FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Desenvolvimento de árbitro automático para competição robótica Robot@Factory

Tony Williams Alves Ferreira



Master in Electrical and Computer Engineering

Supervisor: Paulo José Cerqueira Gomes da Costa

July 29, 2022

This Dissertation was co-supervised by prof. José Luís Sousa Magalhães Lima, INESC TEC.

© Tony Williams Alves Ferreira, 2022

Resumo

A robotização e automatização de tarefas, são processos complexos e de grande importância para o desenvolvimento tecnológico. Capazes de proporcionar competências intangíveis pela humanidade, têm um papel preponderante no que toca a solucionar problemas recorrentes. De forma a integrar e promover a aprendizagem destes processos, as competições robóticas têm também um impacto na educação interligada com o contínuo desenvolvimento e investigação acerca dos aspetos tecnológicos que decorrem na atualidade. Deste modo, este trabalho incide sobre uma competição robótica, Robot at Factory, desenvolvida por membros do INESC TEC, cujo objetivo consiste no desenvolvimento de um robô capaz de transportar peças por intermédio de métodos de navegação e localização, sendo-lhe atribuída uma pontuação consoante a sua prestação.

Dado o contexto tecnológico, pretende-se substituir a ação manual de arbitragem e contabilização dos pontos, por outra autónoma através da extração de dados provenientes de elementos e características presentes na competição, como os objetos a serem transportados e as respetivas zonas de descarga. Portanto, procura-se representar o espaço real num conjunto de informação computacional com recurso a técnicas de visão por computador, localização e mapeamento.

Um dos processos cruciais para alcançar esse objetivo, passa pela calibração adaptativa dos parâmetros de uma câmara digital juntamente com interação de referências visuais, designados por marcadores fiduciais do tipo ArUco.

Assim sendo, este documento retrata o desenvolvimento de uma solução capaz de exercer a tarefa autónoma de arbitragem com recurso a processos cruciais como a calibração adaptativa dos parâmetros intrínsecos e extrínsecos de uma câmara digital e simultaneamente a interação de referências visuais designadas por marcadores fiduciais do tipo ArUco. Em suma, procura-se detetar e estimar a posição das peças de interesse, de modo a verificar e premiar as tentativas do robô, realizadas com sucessos.

Abstract

The robotisation and automation of tasks are complex processes of great importance for technological development. Able to provide intangible skills for mankind, they have a preponderant role when it comes to solving recurring problems. In order to integrate and promote the learning of these processes, robotics competitions also have an impact on education interconnected with the continuous development and research on current technological aspects. Thus, this work focuses on a robotic competition, Robot at Factory, developed by members of INESC TEC, whose objective consists in the development of a robot capable of transporting parts by means of navigation and localisation methods, being awarded a score according to its performance.

Given the technological context, it is intended to replace the manual action of refereeing and counting, by an autonomous one through the extraction of data from elements and characteristics present in the competition, such as the objects to be transported and the respective unloading zones. Therefore, the aim is to represent the real space in a set of computational information using computer vision, localization and mapping techniques.

One of the crucial processes to achieve this goal is the adaptive calibration of the parameters of a digital camera together with the interaction of visual references, called fiducial markers of the ArUco type.

Therefore, this paper portrays the development of a solution capable of performing the autonomous arbitration task by resorting to crucial processes such as the adaptive calibration of the intrinsic and extrinsic parameters of a digital camera together with the interaction of visual references named ArUco fiducial markers. In short, the aim is to detect and estimate the position of the parts of interest, in order to verify and reward the robot's successful attempts.

Acknowledgements

I would like to express my gratitude to my supervisor Prof. Paulo Gomes da Costa and co-supervisor Prof. José Lima for their continuous support, advice and knowledge which guided me throughout the development of the project and made it possible to overcome difficulties.

I would like to express my appreciation to 5DPO Robot at Factory team for their support and make your time available for testing purposes.

I would like to offer my special thanks to my parents and sisters and also my partner who supported and believed in me.

Thanks should also go to my colleague João Pedro Gonçalves dos Santos for providing assistance with materials and suggestions.

Lastly, I would like to thank my colleague Vítor Samuel Pinto Babo and all my friends from "Bros" for all the encouragement and support through my studies.

Tony Williams Alves Ferreira

"Four things for success: work and pray, think and believe."

Norman Vincent Peale

Contents

1 Introduction **1**
- 1.1 Context . . . 1
- 1.2 Problem Definition and Motivation . . . 2
- 1.3 Goals . . . 2
- 1.4 Document Structure . . . 2

2 State of the Art **5**
- 2.1 Robotics Competitions . . . 5
- 2.2 Fiducial Markers . . . 7
  - 2.2.1 Types . . . 7
  - 2.2.2 Comparison . . . 10
- 2.3 Camera . . . 11
  - 2.3.1 Parameters - *Background* . . . 11
  - 2.3.2 Calibration . . . 13

3 Methodology **17**
- 3.1 Competition Concept and Structure . . . 18
  - 3.1.1 Competition and Tasks . . . 19
- 3.2 Selection of the Fiducial Marker . . . 20
- 3.3 Field Mapping . . . 22
- 3.4 Intrinsic Calibration . . . 22
- 3.5 Extrinsic Calibration . . . 23
- 3.6 ArUcos Detection . . . 24
- 3.7 Pose Estimation . . . 24
- 3.8 Pose Verification . . . 26
- 3.9 Competition Modifications . . . 26

4 Implementation **27**
- 4.1 Global Architecture . . . 27
- 4.2 Piece Detection . . . 28
- 4.3 Clock System . . . 31
- 4.4 Referee System . . . 32
- 4.5 Interface . . . 33
- 4.6 Simulation and Scaled Model . . . 35
- 4.7 Fault Tolerance . . . 38

5 Testing and Results **41**
5.1 Raw Type Scenario Through Simulation 41
5.2 Pose Estimation Error 42
5.2.1 Field Tests 43
5.2.2 Piece Tests 46
5.2.3 Camera Analysis 48
5.3 Scoring Results 49

6 Conclusion and Future Work **53**

References **55**

List of Figures

2.1 Different types of fiducial markers for computer vision systems. First row, from left to right: ARToolKit, Maxicode, Cantag, and Topotag. Second row, from left to right: AprilTag, ArTag, CALTag, and ArUco [35]. 8
2.2 Representation of the *pinhole* model and its parameters [43]. 12
2.3 Lens radial distortion: barrel on the left, and *pincushion* on the right. 14

3.1 Flowchart of the global referee system. 17
3.2 Field track layout [7]. 19
3.3 Left: Piece that is used in the Robot at Factory competition; Right: Actual attempt. 20
3.4 Rotations on cartesian axes *x, y* and *z*. 21
3.5 Measurements to map all four zones. 23
3.6 From left to right: Low contrast between piece, marker and wall; low contrast between piece, marker and robot; white paper addition to the piece and wall for higher contrast. 26

4.1 Raspberry Pi and camera supporting structure. 28
4.2 Chessboard marker pictures from the Raspberry Pi Camera Module v2 to perform the intrinsic calibration. 29
4.3 Left: Clock interface with the implemented grid at the bottom; Right: State updates according to the current on-field scenario. 31
4.4 Position smoothing and Infinite Impulse Response filter algorithm 33
4.5 Flowchart of the referee algorithm. 34
4.6 Flowchart of the global architecture interactions. 35
4.7 Interface of the referee system: user interactions on the left and virtual field information on the right. 36
4.8 Referee simulation and testing of a correctly placed green piece. 36
4.9 Pieces information through text storage. 37
4.10 A3 scaled field-testing version with camera setup and an ArUco marker of ID 40. 37
4.11 Referee, Clock, and RPI camera concurrently execution and interaction with the detection of a red type ArUco piece, ID 40. 38

5.1 Test piece simulated displacement over time on *x* and *y* axis. 42
5.2 The final state of the interface is on the middle and the piece path on the right. . . 42
5.3 Position estimation error and number of samples for each detected field marker from a lateral view. 43
5.4 Position estimation error and number of samples for each detected field marker from a longitudinal view. 44
5.5 Position estimation error and number of samples for each detected field marker from a diagonal view. 44

5.6 Position estimation error and number of samples for each detected field marker from a top view. 45
5.7 Vertical camera setup for top-view pose estimation. 45
5.8 Position estimation error and number of samples for each detected field marker from a partial view. 46
5.9 Position estimation error and number of samples for each detected piece marker from Y and X perspectives. 48
5.10 Position estimation error and number of samples for each detected piece marker from a diagonal perspective without (left) and with (right) the IIR filter. 49
5.11 Complete system execution for piece poses verification in round 1 of the competition. 50
5.12 Full system execution for piece poses verification in round 2 of the competition. . 51
5.13 Full system execution for piece poses verification in round 3 of the competition. . 52

List of Tables

2.1 Chapter 2 nomenclature. 5
2.2 Comparison of the camera calibration algorithms [27]. 15

3.1 Chapter 3 nomenclature. 18

4.1 Chapter 4 nomenclature. 27

Abreviations and Symbols

AGV	Automated Guided Vehicle
UAV	Unmanned Aerial Vehicle
FM	Fiducial Marker
PnP	Perspective-n-Point
STEM	Science, Technlogy, Education and Mathematics
CMRE	Center for Marine Research and Experimentation
SAUC-E	Student Autonomous Underwater Challenge - Europe
ERL	European Robotics League
MBZIBC	Mohamed Bin Zayed International Robotics Challenge
RaF	Robot at Factory
SLAM	Simultaneous Localisation and Mapping
AR	Augmented Reality
ID	Identifier
DLT	Direct Linear Transformation
IMU	Inertial Measurement Unit
RPI	Raspberry Pi
ZMQ	ZeroMQ
IIR	Infinite Impulse Response

Chapter 1

Introduction

In this chapter a contextualisation is given about the stated competition with an indication of the objectives to be met through the development of the project as well as the documents structure.

1.1 Context

Over time, we encounter the continuous growth and adaptation of technology whose usefulness falls into a wide and diverse horizon of applications. One aspect of this evolution is featured in automated and robotic systems, where their development and implementation grant the ability to cooperate and perform tasks that exceed human limits. This application can be seen in many components, for example, for industrial purposes, such as industrial manipulators and autonomously guided vehicles (AGV), studies and projects developed by Boston Dynamics, or even the Roomba Robot that you may have at home.

While these applications all focus on the high-end of the spectrum, it does not imply that an individual motivated by the field cannot participate or engage with this technology-intensive environment. In fact, with the right equipment, the integration of low-scale automated and/or robotic systems is possible and therefore tangible to anyone. At the same time, competitions associated with the area in question also represent a great search engine for participants who want to face new challenges.

Therefore, this Dissertation is in the scope of a robotics competition created by members of the INESC TEC [7]. This competition is characterized by a set of participants that try to maximize the score obtained by moving and unloading parts from their original location to a later desired and correct position. Competitors are responsible for moving the parts from an "Incoming Warehouse", ending up at one or several desired later positions depending on the respective round ("Outgoing Warehouse", "Machine-A" and/or "Machine-B"), and are evaluated according to the score they achieve with priority over the completion time.

1.2 Problem Definition and Motivation

A common factor of all the Robot at Factory that has existed in previous versions of the competition is the counting and scoring system. Although this competitive environment is found on a technologically developed context, it can be seen that the counting of the points is done manually by an assistant referee. The competition referee is in charge of counting the correctly transported parts, whose trajectory culminates in a machine or an exit warehouse, according to the round in which the attempt is being made.

In order to speed up this process, it is intended to be autonomous by means of computer vision concepts in a robotics environment, such as the types of fiducial markers and their detection algorithms, camera parameters and calibration, referred to later in this paper. This alternative will serve as a support to the associated members of the event, allowing a better organization of the information associated with the teams, rounds and overall competition.

1.3 Goals

The objective of this Dissertation is to develop a solution that satisfies the functional requirements of the referee system that will be used in the competition. This will be based on the study and analysis of the state of the art for fiducial markers and characterization of the camera in order to obtain all the needed data from the field. In summary, the objectives are as follows:

- **Camera calibration method** – Algorithm to estimate the intrinsic and extrinsic parameters of the camera succeeding a 2-D pixel to 3-D world point projection.
- **Piece Identification and Localization** – Methodology to allow the detection of each piece enabling the correspondent pose estimation.
- **Development of the referee algorithm** – Assurance of a correct scoring system, able to accept or reject the current states of a given round. This component fundamentally has the actual referee verification's with the ability to convert the manual task to an automated one.
- **Verification and Validation** – Environment that will evaluate the concept.

1.4 Document Structure

The rest of the document is structured according to the following chapters:

- Chapter 2: State of the Art – *background*, related work on the subject, and preliminary studies.
- Chapter 3: Methodology – methodology used to achieve the defined requirements.
- Chapter 4: Implementation – tools, technologies and procedures developed to achieve a fully functional system.

- Chapter 5: Testing and Results – analysis of the system performance and requirements validation.
- Chapter 6: Conclusions – main conclusions that are relevant to mention and restating the important information taken from the development of the project.

Chapter 2

State of the Art

This chapter analyses the most relevant topics that focus on the problem associated with this project. A theoretical foundation will be made about the topics in question, along with a current retrospective with articles and studies previously done by several authors.

Table 2.1: Chapter 2 nomenclature.

Variable	Designation
$\mathbf{T}$	Translation Matrix
$\mathbf{R}$	Rotation Matrix
$\mathbf{M}$	Intrinsic Matrix
Z	Camera Focal Plane
f	Focal Length
m	Size on the projected image
γ	Skew Coefficient
(x_p, y_p, z_p)	Real Point
(μ, ν)	Image Point
(μ_d, ν_d)	Distorted Image Point
$\delta(\mu, \nu)$	Distortion Function
$k1, k2$	Radial Distortion
$p1, p2$	Decentering Distortion
$q1, q2$	Thin Prism Distortion

2.1 Robotics Competitions

The areas of automation and robotics are very appealing. Their complexity can have an inherently intimidating aspect from the viewpoint of spectators. Thus, competitions can be held in order to provide reasons and interest to participate in a given imposed challenge, allowing the development of skills and knowledge about the subject. These challenges also represent a great search engine for participants who want to face new challenges. Programming, kinematics, dynamics, control, computer vision, etc., are topics that can be acquired through this environment.

Nowadays, the competition in question and the realisation of similar projects involving robotics and automation concepts (and many other distinct areas), such as the work referred to in this document, are seen as a "vehicle" for representing the fields of Engineering, the realisation of which is expected to generate and captivate interest among technology enthusiasts, causing impact and attraction of these fields to young audiences.

Therefore, several advantages can be associated with this type of competition:

- increased expertise;
- development of hard and soft skills;
- attracting students to Engineering areas;
- encouraging and building the spirit of creativity, innovation and even intra-team cooperation;
- gratification, recognition and education;
- possibility of creating high potential projects that deserve its development continuity.

Montemerlo *et al.* [29] publicise the construction of a robotic vehicle capable of autonomously navigating an urban environment, creating its trajectories, interacting with the environment and even parking, among others. Consequently, their performance was worthy of a podium in a robotic competition held. The competition was focused on autonomous vehicle driving concepts, whose implementation and use are, nowadays, observed in *Tesla* vehicles, for example.

Kitano *et al.* (1997) [21] took the initiative to create a robotics competition to integrate concepts of artificial intelligence, sensory function, multi-collaboration, robot autonomy, etc., through a standard platform for a robotics soccer game. This competition still occurs nowadays, once a year, and has evolved to the point where humanoid robots are now used [16]. Currently, Portugal has a team composed of university students called FCPortugal3D, which retains the title of three-time European Robotic Soccer Champion. Similarly, RobotCupJunior describes the same context of a soccer robotics competition with a slight change with the intent to focus on education and learning through the Science Technology Engineering and Mathematics (STEM) concept. Amy Eguchi [11] emphasises the educational benefits and the impact of the junior version by analysing results based on robotics interest, STEM and expertise learnt.

Additionally to the humanoid robots, DARPA Robotics Challenge encourages the development of ground human-supervised and assistance robots to respond to disastrous scenarios. Radford *et al.* [31] extended its application spectrum by considering extraterrestrial settings for human assistance.

Such competitions are also available in the marine area. Ferri *et al.* [12] refers to challenges held by the Center for Marine Research and Experimentation (CMRE), whose goal is to promote the development and investigation of the current state of the technology involved. The competition is called *Student Autonomous Underwater Challenge - Europe* (SAUC-E), and its positive impact

led first to participate in the euRathlon and later in the European Robotics League (ERL) where emergency robots were developed.

Furthermore, the aerial aspect of robotics, such as unmanned aerial vehicles, is also targeted to be hosted by the robotics competition environment. Bachrach *et al.* [2] showcases the development of micro aerial vehicles with the intent of achieving autonomous navigation in constrained environments to perform rescue tasks, whose validation was also proven through International Aerial Robotics Competitions.

Mohamed Bin Zayed International Robotics Challenge (MBZIBC) hosts competitions that focus on the development of aerial and surface innovative technologies to push the boundaries of automation and robotics applications. Baca *et al.* [1] demonstrates the result of an autonomous landing system for MBZIBC that focuses on trajectory intersection, path planning, control and perception. Similarly to the Robot at Factory (RaF) challenge that is stated in this document, Surpný *et al.* [39] also participated in the MBZIRC Treasure Hunt challenge. It is characterised for pieces the detection and displacement from an initial position to their predefined final destination, rewarded for each correct attempt. While RaF is focused on ground bidirectional and omnidirectional robots, MBZIRC Treasure Hunt is directed to UAVs.

Bellicoso *et al.* [3] showcases the application of quadrupedal robots capable of autonomous navigation, industrial inspection and payload delivery in outdoor scenarios, thus participating in Autonomous Robots for Gas and Oil Sites (ARGOS) and ERL challenges.

Outside the realm of robotics, many other platforms design competitions with the same goal. An illustration of this is the Eurekathon, which seeks to create challenges involving data science skills associated with sustainability.

2.2 Fiducial Markers

Fiducial markers (FM) are visual elements or objects within range of a camera's field of view and contain unique design properties that are automatically detectable by a computer vision system. They can be used in different areas whose interest is based on the detection of these elements to define a reference or measurement point. Their application can be associated with means of navigation, such as planning, localisation and mapping [33, 14], reference for the measurement [28], or object tracking [38].

The associated computerised system requires high detection accuracy and versatility to provide a fast and on-time data association for a given application, enabling navigation methods like Simultaneous Localisation and Mapping (SLAM) [8].In the context of robotics, these markers are used as artificial *landmark* or reference points and are defined as *augmented reality* (AR) markers [35, 45], which are most relevant to be subsequently mentioned.

2.2.1 Types

According to what is presented by M. Hirzen [15], there are some properties to consider for the creation of a fiducial marker. An outer edge characterises the vast majority of the tag types to

enable distinction and consequently the marker detection; and its inner edge, where the information to be decoded is contained. These aspects restrict how these are created. Firstly, the issue of sensitivity to lighting conditions. For this, we choose dual-tone types (black and white), so there is no need to identify shades of grey. Hereafter, it is necessary to decide how much information will be stored on the given marker, which influences the allowable decoding distance. So often, only the identifier (ID) of the tag is used.

Markers can be divided mainly into two categories: circular and square. In the following figure 2.1, some examples are illustrated.

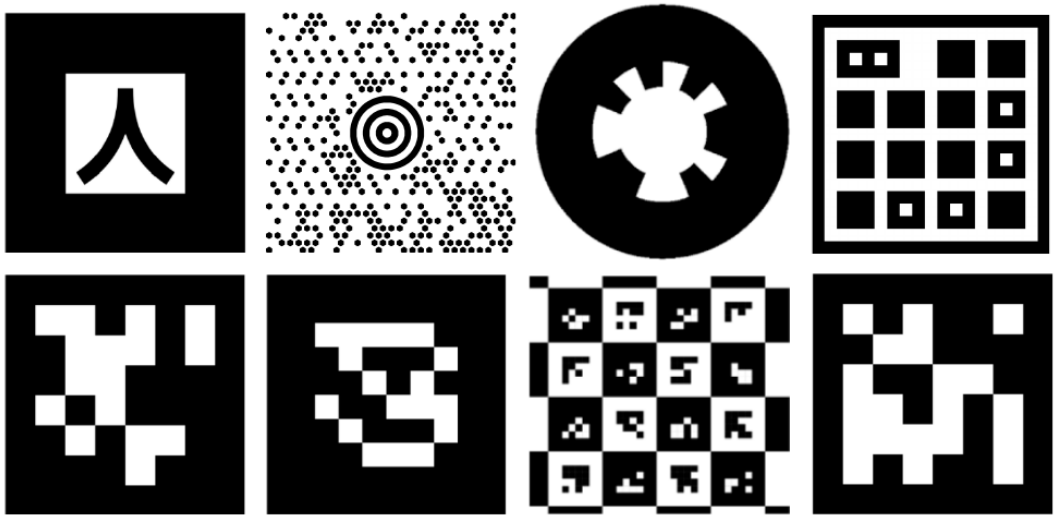


Figure 2.1: Different types of fiducial markers for computer vision systems. First row, from left to right: ARToolKit, Maxicode, Cantag, and Topotag. Second row, from left to right: AprilTag, ArTag, CALTag, and ArUco [35].

With that said, some types of fiducial markers presented by Shabaline *et al.* [35] are considered, among others:

- **ARTag** – marker inspired by the *ARToolKit* ([10], ch. III) and surpasses it by exhibiting greater versatility, resilience to lighting variations and by its ability to avoid identifying false positives. In contrast, its performance is dependent on complex data processing. Edge detection is used to determine the sides of the pixel so that a quadrilateral can be formed. The interior is decomposed by 36 cells, white or black, according to a 6x6 grid. It is on this grid that the ID and redundancy bits data are stored.

  According to Limosani et al., [26] this marker has more reliability, detectability and a greater detection range.

- **AprilTag** – It allows the calculation of the marker identifier, the exact position and orientation with respect to the camera. This marker is characterised by increased internal patterns, reduced false positives, and the total number of bits and sizes.

Additionally, Wang and Olson [42] describe a new detection process distinguished from the original AprilTag by improved robustness, efficiency, performance, lower computational burden, and false-positive detection.

- **CALTag** [5] – marker specifically developed for camera calibration purposes and characterised by its resemblance to a chessboard on its inside. This interior is decomposed by a grid of black and white squares, where it is possible to increase or decrease its density to provide more or less reliability and recognition efficiency. For each potential cell initially detected, its interior is verified and validated. When the point detection fails, the algorithm associated with CALTag is applied for correction and restoration. The defined density influences the size of the marker.

- **Circular Data Matrix** – A circular marker characterised by non-ambiguous identification points, the ability to generate a wide variety of internal codes, and its high detection reliability. It is composed of two black rings (inner and outer) and a white dot in the centre. The identification dots provide orientation, and the one between the rings contains the information to be decoded. The detection algorithm developed by Naimark and Foxlin [30] was designed to solve problems related to lighting variation.

- **TopoTag** [44] – Use of topographical information to improve its robustness and, consequently, detection accuracy. This marker provides additional characteristic points allowing better position estimation. According to the author, it also supports some extension and flexibility regarding occlusion and noise scenarios, therefore being a good element for robotic applications. Additionally, this FM allows interior and exterior shape customisation since it only needs to detect the interior nodes. After the experiments, this FM showed more robustness in position estimation and very noticeable results in orientation detection.

- **CanTag** [34] – An *open-source* creation system whose marker can have two types: circular and square. The first is primarily used for localisation and position estimation scenarios. The second allows the decoding of more information. The system is implemented in C++ and provides a variety of algorithms associated with image processing and marker detection. This variety enables the algorithm to detect multiple markers in a given scenario or dynamically change its processing to meet the application's needs.

- **ArUco** [24] – A well-developed system that allows the creation of a marker without the size limitation of the interior grid. It is run through a dictionary method that generates the desired marker and provides a detection and recognition algorithm that is complete and robust. During occlusion scenarios, it can be supplemented with processes that bypass this problem.

- **STag** – A marker designed to achieve a stable location without using temporal filtering. This type combines the design of a square tag for its exterior and initial homographic estimation; and a circular interior for the estimation refinement. Benligiray *et al.* [4] seek to achieve

robust detection that provides an increased detection range and angle by integrating these two simple geometric features. Their detection includes three parts: candidate detection, where the squares of the image are detected; candidate validation, through its shape (outer square) and internal encoding; and homographic refinement, where the inner circumference is detected for better location estimation.

- **MechaTag** [9] – This marker, unlike all the ones mentioned above, is rigid and is intended to be resilient to physically demanding operating environments with varying lighting, temperature, and humidity. It is a simple and robust marker whose detection only requires two circumferences of different diameters. Its rigidity and detection points provide high reliability, accuracy, and minimisation of position estimation errors.

2.2.2 Comparison

While there are visual similarities between some of the markers, the different designations are mainly associated with the respective detection and decoding algorithm used by each marker type. Consequently, it influences the resilience and versatility properties in occlusion, noise, rotation, illumination, and internal confusion, among other scenarios [20]. Nextly, some studies are illustrated that seek to compare the performance of different types of fiducial markers, evaluating them according to properties.

K. Shabalina *et al.* [35] manually compare the detection of ARTag, AprilTag and CALTag fiducial systems in the presence of occlusion and rotation around the *x, y* and *z* axis. In the first case, the ARTag system was the only one that showed some detection sensitivity. For the second analysis, a simple and a complex object were considered. In the simple case, both the ARTg and AprilTag systems showed sensitivity as opposed to the complex case where AprilTag showed more difficulty than ARTag. In these two cases, CALTag was the system that obtained superior and more acceptable results due to the corner overlap in the first two systems constrained the detection algorithm, whereas, in the CALTag system, it is possible to make a point recovery through the successfully detected squares. The experiment scenario also involved the quality influence of the camera responsible for capturing the image of the markers. Under the same tests, there was a better detection rate relative to rotation in the presence of a higher quality camera, but not enough to achieve the same results as the CALTag system, which was the best performing system.

Then, A. Zakiev *et al.* [45] performed a comparison through a simulator, *Gazebo*, providing a controlled test environment. In this study, the ArUco and AprilTag system were considered against rotations and sensory noise association (Gaussian of zero mean and variable standard deviation). The range from no noise (0%) to moderate noise (5%), ArUco showed greater versatility with respect to normal (x) and lateral (at y or z) rotations. The detection rate is considerably compromised for a sensor with high noise (10%).

M. Kalaitzakis *et al.* [17, 18] seek to evaluate the localisation accuracy and robustness of ARTag, AprilTag, ArUco and STag markers. Among the four systems, the mean deviation, standard deviation and detection rate were measured associated with different detection systems, varying illumination, distance and rotation variation, image quality and computational weight. The experiment is carried out with other cameras (*Raspberry Pi* Camera module 2, for low resolution, and Logitech C270, for high resolution) and different processors (NVidia Jetson TX2 and Intel NUC). Regarding the markers, ARTag and ArUco prove to be the most computationally efficient, resulting in a lower processing burden. AprilTag and Stag were the ones that presented the best results in terms of accuracy. It is worth mentioning the ArUco system's comparable performance to the other two markers (table 13 of the article).

Finally, J. Kallwies *et al.* [19] check the influence of different libraries for the detection of AprilTag type markers. Since they are compatible, it should be mentioned that C++ and *openCV* libraries are used for AprilTag and ArUco. Additional methods were also used to improve the overall performance. In summary, the performance of ArUco *openCV* is noteworthy for its lower computational weight, remarkable localisation accuracy for AprilTag C++ and the detection of small markers for AprilTag 3.

According to the markers identified in the previous section, it is noted that only a few of the mentioned types are studied and compared. In the context of robotics, the markers referenced in this section proved to be the most common and suitable for a robotic environment.

2.3 Camera

Complementary to the previous section, the fiducial markers must be associated with a vision system capable of representing the actual space in a set of computational information. A camera makes this interconnection, and some associated concepts are taken into account for its integration.

Therefore, the camera is the device that ensures a relationship between the real world and the digital world. It behaves as a sensor that receives 3-D (real) information as input and represents it in a 2-D image, where the detection of markers will later be performed. This dimensional transformation uses a mathematical representation that translates the operating model of a camera. The model is characterised by extrinsic and intrinsic parameters that co-relate the coordinates and internal values.

According to this model, it is possible to map a point in the 3D referential to the 2D referential with the help of mathematical expressions that relate external and internal factors to the camera, or as they are called parameters.

2.3.1 Parameters - *Background*

The extrinsic parameters represent the location of the camera in space and are decomposed by 6 degrees of freedom: 3 for the position (x, y and z) and 3 for orientation (α, β and γ), defining a relationship between the coordinates of a point in the real referential and the respective coordinates of the camera through the translation matrix (**T**) for the position, and rotation (**R**), for the

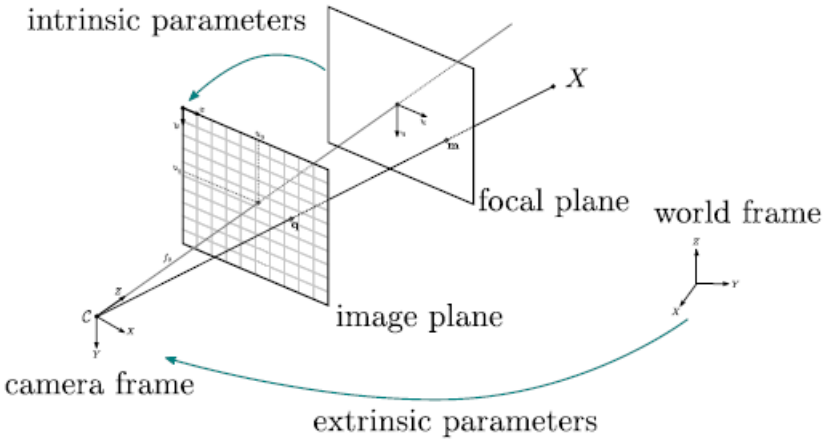


Figure 2.2: Representation of the *pinhole* model and its parameters [43].

orientation [36]. With these matrices, it is also possible to express the camera's centre (C) relative to the real referential: $C = \boldsymbol{R}^T \boldsymbol{T}$.

$$[\boldsymbol{T}_{3x1}]^T = [t_x, t_y, t_z]^T \tag{2.1}$$

$$\boldsymbol{R}_{3x3} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2.2}$$

In turn, the intrinsics constitute all the internal values of the camera and describe how a three-dimensional point is mapped into a two-dimensional image. These parameters include the focal length (f) of x and y (where the relationship is made between the distance from the camera's referential to the image plane of the *pinhole* model, with the corresponding height and width of the projected image, m); the principal point, (u_0, v_0), (ideally at the centre of the image and illustrates the point where the camera optical axis passes); and the skew coefficient, γ, typically equal to 0 depending on the type of the camera, whether it is analogue or digital [36]. From this, the intrinsic matrix **M** results and is represented as follows:

$$\boldsymbol{M}_{3x3} = \begin{bmatrix} f_x.m_x & \gamma & u_0 \\ 0 & f_y.m_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

By defining both parameters, it is possible to project a point in space onto the image plane establishing a singular relationship between a 3-D coordinate and a 2-D point, according to the *pinhole* model and matrices M, R, and T. Through the entire coordinate system, a projection of a real point P = $[x_p, y_p, z_p, 1]^T$ is made to the computerised coordinate system, in pixel, $Z[u, v, 1]^T$, where Z is the focal plane [43].

$$Z\begin{bmatrix}\mu\\ \nu\\ 1\end{bmatrix} = \boldsymbol{M}\begin{bmatrix}\boldsymbol{R} & \boldsymbol{T}\end{bmatrix}\begin{bmatrix}x_p\\ y_p\\ z_p\\ 1\end{bmatrix} \tag{2.4}$$

Knowing how the points in the real referential and the camera referential are related, it is now possible to map a pixel value to the world system and vice-versa. However, the relation obtained from the *pinhole* camera model does not consider the effect resulting from the lens distortion. Therefore, a distinct model arises by correcting the distorted projected point (μ_d, ν_d) concerning the distortion components (δ) with the longitudinal and latitudinal axis of the image plane. As presented in J. Wang *et al.* work [41], the distortion coefficients are defined by k_1, k_2, p_1, p_2, q_1 and q_2, where they are coupled as radial, decentering and thin prism distortion, respectively. Hence, the following mathematical model:

$$\mu = \mu_d + \delta(\mu_d, \nu_d)\nu = \nu_d + \delta(\mu_d, \nu_d) \tag{2.5}$$

$$\delta_\mu(\mu,\nu) = k_1\mu(\mu^2+\nu^2) + k_2\mu(\mu^2+\nu^2)^2 + q_1(\mu^2+\nu^2) + \mu(p_1\mu + p_2\nu\delta_\nu(\mu,\nu) \tag{2.6}$$

$$= k_1\nu(\mu^2+\nu^2) + k_2\nu(\mu^2+\nu^2)^2 + q_2(\mu^2+\nu^2) + \nu(p_1\mu + p_2\nu) \tag{2.7}$$

2.3.2 Calibration

Given the importance of the parameters, the values are determined through a process called calibration, and an estimation of the camera parameters is made. This resolution is generally carried out according to four methods:

- Direct Linear Transformation (DLT): where a camera approximation is made and assumes a perfect lens without distortion. It is a two-step process: solving the homogeneous equation and factoring in the parameters.

  Currently, this method has already changed to make this process more robust by considering more factors that influence the accuracy of the calibration.

  The authors [46] propose an improved method, *"Improved Direct Linear Transformation"*, that relates the intrinsic parameters to the extrinsic ones since an error in one of the parameters can be compensated in the other. This is modelled as a coupling error and allows a linear relationship between the two parameters, minimising calibration errors.

  Furthermore, Shi *et al.* [37] extend the application of this method by giving importance to the lens distortion, which was initially not considered. The camera representation with the tangential and radial lens distortion is added to refine the calibration in question.

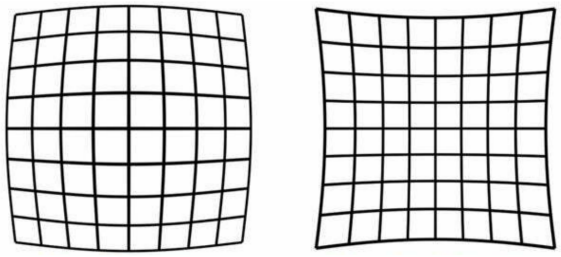


Figure 2.3: Lens radial distortion: barrel on the left, and *pincushion* on the right.

- Traditional Method: based on experimental conditions where is used objects/visual elements with dimensions known a priori. The best known and widely used method in the context of automation and robotics, among others, is the one defined by *Zhang Zhengyou*. This method stands out for its accuracy and operability, and the factor associated with lens distortion (figure 2.3) is already considered when obtaining the intrinsic values. This method takes advantage of the equations 2.2 and 2.3 for the corresponding calibration. Multiple images are also required to calculate the five variables of the internal parameters to subsequently calculate the external parameters.

  X. Liang *et al.* [25], with a monocular camera, demonstrates the application of this method to estimate the intrinsic parameters and combines it with the *Perspective-n-Point* (PnP) to estimate the external parameters. The PnP problem raises the question of how many points are needed to correctly estimate the position of a calibrated camera.

  Furthermore, PnP raises the computational processing weight aspect due to the number of points used and the accuracy accomplished. With the intent to overcome this factor, Lepetit *et al.* [23] have developed a new linear algorithm (O(n)) that overrides the constraint on the number of points that can be chosen for the algorithm. Furthermore, this method also takes advantage of an efficient Gauss-Newton optimisation to refine its accuracy.

  Alternatively, the camera extrinsic parameters can also be determined by applying the *vanishing points*. Fundamentally, this concept is based on the appearing intersecting point of two parallel straight lines on the world when converted from 3-D world coordinates to 2D image coordinates as a consequence of perspective projection [6].

- Autonomous Calibration Method: determines the parameters through the images taken on the scene without needing any reference object. It is a flexible yet not very accurate method. C. Lei *et al.* [22] demonstrate the application of this method supported with the Kruppa Equations.

  Similarly, Z. Tang *et al.* [40] have used *vanishing points* to dynamically estimate the parameters of a static camera employing 3-D object tracking.

- Active Vision-based method: the camera parameters are controlled by a set of external agents that have the ability to adapt over time, providing high robustness to the system

against the quality of the equipment used for calibration. The control is done autonomously, and the movement information is known in the system, allowing a simple and robust calibration, usually resulting in a linear solution.

Rebelo *et al.* [32] exemplify the use of this method. The flexibility of the system to adapt and capture enough information to ensure a given desired accuracy of the ongoing calibration is notable.

According to what has been gathered to this point, the following conclusions can be drawn regarding the methods mentioned, as presented in table 2.2:

Table 2.2: Comparison of the camera calibration algorithms [27].

Calibration Algorithms	Advantages	Disadvantages	Application
Traditional Calibration	High Accuracy	Complexity. The need for known objects	accuracy is a requirement; system with fixed parameters.
Self-Calibration	Object-Independent. Flexibility	Low accuracy	Variable location scenarios. Absence of reference elements.
Active calibration	Simplicity of calculation	High system cost	Current camera information are required.

Chapter 3

Methodology

This chapter will discuss the environment, structure, components, tools, and procedures that are crucial to achieving the correct operation of the referee system to be used in the Robot@Factory robotics competition. The developed system that will be responsible for the scorekeeping consists of 5 main stages, as illustrated in the figure 3.1:

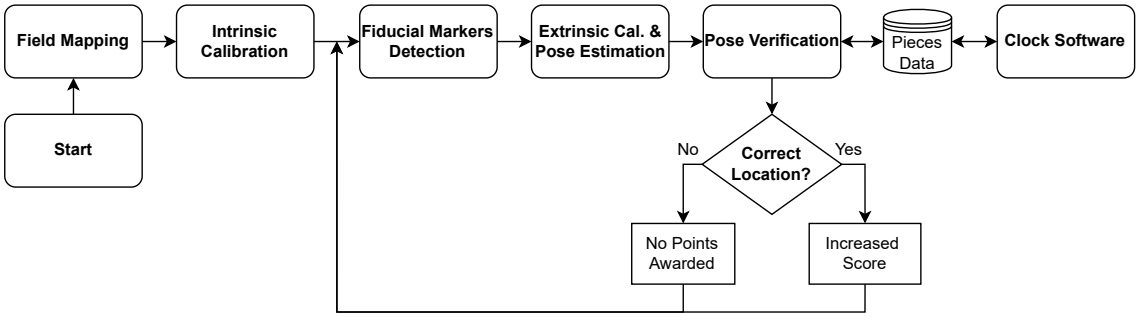


Figure 3.1: Flowchart of the global referee system.

1. **Field Mapping** – Identification of the zones corresponding to the ingoing warehouse, machine A, machine B and outgoing warehouse;

2. **Calibration** – Application of the calibration method to parameterise the camera internally and in space;

3. **Detection of Fiducial Markers** – Use of the detection library for the fiducial marker identification;

4. **Position Estimation** – Estimate the position of the pieces on the field;

5. **Position Verification** – According to the field mapping and estimated position, verify that it is in the right zone with the correct piece type to assign the score.

The steps will be discussed and further developed in the following sections.

Table 3.1: Chapter 3 nomenclature.

Variable	Designation
$\mathbf{T}$	Translation Matrix
$\boldsymbol{T}_3$	Third row of the Translation Matrix
$\mathbf{R}$	Rotation Matrix
$\boldsymbol{R}_3^{-1}$	Third row of the Inverse Rotation Matrix
$\mathbf{M}$	Intrinsic Matrix
$\boldsymbol{M}_3^{-1}$	Third row of the Inverse Intrinsic Matrix
$\mathbf{K}$	Distortion Coefficients Matrix
Z	Camera Focal Plane
(x_p, y_p, z_p)	Real Point
(μ_c, ν_c)	ArUco Centre Point

3.1 Competition Concept and Structure

Before jumping into the procedures needed to achieve the desired performance of the referee system, some understanding of key aspects of the competition must be known to continue.

The Robot at Factory is an academic and international competition within the scope of robotics that showcases Autonomous Guided Vehicles (AGVs) and rewards those who implement the best and fastest methodology in regards of:

1. autonomous localisation, with odometry, Inertial Measurement Unit (IMUs), LiDAR, landmark identification, etc.;

2. and path/trajectory planning supported by, e.g. roadmap algorithm, cell decomposition, potential fields, graph search, A* family algorithms, among others;

3. scheduling, decision-making and optimisation of the piece pick-up order.

This competition is designed to simulate, through an on-scale model, a warehouse and factory system where some initially stored components, with a distinct type, are moved by an autonomous robot to either an operating section that modifies their properties or the expedition section delivering the finished and desired product. Throughout this document, the components that are transported will be called "parts" or "pieces", and they can have one of three types: Raw (red), Intermediate (green) and Final (blue), meaning that a part needs two, one or no operations, respectively. When the part has no operations scheduled, with the blue colour, it means that it is the finished product and can be delivered to the output.

The environment is characterised by a planar 1,7m x 1,2m ground track with multiple black and white square markers and non-filled rectangles on fixed positions.

These 5cm x 5cm squares are a type of fiducial markers, described as Augmented Reality markers, that are known as ArUco. Those are detected with a detection algorithm after being captured by a camera, and the robots use them as landmarks for mapping and localisation purposes.

Each rectangle, or box, represents the area of interest of a given section in the field, which will also correspond to destination points for the robot. Each section has four 9cm x 6cm, or

6cm x 9cm, rectangle areas and they are distinguished into four groups: Incoming Warehouse, Machine-A, Machine-B and Outgoing Warehouse.

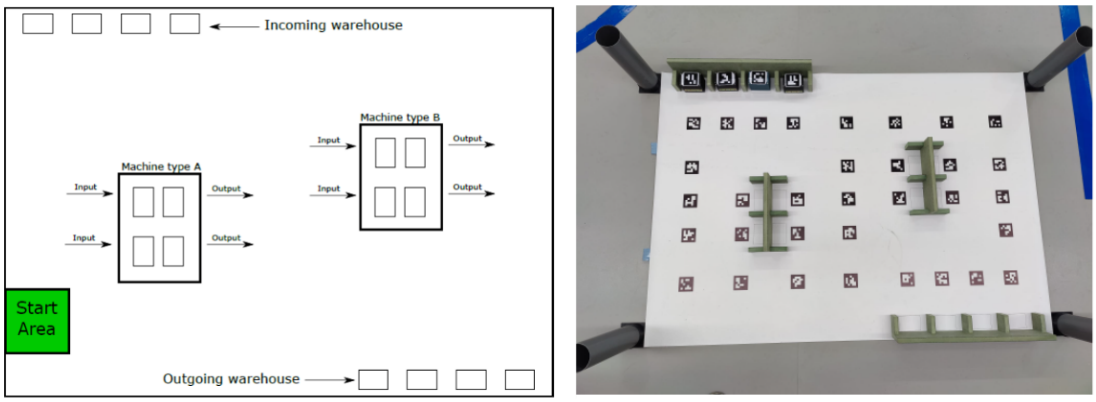


Figure 3.2: Field track layout [7].

The Incoming Warehouse section, on the top-left side, is the spawning (or input) area, which means that four parts will start in this section, one in each box and set its starting position. This is where the robot has first contact and picks up the pieces.

Machine-A and Machine-B, at the middle-left and middle-right, depict the operating areas that will result in a change of the parts type. Parts received as Red will be transformed into Green Parts, and Green ones will result in a Blue and thus finished part. Each machine has two input boxes, where the part will be dropped and processed, and two corresponding output boxes with a successfully processed part with its type updated and ready to be picked. The robot does the loading and unloading of the Parts into the machine. Furthermore, the robot must be able to communicate with the competition server to be synchronised data-wise with the state of an attempt. This communication is done via UDP to know the Ingoing Warehouse configuration for the succeeding piece displacement.

Beforehand, the robot sends an "IWP" message, and the server responds with the colour of the pieces that are placed on the input boxes. For example, if the input configuration has, from left to right, pieces that are green, blue, red and green, then the server responds to "IWP" with "GBRG".

Finally, the Outgoing Warehouse section, at the bottom-right, has a similar configuration as the Incoming Warehouse and represents the expedition process. It is the part's final destination and is used to place the finished components.

These are the most significant sections to be mapped into the referee system to evaluate if a piece is placed in some of these positions. The piece is a 9x6x6cm object with a small metal plate to allow transportation by the robot via electromagnet, as illustrated in the figure 3.3.

3.1.1 Competition and Tasks

Now that the physical aspects of the competition are clarified, we can now redirect the focus to the inside point-of-view in regard to the rules and procedures taking place on the field.

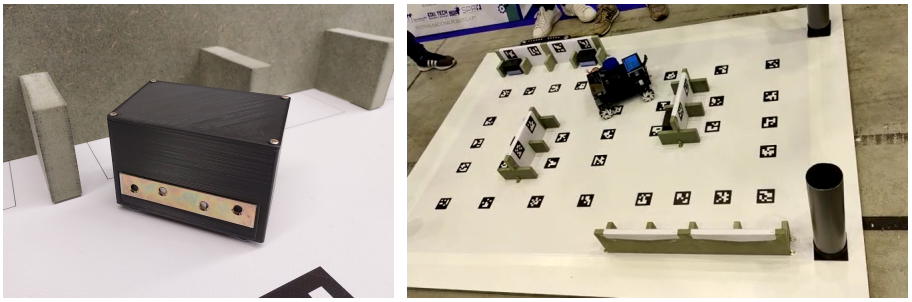


Figure 3.3: Left: Piece that is used in the Robot at Factory competition; Right: Actual attempt.
.

This competition is carried out through 3 incrementally difficult rounds of 10 minutes where the participants must maximise their score in due time with the robot. One point is attributed each time the part is correctly placed on machinery or output, and each round adds a new Part type, meaning that more destinations, operations and points are at stake. On the first and easiest round, there are only spawned blue-type pieces, resulting in 4 points for each fully completed attempt. In the second round, two blue and green types are generated, resulting in 6 possible points. The third and final round is started with a configuration of one blue-type, two green types and one red-type piece resulting in a total of 8 points per attempt, 1 point for a blue type, 2 for greens and 3 for the red piece.

The previous versions of Robot at Factory were assisted with an auxiliary Clock *software* that provides the time left of the given round, the ingoing warehouse configuration with the parts types and outgoing timestamps.

There is still one last component missing to guarantee the correct functionality of the competition and the performance evaluation of the robots: the referee. So far, this aspect has been done manually by an assistant member, and there are several tasks to achieve. Firstly, any time the round is reset, the parts must be manually placed at the Ingoing Warehouse boxes; for each correct loading on the machinery, the correspondent transformed part must be placed from the input box to the correspondent output; the team score must be manually incremented and registered whenever a Part reaches its correct destination; correct management of the clock *software* is needed; and any other correction or unexpected event must be handled, keeping track of all the other tasks. In sum, there is some work to be done by the referee that can be an overwhelming amount of information to be handled and, thus, possibly result in missing out on the correct scoring, for example. With that being said, the number of tasks associated should be minimised, reducing the error.

3.2 Selection of the Fiducial Marker

Firstly, fiducial markers provide the main source of information for the referee system that will be developed. Therefore, their choice should be well studied and considered to ensure a consistent detection rate for any harsh scenario that may be faced.

From the collected markers gathered from the section 2.2, a property strongly considered in the studies to date is the rotation of the marker relative to the detection device. If the marker is considered as the *xOy* plane of the figure 3.4, it is therefore verified normal rotations, in *z (yaw)*, or lateral, in *y (pitch)* or *x (roll)* (identical on a practical view). Depending on the positioning of the camera, certain types of rotations can be more amplified.

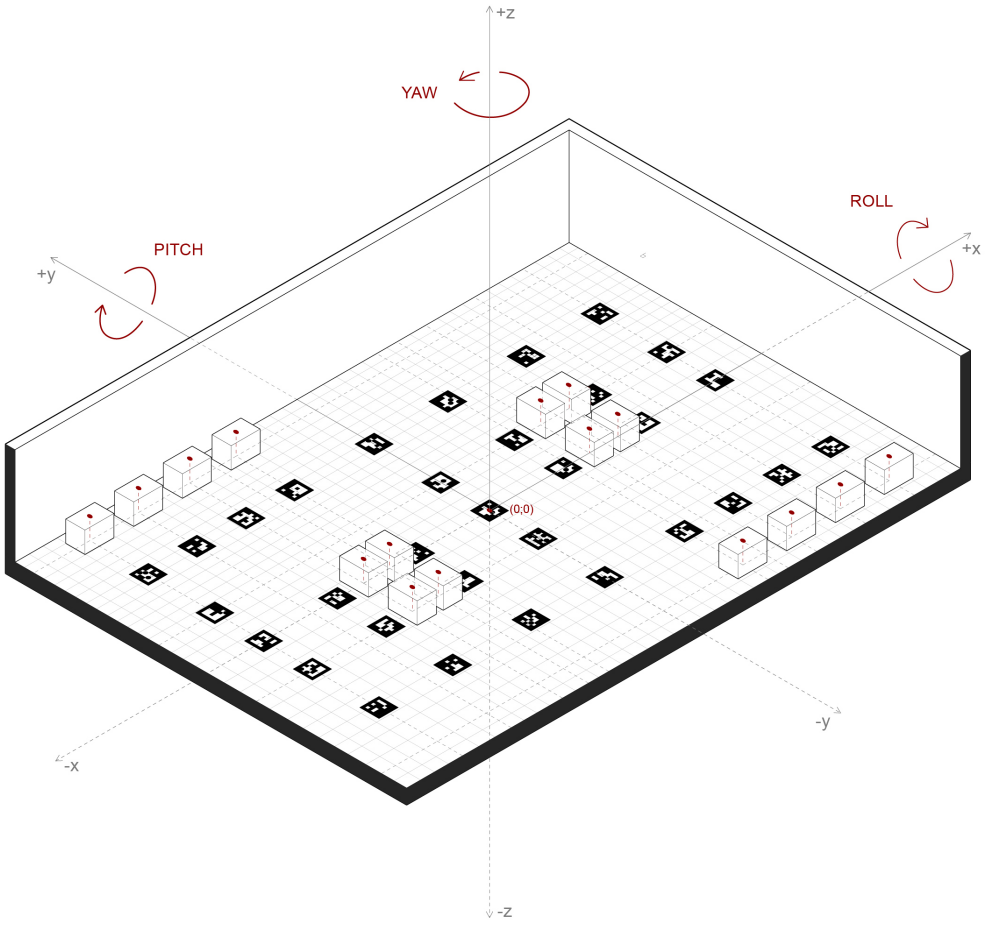


Figure 3.4: Rotations on cartesian axes *x, y* and *z*.

For example, if this device is on the side edge of the competition and close to the ground, not only the *yaw* rotations but also the *pitch* and *roll* rotations will be intensified. In contrast, if the camera is positioned with its optical axis perpendicular to the normal axis of the field, only rotations of the type *yaw* will be witnessed. Therefore, it is important that the choice of the FM is versatile in this respect.

Based on state-of-the-art gathered, a significant portion of the markers presented have an appropriate application to the project case. However, the majority are not available in multiple programming languages. Furthermore, upon detection, additional algorithms should be applied in order to achieve pose detection of the marker. Therefore, the marker type should not restrict the application of these posterior procedures. AprilTag and ArUco Tag have been fiducial markers widely and commonly used among various concepts and in distinct areas throughout the years.

Its popularity has brought advantages that fall on multiple open-source libraries and has been advanced and updated with various algorithms as complementary functionalities.

The AprilTag FM detection process consists first of linear segmentation, in which a method similar to ARTag is applied, differentiating it through segment sequencing in order to form a square and then proceed to its detection. Finally, the internal information is decoded through a system characterised by two parameters: code, or *bits* of internal pattern, and minimum *Hamming distance* between two codes.

On the other hand, according to Ferrão *et al.* [13], the ArUco system consists of methods that perform: an adaptive *threshold* to obtain the necessary contrast to facilitate the subsequent steps to be applied; square detection algorithm, followed by Douglas-Peucker's simplified contour algorithm; marker validation, through defined criteria, such as the quadrilateral sum conjecture, sum of internal angles and geometric composition of the sides; finally, one has the estimation of the position from which *solvePnP* methods are used in *OpenCV*. As tools for implementing the algorithm, libraries of *openCV*, ROS or Alvar can be used.

Additional methods can also be used to improve the localisation accuracy, more precisely edge refinement, corner refinement, extension to chess table corners, bad detection filtering and advanced edge refinement [19].

In addition, computational weight, lighting conditions and digital noise, which will be inevitably present, must be considered. Additionally, regarding the positioning of the camera, several markers may be detected; therefore, computational weight can become a factor to be minimised, highlighting ArUco Tag. Alternatively, the AprilTag marker has also proven helpful with respect to the studies conducted by the authors referenced in the state the art.

On a final note, using ArUco marker also brings the advantage of allowing the simultaneous detection of the pre-existing markers in the field, being beneficial for calibration issues, as will be mentioned later in the chapter 4, thus being the marker of choice.

3.3 Field Mapping

Given the fact that the scoring system of the previous versions of the Robot at Factory competition was a manual task, no measurements were needed in order to assess the correct placement of the pieces as it would be checked visually if their location were inside valid box bounds.

Within the automatic scoring framework, the areas of interest must be mapped beforehand in order to implement the pose verification step. Those areas correspond to the warehouses and machine boxes, and the resulting measurements are shown in the following figure 3.5.

3.4 Intrinsic Calibration

With respect to the intrinsic parameter calibration, Zhang's method, also known as the ''chessboard" calibration, is the most suitable procedure due to its versatility and high accuracy. The chessboard pattern is a seven by ten black and white 1.7625cms squares marker. Hereinafter, the

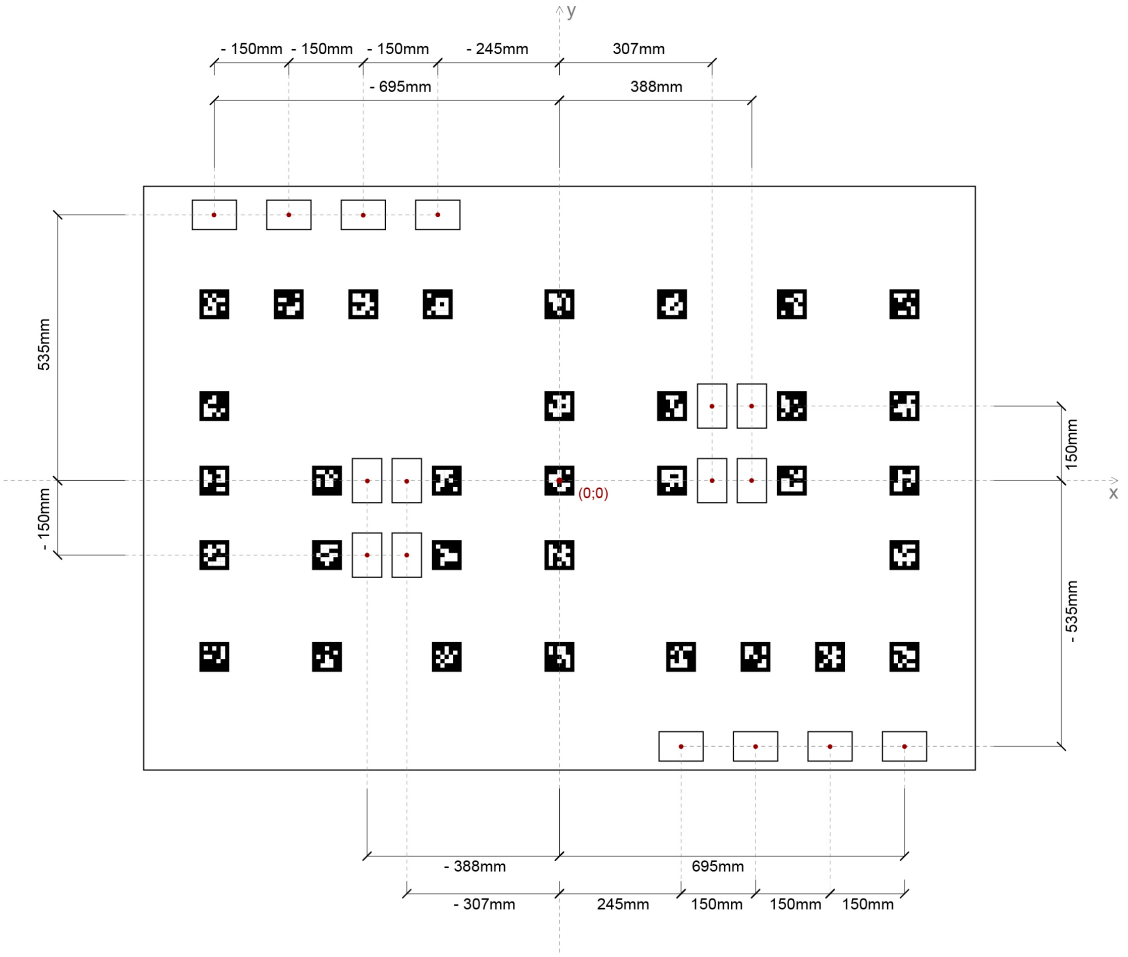


Figure 3.5: Measurements to map all four zones.

.

camera is used to capture images of the marker with various positions and orientations that will be used as input for an *OpenCV* calibration method. As a result, the intrinsic parameters from the matrix (M, equation 2.3) are stored as well as the distortion coefficients derived from the lens imperfections.

$$\boldsymbol{K}_{1x5} = \begin{bmatrix} k1 & k2 & p1 & p2 & k3 \end{bmatrix} \tag{3.1}$$

These variables have a substantial impact on the system since it provides higher accuracy on the results that are obtained on the pixel-to-world conversion.

3.5 Extrinsic Calibration

Once the previous matrices are obtained, the camera parametrisation is ready to be completed by estimating the six degrees of freedom that describe its position through a rotation and translation

along the cartesian axes x, y and z.

For this purpose, Perspective n Point (PnP) is used due to the needed correspondences to correctly apply the algorithm. In other words, an estimate of the rotation and translation matrix of the camera is obtained by taking advantage of known 3-D points and their correspondent coordinates in the image plane. The existing ArUcos on the field are designed to work as landmark points so that the robot can map and guide itself on the field, whereby the same concept can be used by the camera. This approach is also extremely advantageous since the calibration is returned with respect to the field referential frame.

Given any position of the camera, if sufficient markers on the field are detected, then it is possible to compute these values with respect to the competition area referential, correlating the 2-D image points from the camera point of view with the known ArUcos on world coordinates. This algorithm also stands out by assuming a Gaussian error on the gathered points and, therefore, sharper results are expected. Moreover, not only the correlated 2-D and 3-D points and the intrinsic camera matrix are inputted but also the distortion coefficient matrix. This slight detail significantly impacts the succeeding stages since the values obtained are determined and adjusted based on the image distortion, meaning that the *pinhole* camera model remains unaffected.

Additionally, this way grants the ability to dynamically and automatically calculate the extrinsic parameters, providing resilience to camera relocation without re-measurements.

3.6 ArUcos Detection

Concerning the ArUco marker detection, a simple function adapted from an *OpenCV*[1] implementation was developed. Hereby, the ArUco identification (ID) value and the four corners of each marker within the image plane are returned, thus providing each marker's centre point in image pixel coordinates (u_c, v_c). Those resulting outcomes are immediately distinguished from the field to piece ArUcos and then stored with their correspondent pixel values. Finally, these are used as input for the succeeding stage.

3.7 Pose Estimation

The pose estimation procedure has the major task of tracking each piece that is being carried by the robot throughout a full round. However, as seen in the previous figure 3.3, a single piece is described as a simple squared prism without any distinct feature that enables reliable tracking. Hence, unused ArUcos IDs can be used and fixed on top of each part, using the values from the previous step without further modification.

With the *pinhole* formulation, an inverse projection procedure is defined in order to relate an image point (u, v) to its world (x, y, z) coordinates according to the camera parameters, as seen in

[1] https://github.com/GSNCodes/ArUCo-Markers-Pose-Estimation-Generation-Python/blob/main/pose_estimation.py

the equation 2.4. In the context of the developed project, the procedure is defined by the following steps:

1. detect the ArUcos associated with the pieces;
2. obtain the centre point of the ArUco on the image plane (υ_c, ν_c);
3. from the intrinsic parameters (**M**) and the current camera position (extrinsic parameters, **R** and **T**), obtain the 3-D projection of the marker on field $(x\ y,\ z)_p$ knowing that $z_p = 65mm$.

Since the ArUcos are fixed on top of the parts, and these are only displaced on the xOy plane, we have a constant z coordinate throughout the whole piece movement, hence the a priori awareness of the piece altitude that does not need to be computed.

Following this, we can reformulate the projection equation so it meets the system's needs, which results in the following expression:

$$Z \begin{bmatrix} \mu_c \\ \nu_c \\ 1 \end{bmatrix} = \boldsymbol{M}_{3x3} \begin{bmatrix} \boldsymbol{R}_{3x3} & \boldsymbol{T}_{3x1} \end{bmatrix}_{3x4} \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} \tag{3.2}$$

$$\boldsymbol{M}^{-1} Z \begin{bmatrix} \mu_c \\ \nu_c \\ 1 \end{bmatrix} = \boldsymbol{R}_{3x3} \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} + \boldsymbol{T}_{3x1} \tag{3.3}$$

$$\boldsymbol{R}^{-1}\boldsymbol{M}^{-1} Z \begin{bmatrix} \mu_c \\ \nu_c \\ 1 \end{bmatrix} = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} + \boldsymbol{R}^{-1} T \tag{3.4}$$

From equation 3.4 we have a model that translates an image plane into world coordinates from the parameters gathered in the previous steps. However, there is one last parameter to be determined: the focal plane's constant value, Z. Since z_p is constant, the last row of our equation is fully defined by solving Z, resulting 3.6.

$$\boldsymbol{R}_3^{-1}\boldsymbol{M}_3^{-1} Z = z_p + \boldsymbol{R}_3^{-1}\boldsymbol{T}_3 \tag{3.5}$$

$$Z = \frac{z_p + \boldsymbol{R}_3^{-1}\boldsymbol{T}_3}{\boldsymbol{R}_3^{-1}\boldsymbol{M}_3^{-1}} \tag{3.6}$$

Finally, the piece pose projection and estimation can be computed and verified in the following procedure.

3.8 Pose Verification

Once all the previous information is gathered, the last task is responsible for rewarding the competitors whenever the piece is correctly placed in a specific field location.

For this purpose, the referee algorithm will be responsible for verifying and validating the piece position concerning the current state and type, hence the requirement of having the complete information from all the pieces available and the areas of interest localisation. Additionally, the overall system will be deployed within multiple devices working in parallel with distinct functionalities. Thus some communication procedure must be added alongside each device so that the key data is acquired at this point.

3.9 Competition Modifications

It is worth mentioning that the first constraint noticed once the system had been deployed was the lack of contrast between the ArUco marker and its surroundings. Since the piece is black, some changes had to be made to have a functional detection, hence the use of white paper on top of each piece. Furthermore, once the piece is positioned on a box, the back wall that defines the perimeter is a dark tone of green, which also hinders the detection algorithm. Therefore, the white paper was also added to the walls, granting easier marker detection. In the end, the tested system with a moving robot carrying a piece also showed trouble since there was not enough contrast, which would benefit from adding lighter tones to the robot. In the following figure 3.6 are displayed the previous statements.

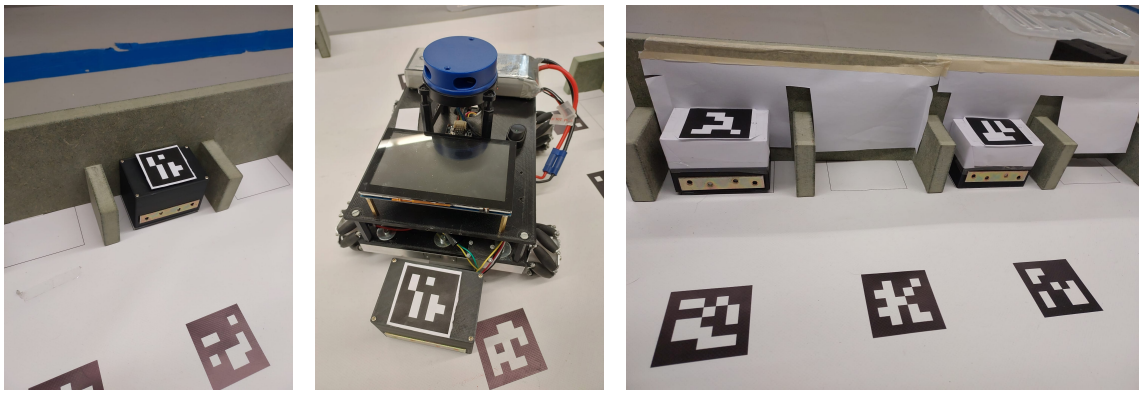


Figure 3.6: From left to right: Low contrast between piece, marker and wall; low contrast between piece, marker and robot; white paper addition to the piece and wall for higher contrast.

Chapter 4

Implementation

This chapter refers to the materials, software, programming languages and methods used to achieve the correct functionality of the procedures involved.

Table 4.1: Chapter 4 nomenclature.

Variable	Designation
N_PIECES	Total number of pieces used
pIDini	ID of the first piece
M	Intrinsic Camera Matrix
K	Distortion Coefficients Matrix
m	Number of field markers detected on a frame
n	Minimum number of points for PnP
ε	Position Error
α	Smoothing Rate
$max_{\varepsilon H}$	Maximum error for moving piece
$max_{\varepsilon L}$	Maximum error for stationary piece

4.1 Global Architecture

Besides the markers on the ground, the competition will have a tag on every piece to be transported, granting us the ability to receive helpful information about these objects, such as position, orientation, identifier and type. This information is being decoded by the previously calibrated camera, positioned at **(x, y, z)**, and will send the corresponded information to the referee system via ZeroMQ (ZMQ). All camera procedures are done by a Raspberry Pi v.4 8GB, providing the ability to reallocate the camera position with ease, self-processing capabilities and inter-communication with other applications, rather than larger equipment which would hinder the overall system.. Camera procedures are done with *OpenCV* python libraries with Thorny IDE as editor. On the other hand, both the referee and clock *software* were done via Lazarus IDE, pascal, which provides the ability to develop the procedures and the visual interface, having the advantage of integrating testing and simulation features.

4.2 Piece Detection

The tags to be fixed on top of each piece are also going to be ArUco-type markers, consequently enabling simultaneous detection of the field markers, which is helpful for the succeeding stages.

Currently, IDs ranging on the [0, 26], [30, 35] and [50, 65] intervals are already in use. Thus the IDs on the interval [40, 47] are dedicatedly used for piece detection. The user defines the number of pieces if more than four parts are used per attempt. From this scenario and for illustration purposes, eight pieces are considered to be involved in the competition despite only four being used, which would define the system with the following: total number of pieces: $N_PIECES = 8$; ID of the first piece: $pIDini = 40$. All the markers used are generated 5x5 from [1].

Hereafter, the camera should be positioned according to one primary requirement: the ArUco marker must be detected throughout all the eight areas of interest (input machine-A/-B and outgoing warehouse). Additionally, the camera was fixed to a tripod with intermediate 3-D printed parts to link the RPI, as illustrated in the figure 4.1, to provide stability and avoid error increases.



Figure 4.1: Raspberry Pi and camera supporting structure.

Since the marker detection algorithm, calibration method and pose estimation process are done within one single device, adequate processing capabilities are desired; thus, the Raspberry Pi v.4

[1] https://chev.me/arucogen/

8GB RAM is used to run the system without any trouble. For the video capture, a Raspberry Pi Camera Module V.2 is used with a resolution of 640 x 480 for a higher frame rate.

Algorithm-wise, firstly the intrinsic parameters are obtained from the images with the chessboard marker, shown in the figure 4.2, with the use of the following *OpenCV* functions (listing 4.1).

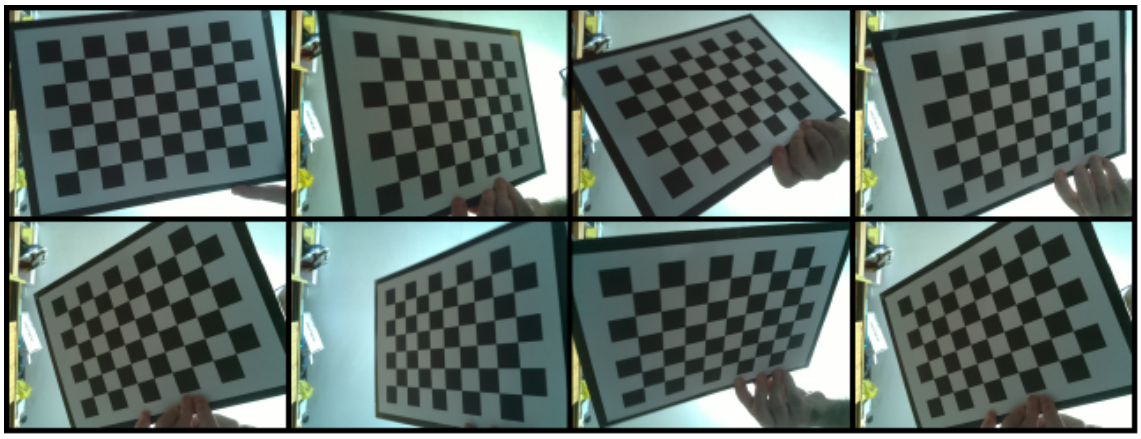


Figure 4.2: Chessboard marker pictures from the Raspberry Pi Camera Module v2 to perform the intrinsic calibration.

```
def calibrate(dirpath, square_size, width=9, height=6, visualize=False):
    define chessboard dimensions: chessb
    Initialize 3DobjectArray
    Initialize 2DimageArray

    get chessboard images
    for each image:
        Find the chessboard corners: cv2::findChessboardCorners

        if corners were found:
            find the sub-pixel location: cv2::cornerSubPix

            add chessb to 3DobjectArray
            add sub-pixel locattion to 2DimageArray
    calibrate camera: cv2::calibrateCamera
    return intrinsic matrix; distortion coefficients
```

Listing 4.1: Intrinsic camera calibration pseudo-code

The procedure returned the following values regarding the intrinsic matrix (equation 4.1) and the distortion coefficients (equation 4.2).

$$\boldsymbol{M} = \begin{bmatrix} 622.84188937 & 0 & 326.88427668 \\ 0 & 622.55192985 & 216.18039264 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

$$\boldsymbol{K} = \begin{bmatrix} 0.20892773 & -0.6219022 & -0.0131884 & 0.00922845 & -0.62808971 \end{bmatrix} \tag{4.2}$$

Secondly, all the ArUco markers on the field are detected with cv2::aruco.detectMarkers and filtered from field markers and piece markers. The data gathered from the first set is used for extrinsic calibration, resulting in an array of size **m** linking each known 3-D world coordinate with the current frame 2-D image coordinates, being **m** the number of field markers detected on a frame. The second set stores the ID and 2-D image coordinates of the centre of the seen pieces on the frame to obtain its location on the field. Note that the 3-D to 2-D projection maps a point to the image plane, but the world position is unknown at this stage.

Hereafter, the resulting 3Dobject and 2Dimage array of points are used as input, as well as the camera intrinsic matrix and the distortion coefficients calculated beforehand, for the Perspective n Point algorithm (cv2::solvePnP), giving rise to the rotation and translation matrix that describes the extrinsic parameters.

The estimate of the object's position is done with the ideal *pinhole* camera model equations (2.4) defined previously in the chapter 2. Note that the distortion coefficients entail an improper use of the model described, but the fact that PnP corrects the rotational and transitional values concerning those imperfections described by **K** grants back the opportunity to use the standard *pinhole* model.

From this point, all the input data to perform pose estimation has been acquired, meaning that this procedure is ready to be used. However, this procedure is done if, and only if, **m** is greater than **n**, being **n** the minimum number of points needed to compute the position assessment and **m** the total number of correlated points detected. In other words, the piece pose estimation procedure is only done if a minimum number of correlated points are detected.

Perspective-n-Point has a minimum requirement of three (non-co-linear) points to perform the algorithm correctly. However, once the camera is within range of the ArUco marker detection, it is relatively easy to detect at least three points but at risk of being all at the same trajectory (co-linear). Therefore, the minimum **n** is reconsidered and adapted to the field setup. In the worst case possible, we consider the detection of a single complete row or column detection, which would be respectively 8 and 5 ArUcos (as seen on the right figure 3.2), concluding that the most undesirable event is shown on the x axis. Note that the camera would have to be positioned so that only a single row would be seen on the frame, which is unusable. Therefore, 8 becomes the minimum number of points needed to perform the pose estimation algorithm: $\mathbf{n} = 8$.

After calculating its position, an open-source asynchronous messaging protocol library, ZeroMQ (ZMQ), is used to update the referee *software*, finishing a complete cycle concerning the

Rasberry Pi system.

4.3 Clock System

As stated previously, the clock system is responsible for generating the Ingoing Warehouse configuration, recording arrival timestamps at the Outgoing Warehouse and displaying the time left of a given round. Moreover, this system does not have any data from the pieces when they leave the Input Warehouse. Thus some adaptations have been made to store and follow the piece's information.

It would be appropriate to include the piece's data so that every member taking part in this challenge has a complete understanding and knowledge of the round's current state. Therefore, a grid was added to provide information about each piece's current type and state, as shown in figure 4.3. The type can be either "Unassigned", "Red", "Green", "Blue' or "White", meaning that it has been reset, is raw, intermediate, final or finished type, respectively. This information is displayed in the second row of the grid by the designation of "Colour". After this, on the third row, the state is defined as "Stored", "OnField", "OnProcess" and "Done" in the event of a piece undetected, detected, successfully placed on machinery and correctly placed on the final destination (Outgoing Warehouse).

From now on, the data is used by the referee system so that the pose validation can be done alongside the pose estimation that the Raspberry Pi does. For this purpose, bi-directional communication between the Clock and the Referee is needed, hence the use of UDP to:

- Send initial states from Clock to Referee;
- Update the new states of the pieces from Referee to Clock when detected or correctly placed.

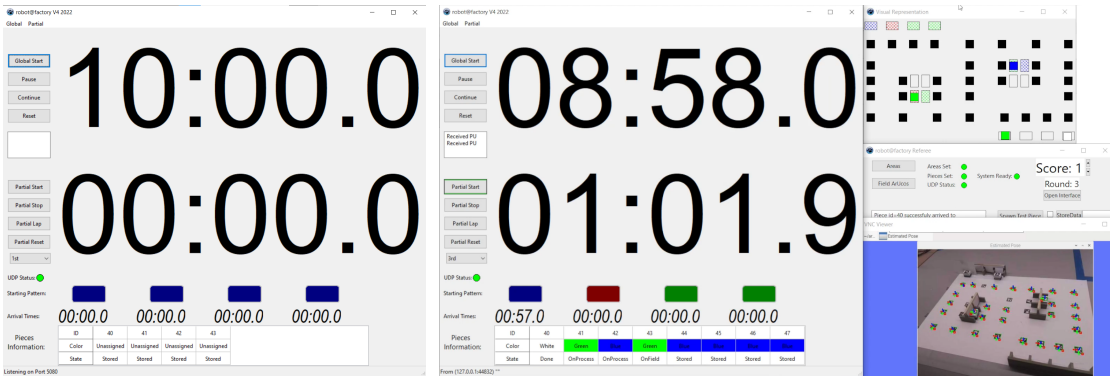


Figure 4.3: Left: Clock interface with the implemented grid at the bottom; Right: State updates according to the current on-field scenario.

The referee interface "Ingoing Warehouse" representation is updated according to the Clock initialisation, and once a piece reaches the final "White" colour (finished state), the arrival timestamp is also registered and shown, as seen on the piece with an ID equal to 40 in the right image of the figure 4.3.

4.4 Referee System

The goal is to validate the pieces' positions to detect the correct placement and reward points. This procedure has two pre-conditions to be validated: setting the areas of interest by reading an XML file containing all the previously measured areas and receiving the piece's details generated from the Clock program. The fulfilment of these conditions enables the verification of the piece's position sent by the Raspberry Pi via ZeroMQ, whose frequency is conditioned by the video frame rate (frames per second). The piece detection and localisation data are sampled and received without further processing, which might evaluate a given position with low reliability and precision.

Therefore, the pieces' positions are stored and updated according to the current and previous location, with the intent to have a comparison measure that provides the system with an error that translates the difference between the present and last position concerning x (ε_x) and y (ε_y). Furthermore, ε is obtained as an average value of the previous (ε_{prev}) and current (ε_{curr}) iteration. This way, any oscillation from the continuous pose estimation can be reduced and used as input for the position smoothing.

The position smoothing procedure identifies three piece cases: stationary, moving and invalid. Thus, a first-order Infinite Impulse Response (IIR) filter for an $\alpha \in [0,1]$ is applied in order to minimise oscillation and showcase a fluid motion when accelerating, designed as:

$$new_{pos} = curr_{pos}\alpha + prev_{pos}(1-\alpha) \tag{4.3}$$

That said, a threshold must be fixed to distinguish the invalid value. According to the previous position, these represent values that describe a displacement that would not be likely to happen and hence are preferably rejected. On behalf of the camera documentation for the frames per second (FPS) with a resolution of 640 x 480 [2] and assuming at worst 5 FPS for the pose estimation procedure gives us intervals of 200 ms from subsequent frames. Additionally, taking that the robot moves at a maximum of 200 mm/s means that a possible 40 mm might bed displaced from one frame to another, which bounds the maximum error from one iteration to another ($max_{\varepsilon H}$) that defines a limit for valid and invalid values. Assuming that the error from the continuous pose estimation of a stationary piece is lower than 10 mm ($max_{\varepsilon L}$), we have defined the boundaries for all three cases of position smoothing

Furthermore, by analysis of equation 4.3 we know that a higher α benefits the current position, meaning that if the previous position is further away, a more significant displacement occurs, and consequently a lesser smoothing will be noticed on its updated value and vice-versa. Therefore,

[2] https://www.raspberrypi.com/documentation/accessories/camera.html

two rates are considered: an α_L that denotes a lower smoothing rate and another α_H desired for a higher smoothing rate, concluding that α_L is used for stationary pieces and α_H for moving ones. These values were defined manually according to the best performance obtained. Hence $\alpha_L = 15\%$ and $\alpha_H = 70\%$.

As a result, the following procedure is implemented, as showcased in the figure 4.4.

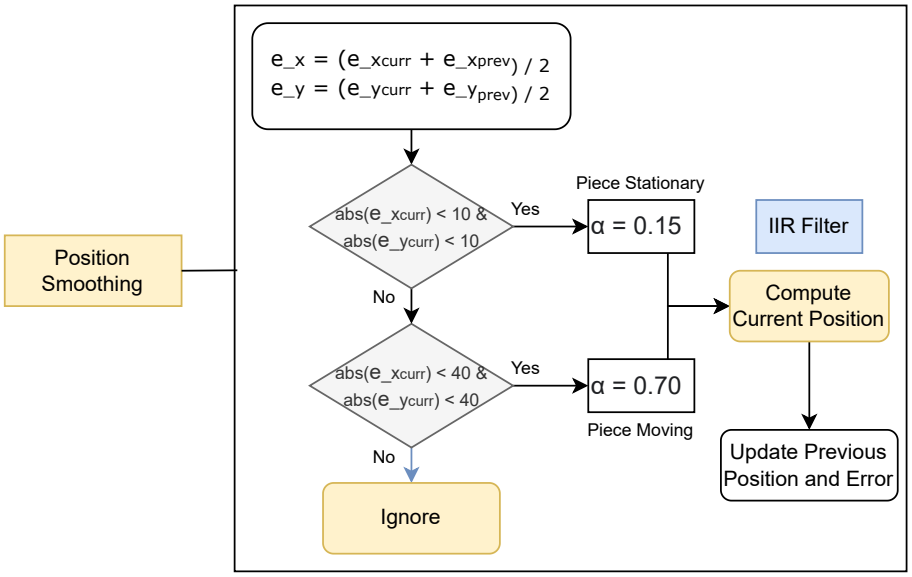


Figure 4.4: Position smoothing and Infinite Impulse Response filter algorithm .

The last procedure needed to complete the system is the pose verification according to its current type and location. For this purpose, each detected piece on the field is verified according to its centre position. If it is within the bounds of a box that is a possible correct outcome, then the piece is updated, and the score is increased, as shown in figure 4.5. This update is also sent towards Clock via UDP. Note that the robot also communicates via UDP with the source of data (server) to be aware of the current configuration of an attempt ("IWP").

Having stated the functionality and procedures of each component that takes part in the overall system, the following diagram (figure 4.6 describes the global architecture with the steps illustrated from a macro level point of view.

Finally, we developed a system that covers all the aspects present at Robot at Factory while providing the maximum information possible from an attempt, including the communication not only for the piece setting and updating data as well as the robot server requests.

4.5 Interface

The information gathered and computed is showcased through a visual interface which concerns the current score, round, events and the piece's positions, as illustrated in figure 4.7. This representation is available by pressing the "Open Interface" button, then "Areas" and finally "Field ArUcos" so that the whole map is displayed. It is worth mentioning that the last two buttons are responsible

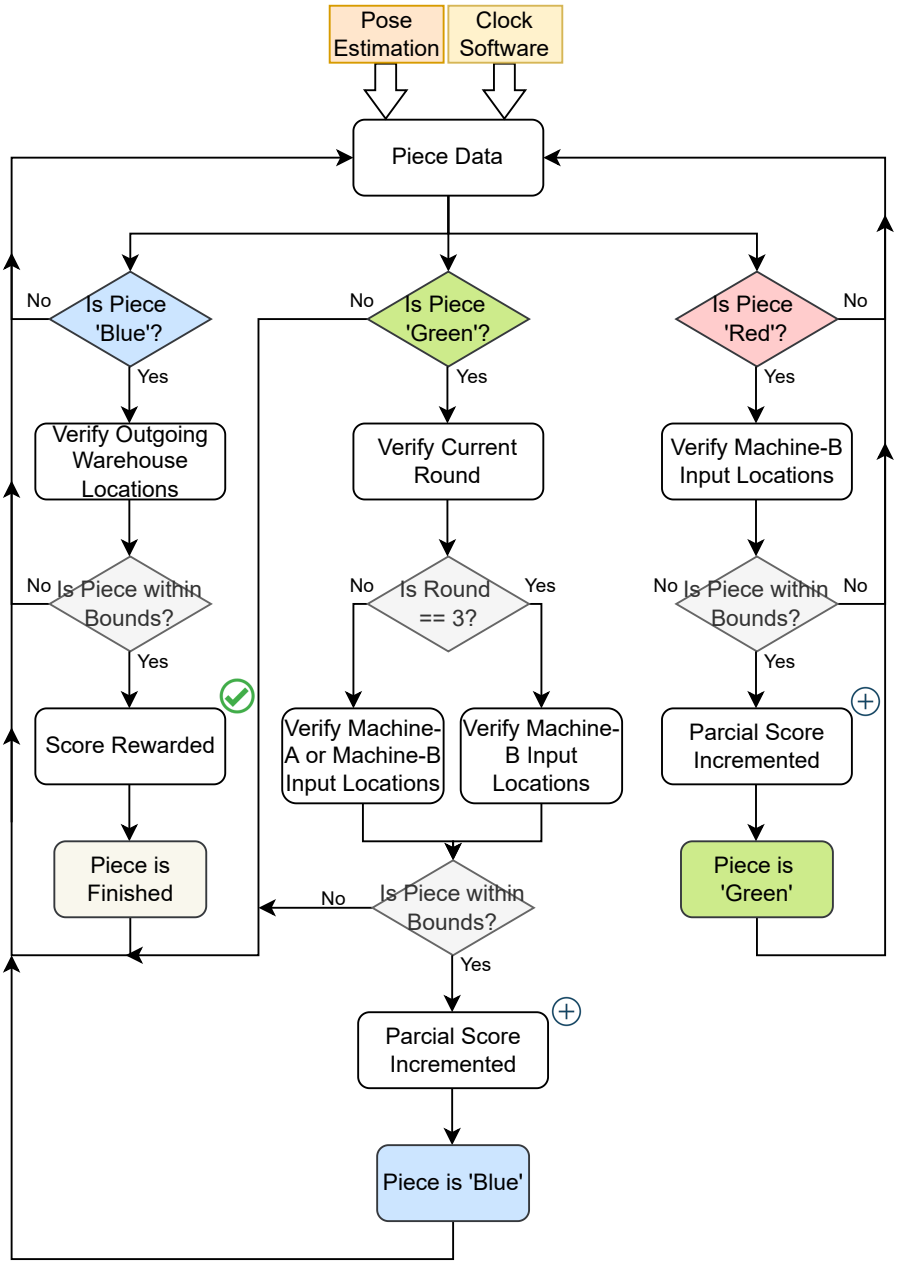


Figure 4.5: Flowchart of the referee algorithm.

for reading the XML files containing the previously measured and mapped positions of the areas and the ArUcos with the intent to store the ground truth mapping data into the referee *software*. Of these two, only the first mentioned is mandatory, considering these values are later used for the pose verification procedure, consequently validating a given piece position, constrained by the round and affecting the score. Additionally, this button also provides the minimum and maximum values concerning the x and y axis, which subsequently is used to replicate an on-scale version of the field through normalisation with the current interface window size.

Throughout an attempt, once a piece is successfully placed into a correct destination, the piece

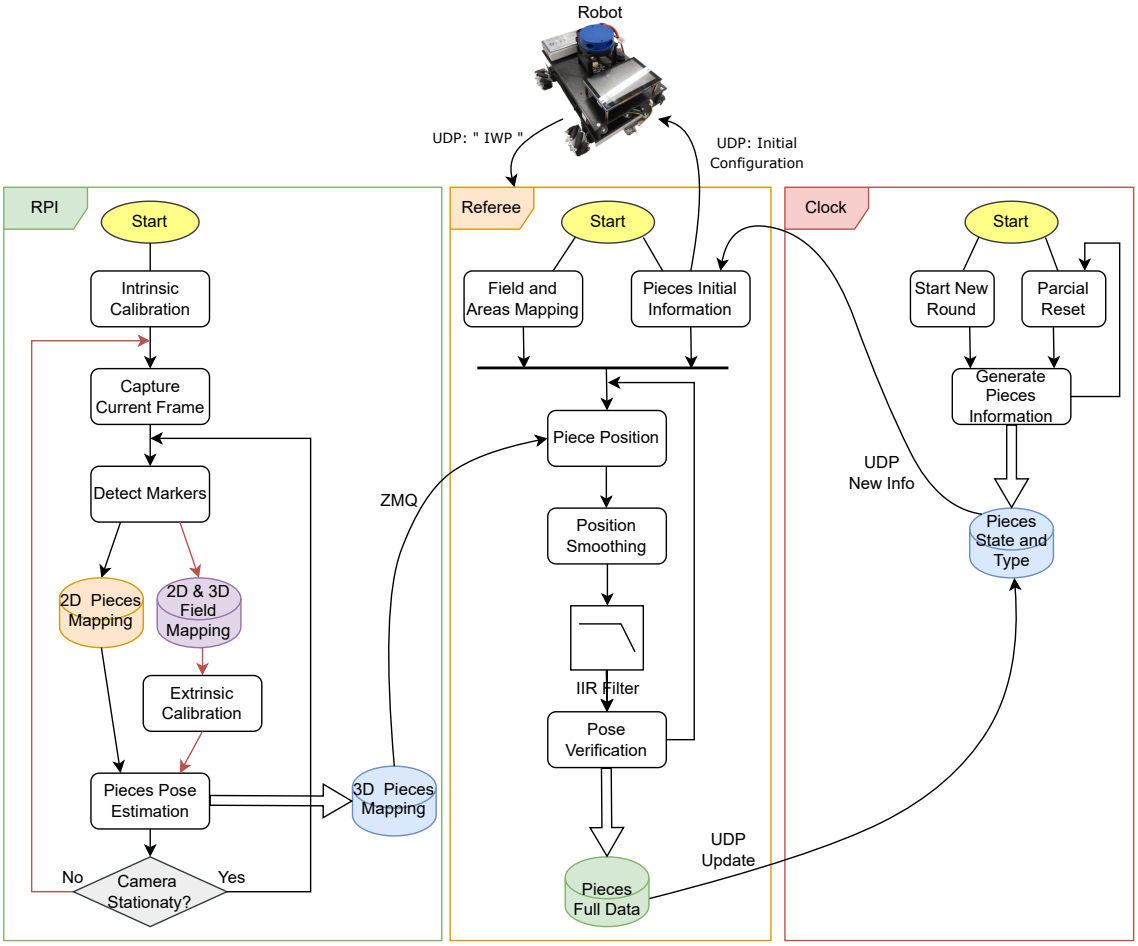


Figure 4.6: Flowchart of the global architecture interactions.

is visually updated. For instance, if a green piece is dropped into Machine B input, it is updated for the succeeding type, blue. Correctly unloading parts to the input machinery section also updates the colour of the input box to the piece entrance colour, and the correspondent output box is updated to the transformed colour, as shown in the figure 4.8.

Moreover, the interface developed was also designed to provide information about the crucial requirements that need to be met to be ready to operate: areas mapped, pieces of information received, and a functional UDP communication. These are all illustrated by a red or green circle indicator, at the "robot@factory Referee" window, and once all three components are set, the system is ready, and the correspondent indicator also turns green ("System Ready").

4.6 Simulation and Scaled Model

Before collecting and analysing the data results from a fully functioning multi-dependable system, advanced testing represented a major key factor in providing enough results to conclude that the system satisfies the requisites and functionalities from a micro-level point of view, as minor

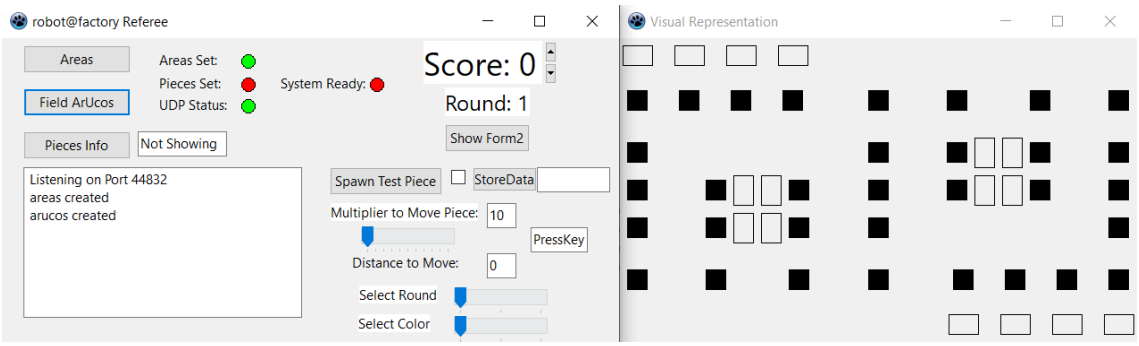


Figure 4.7: Interface of the referee system: user interactions on the left and virtual field information on the right.

procedures from each component, to a macro-level system that has interchangeable information. Furthermore, the competitive environment has not easy accessibility and its sett up and relocation are time-consuming, which hinders the testing and partial validation procedure.

Therefore, the referee interface also had the advantage of developing and implementing a simulation environment which later eases testing scenarios throughout the project's development, as described after this. With that said, additional functionalities were added, allowing simulation capabilities to maximise the testing scenarios with and without the other components.

The simulation side is represented with all the bottom items on the "robot@factory Referee" window and consists of: a text box that provides information on triggered events; spawning of a test piece with the desired colour (type); bindings to replicate piece movement (with multiplier and "PressKey"); selection of the round; and a data button to store pieces data for further analysis, highlighted on figure 4.8. The last button stores all the relevant information on a text file with the latest data updated and the logs occurred (figure4.9).

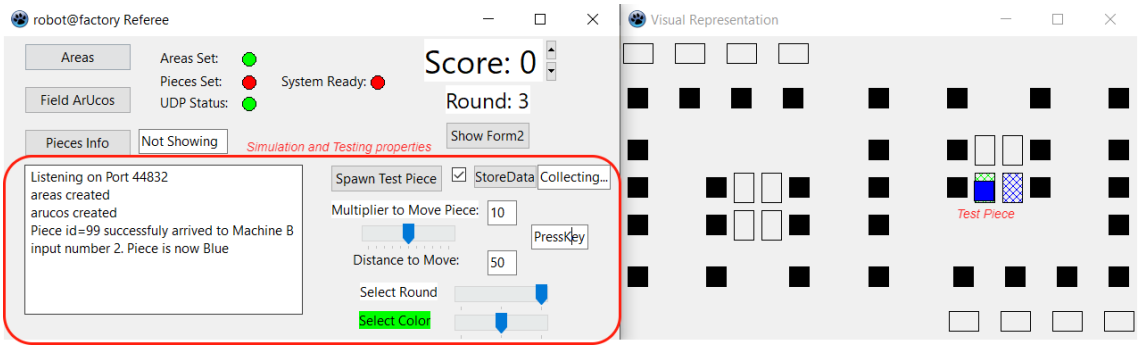


Figure 4.8: Referee simulation and testing of a correctly placed green piece.

Besides simulation and with the intent to globally test the system, an A3 scaled size field was printed that fully replicates the competition and overcomes the accessibility constraints. The following figure 4.10 presents the correspondent scaled field with a tapped ArUco marker with an ID equal to 40.

all.txt - Bloco de notas

Ficheiro Editar Formatar Ver Ajuda

```
22:13:47
40   Stored 0,00 0,00
41   Stored 0,00 0,00
42   Stored 0,00 0,00
43   Stored 0,00 0,00
44   Stored 0,00 0,00
45   Stored 0,00 0,00
46   Stored 0,00 0,00
47   Stored 0,00 0,00
99 Blue OnProcess 295,00 -5,00
```

alloccurences.txt - Bloco de notas

Ficheiro Editar Formatar Ver Ajuda

```
22:13:44 47   Stored 0,00 0,00
22:13:44 99 Green OnField 235,00 355,00
22:13:45 40   Stored 0,00 0,00
22:13:45 41   Stored 0,00 0,00
22:13:45 42   Stored 0,00 0,00
22:13:45 43   Stored 0,00 0,00
22:13:45 44   Stored 0,00 0,00
22:13:45 45   Stored 0,00 0,00
22:13:45 46   Stored 0,00 0,00
22:13:45 47   Stored 0,00 0,00
22:13:45 99 Green OnField 235,00 175,00
22:13:46 40   Stored 0,00 0,00
22:13:46 41   Stored 0,00 0,00
22:13:46 42   Stored 0,00 0,00
22:13:46 43   Stored 0,00 0,00
22:13:46 44   Stored 0,00 0,00
22:13:46 45   Stored 0,00 0,00
22:13:46 46   Stored 0,00 0,00
22:13:46 47   Stored 0,00 0,00
22:13:46 99 Green OnField 235,00 -5,00
22:13:47 Piece id=99 successfuly arrived to Machine b input number 2. Piece is now Blue
22:13:47 40   Stored 0,00 0,00
22:13:47 41   Stored 0,00 0,00
22:13:47 42   Stored 0,00 0,00
22:13:47 43   Stored 0,00 0,00
22:13:47 44   Stored 0,00 0,00
22:13:47 45   Stored 0,00 0,00
22:13:47 46   Stored 0,00 0,00
22:13:47 47   Stored 0,00 0,00
22:13:47 99 Blue OnProcess 295,00 -5,00
22:13:48 40   Stored 0,00 0,00
```

Figure 4.9: Pieces information through text storage.

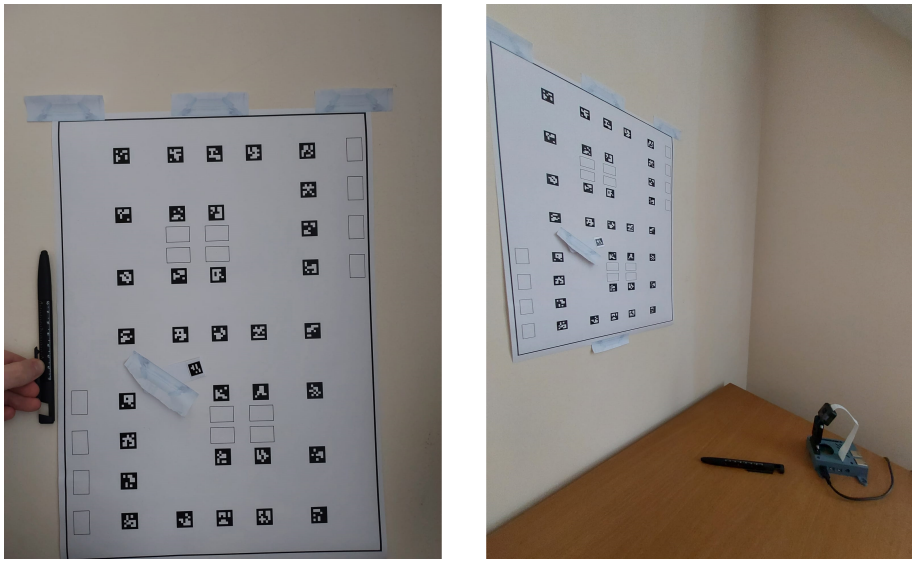


Figure 4.10: A3 scaled field-testing version with camera setup and an ArUco marker of ID 40.

From the example shown in the figure 4.11 and with the information generated from the Clock *software*, is it known that piece number 40 is of the "Red" type. Once detected by the camera, its state is updated to "OnField", and the piece is represented with its current estimated position on the visual representation window from the referee interface.

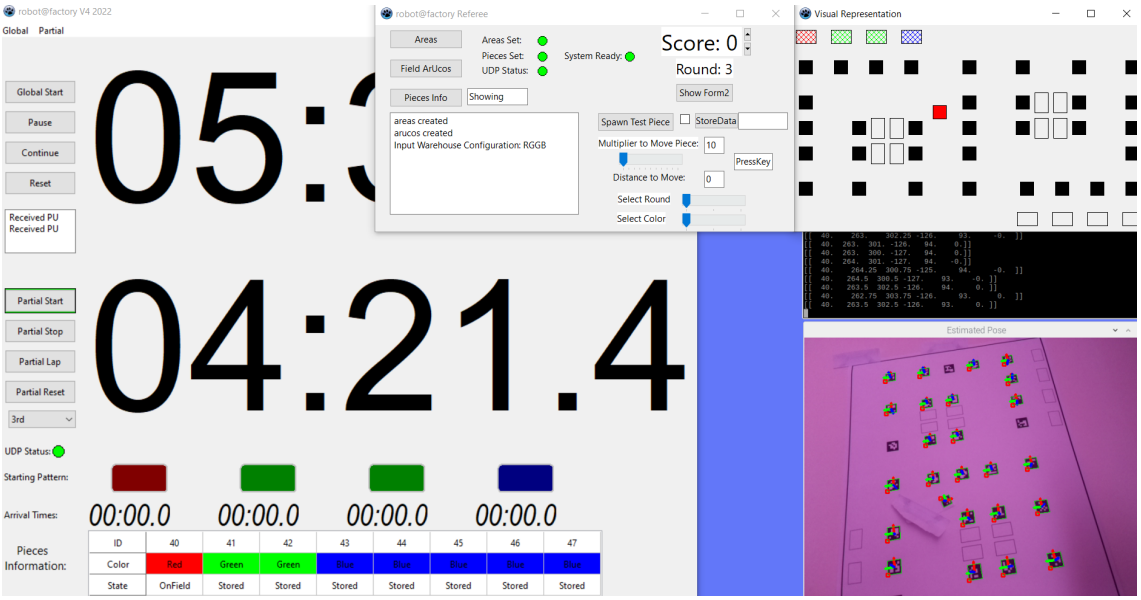


Figure 4.11: Referee, Clock, and RPI camera concurrently execution and interaction with the detection of a red type ArUco piece, ID 40.

4.7 Fault Tolerance

Despite having a fully functional system, some awareness must be taken into consideration due to the imperfections embedded in all procedures, whether illumination changes, noise, miss-calculations, errors in communications, etc. Consequently, a more significant impact might occur on the end-factor of the global architecture algorithm, which is the pose verification. From here, two undesired scenarios may result: having a system rewarding an incorrectly placed piece or having it unaware that the piece has properly reached its destination, thus jeopardising the final score of a team.

Therefore, two mechanisms were implemented with the intent to bypass faulty events that result in a miss-match from the actual competition scenario to the virtual correspondent replication. These are accessible from the referee interface, one on the user interaction window and another on the visual representation. The first one is a manual score incrementor and decrementer that is presented by an upwards and downwards arrow on the right side of the "`Score:_`" label. This functionality enables the score readjustment with ease and does not require detecting the piece on the field, which provides a reliable correction. Secondly, user interaction was added within the visual representation window, which is done by left-/right-clicking a detected piece shown on the interface. By doing so, the piece has its type positively/negatively updated to maintain the correct overall state of the competition. Additionally, if this interaction involves a "White" type transition, the score is also affected by adding/removing points. The procedure used for this purpose is a mouse down click event and was implemented as followed at listing 4.2.

```
procedure PieceOnMouseDown(Sender: Object; Button: MouseButton);
begin
  if MouseButton is LeftClick then begin
   Object.ID is TagID
   if TagID is within PiecesIDs then begin
      Piece[TagID].Type is Updated to the Next Type;
      if  Piece[TagID].Type is Last Type then Increment Score;
      UpdateClock is True;
    end;
  end else if MouseButton is RightClick then begin
   Object.ID is TagID
   if TagID is within PiecesIDs then begin
      if  Piece[TagID].Type is Last Type then Decrement Score;
      Piece[TagID].Type is Updated to the Previous Type;
      UpdateClock is True;
    end;
  end;
  //Piece Type Order: Raw; Intermediate; Final; Finished.
end;
```

Listing 4.2: Piece update Pseudo Code procedure for faulty events

Chapter 5

Testing and Results

In this chapter, test scenarios will be presented to record the system's performance according to the error coming from the position estimation algorithm, along with the validation of the system during a simulated case and over the three existing rounds of the competition.

5.1 Raw Type Scenario Through Simulation

To validate the referee functionality, we will consider the most challenging scenario from the competition, which is the correct transportation of a raw type piece (red). For this purpose, the piece should be placed on one of the inputs of Machine A, then Machine B and lastly at the output, Outgoing Warehouse. If a destination is skipped and the order is not followed no points are awarded since the piece does not reach its final type. Finally, and upon the correct approach, that piece should be valued as three, having the score incremented by the same value.

Therefore, from the simulation point of view, a red typed test piece is spawned on a supposed round 3 attempt. The piece is displaced through the field (movement illustrated at the figure 5.1) to check if its type changes upon correct placement and is not rewarded if it is wrongly placed. Thus, we moved the piece to Machine B input incorrectly and then Outgoing Warehouse instead of Machine A. Once the part is located on the input box 2 of the Machine A section, its type is changed to intermediate (green). Hereafter, the next destination is the input of Machine B. However, we head incorrectly to the Outgoing Warehouse locations and conclude that no update was mistakenly made. So, it is then moved to Machine B number 2 input, and the final type attribute is assigned. The piece is finally placed at the Outgoing Warehouse, which results in a reward of 3 points.

Additionally, the piece path and final destination are shown in figure5.2 where the triggered events shown on the message box should be noted, as well as the last piece type and the score given according to that attempt.

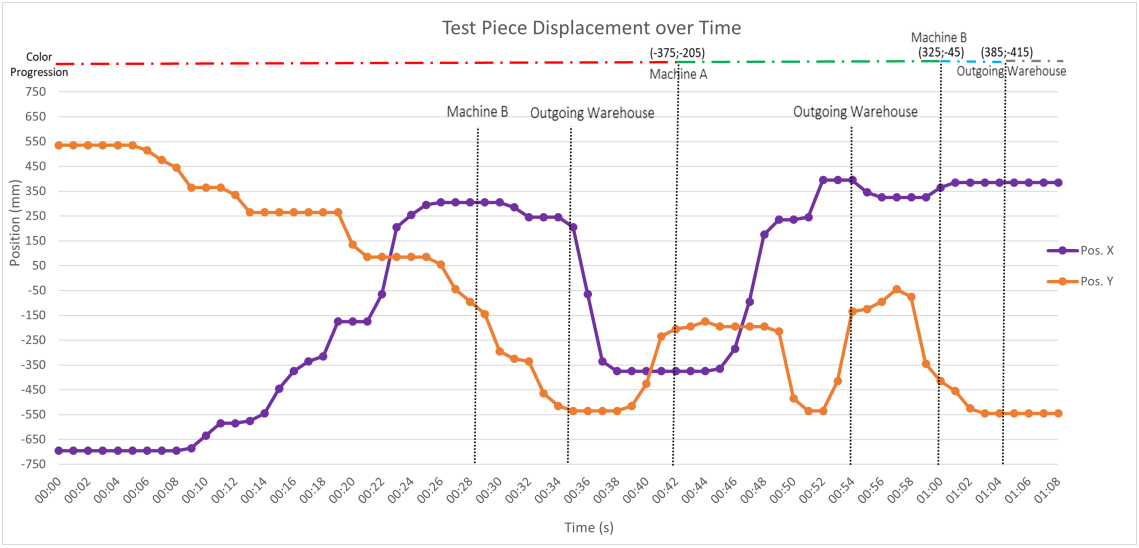


Figure 5.1: Test piece simulated displacement over time on x and y axis.

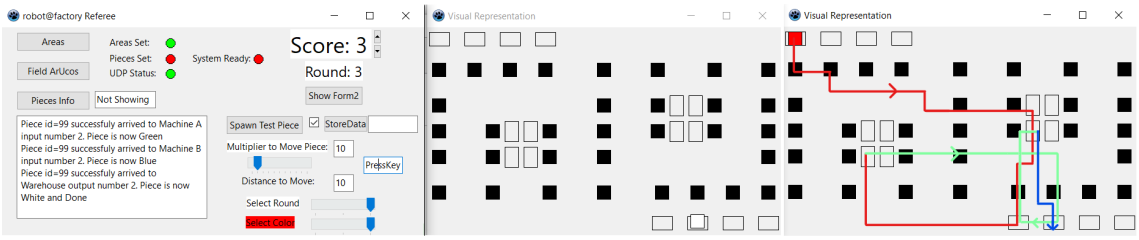


Figure 5.2: The final state of the interface is on the middle and the piece path on the right.

5.2 Pose Estimation Error

Having all the components ready, we must set up every element to have a complete functioning system. Consequently, we consider the referee and clock system, which can be executed on the same or distinct computers, with the ArUco detection and pose estimation known to be running on the Raspberry Pi 4. However, the camera placement on the field is still missing, thus being the main focus of the studies conducted and the results analysed in this section.

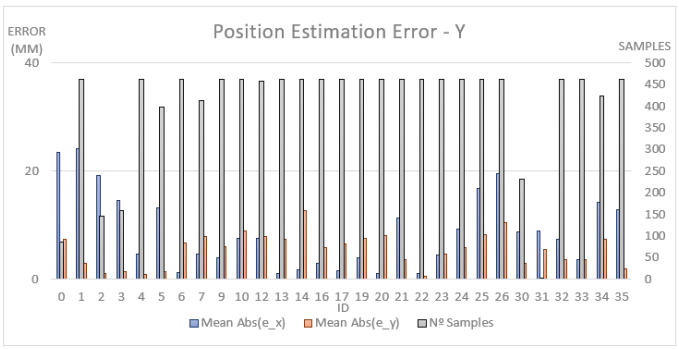
The primary requirement of the camera setup is that every piece should be detectable when placed into any of the boxes representing the machine's input and outgoing warehouse within one frame. After this, the following essential aspect that should be noted is the accuracy of the pose estimation algorithm. In this scenario, the camera is an input device subject to noise due to imperfections, such as lens distortion coefficients, Gaussian noise on the image and world-to-pixel constraints associated with the frame resolution, etc. Consequently, those imperfections are then carried on through the pose estimation procedure, affecting the localisation precision step. In sum, our goal is to conduct a series of experiments to determine the best camera positioning that ensures the previous requisite of complete detection and minimum position estimation error.

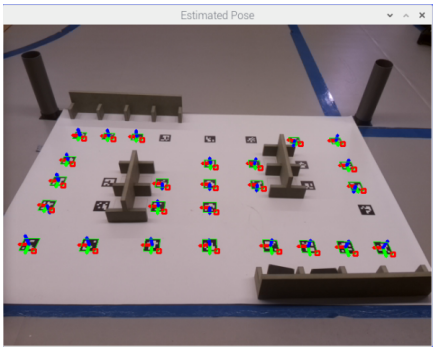
For this purpose, we save all the captures that occurred within one minute for all the pose estimation testing environments for a given camera setup.

5.2.1 Field Tests

Firstly, the camera is placed on the field with distinct positions and rotations with the intent to detect and estimate the position of the field ArUcos. Additionally, this procedure does not need previous marker measurements, providing a fast setup and data gathering system with the respective error in x and y. Note that z estimation is not considered since this value is always known ($z = 0$ for field ArUcos). With that being said, results were acquired according to four camera positions that captured the whole field:

- **Case** Y: x = 0 mm; y = 1200 mm; z = 1100 mm. Camera oriented towards the field origin (laterally, figure 5.3). With respect to an error on the x axis, it is noticeable that the worst results are shown on points whose locations are the furthest from the origin and optical camera centre. Additionally, the impact of the relative position of the marker on the camera appears to be minor. On the other hand, the best results are seen on the points closer to the 0 on the x axis. In contrast, the error on the y axis is higher on ArUcos inside of the field and on the farthest points relative to the camera and optic centre. In this case, the points closer to the camera represent the best results.





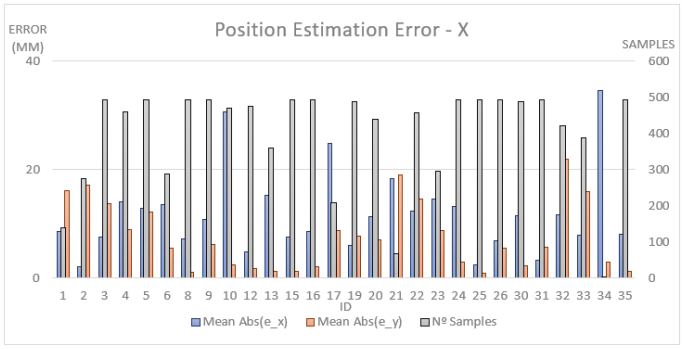
	Abs(e_x)	Abs(e_y)
Total Average	7,8434	5,7154

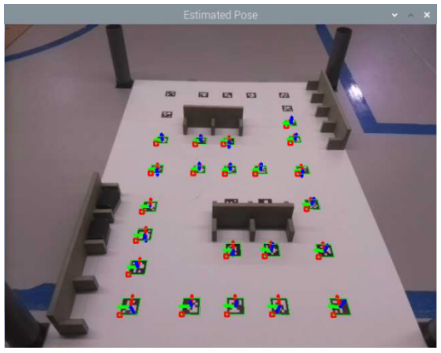
	Max	Total
Samples	463	11575

Figure 5.3: Position estimation error and number of samples for each detected field marker from a lateral view.

- **Case X**: x = 1350 mm; y = 0 mm; z = 1100 mm. Camera is oriented toward Machine A and oriented towards the field origin (longitudinally, figure 5.4). In this case, the furthest row from the camera is the ArUcos on the -695mm position on the field reference. These are severely more impacted to the point that marker ID equal to 0 is not detected from the minute-long exposure. If the point were closer to the camera and well detected, not blocked by the wall, these provided better results. Consequently, the overall average suffered a slight increase. Concerning y, these are similarly affected by the distance to the camera, showing that markers with a higher x position (closer to the camera) has a lower error for the pose

estimation procedure. Marker ID 34 is not considered in the analysis since it has been detected a negligible amount of times. Note that the detection rate of the markers is lowered while the *x* position is decreased.



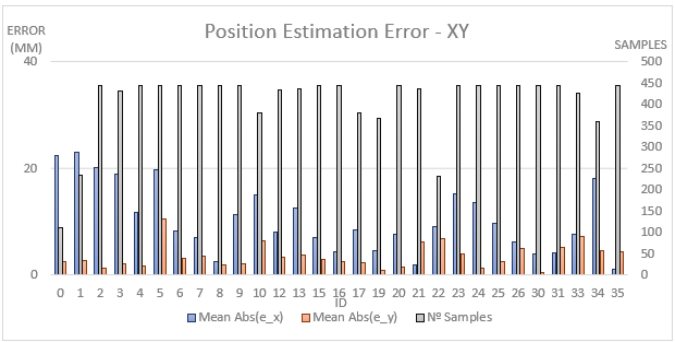


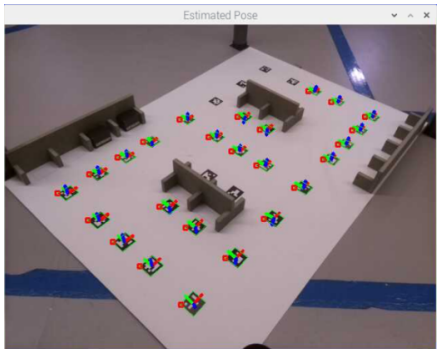
	Abs(e_x)	Abs(e_y)
Total Average	10,3928	6,8974

	Max	Total
Samples	493	11145

Figure 5.4: Position estimation error and number of samples for each detected field marker from a longitudinal view.

- **Case XY**: x = 1100 mm; y = 750 mm; z = 1100 mm. Camera oriented towards the field origin (diagonally, figure 5.5). Interestingly, from this scenario, most of the values with less error on the *x* axis are the points closer to the centre optic ray and vice-versa for the furthest ones. *Y* wise, the overall results were adequate, with a slight increase in the values that are at a greater distance from the camera position. The detection rate is similarly decreased with the influence of the distance.



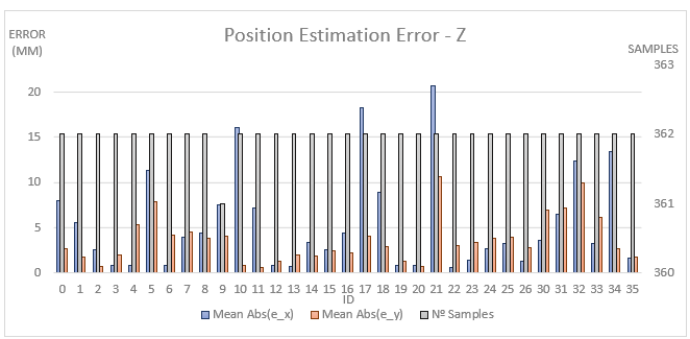


	Abs(e_x)	Abs(e_y)
Total Average	9,8718	3,4661

	Max	Total
Samples	444	11773

Figure 5.5: Position estimation error and number of samples for each detected field marker from a diagonal view.

- **Case Z**: x = 0 mm; y = 0 mm; z = 1250 mm. Camera oriented towards the field origin (vertically, figure 5.6). This last scenario has a similar effect from the *x* and *y* point of view, while an increased error is presented to the points closer to the frame's corners. Globally, the system shows lower mean and maximum values and thus better results with consistent and total marker detection.



	Abs(e_x)	Abs(e_y)
Total Average	5,4386	3,5879

	Max	Total
Samples	362	11945

Figure 5.6: Position estimation error and number of samples for each detected field marker from a top view.

On the other hand, the setup for this experiment was somewhat complex and unreliable to fix (figure 5.7), which would be recommended to build an additional structure to allow a camera setup that is mounted easily with the advantage of providing greater pose estimation results.

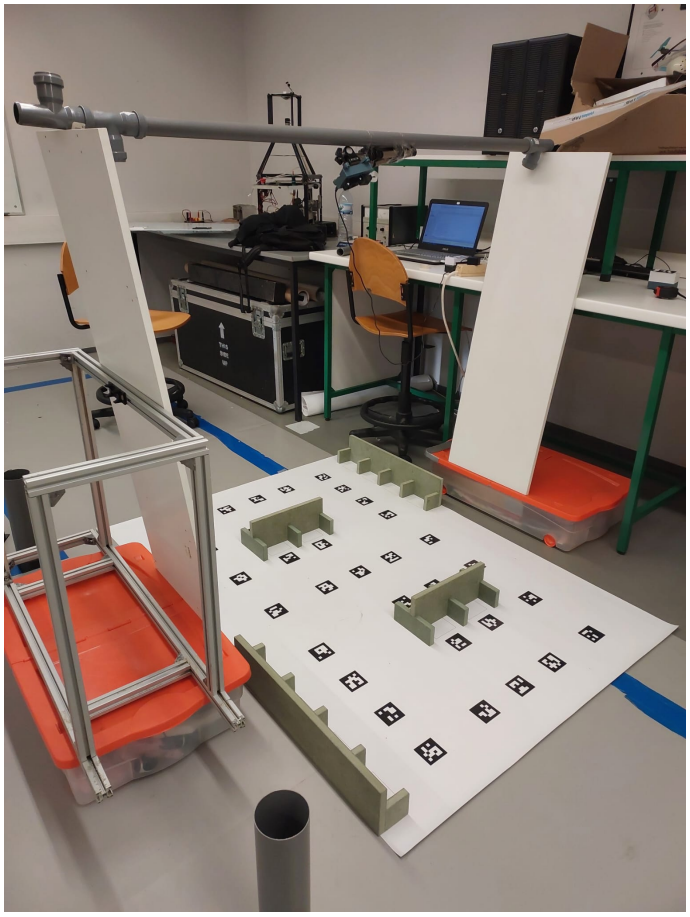


Figure 5.7: Vertical camera setup for top-view pose estimation.

After this, the effect of having a partial field view within the frame was analysed, which provided the following results (figure 5.8). By further analysis, this view did not have a negative impact on the system. The influence of the error is seen according to the distance from the camera while still keeping functional values. A slight difference between "Left" and "Right" values is the field's placement on the frame, which shows that a more centralised view provides improved results.

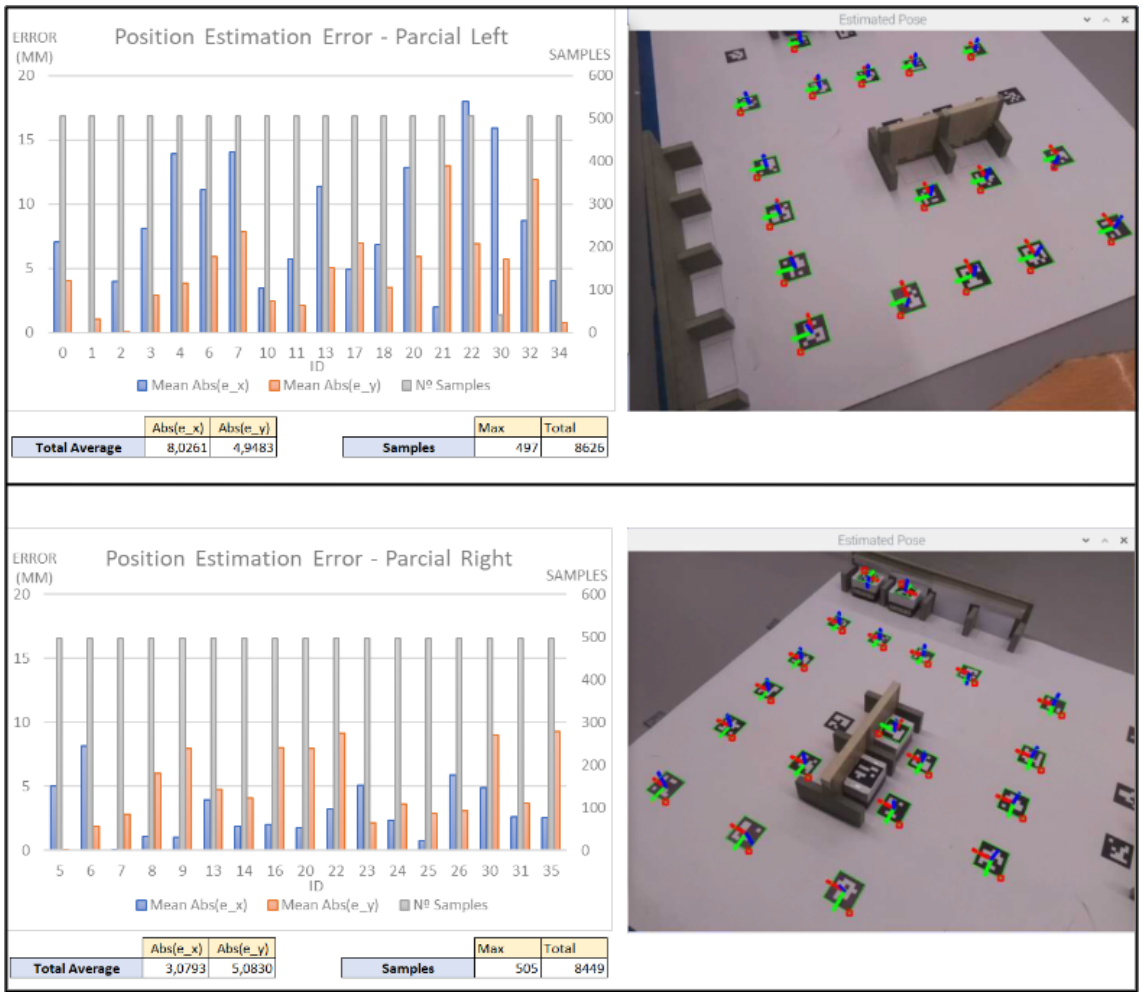


Figure 5.8: Position estimation error and number of samples for each detected field marker from a partial view.

5.2.2 Piece Tests

The piece pose detection is now tested to verify the error with and without the IIR filter to smooth out oscillations. Additionally, the test was done with a camera positioned laterally, longitudinally and diagonally with the intent to grant that all four pieces were detected within a single frame and from distinct locations throughout the areas of interest.

Note that the pieces pose estimation is done by pre-defining z equal to 65 mm and placed as follows:

- piece ID 40: x = -388; y = -150; z = 65;
- piece ID 41: x = 307; y = 150; z = 65;
- piece ID 42: x = 245; y = -535; z = 65;
- piece ID 43: x = 695; y = -535; z = 65;

From here, we should consider the fundamental goal of the piece pose estimation: verifying the detection of a part at a correct destination according to its current type. The validation is done when the piece centre is inside the correspondent box. Assuming that the piece is also positioned at the centre of the box when loaded, then we have half the dimensions of the piece as tolerance to detect, which is 4.5cm laterally and 3cm longitudinally, from a relative position of the piece with respect to the box. Since the box can be aligned vertically or horizontally on the field, we should consider the minimum value for all cases, which is a margin of 3 cm.

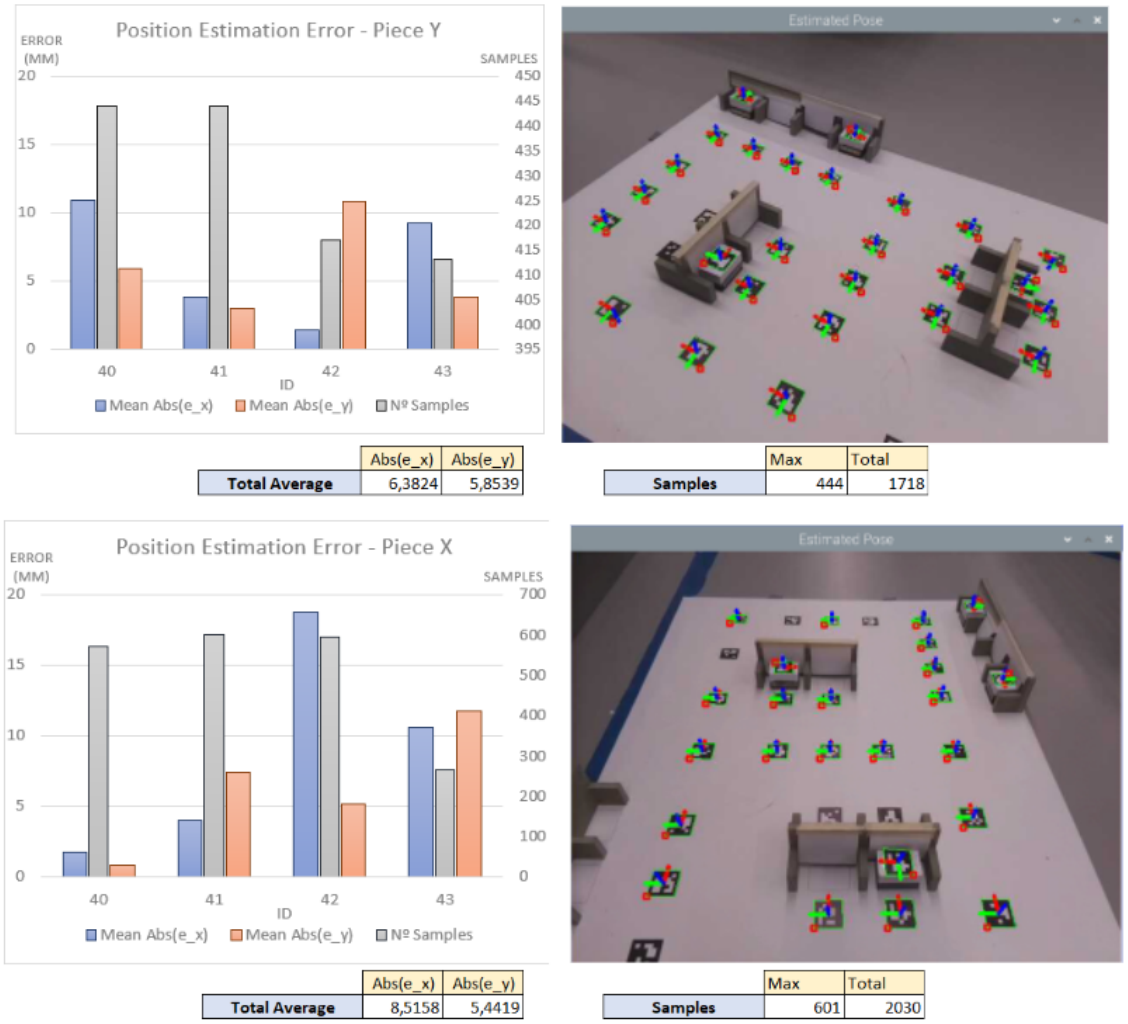
As seen from the figures 5.9 and 5.10 , all these values are within the max error that should theoretically be considered for both axis. However, we want to minimise as much as possible, which favours the "Y" and "X" setup.

These two have identical behaviours, and the benefit of one hinders the other, as seen with ID 40 and 41. Even though the maximum error seen is higher in the second scenario, it also presented higher reliability of detection rate, as opposed to the first one. Finally, the more remarkable result is evident in the marker closer to the camera and vice-versa to the furthest ones.

Moreover, the diagonal point of view, the left image 5.10 provides the worst results compared to the other positions with higher maximum error and more significant oscillations, even though these are usable values. Following this testing scenario, we now verify the impact of the infinite impulse response filter (right image from figure 5.10) used for position smoothing purposes. So, maintaining an identical setup of "XY", we gather data with the intent to verify if the smoothing is beneficial. We conclude that the filter is lowering the error and maintaining more consistent results, which provides additional versatility for positioning purposes.

In conclusion, a camera position with its optical centre aligned with the areas of interest and capturing various non-co-linear points provides better results with the pose estimation aspect. Additionally, the closer the camera is to the field plane and closer to a perpendicular alignment, the better the results. Alternatively, a wider camera lens could be placed enabling a closer setup to the field while maintaining a complete field view and consequently improving results.

It should also be noted that small oscillations, shadows, and reflective light also represents factors to the increase of error in the system. An environment with more controlled variables would also provide additional accuracy to the testing results. Nonetheless, these first analysis of the procedure performance provides acceptable and suitable values for a final and real-time application. However, the black and white contrast should be enhanced by means of white addition or higher illumination towards the field since the furthest piece detection is not as reliable.



	Abs(e_x)	Abs(e_y)
Total Average	6,3824	5,8539

	Max	Total
Samples	444	1718

	Abs(e_x)	Abs(e_y)
Total Average	8,5158	5,4419

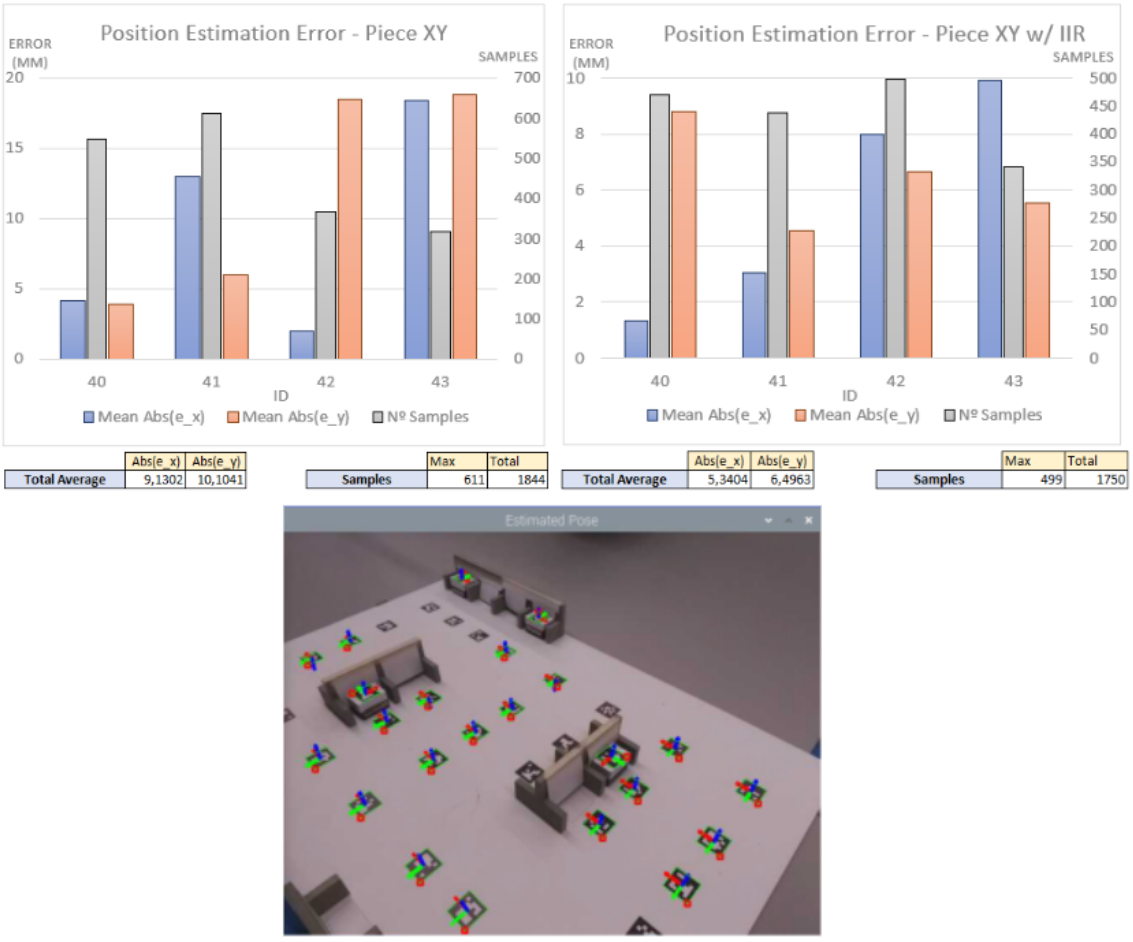
	Max	Total
Samples	601	2030

Figure 5.9: Position estimation error and number of samples for each detected piece marker from Y and X perspectives.

5.2.3 Camera Analysis

With the data gathered from the previous tests, we can be aware of the impact of the pose estimation of multiple markers. The first set of tests estimated a minimum of 28 and a maximum of 33 markers. Additionally, we recorded the maximum value of the number of samples from all detected markers since it was seen that at least one marker would be detected in every frame. This way, knowing that each test was done with a time window of 1 minute and the pose estimation algorithm is done once per frame, the frame rate of the video capture can be estimated with $fps = \frac{No.Samples}{60s}$, which provided approximately 8 and 6 frames per second (fps) respectively from the first set. Concerning the partial view, 18 and 17 markers were detected, which also granted approximately 8 fps.

Finally, considering the piece detection tests, we will direct focus on the lateral and diagonal point of view (case X and XY) since only this setup granted permanent detection of one of the pieces (piece ID 41). A max of 601 and 611 samples results in an approximate frame rate of 10



	Abs(e_x)	Abs(e_y)
Total Average	9,1302	10,1041

	Max	Total
Samples	611	1844

	Abs(e_x)	Abs(e_y)
Total Average	5,3404	6,4963

	Max	Total
Samples	499	1750

Figure 5.10: Position estimation error and number of samples for each detected piece marker from a diagonal perspective without (left) and with (right) the IIR filter.

fps. On the other hand, with some data collected from a recording at half the resolution originally used, 320x240, it was possible to see the impact of a lower resolution on a higher frame rate. The data collected from this setup was similar to the first setup, which granted us a maximum of 859 samples for 22 markers detected on a lateral positioning and a maximum of 598 samples for 33 tags detected on the vertical setup, resulting in 14 and 10 frame rates respectively. While this configuration provides a better frame rate, the camera detection was highly compromised, meaning the camera needed to be much closer to the field to detect markers reliably.

5.3 Scoring Results

Lastly, the scoring system will be tested and validated according to the three existing rounds of the competition. Throughout all the succeeding images, the input configuration shown on the visual representation window of the referee system (top right) should be noticed, as well as the piece states shown on the Clock's grid (left interface). After that, the pieces will be manually moved along the field to be detected by the camera and verified by the referee algorithm.

Round 1

This attempt shows the most straightforward round of the competition, where four blue (final type) pieces are spawned. From here, two pieces are positioned at the outgoing warehouse, where 2 points are rewarded for the correct loading. On the next frame are then placed the other two remaining pieces at the correct destination, which gives the last 2 points of the round for a total of 4 points. Note that the pieces detected have an "OnField" state that is then updated to "Done" when they reach the final place. We can also see the timestamps of each piece's arrival according to the partial timer and its spawning position. Additionally, three out of four pieces have gotten back the "OnField" state since it has not been detected within the last 3 seconds from the current instant. Note that on the figure's 5.11 last frame, the camera was repositioned due to the challenging point of view to detect all the markers while maintaining a correct pose estimation through dynamic calibration.

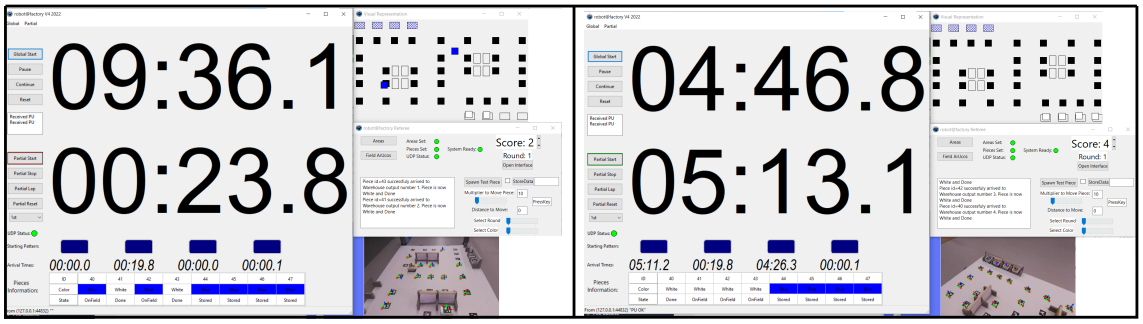


Figure 5.11: Complete system execution for piece poses verification in round 1 of the competition.

Round 2

Hereinafter, we proceed to test an attempt on the second round by spawning two blue and green type pieces whose configuration is seen to have "BGGB". At first, (figure5.12) a blue type piece is placed at the outgoing warehouse number 3, which provides 1 point to the score. Additionally, another blue type piece is placed into input box number 2 of Machine-B, showing that no points are attributed for that incorrect loading. On a final note of the first frame, a green type piece is detected at the input number 1 of Machine-A, which is a correct placement, thus updating the piece state to "OnProcess" and the type to blue. It also showcases the input and output boxes of the machine section to have an updated colour according to the colour of arrival (green) and the colour that has been transformed (blue).

Following this, on the next frame, the remaining green type piece is placed into the input of Machine-B, converting to a blue type piece. The initially green piece is now correctly at the outgoing warehouse, thus rewarding 2 points. Finally, the last two blue-type pieces are spotted at the outgoing boxes number 1 and 4, which grants an additional 1 and 2 points, respectively, for a total of 6 points (2x2 + 1x2). This concludes a successful attempt at the intermediate difficulty.



Figure 5.12: Full system execution for piece poses verification in round 2 of the competition.

Round 3

To end and validate a fully functional referee system, we test the final third round by adding a raw (red) type piece with two greens and one blue (figure5.13).

Firstly, we positioned the only blue type and one green part in the outgoing section, providing the first point and ignoring the wrong green position, and we placed the remaining two pieces at the Machine-A inputs. Since it is the third round, contrarily to the previous attempt, only the red type is correct in this scenario, which ignores the other green type part and updates the red one to the intermediate type. On the next frame, with the global timer at 7:46.2, both the originally red type piece and green type, with IDs equal to 42 and 41, are correctly placed at Machine-B input, thus updating them to the final blue type piece. Notice two details: only one of these two pieces is "OnProcess" since the other had been updated but not been detected within the last 3 seconds, hence updating back to the "OnField" state. Secondly, only three pieces are detected since piece ID 40 is shown at the ingoing warehouse box. This section does not have a white paper that provides additional contrast, which compromises the detection of the marker. In the last two frames, both previous blue-type pieces are placed at the outgoing section, thus incrementing the score by 5 (2 for the green piece and 3 for a red part) as well as the green-type piece that was placed at the input 2 of the Machine-B. Finally, to the output number 1 of the outgoing warehouse, providing the remaining 2 points for a total score of 8.

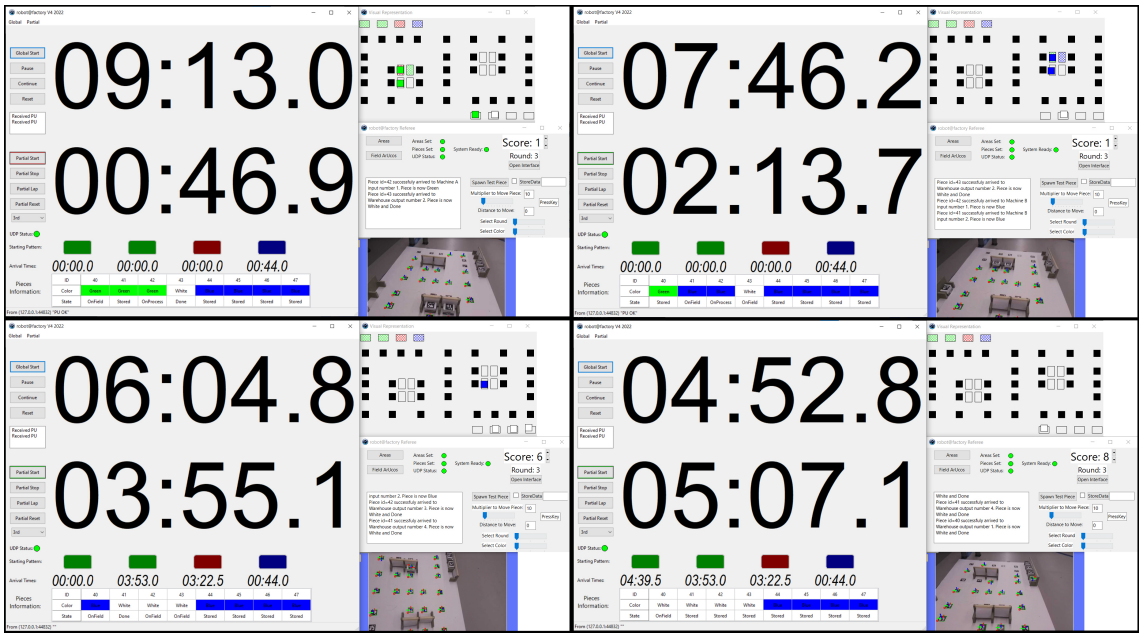


Figure 5.13: Full system execution for piece poses verification in round 3 of the competition.

In summary, the referee system received and updated the piece's information throughout different rounds while rewarding all the possible points from all the attempts made and successfully rejecting missed placements. Additionally, the system is versatile enough to allow repositioning to detect all pieces if some difficulties were apparent regarding the ArUco marker detection, as seen in the first round of the figure 5.11. We consider this a successful test with relevant results, thus validating the overall system.

Chapter 6

Conclusion and Future Work

To conclude, an autonomous-referee-system was developed, capable of taking responsibility for the score checking mechanism, acting as a server and communicating with the different devices to keep the information accessible and updated at all end-points through the use of computer vision concepts through a raspberry pi camera v2 for detection and estimation of objects position. For this purpose, the intrinsic and extrinsic calibration is done with Zhang's method and Perspective-n-Point, respectively, enabling the application of the mathematical *pinhole* camera model to map a 2-D point in the image plane to a 3-D world point in the field referential.

According to the tests carried out, it was possible to verify the full correct functionality of the system with error results quite adequate and improved by applying an IIR filter. The system was successful and gratifying at the validating test that represented the actual application throughout the three existing rounds of the competition.

On the other hand, it was notorious the difficulties of the system to detect the fiducial markers when close to objects that provide less contrast and consequently impair the system. This obstacle could be overcome by using materials with a lighter tone, adding luminosity in the field, using a structure capable of fixing the camera from a higher point of view and perpendicular to the field, and using a lens with a larger field of view (example of the fisheye lens) or using a camera that provides a higher resolution without compromising the frame rate. Improving these aspects could not only increase the detection rate of markers in more challenging positions but also decrease the error coming from the position estimation procedure.

Additionally, with the correct parameterisation and mapping, it would be possible to develop a system based on the same concepts as the referee for other applications based on localization, mapping and navigation concepts.

Nevertheless, the virtual referee proved to be a system capable of being applicable in real competition contexts plus contributing as a base framework to add more functionalities later. With this system, it would be possible to add a component of piece trajectory estimation through the application of a Kalman filter or to use the current information accessible to develop statistical data regarding each team throughout the attempts, such as the piece transportation speed, average time of accomplishment, efficiency, etc.

Furthermore, the positive results drawn from the tests with a partial field view camera encourage the implementation of multiple cameras, increasing the accuracy and detection rate from all possible piece positions. The dynamic calibration procedure has also proven versatile enough, so additional cameras would be added with ease while maintaining the field as referential.

Additionally, a multi-camera configuration, which would enable a detection from all the output boxes from the machinery, and an adjustable total number of pieces used in competition would allow the implementation and automatisation of a new feature that is, to date, manually done: piece displacement from the input boxes of a machinery section to the output zone. This way, a piece could be directly associated with another pre-placed output piece from the moment it would have been correctly dropped into the machine's input.

In summary, the system is left with good results and high hopes to be continued and further developed so that it becomes a reliable source to perform all the referee's related tasks.

References

[1] Tomas Baca, Petr Stepan, Vojtech Spurny, Daniel Hert, Robert Penicka, Martin Saska, Justin Thomas, Giuseppe Loianno, and Vijay Kumar. Autonomous landing on a moving vehicle with an unmanned aerial vehicle. *Journal of Field Robotics*, 36(5):874–891, 2019.

[2] Abraham Bachrach, Samuel Prentice, Ruijie He, and Nicholas Roy. Range-robust autonomous navigation in gps-denied environments. *Journal of Field Robotics*, 28(5):644–666, 2011.

[3] C. Dario Bellicoso, Marko Bjelonic, Lorenz Wellhausen, Kai Holtmann, Fabian Günther, Marco Tranzatto, Péter Fankhauser, and Marco Hutter. Advances in real-world applications for legged robots. *Journal of Field Robotics*, 35(8):1311–1326, 2018.

[4] Burak Benligiray, Cihan Topal, and Cuneyt Akinlar. Stag: A stable fiducial marker system. *Image and Vision Computing*, 89:158–169, 2019.

[5] Atcheson Bradley, Heide Felix, and Heidrich Wolfgang. Caltag: High precision fiducial markers for camera calibration. *VMV 2010 - Vision, Modeling and Visualization*, 2010.

[6] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–139, 1990.

[7] Paulo Costa, José Lima, and Vítor Pinto. Rules for robotatfactory 4.0 2022. *Festival Nacional de Robótica Portuguese Robotics Open*, Nov 2021.

[8] Samira Darabi and Alireza Mohamad Shahri. Vision based simultaneous localization and mapping using sigma point kalman filter. *2011 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, 2011.

[9] Francesca Digiacomo, Francesco Bologna, Francesco Inglese, Cesare Stefanini, and Mario Milazzo. Mechatag: A mechanical fiducial marker and the detection algorithm. *Journal of Intelligent amp; Robotic Systems*, 103(3), 2021.

[10] Alan Brito Dos Santos, Juliel Bronzati Dourado, and Adriano Bezerra. Artoolkit and qualcomm vuforia: An analytical collation. *2016 XVIII Symposium on Virtual and Augmented Reality (SVR)*, 2016.

[11] Amy Eguchi. Robocupjunior for promoting stem education, 21st century skills, and technological advancement through robotics competition. *Robotics and Autonomous Systems*, 75:692–699, 2016.

[12] Gabriele Ferri, Fausto Ferreira, and Vladimir Djapic. Fostering marine robotics through competitions: From sauc-e to erl emergency 2018. *OCEANS 2018 MTS/IEEE Charleston*, 2018.

[13] José Ferrão, Paulo Dias, and António J. Neves. Detection of aruco markers using the quadrilateral sum conjuncture. *Lecture Notes in Computer Science*, page 363–369, 2018.

[14] Mederic Fourmy, Dinesh Atchuthan, Nicolas Mansard, Joan Sola, and Thomas Flayols. Absolute humanoid localization and mapping based on imu lie group and fiducial markers. *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 2019.

[15] Martin Hirzer. *Marker Detection for Augmented Reality Applications*. 2008.

[16] Xin Jin, Zhehai Yang, Hao Wang, Jingyao Yin, Chengye Wang, and Fenghua Wu. Improvements of sycu humanoid robot. *Journal of Physics: Conference Series*, 1693:012204, 2020.

[17] Michail Kalaitzakis, Brennan Cain, Sabrina Carroll, Anand Ambrosi, Camden Whitehead, and Nikolaos Vitzilaios. Fiducial markers for pose estimation. *Journal of Intelligent amp; Robotic Systems*, 101(4), 2021.

[18] Michail Kalaitzakis, Sabrina Carroll, Anand Ambrosi, Camden Whitehead, and Nikolaos Vitzilaios. Experimental comparison of fiducial markers for pose estimation. *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2020.

[19] Jan Kallwies, Bianca Forkel, and Hans-Joachim Wuensche. Determining and improving the localization accuracy of apriltag detection. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[20] Dawar Khan, Sehat Ullah, and Ihsan Rabbi. Factors affecting the design and tracking of artoolkit markers. *Computer Standards amp; Interfaces*, 41:56–66, 2015.

[21] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. Robocup: The robot world cup initiative. *Proceedings of the first international conference on Autonomous agents - AGENTS '97*, 1997.

[22] Cheng Lei, Fuchao Wu, Zhanyi Hu, and H.T. Tsui. A new approach to solving kruppa equations for camera self-calibration. *Object recognition supported by user interaction for service robots*, 2002.

[23] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155–166, 2008.

[24] Boxuan Li, Jiezhang Wu, Xiaojun Tan, and Benfei Wang. Aruco marker detection under occlusion using convolutional neural network. *2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE)*, 2020.

[25] Xufeng Liang, Yuhao Du, and Dongyu Wei. An integrated camera parameters calibration approach for robotic monocular vision guidance. *2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2019.

[26] Raffaele Limosani, Alessandro Manzi, Laura Fiorini, Filippo Cavallo, and Paolo Dario. Enabling global robot navigation based on a cloud robotics approach. *International Journal of Social Robotics*, 8(3):371–380, 2016.

[27] Li Long and Shan Dongri. Review of camera calibration algorithms. *Advances in Intelligent Systems and Computing*, page 723–732, 2019.

[28] Graham G. Monkman, Kieran Hyder, Michel J. Kaiser, and Franck P. Vidal. Using machine vision to estimate fish length from images using regional convolutional neural networks. *Methods in Ecology and Evolution*, 10(12):2045–2056, 2019.

[29] Michael Montemerlo, Jan Becker, Suhrid Bhat, Hendrik Dahlkamp, Dmitri Dolgov, Scott Ettinger, Dirk Haehnel, Tim Hilden, Gabe Hoffmann, Burkhard Huhnke, and et al. Junior: The stanford entry in the urban challenge. *Journal of Field Robotics*, 25(9):569–597, 2008.

[30] L. Naimark and E. Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. *Proceedings. International Symposium on Mixed and Augmented Reality*, 2002.

[31] Nicolaus A. Radford, Philip Strawser, Kimberly Hambuchen, Joshua S. Mehling, William K. Verdeyen, A. Stuart Donnan, James Holley, Jairo Sanchez, Vienny Nguyen, Lyndon Bridgwater, and et al. Valkyrie: Nasa's first bipedal humanoid robot. *Journal of Field Robotics*, 32(3):397–419, 2015.

[32] Jason Rebello, Arun Das, and Steven Waslander. Autonomous active calibration of a dynamic camera cluster using next-best-view. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[33] Ioannis Rekleitis, David Meger, and Gregory Dudek. Simultaneous planning, localization, and mapping in a camera sensor network. *Robotics and Autonomous Systems*, 54(11):921–932, 2006.

[34] A.C. Rice, A.R. Beresford, and R.K. Harle. Cantag: An open source software toolkit for designing and deploying marker-based vision systems. *Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06)*, 2006.

[35] Ksenia Shabalina, Artur Sagitov, Leysan Sabirova, Hongbing Li, and Evgeni Magid. Artag, apriltag and caltag fiducial systems comparison in a presence of partial rotation: Manual and automated approaches. *Informatics in Control, Automation and Robotics*, page 536–558, 2019.

[36] Linda G. Shapiro and George C. Stockman. *Computer vision*. Prentice Hall, 2001.

[37] Z. C. Shi, Y. Shang, X. F. Zhang, and G. Wang. Dlt-lines based camera calibration with lens radial and tangential distortion. *Experimental Mechanics*, 61(8):1237–1247, 2021.

[38] Ansu Man Singh, Quang P. Ha, David K. Wood, and Mark Bishop. Low-latency vision-based fiducial detection and localization for object tracking. *Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC)*, 2017.

[39] Vojtěch Spurný, Tomáš Báča, Martin Saska, Robert Pěnička, Tomáš Krajník, Justin Thomas, Dinesh Thakur, Giuseppe Loianno, and Vijay Kumar. Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles. *Journal of Field Robotics*, 36(1):125–148, 2018.

[40] Zheng Tang, Yen-Shuo Lin, Kuan-Hui Lee, Jenq-Neng Hwang, Jen-Hui Chuang, and Zhijun Fang. Camera self-calibration from tracking of moving persons. *2016 23rd International Conference on Pattern Recognition (ICPR)*, 2016.

[41] Jianhua Wang, Fanhuai Shi, Jing Zhang, and Yuncai Liu. A new calibration model of camera lens distortion. *Pattern Recognition*, 41(2):607–615, 2008.

[42] John Wang and Edwin Olson. Apriltag 2: Efficient and robust fiducial detection. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[43] Oskar Wretstam. Infrared image-based modeling and rendering, 2017.

[44] Guoxing Yu, Yongtao Hu, and Jingwen Dai. Topotag: A robust and scalable topological fiducial marker system. *IEEE Transactions on Visualization and Computer Graphics*, 27(9):3769–3780, 2021.

[45] Aufar Zakiev, Tatyana Tsoy, Ksenia Shabalina, Evgeni Magid, and Subir Kumar Saha. Virtual experiments on aruco and apriltag systems comparison for fiducial marker rotation resistance under noisy sensory data. *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020.

[46] Zhenqing Zhao, Dong Ye, Xin Zhang, Gang Chen, and Bin Zhang. Improved direct linear transformation for parameter decoupling in camera calibration. *Algorithms*, 9(2):31, 2016.