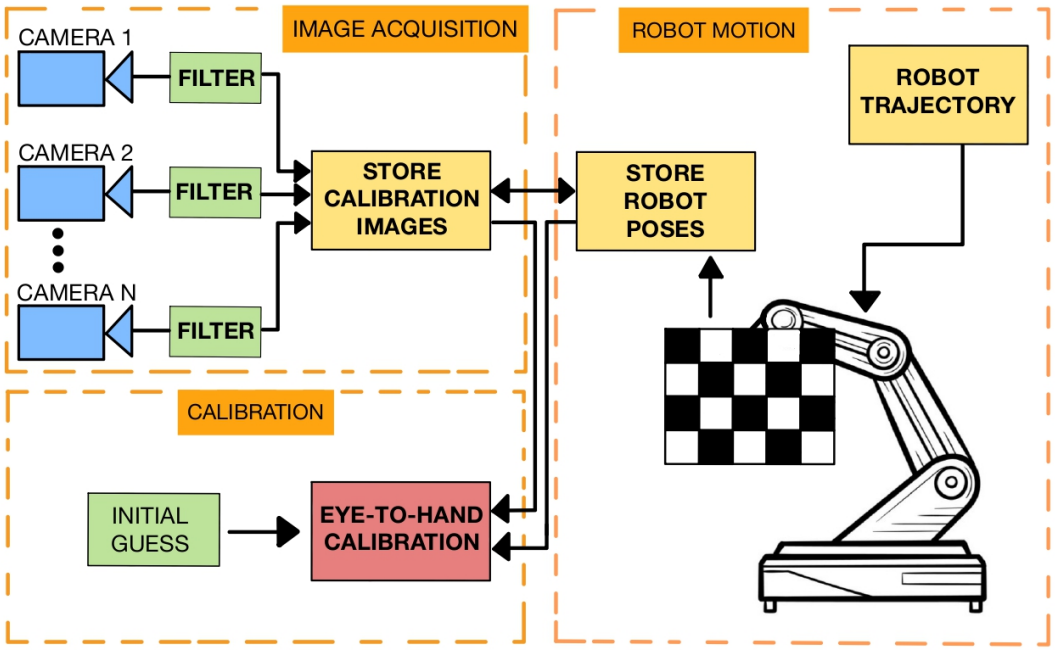


Figure 4.4: *The whole process for multi-camera eye-on-base calibration. It is divided in three subprocesses, one dedicated to the image acquisition by all cameras, one related to the robot motion and the other dedicated to the optimization process for the robotic workcell calibration.*

It can be observed from Figure 4.4 that even in the proposed method for multi-camera eye-on-base calibration, the whole process is split in three different modules which works together. As it was described for the single camera calibration, it can be outlined the **robot motion phase** which controls the execution of a predefined trajectory by the robot arm. In particular the trajectory is defined in order to move the checkerboard mounted on the robot in front of each camera around the robot workspace and moreover the trajectory is

determined in order to have some robot poses in which the checkerboard can be seen simultaneously by couples of cameras. The **image acquisition phase** works in parallel with the robot motion, and it handles the images capture process. In particular at each pose achieved by the robot, the cameras are programmed to capture the images of the checkerboard if and only if they are able to detect the checkerboard. In the event that a camera cannot observe the calibration checkerboard it does not capture the image. Instead when one or more cameras can simultaneously detect a checkerboard, those cameras capture the images and store the pictures in a dedicated folder for each sensor, and the correspondent pose of the robot is stored in a proper folder.
In order to accomplish this phase, it was implemented a state machine[3] to handle the robot and camera behaviours during the image acquisition. The state-machine structure is summarized in the Figure 4.5.

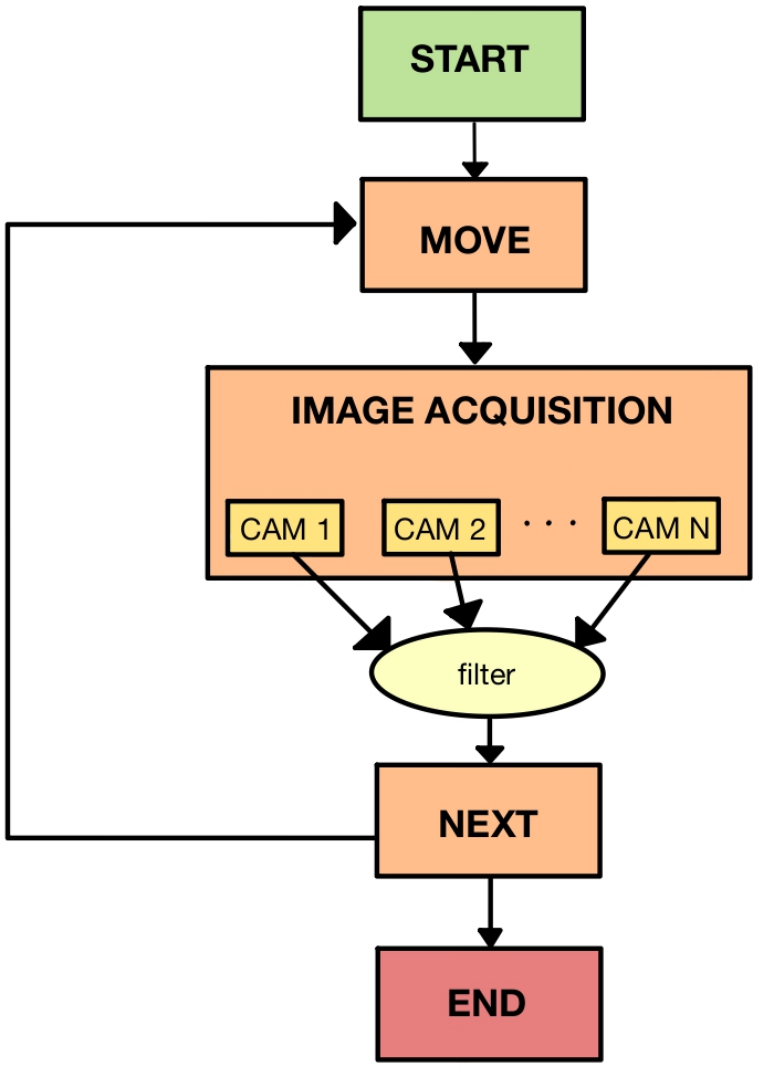


Figure 4.5: *State machine implemented in order to handle the image acquisition phase.*

[3]State machine is a task-level architecture for rapidly creating complex robot behavior.

Each state of the state-machine in Figure 4.5 is defined by the blocks, which have the following roles:

- **START**: the robot is moved to the first position of the trajectory;
- **MOVE**: the robot is moved to the next pose of the trajectory;
- **IMAGE ACQUISITION**: each camera which can detect the checkerboard, captures the image that will be filtered, if correctly detected the image is stored with its corresponding robot pose;
- **NEXT**: it manage the trajectory counter updating, if there exists other poses to be executed, it goes back to MOVE state, else the image acquisition moves to **END** state because the trajectory is concluded.

Then, the stored calibration images and the stored robot poses are given as input to the module related to optimization process. This multi-calibration process is implemented to handle images coming from different cameras and in particular it is realized in order to determine the extrinsic parameters of all camera with respect to each other and with respect to the robot base. Moreover, as for the single camera calibration, this method requires an initial guess as well of the rototranslation matrices that have to be estimated.
This implementation is a novel method, which consists in a single optimization process to calibrate more cameras in a robotic workcell, avoiding to accomplish a calibration for each camera and then to adopt N times the single camera calibration process shown in Figure 4.2.
The multi-camera eye-on-base calibration can be mathematically formalized as reported in equation (4.1) where the unknown matrices X and Z can be derived by observing the robotic workcell setup shown in Figure 4.6.
With respect to the robotic workcell with a single camera, more cameras are added, and their reference frames are denoted with C_1, C_2 and C_3. Moreover the transformations between cameras are denoted by $T_{C_2}^{C_1}$, $T_{C_3}^{C_2}$ and $T_{C_1}^{C_3}$, which have to be estimated. The transformations between the cameras and the planar board are specified by $T_{C_i}^{B}$ with $i = 1, 2, 3$ but they are not exploited for our optimization problem. The transformation matrix T_E^W between the robot base frame and the end effector frame, is given by the robot kinematics. Furthermore the transformations $T_{C_i}^W$ between the cameras and the world reference frame and the transformation T_B^E are the relative poses that have to be estimated through the optimization process.
According to these transformations, considering the set of N control points $P_i^B = [X_i, Y_i, Z_i, 1]$ with $i = 0, \ldots, N$ which are the 3D corners on the checkerboard reference frame and whereas there are K cameras and it can happen that two cameras C_m and C_t can simultaneously detect the checkerboard, the eye-on-base calibration described in the equation 4.1 can be formalized as follow:

$$AX = ZB \Longrightarrow T_E^W T_B^E P_i^B = T_{C_m}^W T_{C_t}^{C_m} T_B^{C_t} P_i^B \tag{4.7}$$

where $(m, t)_i$ are the couples of camera that can simultaneously detect the checkerboard. Moreover, the transformation matrices that have to be estimated are $X = T_B^E$ from the end-effector to the checkerboard, and the transformation matrix $Z = T_{C_m}^W T_{C_t}^{C_m}$ which can

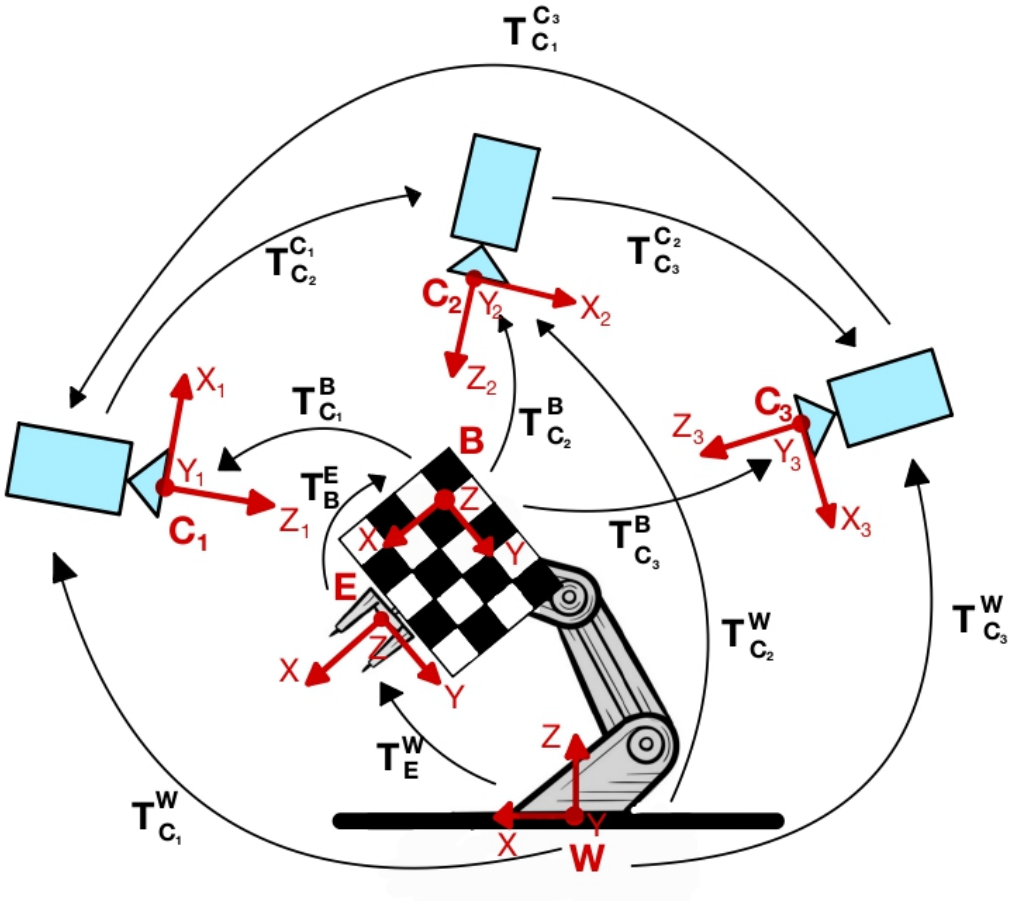


Figure 4.6: *Setup with a camera network of three cameras and the robot arm with a checkerboard mounted on its end-effector, where are denoted the transformations between all reference frames.*

be decoupled in $T_{C_m}^{W}$ and $T_{C_t}^{C_m}$. These two matrices correspond to the relative pose between the generic camera m and the robot base W and the transformation between the camera t and the camera m.

As in the single camera calibration of section 4.1, in order to accomplish the optimization process, the problem is formulated as a minimization problem. The main aim is minimizing the root mean square error between the reprojected 3D points into the 2D points on the image plane and their corresponding detection.

According to the equation 4.7 the functional can be rewritten as follow:

$$\sum_{j=0}^{M-1}\sum_{i=0}^{N-1}\left|\left|T_{C_m}^{C_t}T_{W}^{C_m}(T_{E}^{W})_j T_{B}^{E} P_i^B - T_{B}^{C_t} P_i^B\right|\right|^2 \tag{4.8}$$

where M is the number of trajectory poses and N is the number of the control points on the checkerboard.

Hence, in order to formulate this functional in terms of pixels and to derive the final optimization problem, it is considered the projection of 3D corners onto the image plane through the matrix of intrinsic parameters and the second term is the 2D corners detected p_{ij} for

each image j by the camera t. Thus, the final optimization problem can be written as follow for each pair of cameras (m,t) with $m,t = 1..K$:

$$\underset{T_{C_m}^W, T_B^E, T_{C_t}^{C_m}}{\operatorname{argmin}} \sum_{j=0}^{M-1} \sum_{i=0}^{N-1} \left|\left| K_t \left[\mathbf{I}|0\right] T_{C_m}^{C_t} T_W^{C_m} (T_E^W)_j T_B^E P_i^B - (p_{ij})_t \right|\right|^2 \tag{4.9}$$

where K_t refers to the set of intrinsic parameters of the camera t, $K_t \left[\mathbf{I}|0\right] T_{C_m}^{C_t} T_W^{C_m} (T_E^W)_j T_B^E P_i^B$ is the projection of the 3D point P_i^B on the image plane of the camera t and $(p_{ij})_t$ is the i^{th} 2D corner detected by the camera t at the robot pose j.
Through this single optimization problem which allows to calibrate the whole camera network in the robotic workcell provides the optimized transformation matrices between each camera to the others $T_{C_i}^{C_j}$ with $i,j = 0,\ldots,K$ and the transformation between each camera to the robot's base frame $T_W^{C_i}$. Moreover it is optimized one single matrix T_B^E from the end-effector to the planar board. In particular, in this proposed method it is provided a single common optimization of the matrix T_B^E, unlike the single camera calibration of the previous section 4.1 which optimizes one T_B^E for each performed camera calibration. Furthermore, it has to be highlighted that if the checkerboard at pose j can be detected by a single sensor it is added to the optimization problem the functional described in equation (4.4) as a single camera eye-on-base calibration of the previous section 4.1. So, considering K the number of cameras that can detect the checkerboard at the j^{th} robot pose, the final functional that has to be minimized for the optimization problem can be rewritten as follow:

$$\sum_{j=0}^{M-1} \begin{cases} \sum_{i=0}^{N-1} \left|\left| K \left[I|\mathbf{0}\right] T_W^C (T_E^W)_j T_B^E P_{ij}^B - p_{ij} \right|\right|^2 & \text{if} \quad K = 1 \\ \sum_{i=0}^{N-1} \left|\left| K_t \left[\mathbf{I}|0\right] T_{C_m}^{C_t} T_W^{C_m} (T_E^W)_j T_B^E P_i^B - (p_{ij})_t \right|\right|^2 & \text{if} \quad K = 2 \\ \sum_{\forall(m,t)} \sum_{i=0}^{N-1} \left|\left| K_t \left[\mathbf{I}|0\right] T_{C_m}^{C_t} T_W^{C_m} (T_E^W)_j T_B^E P_i^B - (p_{ij})_t \right|\right|^2 & \text{if} \quad K > 2 \end{cases} \tag{4.10}$$

In particular, it is highlighted that in the event that more than two cameras can simultaneously see the checkerboard ($K > 2$), the optimization just described is computed for each couple of cameras (m,t) with $m,t = 0,\ldots,K-1$ which detect the calibration target.

The pseudo-code algorithm of the multi-camera eye-on-base calibration is reported in algorithm 2. In particular. it can be highlighted that it manages the event that more cameras can detect the checkerboard, but also when only a single camera can see the checkerboard, adding to the minimization problem the residual term $\sum_{i=0}^{N-1} \left|\left| K \left[I|\mathbf{0}\right] T_W^C (T_E^W)_j T_B^E P_{ij}^B - p_{ij} \right|\right|^2$ derived from single camera eye-on-base calibration 4.5. Therefore the proposed multi camera calibration method works even if at each pose the checkerboard is detected by only one camera, leading back to the single camera calibration. However, in order to exploit the advantage of the multi-camera eye-to-hand calibration, so optimize the transformations among cameras, are required robot poses where the checkerboard can be simultaneously detected by more sensors.

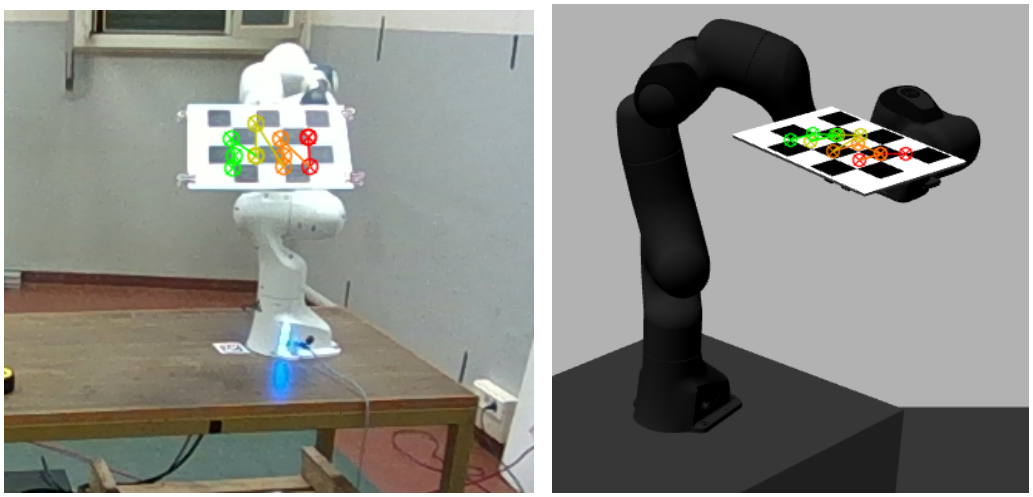
Algorithm 2 Multi camera eye-on-base calibration

$N \leftarrow$number of cameras;
$K_i \leftarrow$ with $i = 0, \ldots, N-1$ matrix of intrinsic parameters of camera i;
$X_0 \leftarrow$initial guess of T_B^E;
$Z_i \leftarrow$ with $i = 0, \ldots, N-1$ initial guess of $T_W^{C_i}$;
$Y_{ij} \leftarrow Z_i(Z_j)^{-1}$ with $i, j = 0, \ldots N-1$ initial guess of $T_{C_i}^{C_j}$
$M \leftarrow$ number of predefined trajectory poses;
$count \leftarrow 0$
$res \leftarrow 0$
for $i = 0 : M-1$ **do**
  Move the robot arm to pose i;
  for $j = 0 : N-1$ **do**
    Camera j try to detect checkerboard
    if $detection == true$ **then**
      Extract corners;
      Filter(extracted corners);
      if $filter == true$ **then**
        Save index j
        $count \leftarrow count + 1$
        Save image i on dedicated folder of camera j;
        Save pose i on dedicated folder of camera j;
      end if
    end if
  end for
  if $count = 1$ **then**
    $res \leftarrow res + \left|\left| K_j \, [I|\mathbf{0}] \, T_W^{C_j} (T_E^W)_i T_B^E P_i^B - p_i \right|\right|^2$
  end if
  if $count > 1$ **then**
    for $\forall (m, t)$ **do**
      $res \leftarrow res + \left|\left| K_t \, [\mathbf{I}|0] \, T_{C_m}^{C_t} T_W^{C_m} (T_E^W)_j T_B^E P_i^B - (p_{ij})_t \right|\right|^2$
    end for
  end if
end for
Robot trajectory concluded;
Given stored calibration images;
Given correspondent poses;
Given the initial guesses;
Calibration phase $\rightarrow \underset{T_{C_m}^W, T_B^E, T_{C_t}^{Cm}}{\text{argmin}} \; res$

Filter(... section 4.3 ...);

4.3 Filter function

As it is mentioned in sections 4.1 and 4.2 related to proposed single-camera and multi-camera calibration method respectively, in order to guarantee a correct checkerboard detection it is implemented a **filter function**. In particular, this function is necessary to check if the checkerboard corners extracted are properly and accurately identified.
A common issue of the functions adopted for the checkerboard corner extraction [80] is the false positives as described in [81]. This corners extraction functions gives as results the control points position of a calibration pattern if and only if they are accurately detected. However it happens that in some events the functions used to extract corners gives the positions of internal corners of the checkerboard as result even if the detection was completely wrong. A practical example is shown in Figure 4.7.



(a) *Wrong checkerboard detection due to high light reflections.*

(b) *Wrong checkerboard detection due to the high checkerboard tilt.*

Figure 4.7: *Example of false positive, namely wrong checkerboard corners extraction.*

It can be easily observed that the corner extraction is wrong. The main issue is that the corners position given is deviated with respect their real position on the image. This issue can be due by different elements, for instance light reflections, shadows, checkerboard too far away from the camera or high checkerboard tilt. In the event of checkerboard too far away from the camera, it is accurately proposed a novel trick in section 6.3 to improve the resolution of the image and overcome this issue. However, regardless the causes, this wrong extraction has to be avoided because it adversely affects the final calibration results. Hence, the filter function is implemented in order to avoid these false positives and then to remove from the calibration process images that are affected by these mistakes. In particular the filter checks that the control points of a generic calibration planar target are arranged as a grid (e.g. checkerboard corners, centers of circles grid, etc.). In order to accomplish this monitoring, the function requires as input the corners extracted and then for each corners row and column it verifies that the image points are perfectly aligned. This is necessary to ensure that the points respect the grid property.

Moreover there are some event where the corners extracted could appear aligned even if they are completely wrongly extracted and the filter it would not able to detect the mistakes, as shown in Figure 4.8, then a further monitoring implementation on the filter was adopted.

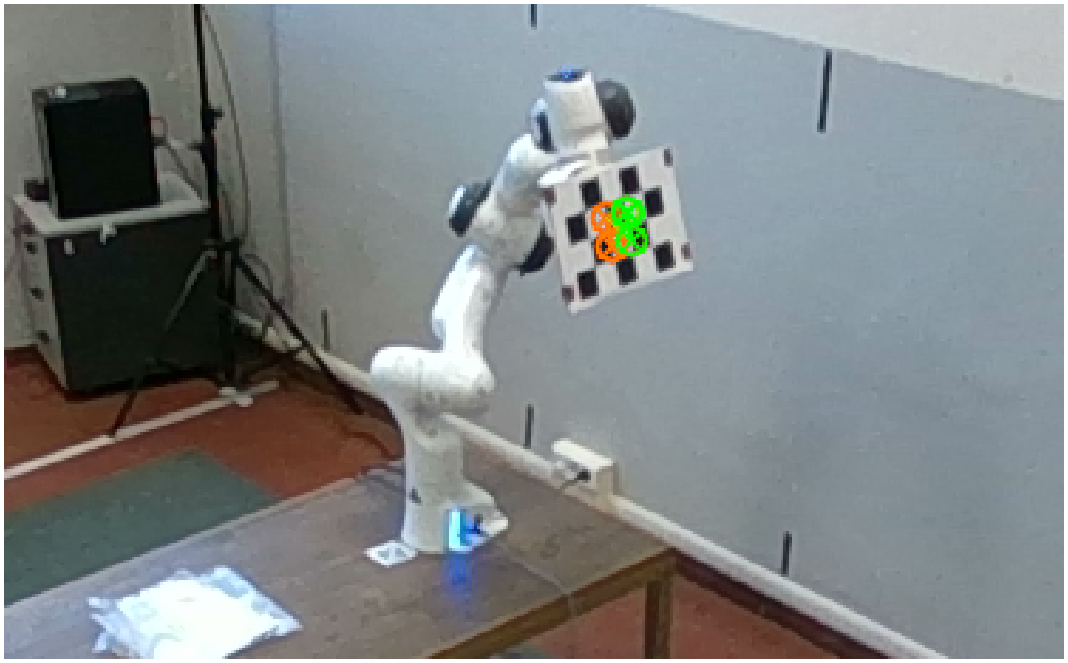


Figure 4.8: *Wrong corners detection due to the high distance between camera and checkerboard. The control points are almost aligned even if they are wrongly detected.*

In particular, in addition to the alignment check, the filter function monitors the distance between each couple of corners. Hence for each row the filter verifies that the corners are placed almost[4] at the same distance and it does the same operation for each column. In this way it is avoided the false positive such as the one shown in Figure 4.8, because the filter can easily verifies that the some checkerboard corners are overlaid not respecting the distance property.
In the section 6 related to the experiments it will be accurately shown that the filter widely improves the calibration performances, removing the images in which the corners are wrongly detected.
The algorithm of the filter function is reported in pseudo-code in 3.

[4]They cannot perfectly be at the same distance, since there can be a little noise on the image and moreover it has to be considered the checkerboard tilt.

Algorithm 3 Filter function

$N \leftarrow$ number of rows
$M \leftarrow$ number of columns
$filter\_state \leftarrow false$
$corners\_det \leftarrow$ number of corners detected
$d_m \leftarrow 0$
if $corners\_det == M \cdot N$ **then**
  for $i = 1 : N$ **do**
    $d_{tot} \leftarrow ||p_{i1} - p_{iM}||^2$
    for $j = 1 : M - 1$ **do**
      $l_j = ||p_{ij} - p_{ij+1}||^2$
      $d_m \leftarrow d_m + ||p_{ij} - p_{ij+1}||^2$
    end for
    if $d_m \simeq d_{tot}$ & $l_j \simeq l_{j+1}$ with $j = 1 : M - 1$ **then**
      $filter\_state \leftarrow true$
    else
      $filter\_state \leftarrow false$
    end if
  end for
  for $j = 1 : M$ **do**
    $d_{tot} \leftarrow ||p_{j1} - p_{jN}||^2$
    for $i = 1 : N - 1$ **do**
      $l_i = ||p_{ji} - p_{ji+1}||^2$
      $d_m \leftarrow d_m + ||p_{ji} - p_{ji+1}||^2$
    end for
    if $d_m \simeq d_{tot}$ & $l_i \simeq l_{i+1}$ with $i = 1 : N - 1$ **then**
      $filter\_state \leftarrow true$
    else
      $filter\_state \leftarrow false$
    end if
  end for
end if
return $filter\_state$

Chapter 5

System setup

In order to accomplish the thesis experiments and testing the proposed calibration method, a suitable setup was implemented, both in terms of *software* and *hardware*.
The adopted system software is mainly based on the **Robot Operating System**[1] (ROS) which is a set of software libraries and tools that allows to build robot apps [82, 83]. It is an open source software development kit for robotics applications running on top of Ubuntu. In particular, it is a software system which provides a built-in and well-tested messaging system that allows the communication between distributed nodes via a publish/subscribe pattern. In our specific case this software allowed us to build an interconnected system, which can manage and connect cameras and the robot arm. The ROS programs developed for our experiments are created with two programming languages: Python and C++.
Furthermore a part of thesis experiments were computed in a virtual environment in order to speed up the whole process and acquire quantitative experiments as it is accurately described in section 6. For this purpose it was adopted **Gazebo** which is a well-designed simulator for testing algorithms, designing robots, performing regression, and training AI system using realistic scenarios [84]. In order to handle the computer vision algorithm for instance for the corner detection, it was exploited the **OpenCV** library [85]. which is an open source computer vision and machine learning software library.
Regarding the hardware elements adopted in thesis experiments, different cameras are used which are accurately described in section 5.1, in particular their technical specifications and their differences are analyzed. In section 5.2 the manipulator adopted for our experiments is described, and the main differences with other manipulators commonly used are analyzed.

5.1 Camera

In order to perform our proposed multi-camera calibration method we tested the algorithm adopting different kind of cameras for the setup. As it is described in section 6 related to the experiment description, we have computed a qualitative analysis and comparison among the cameras, in order to figure out which kind of sensor is more suitable for camera calibration and then more convenient to be adopted in human-robot collaboration tasks. In particular the cameras adopted are illustrated in Figure 5.1.

[1] https://www.ros.org/.

(a) *Kinect V2.* (b) *Depth camera D455.* (c) *LiDAR camera L515.*

Figure 5.1: *Cameras tested on the thesis experiments.*

In our experiments we tested three different kind of cameras: *Kinect V2*, *Intel RealSense depth camera D455* [86] and *Intel RealSense LiDAR camera L515* [87].
The **Kinect V2** is an RGB-D camera[2]. It contains an RGB sensor and infrared projectors and detectors that map depth through *time of flight* calculations, which can be used to perform real-time gesture recognition and body skeletal detection, among other capabilities [88]. In particular the Time-of-Flight principle (ToF) is a method for measuring the distance between a sensor and an object, based on the time difference between the emission of a signal and its return to the sensor, after being reflected by an object [89]. The Kinect sensor also contain microphones that can be used for speech recognition and voice control.
The **depth camera D455** is a stereo-based depth camera of Intel. The stereo vision is similar to 3D perception in human vision and it is based on triangulation of rays from multiple viewpoints. The stereo vision approach can provide full field of view 3D measurements using two machine vision cameras such as in the depth camera D455. Hence the depth perception is achieved by using two sensors a set distance apart to triangulate similar pixels from both 2D planes.
The **LiDAR** is the acronym for "Light Detection And Ranging" and it uses eye-safe laser beams to create a 3D representation of the surveyed environment. Its technology allows to determine the distance by targeting an object with a laser and measuring the time for the reflected light to return to the receiver. In particular, a typical LiDAR sensor emits pulsed light waves from a laser into the environment. These pulses bounce off surrounding objects and return to the sensor. The sensor uses the time it took for each pulse to return to the sensor to calculate the distance it traveled. Repeating this process millions of times per second creates a real-time 3D map of the environment [90].
All the three cameras analyzed and adopted for thesis experiments are based on different vision technologies, but they are linked for their capability of depth perception. In this thesis, even if the proposed camera calibration method is based on the calibration of the RGB cameras, RGB-D cameras are adopted since the main aim is the calibration of a camera network in a robotic workcell for human robot collaboration tasks, hence, it was necessary the usage of cameras that was able to perceive the depth. The main requirement for the cameras used for the experiments is the ROS support, which allows the correct communication among more sensors and the manipulator.

[2]RGB-D camera are a specific type of depth sensing devices that work in association with a RGB camera, that are able to augment the conventional image with depth information.

In the table 5.1 are shown more accurately the technical specifications of the Kinect V2, depth camera D455 and the LiDAR camera L515.

	Kinect V2	**Depth camera D455**	**LiDAR camera L515**
Sensor type	Time of Flight	Stereoscopic	LiDAR
RGB resolution	1920 × 1080	1280 × 800	1920 × 1080
Depth resolution	512 × 424	1280 × 720	Up to 1024 × 768
RGB frame rate	30 fps	30 fps	30 fps
Depth frame rate	30 fps	Up to 90 fps	30 fps
Depth range	0.5 to 4.5 meters	0.6 to 6 meters	0.25 to 9 meters
RGB FoV (h × v)	84.1° × 53.8°	90° × 65°	70° × 43°
Depth FoV (h × v)	70.6° × 60.0°	87° × 58°	70° × 55°

5.2 Robot arm

For the thesis experiments, the manipulator has a main role on performing and testing the single and multi camera calibration methods. As it is accurately described in section 4 the robot arm is adopted to automatize the process of the image acquisition, by executing predefined trajectories and moving the checkerboard in front of the cameras. The advantage of the robot usage is its accurate repeatability, which allows the execution of the same trajectory several times in order to repeat calibration methods with different sensors according to the same chessboard poses.
There are several robot arms suitable for human-robot collaboration tasks such as Kuka LBR iiwa [91], the UR16 [92] and the Franka Emika Panda [93].

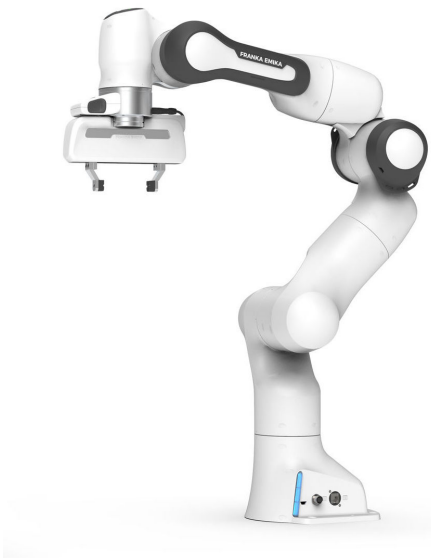


Figure 5.2: *Franka Emika Panda robor arm adopted for thesis experiments.*

In particular the robot arm shown in Figure 5.2 is the Franka Emika Panda robot. It has an high torque sensitivity on all of the 7 axes and it is ideal for optimizing low-weight collaborative processes such as picking, placing, and testing. Considering that it can lift a payload up to 3kg, it has the flexibility to automate almost any manual task [93].
This collaborative robot have an high flexibility thanks to their 7 degrees of freedom, which for our purpose allows to easily execute a complex predefined trajectory as well. Moreover this cobot is known for its accuracy on motion repeatability which is about 0.1 mm.
Hence for our experiments it is adopted the Panda Franka Emika robot arm, however the proposed camera calibration methods can be performed also with other manipulators such as the Kuka, the UR16 or others.
In particular, in order to attach the checkerboard on the end-effector of the robot arm, so that it would remain fixed, it was realized a dedicated custom 3D-printed mount composed by 2 elements shown in Figure 5.3.

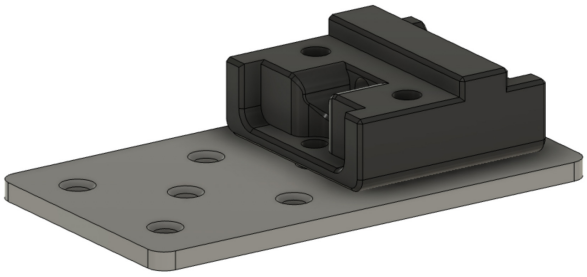
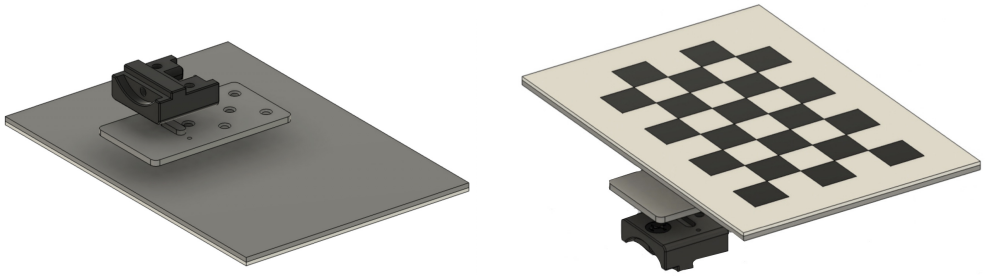


Figure 5.3: *Custom 3D printed mount, composed by two elements necessary to attach checkerboard on the Panda robot arm.*

It is designed in order to perfectly matched with the structure of the end-effector from on side, and from the other side was suitable to attach the checkerboard, as illustrated more accurately in Figure 5.4



(a) *3D printed mount with checkerboard seen from the bottom.*

(b) *3D printed mount with checkerboard seen from the top.*

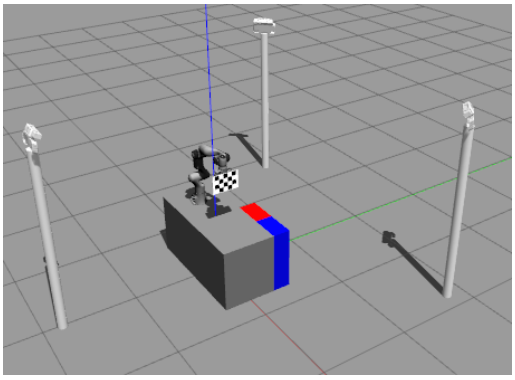
Figure 5.4: *Custom 3D printed mount adopted for to attach the checkerboard to the robot arm.*

Chapter 6

Experimental results

The single and multi camera hand-to-eye calibration processes proposed in section 4 were tested on several and different experiments, to determine their calibration accuracy and their applicability to different scenarios. For the experiments, we considered a dedicated robot workcell composed equipped by an Emika Franka Panda manipulator and several type of cameras such as Kinect v2 and Realsense cameras. Experiments have been performed both in simulation using Gazebo, and in real setups. Figure 6.1 illustrates two different real testbeds for the robot workcell setup in the Robot Vision Lab and in the simulator Gazebo.

(a) *Robotic workcell equipped by a camera network of depth sensors D455, arranged around the robot arm placed on the center of the workcell.*



(b) *Robotic workcell designed for the simulator Gazebo equipped by a camera network, whose sensors are arranged around the robot arm placed on the center of the workcell.*

Figure 6.1: *Example of two different workcell setups of our robotic workcell in Robot Vision Lab and in the simulator Gazebo.*

In this chapter a set of significant performed experiments is accurately described, analyzing in detail the configuration considered and the achieved results. In particular, Section 6.1 focuses on the simulation experiments which aim to evaluate the calibration methods considering different calibration patterns and sizes of the camera network (i.e. distances of the cameras from the robot arm).
Section 6.2 focuses on the real experiments, reporting the results achieved with the pattern

which was proved to achieve the best calibration performances, and tested with the three cameras: *Microsoft Kinect V2*, *Intel RealSense depth camera D455* and *Intel RealSense LiDAR camera L515* described in section 5. Moreover, in section 6.2.2 and 6.2.3 our proposed method is compared and analyzed with respect to other single and multi calibration methods respectively, that are adopted in the state-of-art.
A main issue that has arisen from the experiments is the difficulty of calibrating large networks, due to the insufficient resolution to correctly detect calibration patterns in the image at large distances. Therefore in Section 6.3 we present the calibration results obtained by introducing an image resize step. As demonstrated by the obtained results, this simple trick allows to significantly increase the size of the camera network, making the proposed methods usable with high accuracy even in very large robotic workcells.

6.1 Simulated experiments

As previously mentioned, in this section it is analyzed the set of simulated experiments performed on the simulator Gazebo.
This software can faithfully simulate the real world and it is adopted to accurately reproduce the robotic workcell of our experiments. It allows to perform quantitative experiments of our proposed calibration method, by arranging the cameras network in the most suitable position for each type of experiment, simultaneously providing a specific ground truth of the transformation matrices which describe the relative poses among sensors and the robot arm. It even allowed to avoid to always adopt the real robot arm and then to speed up the calibration process. It can be done by exploiting an higher speed for the trajectory execution which can not be implemented on the real robot for real experiments, since it would be more difficult to keep under control increasing the risk of accident and unexpected collisions with humans or other objects.

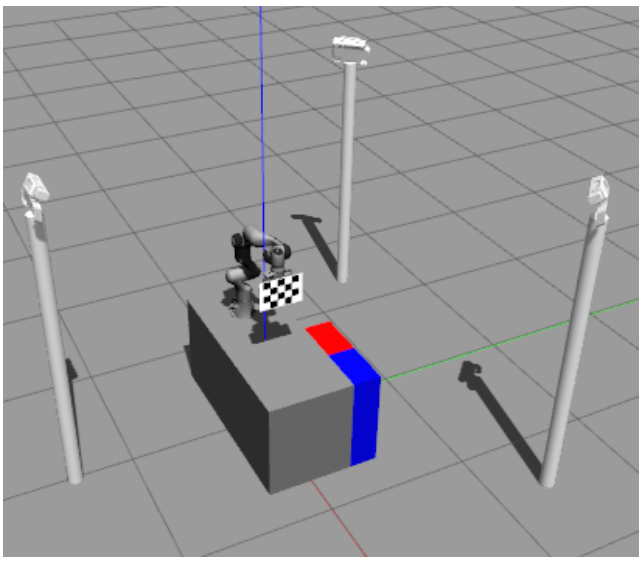
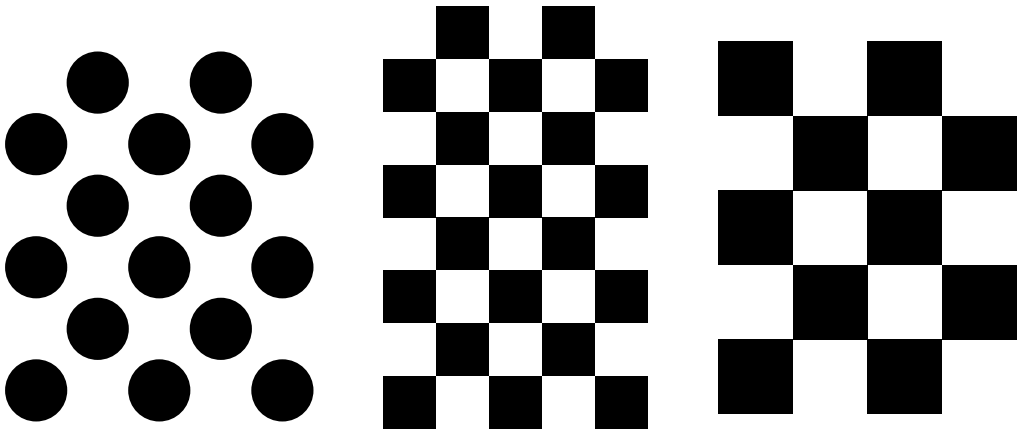


Figure 6.2: *Robotic workcell setup designed for the experiments on simulator Gazebo.*

6.1.1 Analysis with different planar patterns

A first experiments group was focused on the analysis of the calibration with different planar targets, shown in Figure 6.3, in order to determine how much a specific pattern with respect to the others contributes to improve or get worse the calibration performances. In particular, to accomplish these experiments they are attached to the end-effector of the robot arm through the dedicated custom mount described in Section 5.



(a) *Checkerboard with asymmetric circles grid.* (b) *checkerboard with small squares.* (c) *checkerboard with big squares.*

Figure 6.3: *Planar patterns analyzed.*

As introduced in section 1 the above planar patterns sizes are suitable to be printed on an A4 size paper (210 mm $\times$297 mm), so that the checkerboard can be easily attached to the robot and making the robot's movement smoother during the trajectory execution by avoiding collisions with the surrounding environment. In particular the analyzed calibration patterns are the following:

- **Asymmetric circles grid**, Figure 6.3a, with an asymmetric grid of 5×3 circles, whose diameter is $d = 4$ cm;
- **Small squares grid** of Figure 6.3b with a grid of 5×8 squares of size 3 cm;
- **Big squares grid** of Figure 6.3c with a grid of 4×5 squares of size 4.6 cm.

To accomplish an exhaustive and comprehensive comparison, each one of the three planar targets is attached to the end-effector of the robot arm and it is tested with the proposed single camera calibration method. These experiments are performed with the single camera calibration aiming to evaluate the accuracy of the detection of different patterns. In particular, at each of these experiments, the manipulator executes the same trajectory of 20 predefined poses in front of the sensors belonging to the camera network which are placed on the same side of the robot arm at a variable distance d as shown in Figure 6.4.
It was adopted a predefined trajectory with a small amount of poses where the checkerboard is maintained almost stationary and it is only tilted on different orientations with respect

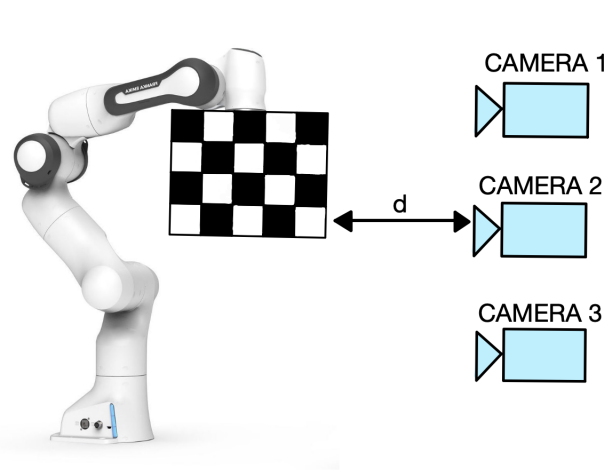
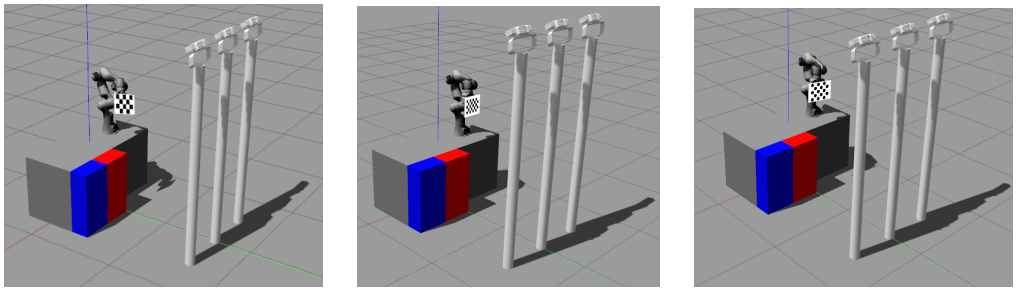


Figure 6.4: *Robotic workcell setup adopted to perform the experiments with different planar patterns.*

to the camera. This is done in order to analyze which pattern allows the best calibration although a small amount of images was acquired, testing their detection robustness and accuracy. When the trajectory is concluded, then the images are completely acquired and the calibration is performed according to the approaches described in Section 4. The processes are repeated one time for each target, by adopting the simulated robotic workcell with three simulated cameras (e.g. Kinect), placed on one side of the panda arm, facing the manipulator positioned on the center of the workspace, as illustrated in Figure 6.5.



(a) *Robotic workcell with checkerboard with big squares.*

(b) *Robotic workcell with small squares checkerboard.*

(c) *Robotic workcell with checkerboard with circles grid.*

Figure 6.5: *Simulated environment adopted with the three calibration patterns analyzed.*

Moreover to implement a quantitative and a qualitative analysis, each planar target is used for the eye-on-base calibrations of the cameras placed at each experiment at an increasing distance d from the robot base (see Figure 6.4). Starting from the first experiment where the cameras are placed at a distance $d = 150\,\text{m}$ from the robot base which is the shortest distance required so that an human operator can move on the workcell avoiding accidental collisions to the cameras, until the last experiments where the cameras achieve the distance $d = 2\,\text{m}$.
The evaluation of the eye-on-base calibration method with different calibration targets was

accomplished according to the **re-projection error** (*pixel*). Indeed, this evaluation error is the most used generic metric to analyze the calibration performances and it will be used on the whole experiments process of this thesis. It consists in computing the root mean squared distances between the 3D control points P_j^B of the checkerboard on the j^{th} robot's pose re-projected on the image plane through the transformations chain $K[I|\mathbf{0}]T_W^C(T_E^W)_jT_B^EP_j^B$, and the detected 2D control points p_j by the camera at j^{th} pose of the robot. In this way, the computed error, reliably describes how accurate the calibration is, since the error is evaluated according the two estimated matrices T_W^C and T_B^E, as follow:

$$err = \sqrt{\frac{\sum_{j=0}^{M-1}\left(K[I|\mathbf{0}]T_W^C(T_E^W)_jT_B^EP_j^B - p_j\right)^2}{M}} = \sqrt{\frac{\sum_{j=0}^{M-1}\left(\hat{y}_j - y_j\right)^2}{M}} \tag{6.1}$$

where M is the number of the trajectory's poses executed and then of the corresponding images analyzed for the calibration process, $\hat{y}_j$ are the corners projected through the estimated matrices, and y_j are the detected corners.
In the following tables are reported the reprojection errors achieved by each camera for each planar target with our proposed single camera hand-eye calibration method, according to the distance between the cameras and the checkerboard. Moreover, since it was adopted the single camera hand-eye calibration method for three cameras, in the first three table are independently reported the achieved results by each of three sensor and in the last table are reported the average reprojection error obtained with all the three sensors.

CAMERA 1	**Small squares grid**	**Big squares grid**	**Circles grid**
Distance 1.5 m	**0.075** pix	0.079 pix	0.110 pix
Distance 1.6 m	0.080 pix	**0.077** pix	0.098 pix
Distance 1.7 m	0.081 pix	**0.078** pix	0.098 pix
Distance 1.8 m	0.088 pix	**0.075** pix	0.099 pix
Distance 1.9 m	0.096 pix	**0.077** pix	0.095 pix
Distance 2.0 m	0.102 pix	**0.081** pix	0.101 pix

CAMERA 2	**Small squares grid**	**Big squares grid**	**Circles grid**
Distance 1.5 m	0.082 pix	**0.081** pix	0.095 pix
Distance 1.6 m	0.081 pix	**0.078** pix	0.098 pix
Distance 1.7 m	0.079 pix	**0.077** pix	0.103 pix
Distance 1.8 m	0.085 pix	**0.078** pix	0.099 pix
Distance 1.9 m	0.089 pix	**0.079** pix	0.101 pix
Distance 2.0 m	0.096 pix	**0.081** pix	0.094 pix

CAMERA 3	**Small squares grid**	**Big squares grid**	**Circles grid**
Distance 1.5 m	0.085 pix	**0.083** pix	0.089 pix
Distance 1.6 m	**0.086** pix	0.087 pix	0.092 pix
Distance 1.7 m	0.092 pix	**0.089** pix	0.091 pix
Distance 1.8 m	0.095 pix	**0.088** pix	0.093 pix
Distance 1.9 m	0.096 pix	**0.085** pix	0.092 pix
Distance 2.0 m	0.094 pix	**0.086** pix	0.094 pix

AVERAGE	Small squares grid	Big squares grid	Circles grid
Distance 1.5 m	**0.081** pix	**0.081** pix	0.098 pix
Distance 1.6 m	0.082 pix	**0.081** pix	0.096 pix
Distance 1.7 m	0.084 pix	**0.081** pix	0.091 pix
Distance 1.8 m	0.089 pix	**0.080** pix	0.097 pix
Distance 1.9 m	0.094 pix	**0.080** pix	0.096 pix
Distance 2.0 m	0.097 pix	**0.083** pix	0.096 pix

It can be easily observed that the proposed camera calibration method achieves an high calibration accuracy, namely a low re-projection error, with each planar pattern at all tested distances, thanks to the high resolution of the captured checkerboard images on the simulator Gazebo. In the experiments with the real setup instead, image resolution and noise will be crucial factors limiting the calibration accuracy, as described in Section 6.2.
Moreover it has to be highlighted that when the distance between the camera and the checkerboard increases, the re-projection error obtained with the checkerboard with small squares slightly increases, since the corners detection get worse as the camera moves away from the checkerboard. The calibration target with big squares and the pattern with the circles grid are not too much affected by the distance issue at least up to distance $d = 2\,\text{m}$. Moreover although the calibration with the pattern with the circles grid allows to achieve very high calibration accuracy, in general the best results are achieved by the camera calibration with the checkerboard with the grid of 4×5 squares of size 4.6 cm. It has to be said that these reported results are only a part of all performed experiments with different planar patterns. However, we could infer that the calibration performances are significantly influenced by the corners detection accuracy, namely by the resolution of the captured images. So the results can variate according to the executed trajectory, since from the captured image the checkerboard can be accurately detected only if the it is properly moved in front of the cameras. However on average the results achieved by the calibration with the checkerboard with bigger squares are the most accurate. The results are accurately illustrated and confirmed by the graph reported in Figure 6.6.

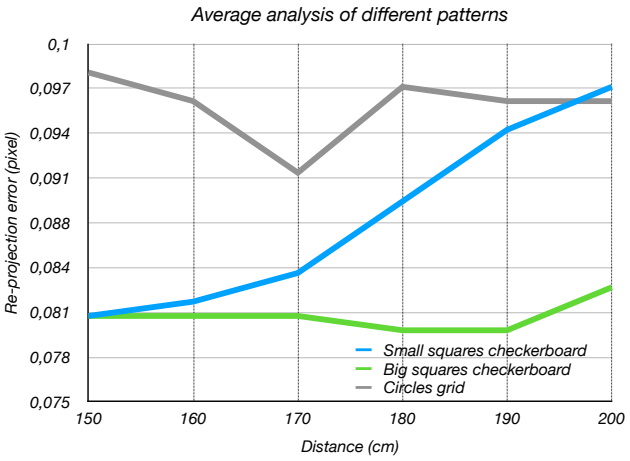


Figure 6.6: *Average results obtained with the three different planar patterns describe above.*

From this graph, it can easily derived that the best performances are achieved by calibration with the chessboard with bigger squares and in particular it can be inferred that the checkerboard with small squares most suffers by the distance issue. In the next simulated and real experiments the proposed camera calibration methods are analyzed using for simplicity the checkerboard with 5×4 squares of size 4.6 cm.

6.1.2 Analysis of the calibration method according to the distance between the camera and the robot arm

From the analysis of the calibration performances with different planar patterns, it can be inferred that the highest accuracy is achieved by the checkerboard with bigger squares shown in Figure 6.3c. Since the main aim of this thesis is determining a novel robotic workcell calibration for human-robot collaboration tasks, this subsection is focused on the analysis of the calibration performances according to the distance, in order to investigate up to which distance between robot and cameras the calibration process provides accurate results.
For this purpose it was designed a dedicated robotic workcell on the simulator Gazebo to simulate a factory environment for human robot collaboration tasks as faithfully as possible. In particular the robotic workcell is equipped by three cameras arranged around the manipulator placed in the center of the cell, as schematically described in Figure 6.7.

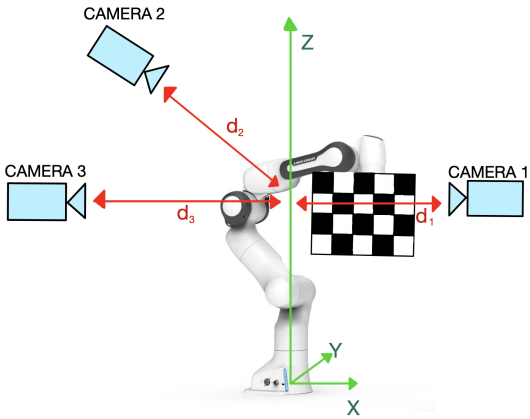
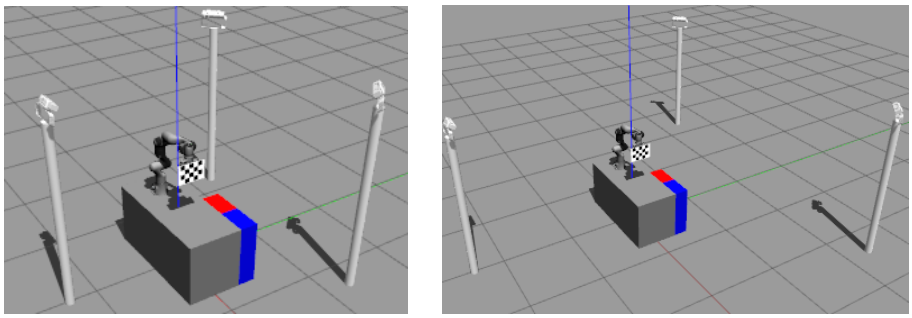


Figure 6.7: *Schematic robotic workcell arrangement for experiments related to the distance.*

The distance between the three cameras and the checkerboard is denoted by d_1, d_2 and d_3. The experiments were performed by increasing at each time the average distance $d = (d_1 + d_2 + d_3)/3$ between the cameras and z axis of the robot base reference frame. More accurately the first experiments set analyzed the calibration accuracy at an average distance $d = 150$ m whose setup is shown in Figure 6.8a and then testing the cameras calibration methods at the maximum distance $d = 3$ m as shown in Figure 6.8b.

For these experiments a dedicated predefined trajectory of 110 poses was executed more times by the robot arm, in order to move the checkerboard in front of the sensors several times and to test both the single and multi-camera hand-eye calibration methods according to the same trajectory depending on the distance. In order to evaluate the experiments



(a) *Robotic workcell with a camera network at a distance $d = 150cm$ from the robot base.*

(b) *Robotic workcell with a camera network at a distance $d = 300cm$ from the robot base.*

Figure 6.8: *Workcell arrangement of the first experiment at the left and arrangement of the last experiments at the right.*

it was adopted the re-projection error of the equation 6.1 introduced and used in previous section 6.1.1. Moreover an additional evaluation metrics was adopted, which consists in computing the euclidean difference between the transformation matrices $T_{C_i}^{C_j}$, $T_W^{C_i}$ and their real value provided by the simulator Gazebo.
Considering our particular setup where the robotic workcell is equipped with three cameras the transformation matrices analyzed are $T_{C_1}^{C_2}$, $T_{C_1}^{C_3}$, $T_{C_2}^{C_3}$, $T_W^{C_1}$, $T_W^{C_2}$ and $T_W^{C_3}$.
In the table 6.1 is reported the calibration accuracy related to the **single camera calibration** implemented one time for each sensor belonging to the camera network, depending on the increasing distance. In particular the values reported in the table 6.1 on the first six columns, denote the distance in mm obtained as norm of the translation part in the transformation matrices from their real values, moreover the errors err_i in the last three columns are the reprojection errors evaluated in pixels with the equation 6.1.

CAM 1-2-3	$\mathbf{T}_{C_1}^{C_2}$	$\mathbf{T}_{C_1}^{C_3}$	$\mathbf{T}_{C_2}^{C_3}$	$\mathbf{T}_{W}^{C_1}$	$\mathbf{T}_{W}^{C_2}$	$\mathbf{T}_{W}^{C_3}$	err_1	err_2	err_3
Distance 1.5 m	2.35	2.61	3.27	0.52	0.66	0.32	0.12	0.22	0.09
Distance 1.6 m	2.42	2.81	3.62	0.48	0.61	0.49	0.15	0.35	0.12
Distance 1.7 m	2.51	3.25	3.75	0.98	0.77	0.55	0.55	0.57	0.32
Distance 1.8 m	2.68	3.49	4.12	1.23	0.68	0.63	0.76	0.87	0.32
Distance 1.9 m	2.97	3.64	3.94	1.12	0.85	0.68	0.79	0.82	0.56
Distance 2.0 m	3.12	3.85	4.05	1.25	0.98	0.75	0.78	0.85	0.63
Distance 2.2 m	3.04	3.95	4.53	0.09	1.12	0.92	0.82	0.91	0.69
Distance 2.4 m	3.53	3.86	4.78	1.68	1.25	1.18	0.93	0.91	0.78
Distance 2.6 m	3.86	4.53	4.93	1.57	1.34	1.38	1.23	1.12	0.89
Distance 2.8 m	4.22	4.82	5.21	2.34	1.49	1.45	1.65	1.58	1.21
Distance 3.0 m	4.89	5.32	5.31	2.17	1.56	1.90	1.75	1.67	1.35

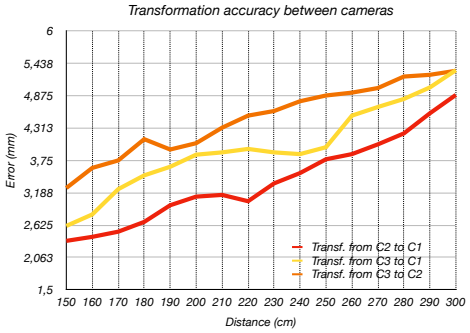
Table 6.1: *Deviation of the transformation matrices from their real values and reprojection error achieved with the single camera calibration method.*

In addition, in the table 6.2 are reported the calibration results according to the distance d obtained through the **multi-camera calibration** process. The same transformation matrices are investigated considering the translation error. The re-projection errors (pixel) achieved by the three cameras are reported as well in the last three columns.

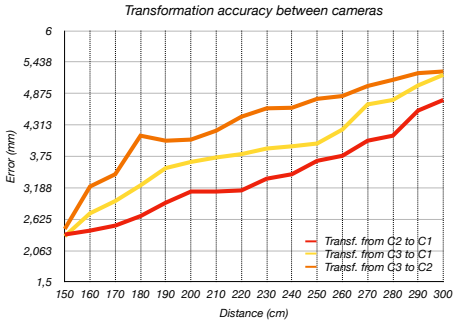
CAM 1-2-3	$\mathbf{T}_{C_1}^{C_2}$	$\mathbf{T}_{C_1}^{C_3}$	$\mathbf{T}_{C_2}^{C_3}$	$\mathbf{T}_{W}^{C_1}$	$\mathbf{T}_{W}^{C_2}$	$\mathbf{T}_{W}^{C_3}$	err_1	err_2	err_3
Distance 1.5 m	2.35	2.32	2.45	0.52	0.69	0.38	0.12	0.24	0.09
Distance 1.6 m	2.42	2.73	3.21	0.49	0.61	0.43	0.15	0.32	0.12
Distance 1.7 m	2.51	2.95	3.43	0.78	0.79	0.63	0.46	0.39	0.24
Distance 1.8 m	2.68	3.23	4.12	1.23	0.73	0.67	0.76	0.65	0.32
Distance 1.9 m	2.92	3.54	4.03	1.25	0.82	0.68	0.79	0.77	0.56
Distance 2.0 m	3.12	3.65	4.05	1.28	0.98	0.75	0.81	0.85	0.63
Distance 2.2 m	3.14	3.79	4.46	1.32	1.07	0.91	0.85	0.88	0.67
Distance 2.4 m	3.43	3.93	4.62	1.56	1.15	1.12	0.95	0.91	0.75
Distance 2.6 m	3.76	4.23	4.83	1.57	1.31	1.28	1.21	1.10	0.93
Distance 2.8 m	4.12	4.76	5.12	2.16	1.43	1.38	1.65	1.42	1.19
Distance 3.0 m	4.76	5.21	5.27	2.19	1.56	1.85	1.79	1.73	1.28

Table 6.2: *Deviation of the transformation matrices from their real values and reprojection error achieved with the multi camera calibration method.*

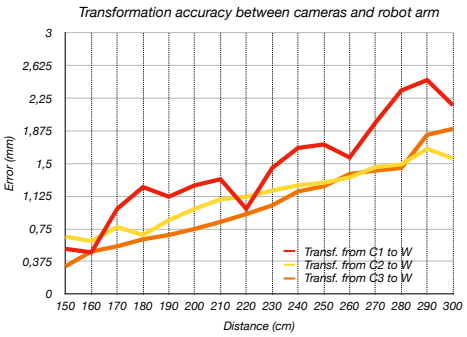
It can be observed how the accuracy of the estimated transformation matrices with respect to their real values significantly decreases as the distance d increases. In particular the average deviation of all the transformation matrix $T_W^{C_i}$ are equal about to 0.5 at the distance $d = 1.5\,\text{m}$ and it becomes more than triple > 1.78 at distance $d = 3\,\text{m}$ for both camera calibration methods. Moreover the re-projection error even increases by an order of magnitude on the last experiments with respect to the first one. These achieved results highlight how the distance significantly affects the calibration performances, the resolution of the checkerboard in the images decreases as the distance enlarges and then causing the consequent reduction of the corner detection accuracy.
In the graphs in Figure 6.9 are reported the results listed in the Tables 6.1 and 6.2, in particular it is clearly illustrated the heavy dependence to the distance of the calibration performances.
Indeed, it can be observed that the graphs 6.9a, 6.9b, 6.9c and 6.9d depict an increasing trend of the transformations deviation with respect to the distance, underlining how much the robotic workcell sizes greatly influences the accuracy of the calibration. However, the obtained results, both at shorter distance such $d = 1.5\,\text{m}$ and at farther distance as $d = 3\,\text{m}$, are very good thanks to the high resolution of the images captured on the simulator. As it can be seen in the next section 6.2, this accuracy will get worse in the real experiments, but these simulated experiments of this section faithfully describes the influence of the distance on the calibration performance.



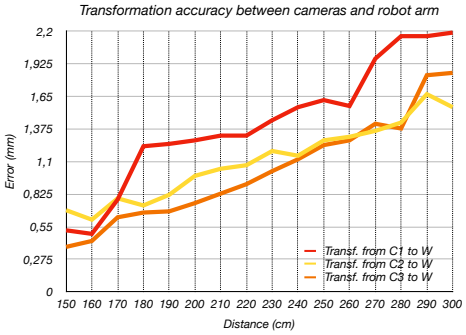
(a) *Deviation of transformations between cameras computed by the single camera calibration method.*



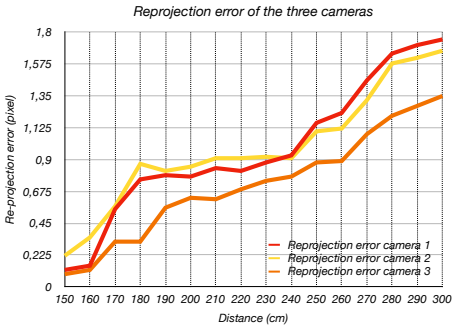
(b) *Deviation of transformations between cameras computed by the multi camera calibration method.*



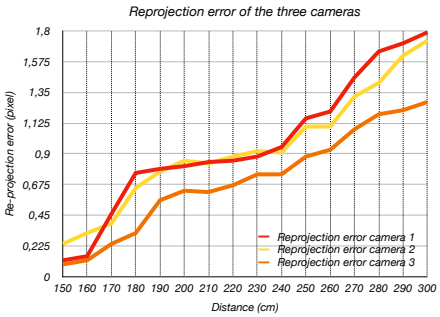
(c) *Deviation of transformations between cameras and robot arm computed by the single camera calibration method.*



(d) *Deviation of transformations between cameras and robot arm computed by the multi camera calibration method.*



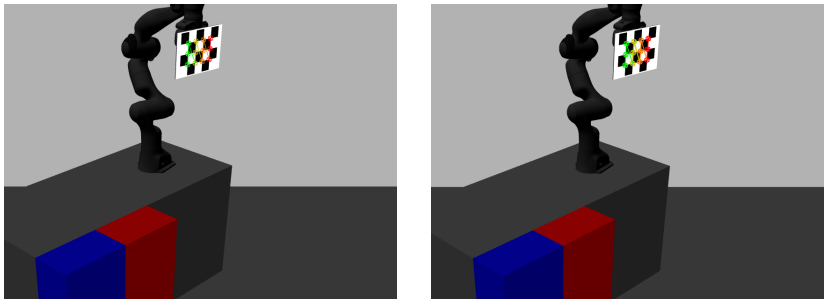
(e) *Reprojection error achieved with single-camera calibration.*



(f) *Reprojection error achieved with multi-camera calibration.*

Figure 6.9: *Deviations of estimated transformation matrices from their real values, achieved achieved by the single and multi camera calibration on the first four graphs. Reprojection error related to the single and multi camera calibration depending on the distance d between the camera and the robot arm on the last two graphs.*

The consideration made above are further confirmed by the graphs 6.9e and 6.9f, which illustrates an increasing trend of the reprojection error according to the increasing robotic workcell sizes. Although the reprojection error increases as the distance is higher, the calibration accuracy achieved with both camera calibration methods is really high as it can be seen in the two Figures 6.10a and 6.10b.



(a) *Reprojected corners through the transformation matrices estimated by the single camera calibration.*

(b) *Reprojected corners through the transformation matrices estimated by the multi camera calibration.*

Figure 6.10: *Reprojected corners on the same image adopting the transformation matrices determined through the single and multi camera calibration at distance $d = 3\,m$.*

In Figure 6.10 can be seen how the calibration at distance $d = 3\,\text{m}$ achieves really accurate calibration results. Indeed in Figures 6.10a and 6.10b are shown the 3D corners of the board, reprojected on the image plane of the camera through the transformation matrices estimated by the calibration process.
Moreover from the results reported in previous tables 6.1 and 6.2, it can be highlighted the slight improvement achieved by the multi-camera calibration which provides the optimization of the transformation between cameras.

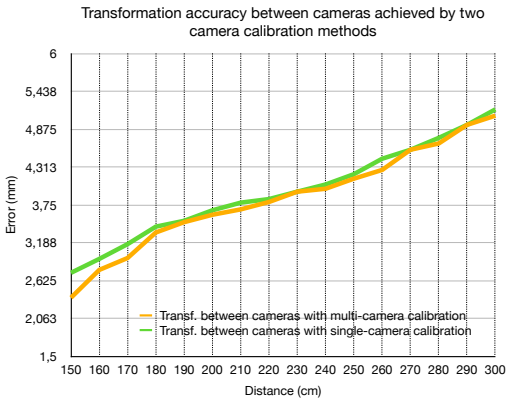


Figure 6.11: *Transformation accuracy achieved by the single and multi calibration methods, computing the deviation mean between the estimated transformation matrix $T_{C_i}^{C_j}$ and their real values at each distance d.*

In the graph of Figure 6.11 it is reported the average deviation computed among the three transformation matrices $T_{C_i}^{C_j}$ at each distance d and it can be observed a slightly lower error obtained with the multi camera calibration (yellow line) according to the distances with respect to the error achieved by the single camera calibration (green line).

6.2 Experiments in the Robot Vision Lab testbed

In order to perform real experiments and then to test the proposed calibration method on a real environment it was designed a dedicated robotic workcell with the Panda robot arm and a camera network as shown in Figure 6.12. In particular in this section are analyzed the calibration performances achieved with different kind of sensors, namely Kinect V2, depth camera D455 and LiDAR camera L515 as listed in section 5. Moreover these sensors are adopted to test the proposed calibration method with the classical checkerboard analyzed in section 6.1 with 5×4 squares of sizes 4.6 cm. The results reported in the following subsection take into account both the single and multi camera calibration. More accurately the subsection 6.2.1 is focused on the analysis of the single and multi-camera calibration with different cameras; in the subsection 6.2.2 it is accurately analyzed the single camera calibration with respect to the other calibration methods most adopted in the state-of-art and finally in the subsection 6.2.3 the multi camera calibration method is compared with some other methods found in literature.

Figure 6.12: *Robotic workcell designed in Robot Vision Lab; in particular in this image there are three depth sensors D455 with a robot arm in the center of the workcell with the chessboard attached on the end-effector.*

It has to be highlighted that the evaluation metrics adopted in the real experiments and reported in the following subsections for every experiments is the re-projection error (pixel), since the deviation of the estimated transformation matrices from their real values (ground truth) can not be computed because they are no longer provided as in the simulator Gazebo.

6.2.1 Analysis of the single and multi camera calibration with different sensors

As previously introduced, this section is focused on the analysis of the single and multi camera calibration with different sensors. In particular the Microsoft Kinect V2, the Intel RealSense depth camera D455 and the Intel RealSense LiDAR camera L515 were adopted. The robotic workcell designed for these experiments is equipped by a camera network of three sensors, due to the availability of hardware in laboratory and they are arranged on the same side of the robot arm at an increasing distance d from the robot arm at each experiment. In particular the cameras are fixed at an height $h = 1.6\,\text{m}$ from the ground and they are are placed at a distance $d_1 = 0.8\,\text{m}$, $d_2 = 1.2\,\text{m}$, $d_3 = 1.8\,\text{m}$, $d_4 = 2.8\,\text{m}$ for each experiment, in order to analyze the cameras properties with respect to the robotic workcell sizes. Hence it was tested the single and the multi camera calibration with the checkerboard with 5×4 squares of sizes $4.6\,\text{cm}$. The same predefined trajectory of 20 poses is repeatedly executed by the panda arm, in order to properly move the checkerboard in front of the sensor and to analyze the different cameras with the same robot poses and the consequent same checkerboard images.
In the following table are reported the re-projection error (pixel) obtained by the three cameras C_1, C_2, C_3, with the three different categories of cameras according to the distance d_i with $i = 1, 2, 3, 4$ through the single camera calibration and in the four next tables the corresponding results achieved with the multi camera calibration method. Note that some values are missing, due to the checkerboard not detected in the images at a given distance.

Single calibration $d_1 = 0.8$	**Reproj. error** $\mathbf{C}_1$	**Reproj. error** $\mathbf{C}_2$	**Reproj. error** $\mathbf{C}_3$
Kinect V2	0.21	0.34	0.27
Depth camera D455	0.22	0.27	0.27
LiDAR camera L515	0.27	0.32	0.33

Single calibration $d_2 = 1.2$	**Reproj. error** $\mathbf{C}_1$	**Reproj. error** $\mathbf{C}_2$	**Reproj. error** $\mathbf{C}_3$
Kinect V2	0.26	0.35	0.36
Depth camera D455	0.22	0.18	0.25
LiDAR camera L515	0.26	0.39	0.32

Single calibration $d_3 = 1.8$	**Reproj. error** $\mathbf{C}_1$	**Reproj. error** $\mathbf{C}_2$	**Reproj. error** $\mathbf{C}_3$
Kinect V2	0.38	0.45	0.41
Depth camera D455	0.86	1.22	0.98
LiDAR camera L515	0.31	0.36	0.40

Single calibration $d_4 = 2.8$	**Reproj. error** $\mathbf{C}_1$	**Reproj. error** $\mathbf{C}_2$	**Reproj. error** $\mathbf{C}_3$
Kinect V2	1.35	1.28	1.25
Depth camera D455	-	-	-
LiDAR camera L515	0.42	0.48	0.41

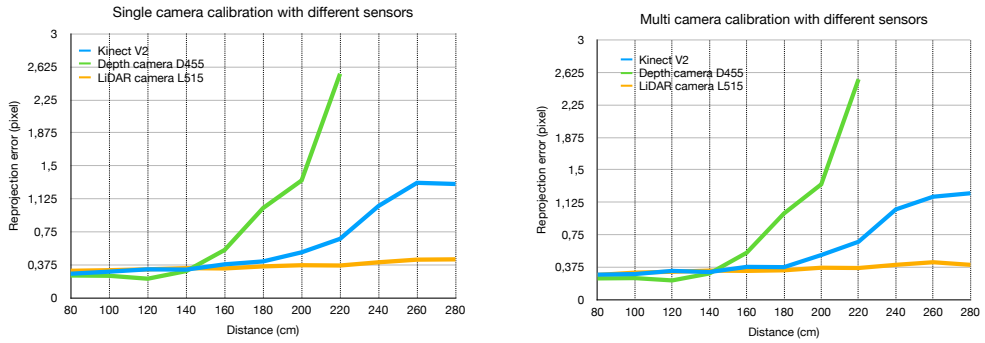
Multi calibration $d_1 = 0.8$	**Reproj. error C_1**	**Reproj. error C_2**	**Reproj. error C_3**
Kinect V2	0.23	0.35	0.29
Depth camera D455	0.21	0.27	0.26
LiDAR camera L515	0.26	0.28	0.32

Multi calibration $d_2 = 1.2$	**Reproj. error C_1**	**Reproj. error C_2**	**Reproj. error C_3**
Kinect V2	0.28	0.36	0.36
Depth camera D455	0.21	0.19	0.27
LiDAR camera L515	0.26	0.39	0.32

Multi calibration $d_3 = 1.8$	**Reproj. error C_1**	**Reproj. error C_2**	**Reproj. error C_3**
Kinect V2	0.36	0.43	0.34
Depth camera D455	0.81	1.15	1.03
LiDAR camera L515	0.35	0.38	0.29

Multi calibration $d_4 = 2.8$	**Reproj. error C_1**	**Reproj. error C_2**	**Reproj. error C_3**
Kinect V2	1.27	1.15	1.27
Depth camera D455	-	-	-
LiDAR camera L515	0.39	0.42	0.40

From data of the previous tables can be derived a more accurate analysis which highlight the difference among the three sensors as reported in graphs of the Figure 6.13. In particular the mean re-projection error of the three cameras obtained by the single and multi camera calibration, at each experiment is taken into account and it is analyzed according to the distance d and the adopted sensor.



(a) *Mean re-projection error obtained with the single camera calibration according to the distance d.*

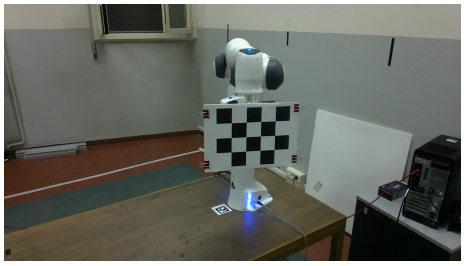
(b) *Mean re-projection error obtained with the multi camera calibration according to the distance d.*

Figure 6.13: *Analysis of the calibration accuracy achieved by each sensor according to the distance through the single and the multi camera calibration method.*

It is interesting to explore the different behaviour of the three sensors according the the distance between the camera and the checkerboard to be detected. It can be immediately seen that the depth sensor D455 achieves the highest calibration accuracy in the shortest

distance until $d \simeq 1.5\,\text{m}$. However as the distance increases, the corresponding re-projection error significantly enlarges, reaching the highest value of $err \simeq 2.5$ pixel both in the single and the multi camera calibration at a distance $d \simeq 2.0 - 2.2\,\text{m}$, and then for higher distance the stereo camera is not able to perform the chessboard detection and consequently failing the cameras calibration (as denoted by the "-" symbols in the tables). This issue is mainly caused by the low RGB resolution of the D455 sensor (1280×800) as reported in section related to the hardware specifications 5.1.
The Kinect V2 and LiDAR camera L515 can achieve excellent calibration results all over the tested distances, thanks to their high RGB resolution equal to (1920×1080). In particular it can be highlighted an higher accuracy achieved by the LiDAR camera at longer distances thanks to its narrower RGB field of view ($70° \times 43°$) than the one of the Kinect V2 ($84.1° \times 53.8°$). This can be seen more easily at shortest distance as illustrated in Figure 6.14.

(a) *Checkerboard image captured by the Kinect V2 at a distance* $d = 1.2\,m$.



(b) *Checkerboard image captured by the LiDAR camera L515 at a distance* $d = 1.2\,m$.

Figure 6.14: *Checkerboard images captured by the Kinect and LiDAR camera from the same distance* $d = 1.2\,m$.

In particular the two Figures 6.14a, 6.14b illustrate the same checkerboard image captured from the same distance, but it can be easily observed that the checkerboard on the LiDAR image covers an higher area than the checkerboard detected by the Kinect thanks to its narrow FoV, then allowing a better corners detection accuracy.

6.2.2 Analysis of the single-camera calibration with other state-of-art methods

This section is focused on the analysis of the proposed single camera calibration method, with an accurate comparison with other single camera hand-eye calibration most adopted in the state-of-art. In particular our proposed single camera calibration method was compared with the following state-of-art methods already introduced in Section 2.2:

- **Tsai method** which introduced an autonomous technique for 3D robotics hand/eye calibration [52];
- **Danillidis method**, also known as dual quaternion method [53];
- **Horaud method** which includes a new linear formulation of classical hand-eye calibration which is linear and numerically efficient [54];

- **Park method** which introduces a cyclic coordinate descent algorithm for optimizing quadratic objective function on $SE(3)$ and applies it to a class of robot sensor calibration problems [55].

In order to perform these experiments the robotic workcell was equipped by a single camera directed to the robot arm, in order to acquire images of the checkerboard with 5×4 squares of size 4.6 cm from different perspective. The predefined trajectory for these experiments consists in a set of 20 poses where the checkerboard is not moved all over the image plane but it is kept almost stationary and it is only tilted at each pose with respect to the image plane of the camera, to capture checkerboard images on different orientations.

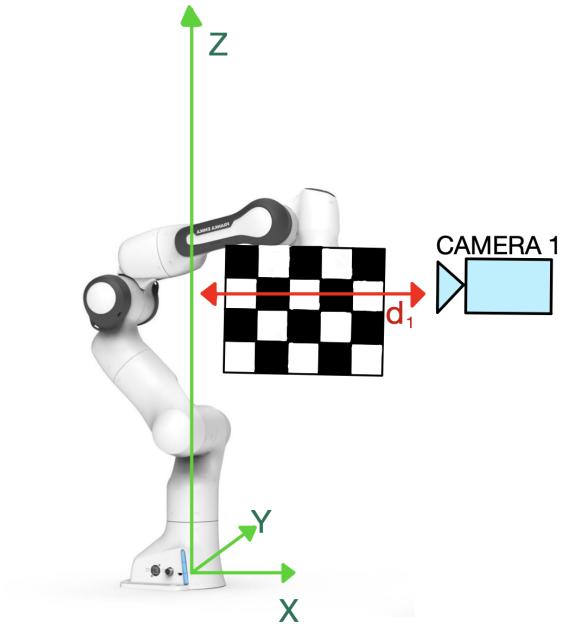


Figure 6.15: *Robotic workcell setup adopted to perform the comparison among different methods of single camera hand-eye calibration method.*

In the following table are reported the average reprojection errors achieved on all the images by the methods previously described and by our proposed method. In particular the reported results are evaluated in pixel terms on the same set of images acquired only one single time for all the optimization processes and they are obtained adopting a robotic workcell with a single camera (Kinect V2) placed from the robot arm at a distance $d = 1.5$ m as schematically shown in Figure 6.15.

SINGLE CALIBRATION	**Reprojection error (pixel)**
Tsai method	4.51 pixel
Danillidis method	7.68 pixel
Horaud method	5.42 pixel
Park method	5.13 pixel
Our proposed method	0.79 pixel

It can be easily observed that among these single camera hand eye calibration methods our proposed method guarantees higher calibration accuracy according to the reprojection error (pixel).

6.2.3 Analysis of the multi-camera calibration with other state-of-art methods

This section takes into account the experiments performed to test the multi camera hand-eye calibration process, by comparing our proposed method with two methods adopted in literature: OpenPTrack [25] and a multi-camera calibration method proposed in [32]. As it is introduced in section 1 the main starting point for this thesis was the multi camera calibration method previously adopted in our research laboratory proposed in the OpenPTrack framework [25], based on a manual calibration performed by a person with a large calibation pattern. For this purpose, a workcell without any panda robot arm and three cameras is adopted for the multi-camera calibration procedure proposed by OpenPTrack, which is implemented by manually moving a big checkerboard 80 cm×90 cm in the workcell without the robot, in order to further facilitate the checkerboard motion as shown in Figure 6.16a. Then it was introduce the robot arm, to accomplish our proposed calibration method with the same cameras network arrangement.

(a) *Manual images acquisition phase for OpenPTrack calibration.*

(b) *Automatic images acquisition phase for our proposed calibration method.*

Figure 6.16: *Two adopted setups for multi-camera calibration where the camera networks are arranged in the same setup and sizes.*

Hence, with the same arrangement of the camera network it was used the robot arm to move the smaller A4 checkerboard as expected by our proposed method and then the multi camera calibration method was performed (see Figure 6.16b). In particular the experiments were performed, considering three LiDARs L515 placed at a distance $d = 2.0\,\text{m}$ from the robot and with an height of $h = 2.1\,\text{m}$ from the ground.
Since multi camera calibration proposed by OpenPTrack provides an optimization of the transformation matrices between cameras, we have evaluated this method with respect to the our proposed technique, considering the re-projection error obtained by projecting the checkerboard 3D points passing through the transformation matrix which describes the relative pose between the two cameras that can simultaneously detect the checkerboard.

This re-projection error is accurately computed as follow:

$$err = \sqrt{\frac{\sum_{k=0}^{M-1} \left(K_j[I|\mathbf{0}] T_{C_i}^{C_j} T_B^{C_i} (P_j^B)_k - (p_j)_k \right)^2}{M}} \quad \forall (i,j) \tag{6.2}$$

where M are the poses in which the couple of cameras (C_i, C_j) can detect the checkerboard, $K_j[I|\mathbf{0}] T_{C_i}^{C_j} T_B^{C_i} (P_j^B)_k$ is the reprojection of the checkerboard 3D corner onto the image plane of camera C_j through the optimized matrix $T_{C_i}^{C_j}$ and $(p_j)_k$ are the 2D detected corners by the camera C_j at the robot's pose k. The achieved results in terms of mean re-projection error (pixel) are reported in the following table, considering the re-projection through each camera C_i with $i = 1, 2, 3$.

MULTI CALIBRATION	**Err CAM1**	**Err CAM2**	**Err CAM3**
OpenPTrack method	6.51 pixel	5.32 pixel	7.24 pixel
Our proposed multi calibration	0.79 pixel	0.81 pixel	0.94 pixel
Our proposed single calibration	1.15 pixel	1.32 pixel	1.21 pixel

It can be observed that although the OpenPTrack calibration exploits a bigger checkerboard, our calibration methods achieve an higher accuracy thanks to the role played by the robot. In particular the checkerboard motion which is handled by the robot arm allows the cameras to capture images of perfectly still planar target, in order to accomplish a more accurate calibration. Instead the OpenPTrack calibration process significantly suffer from the imprecise manual checkerboard motions, indeed the human who moves the checkerboard during the image acquisition phase since he is not capable of standing perfectly still, especially in more awkward positions. Moreover it can be seen that re-projection error achieved by the multi-camera calibration is more accurate than the one obtained by the single camera calibration. This can be expected, since the multi camera calibration provides an optimization of the transformation matrix between the cameras, as accurately explained in section 4.2. A further advantage of our proposed method, in addition to the accuracy, is the time spent to perform the whole calibration which is very fast compared to the manual prodecure used in OpenPTrack, thanks to the marginalisation of the slow and imprecise human intervention on the calibration process.

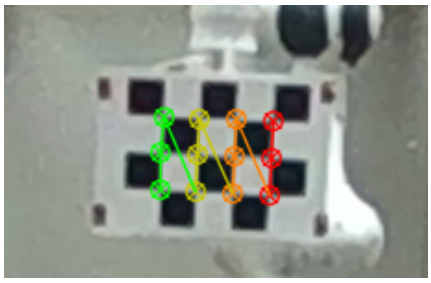
A further comparison of our proposed method was performed with respect to the hand-eye calibration method proposed in [32]. In particular this article proposed a sensors network calibration with a single hand-eye calibration method to be performed one time for each camera belonging to the cameras network. Then it was designed a robotic workcell with three Kinect V2 placed on the same side of the robot arm and directed on the manipulator at a distance $d = 1.1\,\text{m}$. Thus it was performed our proposed method with a predefined trajectory of 20 poses and the mean re-projection error of equation (6.1) achieved by each camera is reported in the following table with respect to the results obtained by the method proposed by the paper.

CALIBRATION at d $= 1.1\,\text{m}$	**Mean reproj. err of 3 cameras**
Method of [32]	0.75 pixel
Our proposed multi calibration	**0.27** pixel
Our proposed single calibration	0.34 pixel

6.3 Image super resolution for large robotic workcells

As it was accurately described in section 6.1.2, the robotic workcell size is a main aspect which significantly affects the calibration accuracy. In particular we have observed that as the distance increases, the accuracy of the corners detection and the resulting calibration precision significantly decreases. Moreover, by excessively increasing the distance between the camera and the checkerboard to be detected, the probability that the checkerboard detection fails increases and then even the resulting calibration.
The proposed trick is based on **bicubic interpolation** to enlarge the images acquired by the cameras [94], thus increasing the number of pixels for the checkerboard in the image: this allow to improve checkerboard detection robustness at further distances (i.e., reduce checkerboard corners wrongly or not detected). Image resizing is a crucial concept which consists in augmenting or reducing the number of pixels in a picture. A common method to obtain this result is given by the interpolation process which works by using known data to estimate values at unknown points. It means that if it is necessary to determine the pixel intensity of a picture at a selected location within the grid at coords (x, y), but only $(x-1, y-1)$ and $(x+1, y+1)$ are known, it has to be estimated the value at (x, y) using linear interpolation. The greater the quantity of already known values, the higher would be the accuracy of the estimated pixel value. There are different interpolation algorithms used to obtain this phenomenon such as the nearest neighbor [95], bilinear [96], bicubic [94], and others. They exploit adjacent pixels during the interpolation and the accuracy of those algorithms is increased by enlarging the set of neighboring pixels considered in order to estimate the new pixel value. Bicubic interpolation considers the 16 closest pixels (4×4), where the pixels that are closer to the one that has to be estimated are characterized by higher weights and the farther pixels have instead lower weights. Moreover the images resampled with bicubic interpolation are smoother and have fewer interpolation artifacts which is a common issue of other interpolation methods. In our specific case this method allows to achieve significant improvement on the corner detection especially with the depth stereo camera which was not able to detect the checkerboard corners at distance larger than $d = 2.4\,\text{m}$ as proved in the previous section and illustrated in Figure 6.17a.

(a) *Original image* (1280 × 720) *where the camera can not detect checkerboard corners.*



(b) *Image with double sizes* (2560 × 1440) *where the corners can be detected.*

It can be easily observed that by resizing the original image, the corner detection can be accurately performed overcoming the distance issue as shown in Figure 6.17b, especially for cameras such as the depth sensor D455 which has a lower RGB resolution. Moreover a

set of experiment was performed in order to analyze how the image resizing improves the calibration accuracy when the distance between the camera and the robot arm increases.

	ORIGINAL IMAGES			**RESIZED IMAGES**		
CAM 1-2-3	Err_1	Err_2	Err_3	Err_1	Err_2	Err_3
Distance 1.5 m	0.17	0.22	0.20	0.16	0.18	0.22
Distance 1.6 m	0.22	0.29	0.29	0.22	0.28	0.21
Distance 1.7 m	0.78	0.71	0.89	0.32	0.43	0.29
Distance 1.8 m	0.86	1.22	0.98	0.29	0.45	0.33
Distance 1.9 m	0.99	1.35	1.42	0.45	0.39	0.38
Distance 2.0 m	1.26	1.55	1.78	0.36	0.42	0.48
Distance 2.2 m	2.35	3.56	2.98	0.55	0.62	0.73
Distance 2.4 m	-	-	-	0.62	0.71	0.69
Distance 2.6 m	-	-	-	0.68	0.75	0.82
Distance 2.8 m	-	-	-	0.65	0.81	0.79
Distance 3.0 m	-	-	-	0.83	0.93	0.88

In the previous table are reported the reprojection error obtained by our single and multi camera calibration methods adopting a robotic workcell equipped by a camera network composed by three stereo cameras placed on the same side of the manipulator at an increasing distance d. It can be observed that, at shorter distances, the calibration accuracy in terms of re-projection error is not too much helped by image resize trick, since the detection on the original images in the distances range $1.5\,\text{m}-1.8\,\text{m}$ is accurate as well. However image resize becomes really helpful when the distance increases such as $d = 2.0\,\text{m}-2.2\,\text{m}$, by significantly reducing the re-projection error. This property can be especially highlighted at larger distance $d > 2.4\,\text{m}$ where the corners detection on the original images completely fails and instead by resizing the image really accurate results can be obtained. In the graph illustrated in Figure 6.17 are reported the mean re-projection error among the three cameras C_1, C_2 and C_3 according to the distance d, obtained by keeping the images with their original size and then by resizing them. It can be easily seen the improvement provided by the image resize especially when the distance enlarges.

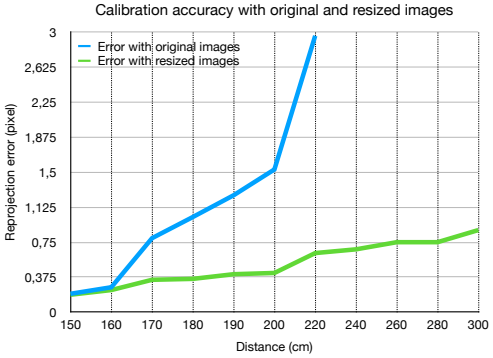


Figure 6.17: *Calibration accuracy in terms of re-projection error obtained by using orginal images and resized images, according to the distance d.*

Chapter 7

Conclusions

In this thesis, we address the problem of camera calibration considering the calibration of the cameras in a robotic workcell for human-robot collaboration. In particular, a novel approach of automatic multi-camera hand-eye calibration for robotic workcells was presented, with the main purpose of overcoming the issues encountered with manual procedures commonly used in literature, as the one proposed in the OpenPTrack framework.
An automatic approach was proposed which allows to remove the human intervention during the checkerboard images acquisition phase, that, as illustrated in section 6 was the main aspect which negatively affected the previous calibration method performances. Moreover, moving the checkerboard with the robot manipulator allows to highly speed up the whole calibration process and then significantly reducing the downtime for a production workflow in a real factory where a robotic workcell calibration is required. As it was described in the previous sections, our proposed calibration process achieved higher accuracy with respect to the other camera calibration methods found and adopted in the state-of-art, allowing to precisely determine the extrinsic parameters among the cameras and the robot arm and then ensuring an high human safety during human-robot collaborations. A further significant contribution of our approach with respect to the state-of-art is the capability of performing the calibration of a camera network in a robotic workcell with large sizes, even with cameras placed at a distance $d = 3\,\text{m}$ from the robot arm, achieving a reprojection error even lower than 1 pixel. This improvement allows an accurate calibration of network of cameras in large robotic workcell, hence our approach is more suitable to be used in real scenario such as human-robot collaboration tasks.
The experiments run with the system outlined that through a classical checkerboard with squares not too small an higher calibration accuracy than other planar targets is achieved. Our method even proved to be simple and easily generalizable. Simplicity is given by its modular process composed by a robot motion phase with the corresponding image acquisition phase and then the final calibration phase; flexibility is demonstrated by quantitative experiments with different sensors and given by the opportunity to adopt even different robot arms. The main critical aspect encountered during the thesis experiments is the need to have a predefined trajectory to be executed by the robot arm in order to correctly move the checkerboard in front of each sensor belonging to the camera network. This is due to the fact that as long as the robotic workcell is not calibrated, the robot is not able to prop-

erly move the checkerboard in front of the cameras in order to execute the images capture and meanwhile to ensure the collision avoidance between robot and other objects in the workspace. Furthermore the predefined trajectory is not perfectly proper for each robotic workcell but a customized trajectory has to be stored for each specific camera network arrangement.
For this reason future developments will be focused on the exploitation of a deep neural network to correctly visual guide a robot arm in front of the camera before the calibration accomplishment, and then to automatically execute a proper trajectory required for the image acquisition phase, with the main purpose of removing completely the human aid.
Overall, the goals set for this thesis were satisfactorily achieved, obtaining higher calibration accuracy with respect to other adopted methods in literature and allowing precise robotic workcell calibrations and then simultaneously ensuring significant improvement for other research works related to human-robot collaboration.

Bibliography

[3] Jens F Buhl et al. "A dual-arm collaborative robot system for the smart factories of the future". In: *Procedia manufacturing* 38 (2019), pp. 333–340.

[5] Paul Baxter et al. "Safe human-robot interaction in agriculture". In: *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction* (2018), pp. 59–60.

[6] Mithun Jacob et al. "Gestonurse: a robotic surgical nurse for handling surgical instruments in the operating room". In: *Journal of Robotic Surgery* 6.1 (2012), pp. 53–63.

[7] Mithun George Jacob, Yu-Ting Li, and Juan P Wachs. "Gestonurse: a multimodal robotic scrub nurse". In: *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction* (2012), pp. 153–154.

[8] Ali Ahmad Malik and Arne Bilberg. "Complexity-based task allocation in human-robot collaborative assembly". In: *Industrial Robot: the international journal of robotics research and application* (2019).

[9] Fen Fang et al. "Self-teaching strategy for learning to recognize novel objects in collaborative robots". In: *Proceedings of the 2019 5th International Conference on Robotics and Artificial Intelligence* (2019), pp. 18–23.

[10] Abdulla Mohamed et al. "Automating active stereo vision calibration process with cobots". In: *IFAC-PapersOnLine* 50.2 (2017), pp. 163–168.

[11] Laura Dipietro, Angelo M Sabatini, and Paolo Dario. "A survey of glove-based systems and their applications". In: *Ieee transactions on systems, man, and cybernetics, part c (applications and reviews)* 38.4 (2008), pp. 461–482.

[12] Marco E Benalcázar et al. "Hand gesture recognition using machine learning and the Myo armband". In: *2017 25th European Signal Processing Conference (EUSIPCO)* (2017), pp. 1040–1044.

[15] Zhiqiang Zhang, Lin Zhang, and Guang-Zhong Yang. "A computationally efficient method for hand–eye calibration". In: *International journal of computer assisted radiology and surgery* 12.10 (2017), pp. 1775–1787.

[16] Florian Beuss et al. "Cobots in maxillofacial surgery–challenges for workplace design and the human-machine-interface". In: *Procedia CIRP* 100 (2021), pp. 488–493.

[19] Olatz De Miguel Lázaro et al. “An approach for adapting a cobot workstation to human operator within a deep learning camera”. In: *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)* 1 (2019), pp. 789–794.

[22] L Li. “Research on camera calibration using new optimization strategy”. In: *Opto-Electronics Review* 19.4 (2011), pp. 454–461.

[24] Renbo Xia et al. “Global calibration of non-overlapping cameras: State of the art”. In: *Optik* 158 (2018), pp. 951–961.

[25] Matteo Munaro, Filippo Basso, and Emanuele Menegatti. “OpenPTrack: Open source multi-camera calibration and people tracking for RGB-D camera networks”. In: *Robotics and Autonomous Systems* 75 (2016), pp. 525–538.

[29] Klaus H Strobl and Gerd Hirzinger. “Optimal hand-eye calibration”. In: *2006 IEEE/RSJ international conference on intelligent robots and systems* (2006), pp. 4647–4653.

[30] Zijian Zhao. “Hand-eye calibration using convex optimization”. In: *2011 IEEE International Conference on Robotics and Automation* (2011), pp. 2947–2952.

[31] Kwang-Hee Lee et al. “High precision hand-eye self-calibration for industrial robots”. In: *2018 International Conference on Electronics, Information, and Communication (ICEIC)* (2018), pp. 1–2.

[32] Justinas Miseikis et al. “Automatic calibration of a robot manipulator and multi 3d camera system”. In: *2016 IEEE/SICE International Symposium on System Integration (SII)* (2016), pp. 735–741.

[37] V Douskos et al. “Fully automatic camera calibration using regular planar patterns”. In: *Int. Arch. Photogram. Remote Sens. Spatial Inf. Sci* 37 (2008), pp. 21–26.

[38] Tianlong Yang et al. “Sub-pixel chessboard corner localization for camera calibration and pose estimation”. In: *Applied Sciences* 8.11 (2018), p. 2118.

[39] Eung-Su Kim and Soon-Yong Park. “Extrinsic calibration of a camera-LIDAR multi sensor system using a planar chessboard”. In: *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)* (2019), pp. 89–91.

[40] Zhengyou Zhang. “A flexible new technique for camera calibration”. In: *IEEE Transactions on pattern analysis and machine intelligence* 22.11 (2000), pp. 1330–1334.

[41] Bernard Schmidt and Lihui Wang. “Automatic work objects calibration via a global–local camera system”. In: *Robotics and computer-integrated manufacturing* 30.6 (2014), pp. 678–683.

[42] Dengqing Tang et al. “AprilTag array-aided extrinsic calibration of camera–laser multi-sensor system”. In: *Robotics and biomimetics* 3.1 (2016), pp. 1–9.

[44] Prashant Ganesh et al. “Extrinsic calibration of camera and motion capture systems”. In: *SoutheastCon 2021* (2021), pp. 01–08.

[45] Ole Kroeger, Johannes Huegle, and Carsten A Niebuhr. “An automatic calibration approach for a multi-camera-robot system”. In: *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (2019), pp. 1515–1518.

[46] Sergio Garrido-Jurado et al. “Automatic generation and detection of highly reliable fiducial markers under occlusion”. In: *Pattern Recognition* 47.6 (2014), pp. 2280–2292.

[47] Yoonsu Park et al. “Calibration between color camera and 3D LIDAR instruments with a polygonal planar board”. In: *Sensors* 14.3 (2014), pp. 5333–5353.

[48] Ankur Datta, Jun-Sik Kim, and Takeo Kanade. “Accurate camera calibration using iterative refinement of control points”. In: *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops* (2009), pp. 1201–1208.

[49] Bin Chen and Bing Pan. “Camera calibration using synthetic random speckle pattern and digital image correlation”. In: *Optics and Lasers in Engineering* 126 (2020), p. 105919.

[50] Po-Chang Su et al. “A fast and robust extrinsic calibration for RGB-D camera networks”. In: *Sensors* 18.1 (2018), p. 235.

[51] Stuart Duncan, Holger Regenbrecht, and Tobias Langlotz. “A fast multi-RGBD-camera calibration”. In: *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (2019), pp. 906–907.

[52] Roger Y Tsai and Reimar K Lenz. “Real time versatile robotics hand/eye calibration using 3D machine vision”. In: *Proceedings. 1988 IEEE International Conference on Robotics and Automation* (1988), pp. 554–561.

[53] Konstantinos Daniilidis. “Hand-eye calibration using dual quaternions”. In: *The International Journal of Robotics Research* 18.3 (1999), pp. 286–298.

[54] Nicolas Andreff, Radu Horaud, and Bernard Espiau. “On-line hand-eye calibration”. In: *Second International Conference on 3-D Digital Imaging and Modeling (Cat. No. PR00062)* (1999), pp. 430–436.

[55] Seungwoong Gwak, Junggon Kim, and Frank Chongwoo Park. “Numerical optimization on the Euclidean group with applications to camera calibration”. In: *IEEE Transactions on Robotics and Automation* 19.1 (2003), pp. 65–74.

[56] Alexander Kirillov Jr. *An introduction to Lie groups and Lie algebras*. 113. Cambridge University Press, 2008.

[57] Gerald Schweighofer and Axel Pinz. “Globally Optimal O (n) Solution to the PnP Problem for General Camera Models.” In: *BMVC* (2008), pp. 1–10.

[58] Shiqi Li, Chi Xu, and Ming Xie. “A robust O (n) solution to the perspective-n-point problem”. In: *IEEE transactions on pattern analysis and machine intelligence* 34.7 (2012), pp. 1444–1450.

[59] Laura Gonçalves Ribeiro et al. “Photogrammetric Multi-Camera Calibration Using An Industrial Programmable Robotic Arm”. In: *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)* (2019), pp. 288–294.

[60] Jorge J Moré. “The Levenberg-Marquardt algorithm: implementation and theory”. In: *Numerical analysis* (1978), pp. 105–116.

[62] Adrian Kaehler and Gary Bradski. *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. " O’Reilly Media, Inc.", 2016.

[63] David Forsyth and Jean Ponce. *Computer vision: A modern approach*. Prentice hall, 2011.

[72] Francesco Barone, Marco Marrazzo, and Claudio J Oton. "Camera calibration with weighted direct linear transformation and anisotropic uncertainties of image control points". In: *Sensors* 20.4 (2020), p. 1175.

[73] Jean-Yves Bouguet and Pietro Perona. "Camera calibration from points and lines in dual-space geometry". In: *Proc. 5th European Conf. on Computer Vision* (1998), pp. 2–6.

[76] Gilbert W Stewart. "On the early history of the singular value decomposition". In: *SIAM review* 35.4 (1993), pp. 551–566.

[77] Franka Emika. "*DATA SHEET ROBOT ARM CONTROL*". In: (2020).

[78] Chris Harris, Mike Stephens, et al. "A combined corner and edge detector". In: *Alvey vision conference* 15.50 (1988), pp. 10–5244.

[79] Alexander Duda and Udo Frese. "Accurate Detection and Localization of Checkerboard Corners for Calibration." In: *BMVC* (2018), p. 126.

[81] Qi Zhang and Caihua Xiong. "A new chessboard corner detection algorithm with simple thresholding". In: *International Conference on Intelligent Robotics and Applications* (2017), pp. 532–542.

[83] Morgan Quigley et al. "ROS: an open-source Robot Operating System". In: *ICRA workshop on open source software* 3.3.2 (2009), p. 5.

[88] Péter Fankhauser et al. "Kinect v2 for mobile robot navigation: Evaluation and modeling". In: *2015 International Conference on Advanced Robotics (ICAR)* (2015), pp. 388–394.

[94] Pankaj S Parsania and Paresh V Virparia. "A comparative analysis of image interpolation algorithms". In: *International Journal of Advanced Research in Computer and Communication Engineering* 5.1 (2016), pp. 29–34.

[95] Olivier Rukundo and Hanqiang Cao. "Nearest neighbor value interpolation". In: *arXiv preprint arXiv:1211.1768* (2012).

[96] Kim T Gribbon and Donald G Bailey. "A novel approach to real-time bilinear interpolation". In: *Proceedings. DELTA 2004. Second IEEE International Workshop on Electronic Design, Test and Applications* (2004), pp. 126–131.

Sitography

[1] BMW Group Harnesses Potential of Innovative Automation and Flexible Assistance Systems in Production. 2017. URL: https://www.press.bmwgroup.com/global/article/detail/T0268199EN/bmw-group-harnesses-potential-of-innovative-automation-and-flexible-assistance-systems-in-production.

[2] Continuation of transformation plan: Mercedes-Benz plant in Bremen is to produce an additional model – the GLC Coupé – and will create 200 new jobs. 2016. URL: https://group-media.mercedes-benz.com/marsMediaSite/instance/ko.xhtml?oid=9978007&filename=Mercedes-Benz-plant-in-Bremen-is-to-produce-an-additional-model--the-GLC-Coup--and-will-create-200-new-jobs.

[4] Underground Robots: How Robotics Is Changing the Mining Industry. 2019. URL: https://eos.org/features/underground-robots-how-robotics-is-changing-the-mining-industry.

[13] How manufacturers can make the most of cobots. 2021. URL: https://www.accenture.com/us-en/blogs/industry-digitization/how-manufacturers-can-make-the-most-of-cobots#:~:text=Cobots%5C%2C%5C%20just%5C%20like%5C%20traditional%5C%20industrial,tasks%5C%2C%5C%20quality%5C%20inspection%5C%20and%5C%20packaging.

[14] How Collaborative Robots Are Being Used in Pharma. 2019. URL: https://www.automate.org/blogs/how-collaborative-robots-are-being-used-in-pharma#:~:text=Collaborative%5C%20robots%5C%2C%5C%20or%5C%20%5C%E2%5C%80%5C%9Ccobots%5C%2C,reliability%5C%2C%5C%20consistency%5C%2C%5C%20and%5C%20precision..

[17] Machine Vision Makes Collaborative Robots Even More Flexible. 2021. URL: https://www.automate.org/industry-insights/machine-vision-makes-collaborative-robots-even-more-flexible.

[18] How Vision Cobots Are Reshaping Factory Automation. URL: https://www.lanner-america.com/blog/vision-cobots-reshaping-factory-automation/.

[20] Extrinsic calibration and aperture angle. URL: https://www.eecs.tu-berlin.de/fileadmin/fg144/Courses/06WS/scanning/Dennis/Extrinsic%5C%20calibration.htm#:~:text=Extrinsic%5C%20calibration&text=get%5C%20the%5C%20orientation%5C%20between%5C%20the,take%5C%20pictures%5C%20from%5C%20one%5C%20object..

[21] Intrinsic camera parameters calibration. URL: https://mphy0026.readthedocs.io/en/latest/calibration/camera_calibration.html.

[23] Camera Calibration using OpenCV. URL: https://learnopencv.com/camera-calibration-using-opencv/.

[26] Tutorial Camera Calibration. 2021. URL: https://boofcv.org/index.php?title=Tutorial_Camera_Calibration.

[27] Calibration by manual alignment. 2015. URL: https://euratom-software.github.io/calcam/html/gui_alignment_calib.html.

[28] OpenCV Camera Calibration. 2020. URL: https://aliyasineser.medium.com/opencv-camera-calibration-e9a48bdd1844.

[33] Hand-eye calibration. 2022. URL: https://doc.rc-visard.com/latest/en/handeye_calibration.html.

[34] Robot camera calibration. 2016. URL: https://www.torsteinmyhre.name/snippets/robcam_calibration.html.

[35] Understanding the importance of 3D hand-eye calibration. 2019. URL: https://blog.zivid.com/importance-of-3d-hand-eye-calibration.

[36] Hand-Eye Calibration. 2021. URL: https://www.ensenso.com/manual/3.1/guides/multi-camera-setups-and-calibrations/handeyecalibration.html.

[43] TagDetect. 2020. URL: https://doc.rc-visard.com/latest/en/tagdetect.html.

[61] Computer Vision: Image formation and representation. URL: https://towardsdatascience.com/computer-vision-image-formation-and-representation-a63e348e16b4.

[64] What Is An Image Sensor?. 2016. URL: https://www.automate.org/blogs/what-is-an-image-sensor.

[65] Explaining Homogeneous Coordinates Projective Geometry. 2014. URL: https://www.tomdalling.com/blog/modern-opengl/explaining-homogenous-coordinates-and-projective-geometry/.

[66] Radius and Refractive Index Effects on Lens Action. 2017. URL: https://micro.magnet.fsu.edu/primer/java/lenses/lensvariations/index.html#:~:text=Higher%5C%20angles%5C%20of%5C%20curvature%5C%20lead,optical%5C%20axis%5C%20of%5C%20the%5C%20lens..

[67] What is the Circle of Confusion — Photography Definition. 2021. URL: https://www.studiobinder.com/blog/what-is-circle-of-confusion-photography/.

[68] Field of View and Angular Field of View. URL: https://www.princetoninstruments.com/learn/camera-fundamentals/field-of-view-and-angular-field-of-view#:~:text=Field%5C%20of%5C%20view%5C%20(FOV)%5C%20is,lens%5C%20and%5C%20the%5C%20sensor%5C%20size..

[69] What Is a Wide Angle Lens? 2021. URL: https://www.adorama.com/alc/faq-what-is-a-wide-angle-lens/.

[70] 3D Reconstruction Using the Direct Linear Transform with a Gabor Wavelet Based Correspondence Measure. 2007. URL: http://bardsley.org.uk/wp-content/uploads/2007/02/3d-reconstruction-using-the-direct-linear-transform.pdf.

[71] Direct Linear Transformation (DLT). 1998. URL: http://www.kwon3d.com/theory/dlt/dlt.html.

[74] Camera Calibration Toolbox for Matlab. URL: http://www.vision.caltech.edu/bouguetj/calib_doc/.

[75] ArUco: a minimal library for Augmented Reality applications based on OpenCV. URL: http://www.uco.es/investiga/grupos/ava/node/26.

[80] Camera Calibration and 3D Reconstruction. URL: https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html#ga93efa9b0aa890de240ca32b11253dd4a.

[82] ROS - Robot Operating System . URL: https://www.ros.org/.

[84] Gazebo. URL: http://gazebosim.org/.

[85] OpenCV. URL: https://opencv.org/.

[86] Intel RealSense Depth Camera D455. URL: https://www.intelrealsense.com/depth-camera-d455/.

[87] Intel RealSense LiDAR Camera L515. URL: https://www.intelrealsense.com/lidar-camera-l515/.

[89] Time-of-Flight principle. URL: https://www.terabee.com/time-of-flight-principle/.

[90] What is Lidar? URL: https://velodynelidar.com/what-is-lidar/#:~:text=and%5C%20many%5C%20more.-,How%5C%20does%5C%20lidar%5C%20work%5C%3F,calculate%5C%20the%5C%20distance%5C%20it%5C%20traveled..

[91] Kuka LBR iiwa. URL: https://www.kuka.com/it-it/prodotti-servizi/sistemi-robot/robot-industriali/lbr-iiwa.

[92] Universal Robot UR16. URL: https://www.universal-robots.com/products/ur16-robot/.

[93] Franka Emika. URL: https://www.franka.de/robot-system.