Final Master Thesis

Master's degree in Industrial Engineering

Improving the Vision System Accuracy for Collaborative Robots

FINAL REPORT

Author: Mireia Graells
Director: Dr. Sandro Schönborn
Speaker: Prof. Lluís Solano

Call: June 2020

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

Abstract

Robots aim to improve and facilitate the human work and their development has extremely advanced during the past years. Year after year, the possible applications of robots is growing and, because of that, technological advances in this industry is constant. ABB is one of the companies that joined this industry and became a leading supplier of industrial robots and robot software around the world. One of the latest research areas within ABB has been towards the study of collaborative robots, meaning robots that need a high interaction with humans both for learning tasks and applying tasks.

Consequently, this research work will be focused on collaborative robots and our aim is to implement statistical methods to specifically improve their vision system, to understand better how it works and facilitate human decisions. Therefore, the purpose of this project is to generate a code which applies the Metropolis algorithm, a Markov chain Monte Carlo method, to explore the uncertainties around the vision system's estimations.

Two main areas will be investigated. On the one hand, in Chapter 8 we will investigate the actual accuracy of the detected object's estimations. For industrial applications, robots may need to carry out work in changing environments (e.g., pick up objects even if they're not always in the same place). For that, a proper vision system is required. However, different tasks may require different levels of accuracy in the vision system. Hence the importance of this part. As a result of this investigation we have provided with a specific decision-making proposal for different types of final uncertainty ranges accepted.

On the other hand, in Chapter 9 we will investigate the impact of the camera calibration into the final accuracy of the calculations. Specific set-ups, depending on the number of reference points, will be observed to analyze the impact of the camera calibration step.

To guide the reader towards these conclusions, there will first be an introduction to the current work in robotics within ABB in Chapter 2. In Chapter 3, the reader will do a quick dive into the theory behind computer vision, especially regarding projective geometries and camera projections. In the same chapter, the statistical framework and the algorithm chosen will be introduced. Then, the used programming language, Scala, is introduced in Chapter 4 and the key parts of the code will be described in Chapter 5. Thereupon, in Chapter 6, the experimental set-up is described and in Chapter 7 prior experiments will be performed to ensure the proper implementation of the code and the algorithm. The two main experiments expressed before will take place, specifying the analysis, hypothesis, and results. Finally, the impact and budget of this project will be presented in chapters 10 and 11.

To wind up, the final conclusions are presented, and further research is encouraged.

Contents

1. Introduction

1.1. Start out and motivation

ABB Future Labs is a department within ABB Group that is solely focused on innovation and research. This department is completely independent from all the business areas from the company, which are: Electrification, Industrial automation, Motion and Robotics & Automation. While all these businesses have their own R&D department, Future Labs works in a corporate level, choosing projects that may be or may not be focused on a specific ABB product. Its goal is to enhance any kind of innovation and boosting research. This was one of the main motivations before joining ABB, the possibility of working in different projects and learning from new technologies.

As a result, I ended up working in very different kind of projects, from techno-economic analysis of waste systems to robotics. Studying the vision system of a robot was something completely new for me but discovering how my background in statistics and coding could be applied in this area of expertise really motivated me to take on this project as my Master Thesis. It was a big challenge to learn about robotics from scratch, and it was completely satisfying.

1.2. Project goals

The vision system of a robot takes care of the new objects' detection, the object localization and orientation, and obtention of gripping coordinates for the robot to grip the object. The vision system then sends this information to the controller (the brain), which is responsible for the task commands and arguments. At the same time, the vision system must receive information from the brain if there's missing needed information for the commands to take place.

The vision system is implemented in the Scala language, receiving the input image seen by a camera and sending information to the brain. As a result of this project, a new vision code with specific purposes has been created, which works with the main vision module.

As explained in the Abstract, the purpose of this project is to use statistical methods and probabilistic analysis to investigate the accuracy of the vision system's calculations. In specific, the statistical method will be Bayesian inference and, the technique used, the Markov chain Monte Carlo method. With this, we will explore the uncertainty around the objects and the whole system.

The specific goals of this project are divided into two main experiments. Firstly, we will

explore the possibilities of improvement of the camera accuracy by testing its performance in different situations. Our objective is to help the user that will interface with the robot by identifying in advance what are the different parameters in the set-up that affect the final uncertainty and how strong is this impact. Secondly, we will explore the influence of the camera calibration to the final uncertainty. Our goal is to find the impact of the calibration of the intrinsic parameters with the camera accuracy and find out possibilities of improvement.

1.3. Project scope

The data used for this project is directly obtained from the experimental set-up and the camera interpretation of the world. Future Labs department in ABB works on various research projects involving collaborative robots. One of ABB's collaborative robots is, for example, YuMi (IRB 14000), which is mostly used for automation of assembly processes.

The scope of this project is based on a collaborative robot's experimental set-up in the Future Labs office of ABB. Nevertheless, it is important to mention that, due to a disruptive situation caused this year 2020 because of the appearance of the virus SARS-CoV-2, responsible for the new coronavirus disease known as COVID-19, all the experiments were needed to be performed at home, using a vision system without any access to the robot.

Thus, the same experimental set-up in Future Labs was simulated in a home space. The representation of the robot's worktable at home has been done using a low-cost camera and low-cost markers (printing them with thick paper). It will be described further in Chapter 6 Experimental set-up.

Figure 1.1: ABB collaborative robot YuMi - IRB 14000 [1]

2. Robotics and ABB

In 1974, ASEA launched the world's first microcomputer controlled electric industrial robot, IRB 6, in Sweden. From then on, more industrial robots were created, and they were exported to several countries such as Germany, UK, USA, Spain, and many more. In 1988, ASEA and BBC, the two biggest European electrical engineering companies at the time, merged to form ABB, with headquarters in Switzerland. In due time, ABB becomes leader in industrial robotics, being the first company in the world to sell 100 000 robots by the year 2002. Nowadays, ABB is a leading supplier of not only industrial robots, but also robot software and equipment, they work in 53 countries around the world and have already installed more than 400 000 robots in the industry.

Year after year, robots were optimized with cutting-edge technology, launching a robot for high production capacity in 2007, the smallest ever multipurpose industrial robot in 2009, the fastest palletizing robot in 2010, a high precision arc welder in 2012, and many more. In 2015, ABB presents the world's first truly collaborative robot with dual arm, YuMi. YuMi presents a high potential for new applications in industrial automation. [1]

Encouraging collaboration between human workers and robots helps companies to become faster and more efficient in adapting to the changes in the market. These robots not only permit a better response to these requirements changes, but also allow operations to be more flexible and productive. For this reason, ABB entered the field of collaborative robots (also known as cobots) as a pioneer and has invested in the launching of the YuMi family of cobots, which are mostly designed for applications such as handling small parts and inspection tasks.

The possibilities with YuMi are highly extensive. Not only it has been proven to work in industrial environments, but also it has been used for some applications in other industries. For example, in healthcare, YuMi has been successfully used for automating the process to assemble and test the "stem", a component of blood separation centrifuges. In addition, in banking, YuMi performed the automation of tests to improve the reliability and quality of ATMs.

To sum up, YuMi was definitely a disruptive technology when it was launched in 2015, and it opened the door to an immense room of opportunities, where humans and robots work collaboratively. A lot of applications are still unknown, and, because of this, it is of great importance that all the components that create this cobot are constantly improved and advanced with new technologies. [1]

This project focuses on the vision system of a collaborative robot, with the aim of improving it and keep advancing on the evolution of collaborative robots.

3. Theory background

In this chapter, the reader will go into the basic theory pillars of this project.

In the first place, a quick step into the concepts of Euclidean geometry, projective spaces, calibration, and more, will be taken, in order to recognize the theory behind a robot's vision system and, thus, behind computer vision. In the second place, and to conclude this chapter, the reader will be introduced into the statistics applied to solve the problem faced in this project. Concepts like Monte Carlo, likelihood and stochastic acceptance will be explained, preparing the reader for the full understanding of how the created code works.

3.1. Geometry in Computer Vision

3.1.1. Projective geometry

When observing a picture, one can quickly realize that, for example, the perfectly rectangular book cover in the real world shows a trapezoidal size in the picture, or that circles may look like ellipses. Thankfully, our brain has stored external information, knowledge, that helps one understand how those objects in the picture really look like in the real world. This is what is known as projective transformation (or homography), which describes the mapping of planar objects into a picture. In projective transformations, none of the following features are maintained: shape, lengths, angles, distances and ratios of distances. Only one property is kept: straightness. Thus, we can describe a projective transformation as any mapping of the points of a plane that preserve straight lines [1].

Euclidean geometry is largely known because it describes the 3D world very well, it describes the angles and shapes of objects. In the Euclidean geometry, length, angles and several more properties are preserved even when there are Euclidean transformations (translation and rotation). On the contrary, when looking a picture, it's clear that the real lengths and angles from the world are not preserved and, in addition, parallel lines might intersect: the imaging process of a camera is not simple. Thus, in an image, Euclidean geometry is insufficient. For example, having two Euclidean parallel lines creates a problematic when modelling them in an image. To solve this concept, it is said that those lines meet at infinity. However, infinity doesn't exist, thus this is solved by adding additional points at infinity called "ideal points". By adding these points at infinity, Euclidean space is transformed into a projective space: the Euclidean space ($\mathbb{R}^n$) is a subset of the projective space ($\mathbb{P}^n$) [7].

Therefore, the projective space is an extension of a Euclidean space where two lines will always meet in a point and, in the parallel case, they meet at points at infinity [1].

3.1.2. From Euclidean space to projective space

To conceptualize how a projective space can be obtained from Euclidean space, let's look at transformations. While in the Euclidean space, after a linear transformation, infinity remains at infinity, in the projective space the points at infinity have the value 0 as third coordinate and, thus, after applying the transformation, these points will be mapped to other arbitrary points. Therefore, points at infinity are not preserved.

Projective space is used as a good representation of the real 3D world, by extending it to a 3D projective space. So, when projecting the world into a 2D representation, a good representation is also a 2D projective space. Therefore, in computer vision, the projective space is used as a way for modeling the imaging process of a camera.

In the real world, there are no points at infinity. By adding a line (or plane) at infinity, the Euclidean space transforms into a projective space [1]. The obtention of these is shown in Figure 3.1.

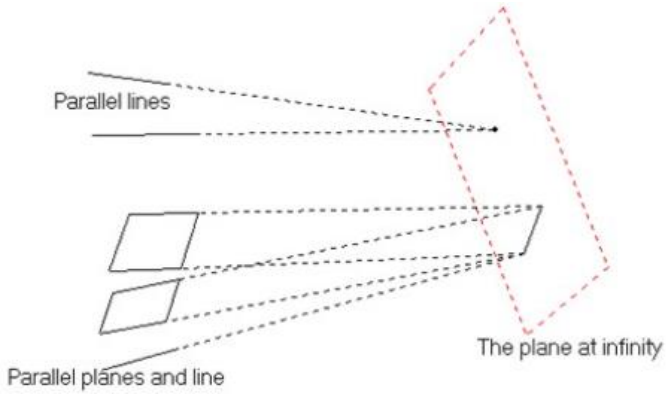


Figure 3.1. The n-dimensional Euclidean space and the entities at infinity make up the n-dimensional projective space [8]

3.1.3. From projective space to Euclidean space

To do this reverse process, one will move from two to three-dimensional projective space. As it has been mentioned before, in a projective space, there is no concept for points at infinity and, consequently, there is no use of parallelism. This is a concept that must be addressed in order to complete this reverse process.

The earth plane and the image photograph are just different ways to view the geometry of a projective plane plus a specific line. The geometry of a projective space and a distinguished line is what creates what is known as *affine* geometry.

Now, by identifying the distinguished line as the line at infinity, the concept of parallelism

of straight lines in a plane is defined.

The next step is to move from affine geometry to Euclidean geometry, which would complete the reverse process of projective geometry to Euclidean geometry. To do that, we will observe the intersection of two circles. In an affine geometry, there is no concept for circles. Because the line at infinity is persevered, circles become ellipses, and one would see 4 intersections. Considering the Euclidean geometry, being the circle a 2-degree equation, two circles would intersect in 4 points. However, geometrically they can only interest in 2 points.

The other two solutions exist in the complex form: (1, $\pm i$, 0). These points are in all circles and, hence, are in the intersection of any 2 circles. As it can be appreciated, the last coordinate is a zero, which means that these points are lying at the line at infinity. These two points lying at infinity are called *circular points* of the plane and, by identifying them, one can define the Euclidean geometry. Therefore, Euclidean geometry appears from a projective geometry by distinguishing a line at infinity and, then, the *circular points* lying on that line.

After defining how to obtain a Euclidean plane in 2D, how to obtain the Euclidean plane in 3D will be briefly described below. This is highly related to camera calibration, which is also briefly explained in the next section.

For the 3D case, the same process is followed: one will observe the intersection of two spheres. All spheres intersect at a plane, which will be the plane at infinity, shaping a specific curve. This curve is called the *absolute conic* and represents a conic curve lying on the infinity plane and only consists of complex points. As a result, the 3D Euclidean plane is defined as projective plane, plus specifying a plane at infinity and, moreover, specifying a particular conic in the plane as the absolute conic. [1]

3.1.4. Camera projections

In this section, the basics around image formation, so transformations from 3D to 2D structures will be explained. This is of a basic knowledge for this project, since a robot's vision system is based on the interpretation of the 3D real world to a projected image.

Usually, this is modelled with a *central projection*: a ray is extended form a point in the 3D space to a fixed point, which would be the *center of projection*. This ray goes on until it intersects with a specific plane known as the *image plane*. This intersection will then be the image of the former point. Such transformation is an action that loses one dimension, except the case where the 3D structure lies on a plane, which won't be the case in this project. This model is the same as the process in a camera, where the rays go through the lens of the camera (which would be the *center of projection*) and interest on a film, which produces the image. The 3D and 2D spaces, when applying the imaging process, are generally modelled

as a 3D projective space $\mathbb{P}^3$ ($\mathbb{R}^3$ with points at infinity) and a 2D projective plane $\mathbb{P}^2$.

For simplifying purposes, one can consider all the points from the ray (which maps the points in $\mathbb{P}^3$ to points in $\mathbb{P}^2$) as equal points. Thus, all images viewed by the same camera center are considered equivalent and, consequently, one can map one image to another through projective transformations, see Figure 3.2.

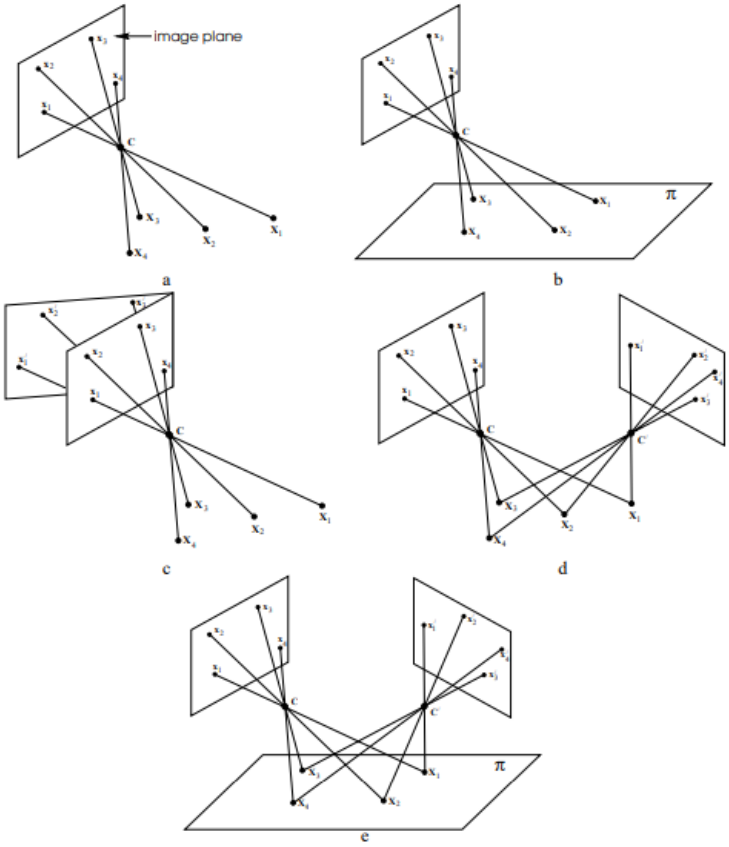


Figure 3.2: Representation of projective transformations illustrating the key role of the camera center [6]

In Figure 3.2, the following relationships are represented: in (a) the process of image formation: the rays from the space points X_i, through the center of projection C, intersect on a plane and form the image points x_i. (b) shows the case where the space points are coplanar. Both (b) and (c) show mapping of planes to one another by rays that go through a center. In (d) and (c), the camera center moves. If that happens, then the images are not related by a projective transformation, except in (e), when all the space points are coplanar. [1]

The camera model in this project is a pinhole camera. In this case, the focal length is measured as the distance between the pinhole (which would be the projection center) and the image projected (in the paper or film at the back of the camera). Shorter focal length represents wide angles, while long focal length represents a *zoom* into the picture. [3]

3.1.5. Geometry of calibration

To completely understand the relationship between the image and the real world, another topic must be discussed: their relative Euclidean geometry.

As previously mentioned, the Euclidean geometry of the 3D world can be derived from identifying a plane laying on a projective space $\mathbb{P}^3$, which would be specified as the plane at infinity, and a particular conic identified in that plane, the absolute conic.

From any point in the image, a ray in space goes through it until it intersects the equivalent single point in the plane at infinity. Consequently, the plane at infinity of the world is mapped one-to-one onto the image plane, this results in the fact that the plane at infinity doesn't add any information of the image point. Therefore, such information is obtained from the absolute conic: the absolute conic, which is laying on the plane at infinity, must project to a conic in the image, this curved image is known as Image of the Absolute Conic (IAC, denoted as ω).

In a calibrated camera, both the plane at infinity and the absolute conic are projected one-to-one onto the image plane and the IAC (ω). Therefore, the relationship between two image points and ω, is equal to the relationship between the projected rays of these two image points intersecting at plane at infinity and the absolute conic Ω_∞.

For a camera to be calibrated, the IAC location in the image must be known, and then one can be able to determine if the trapezoid book in the picture is, in reality, rectangular, or if the ellipse in the picture is, in fact, a circle in the real world. [1]

In this project, the calibration of the camera in needed in two ways:

- First, the calibration to get the intrinsic parameters (parameters that define the camera model, these are the focal length, width and height of the image, radial distortion, tangential distortion and principle points). This calibration is necessary to properly measure the objects. Moreover, it must be done only once and as a first step. In this project it will be done using the *BoofCV* library.

- Second, the calibration to get the extrinsic parameters (location and orientation of the camera with respect to the world frame). This is done by estimation functions and must be done every time the code runs.

After understanding the basic concepts behind computer vision, from Euclidean spaces to image projections, the next section will be an introduction to the statistical algorithm which is the pillar of this project.

3.2. Statistics and fitting problem

3.2.1. Bayesian fitting

For a robot's vision system to work, the step of estimating the camera position must be correctly made. In the experimental set-up for this project, for example, it is done using *fiducial markers*, which are objects placed in the system, seen in the image, and recognized to use them as a measure or reference.

However, obtaining only one solution may lead to errors, because the real-world images can suppose vast complexities. Thus, this optimization problem will be reformulated in a fully probabilistic form using Bayes Theory, which will be explained below.

Once more data is added to an observed distribution, this distribution may change: there is always some uncertainty in a solution. On that account, probabilities help quantify this uncertainty and define how certain one is with a result, what is the current knowledge of the distribution. Bayesian fitting does not try to give the *best unique* solution of the model, but a posterior distribution of the model parameters. To do that, the posterior distribution follows a conditional probability upon the inputs and outputs of the model, such that it is dependent on the prior knowledge of the distribution and the likelihood of the observations. The result is not one unique solution, but a distribution of several model parameters. Having this sequence of models allows one to calculate the certainty of the distribution obtained and, usually, the more data points observed, the lower uncertainty one will have of the distribution.

Bayesian inference is the method of updating the current certainty of the world with new observations, it is about fitting the model to the observed points in a probabilistic way. In that way, a conditional model is obtained: it is fully probabilistic, and it reflects uncertainty. On that account, Bayesian inference is summarized in Figure 3.3. [9]

Figure 3.3: Description of Bayesian inference

When one is interested in the distribution of points conditioned on the known values of other points, then one is dealing with conditional probability, which is defined as in Eq. 3.1:

Conditional probability: $$P(x_1|x_2) = \frac{P(x_1,x_2)}{P(x_2)} \quad (3.1)$$

Marginal probability: $$P(x_2) = \sum_{x_1} P(x_2, x_1) \quad (3.2)$$

Where $P(x_1|x_2)$ is the probability of x_1 conditioned on the exact value of x_2, $P(x_1, x_2)$ is the joint density and $P(x_2)$ is the evaluated marginal density, shown in Eq. 3.2.

Applying this to model general parameters θ and data D (observations), and assuming all the unknown variables are independent, one can factorize the joint distribution such that:

$$P(\theta, D) = P(D|\theta) * P(\theta) \quad (3.3)$$

Where $P(\theta, D)$ will be the joint distribution, $P(D|\theta)$ the likelihood and $P(\theta)$ the prior distribution. Being D the evidence data, P(D) would be the marginal likelihood.

By applying the factorization of the joint distribution to Eq. 3.1, one can derive the Bayes Rule, shown in Eq. 3.4:

$$P(\theta|D) = \frac{P(D|\theta) * P(\theta)}{P(D)} \quad (3.4)$$

In words, this expression can be expressed like $Posterior\ model = \frac{(Likelihood * Prior)}{Marginal\ Likelihood}$.

Therefore, Bayesian inference is a way of updating beliefs, and they are updated with each observation that changes the current knowledge, thus:

$$P(\theta) \rightarrow P(\theta|D_1) \rightarrow P(\theta|D_1, D_2) \rightarrow P(\theta|D_1, D_2, D_3) \rightarrow \dots$$

3.2.2. Algorithmic sampling: MCMC

The posterior joint probability has a lot of unknown variables and a complex structure, which makes it a complex probability to work with. Because of this, the joint probability needs to be factorized and, consequently, a method for approximating the posterior probability is needed.

To apply Bayesian inference to a fitting problem, one needs to generate samples to approximate arbitrary probability distributions to the desired posterior distribution. Thus, representative examples must be generated. This is what is known as stochastic sampling, the process of generating a set of samples, the distribution of which approximates the posterior probability of the model.

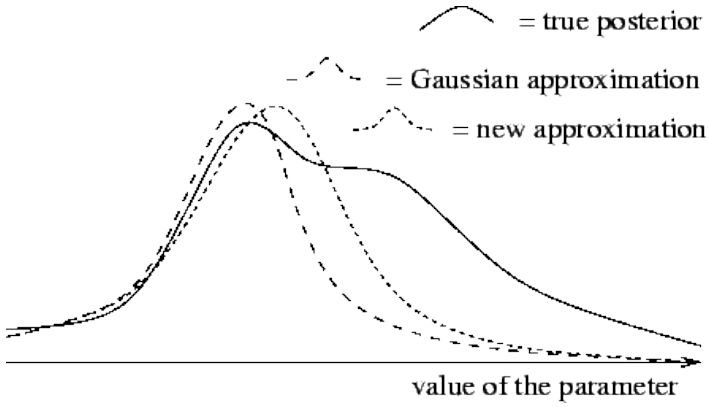


Figure 3.4: Approximation of the posterior distribution [10]

In Figure 3.4, one can see how the new generated approximation is fitted to the true posterior, starting with a Gaussian distribution. The *new approximation* shows the change that could happen in one iteration step. [10]

There are different algorithms to produce different distribution of samples. In this project, the chosen sampling process is called stochastic acceptance. Stochastic acceptance is based on the fact that the algorithm will not only choose the better solutions, but also worse solutions, depending on other parameters. [9]

There are different categories of stochastic processes, and one of them is the Markov process. Their main trait is that they follow the Markov property, which states that the next value of a Markov process depends on the current value, and the behavior in the future is independent on the behavior in the past. A Markov chain is defined as a Markov process in a discrete index set (sometimes representing time).

The Markov processes are the basis for the Markov chain Monte Carlo (MCMC) method, which is a general method to simulate stochastic samples. MCMC methods include several sampling algorithms used to construct and select the probability chain distributions, being one of the most used ones the Metropolis algorithm. The Metropolis algorithm builds the chain by rejecting or accepting the proposals and, in the following section, the reader will learn how to apply it. [11]

3.2.3. Metropolis algorithm

Let's consider the simulation of a target distribution P through random samples x_i. First, in order to apply the Metropolis algorithm, some requirements must be met: the chosen proposal distribution $Q(x'|x)$ must be able to generate samples and must be symmetric and the target distribution $P(x)$ should have a point-wise evaluation.

The aim of the algorithm is to use the known distribution Q to generate simple samples and then filter them into samples from the target distribution P, having as a result a sequence of samples from the distribution $P(x)$, which can only be evaluated point-wise. [9]

The steps of the algorithm are the following: [12]

1. Sample a candidate x' from $Q(x'|x^{t-1})$
2. Calculate $\alpha = min\left\{\frac{P(x')}{P(x^{t-1})}, 1\right\}$. If the new proposal is better than the last one, then $\frac{P(x')}{P(x^{t-1})} > 1$. Thus, α is a probability that represents either that 1 (if the proposal is better), or a lower value (if the proposal is worse)
3. Generate a value U ~ U(0,1), uniformly distributed on (0,1)
4. If U ≤ α, then $x^t = x'$ (so, if the new proposal is better, it will always be accepted). Else, $x^t = x^{t-1}$ (sometimes, even if the new proposal is worse, it will be accepted).
5. Return the sequence $\{x^1, x^2, \ldots, x^n\}$

If the proposal distribution chosen is not symmetric, then the algorithm is called Metropolis-Hastings, and the steps are the same except in step 2, where the new probability would be $\alpha = min\left\{\frac{P(x')}{P(x^{t-1})} * \frac{Q(x^{t-1}|x')}{Q(x'|x^{t-1})}, 1\right\}$.The new added part, would clearly be cancelled if the proposal is symmetric, since $Q(x^{t-1}|x')$ would then be equal to $Q(x'|x^{t-1})$.

Figure 3.5 shows the flow chart of the steps described above, in the case when the proposal distribution is symmetric.

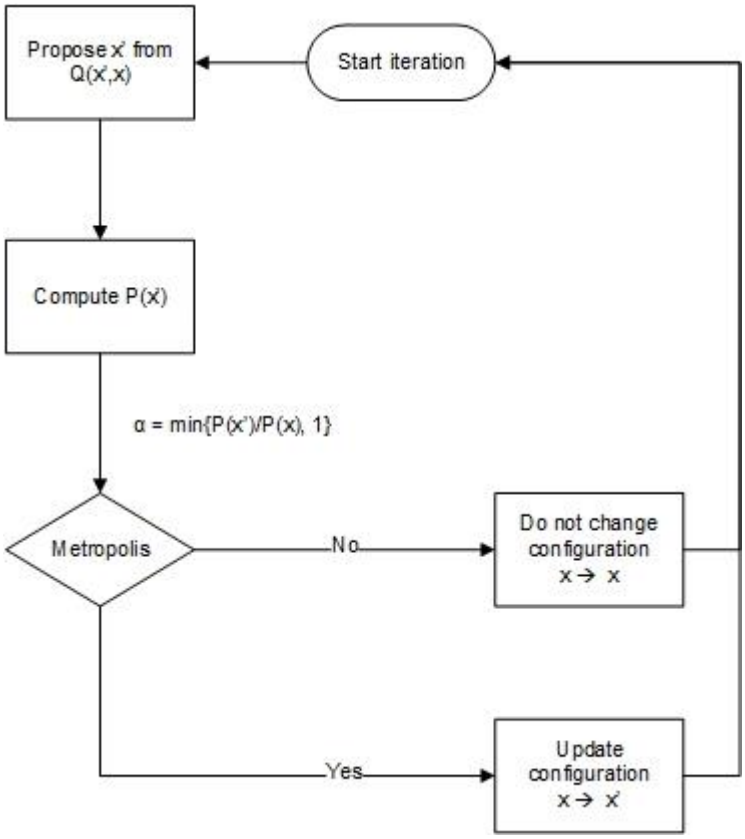


Figure 3.5: Flow chart of the MCMC method

After a long run of iterations, the samples would converge to an equilibrium. When experimenting with this data, one usually takes the last part of the samples.

To sum up, MCMC methods are a type of Monte Carlo methods based on a Markov process where each generated random sample is dependent on the previous sample. So, instead of defining the shape of a distribution beforehand, a Markov chain is created, with several samples that, all together, represent a distribution of probability. The samples are conditionally independent on each other and represent the target distribution. To conclude, the Metropolis algorithm is an example of a MCMC method, through which samples are accepted or rejected following a verification step.

4. Programming language

As it has already been mentioned, Future Labs currently experiments with vision code using the Scala language. The code of this project is based on adding statistical analysis to the vision accuracy and, as a result, it will also be programmed in Scala.

Scala is a high-level language that combines object-oriented (OOP) and functional programming (FP). It is a programming language used for general purposes, and it is built in a static type system. One of the goals of Scala was to improve the vulnerabilities of the Java language and, consequently, Scala language aims to be very concise. A Scala code returns .class files and this resulting executable code runs on the Java Virtual Machine (JVM) and, moreover, Java libraries can easily be accessed and used in Scala. [3]

The Scala version used in this project has been Scala 2.12.8. For the created code in this project, several libraries (from Scala and from Java) are used. Below, the most important ones will be briefly described:

- **BoofCV** [*https://github.com/lessthanoptimal/BoofCV/blob/v0.34/change.txt*]

  BoofCV is an open source library for real-time computer vision. Various packages from this library were used, such as the calibration package, to get the camera's intrinsic parameters. But it was also used to process the image, visualizing the image, enable the fiducial detection and to get the pose estimation of the camera from the fiducials. The version used has been BoofCV 0.34.

- **Scalismo** [*https://github.com/unibas-gravis/scalismo*]

  Scalismo is an open source library for statistical shape modeling and model-based image analysis in Scala that was developed in the University of Basel and is based on the concept of Probabilistic Morphable Models [5]. This library provides different packages that include the MCMC algorithm application: the proposal generators, the evaluators and the Metropolis sampling algorithm are provided in this library. The logger for returning the accepted and rejected proposals is also part of the *Scalismo* library. The version used has been Scalismo 0.17.

5. Creation of the code

In this chapter, the reader will go through the different parts of the created code. First, the conceptual application of the Metropolis algorithm to estimate the camera and object poses of the scene will be explained. Second, the specific parts where the Metropolis algorithm is programmed will be explained in detail, in specific the proposals generation and the likelihood evaluation. Lastly, the most important parts of the main function of the code will be described.

5.1. Conceptual statement

After understanding how the Metropolis algorithm, a Markov chain Monte Carlo (MCMC) method, works in section 3.2 Statistics and fitting problem, it will now be explained how this algorithm will be translated into the programming code.

The first step is to decide which distribution will get the camera uncertainty. In the current scheme, there are several elements to consider:

1. The fiducial: defined by 4 points in the world (the corners).
2. The origin in the world
3. The camera: defined by 6 parameters: three for translation coordinates and three for rotation coordinates. Furthermore, the camera also has its intrinsic parameters that are defined by the initial calibration done to the camera, which include the focal length.

The following Figure 5.1 shows a representation of the current scheme:

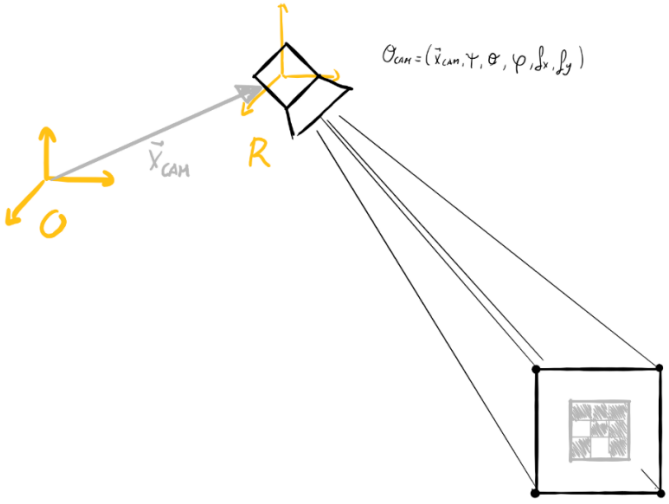


Figure 5.1: Representation of the scheme. Fiducial on the lower-right corner, camera and origin on the top

The camera parameters, orientation, translation and focal length, will define the camera pose. Therefore, the proposed distribution for the camera pose will be defined by these parameters. From now on, the camera pose is mentioned as θ_{CAM}. As the algorithm states, the goal is not to find the optimal pose, but to obtain the posterior distribution of θ_{CAM}.

With the aim of studying different scenarios a robot can face and improve the vision accuracy, the Metropolis algorithm will be applied to the 6 external camera parameters (see Chapter 8 Improving the camera accuracy). However, with the aim of evaluating the importance of calibration, a second analysis (see Chapter 9 Intrinsic Parameters impact) will be made by applying the Metropolis algorithm to 8 parameters: the 6 external ones and 2 intrinsic ones, being those the focal length in the axis x and y.

The camera poses will be obtained given the fiducials detected on the image. So, considering the known parameters from the fiducials seen, this will be used to calculate the posterior probability of the camera parameters. In the experimental set-up, there are three reference fiducials with known parameters. This is represented in Equation 5.1, where F means the number of reference fiducials and $\vec{x}_{FID}^{i}$ is the coordinate vector in the image plane.

$$P(\theta_{CAM} \mid \vec{x}_{FID}^{1},\ \vec{x}_{FID}^{2},\ \vec{x}_{FID}^{3},\ \ldots,\ \vec{x}_{FID}^{F}) \quad F = 3,4,\ldots \tag{5.1}$$

Following what is mentioned in the section 3.2 Statistics and fitting problem, one can apply the same equations and Bayes rule by taking into account the new metrics:

- θ, which are the unknown parameters, is now represented as θ_{CAM}.
- D, which is the observed distribution, is now represented as the fiducials.

Thus, applying Bayes rule, the following relationship is derived:

$$P(\theta_{CAM} \mid Fiducials) = \frac{P(Fiducials \mid \theta_{CAM}) * P(\theta_{CAM})}{P(Fiducials)} \tag{5.2}$$

Where:

- $P(\theta_{CAM} \mid Fiducials)$ is the probability of the new model distribution based on the known fiducials.
- $P(Fiducials \mid \theta_{CAM})$ is the likelihood, which is the probability of the known fiducials conditioned to the new distribution model.
- $P(\theta_{CAM})$ is the probability of the prior model distribution.
- $P(Fiducials)$ is the marginal likelihood, which is the probability the known fiducials.

To apply the Metropolis algorithm, first, the requirements must be met:

- The target distribution is $P(\theta_{CAM} \mid Fiducials)$ and has a point-wise evaluation.
- The proposal distribution is $Q(\theta'|\theta)$ and will be set symmetric, it will be defined as a mixture of Gaussian distributions.

Second, the next steps are followed:

1. Initialization: with sample θ
2. Proposal: Generate next sample of new camera parameters θ' with $Q(\theta'|\theta)$

In this project, θ is made of 6 parameters, which makes the application of the algorithm more complex. This means that, for each dimension, a new proposal must be generated by summing it with a random number from a Gaussian distribution.

After drawing the samples of proposal distributions *Q*, they will be checked with the likelihood, which leads to the third step.

3. Verification: Once these samples are generated, they will be applied to the target distribution, such as $P(\theta')$. The new proposed sample will be accepted if the probability is higher or, sometimes, even if it's not better than the past distribution, it will be accepted. This decision happens in a randomized way.

It is worth mentioning that, since the likelihood probability values can be very small, the values observed will actually be the logarithmic scales of the likelihood.

The calculation of the likelihood will take place the following way. The likelihood is the probability of the known fiducials based on the distribution in the image plane. The pose distributions of the fiducials' points are conditionally independent given the camera pose, meaning that the fiducials are independent once they are dependent on the camera pose. Because of that, the equation can be reshaped such that it is equal to the product of each fiducial point's probability individually (given the camera parameters), which is equal to the product of the normal distributions of the actual pose projected in the image based on the proposed pose, considering a specific standard deviation. This is represented in equation 5.3:

$$\text{Likelihood} = P(Fiducials \mid \theta_{CAM}) = P(\vec{x}_{FID}^{1},\ \vec{x}_{FID}^{2},\ \vec{x}_{FID}^{3},\ \ldots,\ \vec{x}_{FID}^{F} \mid \theta_{CAM})$$

$$= \prod_{i} P(\vec{x}_{FID}^{i} \mid \theta_{CAM})$$

$$= \prod_{i} \mathcal{N}(\vec{x}_{FID}^{i} \mid \vec{x}_{FID}^{i}\ (\theta_{CAM}),\ \sigma) \qquad (5.3)$$

The distribution is evaluated considering how much noise is allowed, which is represented by the standard deviation σ.

This will be carried out for each point of the reference fiducial. As it was mentioned, a fiducial is defined by its 4 corner points. Thus, considering the conditional independence for all 4 points, a distribution will be set for each of them, assuming a Gaussian noise distribution in each point.

4. Finalization (all samples generated)

Several samples have been generated (with the proposal distributions) and compared with the target distribution. As described, the likelihood acts as a filter, returning at the end a set of samples such that:

$$\{\theta_1, \theta_2, \ldots, \theta_N\} \quad where \quad \theta_i \sim P(\theta_{CAM} \mid Fiducials)$$

This will be the set of samples that will be analyzed statistically to get conclusions in this project.

In the next sections, this conceptual statement will be transformed into a code. The reader will go through the code (which can be found in Appendix A), programmed in Scala, with the aim of understanding the source of the experiments that will follow in the next chapters.

This Scala code consists on the creation of a new object called *UncertaintyVision*, which will gather the proposal generators, the likelihood function and several other functions. This code will be part of the rest of the current vision code and, because of that, the reader will see some references to already created functions.

5.2. Proposals generation

As it will be explained in section 7.2.1 Scaling of the different proposals, in Chapter 7 Prior experiments, three proposals were needed to be generated in order for the generated proposals to be close to the target distribution and, consequently, achieve a good number of accepted samples. The proposals are generated using the trait *ProposalGenerator*, which is defined in the *Scalismo* library, a library for statistical shape modeling.

In the translation proposal generator, the standard deviation is added as an input and the new proposal results in a new pose that preserves the same orientation and focal lengths as the prior pose but changes the three translation coordinates. Each coordinate is randomized by summing to it the input standard deviation multiplied by a random number. The same happens for the proposal generator of orientation, and for the proposal generator of the focal length. It is important to remark that, when changing the orientation, not only it is randomized,

but also it is transformed into the remainder of the division to 2π rad. This is done with the aim of having only angles from 0 to 2π rad (positive values), thus making sure that the posterior statistics will not be conflicted.

5.3. Likelihood evaluator

The likelihood function is the second key factor to apply the Metropolis algorithm.

For this class, the inputs are: three measurement uncertainties that should reflect the precision in the real world when placing the reference fiducials, one for each axis, a sequence of all the fiducials detected that will be used to estimate the camera pose, and the camera model, which contains the intrinsic parameters of the camera obtained in the initial calibration.

Then, for each fiducial detected, the corner positions will be compared. Firstly, the corner positions of each fiducial of the sample (the new proposed camera pose) are obtained. This is a direct step, because the Fiducials class already has stored the information of the corners in the image plane (*fid.controls2D*). Secondly, the corner positions specified to be in the real world are obtained. However, as mentioned, these are positions in the real world and they need to be projected into the image plane, which is what is done with *worldToImage()*.

Now all the corner positions (from the fiducials of the proposed sample and from the ones in the real world) are in the image plane, in 2D. However, to apply the likelihood function, the input measurement uncertainty also needs to be projected into uncertainties in the image plane. Therefore, each uncertainty introduced will be transformed into two pixels uncertainties.

The method for this transformation has been created in the function *projectWorldUncertaintyToImage*. The idea behind it is the following: in the real world, it is known the uncertainty through all directions, thus, a cube with this uncertainty as volume is created in every corner of the fiducials, then the cube is projected onto the image plane, together with the fiducial. Then, having the cube projected in the image plane, there's a new standard deviation for each corner, which would be the maximum elongation in X and Y.

Finally, the likelihood function takes place. A Gaussian distribution is created around the calculated position of the sample in the image plane, by considering a noise around it, the projected real-world uncertainty. Then, the log of the Gaussian distribution is calculated in each corner, for each coordinate (X and Y), and evaluated on the actual specified position in the image plane of the respective corner.

Therefore, this ends up giving a likelihood (log of it) for each coordinate of the corner. By summing up the log of the likelihoods of all coordinates, for all corners of all detected fiducials, the output result is the log of the likelihood for that specific sample.

5.4. Main code

The first part of the main function is detecting the image seen by the camera and specifying the input standard deviations values. The first 5 standard deviations are the ones used in the proposals specified above. It can be appreciated that there are a broad and a narrow standard deviation for the translation and rotation proposals. This was made with the aim of generating proposals more similar to the target distribution, as explained in section 7.2.1 Scaling of the different proposals, from Chapter 7 Prior experiments. The last three standard deviations are the real-world uncertainty when placing the fiducials.

There are 5 different proposal generators (two for translation, two for orientation and one for focal length). These proposals are mixed into a mixed proposal, combining all of them in different scales depending on the experiment. The Metropolis algorithm, which is an object from the *Scalismo* library, is called by giving as input the likelihood (using the reference fiducials) and the mixed proposal generator.

The rest of the main function goes through the statistics needed for the posterior analysis. First, the camera object is created by using all the fiducials detected in the image. If a camera is found, all the calculations will take place. Its initial pose will be used as initialization of the Metropolis algorithm. To make sure the angles were properly calculated and placed, a checking experiment was performed in section 7.1, from Chapter 7 Prior experiments.

After obtaining all the samples generated with the Metropolis algorithm, only the latter 5000 samples are considered for statistics, since they are closer to the target distribution. From these samples, both mean and standard deviation will be calculated. For the rotation coordinates, the statistics have more complex equations since it's in a spherical space, and they are unfolded in the function *SphericalGeomStats*. Another important information is the number of accepted and rejected samples, and their respective generator proposal. Some results regarding this are gathered in the section 7.2.2 Acceptance rate.

For each camera sample, an object sample will be obtained, and the analysis results regarding the mean and uncertainty from the objects' poses is gathered in Chapter 8 Improving the camera accuracy and Chapter 9 Intrinsic Parameters impact. Depending on whether the focal length is being explored or not, the code to get the objects is of higher or lower complexity. This is because of a *BoofCV* library feature, which causes that the objects detected before carry the camera model internally and, consequently, they carry the initial focal length values. Therefore, if exploring the focal length, the objects need to be redetected: a new fiducial detector is created and updated for every sample with the new focal length.

After going through the basic theory and the created code, the following chapters will describe the different experiments performed for this project.

6. Experimental set-up

In this section, the experimental set-up will be described. The set-up is the same for all further experiments.

To get the camera position, three stickers (fiducial markers) are placed in the table where the robot will work, these stickers will be called the reference fiducials. They give a total of 12 calibration points (4 per each fiducial). Each sticker has an ID which is recognizable in the Scala language, with the library *BoofCV*. When detecting them, their relative position to the camera is obtained and, knowing their actual position in the world, the absolute camera position can be derived.

As it has been mentioned in 1.3 Project scope, the set-up takes place in a home environment with cheap off-the-shelf materials. Regarding the stickers used, they have been printed in the exact size of 47mm each side, which is important to specify so that the fiducial detection works.

Furthermore, the camera used the detect them is the model Intel RealSense™ Depth Camera D435. The image seen by the camera is 1280x720 pixels of resolution. The illumination can also affect the quality of the image, thus, it's also important to remark that all the experiments were executed in natural light.

The scales of the measures were *mm* for distances and *radians* for rotations. So, all further measures in tables and graphics are in these scales. From Figure 6.1, one can see that the camera is placed approximately 100 cm from the wall, so 1000mm.

The set-up of this experiment is shown in Figure 6.1.

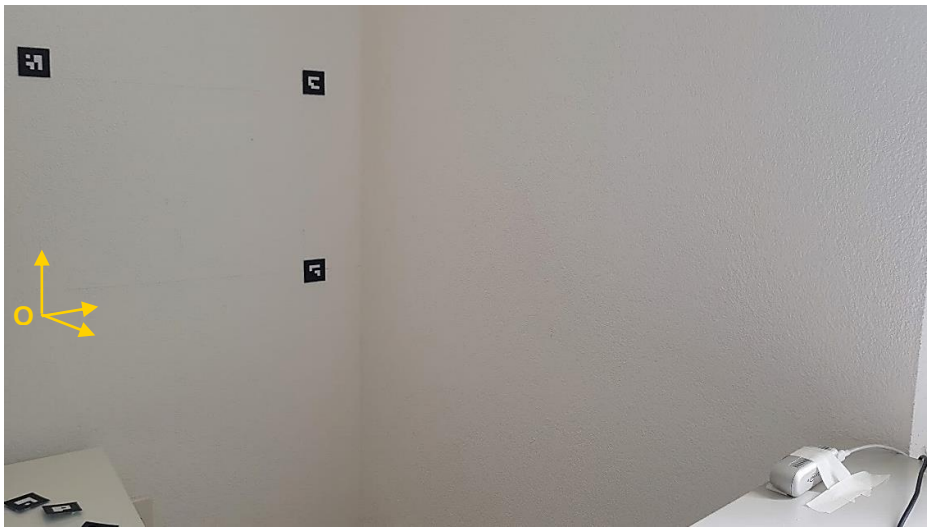


Figure 6.1: set-up of the experiment, showing the reference fiducials in the left and the camera in the right.

7. Prior experiments

Now that the code has been created, several experiments need to be performed in order to test that everything works fine. For instance, different tests were performed during the creation of the code in order to ensure, for example, that all the proposals are being taken into account, the speed of the calculations, the needed outputs for the main analysis, and more.

In addition, more tests were needed to ensure that the results are reasonable and close to reality. Thus, in this section, the reader will go through these basic tests to understand the answer to two questions:

1. Have both the Euler angles transformation and the estimation of the camera position been verified?

2. How are the input parameters of the code specified, and to which value?

To answer the first question, a verification was made observing the real-world orientation and comparing with the output of the code and, consequently, understanding the calculation of the angles.

Regarding the second question, one will need to look deeper into the proposals' generation, since the input parameters are strongly related to them.

7.1. Euler Angles and Estimation Camera position

In this section, the verification of proper calculation of the Euler Angles and the proof of correct estimation of the camera position have been included together. This is because, in order to test the former, the angles of the camera will be checked and compared to the real orientation of the camera. In this way, it is validated both the angles and the estimation of camera position.

In Figure 7.1 it can be seen the image plane, the image seen by the camera. Using Scala code, the coordinate grid of the table and axis of the fiducials and origin were drawn.

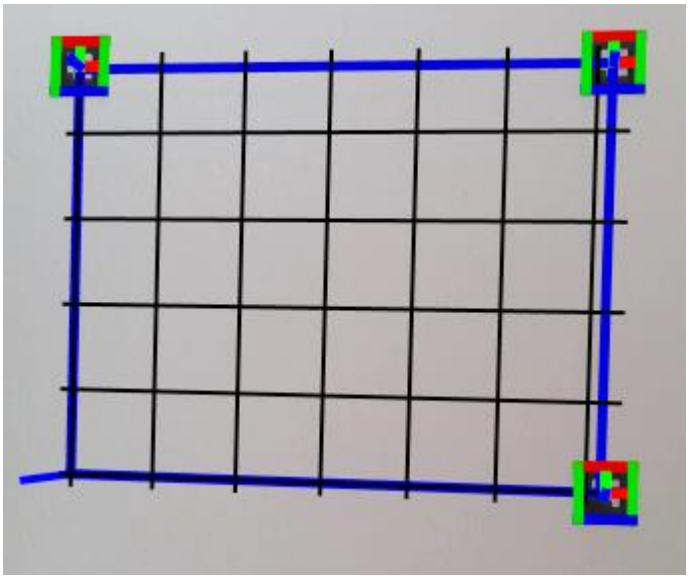


Figure 7.1: image seen by the camera, 3 reference fiducials

Applying the Metropolis algorithm through the code, the output was the following:

	Camera estimated pose					
	x [mm]	y [mm]	z [mm]	xAngle [rad]	yAngle [rad]	zAngle [rad]
Means	417.96	105.59	1070.16	3.08	0.10	0.02
Std. Dev.	14.53	12.43	14.49	0.01	0.01	0.01

Table 7.1: Pose estimation results of the camera

First, it is important to remark that the origin of the world has been placed in the lower-left corner of the fiducials table. This has been specified in a separate code named *ReferenceFiducials* where the real coordinates of reference fiducials chosen are defined considering also the chosen origin, which can be seen in both Figure 6.1 and Figure 7.1.

Looking at the translation coordinates in Table 7.1, it can be concluded that these values are in accordance with the real-word approximate measures observed. The camera is set-up almost in the same distance in the X axis as the fiducials on the right, and a bit above the origin. Regarding to the translation in Z, it can be clearly observed in Figure 6.1 that it is far away from the wall, more than 100cm away from the fiducials. The distances were measured in an approximate way and the values obtained were similar. In section 8.4.2 Accuracy of the calculations, an experiment with exact measures will be carried out. Additionally, the uncertainty values are also accompanying with the expected precision of this estimation.

In order to understand the behavior of the algorithm when estimating the camera

position, the following Figure 7.2 shows the development of the samples for three different starting points. This shows the translation coordinate Z, X and Y can be found in Appendix B.

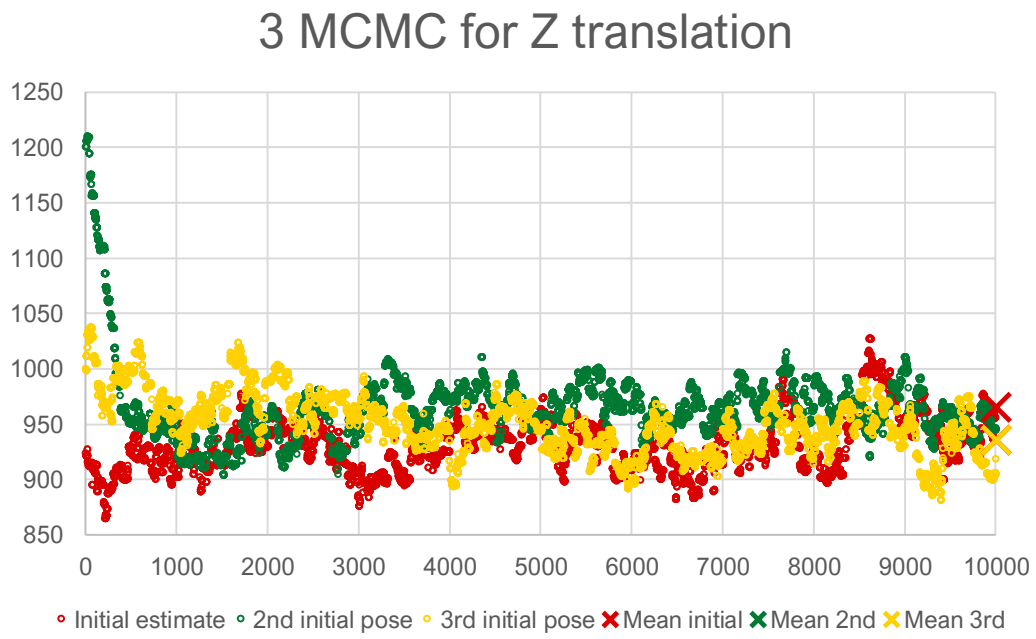


Figure 7.2: Evolution of the Z translation coordinate of the camera, in 3 MCMC methods

Regarding the orientation coordinates, let's first check them in degrees, since it's an easier metric to compare with the real-world observations. The values are, respectively: 176.47°, 5.88°, 1.38°.

Observing Figure 6.1, it can be seen that the origin is in the lower-left corner and the direction of the 3 axis: X axis to the left, Y axis to the top, and Z axis to the outside of the wall. On the one hand, with the real-world set-up, it can be clearly spotted that the camera is facing the wall and, consequently, it is turned approximately 180° on the X axis, which coincides with the results obtained. On the other hand, the table were the camera is placed, although it cannot be clearly seen in Figure 6.1, it is slightly turned: it is not exactly straight as the wall in the right, but it is slightly turned to the left so that it could see all fiducials. This means that the camera is rotated in the Y axis, which also coincides with the results obtained. Finally, regarding the rotated angle through the Z axis, since the camera is placed on a plain table which is supported on the floor, the rotation in this axis should be close to zero, which is also in accordance with the results.

Let's remember that the Euler Angles are computed through a transformation specified in a Scala function (called *rotMatrixToEulerAngles*, from the *Scalismo* library). This function embodies the following calculations explained in the pseudo-code of the document Computing Euler angles from a rotation matrix, by Gregory G. Slabaugh [13].

$$\theta = -\operatorname{asin} R_{31} \tag{7.1}$$

$$\varphi = \operatorname{atan2}\left(\frac{R_{21}}{cos\,\theta}, \frac{R_{11}}{cos\,\theta}\right) \tag{7.2}$$

$$\psi = \operatorname{atan2}\left(\frac{R_{32}}{\cos\theta}, \frac{R_{33}}{\cos\theta}\right) \tag{7.3}$$

Thus, the obtained angles are in the order: phi (φ), theta (θ), psi (ψ). However, when moving them to xAngle, yAngle and zAngle, the order is, respectively: psi (ψ), theta (θ), phi (φ), which can be verified by the observations.

In this experiment, the question *"have both the Euler angles transformation and the estimation of the camera position been verified?"* has been answered by providing the respective verification through observations. Both the camera pose estimation and the calculation of the angles rotated through the 3 axis have been validated.

In the next subsection, the second question *"how are the parameters of the code specified, and to which value?"* will be answered by going through the proposals' generation.

7.2. Input parameters value

As it has been seen in the code details during the last chapter, in the first lines of the main function there are some parameters defined with a specific value. These parameters are:

1. measurementXSdev, measurementYSdev, measurementZSdev
2. proposalBroadTSdev, proposalBroadRSdev, proposalNarrowTSdev, proposalNarrowRSdev
3. FocalLengthProposalSdev

All the parameters are type *Double*. They are standard deviations that are used to determine the final uncertainty of camera and objects.

The first parameters, called *measurement* standard deviations, directly represent the actual input uncertainty in the real-world. These are not parameters to tune the algorithm, they are just an experimental input that should represent the real situation.

Regarding the second and third parameters, these are parameters that should be tuned since they are used by the Metropolis algorithm. In the future experiments, the algorithm will be run with 10 000 iterations, obtaining thus 10 000 samples every time. As it has been explained in section 5.1 Conceptual statement, in Creation of the code, the algorithm creates 10 000 samples using the proposals and then rejects them or accepts them.

This project is divided in two main experiments, one of them maintaining the camera intrinsic parameters as an external input (focal length constant), and the other one creating different proposals of the focal length. The proposals are divided based on this. Thus, the proposals while maintaining the initial focal length constant will use the second parameters, while the proposals of new camera poses by changing the focal length will use the third ones.

7.2.1. Scaling of the different proposals

The Metropolis algorithm only needs one proposal to work. Thus, at the beginning the code was created with only one proposal that changed both the translation and the orientation coordinates at the same time. However, when changing translation and orientation coordinates all at once and independently, the proposal generated would be very far away from reality and, thus, the chance of the proposal being rejected was very high, because the camera was being translated and rotated randomly at the same time.

Hence, with only one proposal with such changes being made, almost all the samples were being rejected. Consequently, it was decided to split the proposals into two new proposals: one that preserved the orientation and changed only the translation coordinates, and one that preserved the translation and changed only the orientation coordinates. In this case, a mixture proposal was created such that 50% of the times it generated the proposal with changes in translation and the other 50% of the times it generated the proposal with changes in the orientation of the camera pose. Like this, the solutions obtained were closer to being accepted.

Furthermore, it was decided to divide all these proposals into two more: one would be for small location changes, while the other would be for broader location changes. Therefore, there would be a small proposal responsible for local exploration and a large proposal responsible for making bigger jumps, and these is how the mixture proposals finally became a mixture of the following 4 proposals:

- Broad exploration for location: proposalBroadTSdev → Only the translation coordinates are being changed to generate a new proposal. Additionally, to make bigger jumps the standard deviation used in this proposal would be high.

- Broad exploration for rotation: proposalBroadRSdev → Only the rotation coordinates are being changed to generate a new proposal. Additionally, to make bigger jumps the standard deviation used in this proposal would be high.

- Local exploration for translation: proposalNarrowTSdev → Only the translation coordinates are being changed to generate a new proposal. Additionally, to make smaller jumps the standard deviation used in this proposal would be low.

- Local exploration for rotation: proposalNarrowRSdev → Only the rotation coordinates are being changed to generate a new proposal. Additionally, to make smaller jumps the standard deviation used in this proposal would be low.

All proposals are scaled equally when running the code. If the first experiments are being run, then the percentage of each proposal would be 25%, since there are 4 of them. Otherwise, if the second experiments are being run and, thus, the focal length is being estimated, then 5 proposals would be being used and, consequently, each one of them would have a scale of 20% probability of being created.

7.2.2. Acceptance rate

The standard deviation of these proposals are the input parameters that one needs to adjust when running the code. The algorithm acts like a filter of fitting the proposal generated into the output distribution. This filter would be more efficient if the proposal is already close to the output distribution, and less if the proposal is far away from the output. To measure the efficiency of the filter, the metric observed will be the accepted / rejected samples ratio.

The goal is a 25% ratio, which means that when running the samples of the Metropolis algorithm, a minimum 25% of the samples are accepted. With this goal in mind, the input parameters (so, the proposals' standard deviation) must be adjusted.

Thus, the current standard deviations have been chosen because they provide with a satisfactory amount of accepted samples. For example. running the code without the focal length unchanged, we obtained the following distribution:

	Accepted	Rejected
CameraRProposal (0.1)	1	2510
CameraTProposal (10.0)	413	2075
CameraTProposal (1.0)	2091	409
CameraRProposal (0.01)	359	2141
Total	**2864**	**7135**

Table 7.2: Map of accepted and rejected samples using the 4 initial proposals

In total, 2864 samples were accepted, which represents a 28.6% of the total samples and is good enough for the experiments that will be performed later. That's the reason why the rest of the experiments will be carried on with the selected standard deviations: 1mm and 10mm respectively for the narrow and broad proposals of translation, and 0.01 rad and 0.1 rad

for the narrow and broad proposals of rotation, respectively.

When adding the proposal of the focal length, a standard deviation of 15mm was fixed. For the other proposals, the previous standard deviations were maintained since the accepted rate was satisfactory. Therefore, the new distribution with the 5 proposals being mixed was the following:

	Accepted	Rejected
CameraRProposal (0.1)	1	1903
CameraTProposal (10.0)	324	1766
CameraTProposal (1.0)	1701	250
CameraRProposal (0.01)	329	1704
IntrinsicProposal (15.0)	719	1302
Total	**3074**	**6925**

Table 7.3: Map of accepted and rejected samples using the 5 proposals

In this case, a 30.7% of the total samples were accepted, which is within our satisfactory rate to run the experiments.

As it can be noticed from both cases, most of the samples from the broad proposal of rotation are being rejected. This is because being the metric of the rotation in radians, a randomized change in the rotation can strongly differ from the reality. Thus, it is presumed that this proposal will have lots of rejected samples. At the same time, comparing the narrow and the broad proposals in each case, it can be expected that the narrow ones will have a higher rate of accepted samples than the broad ones.

To conclude, it's important to understand that the values of the proposals' standard deviations have been chosen so that the acceptance rate of the algorithm was good enough to perform experiments and get convincing conclusions.

In the following sections, the reader will go through the main experiments of the project. The standard deviations chosen in this section will kept constant in the following experiments since they provided with an acceptance ratio of more than 25% of samples, which is a good ratio for the experiments' conclusions to be significant.

8. Improving the camera accuracy

8.1. Objective of the analysis

With the aim of improving the precision with which the object placements are obtained from the calculated camera positions, an analysis focusing on the uncertainties of the objects positions and orientations has been performed. Therefore, the study focuses on the uncertainty distribution around the object localization. For that, several experiments have been carried out.

Each experiment performed has their own abstract, goals, hypothesis and results that will be explained in detail throughout the following sections.

8.2. Description of the analysis

To reach this, the camera uncertainty will be translated into the object uncertainty. This means that, for each camera position obtained in our samples, all the objects will be detected, and each object's pose (including position and rotation) will be obtained. Thus, the objects pose in the scenario will be conducted for each camera sample obtained.

There are two main steps in this analysis:

1. Academic point of view: as a first step, it is important to evaluate the objects uncertainties obtained from the calculations.

   Each object will be calculated for each camera (in this case, 10 000 times) and the mean and standard deviation from all the poses will give an idea of the approximation to reality and the plausibility of the uncertainty obtained.

2. Engineering point of view: as a second step, it is important to deliver practical solutions to improve the robot's efficiency when performing tasks that require visualizing objects.

   In this step, the aim is to obtain conclusions and suggestions to improve the current robot's set-up based on the uncertainty of the final estimations. For that, different scenarios will be set-up and their final object's uncertainties analyzed.

8.3. Proposed tests and hypothesis

8.3.1. Academic perspective

In this section, two tests are performed with the aim of assessing the mean and standard deviations calculation, which are key metrics in this project. The mean of the estimated objects' position and orientation will be compared with the real-world observations, while the standard deviation obtained defines the final uncertainty of the objects and is directly related to the possible applications of the robot.

1st test: General plausibility

The goal of this test is to ensure the general validity and plausibility of the statistics calculation results.

Different scenarios are selected. In these scenarios, the positions and orientations of the objects are decided such that:

- There is a fiducial inside the coordinates grid defined by the reference fiducials, which act as the vertexes of the working table where the objects are usually placed.
- There is a fiducial outside the working table, outside of the coordinates grid.
- There is a fiducial with a specific orientation, different from the default one (which is 0° rotated through the three axis).

The expectations and hypothesis are the following:

- To obtain an accurate position and orientation based on the real-world observations. The resulting means, considering their respective standard deviations, should find within their interval the approximate real-world measures.
- To obtain an uncertainty close to 0 for the rotations and between 0.5 cm for the translation parameters. Since the rotation is expressed in radians, the values obtained for the orientation uncertainty should be very low since, for example, 0.1 rad are already 6° of difference, which represents a big difference from the real orientation. For translation coordinates, for this first check we would expect to see standard deviations lower than 50mm.

2nd test: Accuracy of the calculations

The goal of this test is to have an idea of the actual accuracy of the calculations.

In the second test, one object's position and orientation are carefully calculated in the real-world, and, at the same time, obtained through the algorithm running 10 000 samples (thus, calculating 10 000 camera samples and, for each camera sample, an estimation of the object's pose).

It is expected that all the parameters (3 for translation coordinates and 3 for rotation coordinates) are within the intervals of the mean and standard deviation obtained.

The results of these tests are found in section 8.4 Results of the analysis.

In the following section, the reader will go through the engineering perspective of the analysis. In this section, the experiments are aimed at obtaining practical suggestions for the future engineering projects with collaborative robots.

8.3.2. Engineering perspective

The past two checking experiments and the gathering of necessary information were actions performed with the goal of confirming that the calculation of both the camera and the objects' poses are accurate and plausible. When these points are confirmed, the next phase will be to get some more practical conclusions to improve the robot's vision system accuracy.

Therefore, the goal of this section is to understand how different parameters have an impact to the objects' uncertainty, and how can one adjust these parameters to achieve better performing results.

In total, 17 tests were performed. Each of them is a combination of several parameters that are explained below. In the end of this section, a visual summary of all the tests is shown, with the aim of helping the reader understand the different combinations performed.

The objects placement (2 scenarios)

The aim of these experiments is to conclude if there is a difference between the final estimated uncertainties of the objects' poses by the fact of having them placed in different planes than the reference fiducials plane or in the same one.

With this, one aims to be able to answer these questions: *will one get more precise results if all the objects are placed in the same plane as the reference fiducials? Is the plane of the objects something to consider in the results obtained? Would this impact a lot the certainty and precision of the results?*

Thus, the uncertainties will be calculated for two scenarios depending on the placement of the fiducials detected as objects:

- First scenario: the objects are all in one plane. The chosen plane is defined by all the points of the fiducials being in Z = 0.
- Second scenario: some objects are placed in a plane higher in the Z axis. In this case, some fiducials will have points with the coordinate Z ≠ 0.

The first scenario has a more experimental approach, while the second scenario tries to better reflect real situations.

As it has been mentioned before, the current set-up has the fiducials taped in a wall. This causes that, in the first scenario, not only the reference fiducials will be in plane Z = 0, but also the object fiducials. The reasoning behind the second scenario is that this would represent a more usual scenario situation that the robot can encounter. This is because the objects that are placed in the table usually have the fiducial code on the top, and thus they are found in a different plane than the reference fiducials because of their height.

For the scenario with objects in different planes, the fiducials 5# and 20# have been moved through the Z axis. Also, they have been rotated in the following way: number 5# around the X axis and number 20# around the Z axis.

It is expected that the final uncertainties of the objects' poses don't change, and that they are within the same approximate values in both cases: being in the same plane of the reference fiducials or in different ones.

Measurement standard deviation (2 combinations)

The aim of these tests is to study the impact of the input measurement uncertainty to the final uncertainties of the objects' position and orientation. The question one wants to answer in this case would be: *how precise should one have to be when placing the reference fiducials? Does it strongly impact the final uncertainties?*

In this case, high standard deviation is considered 20 mm and low standard deviation 5 mm. This is the actual uncertainty that corresponds to the real-world measurement precision.

It is expected that, with the lower measurement standard deviation, the final uncertainties of the objects lower significantly. The more precise the fiducials are placed, the more precise results in the calculations will be obtained, and vice-versa. This is because a low input uncertainty will directly increase the likelihood of the proposals generated, and thus, reducing the final uncertainty of the fiducials' poses.

Plane of the reference fiducials (2 combinations)

The aim of these tests is to investigate if placing a reference fiducial in a different plane than Z = 0 improves the final uncertainties of the Z translation coordinates of the objects, answering the question: *does one need to consider the plane of the reference fiducials when placing them, in terms of the final uncertainty?*

As it was observed in the previous tests and mentioned in the first conclusions and observations of the analysis 8.4.1 General plausibility, the uncertainties of the Z coordinates calculation is significantly higher than the uncertainties on the X and Y axis. Because of this, these experiments have been designed to investigate if the uncertainty in Z can be improved.

For logistical reasons, the reference fiducial advanced in the Z axis will always be a reference fiducial added in the lower-left corner of the grid. For that, the new position is accurately calculated. This reference fiducial will be number 22# and will be placed in the position (0, -36, 165) with zero rotation around all the axis.

From these tests, the following is expected:

- In the case were the objects are placed in the same plane (in Z = 0), the uncertainty in Z is not expected to be affected by the action of changing the plane of the reference fiducials.
- In the case were the objects are placed in different planes, the uncertainty in Z is expected to improve for the objects that also have points in the Z direction.

Number of reference fiducials (3 combinations)

The aim of these tests is to see if adding an extra reference fiducial would significantly impact our object calculations precision, or if removing one fiducial would still give positive results, answering *what would be the needed reference fiducials and their position, to improve uncertainty?*

Regarding the specific number of reference fiducials: 3 is the default number and tests will be made by adding an extra reference fiducial (4 in total) and by removing one (2 in total). When adding a reference fiducial in the same place as the others, it will be number 22# and in the point (0,0,0), with no rotation. When removing a reference fiducial, the used reference fiducials were 1234# and 911#. On the other hand, when adding a reference fiducial in a different plane we will add the same specified in the previous subsection and remove other accordingly to the experiment (which can be with 4, 3 or 2 reference fiducials).

It is expected that, with a higher number of reference fiducials, both the uncertainties

for translation and rotation estimated means would decrease, while with a lower number of reference fiducials, they would increase.

As it has been discussed, there are some relationships and impacts between the past presented parameters that are expected, while there are some that are not. For example, it is not expected that the improvement of uncertainty in an object fiducial caused by changing the plane of the fiducial, is dependent on the number of such fiducials. The possibilities of relationships and interactions between those parameters and their impact to uncertainty is very complex.

In order to make this analysis easy to follow, a specific path to follow for these experiments has been marked, and Figure 8.1 shows a summary of the scenarios that have been tested with the aim to obtain more practical-oriented solutions. Let's remember that the tests will be done in two scenarios: one with all the objects on the same plane and one with some objects on different planes.

1. Both scenarios will start at the same point, all the parameters set up with the same values: 3 reference fiducials and high measurement standard deviation. Between these two scenarios it will be observed if there is a significant difference between the obtained uncertainties.

2. The first test to be done in each case will be to change the reference fiducials' plane. In one test all the reference fiducials will be in the plane Z = 0 and the other one, one of them will be in a plane Z > 0. The uncertainty results will be compared. If there is no change, the experiments will continue with the reference fiducials in the plane Z = 0. If there is a difference in the uncertainty results, the following tests will be continued with both cases, so two branches will be created (one representing the reference fiducials in Z = 0, and one representing the case with a reference fiducial in a plane Z > 0).

3. Either if one or two branches are continued from the last point, for each of them two more tests will be performed: one entering low measurement standard deviation, and the other with high standard deviation. The following tests will be continued in both cases if there is a difference in uncertainties and, if not, it will only continue with one of them.

4. Finally, a comparison of uncertainties obtained from three different combinations will be done: with 3 reference fiducials (default set-up), with an extra one and with only 2.

This process will give a good understanding on the impacts of each parameters to the calculated uncertainties of the estimated objects' poses. The process is summarized in Figure 8.1. Moreover, in Table 8.1 the expectation of interaction between all the parameters is summarized.

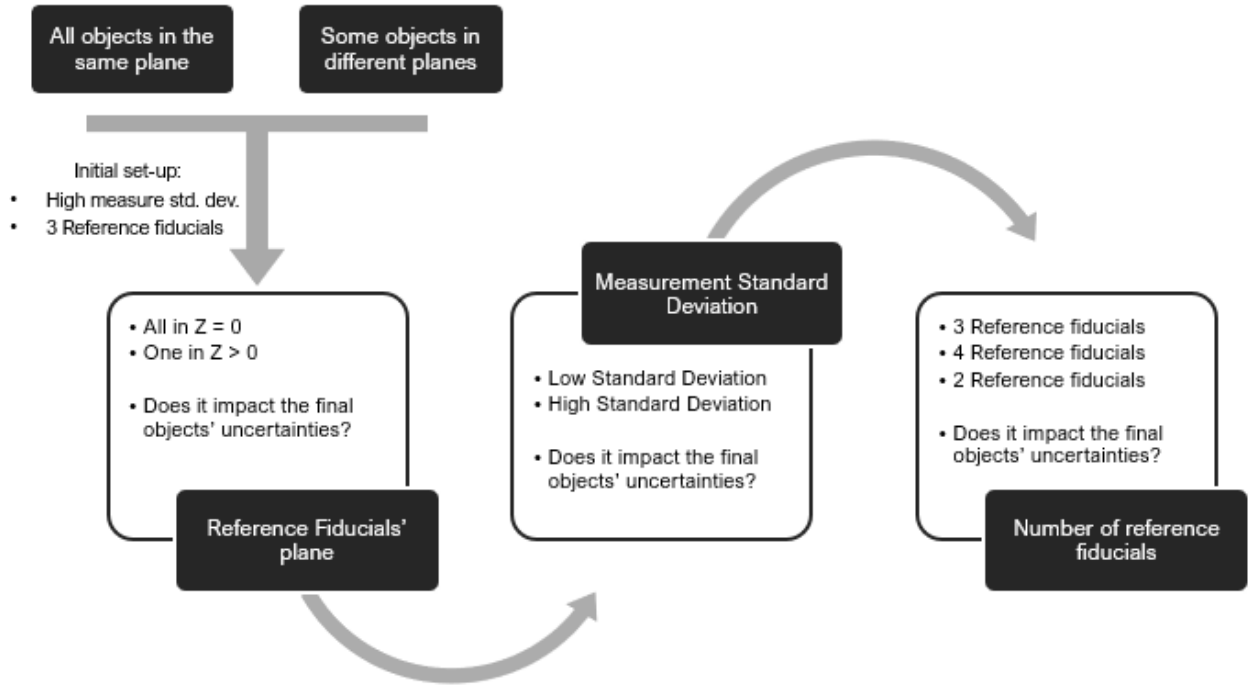


Figure 8.1: Summary of the processes followed for the experiments of objects' uncertainties

	Expected impact to the final uncertainty of the objects?
Plane of the objects	No
Plane of the reference fiducials	Yes, but only to the objects that are in a different plane
Measurement standard deviation	Yes, always
Number of reference fiducials	Yes, always

Table 8.1: Expectations of the parameter's impact to the final uncertainties

In the results section, the uncertainty values will be observed, placing special attention to the relations specified in Table 8.1. In this table, standard deviation and the number of reference fiducials are expected to impact the uncertainty always, independently of the other parameters value. The results will confirm or reject these hypotheses.

8.4. Results of the analysis

8.4.1. General plausibility

An initial checking analysis was made to ensure the proper calculation of the objects' pose, which is defined by the objects' position (3 translation coordinates) and orientation (3 rotation angles around each axis).

The three different scenarios mentioned in subsection 8.3.1 Academic perspective, from the section 8.3 Proposed tests and hypothesis, are described in more detail in the following Table 8.2:

	Fiducial 5#	Fiducial 20#	Fiducial 30#
Scenario 1	• Inside the table • Not rotated	• Outside the table • Not rotated	• At the origin • Rotated around Z axis around 145°
Scenario 2	• Inside the table • Rotated around Y axis roughly 45°	• Outside the table • Rotated around Z axis roughly 360°	• Inside near the origin Advanced in Z axis • Not rotated
Scenario 3	• Inside the table • Rotate around Z axis roughly 45°	• Outside the table • Rotated around Z axis roughly 200°	• Outside the table, Z > 0 • Rotated around X axis roughly full 360°

Table 8.2: Summary of the three scenarios for the *General plausibility* experiment

For example, Figure 8.2 below shows the images from the camera for the Scenario 1 and the Scenario 3:

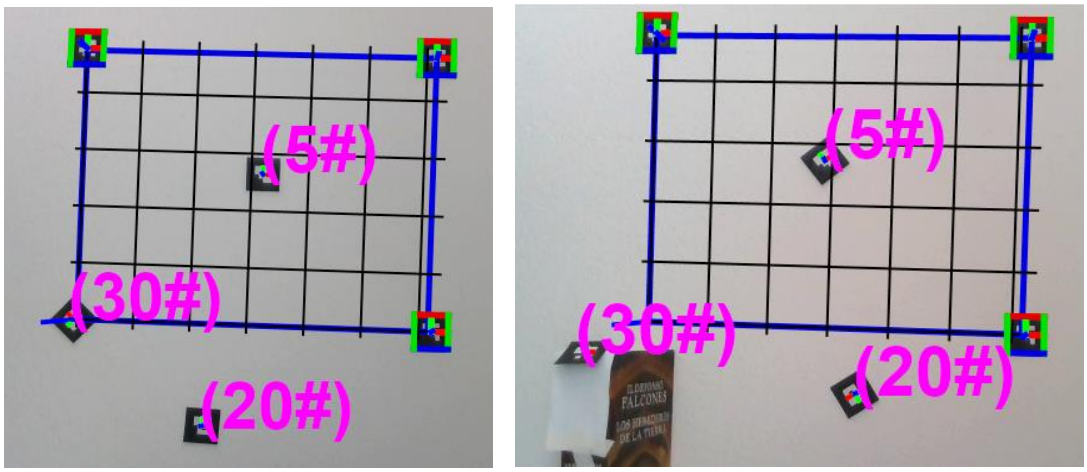


Figure 8.2: Set-up of Scenario 1 and Scenario 3. There are 3 reference fiducials and 3 object fiducials

The results obtained regarding the statistics, mean and standard deviation, of the calculated pose (including both translation and orientation coordinates) of the three detected object fiducials are summarized in the following Table 8.3:

		Scenario 1	
		(x, y, z) - [mm]	**(ψ, θ, φ) - [rad]**
Fiducial 5#	Mean	(253.84, 215.06, -14.56)	(0.32, 0.29, 0.03)
	Std. Dev.	(1.927, 2.627, 4.719)	(0.002, 0.001, 0.001)
Fiducial 20#	Mean	(181.55, -132.77, 18.70)	(6.14, 6.28, 6.23)
	Std. Dev.	(2.020, 2.610, 4.620)	(0.002, 0.001, 0.002)
Fiducial 30#	Mean	(-1.64, 7.61, 8.51)	(0.07, 0.02, 2.39)
	Std. Dev.	(1.963, 2.736, 4.725)	(0.001, 0.002, 0.002)

		Scenario 2	
		(x, y, z) - [mm]	**(ψ, θ, φ) - [rad]**
Fiducial 5#	Mean	(80.94, 198.34, 15.79)	(6.04, 0.67, 6.15)
	Std. Dev.	(1.900, 2.658, 4.778)	(0.002, 0.001, 0.001)
Fiducial 20#	Mean	(351.19, -65.14, 2.49)	(6.09, 6.25, 5.92)
	Std. Dev.	(1.985, 2.535, 4.593)	(0.001, 0.002, 0.002)
Fiducial 30#	Mean	(-11.24, -53.05, 158.12)	(6.26, 6.22, 0.05)
	Std. Dev.	(1.896, 2.493, 4.699)	(0.002, 0.001, 0.002)

		Scenario 3	
		(x, y, z) - [mm]	**(ψ, θ, φ) - [rad]**
Fiducial 5#	Mean	(228.61, 221.78, -10.51)	(6.26, 0.07, 0.68)
	Std. Dev.	(1.910, 2.614, 4.727)	(0.002, 0, 0.002)
Fiducial 20#	Mean	(273.36, -80.85, 4.84)	(0.08, 6.25, 3.71)
	Std. Dev.	(1.995, 2.568, 4.606)	(0.002, 0, 0.002)
Fiducial 30#	Mean	(-12.40, -21.13, 151.28)	(5.23, 0.04, 0.02)
	Std. Dev.	(1.891, 2.507, 4.710)	(0.002, 0.001, 0.002)

Table 8.3: Table of results of the General plausibility experiment

First conclusions and observations

From looking at the mean values of each fiducial position and orientation and contrasting it with the observed pose, one can conclude that these results match the approximate positions and angles in the real world. Thus, the mean derived from all the objects estimated poses is properly calculated. Regarding the standard deviations obtained, on the one hand, the orientation uncertainty is, in all cases, extremely close to 0 radians, which is in accordance with the hypothesis made in section 8.3 Proposed tests and hypothesis. On the other hand, the translation uncertainty is also within the expected values, being between 1 and 5 mm of uncertainty.

It's interesting to notice that the uncertainty for the X and Y translation coordinates is significantly lower (between 1 and 3 mm) than for the translation of the Z coordinate, which is close to 5 mm. Although this was not expected when specifying the hypothesis, it is understandable than the uncertainty of the calculations in the Z axis is higher, since no real-world metric in the Z axis has been stated in the code. All the reference fiducials are in the same plane defined as Z = 0. With this, one can think that having a reference fiducial in a different plane, such that it is advanced in the Z plane and this distance is accurately measured, would maybe improve the uncertainty of the calculations of the Z coordinates. Because of that, an experiment to confirm this idea's probability will be done in the section 8.4.3 Object placement uncertainties. In the figures below one can see the evolution of 3 Metropolis algorithms run with different starting points to estimate the position of one object. In this case, it was outside the grid. Notice how the estimation of the Y coordinate (Figure 8.3) has less uncertainty than the estimation of the Z coordinate (Figure 8.4).

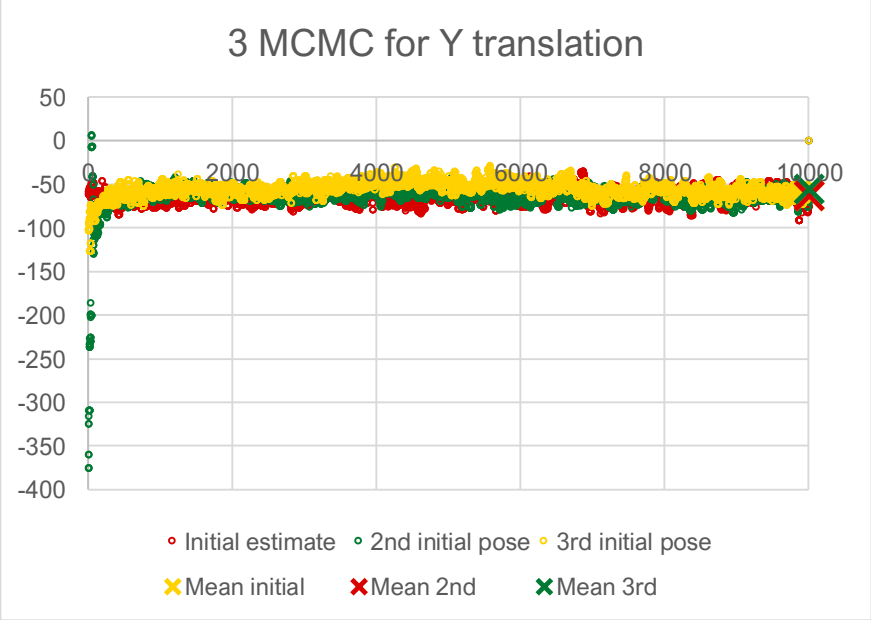


Figure 8.3: Evolution of the Y translation coordinate of an object, in 3 MCMC methods

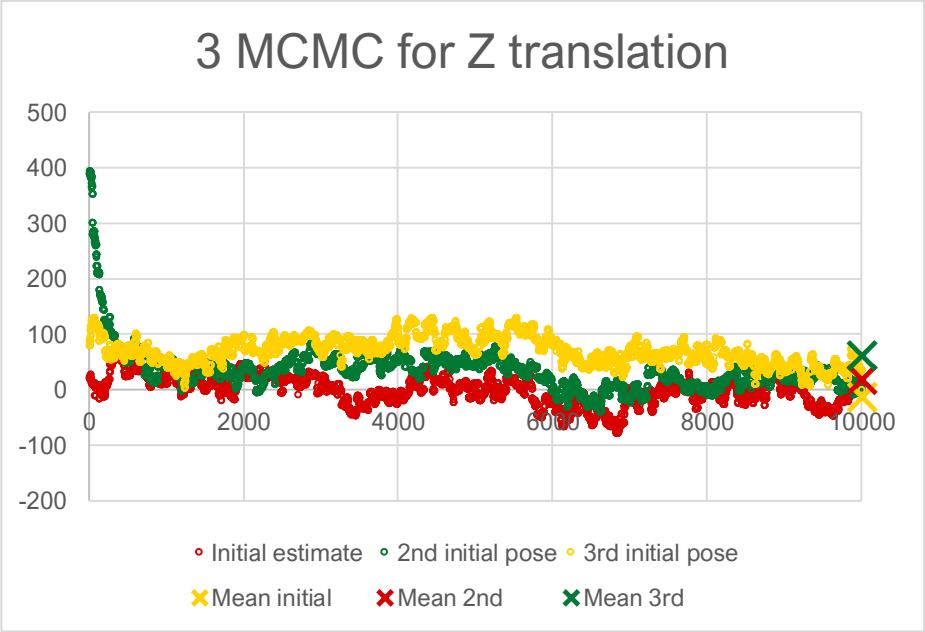


Figure 8.4: Evolution of the Z translation coordinate of an object, in 3 MCMC methods

8.4.2. Accuracy of the calculations

To continue the analysis, and before going into the different object placements analysis, a second checking analysis was made. In this analysis, the objective was to calculate an object position from which we accurately know the exact world position.

In the current set-up, there are three fiducials from which we already know the exact position, since they are the basis of our code to estimate the camera position: these are the reference fiducials. Therefore, for this 2nd checking test, it has been decided to use one of the reference fiducials as the real-world object from which we will compare the estimated pose with the exact real pose.

This means that, in this test, one of the references fiducials will be treated as an object and removed from the reference fiducials list. Thus, its exact position won't be specified in the code anymore and, instead, its calculated position will be estimated for each of the 10 000 camera samples. This position will be compared with the exact real position in the world. If the real-world position is within the limits of the obtained mean position, taking into account its uncertainty (standard deviation), it will be concluded that the precision of the object positions is very accurate.

In this test, the reference fiducial 944#, which is the one placed in the upper right corner of the grid, will be treated as an object. Consequently, only the fiducials 1234# and 911# will be used as reference fiducials.

These have been the obtained results:

Fiducial 944#	***(x, y, z)* - [mm]**	**(ψ, θ, φ) - [rad]**
Real position	**(492, 380, 0)**	**(0, 0, 0)**
Calculated Mean	(495.80, 379.04, 1.20)	(0.066, 6.207, 6.277)
Calculated Std. Dev.	(2.02, 2.05, 4.93)	(0.003, 0.002, 0.007)

Table 8.4: Table showing the results of the Accuracy of the calculations experiment

First conclusions and observations

As it can be appreciated, both the position and orientation have been correctly calculated and their means are accurate with respect to the real position and orientation that the fiducial has. The rotations around the axis are slightly different, having obtained the following angles: 3.78° through the X axis, -4.37° through the Y axis and -0.35° through the Z axis. Although these values, with their uncertainty, don't exactly reach the point of 0°, this can be understood by observing the real set-up: the fiducials are placed in the straightest way possible, however, in the experimental set-up the fiducials are taped in a wall. As a result, the tape can be slightly detached and it could happen that they are a little bit loosen from the corners or separated from the wall, which creates a slight rotation. This slight rotation was not able to be measured in the real world. However, the difference is very low and, thus, the estimated angles seem reasonable.

It's worth mentioning that the experimental set-up in the Future Labs office is quite different from the one at home. There, the fiducials are not taped in the wall but in a table fixed in the floor. Consequently, the position measurement is easier and there's not that high of a risk of the tape detaching.

It is concluded then that the code is giving accurate and precise results regarding the calculation of the objects poses.

8.4.3. Object placement uncertainties

In section 8.3.2 Engineering perspective, the process followed for this analysis has been explained and represented as a summary in Figure 8.1. This part is of great importance for this project, since several realistic situations will be experimented on and the aim is to conclude with specific, practical recommendations to improve a robot's applications depending on the uncertainty of calculations.

8.4.3.1. Experiment with all the objects on the same plane

First, the scenario with all the objects in the same plane is presented. The initial set-up is the following:

- There are the default three reference fiducials in place (944# in the upper-right corner, 1234# in the upper-left corner, and 911# in the lower-right corner)
- The input standard deviation is in the high value (20 mm)
- The number of objects and their position in the grid is indifferent, since it has already been verified in the previous sections that the calculations are correct. For the sake of diverseness, the objects may change positions and orientations from one test to another. However, the important metrics one needs to focus the attention on during the following experiments are the uncertainty metrics.

Reference fiducials' plane comparison

With this initial set-up, the uncertainty results will now be obtained and compared between two modifications applied: on the one hand, the 3 reference fiducials will all be on the same plane (in Z = 0) and, on the other hand, one of the 3 reference fiducials will be in a different plane (advanced in Z > 0). The two set-ups are shown in Figure 8.5.

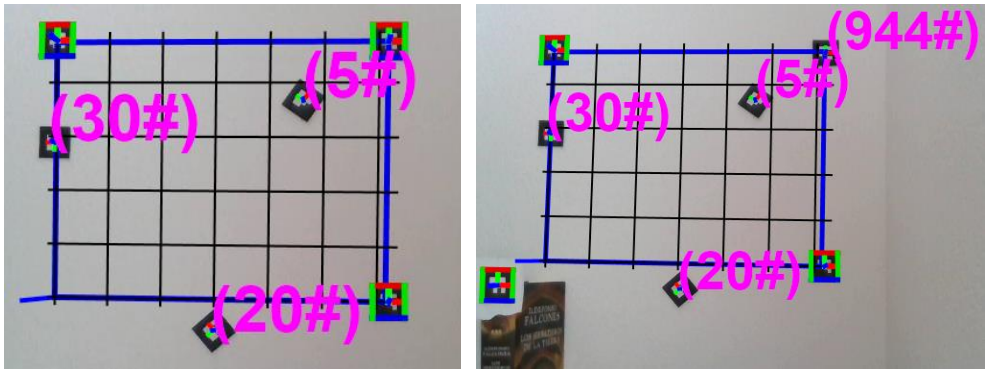


Figure 8.5: Figure of the two scenarios changing the plane of one reference fiducial. Notice how in the right picture the previous reference fiducial 944# is now considered as an object, as the new reference fiducial is in the opposite corner.

Table 8.5 shows the results of this first comparison.

		Reference Fid in Z = 0		Reference Fid in Z > 0	
Fid		(x, y, z) - [mm]	(ψ, θ, φ) - [rad]	(x, y, z) - [mm]	(ψ, θ, φ) - [rad]
5#	Mean	(369.6, 294.38, -16.24)	(6.247, 6.277, 0.955)	(370.25, 295.67, -27.68)	(0.046, 6.265, 0.973)
	Std Dev	(2.01, 2.79, **5.05**)	(0.003, 0.001, 0.007)	(2.24, 4.49, **2.57**)	(0.002, 0.005, 0.006)
20#	Mean	(238.53, -43.71, -2.89)	(0.027, 0.054, 2.286)	(245.18, -45.13, -24.09)	(0.013, 6.259, 2.302)
	Std Dev	(3.23, 3.35, **4.69**)	(0, 0.003, 0.007)	(3.04, 3.91, **2.55**)	(0.005, 0.002, 0.006)
30#	Mean	(-0.76, 234.59, -9.17)	(6.281, 6.239, 3.105)	(1.18, 229.31, -8.27)	(6.214, 6.219, 3.123)
	Std Dev	(2.11, 4.66, **5.57**)	(0.003, 0.002, 0.006)	(2.23, 3.06, **2.41**)	(0.004, 0.003, 0.006)

Table 8.5: Results of the first set-up, comparing the plane of the reference fiducials

From these values, one can see the following behavior:

- In the scenario with all the reference fiducials in the same plane, the uncertainties in X and Y are clearly lower than the uncertainties in Z.
- When moving a reference fiducial in a different plane, all uncertainties, in X, Y and Z are around the same values. There's no clear change between the X and Y uncertainties from one scenario to the other (in some cases they remain the same, in some cases they increase and in some they decrease). However, looking at the uncertainty in Z, one can see that it has now clearly decreased from before.
- The orientation uncertainties are extremely low, very close to zero. However, it can be noticed that the behavior on the increase or decrease of orientation uncertainties follows the same behavior as the uncertainties in translation. For this reason, the following analysis will only show the uncertainties in translation, since it's easier for the reader to get conclusions from it. Nevertheless, the orientation values can still be found in the Appendix B (Table B. 1 to Table B.16).

From this comparison, although it was not expected, a difference between the uncertainties obtained has been seen depending on the reference fiducials plane. For this reason, the following tests will continue for each case.

Measurement standard deviation

For this comparison, 4 experiments were run by combining both the changes of the reference fiducials' plane and the value of the input measurement standard deviation. The results are shown in the following Table 8.6.

		Reference Fid in Z = 0		**Reference Fid in Z > 0**	
Fid	(x, y, z)	High Std Dev	Low Std Dev	High Std Dev	Low Std Dev
5#	Mean	(369.26, 294.44, -15.08)	(368.47, 296.3, -10.56)	(369.88, 295.57, -36.38)	(370.1, 295.59, -31.71)
	Std Dev	(2.03, 2.8, 5.02)	(0.18, 0.41, 0.19)	(2.23, 4.36, 2.53)	(0.18, 0.41, 0.19)
20#	Mean	(238.64, -43.1, 4.91)	(242.77, -43.47, 11.83)	(244.76, -45, -28.62)	(244.53, -44.75, -23.41)
	Std Dev	(3.29, 3.34, 4.67)	(0.18, 0.41, 0.19)	(3.03, 3.8, 2.56)	(0.18, 0.41, 0.19)
30#	Mean	(-1.92, 235.06, -9.36)	(-1.7, 231.5, -1.18)	(0.92, 229.6, -8.51)	(1.6, 229.71, -6.02)
	Std Dev	(2.15, 4.68, 5.55)	(0.18, 0.41, 0.19)	(2.21, 3, 2.42)	(0.18, 0.41, 0.19)

Table 8.6: Translation uncertainties combining the combinations of std dev and reference fiducials plane

From these results, the following can be observed:

- Low measurement standard deviation significantly lowers the final uncertainties of the objects pose estimation. When working with low measurement standard deviation, the plane of the reference fiducial doesn't have an impact to the final uncertainties.
- In the case with high standard deviation, the results are, naturally, the same shown in the previous section. The impact of the reference fiducials' plane to the uncertainties of objects (and only in the Z axis) is only noticeable with a high input standard deviation.

Since there has been a clear difference between working with low or high input standard deviation, the following experiments will also be combined considering these two cases. However, when experimenting with low standard deviation, there will be no further need of trying both the reference fiducials in the same plane or in separate planes, since this doesn't have an impact. Therefore, when experimenting with low standard deviation, all the reference fiducials will be kept at the same plane (default situation).

Number of reference fiducials

The goal is to check whether the number of reference fiducials impact the final uncertainty of the objects. This will be observed in three situations, with low standard deviation

and the reference fiducials in the same plane, and with high standard deviation and changing the plane of the reference fiducials. Although high input uncertainty gives, as seen before, higher uncertainties, this experiment is interesting to be carried out because it may be easier to add reference fiducials than to lower the input uncertainty, and the aim is to see if the number of fiducials can reach the same level of uncertainty as the scenario with low input uncertainty.

As an example, the following Figure 8.6 shows the set-up for the case with 4 reference fiducials, all in the same plane.

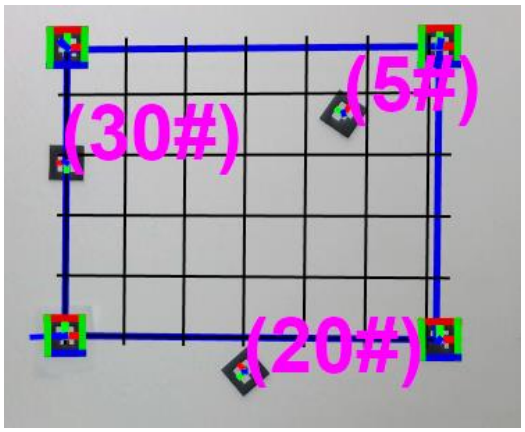


Figure 8.6: Real-world set-up using 4 reference fiducials in the same plane

Table 8.7, Table 8.8 and Table 8.9 show the results. The values in the case of 3 reference fiducials are the same as the obtained in the previous sections.

		Reference Fid in Z = 0		
		Low Std Dev		
Fid	(x, y, z) – [mm]	**2 Ref Fid**	**3 Ref Fid**	**4 Ref Fid**
5#	Mean	(370.15, 295.38, 0.45)	(368.47, 296.3, -10.56)	(370.5, 295.68, -10.02)
	Std. Dev.	(0.21, 0.36, 0.2)	(0.18, 0.41, 0.19)	(0.11, 0.1, 0.19)
20#	Mean	(243.42, -45.22, 1.65)	(242.77, -43.47, 11.83)	(244.41, -44.6, 0.43)
	Std. Dev.	(0.21, 0.36, 0.2)	(0.18, 0.41, 0.19)	(0.11, 0.1, 0.19)
30#	Mean	(4.69, 227.86, 5.16)	(-1.7, 231.5, -1.18)	(4.07, 228.87, 4.95)
	Std. Dev.	(0.21, 0.36, 0.2)	(0.18, 0.41, 0.19)	(0.11, 0.1, 0.19)

Table 8.7: Results comparing the number of fiducials with low std dev and the ref. fid. in the same plane

		Reference Fid in Z = 0		
		High Std Dev		
Fid	(x, y, z) - [mm]	**2 Ref Fid**	**3 Ref Fid**	**4 Ref Fid**
5#	Mean	(372.31, 294.4, -16.88)	(369.26, 294.44, -15.08)	(370.52, 296.37, -20.29)
	Std. Dev.	(2.06, 2.61, 5.11)	(2.03, 2.8, 5.02)	(1.89, 1.5, 4.49)
20#	Mean	(241.34, -44.6, -4.66)	(238.64, -43.1, 4.91)	(239.26, -42.99, -11.22)
	Std. Dev.	(3.42, 2.92, 4.75)	(3.29, 3.34, 4.67)	(2.85, 1.56, 4.38)
30#	Mean	(2.74, 233.39, -9.51)	(-1.92, 235.06, -9.36)	(2.01, 234.63, -7.28)
	Std. Dev.	(2.21, 3.96, 5.63)	(2.15, 4.68, 5.55)	(2, 2.14, 5.06)

Table 8.8: Results comparing the number of fiducials with high std dev and the ref. fid. in the same plane

		Reference Fid in Z > 0		
		High Std Dev		
Fid	(x, y, z) - [mm]	**2 Ref Fid**	**3 Ref Fid**	**4 Ref Fid**
5#	Mean	(367.69, 304.48, -30.19)	(369.88, 295.57, -36.38)	(370.35, 298.38, -19.74)
	Std. Dev.	(7.08, 4.25, 1.42)	(2.23, 4.36, 2.53)	(1.93, 1.57, 1.47)
20#	Mean	(246.32, -38.18, -34.16)	(244.76, -45.00, -28.62)	(239.86, -41.02, -23.52)
	Std. Dev.	(5.14, 3.71, 0.63)	(3.03, 3.80, 2.56)	(2.56, 1.69, 1.52)
30#	Mean	(-1.65, 233.67, -21.98)	(0.92, 229.60, -8.51)	(2.35, 235.41, -4.62)
	Std. Dev.	(6.63, 2.87, 2.84)	(2.21, 3, 2.42)	(1.96, 2.30, 2.02)

Table 8.9: Results comparing the number of fiducials with high std dev and the ref. fid. different planes

From these results, one can reason that there is an impact of the number of reference fiducials to the final uncertainties, and it will be checked if this impact is also dependent on the other parameters' set-up.

To make the observations more comprehensive, the following Table 8.10, has been generated, which gathers the means of uncertainties from all objects and the change achieved from 2 reference fiducials to 4 reference fiducials:

Uncertainty means		2 Ref Fid	4 Ref Fid	Change
Ref Fid in Z = 0 + Low Std. Dev.	x	0.21	0.11	48%
	y	0.36	0.10	72%
	z	0.20	0.19	5%
Ref Fid in Z = 0 + High Std. Dev.	x	2.50	2.21	12%
	y	3.11	1.71	45%
	z	5.15	4.63	10%
Ref Fid in Z > 0 + High Std. Dev.	x	6.23	2.13	66%
	y	3.56	1.83	49%
	z	1.36	1.65	-21%

Table 8.10: Percentage of uncertainties improvement in different cases

The following observations were made:

- The uncertainty slightly decreases with increasing number of reference fiducials. However, the differences for adding 1 reference fiducial are very low. In Table 8.10, it can be observed that, in all experiments, the improvements of uncertainty in X and Y are higher than the improvements in Z, which almost remains constant. So, adding reference fiducials in the case where all objects are in the plane Z = 0, only improves the uncertainty in the X and Y coordinates.

- The behavior is the same both in the cases of low and high standard deviation and in the cases changing the plane of the reference fiducials. Therefore, the number of fiducials is a parameter that has an impact on its own to the final uncertainty, it doesn't depend on other factors.

- Having more than 3 reference fiducials and high input standard deviation, still give uncertainties higher (always more than 1 mm) than the experiment with a smaller number of reference fiducials and low input standard deviation (always lower than 1 mm).

The analysis of objects' uncertainties in the scenario where all the objects are placed in the same plane has now been completed. For each test, the first conclusions have been unfolded, and they will be explained further in the conclusions.

The next section gathers the rest of the experiments in the second scenario, where the objects will be placed on different planes.

8.4.3.2. Experiment with objects on different planes

The set-up for initiating this experiment is the same as before:

- There are the default three reference fiducials in place (944# in the upper-right corner, 1234# in the upper-left corner, and 911# in the lower-right corner)
- The input standard deviation is in the high value (20 mm)
- The number of objects and their position in the grid is indifferent, though, in this case, at least one object will be placed either advanced in Z or rotated in X and Y, with the aim of having it define a different plane from the others.

However, there is one important point to remark before going through the results. In the past experiment, specifically in the first comparison (*reference fiducial's plane comparison*), the results obtained differed from what was expected. Let's remember that, in section 8.3.2 Engineering perspective, from 8.3 Proposed tests and hypothesis, it was stated that the plane of the reference fiducials wasn't expected to have an impact to the uncertainty of the estimated objects' poses. Only in the case where these objects were found in also a different plant, advanced in Z, one would find an advantage in adding reference fiducials in different planes, as it was thought that only those objects would improve their uncertainty of calculations.

Nonetheless, the results from the first experiment proved differently. It was found that the reference fiducials' plane impacted the uncertainty of all the objects (in that case, all the objects were at the same plane Z = 0). Yet, although it impacted all objects, it only had a strong impact to the uncertainty of one coordinate: the z coordinate.

The following experiments were thought with the aim of going into the effects to the uncertainties of having objects in different planes and, mostly, it was expected that having reference fiducials in different planes would make a big difference and, thus, it would be interesting to study all the combinations for that situation. However, it is now expected that everything follows the same pattern: the results comparing the reference fiducials' plane comparison should lead to the same conclusions and, consequently, so should the rest of the tests. For this reason, some tests will now be performed to confirm whether the behavior is the same.

Reference fiducials' plane comparison

The reference fiducial advanced in Z will be the same as the one used in the past experiment.

Figure 8.7 shows the set-up with the objects in different planes. One can notice how

Fiducial 5# is advanced in Z and rotated around the Z axis, and Fiducial 30# is also advanced in Z and rotated mostly around the X and the Z axis.

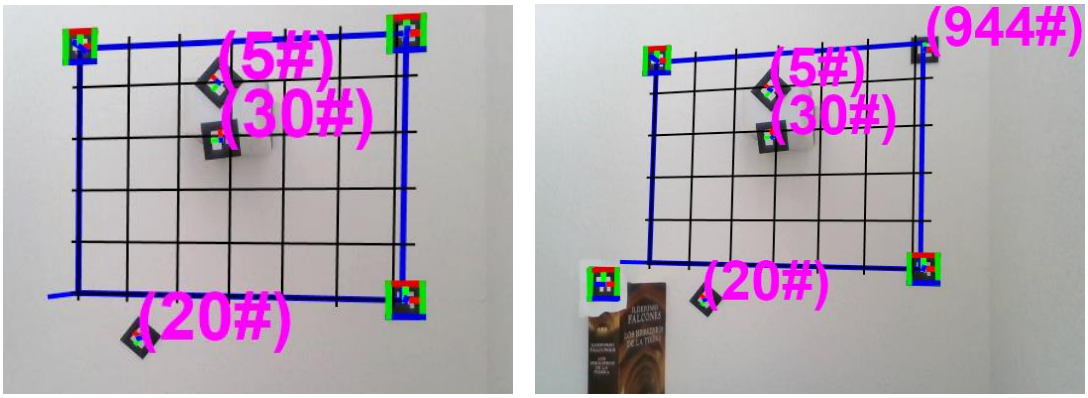


Figure 8.7: Figure of the first two scenarios tested with objects in different planes

Notice how, in Figure 8.7, the reference fiducials selected for each scenario are marked when not only the 3 axis are drawn, but also the sides of the fiducial and, additionally, they don't have the number of fiducial marked. The lower-left corner fiducial is the one advanced in Z, and supported with a book. All the fiducials that are not detected as reference fiducials are considered objects and, thus, they ave the number of the fiducial drawn in the image.

The results from these tests are shown in Table 8.11.

		Reference Fid in Z = 0		**Reference Fid in Z > 0**	
Fid		(x, y, z) - [mm]	(ψ, θ, φ) - [rad]	(x, y, z) - [mm]	(ψ, θ, φ) - [rad]
5#	Mean	(255.31, 283.26, 152.97)	(6.189, 6.27, 2.362)	(267.06, 272.28, 133.29)	(6.187, 6.158, 2.38)
	Std Dev	(1.83, 1.45, **4.64**)	(0.001, 0.002, 0.005)	(0.43, 1.16, **1.62**)	(0.001, 0.002, 0.005)
20#	Mean	(102.76, -60.36, 10.76)	(0.017, 0.004, 2.455)	(108.16, -62.14, -24.97)	(0.01, 6.174, 2.473)
	Std Dev	(3.04, 1.87, **4.57**)	(0.001, 0.002, 0.005)	(2.31, 1.63, **1.71**)	(0.001, 0.002, 0.005)
30#	Mean	(255.89, 212.72, 145.5)	(0.087, 6.059, 1.651)	(268.15, 202.58, 120.45)	(0.168, 5.968, 1.643)
	Std Dev	(1.97, 1.45, **4.57**)	(0.002, 0.001, 0.005)	(0.75, 1.17, **1.57**)	(0.002, 0.001, 0.006)

Table 8.11: Results with objects in different planes, comparing the plane of the reference fiducials

From these results, one can notice that:

- Having a reference fiducial in a different plane, significantly improves the uncertainty of the estimated coordinates in all fiducials, with independence of the plane they are placed in.

- Having a reference fiducial in a different plane equals the values of uncertainties of all coordinates to the same range of values, while by having all the reference fiducials on the same plane one can see lower standard deviations in X and Y compared to Z.

Measurement standard deviation

Having obtained very similar results as before, it is believed that the conclusions will be the same by having objects in different planes.

In the previous experiment, it was concluded that a low measurement standard deviation as an input would significantly decrease the uncertainty in all fiducials, for all coordinates. Moreover, the change was so significant that when running the code with low input standard deviation, having the reference fiducials on the same or in different planes didn't make any difference.

Therefore, the behavior is expected to be the same in this section, either having all the objects in the same plane or in different planes, having a low standard deviation will significantly decrease the final uncertainties of the objects' estimated poses.

To verify it, one test was performed, using the default three reference fiducials on the same plane and with low standard deviation. The results are shown in Table 8.12:

		Reference Fid in Z = 0 + Low Measurement Std. Dev	
		(x, y, z) - [mm]	(ψ, θ, φ) - [rad]
Fiducial 5#	Mean	(261.2, 280.84, 163.92)	(0.325, 6.062, 3.711)
	Std. Dev.	(0.15, 0.37, 0.18)	(0, 0, 0)
Fiducial 20#	Mean	(262.71, 209.04, 151.3)	(5.99, 6.281, 3.189)
	Std. Dev.	(0.15, 0.37, 0.18)	(0, 0, 0)
Fiducial 30#	Mean	(245.03, -29.03, 21.76)	(6.275, 0.088, 2.256)
	Std. Dev.	(0.15, 0.37, 0.18)	(0, 0, 0)

Table 8.12: Results of test with low standard deviation and objects in different planes

As expected, the input of low measurement standard deviation significantly reduced the uncertainties. Thus, the plane of the reference fiducials wouldn't have any impact in this scenario.

Regarding the uncertainty of the orientation coordinates, the actual values were decimals with until 8 zeros. These low values are because, with a low input uncertainty, the

likelihood becomes higher around the mean, and the chance of accepting those proposals is lower. Rotation proposals already represent a high change and, consequently, they are mostly rejected. When the input uncertainty is low, even more proposals are rejected.

Number of reference fiducials

From this same section in the last experiment, it was concluded that uncertainty decreases with increasing number of reference fiducials, and that this is not dependent on the measurement standard deviation nor the plane of the reference fiducials.

So, it was seen in the last part that adding additional reference fiducials only helped in slightly decreasing the standard deviation of all coordinates at the same pace approximately. However, even if adding additional reference fiducials, the uncertainty in the Z axis was still significantly higher that the uncertainty in X and Y and the only way for it to reach the same level was to add one of the reference fiducials in a different plane.

Hence, in this section we will perform a test to verify that this situation shows the same behavior in the case where we have objects in different planes. For that, a test will be performed with the objects in different planes and high measurement standard deviation. The parameters to change will be the number of reference fiducials and the plane of one of the reference fiducials. It is worth mentioning that, since the positions of the fiducials were changed, the test for 3 reference fiducials was also repeated.

The results of objects' uncertainties and means (for translation coordinates) are shown in both the following Table 8.13 and Table 8.14.

		Reference Fid in Z = 0		
Fid	(x, y, z)	**2 Ref Fid**	**3 Ref Fid**	**4 Ref Fid**
5#	Mean	(252.89, 265.19, 155.92)	(259.04, 275.88, 156.31)	(259.4, 277, 148.53)
	Std. Dev.	(1.99, 2.52, 5.37)	(1.91, 2.63, 4.76)	(1.78, 1.41, 4.63)
20#	Mean	(256.1, 191.6, 146.15)	(260.84, 202.45, 145.68)	(262.85, 201.97, 147.64)
	Std. Dev.	(2.17, 2.53, 5.23)	(1.91, 2.65, 4.72)	(1.88, 1.41, 4.55)
30#	Mean	(242.43, -28.33, -18.69)	(238.8, -25.74, -1.59)	(240.08, -28.09, -9.2)
	Std. Dev.	(3.4, 2.85, 4.87)	(2.05, 2.9, 4.64)	(2.74, 1.51, 4.37)

Table 8.13: Uncertainties of objects in different planes, by the number of ref. fid., all in the same plane

		Reference Fid in Z > 0		
Fid	**(x, y, z)**	**2 Ref Fid**	**3 Ref Fid**	**4 Ref Fid**
5#	Mean	(263.17, 268.09, 157.77)	(265.5, 271.63, 146.62)	(259.59, 274.49, 153.48)
	Std. Dev.	(3.37, 2.32, 2.06)	(1.95, 3.7, 2.46)	(1.93, 1.5, 1.57)
20#	Mean	(267.52, 194.47, 146.1)	(272.44, 195.15, 153.62)	(263.91, 199.2, 153.95)
	Std. Dev.	(3.27, 2.36, 1.81)	(1.92, 3.71, 2.42)	(1.93, 1.5, 1.53)
30#	Mean	(249.23, -25.98, -15.57)	(247.94, -30.58, -12.07)	(241.34, -28.03, -5.62)
	Std. Dev.	(3.4, 2.86, 1.2)	(2.88, 3.95, 2.53)	(2.5, 1.62, 1.49)

Table 8.14: Uncertainties of objects in different planes, by the number of ref. fid., one in a different plane

From these results, as expected, the same conclusions from the previous experiment are obtained:

- The uncertainty of the coordinate Z estimate is higher than the other coordinates. However, when adding an additional reference fiducial in Z > 0, this uncertainty can be decreased to the same range as the other coordinates.
- Adding and removing reference fiducials slightly decreases or increases the uncertainty of all objects and all coordinates.

Finally, 17 experiments combining parameters were needed to be performed. In Figure 8.8 one can see the combination of the parameters.

Figure 8.8: Experiments performed to investigate the calculation's accuracy

8.5. Conclusions

The aim of this analysis was to get practical suggestions to improve the accuracy of a robot's vision system. Hence, this analysis has been focused on obtaining a better precision in the object placements estimations. An object placement is defined by six coordinates, three for the translation and three for the rotation around the axis. Therefore, to achieve the goal of this analysis, several scenarios have been tested to investigate what is the range of uncertainties in the estimations and what factors have an effect, strong or weak, to the final uncertainties.

First, all the objects have been tested by placing them in the same plane, and by tuning three external parameters, some conclusions have been reached about whether these parameters have a strong or weak impact to the uncertainty, and how they can be modified to optimize it. These parameters are: the input measurement standard deviation, which defines the actual uncertainty of measurements in the real world, the number of reference fiducials, which are used to estimate the camera position, and the plane of the reference fiducials.

Second, the same tests have been made by adapting the scenario to a more common real-world situation, by placing some objects in a different height than the plane of the reference fiducials.

From this analysis, the most important observations have been gathered in Table 8.15, which compares the stated expectations from the beginning with the final conclusions:

	Expected impact to the final uncertainty of the objects	**Studied impact to the final uncertainty of the objects**
Plane of the objects	*No*	No
Plane of the reference fiducials	*Yes, but only to the objects that are in a different plane*	Yes, to all the objects but only significantly in the Z coordinate
Measurement standard deviation	*Yes, always*	Yes, always and with high impact
Number of reference fiducials	*Yes, always*	Yes, but with a very slight impact (for 1 extra fiducial)

Table 8.15: Comparison between expectations and results for the objects' uncertainty analysis

However, it is important to remark one factor: the robot's application. Depending on the actual robot's application, the desired uncertainty of the calculations may change. For some applications, a higher uncertainty may be accepted, while for others the uncertainty might need to be as low as possible.

On that account, from the test results and observations performed, a suggestion of different experimental set-ups for the robot which have a powerful impact to the uncertainty of the estimated objects' poses have been summarized in the following Table 8.16:

Uncertainty Goal	Measurement uncertainty	Number of reference fiducials	Plane of reference fiducials
< 0.2 mm	Low	4	Indifferent
Between 0.2 and 1 mm	Low	2 or 3	Indifferent
Between 1 and 2 mm	High	4	$Z \neq 0$
Between 2 and 5 mm	High	3	$Z \neq 0$
	High	4	All in the same plane
> 5 mm	High	2	Indifferent
	High	3	All in the same plane

Table 8.16: Proposal of different set-ups that get different object's final uncertainties

Considering the scope of this project, which focuses on collaborative robots used by a top-industry leading company such as ABB, it's important to remember that most of their applications are in the industrial environment and, thus, the operation is critical and the accepted uncertainty is, most of the time, a maximum of 1 mm. Furthermore, another aspect to consider is safety, if the robot is performing a critical task using a critical tool, which could be dangerous for the human, the uncertainty of these calculations is extremely important and should be as low as possible. Another application in this aspect would be in a medical environment, where not only safety is key but also the measurement of the used materials. Therefore, it is recommended that the applied measurement standard deviation is no more than 5 mm.

Nevertheless, sometimes the rate between 1 and 2 mm is also an acceptable uncertainty from some applications, for example, when measuring the ingredients to bake. For this reason, when having a high input uncertainty, it is highly recommended that 4 reference

fiducials are added in the working table, such that one of them is also on the Z > 0 plane. It has been proved that the Z uncertainty is usually the highest one, increasing then the global uncertainty of the object. Thus, by adding a new reference fiducial in Z > 0, it will be ensured that the uncertainty of the Z coordinate is as good as the uncertainties for the other coordinates.

Hence, if the applied measurement standard deviation is very high (such as 20 mm) and cannot be lowered, it can be compensated by having more than 3 reference fiducials and ensuring that one of them is in a plane with $Z \neq 0$.

To sum up, several possible set-ups have been suggested to help the user understand what inputs are needed for the robot to obtain the desired uncertainty in the position calculations of the objects. The desired final uncertainty will depend on the application, based on two main factors: the nature of the application (if it's based on assembling, baking, medical, etc.) and the safety (what critical tasks can it involve).

9. Intrinsic Parameters impact

9.1. Objective of the analysis

In this analysis, the goal is to include in our calculations the uncertainty of the camera intrinsic parameters. Specifically, the effect of the focal length to the uncertainty of the camera and objects will be explored.

The intrinsic parameters of the camera are a key component in order for the vision system of the robot to work. To obtain them, one as to perform a first step which is the calibration of the camera, using an external application from the *BoofCV* library. This first step is very human-dependent (since several pictures need to be taken while changing the position of a square coordinate grid) and very time-consuming.

Therefore, following the goal of suggesting practical improvements for a collaborative robot's vision system, this experimental analysis aims to investigate if the focal length can be estimated from the fiducials, as it has been done in the previous analysis, in Chapter 8 Improving the camera accuracy, with the extrinsic parameters.

9.2. Description of the analysis

The problem statement is defined such that the calibration of the camera will be made in the same step where the pose is estimated. From all the intrinsic camera parameters, the focal length will be studied.

Several explorations will be done around this, with the aim of understanding the effects of an unknown focal length in the fitting problem, and what is its effect to the position and orientation uncertainty of the objects. If, with these experiments, the focal length uncertainty is reasonable and its effect to the objects' uncertainty also, further research is encouraged to completely confirm the proper estimation of all the intrinsic parameters, leading to the possible removal of the initial calibration step.

The idea behind the tests is the following: in principal, with three fiducials the whole system can be solved and, thus, using the image seen by the camera we can get the intrinsic parameters of the camera without the need of performing an initial calibration. For that, different proposals of the focal length will be created, and the Metropolis algorithm will be applied to a mixed proposal made of the proposals of translation parameters, the proposals of orientation parameters and the proposals of the focal length. The likelihood evaluator remains the same as in the past experiments.

9.3. Proposed tests and hypothesis

The analysis is divided in three main areas:

1. Exploring the intrinsic uncertainty implications in the camera pose

The aim of this experiment is to make sure that the focal length estimations are correct and to investigate if this causes an effect to the camera pose uncertainties.

The camera parametrization has now been changed to include the focal length. Consequently, each camera sample will be defined not by 6, but by 8 parameters: 3 location coordinates, 3 orientation coordinates, and 2 focal length values. Since the camera was calibrated in advance to perform the past experiments, it is known that its focal length is 581 pixels both in x and in y.

Moreover, the camera poses estimation should be within the expected values by observing the real world. A comparison will be made between this case and the same case but maintaining the initial focal length constant (so, not using any proposals of focal length).

It is expected that the focal length estimation is correctly made and, thus, both the intrinsic (the focal length) and extrinsic (location and orientation) parameters estimations and their uncertainties are reasonable.

2. Exploring the intrinsic uncertainty implications in the objects' pose

The goal of this analysis is to investigate whether the estimation of focal lengths increases, decreases or doesn't impact the final uncertainties of the objects' poses.

For that, the experiments that were previously performed to obtain the objects' uncertainty are extended to include information about the uncertainty of the camera intrinsic parameters. Hence, not only the uncertainty of the camera extrinsic parameters is studied, but also the uncertainty of its intrinsic ones (such as the focal length), and how this will affect the objects final uncertainties.

Maintaining the same measurement standard deviation, the results by estimating the focal length and by keeping it with the initial value will be compared. It is expected that the uncertainties are similar for both cases, meaning that the estimation of focal length is working.

3. Exploring the intrinsic uncertainty implications in the vision's accuracy

The aim of this experiment is to investigate what should be the specific set-up for a cobot's working table in order to ensure that the intrinsic parameters are accurately estimated.

After exploring the behavior and effects of estimating the focal length to the uncertainties of our system, both camera and objects, if the results are satisfactory the next step is to experiment through different situations with different number of reference fiducials. Eventually concluding if, at some point, the objects' uncertainties equal those of maintaining the focal length constant. If this happens, it would be a first step towards confirming if camera calibration can be avoided.

For that, the experiment will be based on adding reference fiducials and observing the evolution of the objects' uncertainties. The objects observed will be placed such that two are in the grid, in Z = 0 (number 20# and 30#), and the third one will be in a plane advanced in Z (number 5#). The initial scenario will be using 2 reference fiducials in Z = 0. It is expected that more than three reference fiducials will be needed, but not how many.

9.4. Results of the analysis

9.4.1. Effect of focal length to the camera uncertainty

The camera pose was tested in two situations: one considering independence of the focal length, meaning that no initial calibration is needed, and the other one keeping it constant through all the iterations of the algorithm.

It is worth mentioning that the tests have been run with a measurement standard deviation of 5 mm. Regarding the focal length proposal standard deviation, it has been set up at 15 px because it gives a good enough amount of accepted samples. The results are summarized in Table 9.1.

	Mean		**StdDev**	
	Exploration of focal length	Focal length constant	Exploration of focal length	Focal length constant
x [mm]	423	421	4	4
y [mm]	48	17	4	11
z [mm]	1090	1102	15	7
$\boldsymbol{\psi}$ $[\boldsymbol{rad}]$	3.095	3.124	0.003	0.010
$\boldsymbol{\theta}$ $[\boldsymbol{rad}]$	6.228	6.226	0.003	0.004
$\boldsymbol{\varphi}$ $[\boldsymbol{rad}]$	0.011	0.008	0.007	0.005
fx [px]	**569**	**581**	**9**	**0**
fy [px]	**569**	**581**	**9**	**0**

Table 9.1: Camera parameters comparing between estimating the focal length or keeping it constant

For the case of *exploring focal length* (so, estimating the focal length without depending on the initial calibration), 1749 samples were accepted, from which 217 were proposals changing the focal length parameters. Regarding the other case, where the focal length is not changed and only obtained from the calibration file, the number of accepted samples was 1876. It is considered that both amounts are enough to have plausibility.

Two main observations are gathered from the results obtained:

- The focal length values estimated are very close to the ones obtained through the initial calibration of the camera, and their uncertainty is fair. Moreover, the other estimated parameters are also similar in both cases.

- Regarding the uncertainties of the other camera parameters (translation and orientation coordinates), these are within the same range as maintaining the focal length constant. Nevertheless, in the case of uncertainty in Z, one can notice that the uncertainty value is higher in the case where the focal length is estimated than in the case where the focal length is constant. This is because there is a high correlation between the focal length and the camera distance. Therefore, if one is loosened (in this case, the focal length), the camera distance would also loosen and increase uncertainty.

Therefore, it is verified that the focal length estimation is accurate, and it doesn't have a strong effect into the uncertainties of the rest of the camera pose estimation.

9.4.2. Effect of focal length to the objects' uncertainty

This experiment will be performed with an input measurement uncertainty of 20 mm to have higher values of uncertainties and facilitate the comparison. Moreover, three object fiducials are added in the scenario, two of them inside the Z = 0 plane and one of them advanced in Z. Below, the set-up is shown in Figure 9.1 and the results in Table 9.2.

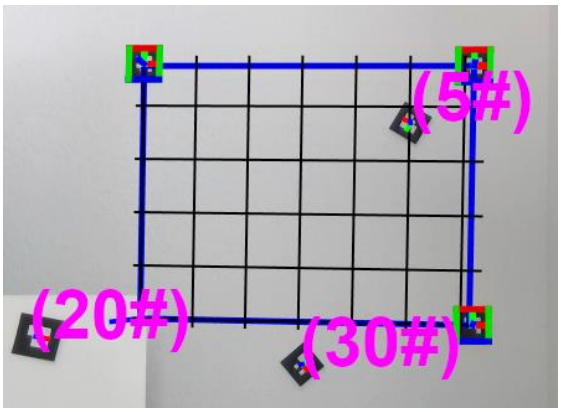


Figure 9.1: set-up of the objects' uncertainty study with the focal length

		Mean		StdDev	
		Exploration of focal length	Focal length constant	Exploration of focal length	Focal length constant
Fiducial 5#	x [mm]	403	402	4	2
	y [mm]	293	298	5	3
	z [mm]	-4	-21	15	5
	$\boldsymbol{\psi}\ [rad]$	0.059	6.269	0.045	0.001
	$\boldsymbol{\theta}\ [rad]$	0.053	0.012	0.022	0.003
	$\boldsymbol{\varphi}\ [rad]$	2.621	2.603	0.002	0.006
Fiducial 20#	x [mm]	-16	-14	4	3
	y [mm]	-12	-6	2	4
	z [mm]	259	262	15	5
	$\boldsymbol{\psi}\ [rad]$	6.045	6.119	0.038	0.002
	$\boldsymbol{\theta}\ [rad]$	0.341	0.236	0.044	0.003
	$\boldsymbol{\varphi}\ [rad]$	6.072	6.068	0.006	0.006
Fiducial 30#	x [mm]	250	245	3	4
	y [mm]	-61	-61	2	3
	z [mm]	5	0	16	5
	$\boldsymbol{\psi}\ [rad]$	6.247	5.917	0.018	0
	$\boldsymbol{\theta}\ [rad]$	0.084	0.095	0.061	0.003
	$\boldsymbol{\varphi}\ [rad]$	5.429	5.402	0.006	0.006

Table 9.2: Comparison of objects' uncertainties by estimating the focal length or maintaining it constant

From this experiment, one can grasp the same conclusions as in the experiment observing the camera pose.

- The estimations of the objects' poses are accurate both when estimating the focal length and when leaving it constant.
- The uncertainties of the poses are not highly affected by the estimation of the focal length, they are maintained within the same range as keeping the focal length constant.

- The uncertainty of the translation in Z is the only one where there's a visible change comparing the case of estimating the focal length or keeping it constant. As mentioned, when zooming the focal length, everything is scaled, while moving the camera through the Z axis scales the objects differently depending on how far away they are from the camera. Thus, a change in the focal length is strongly correlated with the distance of the camera: the more changes done in the focal length, the higher the uncertainty of the Z translation coordinate would be.

For the sake of completion, the same experiment has been performed using an input uncertainty of 5 mm instead of 20 mm.

In Table 9.3, the mean and the uncertainties of each coordinate are shown (considering all detected objects). The specific statistics (mean and uncertainties) for each fiducial can be found in the Appendix C (Table C. 1 and Table C. 2).

		StdDev / Uncertainty	
		Exploration of focal length	Focal length constant
Objects parameters	x [mm]	0.8	0.2
	y [mm]	0.4	0.4
	z [mm]	2.9	0.2
	$\boldsymbol{\psi}\ [rad]$	0.018	0
	$\boldsymbol{\theta}\ [rad]$	0.017	0
	$\boldsymbol{\varphi}\ [rad]$	0.001	0

Table 9.3: Uncertainties of all camera parameters comparing the two focal length experiments

As expected, the uncertainties are also similar, except for the uncertainty in the translation coordinate of Z, which increases because the focal length is more loosened also.

9.4.3. Effect of the focal length to the vision's system accuracy

As it has been mentioned in section 9.3 Proposed tests and hypothesis, in this experiment the focal length and the objects' pose uncertainties will be estimated in several scenarios. An example of the scenarios can be seen in Figure 9.2. In this case, as it can be appreciated, there are three reference fiducials in the plane Z = 0 and the objects are positioned such that: one is advanced in Z, one inside the grid, and one outside the grid (5#, 20# and 30# respectively), all of them rotated.

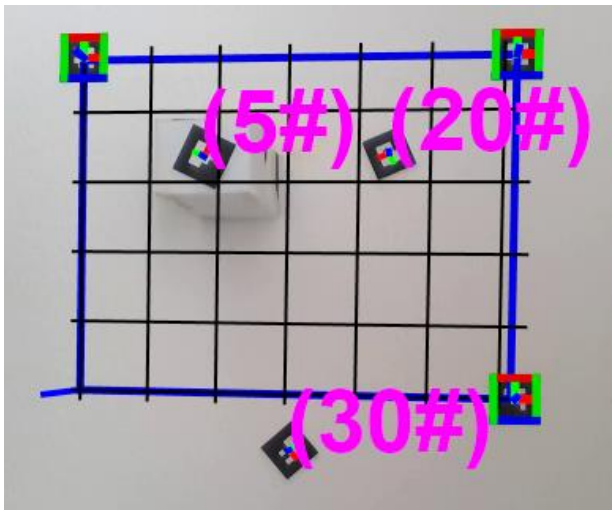


Figure 9.2: Example of real-world set-up for the focal length experiment

In this scenario, the code was run several times through changing the reference fiducials: by adding them and moving them forward in the plane Z test after test, until the last iteration which was 6 reference fiducials and with one of them in a plane Z > 0. These experiments were carried out with an input uncertainty of 20 mm with the aim of having higher scales to facilitate comparison. Figure 9.3 shows the set-up of the last test:

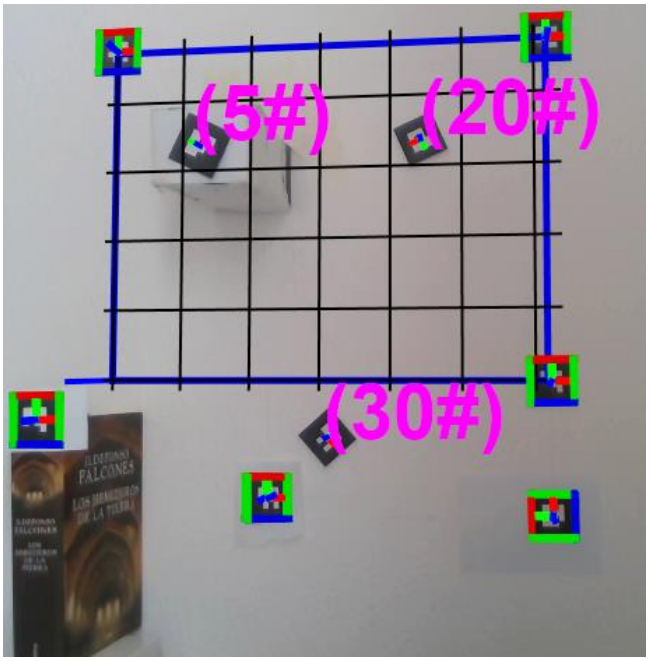


Figure 9.3: Example of real-world set-up for the focal length experiment

First, a quick look was made into the focal length values obtained in all experiments, to make sure they are as expected and that their uncertainties are plausible. For instance, Table 9.4 shows the focal length results of all scenarios:

		Mean [px]		Std. Dev. [px]	
		fx	fy	fx	fy
All in Z = 0	2 Ref Fid	594	511	98	37
	3 Ref Fid	636	676	75	90
	4 Ref Fid	531	524	35	36
	5 Ref Fid	702	709	30	29
	6 Ref Fid	609	604	38	35
One in Z > 0	2 Ref Fid	579	390	63	44
	3 Ref Fid	654	576	34	29
	4 Ref Fid	731	704	31	33
	5 Ref Fid	681	674	26	28
	6 Ref Fid	584	583	29	29

Table 9.4: Focal length mean and uncertainties for each scenario, with 20mm input uncertainty.

As observed, all means of the focal length are within the same range. Note that the standard deviations are higher than Table 9.1 because of the input uncertainty.

Hereinafter, the objects' uncertainties will be checked for each scenario, with the aim of seeing their evolution through the scenarios. Each scenario will be different in terms of the number and position of the reference fiducials.

First, with the aim of having higher magnitude uncertainty values (in the objects) and facilitate to the reader the comparison between scenarios, it was decided to enter a high input uncertainty, 20 mm, because, when then the distribution is broader and the overall likelihood lower. The algorithm accepts more proposals, causing the uncertainty to go up. The results, of this experiment will be given below.

In Table 9.5, all the uncertainty results obtained by each fiducial is gathered. Additionally, the complete output of the code can be found in the Appendix C (Table C. 3 to Table C. 12).

First of all, it's important to remark that in all the results, the uncertainty in Z coordinate is in general always higher that the uncertainties is X and Y: this is expected, and has been explained why in the past section 9.4.2 Effect of focal length to the objects' uncertainty.

		All reference fiducials in Z = 0					One reference fiducial in Z > 0				
		2 Ref Fid	3 Ref Fid	4 Ref Fid	5 Ref Fid	6 Ref Fid	2 Ref Fid	3 Ref Fid	4 Ref Fid	5 Ref Fid	6 Ref Fid
5#	x	4.81	4.12	1.56	1.54	1.53	4.69	2.69	0.67	0.93	0.87
	y	1.58	1.82	1.55	1.56	1.56	0.67	0.84	1.16	0.54	0.56
	z	7.87	7.09	4.94	4.93	4.94	17.66	14.71	5.02	4.70	4.82
	ψ	0.03	0.03	0.01	0.01	0.01	0.03	0.02	0.01	0.01	0.01
	θ	0.03	0.04	0.00	0.00	0.00	0.04	0.02	0.01	0.01	0.01
	φ	0.00	0.01	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00
20 #	x	1.18	1.06	1.39	1.36	1.35	1.73	2.17	0.88	0.74	0.79
	y	3.62	4.46	1.71	1.68	1.69	3.24	2.40	1.63	1.49	1.61
	z	4.55	7.54	5.45	5.45	5.46	13.69	17.70	9.14	8.84	9.13
	ψ	0.03	0.05	0.01	0.01	0.01	0.04	0.02	0.01	0.01	0.01
	θ	0.07	0.04	0.01	0.01	0.01	0.03	0.02	0.01	0.01	0.01
	φ	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00
30 #	x	3.38	2.67	1.51	1.47	1.46	3.54	2.88	2.30	0.47	0.45
	y	1.37	1.51	1.53	1.54	1.55	1.59	0.69	0.52	0.51	0.52
	z	10.01	8.23	5.82	5.82	5.82	20.00	17.49	6.31	5.70	5.76
	ψ	0.04	0.04	0.01	0.01	0.01	0.03	0.02	0.01	0.01	0.01
	θ	0.04	0.08	0.01	0.01	0.01	0.08	0.04	0.03	0.02	0.03
	φ	0.00	0.01	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00

Table 9.5: Results of the focal length experiments to reduce the calibration step

In the following Table 9.6, the information has been simplified to show the mean of the uncertainties per each parameter (considering the three objects).

The mean of the standard deviations is obtained by calculating the Root Mean Square (RMS) over all the objects.

	Total uncertainty for all fiducials in the scenario									
	All reference fiducials in Z = 0					One reference fiducial in Z > 0				
	2 Ref Fid	**3 Ref Fid**	**4 Ref Fid**	**5 Ref Fid**	**6 Ref Fid**	**2 Ref Fid**	**3 Ref Fid**	**4 Ref Fid**	**5 Ref Fid**	**6 Ref Fid**
x [mm]	2.7	2.3	1.5	1.5	1.4	3.1	2.6	1.1	0.7	0.7
y [mm]	2.0	2.3	1.6	1.6	1.6	1.5	1.1	1.0	0.7	0.8
z [mm]	7.1	7.6	5.4	5.4	5.4	16.9	16.6	6.6	6.2	6.3
ψ [rad]	0.03	0.04	0.01	0.01	0.01	0.03	0.02	0.01	0.01	0.01
θ [rad]	0.04	0.05	0.01	0.01	0.01	0.05	0.02	0.02	0.02	0.01
φ [rad]	0.00	0.01	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00

Table 9.6: Evolution of total uncertainties in the focal length experiment

Looking at Table 9.6, the information is now easier to be interpreted. Nevertheless, both tables lead to the same observations:

- The uncertainties decrease with increasing number of fiducials. In general, for increasing number of fiducials, there's a decreasing value in uncertainties, which is what was expected. This behavior can be also noticed in the following Figure 9.4:

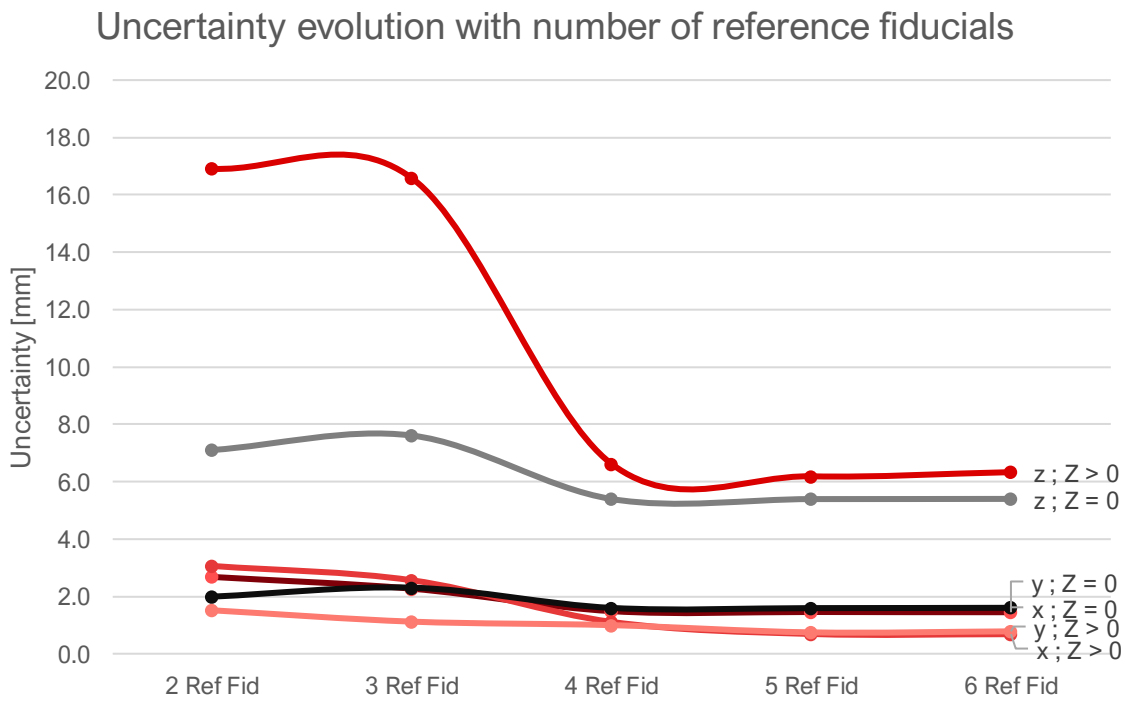


Figure 9.4: Evolution of uncertainty exploring the focal length and changing the reference fiducials' pose

- However, this decrease is not always proportional for each added fiducial. From 4 fiducials on, the uncertainties keep decreasing but not significantly. The values are almost maintained constant even while adding reference fiducials.

Let's also remember that the experiment was done in a home environment and, as a consequence, the precision of the reference fiducials placement may not be so accurate as in a laboratory. With the aim of checking if this fact was a big influence, it was experimented with the pose estimation algorithm. The current estimation algorithm does the average of the camera position estimated for each reference fiducial and, if one is bad placed, it can have a big impact. Thus, it was decided to test the set-up with another estimation algorithm, such as the RANSAC algorithm (random sample consensus), which works better in the presence of conflicting information. The experiment for 20 mm was repeated, now using the RANSAC algorithm for the camera pose, only in the case of having 6 reference fiducials and one of them at a plane Z > 0. The results are shown in the Table 9.7 and the output is in Appendix C (Table C. 13). The values for the same experiment with the average algorithm have also been added for comparison purposes, they are taken from Table 9.6.

	Total uncertainties for all objects found	
	RANSAC algorithm	**Average estimation algorithm**
x [mm]	0.7	0.7
y [mm]	0.7	0.8
z [mm]	6.3	6.3
ψ [rad]	0.01	0.01
θ [rad]	0.02	0.01
φ [rad]	0.00	0.00

Table 9.7: Results with 6 Ref Fid, one in Z > 0. 20 using different estimation algorithms

As it can be appreciated, in this case the uncertainties are almost the same. Since the RANSAC algorithm would improve results in the face of misplacement (outliers), the obtained values mean that, even with a home set-up, there's almost no misplacement in the reference fiducials.

To conclude this experiment, the uncertainty results with estimating the focal length need to be compared with the results with maintaining the focal length constant.

The uncertainty values without estimating the focal length were already gathered in the past experiments (in section 8.4.3.2 Experiment with objects on different planes from Chapter 8 Improving the camera accuracy).

The following comparison will be made between: estimating the focal and not exploring it with an input uncertainty of 20 mm. The values from the tables Table 8.8 and Table 8.9 were used. From these values, the RMS of the standard deviations was taken and, as a result, the comparison in Table 9.8 is obtained.

	All reference Fiducials in Z = 0					**One reference fiducial in Z > 0**				
	2 Ref Fid	3 Ref Fid	4 Ref Fid	5 Ref Fid	6 Ref Fid	2 Ref Fid	3 Ref Fid	4 Ref Fid	5 Ref Fid	6 Ref Fid
	Exploring focal length.					Exploring focal length.				
x [mm]	2.7	2.3	**1.5**	**1.5**	**1.4**	3.1	2.6	1.1	0.7	0.7
y [mm]	2.0	2.3	**1.6**	**1.6**	**1.6**	1.5	1.1	1.0	0.7	0.8
z [mm]	7.1	7.6	**5.4**	**5.4**	**5.4**	**16.9**	**16.6**	**6.6**	**6.2**	**6.3**
	Focal length constant.					Focal length constant.				
x [mm]	2.4	2.0	2.1	-	-	3.3	2.2	2.1	-	-
y [mm]	2.6	2.7	1.4	-	-	2.5	3.8	1.5	-	-
z [mm]	5.2	4.7	4.5	-	-	**1.6**	**2.5**	**1.5**	-	-

Table 9.8: Comparison of objects' uncertainties between estimating the focal length with 5 mm of uncertainty or keeping it constant with 20 mm of uncertainty

As it can be appreciated in Table 9.8, for both scenarios (regarding the reference fiducials' pose), the uncertainty of the objects when exploring the focal length is, in general, always higher than when keeping it constant. However, from 4 fiducials on one can see that the X and Y uncertainties start to be closer to the uncertainties without exploring the focal length, although the uncertainty in Z is still significantly higher. So, with more than 4 fiducials and exploring the focal length, it seems that similar ranges can be obtained. Although it is not a clear improvement in uncertainty, because the one in Z is still higher. From 4 fiducials on, adding an extra reference fiducial doesn't significantly improve the uncertainty.

Nevertheless, there is one important observation to point out: in all cases, when exploring the focal length, the uncertainty in the Z axis is higher than the X and Y uncertainty. This fact causes that, in the case where one reference fiducial is in Z > 0, even if the uncertainties by estimating the focal length were similar than those keeping it constant, the one in the Z axis will always be higher, since this is a direct consequence of the focal length estimation (explained in section 9.4.2 Effect of focal length to the objects' uncertainty). So, while exploring the focal length one would never get the same uncertainty results as when not exploring it and having one reference fiducial in Z > 0.

From Table 9.3 and Table 9.2, it is known that the same behavior is seen with an input uncertainty of 5 mm or with 20 mm: the uncertainties exploring the focal length are higher than when not exploring them and even more in Z. Thus, with 5 mm of input uncertainty, it is expected that one would reach the same conclusions when doing the experiment of adding reference fiducials.

9.5. Conclusions

The aim of this experiment was to look further into the intrinsic parameters of the camera, specifically the focal length, and to understand how their estimation could have an effect to the uncertainty calculations of the camera pose and objects poses. The reason behind it was that, if all intrinsic parameters of the camera could be estimated, then one wouldn't need to perform the calibration of the camera, which is a mandatory initial step that is very human dependent and time consuming.

After planning the analysis and obtaining the results, the conclusions are the following:

- The focal length estimation can be done, and the mean and standard deviation obtained are plausible with the real values obtained through calibration.

- When estimating the focal length, the uncertainty values of the Z translation coordinate are higher than in the case where the focal length from the calibration is maintained. This is because there is a high correlation between the focal length and the camera distance and loosening one causes a higher uncertainty in the other. This means that, in the case where one reference fiducial is in $Z > 0$, the uncertainty values by exploring the focal length would never reach the same range as by maintaining it constant.

- With the current data, it is thought that any final uncertainty obtained by exploring the focal length will likely be higher than by keeping the focal length constant. However, with more than 4 reference fiducials and all in $Z = 0$, the uncertainties seem to reach more similar values, although from 4 fiducials on the values remain almost constant. Nevertheless, more experiments should be done while increasing the number of reference fiducials.

The initial thoughts, thus, are that if one wants to achieve the same uncertainty when exploring or keeping constant the focal length, probably a minimum of 6 reference fiducials should be added in the scenario, and all of them in $Z = 0$. With this number of reference fiducials, the results were closer to the results with 3 or 4 reference fiducials and keeping the focal length constant.

Moreover, if there's a reference fiducial in $Z > 0$, the uncertainty in Z doesn't seem to

reach the same level when exploring the focal length as when keeping the focal length constant, because the exploration significantly increases the uncertainty in the Z axis.

To finish, we can summarize the following general conclusions:

- If all reference fiducials are in the same plane → with exploring the focal length, the increasing of reference fiducials (to 5 or 6 approximately) can reach a similar uncertainty with not exploring the focal length and a smaller number of reference fiducials.

- If one reference fiducial is in a different plane → the uncertainty in Z when exploring the focal length will always be higher than the uncertainty when keeping the focal length constant.

One could try to estimate the rest of the intrinsic parameters and observe the final uncertainties. With this, a new step would be taken towards the understanding of the camera calibration impact.

10. Project impact

Any project has an impact to society and the world, and it's very important to understand and study in detail this impact before implementing the project. The impact of this project is presented in two perspectives: environmental and socioeconomic.

10.1. Environmental impact

As mentioned in the section 1.3 Project scope, from the Introduction, this project was conducted at home due to the consequences of COVID-19. As a result, the only material used for the project has been the fiducials (which are printer in paper), the camera, and the computer. When the restrictions of going to the office will be lifted, applying this project to a specific cobot in Future Labs will only suppose the extra energy required for the robot to work, which would not be different from its daily impact due to the fact that the robot will also be used for the rest of the current projects, since it's in the research lab.

Therefore, this project has almost null impact to the environment. However, it can be important to remark that, when the life of the computer or the camera comes to an end, the residues must be well taken care of because most of their components can be reused. When some of the components are unable to be used again it's important to bring them back to the original factory or a green point, where they can be transformed and used again for another purpose.

10.2. Socioeconomic impact

This project was a first step to a large process of improving the vision system accuracy of a collaborative robot. If the suggestions of different set-ups to obtain specific final uncertainties are followed, it is expected that the efficiency of the places where this is followed will improve. Moreover, the conclusions regarding the impact of calibration to the final uncertainties show that there is potential in estimating all the intrinsic parameters instead of getting them from the calibration, consequently meaning that if this is investigated further, it could lead to important improvements on the calibration requisite for using the robot.

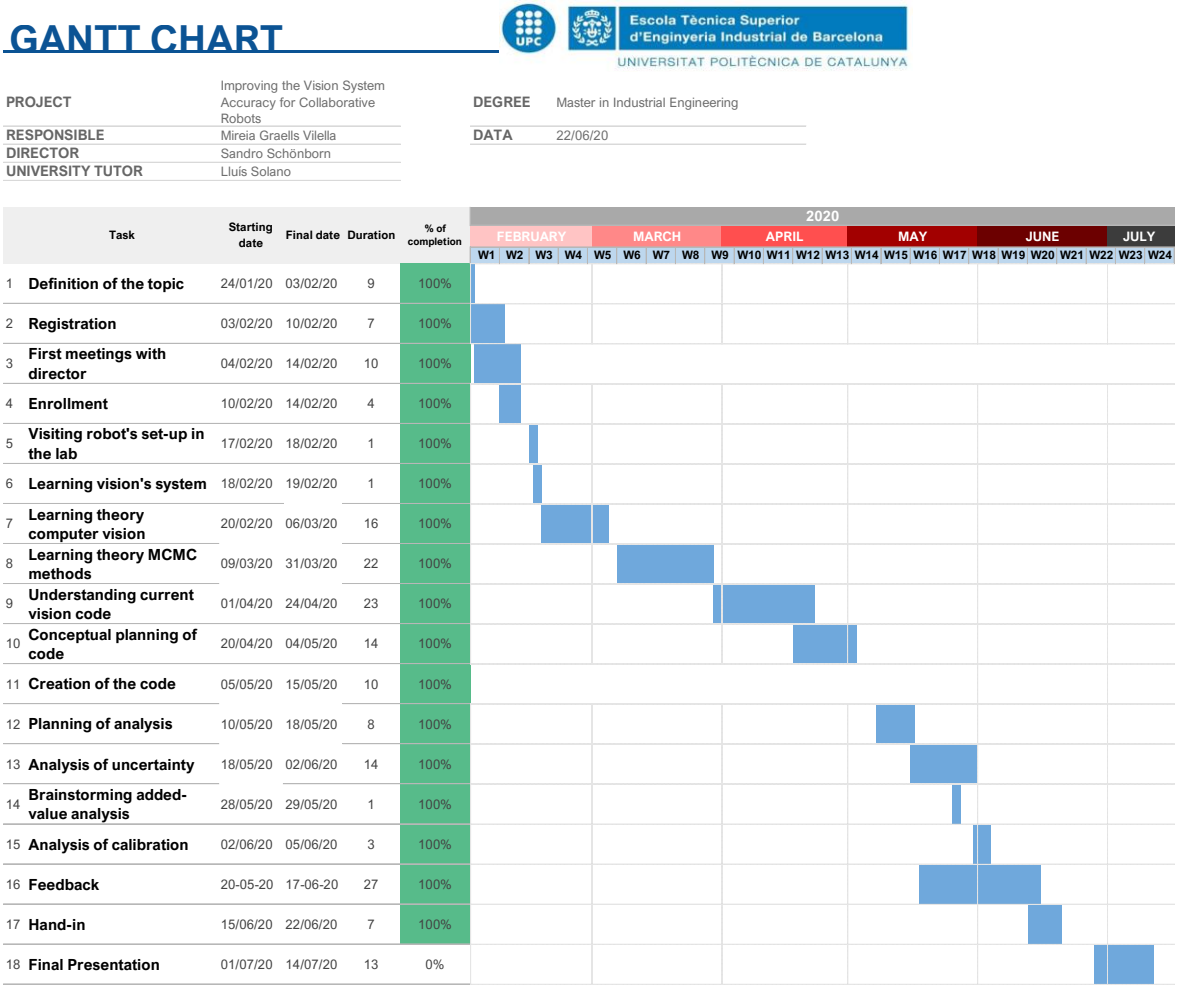
Therefore, this project is one more step to a higher efficiency and reliability of industrial collaborative robots and, as a result, it could bring a strong socioeconomic impact in the future.

Nevertheless, it is important to remember that this is a research project, with the aim of investigating, exploring and learning about the vision system of a collaborative robot. Further research has been encouraged and, thus, the more advances are made the higher will be the socioeconomic impact of these discoveries in the industry.

11. Planning & Budget

Before going through the final conclusions and suggestions of further research, it is important to describe how the planning of this project has unfolded itself and what are the billing costs.

The project development is represented in the Figure 11.1 below, through a Gantt chart:



	Task	Starting date	Final date	Duration	% of completion
1	Definition of the topic	24/01/20	03/02/20	9	100%
2	Registration	03/02/20	10/02/20	7	100%
3	First meetings with director	04/02/20	14/02/20	10	100%
4	Enrollment	10/02/20	14/02/20	4	100%
5	Visiting robot's set-up in the lab	17/02/20	18/02/20	1	100%
6	Learning vision's system	18/02/20	19/02/20	1	100%
7	Learning theory computer vision	20/02/20	06/03/20	16	100%
8	Learning theory MCMC methods	09/03/20	31/03/20	22	100%
9	Understanding current vision code	01/04/20	24/04/20	23	100%
10	Conceptual planning of code	20/04/20	04/05/20	14	100%
11	Creation of the code	05/05/20	15/05/20	10	100%
12	Planning of analysis	10/05/20	18/05/20	8	100%
13	Analysis of uncertainty	18/05/20	02/06/20	14	100%
14	Brainstorming added-value analysis	28/05/20	29/05/20	1	100%
15	Analysis of calibration	02/06/20	05/06/20	3	100%
16	Feedback	20-05-20	17-06-20	27	100%
17	Hand-in	15/06/20	22/06/20	7	100%
18	Final Presentation	01/07/20	14/07/20	13	0%

Figure 11.1: Gantt chart of the project

Regarding the billing costs of this project, they have been elaborated considering in two sections:

- Human costs: the people who contributed to this project are the Dr. in computer science who, as a senior scientist, developed the previous vision code of the cobot and directed this project, and the engineering student who developed the project.

- Material costs: the cost of the computer and camera used for this project.

Regarding the human costs, the engineer cost is calculated taking into account the hours dedicated to this study. The Master Thesis equals to 12 ECTS credits, which, represents a total of 240h. The standard salary of a junior engineer in Spain is, approximately, 28 000 € per year, 14 € / hour. Furthermore, there has been a continuous supervision of the thesis director, approximately dedicating 2-3h per week, which makes a total of 20h. As a senior scientist the general salary in Spain is around 30 € / hour. Finally, it is estimated that the senior scientist dedicated around 100h to develop the previous vision code.

Regarding the material costs, since they have only been used temporarily, one needs to consider the amortization. The following Table 11.1 and Table 11.2 quantify the costs:

Human costs	**Hours**	**Salary [€ / h]**	**Cost [€]**
Engineering student	240	14	**3360**
Senior Scientist – supervision	20	30	**600**
Senior Scientist – previous vision code	100	30	**3000**

Table 11.1: Human costs of the project

Material costs	**Price [€]**	**Payback [years]**	**Yearly payback [€/year]**	**Project duration [years]**	**Accumulated amortization**
Laptop	**1160** [15]	**4**	**290**	**0.2**	**58**
Camera	**180** [14]	**3**	**60**	**0.2**	**12**

Table 11.2: Material costs of the project

To conclude, the total cost of this project, considering the direct cost of development and the previous costs, has been **7 030 €**, being the human cost significantly higher than the material cost.

Conclusions and further research

The main goal of this project was to explore in depth the vision system of a collaborative robot with the purpose of distinguishing the main aspects that could improve the vision system's accuracy and, consequently, propose ways on how to achieve them.

Therefore, in this project, instead of obtaining the optimal position of an object, the code has been adapted so that it applies a Markov chain Monte Carlo (MCMC) method to generate stochastic samples for each object and return a representative distribution of the position.

After successfully applying the Metropolis algorithm (the MCMC method chosen) to the code, the evolution of the project has been mainly focused on two areas: observing the uncertainty of the estimations obtained for detected objects and investigating the importance of the camera calibration step.

From the first part, we have concluded that there are important external parameters that strongly affect the final uncertainty of the results. These parameters must be considered and tuned in the proper way in order to obtain the wanted uncertainty, which will depend on the specific application of the robot. The most important suggestions are that:

- In order to ensure a final uncertainty of calculations of less than 1 mm, which is typically what is expected in an industrial setting, the operator must make sure that the precision of the measurements in the worktable of the robot is no more than 5 mm. If this precision is not met, the final uncertainties won't be able to be less than 1 mm.

- If the uncertainties are accepted to be between 1 and 2 mm, then the operator has more flexibility: the input uncertainty of the measurement can be higher than 5 mm. However, there must be, at least, 4 reference fiducials in the worktable and, additionally, one of them must be placed in a different plane than the other, since the experiments results have shown a great improvement in the uncertainty of the Z coordinates if this is done.

- Supplementary set-ups have been suggested to prepare the operator in case the final uncertainties could be higher. Although final uncertainties of 2 mm are not common, the set-ups can be found in Table 8.16 in case it should be necessary.

From the second part, we have concluded that the camera calibration has a high impact to the uncertainties of the objects' estimations. We have seen that, in most of the cases, the results with the initial calibration were significantly better than the results where the focal length was estimated. This is because estimating the focal length caused an increase in the uncertainty in the translation coordinate on the Z axis. Nevertheless, it was concluded that, if

more than 4 or 5 reference fiducials are added, then it seems that the estimation of the focal length give the same results as not estimating it but with less reference fiducials, such as 3 or 4. This leads us to believe that, if further research is done in this area, and, for example, all of the intrinsic parameters are estimated, a bigger step will be taken into the possible complete removal of the calibration step.

To finish, some specific suggestions and further research are proposed in order to continue improving the vision system of collaborative robots.

First of all, the calculation of the uncertainty of the fiducials can be improved. Currently, the likelihood is obtained by comparing the positions of the 4 corners, independently of each other, and in each corner a specific uncertainty is assumed. However, these 4 corners never move independently, they always move and rotate together. Thus, the uncertainty could be calculated for the 4 corners at once, considering their relation. This would be more realistic.

Second of all, another improvement of this project would be to analyze an uncertainty that joined the translation and the rotation uncertainty. This would be interesting because, due to the low orientation values, the analysis has mostly been focused on translation uncertainties. There are specific coordinates in the vision code that are obtained both from translation and rotation, such as the gripping points of the robot's wand, and, thus, they have the information of both. Working with the uncertainty of the gripping points would also give more strength to the analysis performed.

Finally, the focal length experiments have been introduced as a first step of a large process that could lead to the removal of the initial calibration step of the camera. As such, it is highly encouraged that the experiment is continued with three paths: first, keep adding more reference fiducials and studying the final uncertainties; second, estimating not only the focal length, but also the rest of the intrinsic parameters; and, last but not least, perform the experiment starting with a random initial value for the focal length, instead of taking it from the calibration. By experimenting these paths, we would be able to conclude if avoiding the calibration step can be done.

Acknowledgments

I would like to express my sincere gratitude towards all the people who has given me support during the development of this research project.

First, I would like to specially thank Dr. Sandro Schönborn for introducing me to computer vision and to the challenges of the statistics behind the improvements of a robot's vision system. Thank you for your trainings, your guidance, and for your help through all the obstacles we've encountered. Thank you also to all the ABB team, for giving me the opportunity of working with such an impressive team of researchers.

I would also like to thank Lluís Solano, for the tutoring of the project. Thank you for supporting the subject and helping me improve and complete this report.

Finally, I would like to thank my family and friends, for providing me with their support and motivation not only throughout this project, but also throughout my whole university career.

References

[1] ABB Asea Brown Boveri Ltd. *ABB Robotics Historical milestones.* Switzerland, 2020.

[https://new.abb.com/products/robotics/about-us/historical-milestones, 4th May 2020]

[2] ABB Asea Brown Boveri Ltd. *ABB Robotics – Collaborative Robots*. Switzerland, 2020.

[https://new.abb.com/products/robotics/collaborative-robots, 4th May 2020]

[3] FABBRI, M. *How a pinhole camera works*. Alternative Photography. 2014.

[http://www.alternativephotography.com/how-a-pinhole-camera-works/, 5th May 2020]

[4] LACHKAR, M., ALEXANDER, A. *Scala Book. Prelude: A taste of Scala*. 2020.

[https://docs.scala-lang.org/overviews/scala-book/prelude-taste-of-scala.html, 1st June 2020]

[5] UNIVERSITY OF BASEL. GRAVIS GROUP. *Scalismo - Scalable Image Analysis and Shape Modelling*. Basel, 2016.

[https://scalismo.org/, 1st June 2020]

[6] HARTLEY, R., ZISSERMAN, A. *Multiple View Geometry in computer vision*. United Kingdom: University of Cambridge, 2003. p. 1-19

[7] BIRCHFIELD, S. *An Introduction to Projective Geometry (for computer vision)*, California: Stanford University, 1998.

[8] HU, Z. *Projective Geometry*. Institute of Automation, Chinese Academy of Sciences. 2016. [Figure].

[9] SCHÖNBORN, S., EGGER, B., MOREL-FORSTER, A., VETTER, T. *Markov chain Monte Carlo for Automated Face Image Analysis*. International Journal of Computer Vision. 2016.

[10] VALPOLA, H. *Bayesian Ensemble Learning for Nonlinear Factor Analysis*. Helsinki University of Technology, 2000. [Figure].

[11] SERFOZO, R. *Basics of Applied Stochastic Processes*. Atlanta: Georgia Institute of Technology, 2009.

[12] LEE, P.M. *Bayesian Statistics: An Introduction*. West Sussex, United Kingdom, 2012. p.317 – 322

[13] SLABAUGH, G. *Computing Euler angles from a rotation matrix.* United Kingdom, 2017.

[14] Intel RealSense Technology. *Intel RealSense Depth Camera D435*.

[https://store.intelrealsense.com/buy-intel-realsense-depth-camera-d435.html, 8th June 2020]

[15] Lenovo. *ThinkPad X Series*.

[https://www.lenovo.com/ch/en/laptops/thinkpad/x-series/c/thinkpadx. 8th June 2020]