




# Automatic identification and validation of buildings' structure dimensions

# Final Report

REVISION HISTORY AND APPROVAL RECORD

| Revision | Date | Purpose |
|---|---|---|
| 0 | 17/04/2023 | Document creation |
| 1 | 20/04/2023 | Document task distribution |
| 2 | 24/04/2023 | Document update |
| 3 | 27/04/2023 | Document update |
| 4 | 02/05/2023 | Document update |
| 5 | 04/05/2023 | Document update |
| 6 | 08/05/2023 | Document update |
| 7 | 11/05/2023 | Document update |
| 8 | 18/05/2023 | Document revision |

DOCUMENT DISTRIBUTION LIST

| Name | E-mail |
|---|---|
| Victor Castellón Martín | victor.castellon@estudiantat.upc.edu |
| Xavier Lorenzo Lobato | xavier.lorenzo@estudiantat.upc.edu |
| Marc Lucena González | marc.lucena@estudiantat.upc.edu |
| Isi Bardají Bofill | isi.bardaji@estudiantat.upc.edu |
| Alejandro Amat Payá | alejandro.amat@estudiantat.upc.edu |
| Elias Ismael Estévez Macas | elias.ismael.estevez@estudiantat.upc.edu |
| Esteve Tomasa Costa | esteve.tomasa@estudiantat.upc.edu |
| Felip Martinez Quiros | felip.martinez@estudiantat.upc.edu |
| Llorenç Sastre Galmés | llorenc.sastre.galmes@estudiantat.upc.edu |
| Óscar Díaz Almendros | oscar.diaz@estudiantat.upc.edu |
| Pablo Diaz Soley | pablo.diaz.soley@estudiantat.upc.edu |
| Jaume Comellas Colome | jaume.comellas@upc.edu |
| Anna Umbert Juliana | anna.umbert@upc.edu |
| Ramon Viedma | ramon.viedma@hp.com |

| WRITTEN BY: | | REVIEWED AND APPROVED BY: | |
|---|---|---|---|
| Xavier Lorenzo Lobato<br>Isi Bardají Bofill<br>Marc Lucena González<br>Elias Ismael Estévez Macas<br>Victor Castellón Martín<br><br>Date 17/04/2023 | | Alejandro Amat Payá<br>Elias Ismael Estévez Macas<br>Esteve Tomasa Costa<br>Felip Martinez Quiros<br>Llorenç Sastre Galmés<br>Óscar Díaz Almendros<br>Pablo Diaz Soley<br><br>Date 18/05/2023 | |
| Name | Victor Castellón Martín | Name | Esteve Tomasa Costa |
| Position | e.g. Docum. Resp. | Position | e.g. Project leader |

# 0. CONTENTS

# 1. DOCUMENT SCOPE

The purpose of this document is to present a final report of the project, developed by a group of UPC students for the company HP. This document will explain in detail how it has been developed, the initial approach, the obstacles or problems encountered and how they have been solved.

This analysis will include a summary of the main objective of the project and the organization of the work time spent. It will also compare the design idea, implementation and characterization with the final work that has been done.

In addition, this document will discuss an overview of the costs, their management and the project strategy. The business model canvas will be followed by a study on sustainability and work ethics.

To conclude, this report will outline the conclusions, lessons learned, details for improvement and the team's performance.

## 2. PROJECT SUMMARY

In the construction sector one of the tasks to be carried out, once the structure of the building has been constructed, is to mark the internal divisions that must be built, i.e. internal walls, situation of the doors, etc. Traditionally this task is done manually by workers who paint the marks on the ground from a distribution made on a plan. In this context HP has developed a robot that, based on the information on a map, paints tha marks on the ground, the HP SitePrint [1] (Figure 1.1). This new tool makes work easier and faster, while reducing the number of workers needed. However, sometimes there are errors between the map and the structure that was built, which means that the painted marks have to be erased and repainted.

By design, the HP SitePrint has a built-in camera Intel RealSense D435i (Figure 1.2), and the challenge that HP engineers have proposed to us is to develop an algorithm that, based on the measurements that can be taken with this camera, detects if there are errors between the constructed reality and the map, and warns the user, so that wrong marks are no longer painted.

The Python programming language is used to develop the algorithm. Although any other programming language could have been chosen, Python was chosen because of the direct recommendation of HP due to its wide range of available libraries that facilitate the development of the code.

To achieve these objectives, the team is divided into three sub-teams: the first one is dedicated to code implementation and testing, the second one writes all the documents reporting the progress of the project and the third one is focused on the design and construction of a mock-up with the members of the second group.

In terms of specifications, HP has proposed three requirements:

- Measure to an accuracy of 3 mm.
- Calculate the alignment and position of columns and walls.
- Verify that the calculated surfaces are where they are expected to be on the building plan.

Finally, this document also explains the successes and failures of the project, as well as lessons learned and recommendations for future projects. In addition, it argues the limitations and results obtained in the development process.

*Figure 2.1. HP SitePrint*

*Figure 2.2. Intel RealSense D435i*

# 3. TIME PLAN UPDATED

The following are the changes that have occurred in the planning of the different tasks due to various setbacks or problems that have appeared. The delays or changes in each of the tasks are explained sprint by sprint.

In the first sprint there has not been any variation or delay, all the tasks have been developed according to the deadlines and times planned, because in this sprint we started to investigate the challenge and to find out how to approach the idea.


*Figure 3.1. Sprint week 1-2*

In the second sprint, once the main idea was focused, the code was developed. The document and layout parts continued to progress well. All tasks were completed on schedule.


*Figure 3.2. Sprint week 3-4*

As the vision and the objectives of the project were not properly understood from the beginning, it was decided to talk with Hp in order to gain new insights for the third sprint. This led to a change in the approach and development of the code part of the project. Therefore, all the tasks related to software development were affected. On the other hand, the documentation part of the project continued to develop as planned.

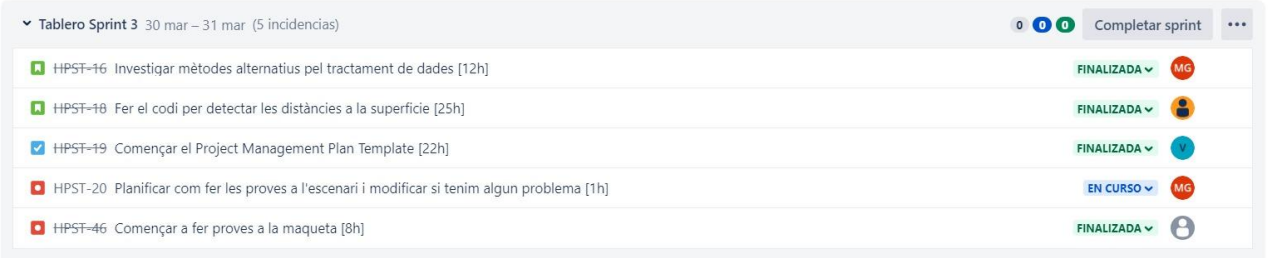


*Figure 3.3. Sprint week 5-6*

The fourth sprint was also affected by the aforementioned setback and all tasks were delayed.

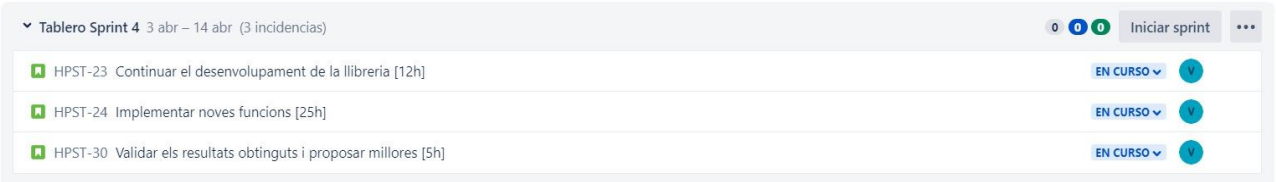


*Figure 3.4. Sprint week 7-8*

In the fifth sprint, the document part of the project was started on schedule and on time. On the software side, since the Critical Review presentation had already been made, the project was re-directed more according to the specifications and changes that HP proposed. Therefore, the tasks were adapted to these changes and new priorities, which meant that the other tasks were modified and in some cases there were delays.

From the meeting with the HP manager certain relevant values or final objectives were proposed and we focused especially on these. This caused the focus towards the final part of the project to be in a small part different. Because of this, many tasks on the software side had to be modified, but the hours spent on them were not affected.

The documents part was developed according to the established times and without any incidence or change.

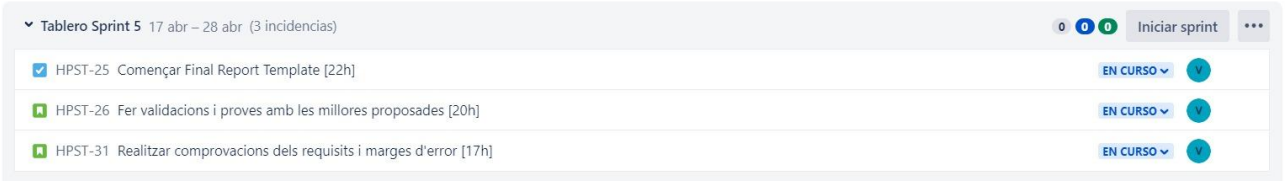


*Figure 3.5. Sprint week 9-10*

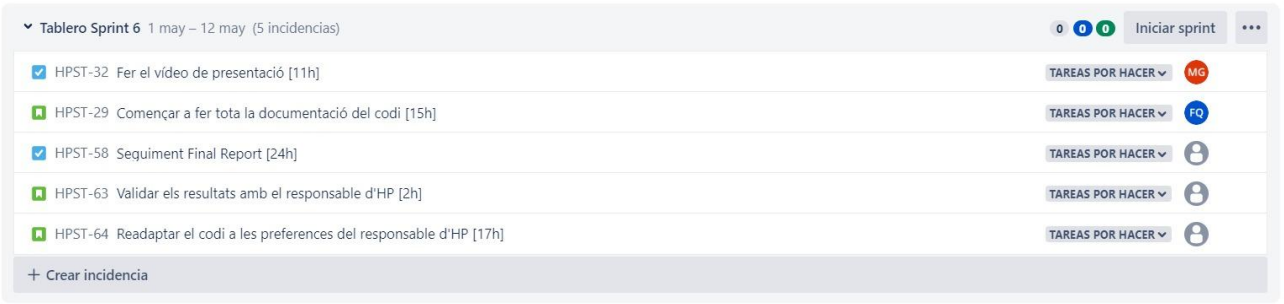


*Figure 3.5. Sprint week 11-12*

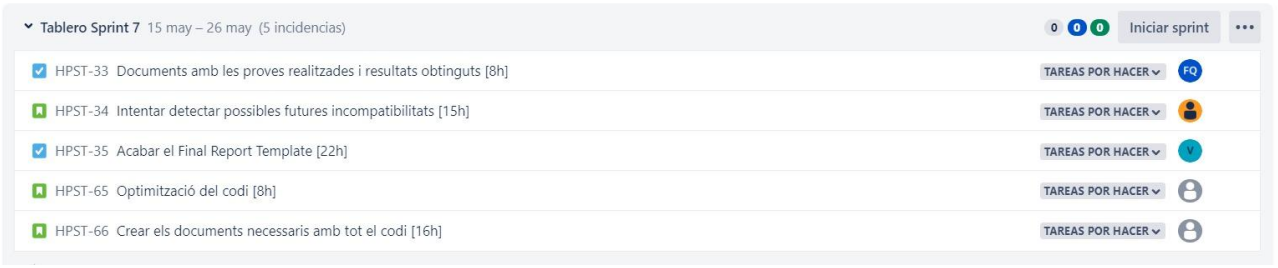


*Figure 3.7. Sprint week 13-14*

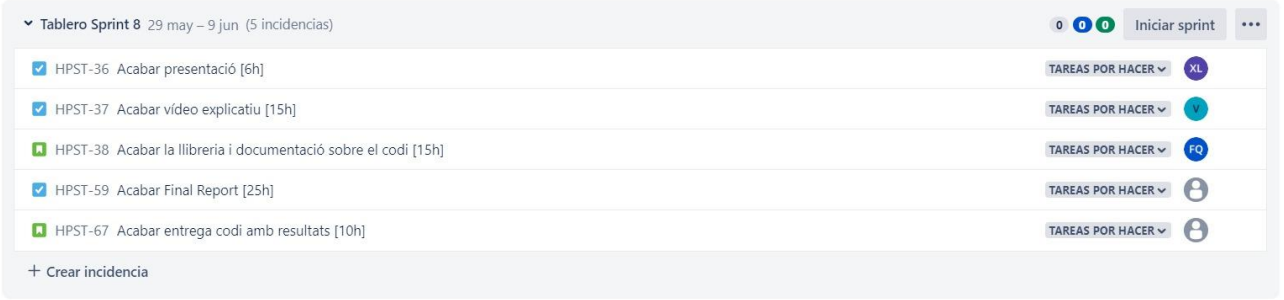


*Figure 3.8. Sprint week 15-16*

# 4. SYSTEM DESIGN DOCUMENTATION

Since the project consisted in infrastructure and construction error detection, the solution was to make two distance measurements. Comparing the theoretical distance with the real one, obtained with the camera information, it was possible to obtain the error computation.

It was decided that the final product would be a Python library which would be uploaded to the official python repository; Also known as Pypi.

Doing this has certain advantages. The first thing is that By uploading your package to PyPI, you make it easier for users to discover, download, and install your package using standard tools like `pip`. This simplifies the distribution process and ensures that your package can be easily installed on different systems and environments. It also provides versioning support, allowing you to release new versions of your package as you make improvements or introduce new features. It is also helpful as  Users can specify the desired version when installing your package. FInally, as many tools rely on piPY repositories, it has a great integration with the system as it may be in the CI process or in code editors.

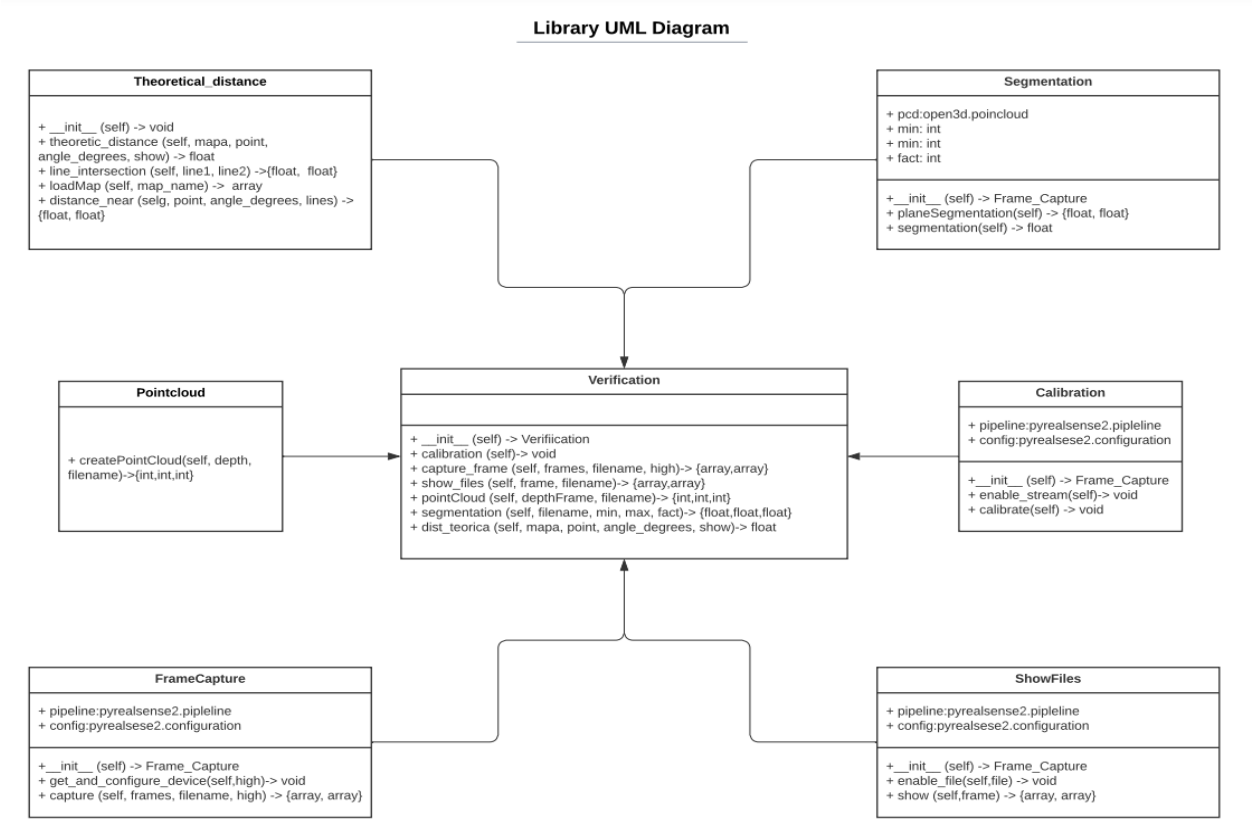


*Figure 4.1. Project block diagram*

An Object-Oriented Paradigm has been implemented (As shown in 4.1). This not only gives a much more neat and structured code but it enables to have all the functionality in one class. This is helpful because when installing our library it is only necessary to import the Verification module. This class is connected to all the other classes and abstracts all their complexity into short and easy functions.

To obtain the theoretical distance it is necessary to know the robot's position and the digital representation of the theoretical set up. With this data, the computation is performed. The calculation is purely geometric. An example of measuring the distance between the HP SitePrint and any obstacle can be seen below:

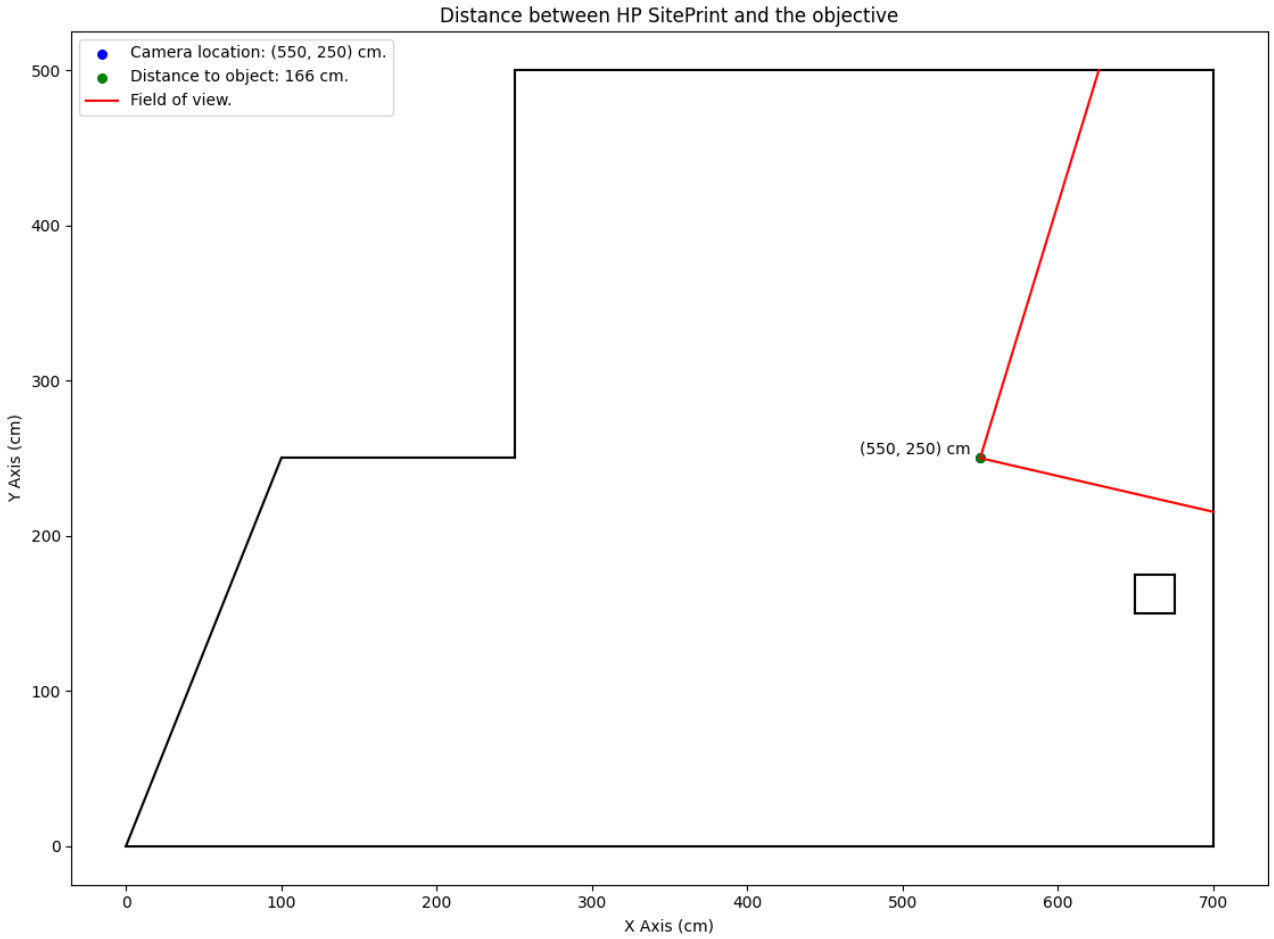


*Figure 4.2. Realization assuming position (550,250) and an angle of 30°*

To obtain the actual distance the Intel RealSense d435i camera has been used together with a python library created by the team. Thanks to the implementation of Point Cloud data structure segmentation algorithms, it is possible to to estimate the real distance from the camera plane to the object.

Once the two distances have been obtained, the only thing left is to compare them to see if there has been any construction error with respect to the initial plan.

To perform all these calculations have been used use the following python public libraries:

- **pyrealsense2:** Provides an interface to interact with Intel's RealSense cameras. This library allows access to data captured by cameras, such as images, depths and spatial coordinates, using this data to develop applications that can capture and process information in real time.

- **numpy:** Provides support for efficient numerical calculations and advanced mathematical operations.

- **cv2:** Provides a set of functions for processing images and videos in real-time, such as reading and writing images, converting colored spaces, segmentation of images, etc.

- **open3d:** Provides a set of functions for computational geometry such as alignment and recording of images, object segmentation, etc.

- **imageio:** Provides functions for reading and writing file images and videos.

- **matplotlib:** This library is used to create graphics in order to represent the plans of the building.

- **math:** It provides us with a set of basic mathematical functions for scientific and mathematical calculations. It will use this for trigonometric functions for theoretical distance.

- **ast:** It provides us with the necessary functions for the construction of specific programming languages for mathematical tasks.

- **OS:** This library provides us with functions for creating and deleting directories, obtaining current directories and all kinds of cross-file management.

# 5. SYSTEM IMPLEMENTATION DOCUMENTATION

A python library has been designed to be used by its own or to complement the already developed HP Site-Print product. The new library uses other complementary libraries published in the PyPi repository. The structure has been developed following the block diagram that has been shown in figure 4.1. By using this architecture, we are able to locate all the functionality into one class, making easy and short methods in order to simplify the usage.

First, the data captured by the sensor is displayed. For this, verif.capture_frame() and verif.show_files() are used. Firstly, the device is configured. This enables depth and RGB streams so that the gathering data process starts. The number of depth and color frames specified are captured. The depth frames give the information of distance (depth) of the points of the frame to the camera plane.

Depth frame information is used as it is enough to achieve the objective of high precision structure verification. However, it was desired to create functions as general and portable as possible so color frames were added in case of other upgrades or algorithms needed to use this information.

An example of the output of those functions is shown in figure 5.1:

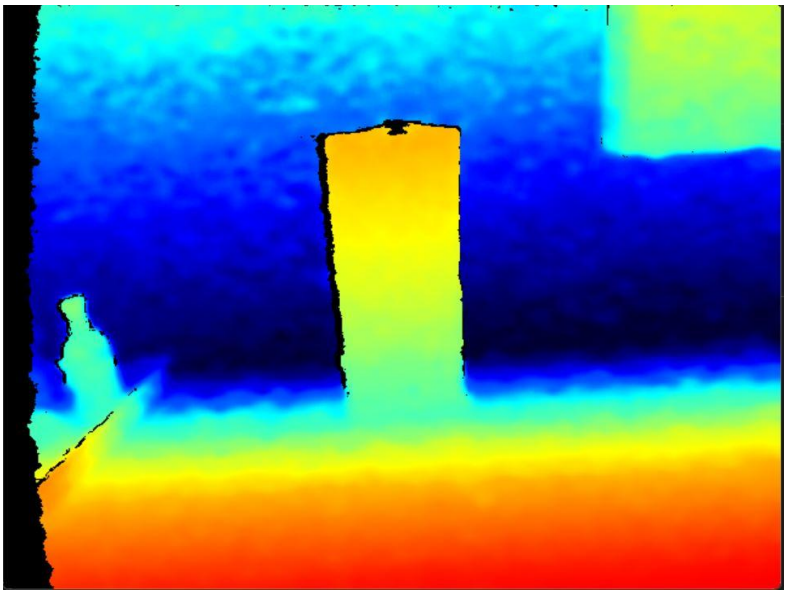

*Figure 5.1. First data collected by the sensor (depthframe)*

This frame is converted to a PNG image to apply some algorithms to fill gaps and correct imperfections. Once it is done, our algorithm is able to create the first Point Cloud with *verif.PointCloud().* A first view of the pointcloud is the figure 5.2:

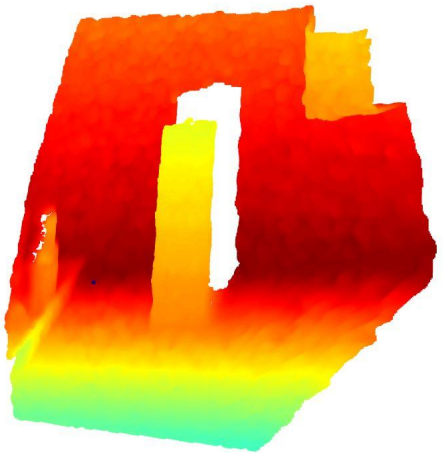

*Figure 5.2. First Point Cloud version*

Depth frames store the depth values of a frame for each pixel. This type of storage is not enough to continue processing the data in an efficient way. That is why Point Cloud data structures are needed. PointClouds are just an accurate and detailed 3d representation of the depth frames. They are beneficial as they are a flexible data structure that allow efficient storage for processing 3d information. They are also sensor independent. When the camera computes the depth frames it relies on its intrinsic parameters, such as the optical center and its focal length. But by the way the PointClouds have been constructed, the coordinate transformation erases these dependencies. Finally, there exist many algorithms and libraries that have already been implemented for these types of data structures, so it eased the work.

The verif.PointCloud() function processes the point cloud generated in the previous step. It segments the image by identifying the planes or using algorithms such as DBSCAN (RANSAC) to define to which surface the distance is being calculated.

The algorithm begins by creating a precise 3D representation of the scenario using a point cloud and real plan details. Segmentation is applied to accurately calculate distances to desired objects by separating them from irrelevant points. This approach works effectively in different scenarios and allows for distinguishing and calculating distances to multiple objects.

To achieve this, the algorithm uses plane segmentation to detect walls and columns that are not tilted or angled. For structures that are tilted or angled, the algorithm incorporates the DBSCAN algorithm. This update ensures accurate detection and segmentation of these structures. By selecting relevant clusters based on variance and the number of points, the algorithm successfully identifies desired objects. With the calculated distances, it compares them to theoretical values, achieving an error rate of less than 3% throughout the camera's operating range, fully utilizing the capabilities of the Intel RealSense camera.

In the figure (Figure 5.3) you can see a red plane created from the points detected as a wall. This would be useful to calculate the distance to it. But in this case we are looking for the distance to the object that can be observed in the center, so it is only interesting to detect the wall to eliminate it.

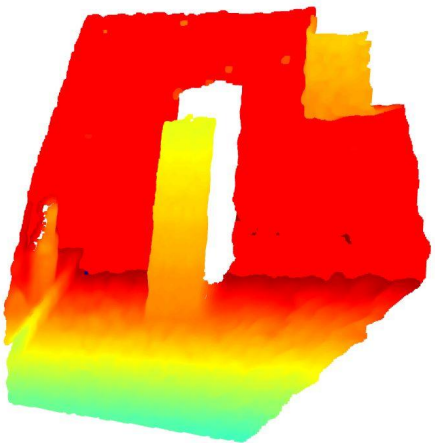

*Figure 5.3. Detecting the wall as a plane.*

Then we proceed to eliminate the floor and all the points detected as uninteresting. To eliminate the floor are chosen all those points that are all those that generate a relatively flat surface (due to the distortion of being so close) and with an inclination practically perpendicular to the wall.

In this case, the surface detected as floor and deleted is the following one (Figure 5.4):

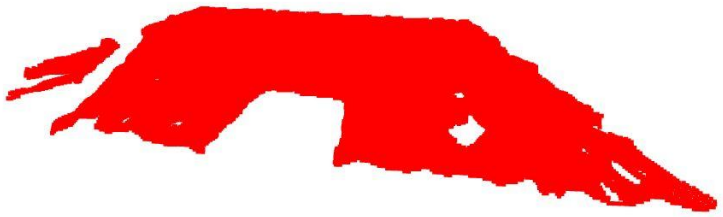

*Figure 5.4. Surface detected as floor and eliminated*

After having eliminated almost all flat surfaces that are of no interest, floating points that are noise or small surfaces that were also of no interest are eliminated. The DBSCAN algorithm allows to know how many points belong to that object, and only the largest one is retained, which is usually the one of greatest interest. This behavior can be seen in the figure (Figure 5.5).

*Figure 5.5. Point cloud after DBSCAN algorithm*

Finally, the largest area is saved and the distance to it is calculated. This distance will be compared with the theoretical one to find the error and to warn in case it is higher than the established limit, in which case the user will be warned. The calculated distance is to the largest and central object, as can be seen in the following image:

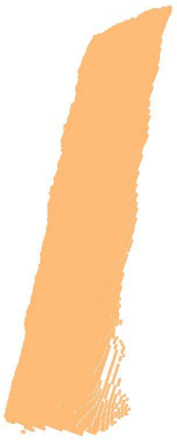

*Figure 5.6. Biggest object detected*

# 6. SYSTEM CHARACTERIZATION

To carry out the verification of the measurements obtained by the camera, first a scenario simulating a smaller scale empty room was set up. This room had a floor with millimeter markings to determine the exact distance from the camera to the walls. Thanks to these measurements, the exact physical distance was determined and compared with the distance calculated by the software, in order to ensure its correct operation. The mock-up can be seen in the following image:

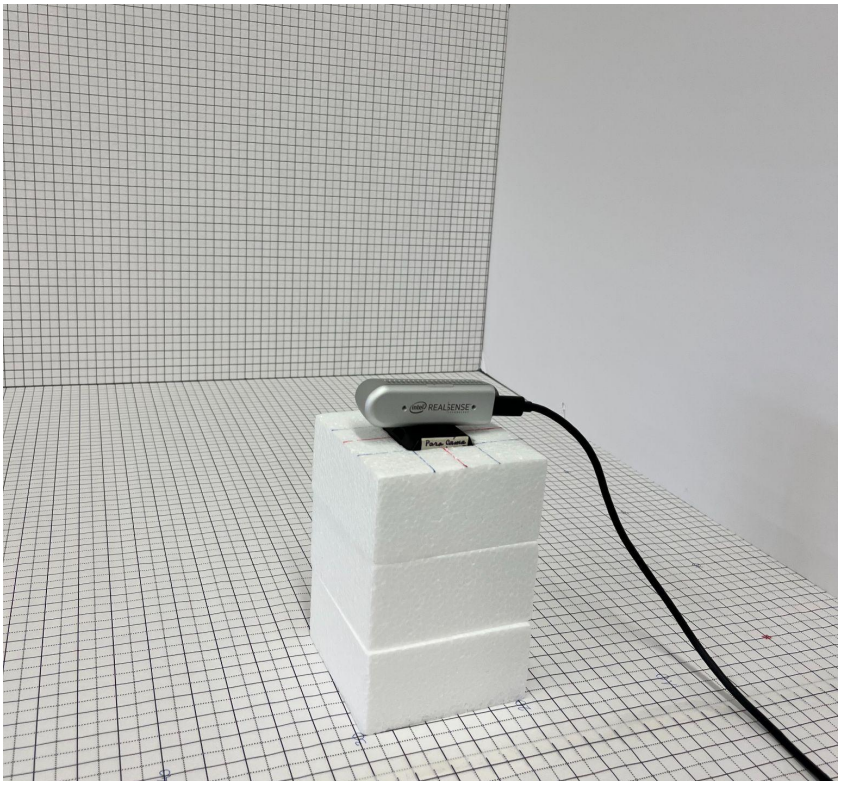

*Figure 6.1. First set-up*

After checking and adjusting the camera for short distances, a maximum threshold of acceptable accuracy was established. It was agreed that an accuracy level of no more than 2% of the measured distance would be adequate.

To determine this threshold, a more realistic scenario was applied. Since the software was more developed at this point, the first scenario was replaced by this one for all the following

measurements. This allowed a more realistic scenario to be achieved and the algorithm to be refined for greater accuracy.

The graph below illustrates the evolution of the precision improvement achieved, as evidenced by the reduction in the percentage of error between the theoretical and calculated distances.

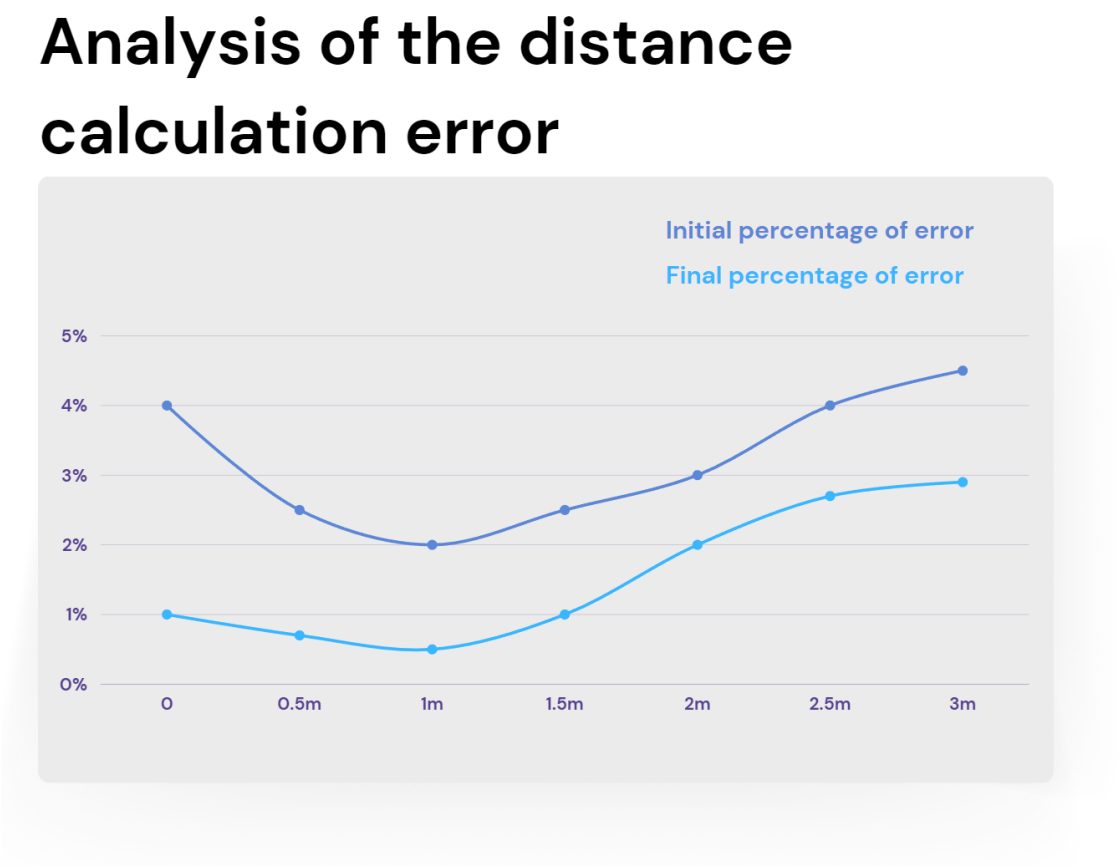


*Figure 6.2. Graphic about Analysis of the distance calculation error*

The optimal performance of the algorithm is found between distances of 0.6 to 1.8 meters. Beyond this range, as the distance increases, the error also increases until it reaches 3% calculation error.

During the performance of the first measurements, it was concluded that any kind of interference of the light beam (whether artificial or natural) affected the correct calculation.

However, there are different situations that must be taken into account in order to make a correct calculation of the measurements, such as:

- There can be no direct illumination to the camera sensor.
- Distances cannot be measured against a reflective surface such as glass.
- The distance from the camera to the object must be within the specified values, otherwise the error is very large.

Below, you can see the camera in a real scenario like the laboratory wall.

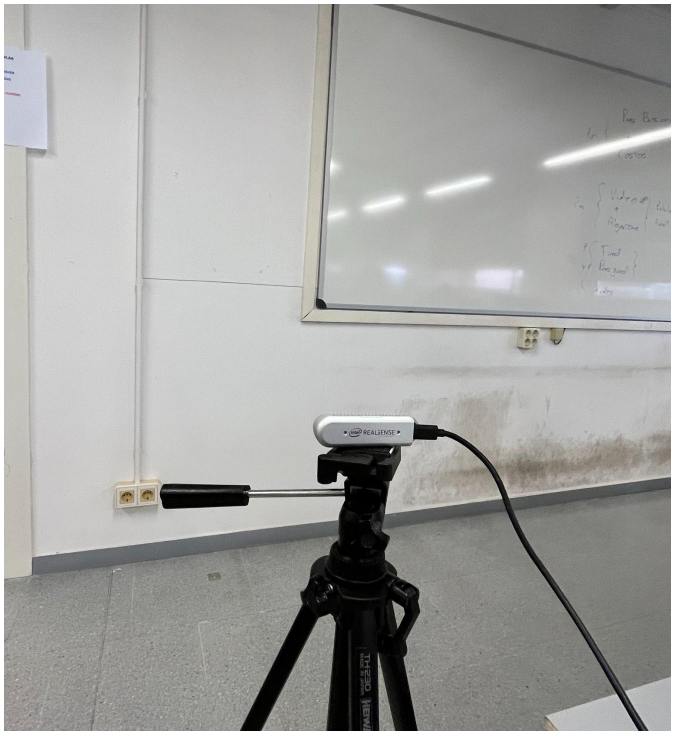

*Figure 6.3. Second and more realistic scenario*

# 7. COSTS

The project proposed by HP does not impose any budget constraints. For this reason, the cost analysis shows the expenses generated during the realization of the project.

To calculate the cost of the project, it is divided into 3 categories: personnel expenses, costs related to materials and costs associated with the laboratory.

Personnel costs:

First of all, to determine the salaries of the team it will be taken into account that despite being a start-up, the workers are considered to be trainee engineers and therefore the salary will be 9 € per hour. Consequently, despite the different positions assigned in the team, the salary will be the same for all of them.

The social security costs to be paid for each employee must be taken into account. A cost that represents an additional 30% of the monthly salary is assumed. This factor means that the total cost of employees is 36,995 euros, as can be seen in Figure 7.1:

| Personal | Quantity | Work Hours | Salary/Hour | Salary/Person | Total Salary | S.C. (30%) | Total cost |
|---|---|---|---|---|---|---|---|
| Software developers | 6 | 282 | 9 € | 2.538 € | 15.228 € | 4.568 € | 19.796 € |
| Document managers & Scenario makers | 5 | 294 | 9 € | 2.646 € | 13.230 € | 3.969 € | 17.199 € |
| TOTAL PERSONAL COST | | | | | | | 36.995 € |

*Figure 7.1. Salaries Costs Table*

Resources costs:

For the execution of the project the university provides a work space suitable for 11 workers. This space is not an expense, but it is taken into account in the cost analysis. In addition, the basic resources of the internet, water and electricity will be added. The resources are divided into direct cost and indirect cost. Electricity and internet are considered direct costs because they are used to produce a good or provide a service. On the other hand, water and rent are indirect costs because they are attributed to specific services or activities.

| | Cost per month | Total Months | Cost |
|---|---|---|---|
| ***Direct Cost*** | | | |
| Electricity | 80 € | 4 | 320 € |
| Internet | 70 € | 4 | 280 € |
| ***Indirect Cost*** | | | |
| Water | 40 € | 4 | 160 € |
| Rent | 2.000 € | 4 | 8.000 € |
| **TOTAL** | 2.190 € | | 8.760 € |

*Figure 7.2. Resources Costs Table*

Material costs:

To calculate the cost of materials, depreciation has to be taken into account. It is assumed that the useful life of intangible materials is for immediate use and therefore is not taken into account, that of electronic equipment is 5 years and that of office furniture is 10 years. Moreover, due to the fact that Python is an open source language, the cost of its license is not counted.

Taking these assumptions into account, depreciation has been calculated as follows:

$$Depretiation = \frac{Total\ Costs - [Total\ Costs * Residual\ Value(\%)\ ]}{Depretiation\ Years}$$

Depreciation refers to the process of decrease in value of a tangible or intangible asset over time due to use. Calculated as the result of the total value minus the residual value, divided by the years of useful life. This residual value is the value that the product will have after the years of useful life, which is the cost of the product times the percentage of the residual value.

| Material | Provider | Quantity | Price | Total costs | Residual Value(%) | Residual Value | Depreciation years | Depretation in 1 year |
|---|---|---|---|---|---|---|---|---|
| *Direct Cost* | | | | | | | | |
| Laptop | UPC | 11 | 1.549 € | 17.039 € | 30% | 5.112 € | 5 | 2.385 € |
| Featherboard | Amazon | 5 | 14 € | 70 € | 0% | 0 € | 0 | 0 € |
| Grid paper | Amazon | 1 | 28 € | 28 € | 0% | 0 € | 0 | 0 € |
| Millimeter paper | Amazon | 1 | 11 € | 11 € | 0% | 0 € | 0 | 0 € |
| Foam blocks | Amazon | 1 | 12 € | 12 € | 0% | 0 € | 0 | 0 € |
| Black paint spray | Amazon | 1 | 6 € | 6 € | 0% | 0 € | 0 | 0 € |
| Red paint spray | Amazon | 1 | 5 € | 5 € | 0% | 0 € | 0 | 0 € |
| Intel RealSense D435i | HP | 1 | 355 € | 355 € | 30% | 106 € | 5 | 50 € |
| *Indirect Cost* | | | | | | | | |
| Table | UPC | 11 | 52 € | 572 € | 50% | 286 € | 10 | 29 € |
| Chair | UPC | 11 | 85 € | 935 € | 50% | 467 € | 10 | 47 € |
| Projector | UPC | 1 | 659 € | 659 € | 30% | 198 € | 5 | 92 € |
| Whiteboard | UPC | 1 | 112 € | 112 € | 50% | 56 € | 10 | 6 € |
| **TOTAL** | | | | 19.803 € | | 6.225 € | | 2.608 € |

*Figure 7.3. Material Costs Table*

The following table shows the cost summary of the project with the costs mentioned in the previous tables, pertaining to hardware, monthly expenses, personnel and materials.

| Total costs of the project | |
|---|---|
| *Direct cost* | |
| Hardware | 17.394 € |
| Monthly costs | 600 € |
| Team members | 36.995 € |
| Materials | 132 € |
| **TOTAL DIRECT COST** | 55.121 € |
| *Indirect cost* | |
| Monthly costs | 8.160 € |
| Materials | 2.278 € |
| **TOTAL INDIRECT COST** | 10.438 € |
| Depretiation in 1 year | -2.608 € |
| **TOTAL with depretiation** | 62.950 € |
| **TOTAL** | 65.559 € |

*Figure 7.4. Total Costs Table*

On the one hand, the total cost of the project without taking into account depreciation is 65,558€. On the other hand, taking into account the depreciation of the materials in one year, the total price will be 62,950 €.

# 8. BUSINESS MODEL CANVAS

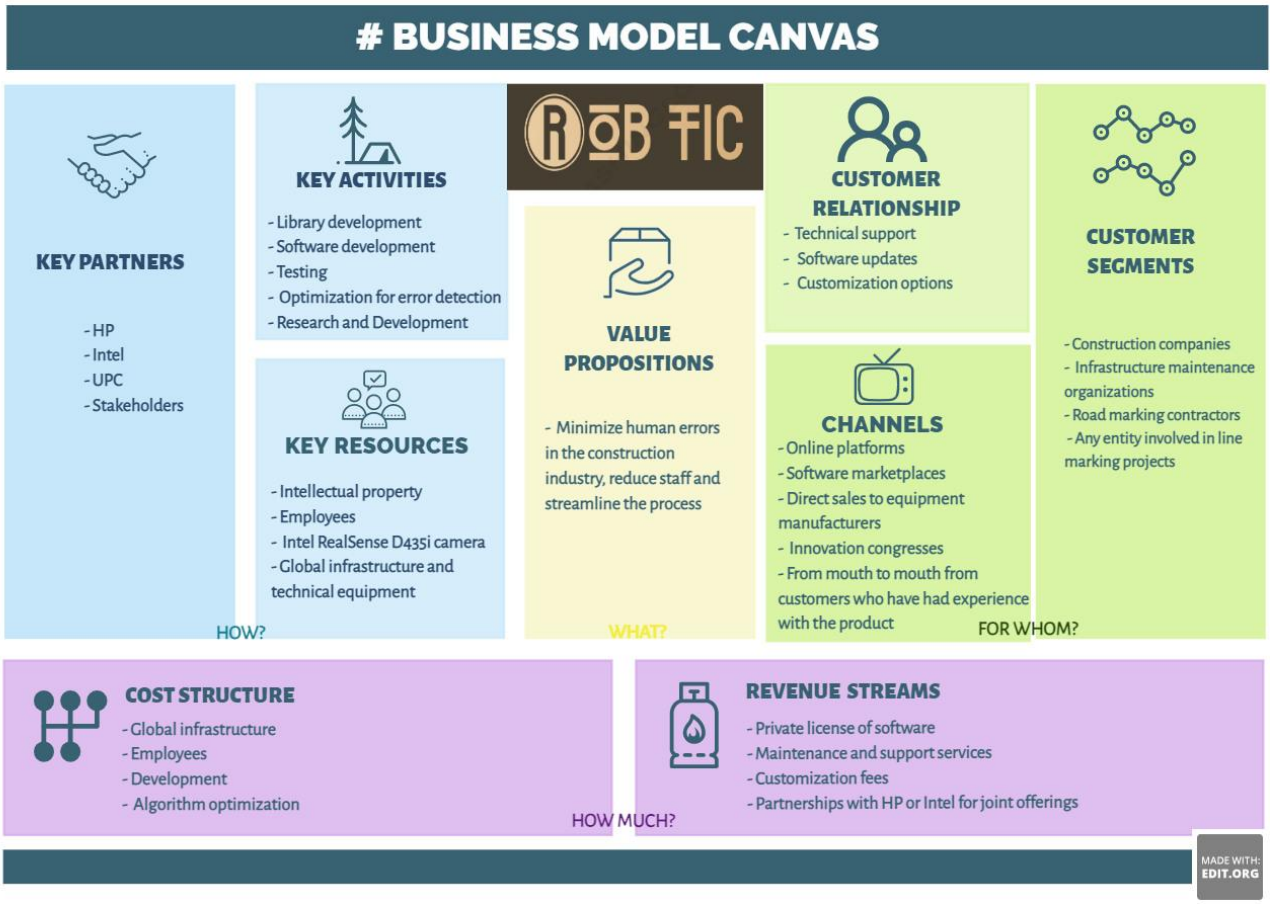


*Figure 8.1. Business Model Canvas*

**Key partners:**

Our key partners are those interested in developing an algorithm to validate the error when printing lines on buildings under construction with respect to the theoretical plan. In this case it would be the company HP and Intel for access to the RealSense D435i camera technology. Collaborations with construction industry stakeholders, software vendors, or research institutions could also be beneficial. In addition to the UPC institution, which gives us support and help to develop this project.

**Key activities:**

Our company stands out for providing a library with the necessary software to detect the error between the distance made with the Intel D435i camera and a theoretical map of a plane.

In addition to offering customer support and a research and development team that is constantly informed and updated on any progress to update the software or incorporate improvements or solve any problem.

**Key resources:**

Resources have been divided into two categories. On the one hand, physical resources, which include employees, facilities and equipment. On the other, intellectual resources, which are intangible and include intellectual property and technology.

**Value propositions:**

The most important value of our project consists of designing and implementing an algorithm that can measure the position of walls and other obstacles with the HP SitePrint and compare them with a theoretical floor plan.

This project can be a solution to minimize human errors in the construction industry, reduce staff and streamline the process.

**Customer relationships:**

Our customer relationships will have a confidentiality agreement not to discuss other business with your competitors and maintenance software. In addition to providing them with software adapted and customized to their needs.

**Channels:**

The algorithm can be distributed through software integration with the HP SitePrint system or provided as a standalone software package. It can be accessed through online platforms, software marketplaces, or direct sales to equipment manufacturers.

**Customer segments:**

The customer segments would be companies that use a robot to print lines on buildings and need to validate the correct or incorrect position of the lines or are looking for different applications of the intel D435i camera.

**Cost structure:**

Our cost structure would be based on working at the UPC institution, with its equipment and facilities and the salaries that our workers will demand. On the other hand, HP provided us with the appropriate hardware material for the proposed project. Would also include expenses related to research and development, software engineering, algorithm optimization, employee salaries, marketing, and ongoing support and maintenance.

**Revenue streams:**

Our main revenue would come from software licensing or subscriptions, maintenance and support services, customization fees, or partnerships with HP or Intel for joint offerings.

## 9. IP CONSIDERATIONS

At the beginning of this project, in order to be part of it, we had to sign with HP a declaration of assignment of rights and confidentiality. This document was made because the project that was offered to us was related to the improvement of the software of a product, which was going to be launched to the market. Therefore, there was certain information that could only be used to understand and extract the optimal solutions for the project and could not be passed on outside this scope. One of these reasons could be the arrival of information to certain competitor companies with similar ideas that might have similar products and different specifications.

On the other hand, as the project has been carried out, it has been possible to see that certain ideas that have arisen and have been implemented could be useful in the improvement of the product made by HP. In order to protect these ideas that can be applied in their already established product to generate specific or general improvements, an intellectual property contract was established.

The mentioned ideas have been dedicated to the field of software development. This part of the project has been approached in a public way by generating a library with all the developed code, available to anyone. The established approach is due to the fact that it has not been possible to innovate in the algorithm, that is to say, code already implemented has been used, allowing us to draw some conclusions in relation to the project.

In conclusion, it can be seen that this confidentiality generates a benefit for both parts of the project and that if an argued idea had been generated that had not yet been created, it could have been implemented privately, obtaining benefits if it had been patented.

# 10. SUSTAINABILITY ANALYSIS

The sustainability analysis has been divided into three parts:

- The Development of the project, which starts with the acceptance of the project by us and ends when the project report is submitted.
- The exploitation of the project, which begins once the project has been implemented and ends when it is dismantled (estimation of the hypothetical exploitation of the product or service presented).
- The risks and limitations inherent to the project during its construction, useful life and dismantling.

The environmental, economic and social impact of each of these has been studied, as shown in Figure 1.2.

| | **Project Development** | **Exploitation** | **Risks/ limitations** |
|---|---|---|---|
| **Environmental** | Consumption of the design | Carbon footprint | Environmental risks and limitations |
| **Economic** | Invoice | Viability plan | Economic risks and limitations |
| **Social** | Personal impact | Social impact | Social risks and limitations |

*Figure 10.1. Project sustainability matrix*

A matrix of questions (Annex II) has been used to break down each point, which has made it possible to go into detail. For each section, two points of view are presented, one relating to the initial part of the project and the other as the project develops.

## 10.1 Project Development

### 10.1.1 Consumption of the design

The estimation of the environmental impact at the beginning of the project was based on the calculation of the amount of CO2 produced by the resources that had been foreseen at the time of planning the project. These include the material used to create the scenario for testing with the camera (by online-market), the use of social networks for internal and external communication (HP employer and teachers), as well as using them to share documents; the use of public and private transport for mobility; the electricity consumed by the different devices and the access to the network and cloud.

It can be observed, the summary table of the estimated amount of CO2 emitted by the project (for more details see Annex II).

| | Impact (kg CO2) |
|---|---|
| Electronic device | 207 |
| Internet tools | 43 |
| Movility | 322 |
| Model material | 0 |
| TOTAL IMPACT | 572 |

*Figure 10.2. Summary Table of the Estimate Environmental Impact*

At the beginning, no objective was set to minimize the environmental impact, so no strategies to approach the project in a sustainable way were proposed at any time. Also, as it was the creation of an algorithm, there were no quick connections with sustainability issues, such as the recycling of materials or the origin of the materials. As for the purchased material, the only thing that was considered was to buy it from the same webmarker to speed up the shipping process. The final environmental impact can be observed in the summary table below (for more details see annex 2):

| | Impact (kg CO2) |
|---|---|
| Electronic device | 107 |
| Internet tools | 32 |
| Movility | 223 |
| Model material | 0 |
| TOTAL IMPACT | 362 |

*Figure 10.3. Summary table of Finally Environmental Impact*

During the development of the project, the resources previously proposed have been maintained. However in order to reduce the impact, it has been agreed not to buy more material for the realization of scenarios, but to use reusable material provided by the UPC to simulate columns and/or obstacles. Another measure has been to dispense with the printing of content on sheets, and in the case of communication with teachers or HP managers to do so by means of a single email, which compiles the problems encountered, sent by the project leader, and finally, the use of github and other data stores as tools for sharing code and documents, thus avoiding the abuse of constantly sending new versions through other networks. Moreover, the usage time of some has been reduced.

The ecological footprint by the project providers(HP, Intel and Amazon) has been analyzed to see the impact of the materials and services used. The materials provided by HP at the start of the project, which are Intel products, both the camera and the USB cable are manufactured in China, and no information has been found on the web regarding the materials they are made of. However, Intel reports that 100% of its strategic suppliers (representing approximately 80% of its purchasing spend) have established environmental sustainability targets and are evaluated annually to assess their compliance. In addition, 92% of Intel's direct suppliers (representing approximately 90% of its purchasing spend) have been evaluated for compliance with Intel's environmental sustainability standards.

On the other hand, the materials purchased from Amazon used for the creation of the model are all sourced from China and contain no conflict materials. Amazon reports that 72% of its suppliers by purchase spend have established environmental sustainability targets. In addition, Amazon

evaluates its suppliers on their environmental sustainability policies and takes steps to improve the environmental performance of its suppliers where necessary.
HP is committed to the "Cradle to Cradle" philosophy (as are Intel and Amazon). In addition, it uses renewable electricity for 43% of global operations and plans to increase that number to 100% by 2035 [5]. It claims that 87% of its suppliers per purchase spend have set environmental sustainability targets. In addition, 99% of suppliers per spend have been assessed for compliance with HP's environmental sustainability standards.

To conclude this section, all three material supplier companies follow RoHS directives and the RBA Code of Conduct. Of particular note are Intel and HP, both of which are members of the RBA.
Overall, all three companies have environmental and social policies and standards and are committed to reducing their environmental and social impact in their operations and supply chain.

### 10.1.2 Invoice

The estimated costs of the project (in terms of human and material resources) can be seen in the summary table( Figure 7.4.).For more details, please see section 7 in full. The quantification of total project costs is presented in the following table:

| Material | Provider | Quantity | Price | Total costs | Residual Value(%) | Residual Value | Depreciation years | Depretation in 1 year |
|---|---|---|---|---|---|---|---|---|
| | | | | ***Direct Cost*** | | | | |
| Laptop | UPC | 11 | 1.549 € | 17.039 € | 30% | 5.112 € | 5 | 2.385 € |
| Featherboard (100cmx140cmx5mm) | Amazon | 5 | 14 € | 0 € | 0% | 0 € | 0 | 0 € |
| Grid paper (80cmx15m) | Amazon | 1 | 28 € | 0 € | 0% | 0 € | 0 | 0 € |
| Millimeter paper with 50 sheets | Amazon | 1 | 11 € | 0 € | 0% | 0 € | 0 | 0 € |
| Foam blocks (10,2x10,2x5 cm) with 12 pieces | Amazon | 1 | 12 € | 0 € | 0% | 0 € | 0 | 0 € |
| Black paint spray | Amazon | 1 | 6 € | 0 € | 0% | 0 € | 0 | 0 € |
| Red paint spray | Amazon | 1 | 5 € | 0 € | 0% | 0 € | 0 | 0 € |
| Intel RealSense D435i | HP | 1 | 355 € | 0 € | 30% | 0 € | 5 | 0 € |
| | | | | ***Indirect Cost*** | | | | |
| Table | UPC | 11 | 52 € | 572 € | 50% | 286 € | 10 | 29 € |
| Chair | UPC | 11 | 85 € | 935 € | 50% | 467 € | 10 | 47 € |
| Projector | UPC | 1 | 659 € | 659 € | 30% | 198 € | 5 | 92 € |
| Whiteboard | UPC | 1 | 112 € | 112 € | 50% | 56 € | 10 | 6 € |
| **TOTAL** | | | | 19.317 € | | 6.119 € | | 2.559 € |

*Figure 10.4. Final Material Costs Table*

| Personal | Quantity | Work Hours | Salary/Hour | Salary/Person | Total | S.C. (30%) | Total cost |
|---|---|---|---|---|---|---|---|
| Software developers | 6 | 282 | 9 € | 2.538 € | 15.228 € | 4.568 € | 19.796 € |
| Document managers & Scenario | 5 | 264 | 9 € | 2.376 € | 11.880 € | 3.564 € | 15.444 € |
| TOTAL PERSONAL COST | | | | | | | 35.240 € |

*Figure 10.5. Final Salaries Costs Table*

| Total costs of the project | |
|---|---|
| ***Direct cost*** | |
| Hardware | 17.039 € |
| Monthly costs | 600 € |
| Team members | 35.240 € |
| Materials | 0 € |
| TOTAL DIRECT COST | 52.879 € |
| ***Indirect cost*** | |
| Monthly costs | 8.160 € |
| Materials | 2.278 € |
| TOTAL INDIRECT COST | 10.438 € |
| Depretiation in 1 year | -2.559 € |
| TOTAL with depretiation | 60.759 € |
| TOTAL | 63.317 € |

*Figure 10.6. Final Total Costs Table*

The decisions taken to reduce costs have been:

- Human resources reduction: After realizing that one department was completing their tasks more quickly, it has been decided to reduce the allotted time for those tasks.
- Reduction material cost: An agreement has been reached with the UPC and HP to provide the necessary materials for the creation of the scenery and the camera free of charge.

Finally, it was noted that the cost reduction was 4%, which indicates that the measures taken to reduce the price do not have a significant impact. This is due to the fact that more than 50% of the project cost comes from salaries.

### 10.1.3 Personal impact

First of all, the choice of the project was focused on "work" enrichment, as it was the first time to be involved in a project with similar characteristics to those encountered in the professional environment (propose the given project, develop and implement it and present the final result). In addition, the fact that HP was an eye-catching project encouraged us to choose it, as it is well known that it is a modern, innovative company with a great reputation in the technological world.

Throughout the project it has not considered using inclusive language or highlighting the figure of women as engineers or construction workers. Nor has there been any attempt to highlight the male figure, but the male gender has been used to refer to individuals.
Currently, the construction sector is a largely male-dominated sector (65%), which is not very modernized and is still facing challenges in terms of environmental and social sustainability. Therefore, the introduction of new robots which can automate the work but do it in a more precise way, as well as the introduction of new and more sustainable materials can cause a revolution in this sector.

Finally, it is worth noting that the companies involved in this project such as HP, Intel and Amazon, which are international companies, each have their own code of ethics and conduct that sets out their values, as well as ethical principles and standards of conduct to be followed by their employees and collaborators. In general, these companies strive to comply with their codes of ethics and conduct, although there may be cases where they have failed to comply with these standards. Therefore, due to the electronic components, international shipments and servers that have been used, there can be no 100% assurance that public codes of ethics or conduct have been complied with at all times.

## 10.2 Exploitation

### 10.2.1. Carbon footprint

By developing and implementing the algorithm and being able to incorporate it into the other functions performed by SitePrintHP, the environmental impact can be reduced in two main areas. One of them is the reduction of material used in case of errors by marking the plan lines with high precision, controlling the amount of paint used to mark the lines, also avoiding the collapse and reconstruction of walls, thus avoiding the huge waste of material and energy. On the other hand there is the reduction of material that would be needed to have workers performing this work, such as adequate clothing, cleaning material and protection material against toxic substances.

The resources used during the useful life of the project are basically focused on the servers and communication networks, which will allow the algorithm to be downloaded either to be implemented in the robot or to be useful in other projects that will make use of the algorithm. It is worth mentioning that the algorithm as such will not reduce any resources directly as it will be stored on a server until it is used, but it will be useful (in terms of sustainability) when it can be implemented in the robot, significantly reducing both resources and environmental impact. The useful life of the algorithm will end when it becomes obsolete and is deleted from the server or because the server stops working. The algorithm will not generate waste at the end of its life.
During the project there has only been waste from the materials used in the model, which have been donated to the UPC and the rest has been recycled, and waste emitted by the servers and communications network. It has been observed that a distributor, server and communication network that emits less CO2 could have been chosen.

### 10.2.2. Viability plan

The economic improvements that the algorithm can provide are:

- Reduced correction costs: The algorithm can detect errors in construction before they become too costly to correct. This can save money by allowing problems to be fixed in a timely and efficient manner, before they become major problems.

- Reduced construction time: Early detection of errors can reduce overall construction time. This can save money on labor costs and speed up project completion.

- Reduced insurance costs: Early detection of errors can reduce the likelihood of accidents and therefore reduce insurance costs associated with the project.
- Reduced material costs: Automating error detection will eliminate the need for all protective and cleaning material dedicated to the worker.

In order to reduce the estimated costs ( Figure 7.4), it has been proposed to find cheaper materials or to look for second-hand and reusable materials, in terms of cost for the creation of the model. In terms of personnel, it was found that certain areas were completing their tasks much earlier than planned, so the hours planned for each task had to be readjusted and working hours had to be reduced. However, it was found that this does not represent a large reduction in the estimated cost.

The costs of updates/repairs/adaptations of this project during its lifetime have not been considered, but in general the cost will not increase much, since the updates would be software (which would only increase the salary + cost of renting the laboratory) and in the case of the model, the material necessary to repair or replace defective parts will be accounted for, in our case, since it is a small model, the cost involved is minimal.

At the end of this project no disassembly costs will be incurred and everything developed can serve as a reference for subsequent algorithms, developed exclusively by HP or by anyone interested in the subject.

### 10.2.3. Social impact

In terms of the improvements in quality of life that the algorithm and its implementation in the robot can bring are as follows:

- Time saving: An algorithm can perform a task faster and more efficiently than a human being, which can save time and allow people to spend their time on other activities or make further progress in that field of research and development.
- Increased safety: Automating the accurate painting of lines for subsequent construction reduces the risk of accidents for workers and also leads to safer construction.
- Increased efficiency and reduced stress: The automation of line painting by a robot can increase productivity, thus enabling construction to be completed sooner. In

addition, workers can focus on less repetitive and more complex tasks, which can be more satisfying and less stressful.

The project is focused on the construction sector but no stereotypes or gender roles are presented, neither in the development and use of the algorithm, nor in the construction area. However, being a technical project, the usability and full understanding of the algorithm will present the following barriers:

- People with cultural and functional diversity may have difficulties in understanding the technical terminology used in the algorithm, as well as the instructions and error messages that are generated.
- People with physical disabilities may have difficulty interacting with the user interface of the algorithm, which may affect their ability to use it effectively.
- People with visual disabilities may have difficulty accessing the visual information generated by the algorithm, while people with hearing disabilities may have difficulty accessing auditory information.
- Inexperienced people may have difficulty adapting to new technologies and processes, which may affect their ability to use the algorithm effectively.

## 10.3 Risks/Limitations

### 10.3.1. Environmental risks and limitations

- Energy use for the operation of the algorithm and the machinery involved in the construction process can have an impact on the carbon footprint of the project.
- The disposal of electronic equipment related to the algorithm and the technology involved in the construction process can generate e-waste, which can be hazardous if not handled properly.
- If the application involved in the construction process requires the combustion of fossil fuels, this could generate emissions that affect air quality in the project area.

- The use of natural resources for the production and maintenance of the technology may have an indirect impact on the environment, such as the extraction of minerals and metals for the manufacture of electronic devices.

### 10.3.2. Economic risks and limitations

- Integration cost: Integration of the algorithm and robot into existing construction processes could require modification of existing systems and processes, which could result in additional costs.

- Maintenance cost: In addition to the initial cost, maintenance of the robot and algorithm could be costly, which could affect the overall cost of the project.

### 10.3.2. Social risks and limitations

- Displacement of workers: If the algorithm is used to improve construction efficiency, it could result in a decrease in the need for labor, which could lead to the displacement of workers.

- Lack of training: If workers are not properly trained to use the algorithm, this could result in decreased efficiency and increased construction time, which in turn could affect project cost and customer satisfaction.

- Privacy and data security: The use of the algorithm to collect construction and worker data could raise privacy and security concerns regarding the personal and sensitive information that is collected.

# 11. ETHICAL IMPLICATIONS

The ethical issues involved in the design, production and use of the product are:

- Privacy issues: The algorithm requires some kind of camera to obtain the inputs, therefore it could inadvertently capture images or data of people or private property, which could raise privacy concerns.
- Accuracy and bias: Due to the short development time of the algorithm, the algorithm will not be fully optimized and reliable and may not identify errors accurately, leading to incorrect repairs or waste of resources. In addition, the algorithm may be biased or discriminate against certain groups or types of property, creating unequal treatment.
- Potential job losses: If the solution replaces human workers in error detection and line painting (in the case of SitePrint), it may lead to job losses for these workers.

In terms of damage caused by intentional, reckless or negligent use, on the one hand there are safety issues, because the algorithm is not fully optimized and tested in all environments, it can lead to unexpected bugs, robot overload, calculation errors, resulting in inaccurate repairs, leading to potential risks to the safety of people and property. On the other hand, there is material damage where the algorithm may suggest unnecessary repairs, may cause unnecessary work and material damage, both to the robot hardware (system overload) and to the building material.

The approach to communication with affected parties is as follows:

- Clear communication about the limitations and capabilities of the algorithm is required to ensure proper use.
- Privacy concerns may need to be addressed by providing information about data collection and use and obtaining consent.
- Clear communication with workers to address any concerns about potential job losses and to ensure that the solution is used in a way that complements human workers.

Other corrective measures include regular testing and quality control to ensure the accuracy and fairness of the algorithm, as well as providing training and resources for workers to adapt to new roles or develop skills complementary to the solution.

As information can be obtained not only from the construction area, appropriate measures should be taken to protect the privacy of individuals, such as obtaining informed consent, anonymising data or implementing security measures to prevent unauthorized access or data breaches.

# 12. CONCLUSIONS

The general conclusions of the project will be divided into three areas: the technical, the business and the sustainability.

In the first place, as far as the technical part is concerned, the conclusion has been drawn that this type of sensor should not be used since they are focused on object recognition or the use of the RGB camera and the sensor simultaneously. In addition, it has been found that in a wide variety of situations, including the most likely ones where this product is expected to be used, they do not work properly. An example would be that direct light or reflective surfaces prevent its use because the infrared rays are camouflaged by those of the environment. Another feature that limits this sensor is the distance at which it can be measured since its optimum range is from 0.3 to 3 meters ,and the planes for which this product was intended to be used exceed these limits by orders of magnitude.

On the business side, HP did not present a budget limit for the project. Therefore, the initial proposed budget estimate was 43.064€. Finally, this amount has increased to 63.317€ due to the fact that the initial budget was an approximation and did not take into account the specific costs of all the products used. Also, the business model canvas helped us to focus on the direction of the business we represent, what we do and the direction we are going to take.

In general, the sustainability part has been useful to deepen the superficial knowledge about the environmental, social and economic impact of a project regardless of its magnitude.
As a software project, we have studied the sustainability measures to be taken into account. We have mainly tried to minimize energy consumption, reduce physical resources and dispose of materials appropriately. In addition, we had the opportunity to check if the partner companies were environmentally friendly and if they also followed the RoHS directive. Furthermore, we have learnt the "cradle to cradle" concept, which we have implemented in our project by being more aware of the possibility of reusing all the machines used in our project. In fact, we also try to minimise the CO2 emissions of the transport used by our employees to go to the office by encouraging the use of public transport.

As for the cost savings of the project, we also try to save as much as possible by taking some measures to reduce the overall costs of the daily use of an office.

After this brief recapitulation of the measures studied to increase the sustainability of the project, we can conclude that there are many more aspects to take into account when trying to improve the sustainability of a software project than one might think at first glance. We have been impressed by how much environmental and social impact a project as small and limited as ours can have, leaving us uncertain about the impact that large industries can have, and even more so if they claim to be ignorant of sustainability analysis.
Also how many different aspects affect sustainability, from human resources to all the companies involved in the projects that provide raw materials or tools.

As far as the scope of the project and the stakeholders are concerned, as they were defined once we had a clear vision of the project, they have not been modified and no changes have been introduced. Some examples of stakeholders related to the project are: customers, employees, shareholders and investors or competitors.

On the risk side, there has not been much variation from those initially presented, and the ways to reduce their impacts have not been altered. The most worrying risk was the possibility of not being able to integrate it into the set of functions developed by the robot, however, given the problems presented in meeting the accuracy objectives, we were informed that the algorithm would be used to determine whether the camera was useful for more functions or presented many adversities.

# 13. REFLECTION DOCUMENT

The most important aspect that we have learned during the realization of this project has been the difficulty of understanding, defining and clarifying the objectives of the project. In addition to determining the relevance of each of them and that this is shared by both the company, in our case HP, and the team that will develop it.

Another lesson learned has been to know how to adapt to changes and possible unforeseen events, and that these do not paralyze us to move forward with the project and that its evolution is affected as little as possible.

As for the company, HP, the fact that they did not define the project precisely from the first day and delayed the explanation of the project by a week meant that we lost a whole week that we could have used to get to know the project better and make more progress on it.

On the other hand, they did not define clear objectives with an established range of priorities, in addition to changing these objectives with each passing week. This caused, to some extent, that some of the advances we made in the code we developed were unnecessary and were frustrated and replaced by new objectives. This caused us to lose hours of work, which with more determined objectives we would not have lost. In spite of this, we finally determined fixed objectives that allowed us to move forward with the project.

As for the team, from the beginning we defined a team leader who was in charge of managing the subgroups and organizing the tasks throughout the development of the project. In addition, he assigned tasks to each team member according to their capabilities so that they could give their best and contribute the most to the project as long as they felt comfortable in that position.

In addition, the entire team adapted easily to the changes in objectives provided by the project manager at HP. Although the work done so far was frustrating, this served to face the changes as a new challenge and an opportunity to give the best of ourselves instead of becoming demotivated. This made us work more and more efficiently and optimize processes at every step. Therefore, teamwork was really easy and dynamic thanks to the great atmosphere in the team and the good organization.

All in all, we feel very fortunate to have been able to be part of this challenge and we are aware of what a great opportunity it was to work with a company like HP.

# 14. ANNEX I: CODE

## 14.1 Main.py

```
import verification  as v

verif = v.Verification()

#DISTANCE DETECTED BY CAMERA
#dp,dc = verif.capture_frame(100, "prueba_1m.bag", 0)
dist = verif.dist_teorica('TestMap', (150,100), 270, True)
dp,dc,p = verif.show_files(26,"../test_1m.bag")

min, max, dist_cam= verif.PointCloud(dp,"test_1m")
print("Measured distance: " + str(dist_cam))
#verif.Segmentation("./gg_final/gg_final.ply",min,max)

print("The difference between the theoric and measured distance is: " +
str(abs(dist*10-((dist_cam)*1000))) + " millimeters.")
```

## 14.2 Class Verification.py

```
from Calibraton import Calibration
from frameCapture import Frame_Capture
from showFiles import Show_Files
from frameCapturePLY import Frame_Capture_PLY
from pointCloud import PointCloud
from segmentation import Segmentation
import numpy as np
import theoretical_distance_iterative as dt



class Verification:
    def __init__ (self):
        pass


    def calibration(self):
        calibrate = Calibration()
        calibrate.enable_stream()
        calibrate.calibrate()


    def capture_frame(self,frames, filename, high):
        frame_capture = Frame_Capture()
        frame_capture.get_and_configure_device(high)
        return frame_capture.capture(frames, filename, high)

    def capture_frame_ply(self,filename, high):
        frame_capture = Frame_Capture_PLY()
        frame_capture.get_and_configure_device()
        frame_capture.capture(filename, high)
```

```
    def show_files(self, frame,  filename):
        show = Show_Files()
        show.enable_file(filename)
        return show.show(frame)

    #def kmeans (self,matrix, k):
      #  km = Kmeans()
       # km.CalculateKMeans(matrix, k)

    def PointCloud (self, dp, filename):
        pc = PointCloud()
        return pc.createPointCloud(0.001*np.asanyarray(dp.get_data()),filename)


    def Segmentation (self, filename, min, max ):
        seg = Segmentation(filename,min,max)
        seg.removeBackgorund()
        seg.segmentation()

#TEORIC DISTANCE Test, Camera Position(X, Y), Angle(degree), Map Plot(Boolean))
    def dist_teorica(self, mapa, point, angle_degrees, show):
        d = dt.Theoretical_distance()
        dist = d.theoric_distance(mapa, point, angle_degrees,show)
        print("Distància teorica: " + str(dist))
        return dist
```

## 14.3 Class Calibration.py

```
import pyrealsense2 as rs
import numpy as np
import cv2


class Calibration:

    def __init__ (self):
        self.pipeline = rs.pipeline()
        self.config = rs.config()

    def enable_stream(self):

        # Get device product line for setting a supporting resolution
        pipeline_wrapper = rs.pipeline_wrapper(self.pipeline)
        pipeline_profile = self.config.resolve(pipeline_wrapper)
        device = pipeline_profile.get_device()
        device_product_line = str(device.get_info(rs.camera_info.product_line))

        found_rgb = False
        for s in device.sensors:
            if s.get_info(rs.camera_info.name) == 'RGB Camera':
                found_rgb = True
                break
        if not found_rgb:
            print("The demo requires Depth camera with Color sensor")
            exit(0)

        self.config.enable_stream(rs.stream.depth, 640, 480, rs.format.z16, 30)

        if device_product_line == 'L500':
```

```
            self.config.enable_stream(rs.stream.color, 960, 540, rs.format.bgr8, 30)
        else:
            self.config.enable_stream(rs.stream.color, 640, 480, rs.format.bgr8, 30)




    def calibrate (self):
    # Start streaming
        self.pipeline.start(self.config)
        depth_frame_set={}
        color_frame_set={}
        for x in range(20):
            # Wait for a coherent pair of frames: depth and color
            frames = self.pipeline.wait_for_frames()
            depth_frame_set[x] = frames.get_depth_frame()
            color_frame_set[x] = frames.get_color_frame()
            if not depth_frame_set[x] or not color_frame_set[x]:
                continue

            # Convert images to numpy arrays
            depth_image = np.asanyarray(depth_frame_set[x].get_data())
            color_image = np.asanyarray(color_frame_set[x].get_data())

            # Apply colormap on depth image (image must be converted to 8-bit per pixel first)
            depth_colormap = cv2.applyColorMap(cv2.convertScaleAbs(depth_image,
alpha=0.03), cv2.COLORMAP_JET)

            depth_colormap_dim = depth_colormap.shape
            color_colormap_dim = color_image.shape

            # If depth and color resolutions are different, resize color image to match depth image
```

```
for display
            if depth_colormap_dim != color_colormap_dim:
                resized_color_image = cv2.resize(color_image, dsize=(depth_colormap_dim[1],
depth_colormap_dim[0]), interpolation=cv2.INTER_AREA)
                images = np.hstack((resized_color_image, depth_colormap))
            else:
                images = np.hstack((color_image, depth_colormap))

            # Show images
            cv2.namedWindow('RealSense', cv2.WINDOW_AUTOSIZE)
            cv2.imshow('RealSense', images)
            cv2.waitKey(1)



        # Stop streaming
    self.pipeline.stop()


    f = open ("./output.txt", "w")
    f.write( depth_frame_set[1])
    f.close()
```

## 14.4 Class frameCapture.py

```
import pyrealsense2 as rs
import numpy as np
import cv2


class Frame_Capture:

    def __init__ (self):
        self.pipeline = rs.pipeline()
        self.config = rs.config()



    def get_and_configure_device (self, high):

        if(high<2):
            pipeline_wrapper = rs.pipeline_wrapper(self.pipeline)
            pipeline_profile = self.config.resolve(pipeline_wrapper)
            device = pipeline_profile.get_device()
            device_product_line = str(device.get_info(rs.camera_info.product_line))

            found_rgb = False
            for s in device.sensors:
                if s.get_info(rs.camera_info.name) == 'RGB Camera':
                    found_rgb = True
                    break
            if not found_rgb:
                print("The demo requires Depth camera with Color sensor")
                exit(0)

            self.config.enable_stream(rs.stream.depth, 640, 480, rs.format.z16, 30)
```

```
        if device_product_line == 'L500':
            self.config.enable_stream(rs.stream.color, 960, 540, rs.format.bgr8, 30)
        else:
            self.config.enable_stream(rs.stream.color, 640, 480, rs.format.bgr8, 30)


    else :
        self.config.enable_device('017322075378')
        self.config.enable_stream(rs.stream.depth, 1280, 720, rs.format.z16, 30)
        self.config.enable_stream(rs.stream.color, 1280, 720, rs.format.bgr8, 30)
        self.config.enable_stream(rs.stream.accel)
        self.config.enable_stream(rs.stream.gyro)


def capture (self, frames ,filename, high):

    self.config.enable_record_to_file(filename)

    # Start streaming
    pipe_profile = self.pipeline.start(self.config)

    if(high>2):
        depth_sensor = pipe_profile.get_device().first_depth_sensor()

        #Activem High Accuracy Mode (3); High Density (4)
        visualpreset = depth_sensor.get_option_value_description(rs.option.visual_preset,4)
        depth_sensor.set_option(rs.option.visual_preset, 4)

    # rs.recorder() #Start recording
    try:
        for i in range(frames):
```

```
            # Wait for a coherent pair of frames: depth and color
            frames = self.pipeline.wait_for_frames()
            depth_frame = frames.get_depth_frame()
            color_frame = frames.get_color_frame()
            if not depth_frame or not color_frame:
                continue

            # Convert images to numpy arrays
            depth_image = np.asanyarray(depth_frame.get_data())
            color_image = np.asanyarray(color_frame.get_data())

            if(i==10):
                fin_depth_image = depth_image
                fin_color_image = color_image

            # Apply colormap on depth image (image must be converted to 8-bit per pixel first)
            depth_colormap = cv2.applyColorMap(cv2.convertScaleAbs(depth_image,
alpha=0.03), cv2.COLORMAP_JET)

            depth_colormap_dim = depth_colormap.shape
            color_colormap_dim = color_image.shape

            # If depth and color resolutions are different, resize color image to match depth image
for display
            if depth_colormap_dim != color_colormap_dim:
                resized_color_image = cv2.resize(color_image, dsize=(depth_colormap_dim[1],
depth_colormap_dim[0]), interpolation=cv2.INTER_AREA)
                images = np.hstack((resized_color_image, depth_colormap))
            else:
                images = np.hstack((color_image, depth_colormap))
```

```
            # Show images
            cv2.namedWindow('RealSense', cv2.WINDOW_AUTOSIZE)
            cv2.imshow('RealSense', images)
            cv2.waitKey(1)

        finally:
            # Stop streaming
            self.pipeline.stop()
            return fin_depth_image,fin_color_image
```

## 14.5 Class pointCloud.py

```
import open3d as o3d
import cv2
import numpy as np
import os
import imageio.v3 as iio

class PointCloud:

    def __init__(self) -> None:
        pass


    def createPointCloud (self, depthframe, filename):


        os.makedirs(filename, exist_ok=True)


        # Load the depth frame as a 16-bit grayscale image
        depth_image = np.array(depthframe)
        height, width = depth_image.shape
        min =  np.min(depth_image)
        max =  np.max(depth_image)
        # Normalize the depth values between 0 and 255
        #print(height,width)
        depth_frame_norm = (((depth_image - min) / max)*1024) #225,2048,1024
        #print ("Dist: "+ str(depth_image [320][400]))
        # Convert the depth frame to a color map
        dist = depth_image[320][400]
```

```
        # Save the depth map as a PNG image
        cv2.imwrite("./"+filename + "/"+filename+'.png', depth_frame_norm)

        '''
        depth_image = cv2.imread("./"+filename + "/"+filename+'.png',cv2.IMREAD_ANYDEPTH)
        print(depth_image)
        hole_mask = depth_image == 0
        cv2.imwrite("./"+filename + "/"+filename+'_mask.png', hole_mask.astype(np.uint8)*255)

        # Apply the inpainting algorithm
        filled_depth = cv2.inpaint(depth_image, hole_mask.astype(np.uint8), 5,
cv2.INPAINT_TELEA)

        # Save the filled depth frame
        cv2.imwrite("./"+filename + "/"+filename+ '_filled.png', filled_depth)

       #depth_image = filled_depth.astype(np.uint16)

        #depth_o3d = o3d.geometry.Image(depth_image)
'''
       # print(depth_o3d)
        depth_image = iio.imread("./"+filename + "/"+filename+ '.png')



        '''
        # Create a depth image from the numpy array

        intrinsic = o3d.camera.PinholeCameraIntrinsic(640, 480, 616.962, 616.962, 329.215,
237.106)
        im = o3d.geometry.PointCloud.create_from_depth_image(depth_o3d,intrinsic)
        '''
        CX_DEPTH = 322.282 # 322.282
```

```
CY_DEPTH = 322.282

FX_DEPTH=320.818 #320.818
FY_DEPTH=178.779 #178.779

pcd = []
height, width = depth_image.shape
for i in range(height):
    for j in range(width):
        z = depth_image[i][j]
        x = (j - CX_DEPTH) * z / FX_DEPTH
        y = (i - CY_DEPTH) * z / FY_DEPTH
        pcd.append([x, y, z])
'''
for i in range (100):
    k=i

    if (i<200):

       pcd.append([0, 0, 2*i])
    for j in range(100):
        t=j
        if(i%2):
            k= i*(-1)
        if(j%2):
            t= j*(-1)
        pcd.append([k, t, 0])
'''

pcd_o3d = o3d.geometry.PointCloud()  # create point cloud object
pcd_o3d.points = o3d.utility.Vector3dVector(pcd)  # set pcd_np as the point cloud points
# Visualize:
```

```
    o3d.visualization.draw_geometries([pcd_o3d])


    #s = verif.matrixDiagn(dp)

    o3d.io.write_point_cloud("./"+filename + "/"+filename+".ply", pcd_o3d)
    return min, max, dist
```

## 14.6 Class showFiles.py

```
import cv2                          # state of the art computer vision algorithms library
import numpy as np                  # fundamental package for scientific computing
import open3d as o3d
import pyrealsense2 as rs


class Show_Files:

    def __init__ (self):
        self.pipeline = rs.pipeline()
        self.config = rs.config()
        self.pc = rs.pointcloud()




    def enable_file (self, file):


        rs.config.enable_device_from_file(self.config , file)
        self.config.enable_stream(rs.stream.depth, rs.format.z16, 30)
        self.config.enable_stream(rs.stream.color, 640, 480, rs.format.bgr8, 30)

        profile = self.pipeline.start(self.config)



    def show (self, frame):

        cv2.namedWindow("Depth Stream", cv2.WINDOW_AUTOSIZE)
```

```
pointcloud=rs.pointcloud()



# Create colorizer object
colorizer = rs.colorizer()
iter = 0
   # Streaming loop
while True:
      iter = iter + 1
      # Get frameset of depth
      frames = self.pipeline.wait_for_frames()

      # Get depth frame
      depth_frame = frames.get_depth_frame()
      color_frame = frames.get_color_frame()

      #points = pointcloud.calculate(depth_frame)

      #verts = np.asarray(points.get_vertices())
      #rgbs = np.asarray(color_frame.get_data())
      #pcd = o3d.geometry.PointCloud()
      #pcd.points = o3d.utility.Vector3dVector(verts)
      #pcd.colors = o3d.utility.Vector3dVector(rgbs)

       # Visualizar el pointcloud
      #o3d.visualization.draw_geometries([pcd])

      if(iter == frame-6):
          final_depth = depth_frame
          final_rgb = color_frame
      # Colorize depth frame to jet colormap
      depth_color_frame = colorizer.colorize(depth_frame)
```

```
# Convert depth_frame to numpy array to render image in opencv
depth_color_image = np.asanyarray(depth_color_frame.get_data())

# Render image in opencv window
cv2.imshow("Depth Stream", depth_color_image)
key = cv2.waitKey(1)
# if pressed escape exit program
if key == 27:
    cv2.destroyAllWindows()
    return final_depth, final_rgb, self.pc
```

## 14.7 Class frameCapturePLY.py

```
import pyrealsense2 as rs
import numpy as np
import cv2


class Frame_Capture_PLY:

    def __init__ (self):
        self.pipeline = rs.pipeline()
        self.config = rs.config()
        self.points = rs.points()
        self.spatial = rs.spatial_filter()
        self.temporal = rs.temporal_filter()
        self.hole_filling = rs.hole_filling_filter()


    def get_and_configure_device (self):

        self.config.enable_device('017322075378')
        self.config.enable_stream(rs.stream.depth, 1280, 720, rs.format.z16, 30)
        self.config.enable_stream(rs.stream.color, 1280, 720, rs.format.bgr8, 30)
        self.config.enable_stream(rs.stream.accel)
        self.config.enable_stream(rs.stream.gyro)
        self.spatial.set_option(rs.option.filter_magnitude, 2)
        self.spatial.set_option(rs.option.filter_smooth_alpha, 0.5)
        self.spatial.set_option(rs.option.filter_smooth_delta, 20)

    def capture (self,filename, high):
```

```
        # Start streaming with chosen configuration
        pipe_profile = self.pipeline.start(self.config)

        if(high>2):
            depth_sensor = pipe_profile.get_device().first_depth_sensor()

            #Activem High Accuracy Mode (3); High Density (4)
            visualpreset = depth_sensor.get_option_value_description(rs.option.visual_preset,4)
            depth_sensor.set_option(rs.option.visual_preset, 4)
        # We'll use the colorizer to generate texture for our PLY
        # (alternatively, texture can be obtained from color or infrared stream)
        colorizer = rs.colorizer()

        try:
            # Wait for the next set of frames from the camera
            frames = self.pipeline.wait_for_frames()
            colorized = colorizer.process(frames)
            depth = frames.get_depth_frame()
            depth = self.temporal.process(depth)
            depth = self.spatial.process(depth)
            color = frames.get_color_frame()
            # Create save_to_ply object
            ply = rs.save_to_ply(filename+ ".ply")
            # Set options to the desired values
            # In this example we'll generate a textual PLY with normals (mesh is already created by
default)
            ply.set_option(rs.save_to_ply.option_ply_binary, False)
            ply.set_option(rs.save_to_ply.option_ply_normals, True)

            print("Saving to 1.ply...")
            # Apply the processing block to the frameset which contains the depth frame and the
texture
```

```
        ply.process(colorized)
        print("Done")
    finally:
        self.pipeline.stop()
```

## 14.8 Class theorical_distance_iterative.py

```
import matplotlib.pyplot as plt
import numpy as np
import math
import ast

class Theoric_distance_iterative:

    def __init__ (self):
        pass

    def theoric_distance_iterativa(self, mapa, point, angle_degrees, show):
        #Carreguem mapa
        lines = self.loadMap(mapa)

        # Definim posició i angles (graus)
        #point, angle_degrees = (150, 100), 30

        #Ample de visió de la càmara d435i
        angle_FOV = 86/2

        #Calculem distancia central
        dist, interseccio_paret = self.distancia_propera(point, angle_degrees, lines)

        #Camp de visió càmara
        FOV_right, interseccio_paret_right = self.distancia_propera(point, angle_degrees -
angle_FOV, lines)
        FOV_left, interseccio_paret_left = self.distancia_propera(point, angle_degrees +
angle_FOV, lines)

        angles = array = np.arange(-angle_FOV, angle_FOV, 0.5)
        scalar_angles = [math.radians(angle) for angle in angles]
```

```
        d = []                      #Vector de distancies
        interseccio_paret_aa = []       #Vector de les interseccions
        distancia_max = 0               #inicialitzacio pel calcul de la distancia max
        distancia_min_c = 1500          #inicialitzacio pel calcul de la distancia min
        cares_columna = 0               #Indica el nombre de cares d'una columna que veu la camara
        estam_veiem_columna = False     #boolean que indica si en quin moment s'esta observant
la columna dintra de la iteracio
        index_columna_final = -1        #inicialitzacio per a la distancia de la columna
        index_columna_inici = -1        #idem
        index_min_c = 1500
        hem_estat_columna = False

        for i in range(len(scalar_angles)):
            distancia, interseccio = self.distancia_propera(point, angle_degrees + angles[i], lines)
            distancia = distancia*math.cos(scalar_angles[i])
            #print(i,distancia, angles[i])
            d.append(distancia)
            interseccio_paret_aa.append(interseccio)
            if distancia > distancia_max: #Guardem el valor i index de la distancia maxima i per tant,
cantonada
                index_max = i-1
                distancia_max = distancia

            if abs(distancia-d[i-1]) > 10: #Si hi ha salts de valors "grans" significa que la camara veu
una columna
                if not estam_veiem_columna:
                    #print("entrem de la columna",i)
                    index_columna_inici = i
                    estam_veiem_columna = True
                    hem_estat_columna = True
```

```
            else:
                #print("sortim de la columna",i)
                index_columna_final = i-1
                estam_veiem_columna = False

        if distancia < distancia_min_c and estam_veiem_columna: #Veiem la cantonada de la columna
            index_min_c = i
            distancia_min_c = distancia

    #La columna es recte, per tant si el valor maxim esta al principi o al final, ja sabem que la camara no veu
    #cap cantonada, en el cas que el maxim no coincideixi amb inici o final, sabem que hi haura una cantonada
    #si hi ha cantonada sabem que la camara veu dues cares de la columna
    if hem_estat_columna:
        if abs(distancia_min_c-d[index_columna_inici]) < 0.01 or abs(distancia_min_c-d[index_columna_final]) < 0.01:
            cares_columna = 1
        else:
            cares_columna = 2

    plt.plot(angles,d)
    plt.xlabel("Degrees")
    plt.ylabel("Distancia")
    plt.xlim(-45,45)
    plt.ylim(0, 300)

    #print(distancia_max,angles_distancia_max, index_max)


    #La paret es recte, per tant si el valor maxim esta al principi o al final, ja sabem que no
```

```
topem cap cantonada
    #En el cas que el maxim no coincideixi amb inici o final, sabem que hi haura una cantonada

    if(abs(distancia_max-d[0]) < 0.01 or abs(distancia_max-d[len(scalar_angles)-1]) < 0.01):
        veiem_cantonada = False
    else:
        veiem_cantonada = True

    #--------------------< CALCUL DE LES DISTANCIES FINALS >--------------------

    print("Faces the column:", cares_columna)

    if cares_columna == 0:
        if not veiem_cantonada:
            print("Wall seen")

            #Distancia mitja de la paret
            dist = sum(d) / len(d)

        else:
            print("Corner seen (2 walls)")
            mean_1_np = np.mean(d[:index_max-1],dtype=np.float64)
            mean_2_np = np.mean(d[index_max:],dtype=np.float64)

            #Distacia mitja de les dues parets

            if index_max > len(scalar_angles)/2:
                dist = [mean_1_np, mean_2_np]
            else:
                dist = [mean_2_np, mean_1_np]
```

```
        elif cares_columna == 1:
            print("Column seen")

            mean_columna =
np.mean(d[index_columna_inici:index_columna_final-1],dtype=np.float64)
            mean_paret_1 = np.mean(d[:index_columna_inici-1],dtype=np.float64)
            mean_paret_2 = np.mean(d[index_columna_final+1:],dtype=np.float64)

            #Distancia mitja de la de la columna i la dels dos trossos de paret
            if (len(scalar_angles) - index_columna_final > index_columna_inici):
                dist = [mean_paret_2, mean_paret_1, mean_columna]
            else:
                dist = [mean_paret_1, mean_paret_2, mean_columna]

        elif cares_columna == 2:
            print("2 faces of a column seen")

            mean_columna_a = np.mean(d[index_columna_inici:index_min_c-1],dtype=np.float64)
            mean_columna_b = np.mean(d[index_min_c:index_columna_final-1],dtype=np.float64)
            mean_paret_1 = np.mean(d[:index_columna_inici],dtype=np.float64)
            mean_paret_2 = np.mean(d[index_columna_final+1:],dtype=np.float64)

            #Distancia mitja de les dues cares (a i b) de la columna i la dels dos trossos de paret
            if index_columna_inici > len(scalar_angles) - index_columna_final:
                if index_min_c - index_columna_inici > index_columna_final - index_min_c:
                    dist = [ mean_paret_1, mean_paret_2, mean_columna_a,mean_columna_b]
                else:
                    dist = [ mean_paret_1, mean_paret_2, mean_columna_b,mean_columna_a]
            else:
                if index_min_c - index_columna_inici > index_columna_final - index_min_c:
                    dist = [ mean_paret_2, mean_paret_1, mean_columna_a,mean_columna_b]
                else:
```

```
                dist = [ mean_paret_2, mean_paret_1, mean_columna_b,mean_columna_a]



        #--------------------< REPRESENTACIÓ >--------------------
        if(show):
            #Creem mapa
            fig, ax = plt.subplots()
            # Setejem la nostra posició
            ax.scatter(point[0],point[1], label=f'Camera location: {(point[0], point[1])}' + ' cm.',
color='blue', marker='o', s=30)
            # Graficar mapa
            for i in range(len(lines)):
                x1, y1 = lines[i][0][0], lines[i][0][1]
                x2, y2 = lines[i][1][0], lines[i][1][1]
                ax.plot([x1, x2], [y1, y2], 'k-')

            #Mostrem distancia
            #ax.plot([point[0],interseccio_paret[0]], [point[1],interseccio_paret[1]], 'g-', label=f'Central
distance: {round(dist, 3)}' + ' cm.')
            #ax.scatter(point[0],point[1], label=f'Distance to object: {round(dist[0])}' + ' cm.',
color='green', marker='o', s=30)

            #Mostrem rang de visió de la càmara
            ax.plot([point[0],interseccio_paret_right[0]], [point[1],interseccio_paret_right[1]], 'r-',
label=f'Field of view.')
            ax.plot([point[0],interseccio_paret_left[0]], [point[1],interseccio_paret_left[1]], 'r-')

            ax.legend(loc='upper left')

            #Redimensionar finestra
            ax.set_aspect('equal')
```

```
        plt.title('Distance between HP SitePrint and the objective')
        plt.xlabel('X Axis (cm)')
        plt.ylabel('Y Axis (cm)')

        #Establim valor de la distància teòrica
        #plt.text(interseccio_paret[0], interseccio_paret[1], f"{round(dist, 3)}" + " cm", ha='right',
va='bottom')
        plt.text(point[0], point[1], f"{(point[0], point[1])}" + " cm  ", ha='right', va='bottom')



        plt.show()
        self.print_distance(dist)

    if cares_columna > 0:
        columna = 1
    else:
        columna = 0

    return dist, columna

  def line_intersection(self, line1, line2):
    x1, y1 = line1[0]
    x2, y2 = line1[1]
    x3, y3 = line2[0]
    x4, y4 = line2[1]
    det = (x1 - x2) * (y3 - y4) - (y1 - y2) * (x3 - x4)

    if det == 0:
        return (-1,-1) #si no hi ha interseccio
    else:
        px = ((x1*y2 - y1*x2)*(x3 - x4) - (x1 - x2)*(x3*y4 - y3*x4)) / det
```

```
            py = ((x1*y2 - y1*x2)*(y3 - y4) - (y1 - y2)*(x3*y4 - y3*x4)) / det
            if (x1-1 <= px<= x2+1 or x1+1>=px>=x2-1)and(y1-1<=py<=y2+1 or y1+1>=py>=y2-1) and
(x3-1 <= px <= x4+1 or x3+1>=px>=x4-1)and(y3-1<=py<=y4+1 or y3+1>=py>=y4-1):
                return (px, py)
            else:
                return(-1,-1)

    def distance_between_points(self, point1, point2):
        x1, y1 = point1
        x2, y2 = point2
        return math.sqrt((x2 - x1)**2 + (y2 - y1)**2)

    def loadMap(self, map_name):
        if '.txt' in map_name:
            map_name = map_name.replace(".txt", "")
        with open(map_name + '.txt', 'r') as file:
            lines = file.readlines()
            #Assignem a un vector els valors de archiu
            lines_strip = [line.strip() for line in lines]
            #Separem les coordenades dels punts per poder-hi accedir posteriorment
            lines_clean = [ast.literal_eval(linea) for linea in lines_strip]

            return lines_clean

    def distancia_propera(self, point, angle_degrees, lines):

        # Convert the angle to radians
        angle_radians = (angle_degrees*math.pi)/180

        # Calculate the coordinates of the end point of the ray
        dx = 1000 * math.cos(angle_radians)
        dy = 1000 * math.sin(angle_radians)
```

```
        ray_end = (point[0] + dx, point[1] + dy)

        # Check which line the ray intersects with
        intersection_points = []
        for line in lines:
            intersection_point = self.line_intersection((point, ray_end), line)

            if intersection_point[0] >= 0 and intersection_point[1] >= 0:
                intersection_points.append(intersection_point)

        # Find the closest intersection point to the point
        closest_intersection_point = None
        closest_distance = float("inf")
        for p in intersection_points:
            distance = self.distance_between_points(p, point)
            if distance < closest_distance:
                closest_intersection_point = p
                closest_distance = distance

        return(closest_distance, closest_intersection_point)

    def print_distance(self,dist):
        if isinstance(dist, float):
            print("Distance to the wall: ",dist,"cm")
        elif len(dist) == 2:
            print("Distance to the most seen wall: ", dist[0],"cm", "\nDistance to the least seen wall: ",
dist[1],"cm")
        elif len(dist) == 3:
            print("Distance to the most seen wall: ", dist[0],"cm", "\nDistance to the least seen wall: ",
dist[1],"cm", "\nDistance to the column", dist[2],"cm")
        elif len(dist) == 4:
            print("Distance to the most seen wall: ", dist[0],"cm", "\nDistance to the least seen wall: ",
```

```
dist[1],"cm", "\nDistance to the most seen face of the column: ", dist[2],"cm", "\nDistance to the least seen face of the column: ", dist[3],"cm")
```

## 14.9 Class segmentation.py

```
import numpy as np
import open3d as o3d
import matplotlib.pyplot as plt



class Segmentation:

    def __init__(self,filename,min,max) :
        self.pcd =   o3d.io.read_point_cloud(filename)
        #pcd =self.pcd.voxel_down_sample(voxel_size=0.02)  #down sampling por si imagen muy compleja
        self.pcd.remove_statistical_outlier(nb_neighbors= 10, std_ratio=2.0)
        o3d.visualization.draw_geometries([self.pcd])
        self.min = min
        self.max = max
        #pcd.remove_radius_outlier(nb_points=16, radius=0.05) # esta eliminacion tarda mucho si la imagen tiene muchos puntos



    def removeBackgorund(self):

        plane_model, inliers =self.pcd.segment_plane(distance_threshold=3.5, ransac_n=3,num_iterations=1000) #0.01. El q funcionaba bien era 0.008
        inlier_cloud = self.pcd.select_by_index(inliers)
        inlier_cloud.paint_uniform_color([1.0, 0, 0])
        outlier_cloud =self.pcd.select_by_index(inliers, invert=True)
        #o3d.visualization.draw_geometries([inlier_cloud, outlier_cloud])
        self.pcd = outlier_cloud

        depth_values_plane= np.asarray(inlier_cloud.points)[:,2]
```

```
        depth_value_plane= ((np.mean(depth_values_plane)/255)*self.max)+self.min
        print("distancia pared final: " + str(depth_value_plane))



        plane_model, inliers =self.pcd.segment_plane(distance_threshold=2.8,
ransac_n=3,num_iterations=1000) #2.8
        inlier_cloud =self.pcd.select_by_index(inliers)
        outlier_cloud =self.pcd.select_by_index(inliers, invert=True)
        inlier_cloud.paint_uniform_color([1.0, 0, 0])
        outlier_cloud.paint_uniform_color([1.0, 0, 0])
        o3d.visualization.draw_geometries([inlier_cloud])
        o3d.visualization.draw_geometries([outlier_cloud])
        self.pcd = outlier_cloud

        depth_values_plane= np.asarray(inlier_cloud.points)[:,2]
        depth_value_plane= ((np.mean(depth_values_plane)/255)*self.max)+self.min
        print("distancia suelo/objeto: " + str(depth_value_plane))




    def segmentation(self):

        with o3d.utility.VerbosityContextManager(
        o3d.utility.VerbosityLevel.Debug) as cm:
            labels = np.array(
              self.pcd.cluster_dbscan(eps=3.2, min_points=20, print_progress=True)) # inclinado: 3.2,
y min_p=50.     1.8

        max_label = labels.max()
        print(f"point cloud has {max_label + 1} clusters")
```

```
        colors = plt.get_cmap("tab20")(labels / (max_label if max_label > 0 else 1))
        colors[labels < 0] = 0
        self.pcd.colors = o3d.utility.Vector3dVector(colors[:, :3])
        o3d.visualization.draw_geometries([self.pcd])



        #Buscamos la region que nos interesa
        target_label= 0
        selected_indices = np.where(labels == target_label)[0]
        selected_pcd =self.pcd.select_by_index(selected_indices)
        max_pcd= selected_pcd


        for i in range(max_label):
            target_label = i
            if(target_label> 0):
                selected_indices = np.where(labels == target_label)[0]
                selected_pcd =self.pcd.select_by_index(selected_indices)

                if(i==1):
                    max2_pcd= selected_pcd

                if( len(selected_pcd.points) > len(max2_pcd.points) ):
                    max2_pcd = selected_pcd

                if( len(selected_pcd.points)> len(max_pcd.points)):  #
np.asarray(selected_pcd.points).size> np.asarray(max_pcd.points).size
                    max2_pcd = max_pcd
                    max_pcd= selected_pcd

        o3d.visualization.draw_geometries([max_pcd])
```

```
o3d.visualization.draw_geometries([max2_pcd])
max_points = np.asarray(max_pcd.points)
max_points2 = np.asarray(max2_pcd.points)

print("varianza:" + str(np.var(max_points2[:,2])))

print("varianza:" + str(np.var(max_points[:,2])))

if(np.var(max_points2[:,2])<np.var(max_points2[:,2])):
    pcd_final=max2_pcd
else:
    pcd_final=max_pcd



o3d.visualization.draw_geometries([pcd_final])

depth_values_obj= np.asarray(pcd_final.points)[:,2]
depth_value_plane= ((np.mean(depth_values_obj)/255)*self.max)+self.min
print(depth_value_plane)
```

# 15. ANNEX II: MATRIX OF SUSTAINABILITY ANALYSIS QUESTIONS

| | | Project Development | Exploitation | Risks |
|---|---|---|---|---|
| Environmental | I | I.EN.D. Estimate the environmental impact of the project. Have you considered how to minimize the impact, for example, by reusing resources? | I.EN.X. How will your solution environmentally improve current solutions? | I.EN.R. What limitations will the calculation of the environmental impact have? What will not be taken into account? What will be its scope? |
| Economic | I | I.EC.D. Estimate the cost of the project (human and material resources). | I.EC.X. What could economically improve your solution as regards current solutions? | I.EC.R. What limitations will the calculation of the economic impact have? What will not be taken into account? What will be its scope? |
| Social | I | I.SO.D. What do you think will personally bring to you the realization of this project? | I.SO.X. What could socially improve (quality of life) your solution as regards previous ones? | I.SO.R. What limitations will the social impact calculation have? What will not be taken into account? What will be its scope? |

Table 3a. Questions to answer in the PAE/INT sustainability analysis of your PAE/INT report. (I: Initial phase of the project definition))

| | | Project Development | Exploitation | Risks |
|---|---|---|---|---|
| Environmental | F | F.EN.D.1. Quantify the environmental impact of the project development. What steps have you taken to reduce the impact? Have you quantified this reduction? Does your design follow the cradle-to-cradle philosophy? | F.EN.X.1. What resources do you estimate will be used during the lifetime of the project? What will be the environmental impact of these resources? | F.EN.R.1. Could any scenarios that might increase the footprint of the project arise? |
| | | | F.EN.X.2. Will the project reduce the use of other resources? Overall, will the use of the project improve or worsen the ecological footprint? | F.EN.R.2. If you did the project again, could it be done with fewer resources? Can it be designed again with reused material? |
| | | F.EN.D.2. What is the origin of the raw materials and / or materials used? Do your suppliers publish environmental reports? | F.EN.X.3. When the life of the project comes to an end, what waste is generated? How the environmental impact of dismantling can be reduced? | F.EN.R.3. What have been the main limitations of the environmental analysis of your proposal? |
| | | F.EN.D.3. Do your suppliers follow the RoHS directive? Do your suppliers follow the RBA Code of Conduct? | F.EN.X.4. Could the project be carried out with less environmental impact? | |
| Economic | F | F.EC.D.1. Quantify the cost (human and material resources) of the project. What decisions have you taken to reduce the cost? Have you quantified the savings? | F.EC.X.1. What is the estimated cost of the project over its lifetime? Could this cost be reduced to make the project more feasible? | F.EC.R.1. Could any scenarios arise that may jeopardize the viability of the project? |
| | | F.EC.D.2. Is the estimated cost similar to the final cost? Justify the differences (lessons learned). | F.EC.X.2. Have you taken into account the cost of adjustments / updates / repairs over the life of the project? | F.EC.R.2. What have been the main limitations of the economic analysis of your proposal? |
| | | | F.EC.X.3. Would the dismantling of the project incur any additional costs? | |
| | | | F.EC.X.4. Could any other project benefit from the results of this one? | |
| Social | F | F.SO.D.1. Does this project involve significant reflections on the personal, professional or ethical standards of the people working in the project? Has inclusive and non-sexist language been used? | F.SO.X.1. Who benefits from the use of the project? Is there any group that may be adversely affected by the project? If so, to what extent? | F.SO.R.1. Could any scenarios arise to make the project detrimental to any particular segment of the population? |
| | | | F.SO.X.2. To what extent does the project solve the problem initially raised? | |
| | | F.SO.D.2. What is the current situation of the sector related to the project? | F.SO.X.3. Are other ways of implementing the project that lead to different social impacts? | F.SO.R.2. Could the project create any kind of dependency that might leave users in a weak position? |
| | | F.SO.D.3. Do the distributors, manufacturers, suppliers, and retailers meet public ethical or conduct codes? | F.SO.X.4 Does the project avoid biases, stereotypes and gender roles? F.SO.X.5. Have you considered the usability of your product for people with diverse needs (age, gender, sex, functional diversity, cultural diversity, etc.)? Are there barriers to using it? | F.SO.R.3. What have been the main limitations of the social analysis of your proposal? |

Table 3b. Questions to answer in the PAE/INT sustainability analysis of your PAE/INT report. (F: final phase of the project)

# 16. ANNEX III: ENVIRONMENTAL IMPACT TABLES

## Tables of Environmental Impact Estimation

| Electronic device | Quantity | Hours/week | Hours/month | Power usage (kW) | kg CO2/kWh | Impact (kg CO2) |
|---|---|---|---|---|---|---|
| Pc equipament | 8 | 6 | 72 | 0,58 | 0,43 | 143,6544 |
| Router | 4 | 25 | 300 | 0,012 | 0,43 | 6,192 |
| Movil | 11 | 25 | 300 | 0,008 | 0,43 | 11,352 |
| Projector | 1 | 3 | 36 | 0,374 | 0,43 | 5,78952 |
| Lapdop | 7 | 25 | 300 | 0,03 | 0,43 | 27,09 |
| Lights | 10 | 25 | 300 | 0,01 | 0,43 | 12,9 |
| | | | | | TOTAL | 207 |

| Comunication tools | Usage/week | Total usage (4 months) | Consumption (kW) | kg CO2/kWh | Impact (kg CO2 of 11 workers) |
|---|---|---|---|---|---|
| Email | 2 | 24 | 0,000001 | 0,43 | 0,00011352 |
| Github | 23 | 276 | 0,00004 | 0,43 | 0,0522192 |
| Jira | 6 | 72 | 0,00005 | 0,43 | 0,017028 |
| Drive | 25 | 300 | 0,00005 | 0,43 | 0,07095 |
| Google Meet | 3 | 36 | 0,25 | 0,43 | 42,57 |
| | | | | TOTAL | 43 |

| Movility | Quantity | Distance to work(km) | kg CO2/km | Total distance( km) | Impact( kg CO2) |
|---|---|---|---|---|---|
| Cars | 1 | 43 | 0,2113 | 688 | 145,3744 |
| Train | 4 | 28 | 0,07 | 448 | 31,36 |
| Metro | 6 | 50 | 0,0625 | 800 | 50 |
| Bus | 3 | 135 | 0,0437 | 2160 | 94,392 |
| TOTAL | | 256 | | 4096 | 322 |

| Commute | Distance to work (km) |
|---|---|
| Esteve | 62 |
| Felip | 61 |
| Isi | 43 |
| Elias | 35 |
| Victor | 10 |
| Marc | 12 |
| Xavier | 10 |
| Alejandro | 10 |
| Pablo | 0,5 |
| Llorenç | 1 |
| Oscar | 12 |
| Total | 256,5 |

## Tables of Final Environmental Impact

| Electronic device | Quantity | Hours/week | Hours/month | Power usage (kW) | kg CO2/kWh | Impact (kg CO2) |
|---|---|---|---|---|---|---|
| Pc equipament | 3 | 6 | 72 | 0,58 | 0,43 | 53,8704 |
| Router | 4 | 25 | 300 | 0,012 | 0,43 | 6,192 |
| Movil | 11 | 4 | 48 | 0,008 | 0,43 | 1,81632 |
| Projector | 1 | 1,5 | 18 | 0,374 | 0,43 | 2,89476 |
| Lapdop | 7 | 24,5 | 294 | 0,03 | 0,43 | 26,5482 |
| Lights | 12 | 25 | 300 | 0,01 | 0,43 | 15,48 |
| | | | | | TOTAL | 107 |

| Comunication tools | Usage/week | Total usage (4 months) | Consumption (kW) | kg CO2/kWh | Workers | Impact (kg CO2 of workers) |
|---|---|---|---|---|---|---|
| Email | 1 | 12 | 0,000001 | 0,43 | 1 | 0,00000516 |
| Github | 23 | 276 | 0,00004 | 0,43 | 6 | 0,0284832 |
| Jira | 6 | 72 | 0,00005 | 0,43 | 3 | 0,004644 |
| Drive | 25 | 300 | 0,00005 | 0,43 | 11 | 0,07095 |
| Google Meet | 1 | 12 | 0,25 | 0,43 | 11 | 14,19 |
| | | | | | TOTAL | 15 |

| Movility | Quantity | Distance to work(km) | kg CO2/km | Total distance | Impact( kg CO2) |
|---|---|---|---|---|---|
| Cars | - | - | - | - | 0 |
| Train | 5 | 62,4 | 0,07 | 998,4 | 69,888 |
| Metro | 7 | 58,6 | 0,0625 | 937,6 | 58,6 |
| Bus | 3 | 135 | 0,0437 | 2160 | 94,392 |
| TOTAL | | 256 | | 4096 | 223 |