



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería en Electrónica Industrial y Automática

**Bin-picking of empty beverage containers
using a sheet-of-light vision setup**

Autor:

Gutiérrez Ácimas, Alejandro

Mateos Pedraza, Ismael

Tutor:

Verheyen, Maarten

ACRO

Valladolid, Junio 2017.

Valladolid, Junio 2017.
TFG REALIZADO EN PROGRAMA DE INTERCAMBIO

TÍTULO: Bin-picking of empty beverage containers using a sheet-of-light vision setup.

ALUMNO: Gutiérrez Ácimas, Alejandro
Mateos Pedraza, Ismael

FECHA: 15/06/2017

CENTRO: ACRO

TUTOR: Verheyen, Maarten

Acknowledgements

There are many people we would like to express my gratitude to.

First of all, we have to thank the research group ACRO for their help during these months, especially Maarten Verheyen for his assistance, help and patience and for always answering all our doubts.

Secondly, our regards go to our Erasmus coordinator, Wim Claes, as well as our promoters in Belgium, Eric Demeester, and Spain, Javier García Ruiz, for their willingness to answer all our questions and provide us with all the information I needed.

We cannot forget to say thank you to all the people that have helped us from our country, Spain. Our huge gratitude to our families because their support, care and encouragement in every single moment during these five months.

Last but not least, we want to say thank you to all the students that have made of this experience one of the best ones in our life: all the Erasmus students in Hasselt with whom we have shared meals, trips, cycling routes and many other unforgettable moments.

Index.

List of figures	7
Abstract	10
1. Introduction	11
1.1 Description	11
1.2 Research objectives.....	12
1.3 Overview	13
2. 3D VISION FOR INDUSTRIAL ROBOTS.....	14
2.1 Introduction	14
2.2 3D vision techniques	15
2.2.1 Stereo imaging	15
2.2.1.1 Stereo vision - Principle.....	16
2.2.2 Projected textures.....	17
2.2.3 Motion stereo.....	17
2.2.4 Structured light	18
2.2.4.1 Sheet of light principle	18
2.2.5 Time of flight	19
2.3 Comparison between the different vision techniques.....	20
3. Used setup and technique	21
3.1 Introduction	21
3.2 Sheet of light	21
3.2.1 Sheet-of-light method.....	21
3.2.2 The Measurement Setup.....	22
3.3 System Installation	26
3.3.1 Description	26
3.3.2 Camera	26
3.3.3 Laser Line Projector.....	28
3.3.4 Positioning system	29
3.4 Connections and Configurations	32
3.5 Conclusion	38
4. Programming with Halcon.....	39
4.1 Introduction.....	39
4.2 Halcon.....	39
4.3 Calibration	40
4.3.1 Introduction.....	40

4.3.2	Camera calibration with its rules.	40
4.3.3	Calibration of the Light plane.	45
4.3.3	Calibration of the movement pose.	47
4.4	Main program.....	49
4.4.1	Introduction.....	49
3.3.1	Choice of the acquisition image mode and parameters definition.....	50
4.3.2	Image acquisition	50
4.3.3	Data files preparation.....	51
4.3.4	Develop of the front surface of containers detection algorithm	55
	<i>Procedure parameters definition</i>	57
	<i>Create detection zones</i>	57
	<i>Detect the rack</i>	57
	<i>Detect the containers shapes</i>	59
	<i>Find the centre of the container front surface</i>	60
	<i>Divide the rack into zones</i>	60
	<i>Choose a line that we want to pick</i>	61
	<i>Search for the selected container front surface.</i>	62
	<i>Clear extraction checks</i>	62
4.5	Conclusion	68
5.	Conclusion	69
5.1	Overview	69
5.2	Future work	69
	References.....	70
	Attachments.....	71

List of figures

Figure 1: Measure of the racks.....	11
Figure 2: Position ambiguity in the scene using a single 2D image.....	14
Figure 3: Basic principle of stereo vision.....	15
Figure 4: Ambiguities in the stereo matching.....	15
Figure 5: Segmentation and feature correspondence for stereo matching.	15
Figure 6: Stereo vision principle.....	16
Figure 7: Projected texture method.....	17
Figure 8: Real Scene and stereo 3Dmodel.	17
Figure 9: Structured light measurements using several light planes.....	18
Figure 10: Sheet of light set-up.....	19
Figure 11: Measuring the time system.....	19
Figure 12: Phase shift-base flight time system.	20
Figure 13: Disparity image: the disparity obtained from each profile is stored in a row of the disparity image.....	22
Figure 14: Example of a sheet of light technique.....	22
Figure 15: Vision setup of the system.....	23
Figure 16: Triangulation setup 1.	23
Figure 17: Triangulation setup 2.	24
Figure 18: Triangulation setup 5.	24
Figure 19: Triangulation setup 6.	25
Figure 20: With the camera being orthogonal to the object, an orthogonal projection of the object is possible: Left is an orthogonal camera view and Right is a perspective view.....	25
Figure 21: Photonfocus MV1-D2048x1088-3D03-760-G2-8 camera.	27
Figure 22: Single A/B Mode.....	28
Figure 23: Z-laser z12018s-f-660-lp45.....	28
Figure 24: AC Motor Vemat, type VTB 63B.	29
Figure 25: Aluminum linear guide rail.....	30
Figure 26: The frequency converter Omron j7 mini J7AZBOP.....	30
Figure 27: Limit switch Sick IME08-04NPSZW2S.....	31
Figure 28: PLC SIEMENS S7 300 CPU314C-2DP.....	31
Figure 29: Encoder DFS60B-S4PM10000.....	31
Figure 30: Relay RLY08-B.....	32
Figure 31: Three-wire sequence setup.....	32
Figure 32: Electric diagram of the frequency converter.	33
Figure 33: Configuration of the motor speed.	33
Figure 34: Electric diagram of the laser line projector.....	34
Figure 35: Electric diagram of the limits switch.	34
Figure 36: Electric diagram between camera and encoder.	35
Figure 37: Electric diagram of the relay card.	36
Figure 38: Contactors Diagram for stop process.....	37
Figure 39: Contactors Diagram for forward process.....	37
Figure 40: Contactors Diagram for reverse process.....	38
Figure 41: Halcon starting interface.....	39
Figure 42: Camera calibration code flow diagram.	41
Figure 43: Set camera parameters code.	41

Figure 44: Test camera performing code.....	42
Figure 45: RLY_OPEN code.....	42
Figure 46: Image taken while program is checking the laser-camera alignment.....	43
Figure 47: Images taken while program is covering the whole view.....	43
Figure 48: Image taken while program is waiting for Light plane caltab calibration plate detection.....	44
Figure 49: Final image for Light plane caltab 1 y caltab 2 after the detection.....	44
Figure 50: Obtaining camera parameters and pose.....	44
Figure 51: Light plane calibration code flow diagram.....	45
Figure 52: Image taken while program is waiting for Light plane laser calibration plate detection.....	45
Figure 53: Final image for Light plane laser 1 after the detection.....	45
Figure 54: Obtaining Light plane 1 pose.....	46
Figure 55: Getting the light plane pose.....	46
Figure 56: Movement pose calibration code flow diagram.....	47
Figure 57: Final image for DY-start after the detection.....	47
Figure 58: Final image for DY-stop after the detection.....	48
Figure 59: Results of defining the magnitude of the Y Axis.....	48
Figure 60: Final message of everything was saved correctly.....	48
Figure 61: Main program flow Diagram.....	49
Figure 62: Set the sheet of light parameters.....	52
Figure 63: Profile image taken in the middle of the trayectory.....	52
Figure 64: Getting the results images.....	52
Figure 65: Images resulting from X, Y (top), Disparity (middle) and Score (bottom) filter.....	53
Figure 66: 3D interactive representation of the rack.....	54
Figure 67: Detect containers code flow Diagram.....	56
Figure 68: Defining the zones where borders of the rack should be detected.....	57
Figure 69: Detecting the top border of the rack.....	57
Figure 70: Detecting right and left borders of the rack.....	58
Figure 71: Generating the smallest surrounding rectangle with any orientation.....	58
Figure 72: Detecting the containers front surfaces.....	59
Figure 73: Detecting the containers shapes, defining an area and then with vertical lines.....	59
Figure 74: Remarking the top borders of the containers.....	59
Figure 75: Joining the points and find the center point of the line.....	60
Figure 76: Generating the small bounding box surrounding the 3D model.....	60
Figure 77: Different divided parts of the 3D model. Front and middle at top, back and top at middle, right and left at bottom.....	61
Figure 78: Selecting the best located container.....	62
Figure 79: Creating a surrounding box of the front part of the selected container.....	62
Figure 80: Checking if there is enough space to take the box.....	63
Figure 81: Looking for the back border of the container.....	63
Figure 82: Looking for the left and right border of the container.....	64
Figure 83: Showing the 3D representation of the container.....	64
Figure 84: Checking if there is something on the top or in front of the container.....	65
Figure 85: Show the final representation, location and pose of the selected container.....	65

Figure 86: Show the final representation, location and pose of the next containers of this scan.	66
Figure 87: Differences in the detected area between first and second scanner.	66
Figure 88: Show the final representation, location and pose of the second scanner containers.	67

Abstract

The aim of this project is to develop a bin-picking setup for emptying metal racks filled with empty beverage containers. The main problem is that currently employees have to perform this task manually which is very demanding and hard physical work. They can only perform this job for a maximum of one hour and then they are assigned to other work.

This project focusses on the vision part and not so much on the robot picking part. One of the biggest difficulties regarding the vision aspect is that every container can be different in size and shape and every metal rack is stacked in a different order. This makes it impossible for the robot to follow a pattern to pick the containers.

This project was started by ACRO and our goal is to improve the algorithm for the detection of the different container. First the hardware was setup for moving the 3D vision system by programming a PLC, computer, encoder, and relay card. Next the vision system was calibrated using a calibration plate so calibrated 3D scans of the whole container could be taken. Calibration of the vision system consists out of three main parts. The camera is calibrated, the light plane (positioning of the laser relative to the camera) and a movement position.

Finally, the algorithm currently available was analyzed and improved to get a better detection of the separate containers. It is necessary to get a good detection of the position and orientation of the container so in a final step a robot is able to pick the container.

Keywords

Rack / Container / 3D / Sheet-of-light / profiles

1. Introduction

1.1 Description

This current project has been developed in ACRO research center. The aim of the project is developed a bin-picking setup for emptying metal racks filled with empty beverage containers.

There are many companies that engage of warehouse and distribution of empty beverage containers. They usually use metal racks for a more easy and safe transport. The main problem is that handling the containers for a long time is a heavy work. Nowadays emptying the racks is performed manually. Employees perform different depending on health, personal problems, and number of worked hours. Because of that it is difficult to assure the same results every day. As well as workers need to take rest and vacation decreasing their productivity, however an automatic process could run without stops.

Thus, it could be interesting to automate this process. Is here where our project makes sense and acquires a great value.

To define the problem, first step of the research is analyzed the problem:

Each bottle brand has a different shape of container, also there are two different heights of racks and each rack has more than one columns and rows of stacked containers.

This makes it impossible to follow a predefined pattern for the robot to empty the beverage racks.

The measurements of the racks are displayed in figure 1:



Figure 1: Measure of the racks.

The displayed number between brackets in figure 1 is the actual size between the tubes of the rack. (Tube diameter is +/- 3cm)

In addition, there is also a rack with an intermediate platform, that it has the same size as the big which in the figure 1 is the right one.

All of these differences should be taken into account when developing the hardware setup and the programming code.

1.2 Research objectives

Making a deep analysis about the previous section, it could be interesting to develop an automatic system that is both economically friendly and user friendly so that it could be installed by ordered company. The aim of the project is to automatize the process of picking the containers from the metal racks.

First the hardware will be setup for moving the 3D vision system by programming a PLC, computer, encoder, and relay card. Next the vision system will be calibrated using a calibration plate so calibrated 3D scans of the whole container could be taken. Calibration of the vision system consists out of three main parts. The camera will be calibrated, the light plane (positioning of the laser relative to the camera) and a movement position

Using a 3D technique, it is possible to reconstruct the 3D object in form of a point cloud. This point cloud can be directly compared with a CAD model of the object that is being inspected. This pose detection method is called surface-based matching and is the one that has been used in this project.

For these tasks the equipment that is available at ACRO consists on:

- 3D sensors
 - Photonfocus MVI-D2048x1088-3D03-760-G2-8 camera is used in the SOL setup.
- A laser line
 - Z-laser z12018s-f-660-lp45
- A Siemens PLC
 - CPU314C-2D
- Relay card
 - USB-RLY08-B
- Limit switch
 - Sick IME08-04NPSW2S
- Frecuency converter
 - Omron J7 mini j7AZB0P2
- Encoder
 - SICK DFS60B- S4PM10000
- Motor to move the equip
 - VEMAT VTB 63-2A
- Software: Halcon 12.1 to develop the code

1.3 Overview

The project work is divided into following main sections:

- 3D Vision for industrial robots: This section includes a literature study on the possible methods that exists designed for the purpose.
- User Setup and techniques: The technology used in this thesis is described: vision sensors, installation, and communication.
- Programming with Halcon: Developed code divide in two parts: calibration, and the detection algorithm.
- Conclusions.

2. 3D VISION FOR INDUSTRIAL ROBOTS

2.1 Introduction

Analyzing the rack, it can be seen like a 3D workspace. This environment is perfect to use a robot because it can move in three dimensions. The biggest challenge with robotic bin picking is quantifying the operation's cycle time. With most traditional robotic applications, the cycle time will be very repeatable. This gives us a better productivity than a worker's productivity.

If we can mix a robot with a recognize system, we would create a complete bin picking system, which is able to detect and know the orientation before pick the object.

These systems are fast becoming an enabling technology for the automation of many different processes within many different industries.

For the recognition part, several techniques are available, both 2D and 3D.

The so-called shape-based matching method was used to find the object in an ordinary 2D image: the image is compared to a shape model that consists of 2D projections of a 3D object seen from different points of view.

This is a quite fast method because it only needs the acquisition of a single ordinary image. Moreover, it is also an economical approach since the required equipment consists of a common camera. Nevertheless, the matching between 2D and 3D features is not always easy and relies on a correct detection of the edges of the object in the image.

This makes the process sensitive to changes in illumination, noise and changes in inclination or scale, for instance. Another drawback is that although simple objects could be correctly found, this approach is not suitable when more complex objects are being inspected. Overall, it is not a method robust enough for industrial applications.

Because of these reasons, a 3D technique has been adopted in this thesis to enhance the results.

The problem that is found when a 3D measurement is to be taken using a single common camera is shown in Figure 2. There are several objects which have different dimensions, which depend on where they are placed. They have the same representation in the image. Due to this ambiguity, it is not possible to determine the position of the object in the scene.

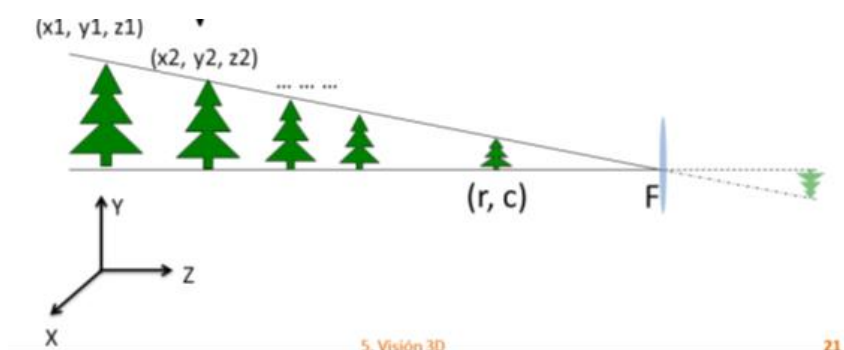


Figure 2: Position ambiguity in the scene using a single 2D image.

There are several methods available nowadays to solve this problem. The information below has been mainly obtained from [1].

2.2 3D vision techniques

2.2.1 Stereo imaging

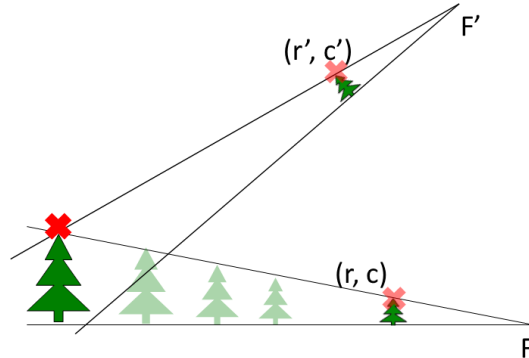


Figure 3: Basic principle of stereo vision.

This method requires the use of 2 cameras which are used to capture 2 images of the scene from 2 different viewpoints. The triangulation principle makes possible to calculate the depth of each point that can be recognized in both images. The further the point is, the smaller the difference of its position in both images is.

The main difficulty is to determine the pair of points in each image that correspond to the same point in the real scene. This process is called stereo matching and some of the difficulties encountered are shown in figure 4.

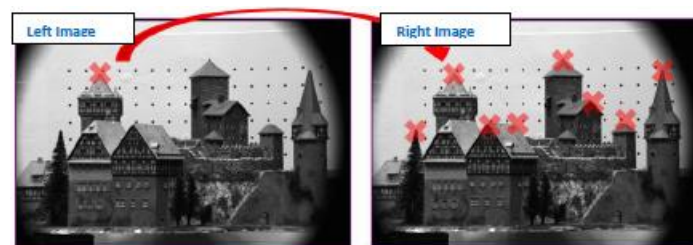


Figure 4: Ambiguities in the stereo matching.

A good way to fix this, is finding similar features in both images (Figure 5). However, this is not always possible if large regions without texture regions appear in the image and the stereo matching fails to find corresponding points.

To address this problem projectors are used in many cases. Knowing the pattern that is projected it is getting easier to find resembling points between the 2 images. For example, the "ENSENSO" from IDS is using this principle

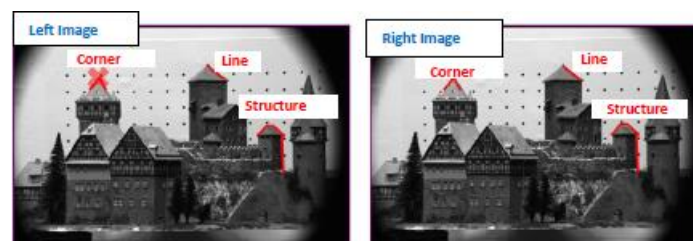


Figure 5: Segmentation and feature correspondence for stereo matching.

The relative positions of the cameras, as well as their internal parameters, should be precisely calculated. Because of this reason, it is important to have 2 cameras which are already calibrated.

Disparity values obtained are directly proportional to the distance the cameras are apart and inversely proportional to the distance an object is away from the cameras. Thereby, the only way to improve the accuracy is placing the dual cameras closer to the object to be examined or using two independent cameras and augmenting the distance between them.

2.2.1.1 Stereo vision - Principle

The following information appears in one of Halcon's manuals [2].

Using this approach, it is possible to get 3D information from a scene employing 2 common cameras which take pictures from different points. Comparing both images it is possible to determine the depth of a point in a scene.

The principle can be explained considering two parallel looking 1D cameras with identical parameters (Figure 6).

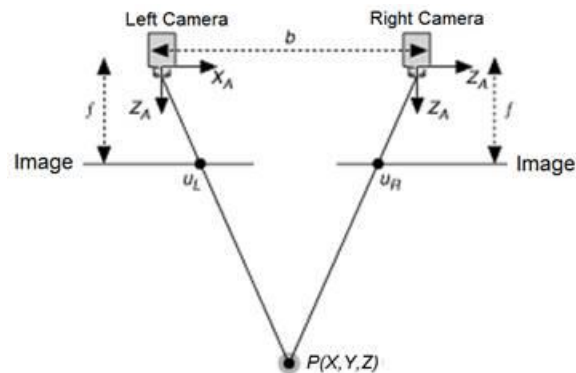


Figure 6: Stereo vision principle.

The image plane coordinates of the projections of the point $P(x^c, y)$ into the 2 images is expressed by:

$$u_1 = f \frac{x^c}{z^c}$$

$$u_2 = f \frac{x^c - b}{z^c}$$

Being f is the focal length and b the length of the basis (line that connects the two optical centers).

The difference between the two image locations of this point P is called the disparity distance:

$$d = (u_2 - u_1) = -f \frac{b}{z^c}$$

The camera parameters and the image coordinates of the two conjugate points are given, the z^c coordinate of point P . i.e., its distance from the stereo system, can be calculated by the following:

$$z^c = -f \frac{b}{d}$$

From the above formula, we conclude that the following points should be known:

- The relative pose of the second camera in relation to the first one.
- Internal camera parameters.
- The conjugate points.

The system calibration achieves the two first goals. The stereo matching process does the last task, getting a normal calibration and then a stereo calibration.

Multi-View Stereo Vision: This method uses more than two cameras. Though it is internally based on binocular stereo vision in contrary it doesn't return disparity image as a result.

2.2.2 Projected textures

Projected texture is a method used to add an artificial texture on very homogeneous surfaces where it is not possible to find correspondence between features. Normally a random pattern is projected in the scene and the images are taken by cameras. Then, it is possible to find the same projected point in both images and consequently the same point in the object.

Stereo vision can be used along with active lighting. The projected pattern helps perform the stereo matching. Then, it is the same the process to determine the depth of the point in the scene as in the common stereo vision. This method gives a better result when the object that is being rebuilt does not have well defined edges.

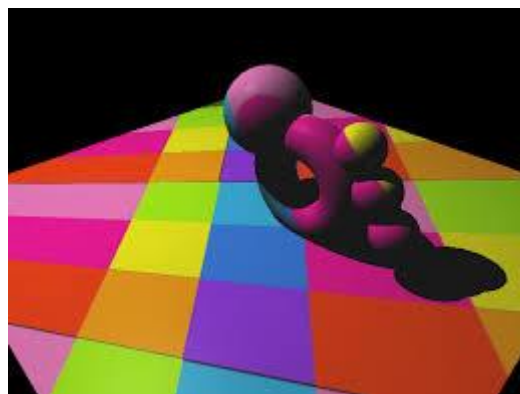


Figure 7: Projected texture method.

2.2.3 Motion stereo

The principle to obtain the disparity image is the same as in the first method. The difference is that only one camera is used. This camera is mounted on a robot and takes two or more images of the scene from different positions. One drawback is that the positioning of the robot has to be extremely precise. The main advantage is that generally the distance between the positions in which the images are taken is bigger than in the case of preconfigured dual cameras allows to obtaining a higher accuracy, like it has been said that the disparity values are higher when the distance between pictures increases.



Figure 8: Real Scene and stereo 3Dmodel.

2.2.4 Structured light

To generate a 3D image using structured light, a camera records the projected laser beam which is reflected from the object's surface. In this approach, it is essential to know the relative position between the camera and the laser, which permits to calculate the 3D coordinates of the projected laser beam by triangulation.

For structured light, often a projector is used which projects different known patterns onto the object. Therefore a movement is not needed the system moves across the field of view of the object, neither X, Y, Z coordinates are calculated neither used to generate a point cloud and disparity image of the external surface of the object. That is actually, one of the main advantages of structured light.

The laser/camera system can be configured separately or forming part of a pre-calibrated setup. The calibration of the setup must be very precise in every case. Any camera can be used to take the image of the reflected laser line. However, because of great variations in the lighting the use of cameras with high dynamic range is suggested.

Not only can a laser line be used, it is used also several planes at the same time, as shown in Figure 9.

Moreover, this would make the process faster. The use of more than one plane or cameras could also help reduce the occlusions and get a better detection.

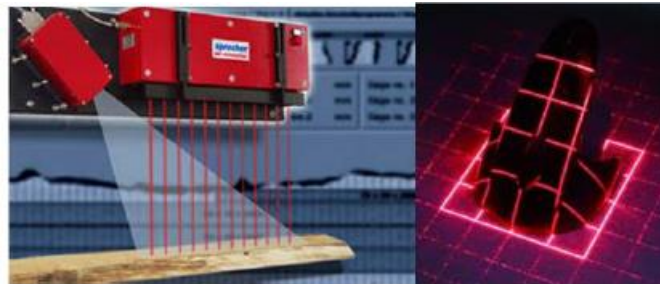


Figure 9: Structured light measurements using several light planes.

2.2.4.1 Sheet of light principle

The basic idea of the sheet-of-light technique is that a laser projector emits a line or multiple lines of laser light which is projected onto the scene to be scanned. The projected laser line is getting deformed by the to be scanned object. The amount of deformation then can get you the required depth information. Objects that are higher will offset the line from the base line by an amount equal to their height, as can be seen in figure 10. This offset from the baseline can be classed as a disparity from the normal and in the same way as above can be used to calculate the depth, i.e. the height of a given object. By scanning the object with the laser line, it is easy to obtain a full profile of the object being scanned.

The sheet-of-light technique can be applied either to a calibrated measurement setup or to the un-calibrated setup. If the setup is calibrated, the measurement returns the disparities, the x, y, and z coordinates of the points that build the profiles in the world coordinate system (WCS, see figure 10), and a 3D object model that is derived from the x, y, and z coordinates.

The angle that is formed by the optical axis of the camera and this plane is called angle of triangulation. The points of intersection between the object and the laser plane depend on the

height of the object. In consequence, if the object that is being scanned differs in height the laser is not seen as a line but as a profile image of the object. Only information about the points of the surface that are being illuminated by the laser can be obtained. Therefore, it is necessary to have a relative movement between the object and the measurement setup, that is to say, the camera and the laser, in order to reconstruct the complete object.

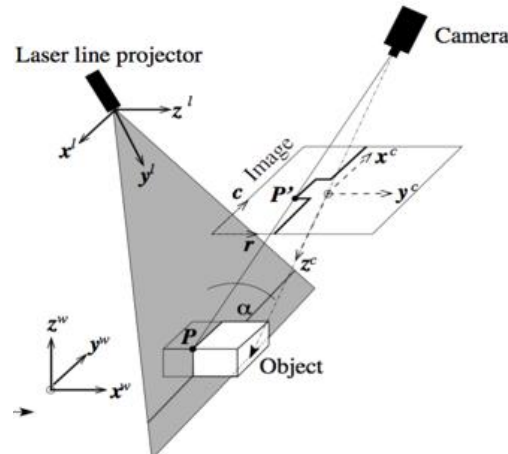


Figure 10: Sheet of light set-up.

The relation between the laser, the camera and the positioning system must not be changed or the system should be recalibrated. Only a relative movement of the object in relation to the vision set up must take place.

The result of the measurement can be affected by several conditions. Some of the factors that can have influence are the extracting algorithm, the quality of laser projector, the characteristics of the lens, the ambient lighting and the scattering properties of the material that the object is made of.

It is necessary to decide which options are more appropriate for a specific application in order to guarantee that the required quality of the results is achieved.

2.2.5 Time of flight

Previous techniques require the use of illumination sources and solid-state imagers as well as a dark work environment to get better results. Objects within a scene are illuminated by a modulated light source. There are two possible methods that can be used to perform the measurement.

1. The first method consists on measuring the time that it takes the light to go and come back.

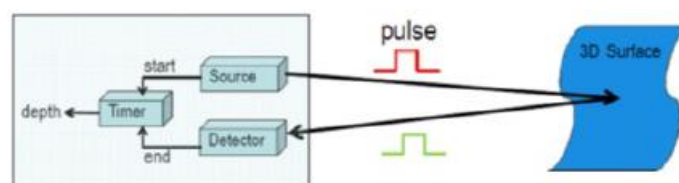


Figure 11: Measuring the time system.

2. In the second one, the phase shift between the illumination and the reflection beam is measured in order to determine the distance. The object is either illuminated by a pulsed mode or continuous wave and various methods can be used to determine the phase shift.

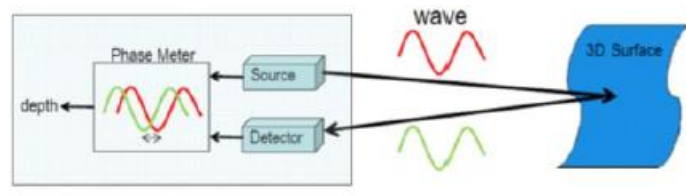


Figure 12: Phase shift-base flight time system.

2.3 Comparison between the different vision techniques

Having this theoretical knowledge about the available techniques, following schedule show all favor and drawbacks of each one.

Stereo imaging: This mode has de capacity of have a fast images acquisition however has problems like that finding corresponding points may not possible without textures surfaces or like it has a low precision.

Motion Stereo: This mode has as well a good capacity of fast images acquisition. Moreover, it has also precision that can be improved only increasing the distance between successive pictures. Unfortunately, it has the same problem as Stereo vision method with the affix of it requires high accuracy in the pose of the robot.

Textured projected stereo: This mode has same problems as Motion Stereo mode; although it has de possibility of performing the stereo matching even in scenes without texture. This method has de inconvenient that the result is more precise but the accuracy is poor, especially in object borders.

Sheet of light: This mode has de capacity of have a precision of the reconstructed object and a fast and easy computation of each profile picture. However, it has a slow process because several pictures per object should be taken and the accuracy of the calibration and positioning system are decisive.

Structure light: This method has the advantage of an easy segmentation and analysis as well as a high precision. It is one of the fastest methods because it can have a full 3D scan in like 2 seconds.

Time of flight: this method is much recommended when it is a long distance to measure or a lit-up work environment because it is not sensitive to light ambient. Nevertheless, it has a lower precision in compeer with the other modes.

3. Used setup and technique

3.1 Introduction

This chapter includes a description of the method and set of devices used in the setup. Firstly, the used technique will be explained in detail. Secondly, a description of the different components and devices used in the vision system will be given, including the communication and programming between the different parts.

3.2 Sheet of light

To achieve the goal of 3D object surface reconstruction by approximating it via a grouping of height profiles, laser triangulation technique can be used. Sheet-of-light technique is a special type of laser triangulation that HALCON provides by mean of some operators.

3.2.1 Sheet-of-light method

There are different laser techniques; structured light triangulation and sheet of light triangulation are considered the fastest when acquiring 3D data information from an object in real time. Because we have the necessary material, it was decided to use the second technique. Sheet of light is a technique that is used to get 3D model, used in machine vision applications. This technique consists of projecting a bright straight line, normally generated by a laser projected light line with a camera. A laser line is projected onto the scene to be scanned and is observed using the camera. The objects which are in the scene deform the shape of the laser stripe and this is what is called light plane or sheet of light.

The optical axis of the camera and the light plane form an angle, which is called angle of triangulation, α . The height of the object depends on the points of intersection between the camera view and the laser line. Thus, the line is not shown as a straight line however it represents a profile of the object if the object onto which the laser line is projected differs in height. We can get the height differences of the object using this profile image.

To reconstruct the whole surface of an object, the object must be moved relative to the setup (camera and laser). Sheet of light method can be applied either to a calibrated measurement setup or to uncalibrated setup. If the camera and the light plane are calibrated, the setup returns the disparities, the x , y , and z coordinates of the points that build the profiles in the WCS, world coordinate system. And at the same time from the x , y , and z derive a 3D object model.

The disparities are returned by means of a disparity image; the disparities of each measured profile are stored in one row of the disparity image. Because of this, the camera must be oriented such that the profiles are roughly parallel to the rows of the image, as it can be seen in the figure 13. The x , y , and z coordinates are not returned like values however they are expressed as values of pixels within images.

Three images are returned, one for each coordinate (X , Y and Z). Mapping 2D images are obtained the 3D coordinates for make the 3D object model. If the setup was uncalibrated, only

the disparity image and score image that describes how reliable the measurement result is that can be returned.

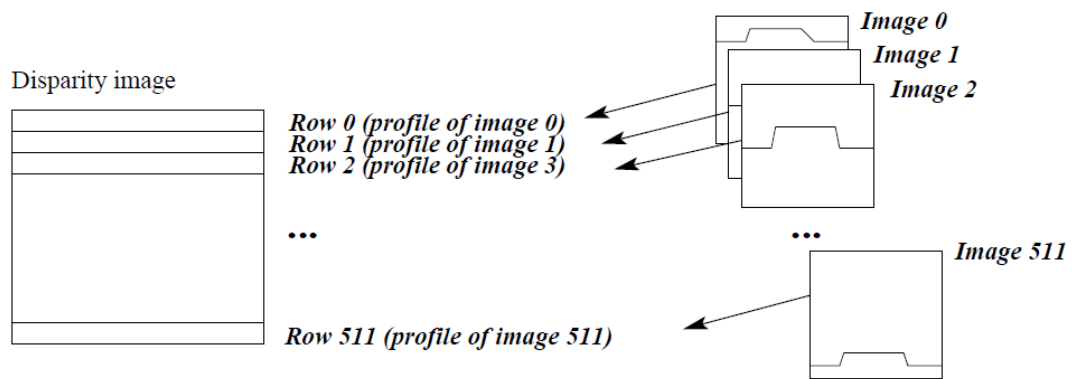


Figure 13: Disparity image: the disparity obtained from each profile is stored in a row of the disparity image.

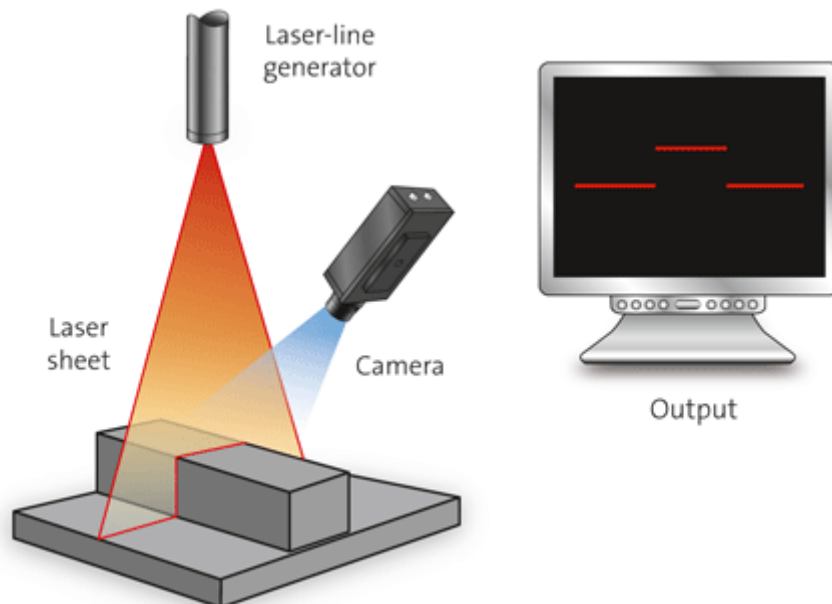


Figure 14: Example of a sheet of light technique.

3.2.2 The Measurement Setup

The hardware needed for a sheet-of-light measurement is composed of a projector that is able to project a thin luminous line, a camera, a positioning system, and the object to scan. For the present thesis, the projector is a laser line projector and the positioning system is a linear guide rail that is moved using a motor with a belt and that the vision setup (laser and camera), very common in laser triangulation applications. This setup is displayed in the figure 15.

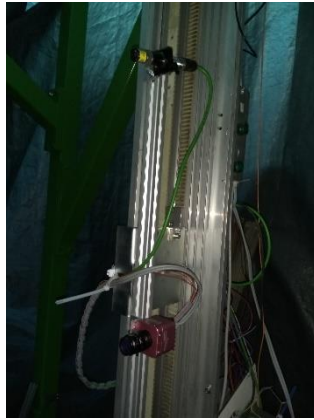


Figure 15: Vision setup of the system.

The relation between the projector, the camera, and the linear positioning system must not be changed, whereas the position of the scanned object that is transported by the positioning system changes in relation to the projector-camera unit. In our setup, this object is an aluminum bracket. Due to the profile images are processed column by column. The profiles must be oriented roughly horizontal, parallel to the rows of the image.

In order to create a 3D model the triangulation setup contains a laser line generator and the camera, but there are several configurations with them. Depending of the scattering of the material to be scanned will be selected a setup used in the application. There are setups for highly scattering and for nearly reflecting surfaces. The accuracy of the triangulation system is determinate by the line created by the algorithm, the optical setup, the parameters of the lens and the parameters of the laser line.

According to the possible setup before exits the following triangulation setup:

- Triangulation setup 1: In this setup, the camera has an angle α respect to the laser line projected from the top. A larger angle provides a higher resolution, but the range of height is reduced. With small angles is reduced the occlusions. This setup is represented in the following image.

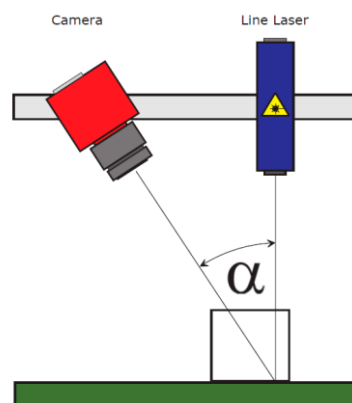


Figure 16: Triangulation setup 1.

- Triangulation setup 2: in contrast to the previous setup, the laser has an angle α respect to the camera. With this setup, the resolution at same triangulation angle is a little higher but is possible that appears artifacts during the measurement at the borders of the object. This problem must be suppressed by software. This setup is represented in the following image.

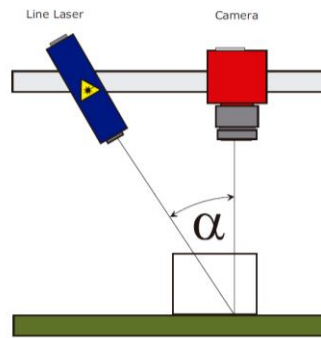


Figure 17: Triangulation setup 2.

- Triangulation setup 3: in this setup, the camera and the laser line projector are located with a slight inclination. This setup is used for dark or matte surfaces because gives a better signal. Also, can be used for reflecting surfaces due to this configuration help to get good results.
- Triangulation setup 4: this setup is the best option for high scattering material or objects with strong reflections. The resolution is reduced due to the relations of the angles.
- Triangulation setup 5: due to this camera has the possibility to detect two laser lines, this triangulation setup is used to avoid occlusions. The angles of the laser lines could be equal or different to reduce the calibration effort. . This setup is represented in the following image.

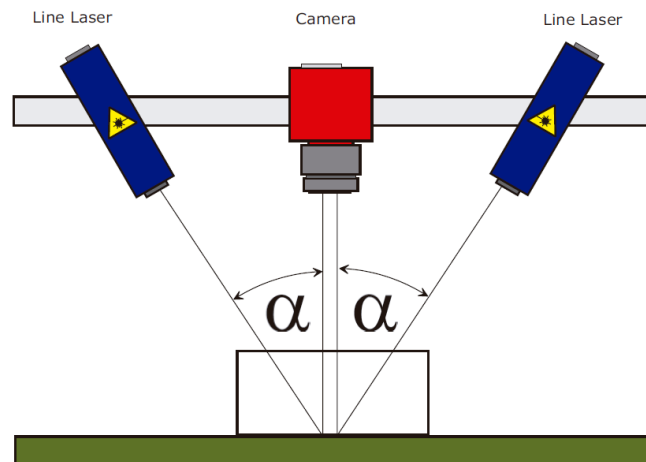


Figure 18: Triangulation setup 5.

- Triangulation setup 6: by means of the detection of two lasers lines this camera gives the possibility to measure glasses or other transparent materials. The different reflexes which occur could suppress determining an appropriate ROI and geometric arrangements. This setup is represented in the following image.

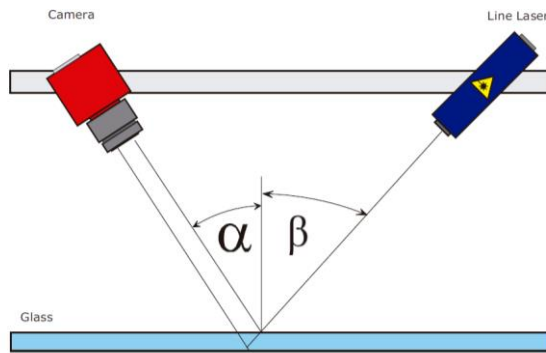


Figure 19: Triangulation setup 6.

- Triangulation setup 7: using the detection of two lasers lines can be suppressed vibrations and elastic deformations of the scanned object. This method is often performed when there are fast profile scanners where the profile direction is on the limits. The camera detects this vibrations and elastics deformations like height gradients of two laser lines. This fact can be used with software algorithms to compensate vibrations effects and get better measurements.

The use of a determinate configuration depends of the purpose of the measurement and the geometry of the object. For example, the configuration of a title camera view and an orthogonal light plane is suitable if is needed an orthogonal projection of the object in the image, resulting a cuboid is a rectangle. If the camera is not placed orthogonal, the image would be a trapezoid because of the perspective deformations.

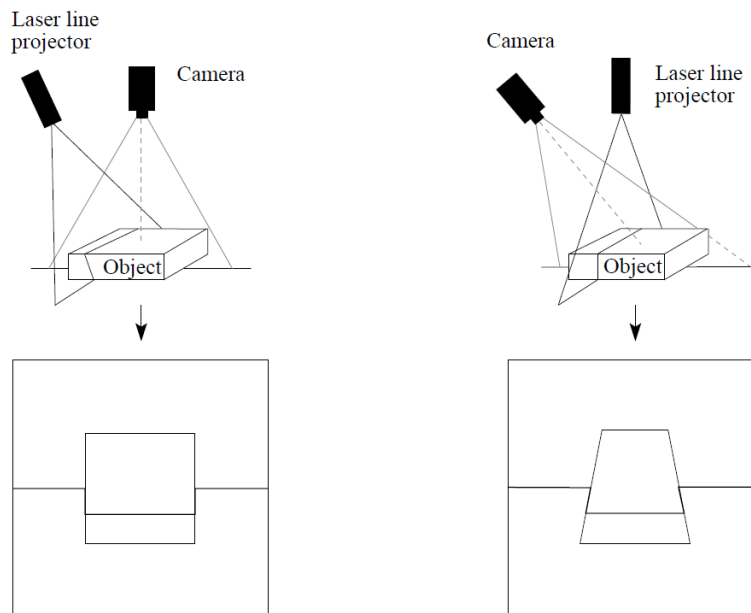


Figure 20: With the camera being orthogonal to the object, an orthogonal projection of the object is possible: Left is an orthogonal camera view and Right is a perspective view.

The geometry of the object is the most important criterion for the choice of the measurement setup. The configuration should be chose trying that the amount of shadowing effects and occlusions is minimized. If an object point is illuminated by the laser line but is not visible in the image, occlusions are happened, because other object parts are between the object point and the line of sight of the camera. Shadowing effects appear when an object point is visible in the image however it is not illuminated by the laser line, because other object parts are between the object point and the laser projector.

For these three setup configurations, the triangulation angle, that is to say, the angle between the optical axis of the camera and the light plane should be in a range of 30° to 60° to achieve a good measurement precision. If the angle is smaller, the accuracy decreases. Conversely if the angle is bigger, the accuracy increases, however more problems because of occlusions and shadowing effects will appear. Therefore, a balance must be found between the accuracy and the entirety of the measurement.

Moreover, it is also important that the precision decreases if the light plane is out of focus of the camera. If the triangulation angle is too small maybe this could happen, that is to say, the angle between the focus plane and the light plane is too large, and therefore only a small part nearby the intersection of both planes is in focus.

Having a unique laser line projector has been chosen the configuration number two. Because it offers a higher resolution than the first setup and later with an algorithm will be selected an area where the edges of the images are not used since they are not those that offer exact measurements.

3.3 System Installation

3.3.1 Description

A measurement setup is composed of a laser, a camera, and a positioning system. Because each component has a certain function, it must be adapted to our system to obtain the best results. Each of them will be described in detail below.

3.3.2 Camera

In the industrial sector, with a long list of options it difficult to choose the perfect camera but, it must be selected one that meets the following requirements. First, the camera resolution must be able to cover all the workspace and with high quality. Secondly, the camera should be compatible with the vision program which we use. For this project, the chosen program is Halcon, a vision machine application used for 3D vision techniques. This program will be explained in depth in the followings sections.

The technology of the sensor must be of type CMOS since these are the ones that offer a better result in the sector of the laser triangulation. Photonfocus is one of the leading specialists in the development of CMOS image sensors and corresponding industrial cameras for machine vision. Photonfocus manufactures with the latest generation of CMOS with Active Pixel Sensor (APS) and global shutter technologies enable high speed and high dynamic range applications, while avoiding disadvantages like image lag, blooming and smear.

The model used is the MV1-D2048x1088-3D03-760-G2-8 camera with an 8mm / f1.4 lens. The information in this section appears in the user's manual of the camera.

This camera has a Gigabit Ethernet interface called GigE Vision optimized for high speed laser triangulation applications, reaching up to 10200 profiles per second. In addition, this camera contains a 2.2 megapixel Monochrome CMOS image sensor developed by CMOSIS.



Figure 21: Photonfocus MV1-D2048x1088-3D03-760-G2-8 camera.

Others principal characteristics of this camera are:

- A non-linear interpolation technique in the laser line detection algorithm that achieve a high reliability and accuracy of 3D reconstruction.
- Laser line detection with up to 1/64 sub pixel accuracy.
- With the Dual Peak isn't necessary a second camera for various setups.
- Automatic vertical centering of ROI (Region of interest) through Moving ROI. The ROI is freely selectable in x and y direction.
- Global shutter
- Combined 2D/3D applications using the 2D/3D mode of the camera.
- Image sensor with high sensitivity.
- 2 Isolated trigger inputs, 2 differential isolated inputs, and 2 isolated outputs.
- Programmable Logic Controller for powerful operations on inputs and outputs signals.
- The power input range goes from 12V to 24V (+10 %).
- Column Fixed Pattern Noise Correction for improved image quality.
- Strobe output with programmable delay.
- An intelligent A/B shaft encoder interface.
- The trigger signal input range is from +5V to +30V DC

Features of 3D camera

The camera transmits data to the user in three different modes:

- 2Donly: the camera turns off the laser detection and acts as a normal area scan camera. This mode is used for determinate the possible ROI and serves as preview mode in the setup.
- 2D&3D: the camera turns on the laser detection, and transmits 2D image and 3D data. This mode is used for the configuration of the triangulation system.
- 3Donly: the camera turns on the laser detection, and transmits only 3D data plus an additional image row. This mode is faster than 2D&3D at the scan rate and works in the same way as the previous one.

The trigger of this camera can be configured to be active high or active low by the command "TriggerActivation". It could be selected one of the followings modes:

- Free running: in this mode, the camera internally generates the trigger. The camera starts recording immediately. It is used when continuous recording is required. For example, to perform the calibration and grab the correct images.

- Software Trigger: The trigger signal is applied through a software command.
- Line1 Trigger: The trigger signal is applied directly to the camera by the power supply connector through pin ISO_IN1.
- ABTrigger: Trigger from incremental encoder, used in our setup. In order to synchronize the camera triggers to the speed of a conveyor belt, can be used an incremental encoder with A/B outputs. These A/B outputs can be directly connected to the camera in order to generate suitable triggers inside the camera. Specifically, for our setup, it has been used "A/B Mode". ABMode could be selected by three possible configurations: Single, Double and Quad.

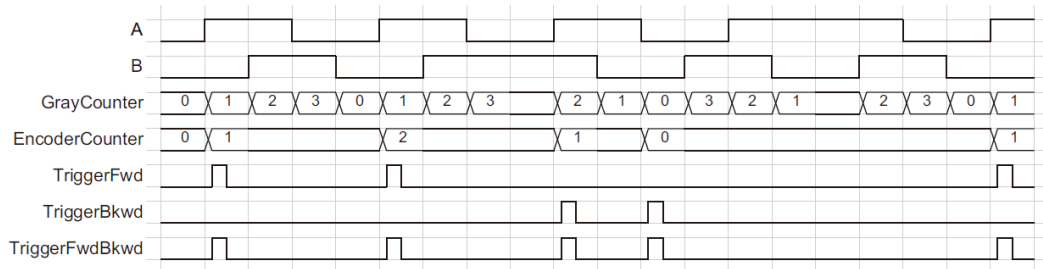


Figure 22: Single A/B Mode.

For this application has been selected Single. Single mode generated a trigger every A/B sequence.

3.3.3 Laser Line Projector

One of the specifications of the camera indicates that the Spectral range standard sensor ranges goes from <350 to 900 nm (to 10% of peak responsivity). The second restriction that must be considered on choosing the laser line projector should be the power of it. The laser must be to fully cover the object to be scanned.



Figure 23: Z-laser z12018s-f-660-lp45.

The model used was a Z-laser z12018s-f-660-lp45. Some of the main characteristics of this laser line projector are:

- The power supply voltage goes from 5 to 30 VDC \pm 5%
- The mode of operation is APC with current limiting
- The modulation is analog intensity control up to 32 steps
- The modulation is also able to use a digital TTL Trigger up to 1KHz
- Integrated microcontroller
- The output power is from 1 to 200mW depending on wavelength
- The wavelength is 660nm (red)
- The optic lens is lp45 (a line)
- Hand focusable

Because this laser has a powerful beam of light, for manage it is necessary use special protection glasses.

3.3.4 Positioning system

One of the components of measurement system is the positioning system. This is composed by different devices that together allow obtaining a continuous and controlled movement of the combination of the laser and camera. For this thesis, a vertical conveyor belt has been chosen. This movement system affords a fast adjustment of speed and distance of the total travel of this. In order to control this system through a computer system we must add some components to a current conveyor system.

The system of positioning by means of linear guide rail and controlled by computer is composed by:

- AC Motor

The engine chosen is a Vemat, type VTB 63B. It is a three-phase motor with thermal insulation that allows it to reach a temperature of 130° cause being of type B. Its nominal power is 0.18 KW which is equivalent to 0.25 HP. Also have a efficiency of 54% and a power factor of 0.70.

Working at 230/400 V and 50 Hz and reaching 1380 rpm as the nominal value. The consumption of this is situated about 1.30 A at the input of 230 V.

Associated with the motor is a worm screw reducer which is the failure to connect the motor with the toothed belt. The model is the Motovario NMRV / 030.

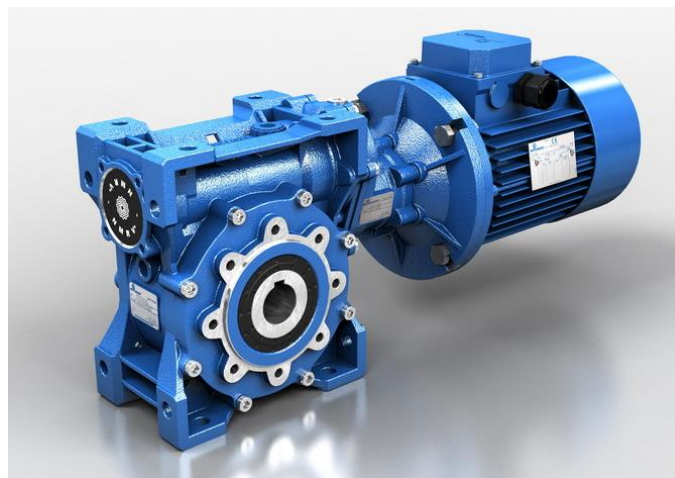


Figure 24: AC Motor Vemat, type VTB 63B.

- Linear guide rail

The conveyor belt system consists of an aluminum structure, a toothed belt, a bracket for the camera and laser and bearings blocks for smooth and continuous movement.

The structure was designed previously for previous projects, has a height of three meters to be able to look over the racks completely, in the lower part is maintained by 4 legs arranged radially from the center.

The bracket that supports the camera and laser assembly is formed by two parts. One that is mounted on the belt and attached to it carries an aluminum profile. This profile allows moving the position of the camera or the laser until finding the best position for the system of triangulation. The bearings blocks allow a circular movement of the toothed belt over the whole structure.

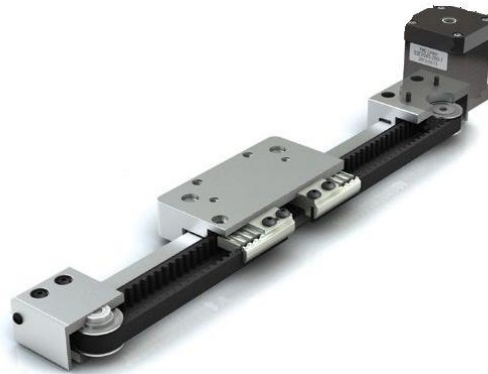


Figure 25: Aluminum linear guide rail.

- Frequency converter

One of the most important components in the positioning system is the frequency converter, as it oversees controlling the speed of the motor, as well as its movement start or stop. The converter must be selected with the requirements of the parameters of our engine. In this case, an Omron will be used, namely the model Omron j7 mini J7AZB0P2. The great versatility of this converter makes it perfect for our system. It has a digital operator provided with potentiometer for a fast speed selection. The output is OP2 type that means up to 0.25 kW, and the input is Single-phase 200VAC. The parameters of configuration will be explained in the next section



Figure 26: The frequency converter Omron j7 mini J7AZB0P.

- Limit switch

A limit switch is a switch operated by the presence of an object, for this system they will be used of magnetic type. The first one will be located in the upper part of the trajectory of the support and one second in the lower zone, preventing in both cases that collision with the end of the belt.

The chosen model is the Sick IME08-04NPSZW2S. These inductive proximity sensors detect the metallic zone of the bracket where the camera and laser are located.

These will send a signal to the plc when they locate a metallic object in its zone of detection, thus the PLC is able to know when must stop the movement, either in the high or low zone.



Figure 27: Limit switch Sick IME08-04NPSZW2S.

- Programmable logic controller

The PLC is in charge of controlling the movements of the motor, for this purpose it sends the instructions to the frequency converter. The model used has been the SIEMENS S7 300 CPU314C-2DP, within 24 digital inputs and 16 digital outputs. It works with a memory card where the program is stored, and require an external 24 V DC supply. The programming of this PLC will be explained in the next section.



Figure 28: PLC SIEMENS S7 300 CPU314C-2DP.

- Rotary Encoder

An encoder is an electro-mechanical device that converts the angular position to digital code. It is connected to the inputs of the trigger of the camera so that it can know the speed of the positioning system descent and thus calculate the number of images that it must to take in each second. For this system will use the SICK encoder with model number DFS60B-S4PM10000. It has a resolution of 10000 pulses per revolution, which is enough for the speed that our engine reaches. This device is connected to the worm screw reducer to give an exact position. Moreover, the power supply goes from 4.5-32V.



Figure 29: Encoder DFS60B-S4PM10000.

- Relay Card

A Relay is a switch that uses electromagnet to mechanically operate. It is possible with Halcon to communicate with a PLC just over an ordinary socket communication. But the PLC you have at your disposal doesn't have an Ethernet port so this was an alternative method to send signals from Halcon to the PLC. This relay card called RLY08-B is connected via USB to the computer and through a COM interface. It can be operative from the application. This relay card controlled the inputs of the PLC from the computer and also is used to turn on the laser line projector.



Figure 30: Relay RLY08-B.

3.4 Connections and Configurations

In this section are explained the configurations of some devices, explained previously, that need programming like the PLC and the converter frequency, besides, the electric connection of all the different components between them.

Motor and frequency converter

Continue the order of explanation of the different devices in the previous section. First of all, it will see the connections between the frequency converter and the motor. The motor is a three-phase motor. So, you need to vary the currents of your windings to be able to rotate, and this function is carried out by the converter. The frequency converter has several possible configurations to control the motor. Our system must be able to move the conveyor belt on one direction and the opposite and stop when then program require that.

The following information is detailed in the user's manual of the frequency converter. The configuration selected is operation in three-wire sequence. To program this, a zero in the parameter n37 must be set. By making this setting, the set value in n36 is ignored. The following operate diagram shows the wiring setup of the terminals in 3-wire.

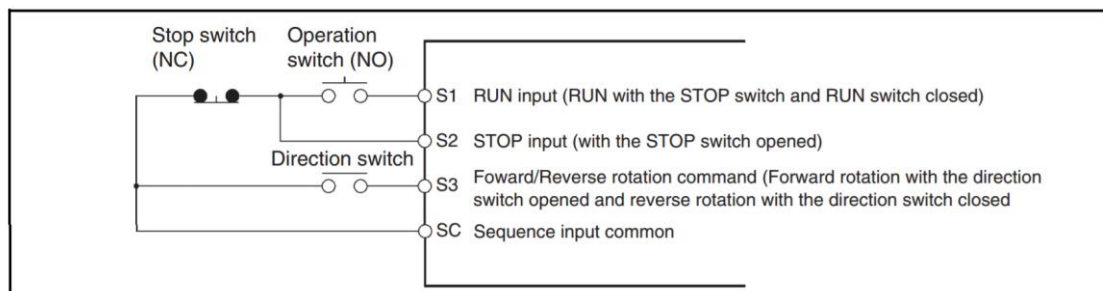


Figure 31: Three-wire sequence setup.

The electrical assembly between the motor and the frequency converter is as follows.

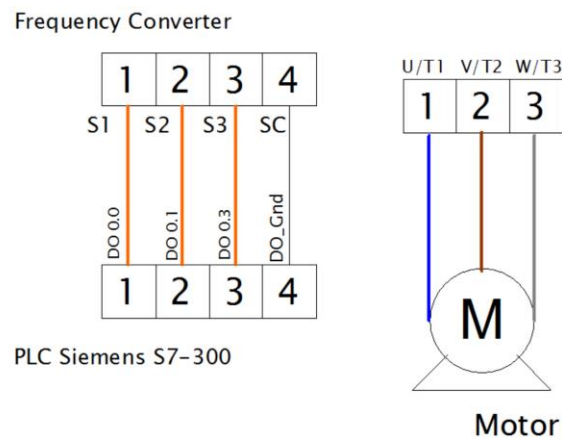


Figure 32: Electric diagram of the frequency converter.

In the above diagram, also is defined the connection between the PLC and our frequency converter. The cables have been defined with the label that is externally in the PLC, although internally it will use another numbering that will be explained later in the section dedicated to this one.

Moreover, the speed must be configured. To achieve this is necessary that the following steps must be followed.




Key sequence	Indicator	Display example	Explanation
	FREF	0.0	Press the Mode Key to turn on the FREF indicator. Monitors the frequency reference.
	FREF	0.0	Press the RUN Key. The RUN Indicator will be lit.
 MIN MAX FREQUENCY	FREF	10.0	Turn the FREQ adjuster clockwise slowly. The monitored frequency reference will be displayed. The motor will start rotating in the forward direction according to the frequency reference.

Figure 33: Configuration of the motor speed.

The laser line projector

For the installation of the laser we must connect this one with a type cable KB4. The Kb4 is a special cable provided by the manufacturer, it is female connector with "screw plug", straight, and 4 wires. The wiring for this device is shown in the diagram below.

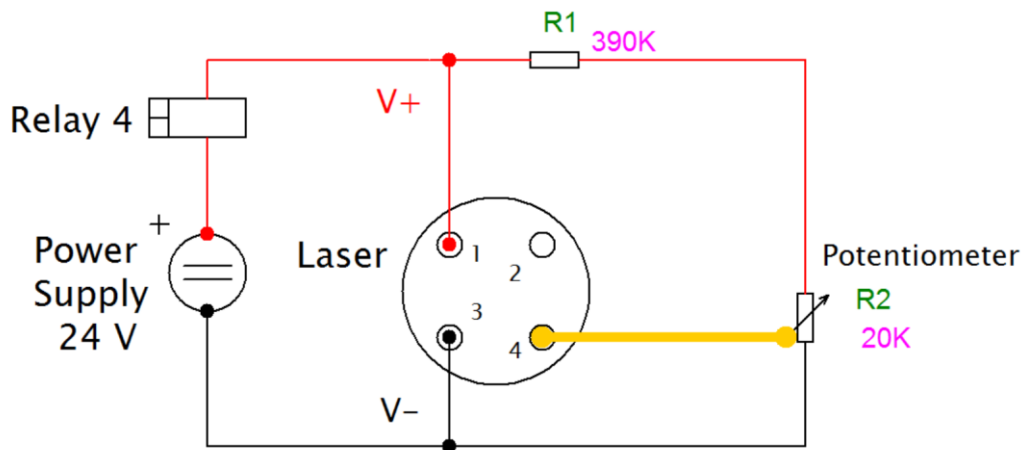


Figure 34: Electric diagram of the laser line projector.

The power supply will be connected to the transformer used for powering the whole system. For the control of the intensity of the laser, this has an analog modulation with continuous wave in Pin 4, with yellow wire. The values of the modulation must be from 0 to 1V. To achieve this value a voltage divider is performed. With nominal values of resistance and a standard potentiometer, we obtain values of resistances of 390K Ω for R1 and a potentiometer of 20K Ω . With these values we obtain a maximum output value of 1.17 V which allows adjusting the modulation of the laser throughout the range. In addition, we have a margin of 0.17V to check the zone of maximum power of the laser and to be able to adapt the power to our system. The power supply of the laser has a relay to the input to be able to control in turn on and turn off it from the Halcon program. In particular, will be relay number 4 of the relay card used for this.

The limit switch

These devices are located in the upper and lower areas of the slide travel of the conveyor belt, their function is to detect the position of the bracket and send a signal to the PLC. They have a connection of three cables, two of them are for the electrical power supply and a third one of blue will be the one connected to the PLC.

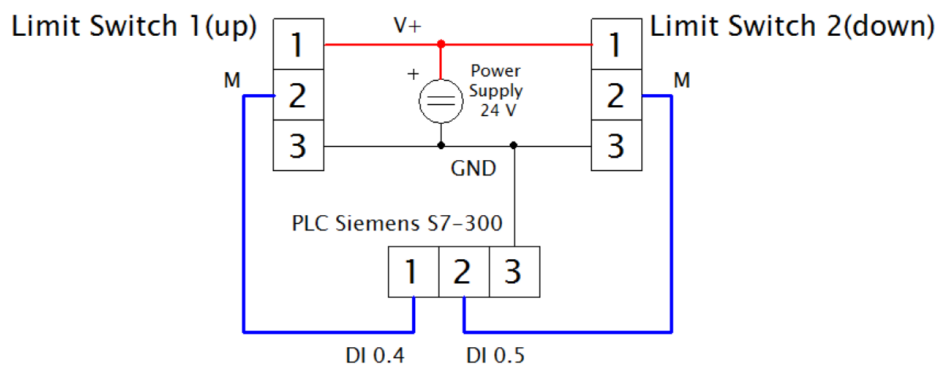


Figure 35: Electric diagram of the limits switch.

The following wiring diagram appears in the wiring diagram. In the PLC the numbered connectors will be used, DI 0.4 for the limit switch located in the upper zone and DI 0.5 for the lower zone.

Camera and encoder

The camera as well as the PLC operates at 24 V supply power, but for the encoder a 9V supply power has been chosen to be independent and also not to exceed the power limit of the 24V transformer.

For correct operation of the trigger in the camera, it must be provided with reference values, both at high and low level. Because the encoder operates at a lower voltage, it must be wired in the camera with the encoder supply as a differential input for the A/B trigger. In addition, for the camera is powered by pins 1 and 2 to 24 V DC. The following diagram shows the different connections between the two devices and their different label, also the number of the corresponding pins and the colors of the wiring used.

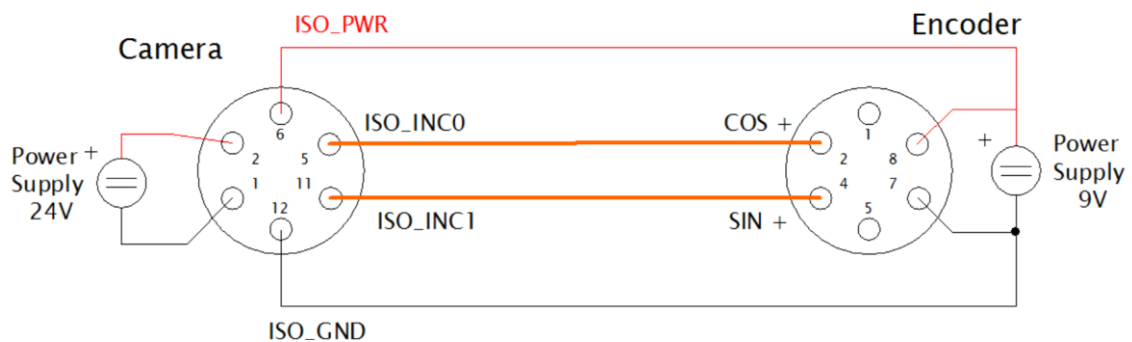


Figure 36: Electric diagram between camera and encoder.

The outputs in the encoder chosen are: pin 2 corresponding to Cos + and with white cable and pin 4 with output Sin + and pink color. Respectively in the camera correspond to pins 5 and 11 that are defined as ISO_INCO and ISO_INC1. The pins 6 and 12 of the camera are used for the supply of the trigger with the reference values of this one.

Relay card

The relay card will be in charge of connecting the Halcon application with the PLC and the laser. Your connection to the computer is via USB. To switch the laser on and off, like is already explained in a section before, relay number 4 will be used. Relays 1, 2 and 3 will instead be used for communication with the PLC. Specifically, the number 1 will serve to increase the measurement system, the second to lower it and finally the third will serve to stop this in an area that is neither the beginning nor the end of the route. The wire connection follows the scheme belong.

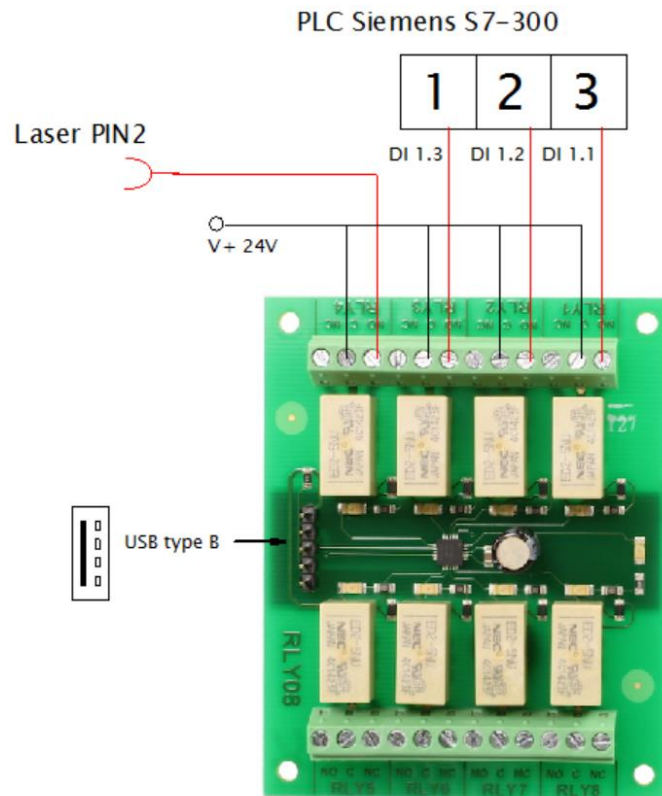


Figure 37: Electric diagram of the relay card.

The method used is to feed the common connector of the relays with positive voltage to 24V and connect the pins of normally open the output of this. This way when activating a relay, cause of this the current flows and activates the input of the plc or in the laser. The inputs chosen for this function have been: DI 1.1 to raise the support, DI 1.2 to lower it and DI 1.3 to stop.

The Siemens S7-300 PLC

This PLC has a removable card, where the programming code is stored. This device has a MPI connection for programming the PLC using the application Simatic step 7. Simatic Step 7 is a program developed by Siemens to program its devices with a high-level language. KOP has been the programming language chosen, it is also known as ladder diagram. And Outputs from the relay card are connected to inputs of the PLC.

In a single function are collected the three segments used, one to program the rise, down and stop. Segment 1 is about the stop function, this is activated when any of the three possible inputs is fulfilled:

- E125.3 is the input from the relay card specific the third one and physically is the input DI 1.3.
- E124.4 is the input from the top limit switch and physically is the input DI 0.4
- E124.5 is the input from the bottom limit switch and physically is the input DI 0.5

☐ Segm. 1 : Stop

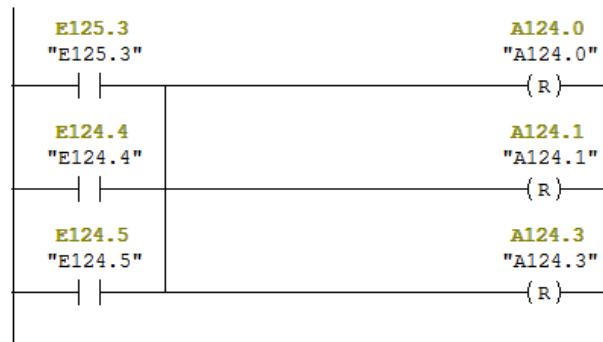


Figure 38: Contactors Diagram for stop process.

Any of these three reset inputs, setting low or zero, the three outputs.

- A124.0 is connected to the input S1 of the frequency converter and physically is the output DO 0.0.
- A124.1 is connected to the input S2 of the frequency converter and physically is the output DO 0.1.
- A124.3 is connected to the input S3 of the frequency converter and physically is the output DO 0.3.

Segment 2 is about move up. This is activated when E125.1 has high level. E125.1 is the input from the relay card specific the first one and physically is the input DI 1.1. In this case only activates the outputs: A124.0 and A124.1, and put in low level A124.3.

☐ Segm. 2 : Rise

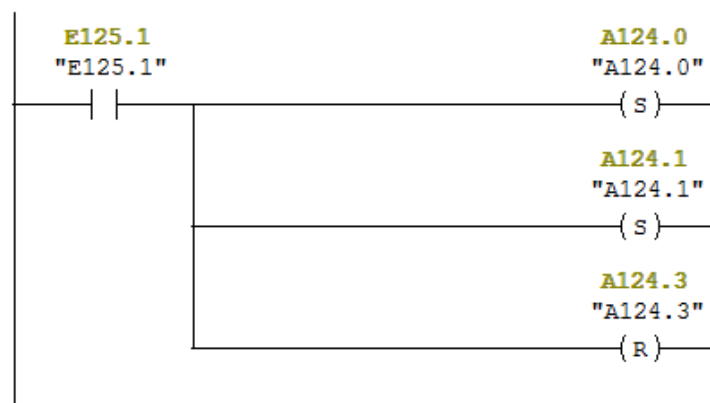


Figure 39: Contactors Diagram for forward process.

Segment 3 is about move down. This is activated when E125.2 has high level. E125.2 is the input from the relay card specific the second one and physically is the input DI 1.2. In this case only activates the three outputs: A124.0, A124.1 and A124.3.

☐ Segm. 3 : Go Down

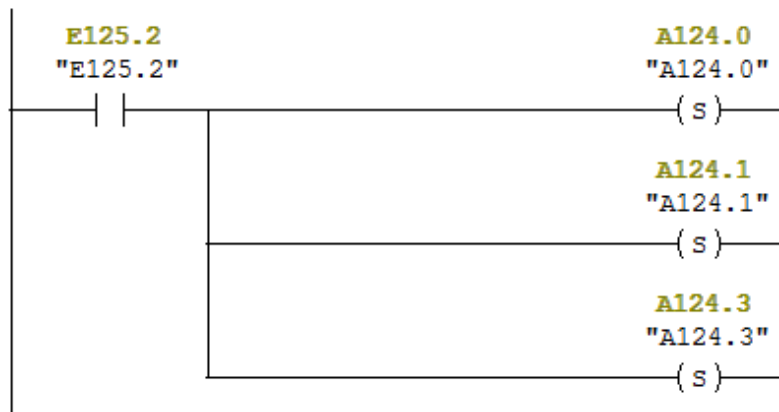


Figure 40: Contactors Diagram for reverse process.

3.5 Conclusion

The assembly of the measuring system is composed of different devices that must be connected between them properly. First, the most relevant components were selected for the project, such as the camera and the laser line projector.

Then the positioning system, which comprises more devices: motor, encoder, conveyor belt, reducer, and PLC. In addition, have been added components such as the card to be able to support the plc from the application Halcon itself.

4. Programming with Halcon

4.1 Introduction

In this section of the thesis, all the programming referring to the one realized in Halcon is going to be deeply explained, as well as some basic concepts about how Halcon works and the possibilities that it offers in artificial vision programming.

4.2 Halcon

MVTec HALCON is the comprehensive standard software for machine vision with an integrated development environment (HDevelop) that is used worldwide. It enables cost savings and improved time to market. HALCON's flexible architecture facilitates rapid development of any kind of machine vision application.

MVTec HALCON provides outstanding performance and a comprehensive support of multi-core platforms and special instruction sets like AVX2 or NEON, as well as GPU acceleration. It serves all industries, with a library used in hundreds of thousands of installations in all areas of imaging like blob analysis, morphology, matching, measuring, identification, and 3D vision.

The software secures your investment by supporting a wide range of operating systems and providing interfaces to hundreds of industrial cameras and frame grabbers, in particular by supporting standards like GenICam, GigE Vision, and USB3 Vision.

The interface has an easy-understanding design, with many tools at hand in just a few clicks.

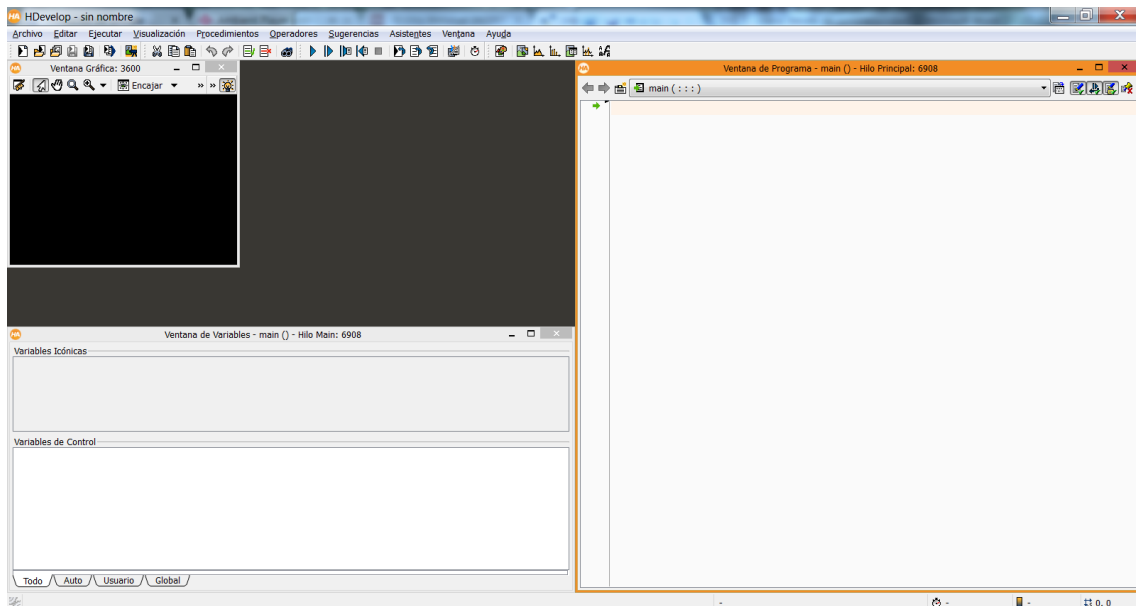


Figure 41: Halcon starting interface.

4.3 Calibration

4.3.1 Introduction

This part of the thesis describes deeply all the calibration process. The steps that have to be followed and the final result are detailed. This chapter is divided in the following sections:

- Camera calibration with its rules.
- Calibration of the light plane.
- Calibration of the movement pose.
- Final step: Store the calibration files.

Like it has been told before, the SOL that we developed is formed by an independent camera and a line laser. It is essential to calibrate setups, camera and laser setup, to get the measurement of the profile pictures. In the following sections both calibration programs have been explained.

4.3.2 Camera calibration with its rules.

The results of the calibration process are stored in a file with all the internal parameters of the camera that are need for almost all procedures. Some of these values are the size of the cells, focal length, and other parameters that model the distortions.

However the internal calibration could be performed using the Calibration Assistant that is available in HALCON, a script has been written to calculate and store all the results. A 300 mm calibration plate has been used for this purpose. During the process the camera has to be in a fixed position while the calibration plate is moved manually around the field of view.

The information related to the calibration process has been obtained from Halcon's manual [3].

It is necessary to know some details in order to get a good calibration images:

- ✓ The orientation of the plate must vary around 45° in some of the images.
- ✓ The complete field of view must be covered by the sequence of images.
- ✓ The plate must not be overexposed. This means that no pixels should have a grey value of 255.
- ✓ The calibration plate must be clean.
- ✓ The light parts of the plate must have a grey value above 100.
- ✓ The contrast between the dark and light parts must be higher than 100.
- ✓ Recommendations for the size of the plate: If it was rectangular plate is being used, this should cover 25% of the image.
- ✓ The orientation must vary in the images: the measurement is performed in other Z planes.
- ✓ The more images are used, the better is the final accuracy. In the manual it is said that at least 15 calibration images should be used, in our case 25 images have been taken.
- ✓ The images should be sharply focused, with transitions clearly delimited.

The camera calibration follows the following flow diagram:

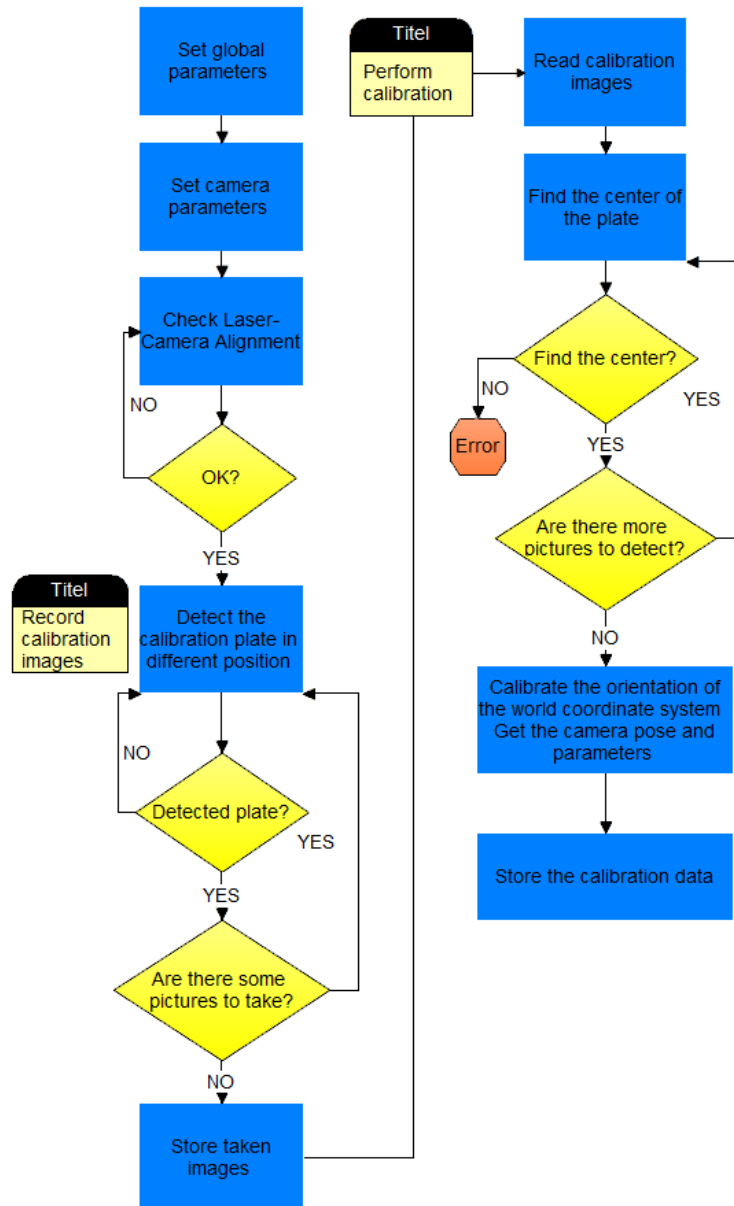


Figure 42: Camera calibration code flow diagram.

This part is going to be explained before the other calibrations because it is the essential one. It is necessary to have a good physical setup to get a good calibration result

First step consist in fixing some general parameters like total images taken, trigger mode and exposure time of the camera or routes to folder where all files are going to be saved.

After that the program set camera settings, as it can be seen in the image 43.

```

*-----*
* Set Camera Settings :
open_framegrabber ('GigEVision', 0, 0, 0, 0, 0, 0, 'progressive', -1, 'default', -1, 'false', 'default', '00111cf5a5b5_PhotonfocusAG_IVID2048x10883003760', 0, -1, AcqHanc
set_framegrabber_param (AcqHandle, 'AcquisitionMode', 'Continuous')
set_framegrabber_param (AcqHandle, 'TriggerMode', 'Off')
set_framegrabber_param (AcqHandle, 'ExposureTime', ExposureTime * 1000)
*-----*
  
```

Figure 43: Set camera parameters code.

Function “open_famegrabber” is used to start the connection between camera and Halcon. “Set_framegrabber”, instead, set camera parameters.

Then it performs a test of the camera and laser alignment, like it can be seen in figure 44. This step is necessary to get a good reconstructed final image.

```

* Create a HALCON calibration data model
create_calib_data ('calibration_object', 1, 1, CalibDataID)
set_calib_data_cam_param (CalibDataID, 0, 'area_scan_polynomial', StartCamPar)
set_calib_data_calib_object (CalibDataID, 0, CaltabDescr)

if (collect_images=1)

    if(usb=1)
        RLY_OPEN ('COM8', SerialHandle)

        LASER_ON(SerialHandle)
    endif

    * ----- *
    * camera and laser alignment
    while (1)
        dev_set_window (WindowHandle)
        set_framegrabber_param (AcqHandle, 'ExposureTime', ExposureTime * 1000)
        grab_image_async (Image, AcqHandle, -1)
        dev_display(Image)
        disp_message (WindowHandle, 'Aligning the laser line', 'window', 12, 12, 'black', 'true')
        disp_message (WindowHandle, 'aim the laser line +/- parallel to the center line', 'window', 35, 12, 'red', 'true')
        disp_message (WindowHandle, 'click a mouse button to continue ...', 'window', 60, 12, 'red', 'true')

        *draw rectangle
        dev_set_color ('magenta')
        gen_rectangle1(ProfileRegion,170,30,450,2000)
        disp_region(ProfileRegion,WindowHandle)

        *draw some parallel lines

        dev_set_color ('green')
        disp_line (WindowHandle, Height/2, 0, Height/2, Width)
        disp_line (WindowHandle, 0,Width/2, Height, Width/2)

        * When pressed the button, continue ...
        try
            get_mposition (WindowHandle, Row, Column, Button)
            if (Button = 1)
                disp_message (WindowHandle, 'Put the laser OFF ...', 'window', 85, 12, 'red', 'true')
                break
            endif
        catch (Exception1)
        endtry
    endwhile

    stop()
    LASER_OFF(SerialHandle)
* ----- *

```

Figure 44: Test camera performing code.

After camera parameters are set, the connection between PLC and Pc is necessary to initialise. Function “*RLY_OPEN*” is the responsible for creating this connection ussing “*open_serial*” like principal command to achieve it, as it is shown in figure 45.

```

open_serial(COM_Port,SerialHandle)
set_serial_param (SerialHandle, 9600, 8, 'none', 'none', 1, 1000, 'unchanged')

*Get software version
write_serial (SerialHandle, 0x5A)
read_serial (SerialHandle, 2, Data)
Module_ID := 8
Software_version := 2
*Check version
if (Data[0]=Module_ID and Data[1]=Software_version)
    * Correct USB module detected
else
    * andere module / softwareversie gedetecteerd
    stop()
endif
return ()

```

Figure 45: RLY_OPEN code.

Thanks to this link, laser could be switched on and switched off. A few lines are displayed in the screen to help with the camera-laser alignment. The command “*grab_image_async*” get an image from the camera. In the figure 46 it can be appreciated the displayed lines and the laser line perfectly calibrated.

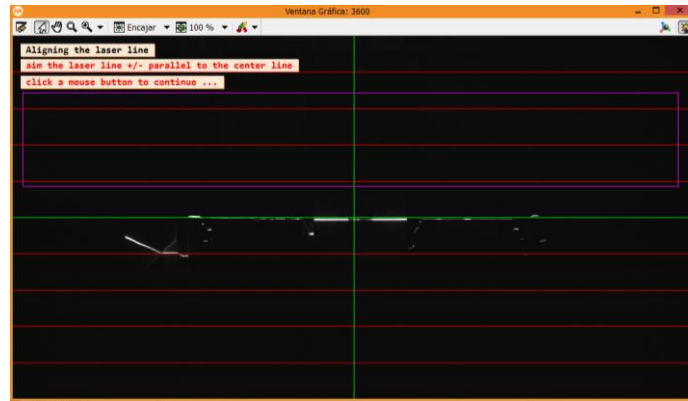


Figure 46: Image taken while program is checking the laser-camera alignment.

Once that this step is checked, the program is going to advise the calibration images. The number of this amount of images have to be fixed previously. On this part of the code, the program looks for the calibration plate, which has to be located by the calibrator person, around all camera view. Always holding it, fixed in the same point of the trail. In order to chose the location of the camera is recommended in the middle of the trayectory.

First of all the program take the whole covered view photos, in our case 25 pictures, as it can be seen in the figure 47, the calibration plate is moving around the view volume and with diferents pose to get more acuracy in the final calibration.



Figure 47: Images taken while program is covering the whole view.

The important function to achieve this is “find_calib_object” which is be able to find the black contour of the calibration plate, as well as de dots of it. It Is also remarkable the function “Caltab_exposure_regulator” which adapts the exposure of the camera until a fixed values interval(190/210).

Then it calibrates Z axis with few pictures with and without laser, far and close the camera.

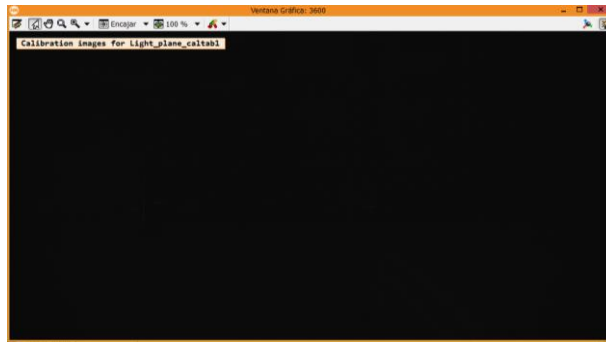


Figure 48: Image taken while program is waiting for Light plane caltab calibration plate detection.



Figure 49: Final image for Light plane caltab 1 y caltab 2 after the detection.

When all necessary images have been taken, the calibration performing start creating a new calibration object and set some general parameters. Then it reads all calibration images trying to find the calibration plate again, in order to get the initial pose and the most important thing, define the World Coordinate System(WCS) as it is shown in the figure 50.

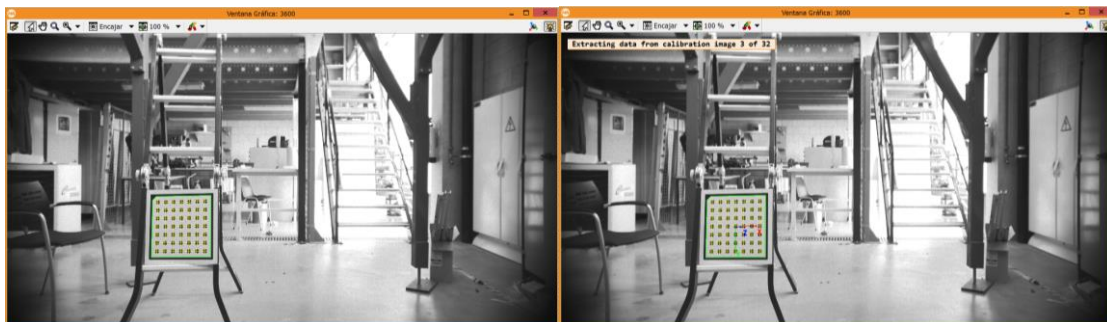


Figure 50: Obtaining camera parameters and pose.

4.3.3 Calibration of the Light plane.

The Light plane calibration follows the following flow diagram:

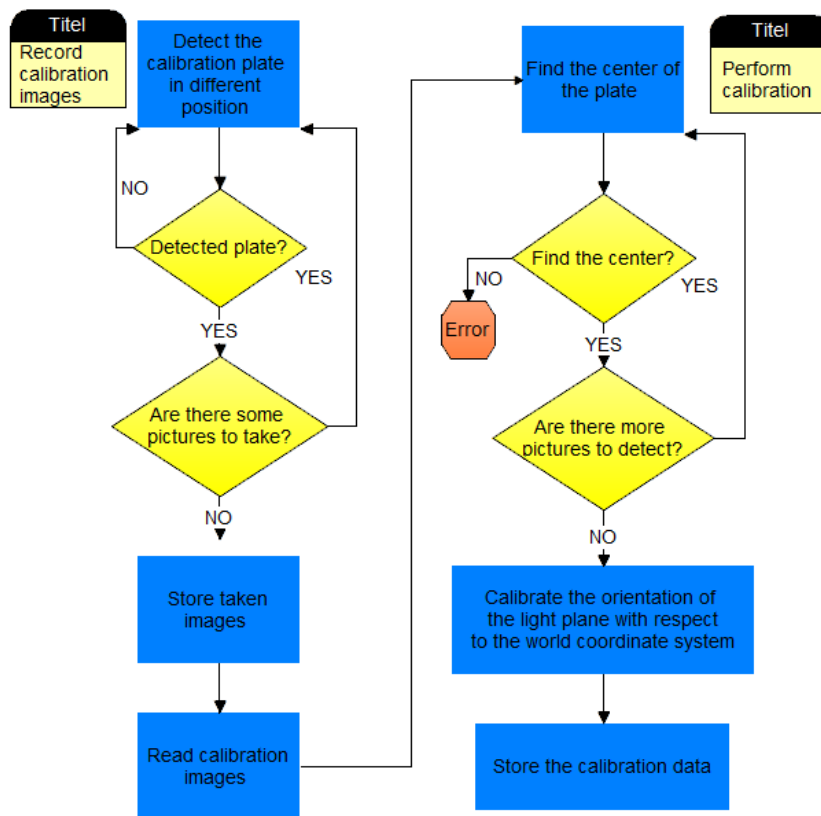


Figure 51: Light plane calibration code flow diagram.

Once it has the WCS, next step is calibrating the light of plane respect to the WCS in order to get the light plane pose. To achieve that, few images, with and without laser exposition, have to be taken, like it is possible to see in the figures 52 and 53.

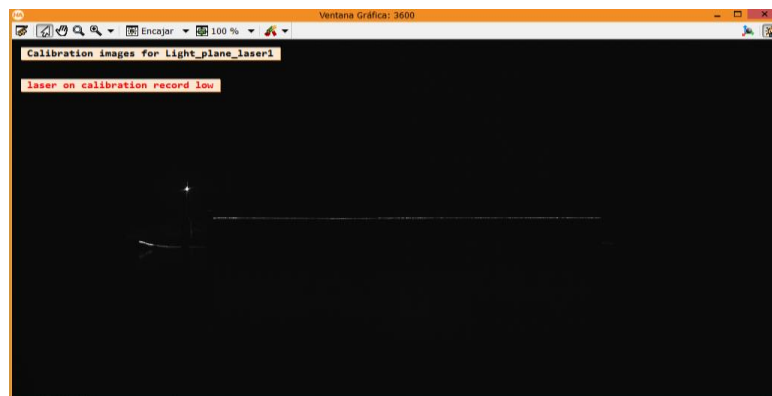


Figure 52: Image taken while program is waiting for Light plane laser calibration plate detection

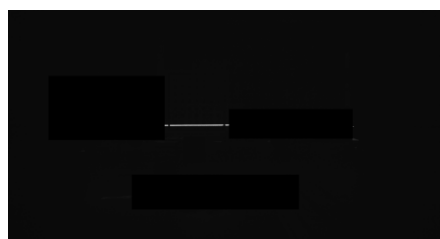


Figure 53: Final image for Light plane laser 1 after the detection.

It is important to realize that the caltab picture is taken with low exposure time and with laser turned off. However the Light plane laser picture is taken with high exposure time and the laser turned on.

Program take the word coordinate system by chosen image, getting the calibration data by means of *“get_calib_data”* function. After that it defines temporary coordinate systems, here the laser line image in ‘Light plane laser1’ and ‘Light plane laser2’ are taken in the plane defined by $z=0$ in this coordinate system.

Later it compute the 3D coordinates of the light line points in the plane $z=0$ of the temporary coordinate systems setting the input control *“LocalCameraPose”* to the TCS1 and TSC2.

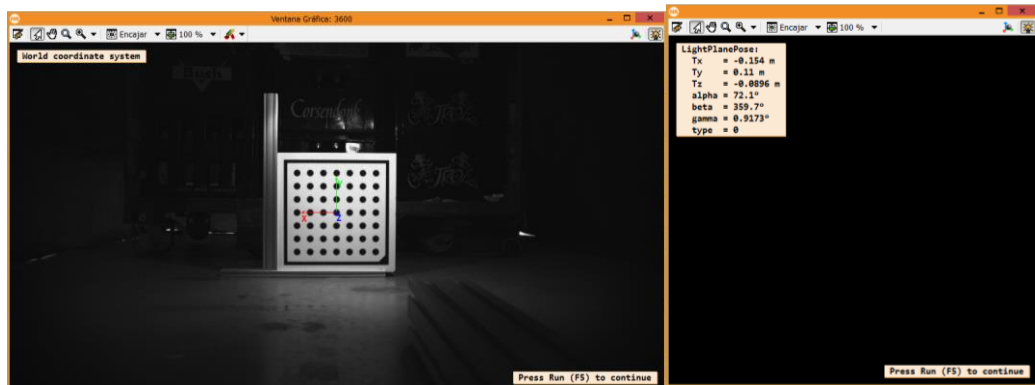


Figure 54: Obtaining Light plane 1 pose.

The key operator here is *“compute_3d_coordinates_of_light_line”*, in which the extraction of the coordinates of light line points in the image coordinate system is performed by using the operator *measure_profile_sheet_of_light*. The 3D-coordinates are computed by projecting the extracted coordinates to the plane $z=0$ of the local coordinate system. Then those coordinates are transformed to the reference pose. However in necessary to know that if no local coordinate system is explicitly provided, the reference coordinate system is used instead. It is also important that the profile has to be oriented roughly horizontal to don't get a empty tuples for X, Y and Z.

Next step is fit the light plane in the 3D coordinates of the line points computed previously. This requires nearly coplanar points so it must provide line points recorded at -at least- two diffents heights, in order to get an unambiguous Solution. To obtain stable and accurate results, it acquire the light line points at the bottom and at the top of the measurement volume. Done it, last step is get the light plane pose of this results.

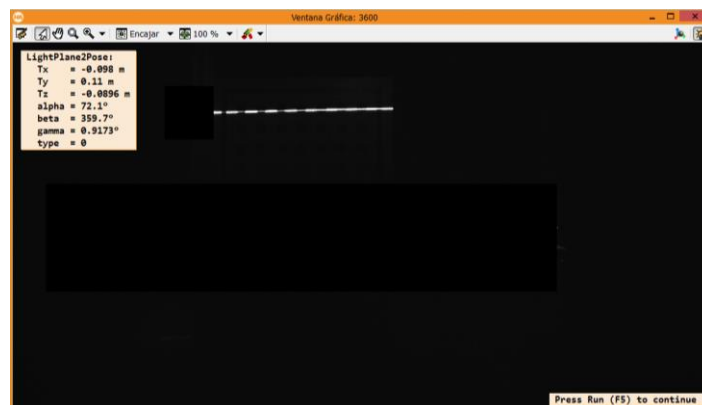


Figure 55: Getting the light plane pose.

4.3.3 Calibration of the movement pose.

In order to determine the movement of the object between two successive profile images, program acquire two images of a calibration table which describe the same movement. As a suggestion to get a good accuracy, it is better move the calibration plate by more than one step.

The movement pose calibration follows the following flow diagram:

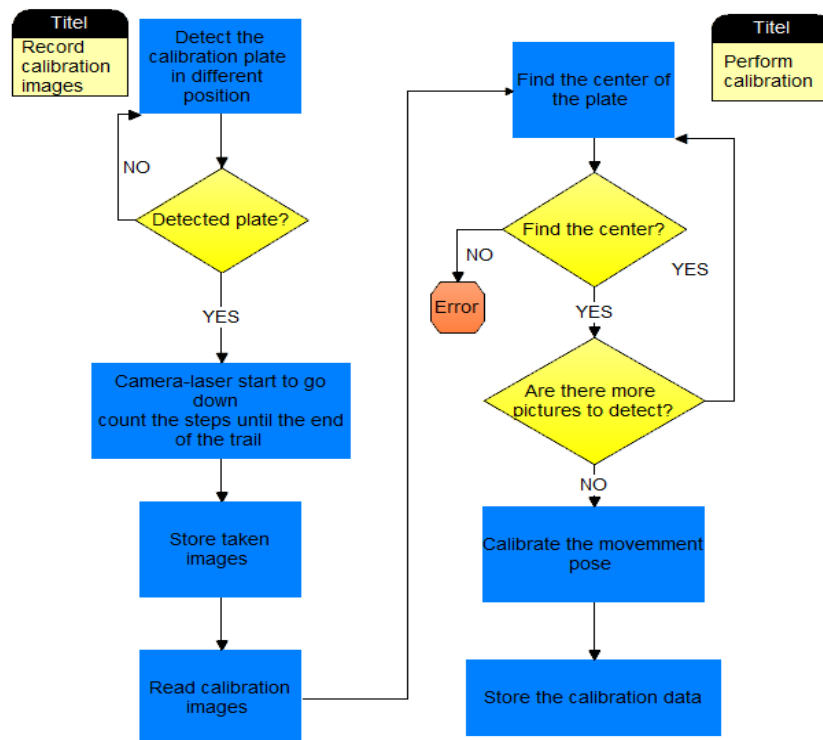


Figure 56: Movement pose calibration code flow diagram.

In this part of the process, to simulate this object movement, is necessary that the camera goes up and down in order to get vertical axis parameters. To get this purpose, the code needs a image going up until caltab is still just in the picture.

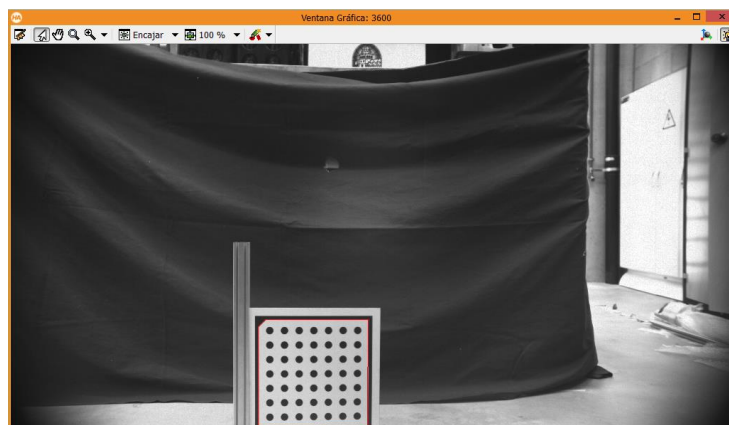


Figure 57: Final image for DY-start after the detection.

In that moment the set camera-laser must to stop, take the image (figure 57) and then start to go down until the end of the trail(figure 58). It is an important part of the calibration, because depending on how many steps the encoder counts, the accuracy the Y axis will vary.

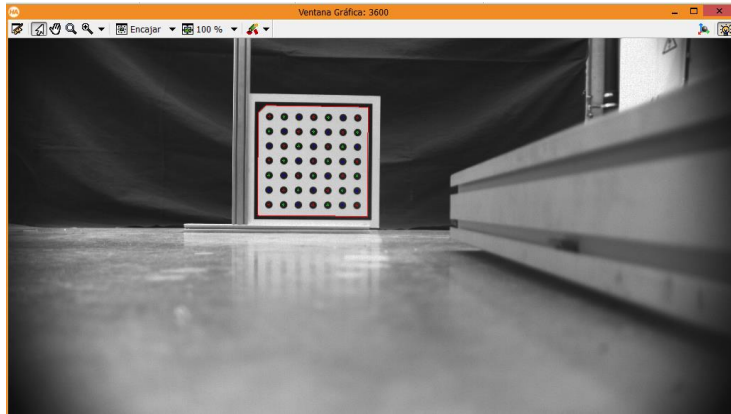


Figure 58: Final image for DY-stop after the detection.

When both pictures have been taken and stored. The program based on the number of step that the encoder count is able to calculate the distance between both images, returning an array of values with the information about the movement performed.

The responsible function to make this possible is “*movement_pose*” function which get a movement vector that define the vertical movement of the camera-laser set. Then the results are displayed on the screen.

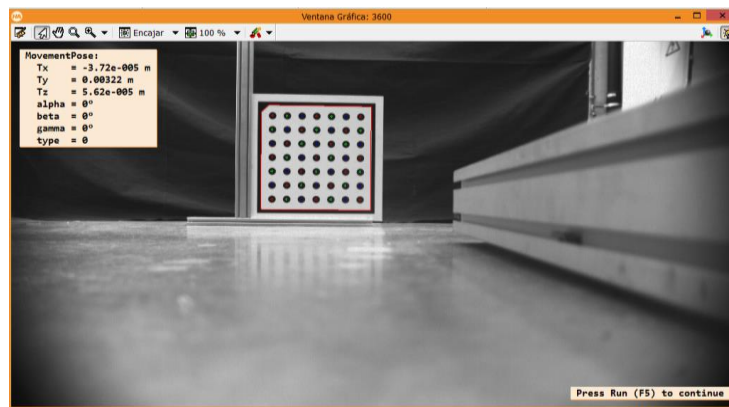


Figure 59: Results of defining the magnitude of the Y Axis.

Final part: Store the calibration data

Finally when program have all data needed, last part is the saved of all this files. Saving first the camera parameters in the file ‘CamPar.dat’, then camera pose in ‘PoseCamera.dat’. After that ‘LightPlanePose.dat’ and finally ‘PoseMovement.dat’.

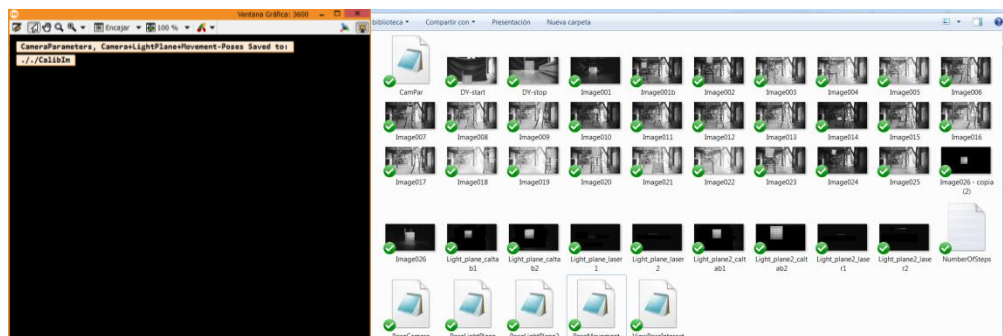


Figure 60: Final message of everything was saved correctly.

4.4 Main program

4.4.1 Introduction

This part of the thesis describes deeply the whole container detection algorithm, from the acquisition images by scanner of the racks, to the front surface of containers detection and the visualization of the results.

The steps that have to be followed and the final result are detailed. This chapter is divided in the following sections:

- Choice of the image acquisition mode and parameters definition.
- Data 3D files preparation.
- Develop of the front surface of containers detection algorithm.

The main program follows the next flow diagram:

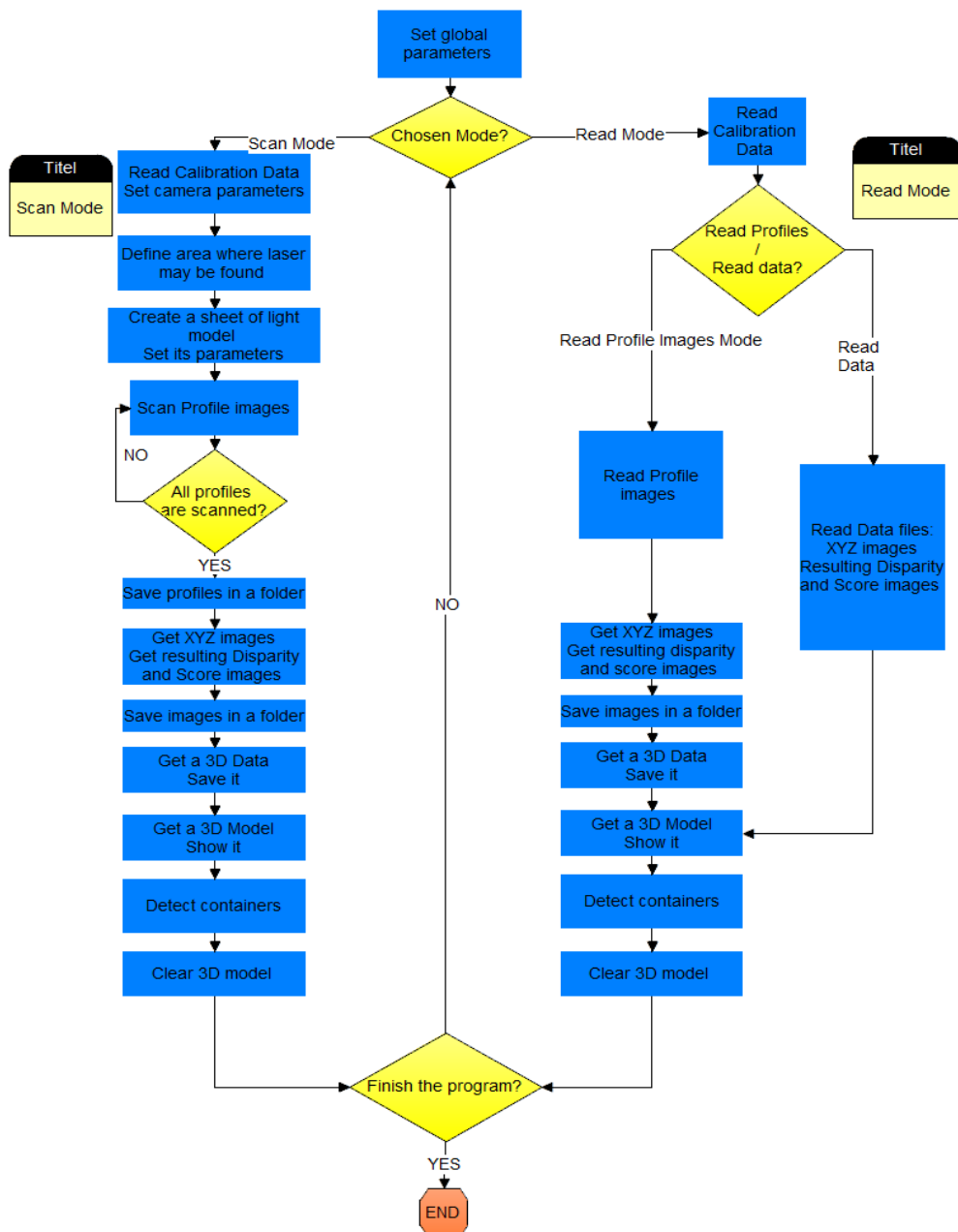


Figure 61: Main program flow Diagram

As showed in the flow diagram, there are two different methods of getting a 3D model. One of them is scan the rack and other is read previous scanners data. This last one has two performing ways, either reading images or reading a data file. From that resulting 3D model, the container detection will be performed. The main mode is scan mode, since it is the one that performs the complete process, from scanning to container detection. Being the other two modes a useful tool to check previous scanners or to finish incomplete analysis.

3.3.1 Choice of the acquisition image mode and parameters definition

First of all like every program, is recommended fix some global parameters which are going to be used during the program. Parameters like the size of the windows that will show the 3D models, paths to the folder where program can find the Calibration data and save the results files or number of profile images that we have to take or read, are defined to ensure the correct functioning. It is also representative the definition of the work program mode.

To get a good 3D model, three options are available to acquire a 3D model in this code:

- Scan profiles
- Read image profiles
- Read 3D data

The first one is essential to have camera-laser structure connected to the computer, this mode can be called 'online' mode, however the other two can be executed without have the devices nearby and these two can be called 'offline' modes. Each method has been explained in detail in the following section.

4.3.2 Image acquisition

In order to give more flexibility to the program, three modes have been designed for this purpose.

Case Scan profiles: This mode allows perform a complete scan of the rack and from that create all necessary files to get the 3D model as well as show on the screen the result of the boxes detection. If we chose this mode, it will enable procedures of "Read_Calibration_Camera", "usb", "Prepare_Camera", "Scan_profiles", "Display_Profiles"(this is optional, only to check), "Write_Profile_images", "Get_XYZ_images", "Measure_SOL" , "Write_3D_Data" and "Display_COP". On the other hand the program will disable procedures like "Read_Profiles_Images"and "Read_3D_Data".

If we chose Scan profiles mode the order of procedures are the following:

- ✓ "Read_Calibration_Camera" → this procedure permit to get all of the calibration data, including camera parameters as well as light plane pose and movement pose.
- ✓ "usb" → It establishes the connection among PC, PLC and laser opening the COM port. Besides it sends to PLC an order to rise up the camera-laser set.
- ✓ "Prepare_Camera" → In this process, the camera connection are initialized with the function "open_framegrabber". Moreover parameters like Exposure Time, Acquisition Mode (Continuous) or Multislope Mode (LowCompression) are set with the function "set_framegrabber_param".

It also set all parameters related to trigger like 'TriggerSource', 'ABTriggerMode', 'ABTriggerDirection', 'ABTriggerDivider', 'ABTriggerCountAlways'

✓ "Scan_profiles" → The first thing that this procedure do is turn on the laser and once this is ready (because it needs around 2 second to shine), it sends an order to lower the camera – laser and after that, call to function "Get_Scan_Profiles", which take images until number of profile images that we should fix previously, and store in a vector variable. If this procedure fails, it would show a error message, telling us that we would need to restart the program.

✓ "Display_Profiles" → This is an optional function, it show on screen the whole vector variable with the profiles images.

✓ "Write_Profile_images" → It saves every image in the defined folder in order to they can be read later.

Case Read_profiles: This mode allows check a complete scan of the rack that was got before and from that create all necessary files to obtain the 3D model as well as show on the screen the result of the boxes detection. If we chose this mode, it will enable procedures of "Read_Calibration_Camera", "Read_Profiles_Images", "Get_XYZ_images", "Measure_SOL", "Write_3D_Data" and "Display_COP". On the other hand the program will disable procedures like "usb", "Prepare_Camera", "Scan_profiles", "Display_Profiles", "Write_Profile_images" and "Read_3D_Data".

If we chose Scan profiles mode the order of procedures are the following:

✓ "Read_Calibration_Camera" → this procedure permit to get all of the calibration data, including camera parameters as well as light plane pose and movement pose.

✓ "Read_Profile_images" → It reads every image in the defined folder in order to bring to program the profile images.

Case Read Data: This mode allows check a complete scan of the rack that was got before but without create the files to obtain the 3D model because it reads directly the files created, skipping the step of data files preparation. If we chose this mode, it will enable procedures of "Read_Calibration_Camera", "Read_3D_Data" and "Display_COP". On the other hand the program will disable procedures like "usb", "Prepare_Camera", "Scan_profiles", "Display_Profiles", "Write_Profile_images", "Read_Profiles_Images" "Get_XYZ_images", "Measure_SOL" and "Write_3D_Data".

If we chose Scan profiles mode the order of procedures are the following:

✓ "Read_Calibration_Camera" → this procedure permit to get all of the calibration data, including camera parameters as well as light plane pose and movement pose.

✓ "Read_3D_Data" → It reads all files necessities to get a good 3D model. They are X, Y, Z images as well as Disparity and Score images.

4.3.3 Data files preparation

Once the program has all images in the memory, it is time to create the files that give us important information for the 3D model. These necessary files are the X, Y, Z images, Disparity and Score image.

First of all a rectangular area in which the profiles are measured is generated, as well as a sheet of light model in order to measure profiles according to the sheet of light technique. Simultaneously set some parameters for the model.

```
create_sheet_of_light_model (ProfileRegion, ['min_gray','num_profiles','ambiguity_solving','score_type'], [70,290,'first','width'], SheetOfLightModelID)
set_sheet_of_light_param (SheetOfLightModelID, 'calibration', 'xyz')
set_sheet_of_light_param (SheetOfLightModelID, 'scale', 'mm')
set_sheet_of_light_param (SheetOfLightModelID, 'camera_parameter', CamParam)
set_sheet_of_light_param (SheetOfLightModelID, 'camera_pose', CameraPose)
set_sheet_of_light_param (SheetOfLightModelID, 'lightplane_pose', LightPlanePose)

set_sheet_of_light_param (SheetOfLightModelID, 'movement_pose', MovementPose)
```

Figure 62: Set the sheet of light parameters

Next step is measuring the profile from successive images, reading this images, as it can be seen in the figure 62, and store in a variable. The operator “*measure_profile_sheet_of_light*” processes the ProfileImage and stores the resulting disparity values to the sheet-of-light model.

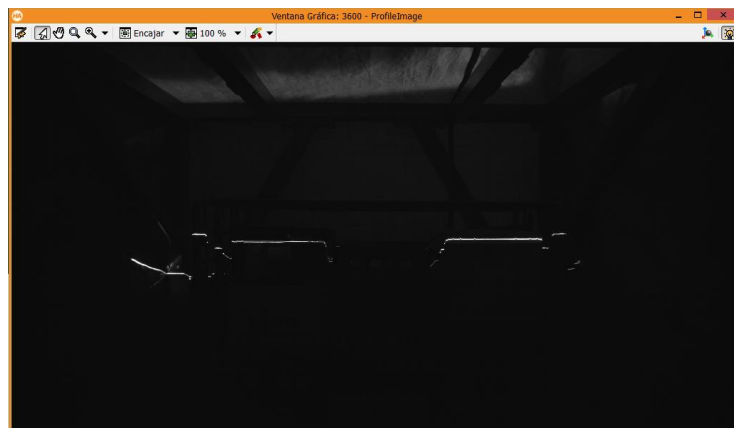


Figure 63: Profile image taken in the middle of the trayjectory.

Then the resulting disparity and score images are gotten by the shown functions in figure 64:

```
get_sheet_of_light_result (ImageDisparity, SheetOfLightModelID, 'disparity')
get_sheet_of_light_result (ImageScore, SheetOfLightModelID, 'score')
get_sheet_of_light_result (X, SheetOfLightModelID, 'x')
get_sheet_of_light_result (Y, SheetOfLightModelID, 'y')
get_sheet_of_light_result (Z, SheetOfLightModelID, 'z')
```

Figure 64: Getting the results images.

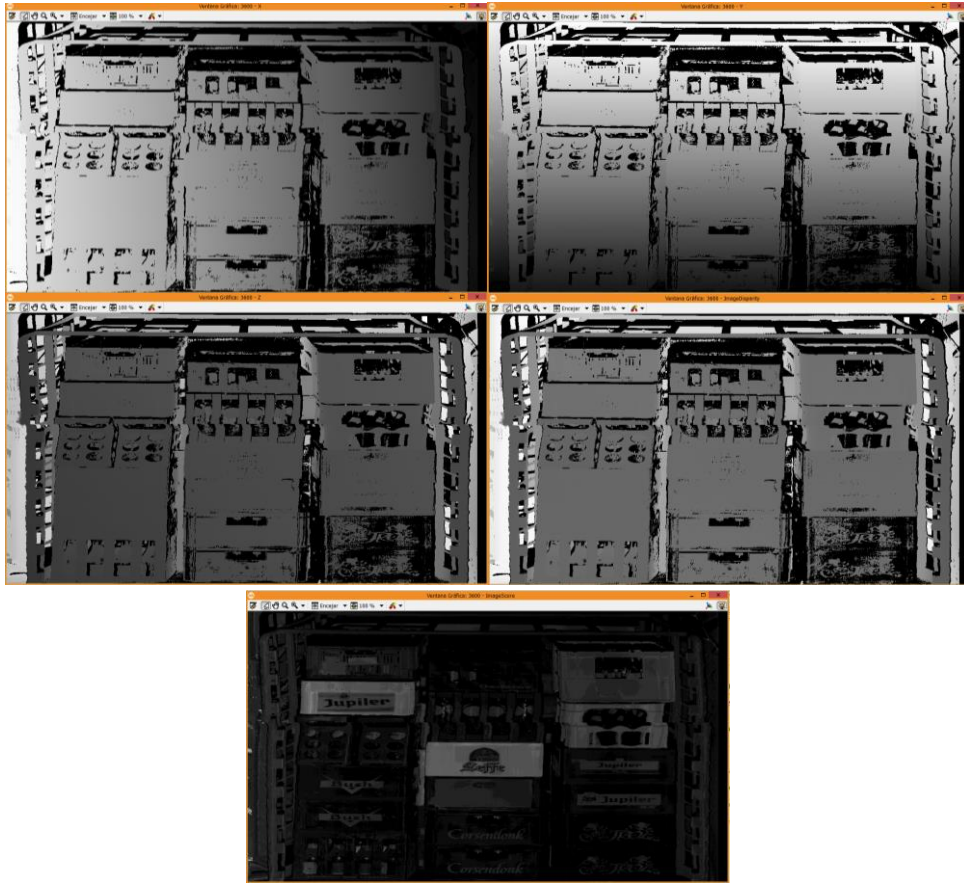


Figure 65: Images resulting from X, Y (top), Disparity (middle) and Score (bottom) filter.

These images are very useful because it is the measured reconstruction of the real rack, giving the distances of the boxes in the X, Y and Z axes.

The following step is call the procedure *“Write_3D_SOL_Result_To_Disk”* to save this images en the defined folder in where the Read_3D_Data method is going to read these images.

After that, the SOL measurement is performed with the function *“Measure_SOL”* . This procedure performs the real measurements required to generate the calibrated 3D data. Calibration data is assumed ahead set to the SheetOfLightModel3D using *“set_sheet_of_light_param”*. This procedure also measure and returns the effective execution time of this procedure. Immediately after, the operator *“get_sheet_of_light_result_object_model_3d”* get the result of a calibrated measurement performed with the sheet-of-light technique as a 3D object model.

Finally the 3D model are totally finished setting a view pose from which 3D model are shown, visualizing it on the screen thanks to the operator *“visualize_object_model_3d”*. This procederure can be used to display one or more 3d object models and to interactively modify the object poses by using the mouse. The pose can be modified by moving the mouse while pressing a mouse button. The default settings are:

- * Rotate: Left mouse button
- * Zoom: Shift + Left mouse button (or Center mouse button)
- * Pan: Ctrl + Left mouse button

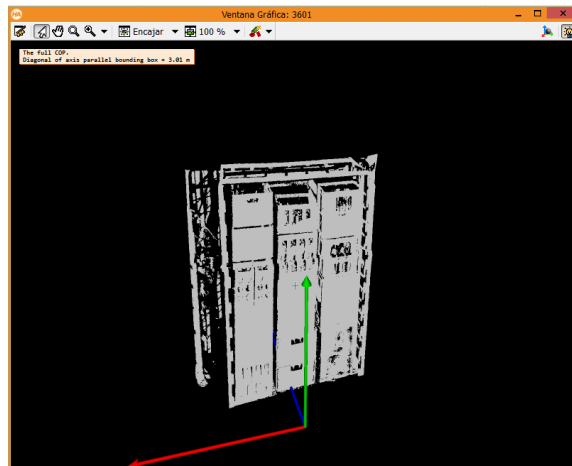


Figure 66: 3D interactive representation of the rack.

There are some display COP modes more implemented, however they aren't so much useful for our purpose.

Next step is start directly to the boxes detection.

4.3.4 Develop of the front surface of containers detection algorithm

This part of the thesis describes the containers detection algorithm. The step is a procedure which it is called from the main program. However its complexity is bigger than the main program. The steps, that must be followed, and the final results are detailed in the next lines.

This chapter is divided in the following sections:

- Procedure parameters definition.
- Create detection zone.
- Detect the rack.
- Detect the containers front surfaces.
- Detect the containers shapes.
- Find the centre of the container front surface.
- Divide the rack into zones.
- Choose a line that we want to pick.
- Search for the selected container front surface.

These steps are summarized in the following flow diagram:

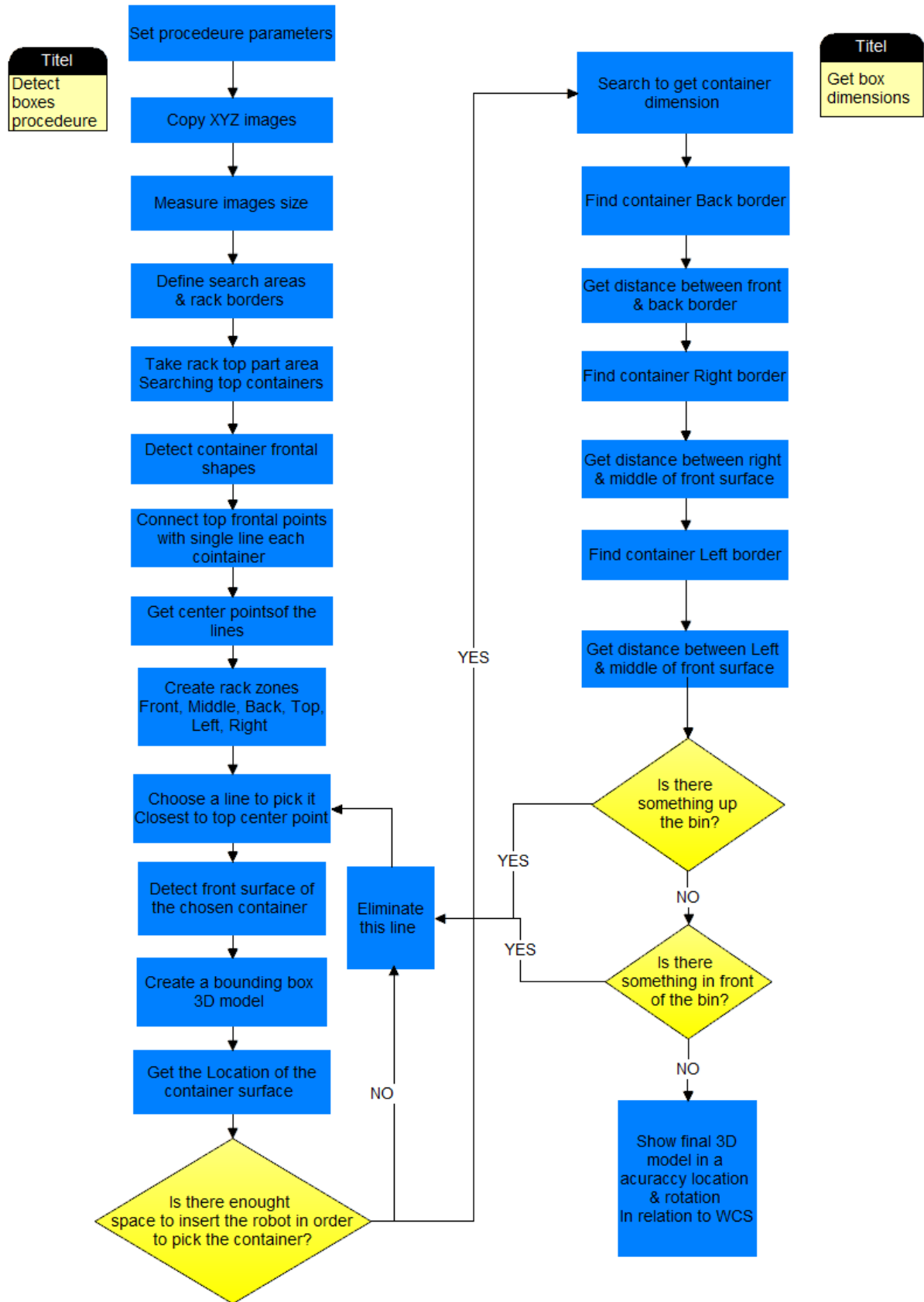


Figure 67: Detect containers code flow Diagram.

Procedure parameters definition

It is necessary define some parameters which will be used later. Because this is a procedure of main program, some parameters come given from it. Some of them are values or value arrays other is the 3Dmodel and others are the XYZ images and Image Score.

Create detection zones

The way to know the zone that we want to detect is measuring the image size (function *“get_image_size”*); from that program generate three rectangles (function *“gen_rectangle1”*), one of them in the upper part of the image and the other two in the right and left part. This area is where the program will look for the external brackets of the rack.

Detect first the external tubes of the rack is very useful to have limits of the area where we can find the containers.



Figure 68: Defining the zones where borders of the rack should be detected.

Detect the rack

The first part that we are going to detect is the upper tube. Thanks to the operators *“reduce_domain”* and *“threshold”* we are be able to find it. Once this tube is selected and like the standard precision that is fixed in the *“threshold”* is not enough accurate, it is necessary that the program dilates the selected shape with the function *“dilation_circle”*.

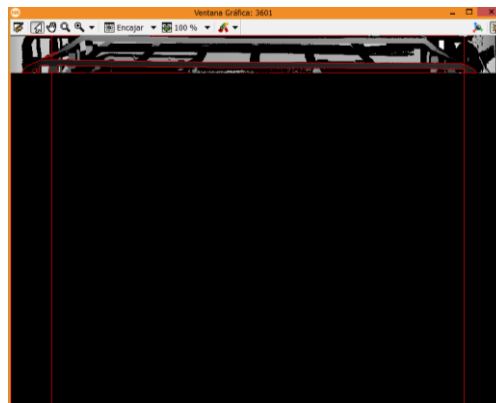


Figure 69: Detecting the top border of the rack.

After that, left and right tubes are detected by the same method.

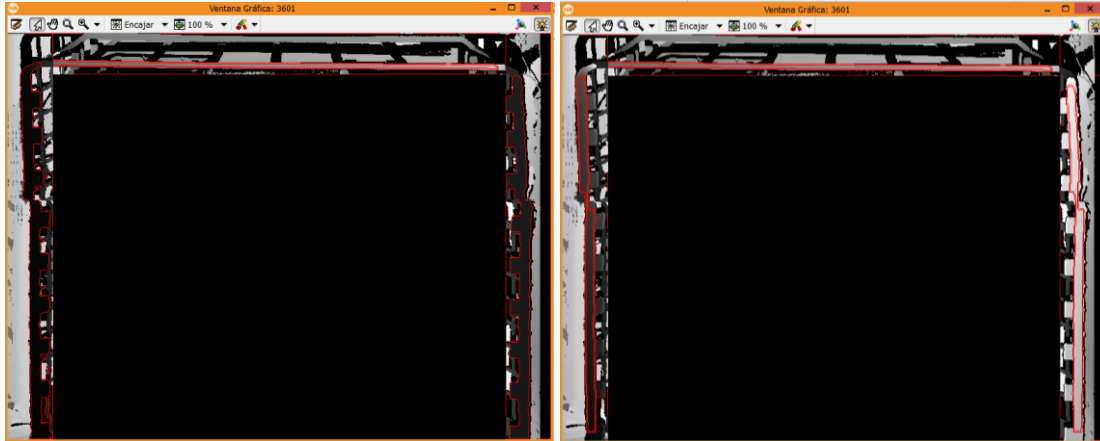


Figure 70: Detecting right and left borders of the rack.

By means of the function “*union1*” both regions, right and left are joined to work only with a single region instead of two.

Next step is transform this resulted area in a easy-work rectangles, the operator “*smallest_rectangle2*” is the best for this purpose, because it generates the smallest surrounding rectangle with any orientation.

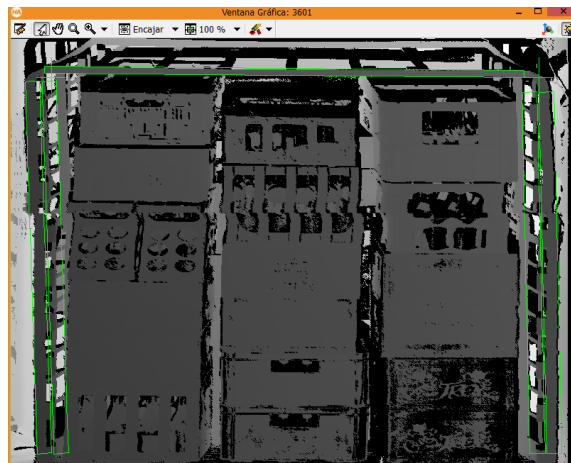


Figure 71: Generating the smallest surrounding rectangle with any orientation.

Detect the containers front surfaces

In this part of the code, the program is going to look for the front part of the containers. To achieve this, “*reduce_domain*” and “*threshold*” operators are been used with the correct parameters in order to take the important area that we want. “*Threshold*” function permit to select pixels from the input image whose gray values g fulfill the following condition:

$\text{MinGray} \leq g \leq \text{MaxGray}$. “*Reduce_domain*” function reduces the definition domain of the given image to the indicated region.

The combination of the reduction in the X-image, Y-image and Z-images and then complemented them with the functions “*xyz_to_object_model_3d*”, which transforms the calibrated X-image, Y-image and Z-image 1D points and combines them together to form a 3D object model, and “*object_model_3d_to_xyz*”, which transform 3D points from a 3D object model to images.



Figure 72: Detecting the containers front surfaces.

Detect the containers shapes

Once it has the front surfaces, the program defines an area in the top part where container top borders will try to be found.

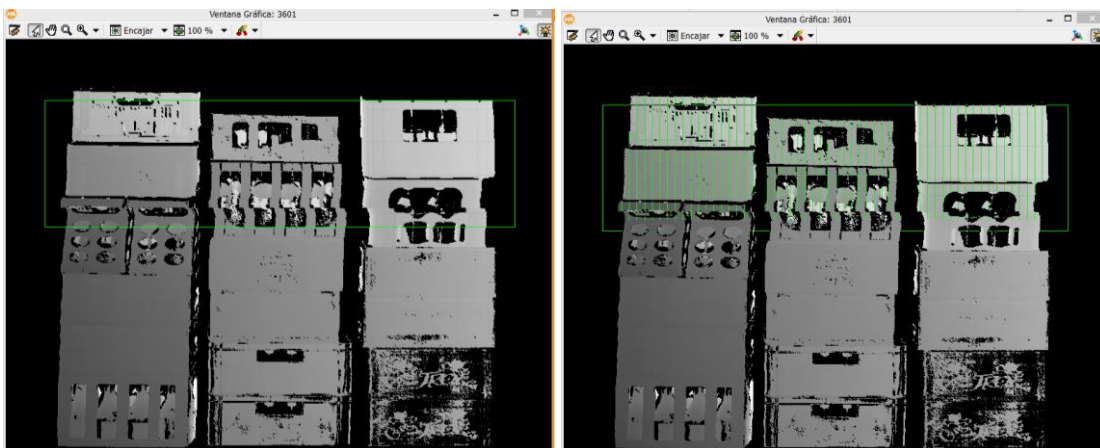


Figure 73: Detecting the containers shapes, defining an area and then with vertical lines.

This is possible because of the procedure “*detecting_transitions_lines*”, which is able to perceive the change in gray values and generate a line between dark zones, pointing thus the lightest zones.

After that, program made cheks, line by line, storing and showing on the screen the beginning and the end of each line. Red points represent start point of the line and green points represent end of the line.



Figure 74: Remarkng the top borders of the containers.

Find the centre of the container front surface

Mainly we are interested in the red ones, which define a rect line in the top border of the boxes. Program call to “xyz_to_object_model_3d” in order to transform the calibrated X-image, Y-image and Z-image 1D points and combines them together to form a 3D object model due to work in 3D is easier. Then call “connection_object_model_3d” “select_object_model_3d” functions to join these point making a line in 3D location.

Next step is create a line with the selected points, the simple way is using “smallest_rectangle2”, which generate a rectangle surrounding the points. Once this is done, it find the center of these lines, which are the centre of the container front surface as well.



Figure 75: Joining the points and find the center point of the line.

Divide the rack into zones

Next step of the container detection is dividing the rack into different zones in order to check if some container that we are going to pick out is in a difficult location to insert the robot. The areas are selected using “threshold” and “reduce_domain” with X image, in order to take the left and right tube and its surroundings, and Z image in order to take top part of the rack. To get the 3D model the function “xyz_to_object_model_3d” is the best option.

Once that program has a 3D model, it calls “smallest_bounding_box_object_model_3d” and “gen_box_object_model_3d” operators which generates a box that surround whole 3D model points that we selected.

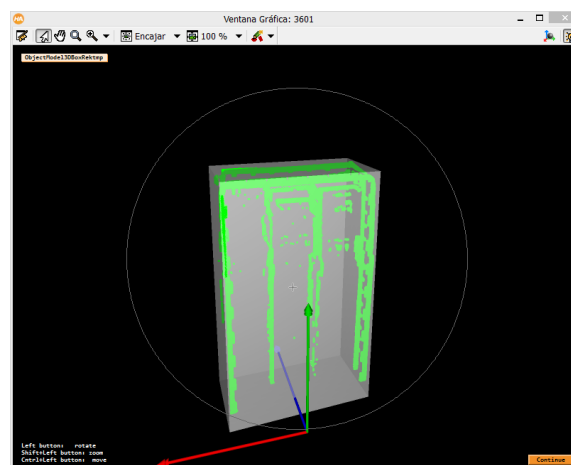


Figure 76: Generating the small bounding box surrounding the 3D model.

Starting from this base, program works with box measures to make different volumes. It use “create_3dbox2check” to choose the offset and dimensions of the boxes respect the first one.

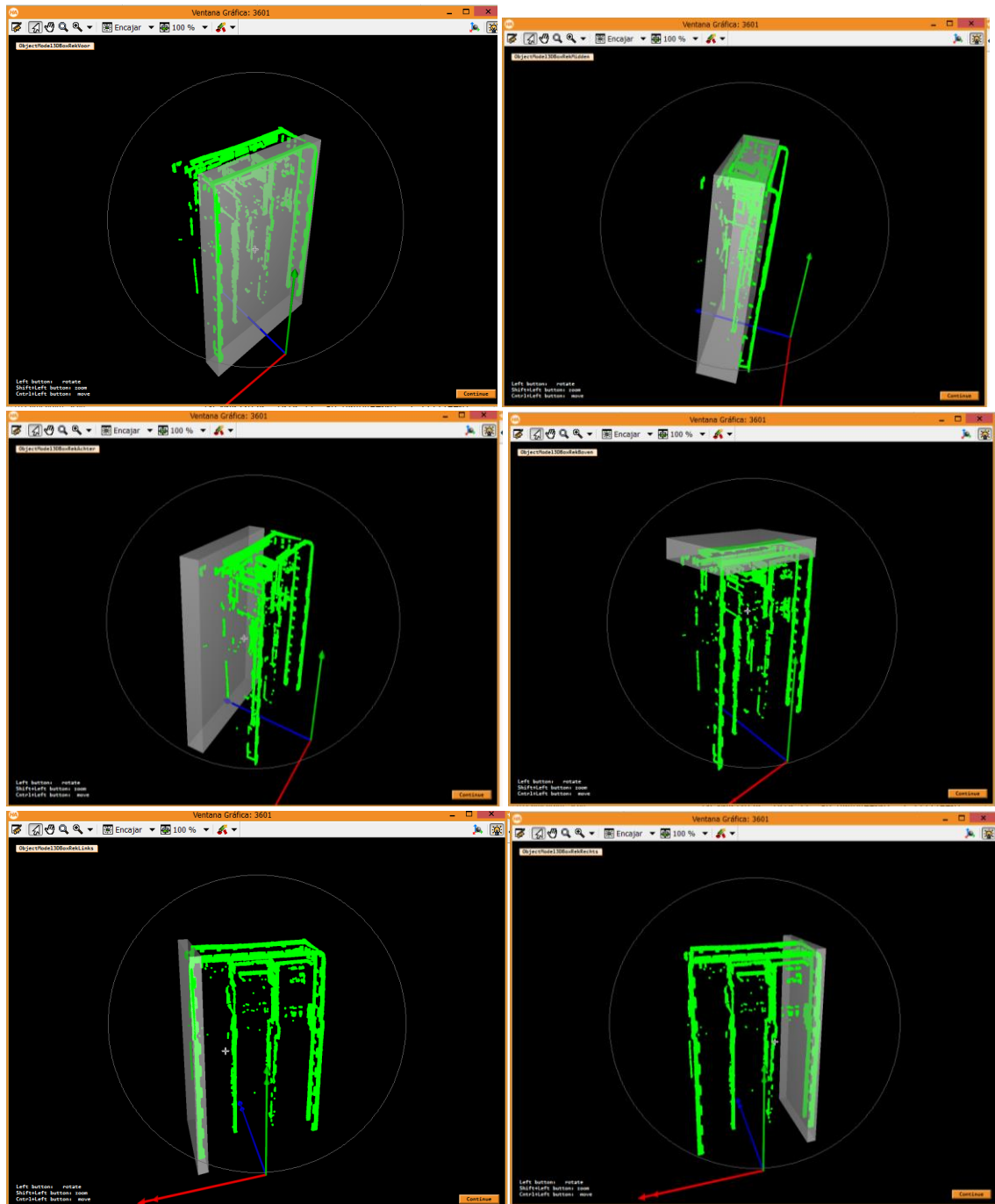


Figure 77: Different divided parts of the 3D model. Front and middle at top, back and top at middle, right and left at bottom.

Choose a line that we want to pick

In this step, the program is going to calculate the distance between a point that hangs in the middle of the container and a point in the top of the rack, to be prepared in that, this point always depends on the height of the highest tray edge. This is to get the rack still about 'cascade' empty .

For getting distances between container center points and the reference point it calls procedures *"distance_object_model_3d"* *"get_object_model_3d_params"*. Just after that *"tuple_sort_index"* , which sorts the elements of a tuple and returns the indices of the sorted tuple, with is element sorted, program is going to iterate to get the exact position of the

containers. The closer container respect to the reference point, will be chosen to get out. The chosen function to select the line is “*select_obj*”.

Search for the selected container front surface.

This part of the code is performed in its majority by “*search_rect2_front_area*”, which generates a detection rectangle beginning with the line chosen and then it takes X, Y and Z images to reduce the domain only to the chosen container. Separating the points of front surface.



Figure 78: Selecting the best located container.

After that, “*smallest_bounding_box_object_model_3d*” and “*gen_box_object_model_3d*” are the in charge of create a box that surrounding the surface of the container.

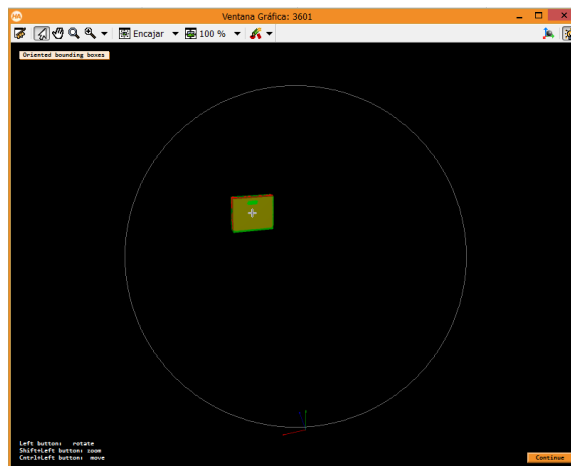


Figure 79: Creating a surrounding box of the front part of the selected container.

This representation is showing the exact location and pose that the container front surface has in the reality.

Thanks to the operator “*get_pose2pick*”, the program achieves the distances and rotation respect to the three axes from the World Coordinate System to the top of the middle of the current chosen box surface.

Clear extraction checks

In this point of the program, in which it has a precise position, it is interesting to know if the robot can enter until the location that the chosen container is. This check is easy using the

following technique: create a box in the top of the container surface and perform “*intersect_box_model_3d*” operator, which is look for shared points between top part of the rack and current surface 3D model.

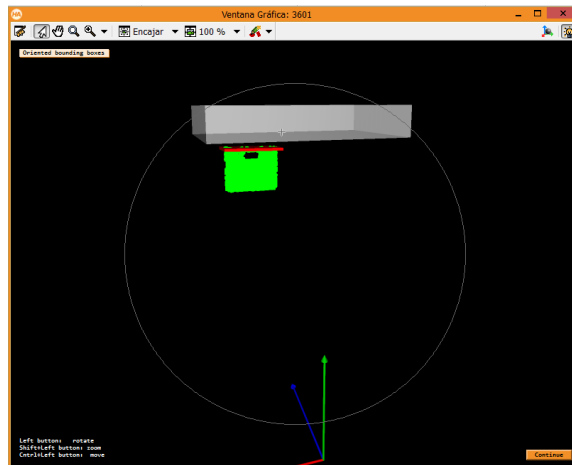


Figure 80: Checking if there is enough space to take the box.

If this number of points is higher than 1, the container is too high and it move on to the next container. If this number of points is 0, there is enough space to take the container, so next step is try to know the dimensions of the container.

Search for the container dimensions

In this part of the code, program is going to look for back part of the container first and then left and right sides. Both checks are performed creating a box with the previously discussed function “*create_3dbox2check*” and then making an intersection between both 3D models by means of “*intersect_box_model_3d*”

In the first case, a perpendicular box to the container surface is created trying to found the back part, and then a narrow box in the upper edge of the surface is created as well.

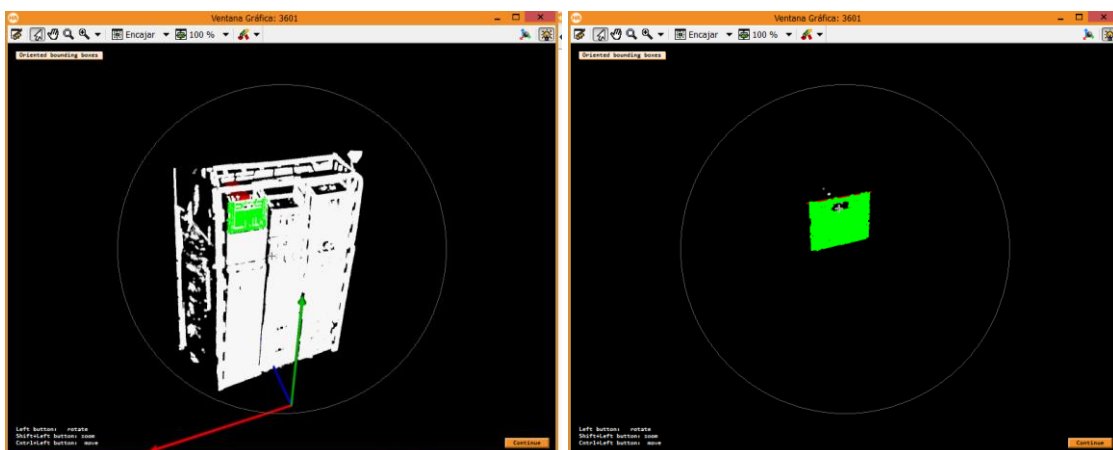


Figure 81: Looking for the back border of the container.

Through “*distance_object_model_3d*”, “*get_object_model_3d_params (&distance)*” functions, which get every distance between both 3D models, and “*tuple_max*” operator, which give us the maximum distance, program gets the depth of the chosen container.

In the second case, the process is similar, however, the box is created in the borders of the surface and the narrow box is perpendicular to the surface to be able to calculate the distances. The used function is the same than in the back part detection.

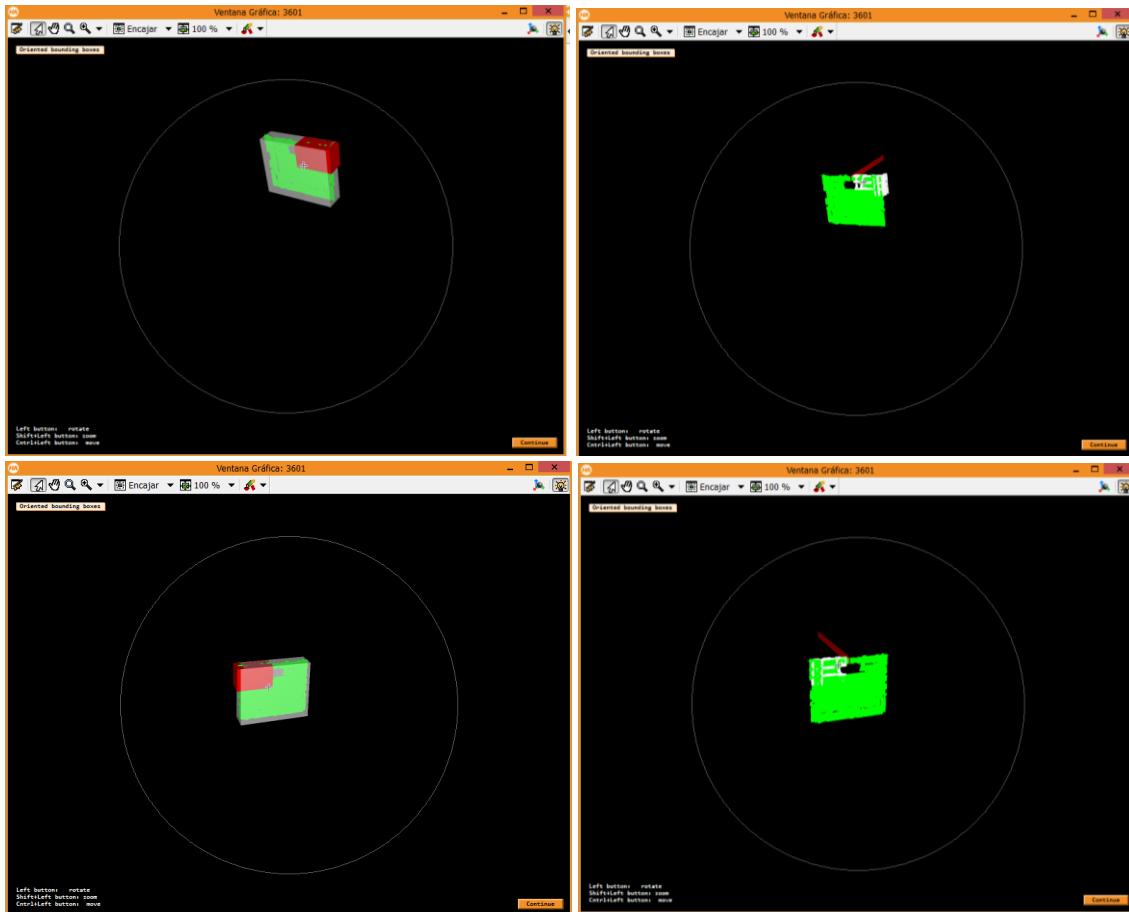


Figure 82: Looking for the left and right border of the container.

Once the program already got the three dimensions of the boxes following the same process, it will create a box simulating the size of the container in the same pose that it has in the reality.

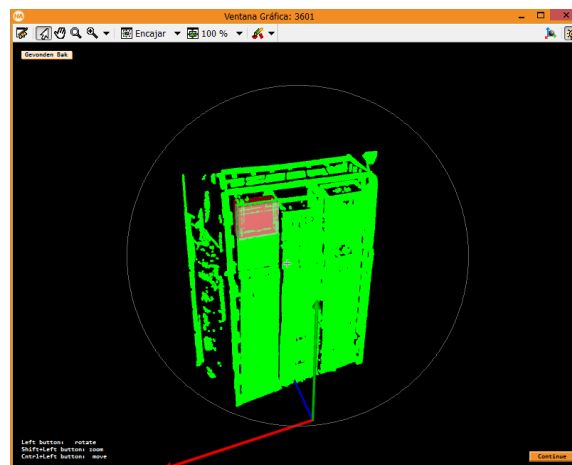


Figure 83: Showing the 3D representation of the container.

Last checks of the process

Finished this representation and knowing the container dimensions are necessary to check if the chosen container is removable. This can be known checking if there is something up or in front of. The technique is like the previous steps. First is to create a box in the correct pose, then intersect and finally check if there are any points. If this happens, it means that there is another box and the program jumps to analyze the following container

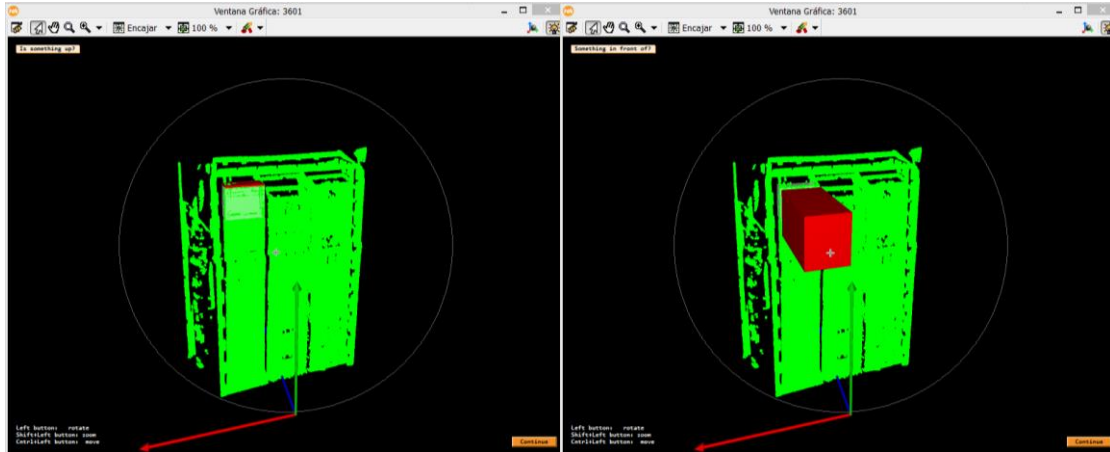


Figure 84: Checking if there is something on the top or in front of the container.

Showing the results

If the selected container has passed all tests, it is considered like validated container to be removed. Then the program show on screen the location and pose, as well as the detected height of the container. This information should to be sent and treated to be interpreted by robot before picking the container.

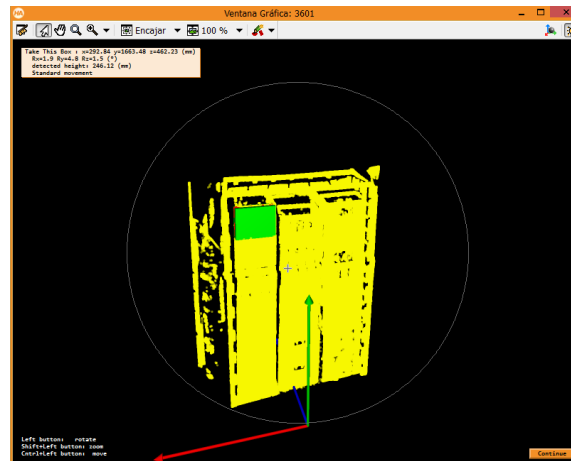


Figure 85: Show the final representation, location and pose of the selected container.

The following step is to go to the next iteration taking the 2nd closest container from the referent point fixed in the middle of the rack top tube. This means the whole detection process is going to be performed until the upper containers have been removed.

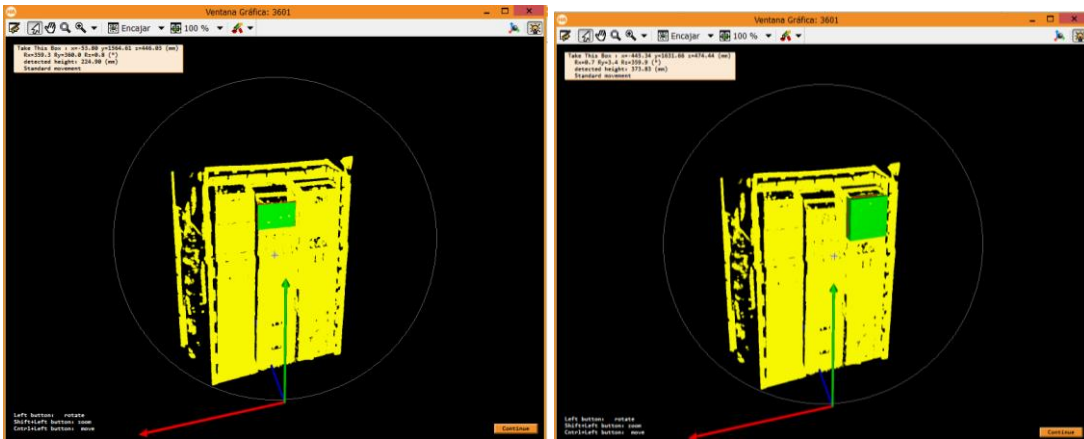


Figure 86: Show the final representation, location and pose of the next containers of this scan.

Finally when the program has been executed the iterations, it is going to jump to take a new scan. However not before releasing the memory, which is occupied by 3D models with the function *“clear_object_model_3d”* and the images with *“dev_clear_obj”*. As well as decrease the value for the detection area rectangle because of the assumed lower height that the upper containers have in the second scan.

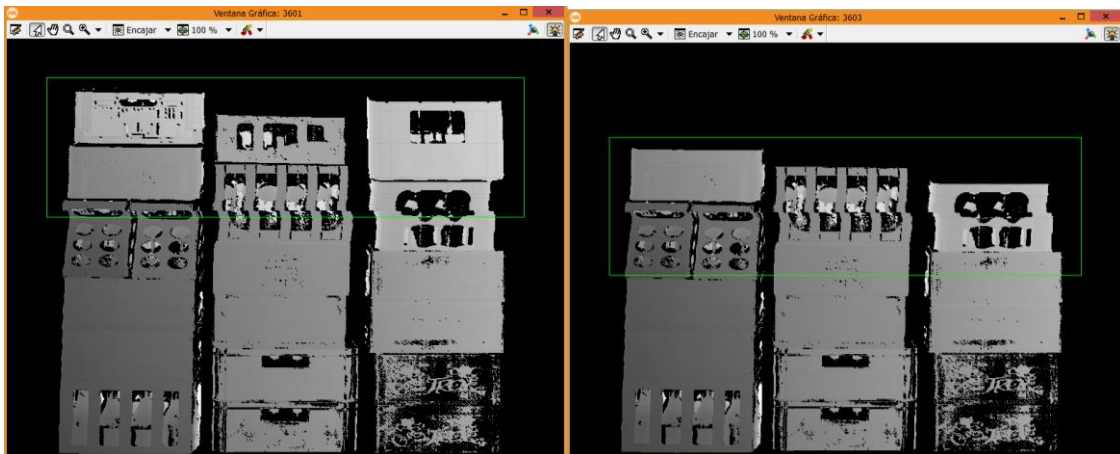


Figure 87: Differences in the detected area between first and second scanner.

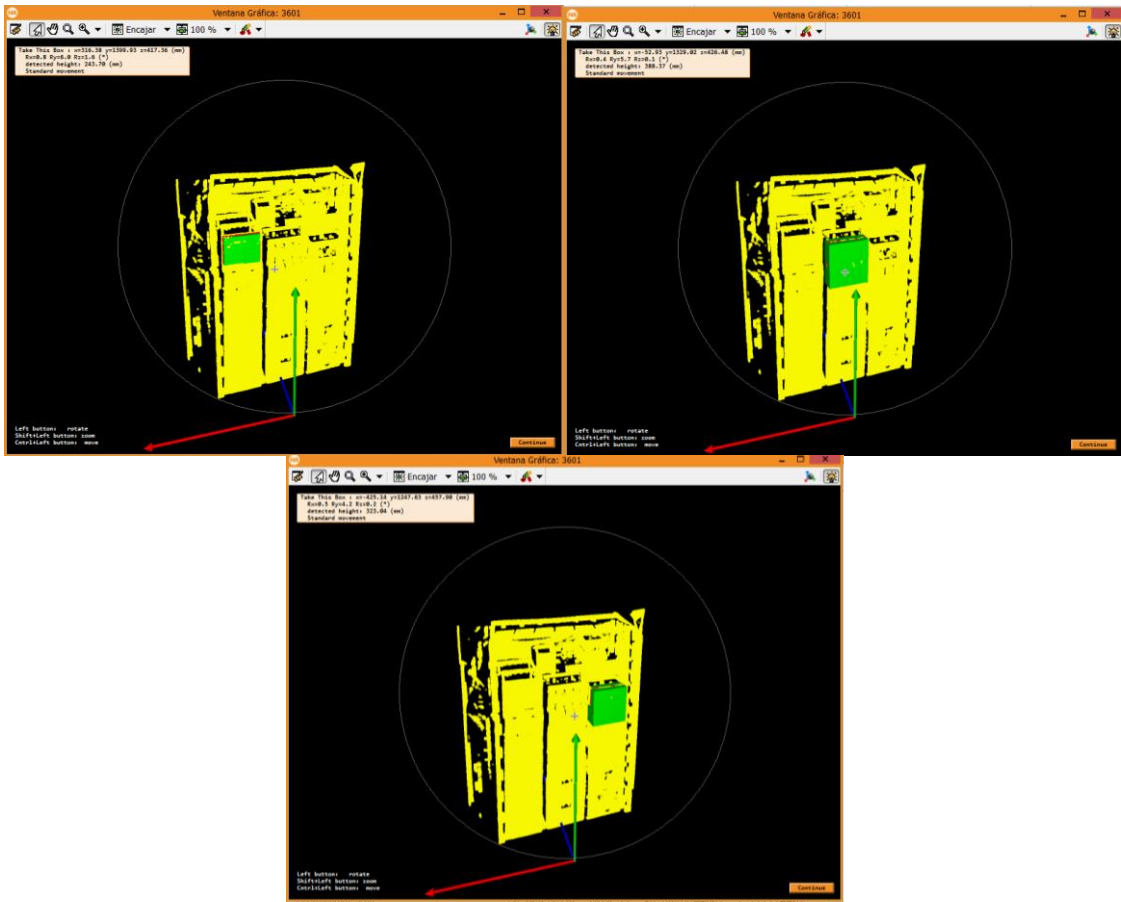


Figure 88: Show the final representation, location and pose of the second scanner containers.

4.5 Conclusion

To obtain accuracy results is needed to obtain a 3D model as similar as reality. Therefore, special attention must be paid to the factors that affect the creation of the cloud of points that would form the 3D model. These factors are mainly, first the parameters of the calibration of the sheet of light setup and second the number of images taken per encoder revolution.

It was seen that the calibration of the movement has a high influence, because changes in the number of steps between two successive photographs cause inaccuracies in the results of measurements. In addition, a smaller distance between the measurement system and the object to be measured produces better results. For this reason, successive changes were made to the position of the rack to find the location where the best results of the 3D model are obtained.

5. Conclusion

5.1 Overview

The aim of the project is to develop a bin-picking setup for emptying metal racks filled with empty beverage containers. For this task, a measurement system based on 3D vision has been developed. A system based on Sheet-of-light setup has been implemented. This thesis involves all the steps of the vision system, from the installation of the hardware and its calibration to the acquisition process and the creation of a code on the Halcon application that controls the procedure.

5.2 Future work

If we analyze the current system, the following steps are the improvements of the accuracy of the scanner and the calibration. Afterwards, try to reduce the influence of the other factors that cause disturbances in the scanner.

For the calibration, it could consider the use of better rail structure that does not have an assembly defect in the middle of the trajectory.

Finally, could be considered for the current project, the improvement of the recognition of the selected container sizes. Because it could have some problems in the detection when there are containers too close each other.

However, to adopt this method it is necessary get better results in the container reconstruction. This could be achieved with more precision calibration, as well as add a new laser, and that permits have two sheets of light instead of only one.

References

- [1] Andrew Wilson. (2014, 12). Choosing a 3D vision system for automated robotic applications. <http://www.vision-systems.com/articles/print/volume-19/issue-11/features/choosing-a-3d-vision-system-for-automated-robotics-applications.html>.
- [2] MVTec-Halcon-12.0. (n.d.). Solution guide III-C 3D Vision. Chapter 5, page 137.
- [3] MVTec-Halcon-12.0. (n.d.). Solution guide III-C 3D Vision. Chapter 8, page 213.
- [4] MVTec-Halcon-12.0. (n.d.). Solution Guide III-C 3D Vision. Chapter 6, page 175.
- [5] Photonfocus. (n.d.). User Manual- MV1-D2048-3D03 Camera Series. Man052 09/2014 V2.6.

Attachments

In this section, it will see the index of all datasheet of the used devices.

- PLC
- Encoder
- Frequency converter
- Induction sensors
- Photonfocus Camera
- AC Motor
- Worm gear reducer

All of this datasheet are in the extended version in the folder destined to store these files, which is called ('PDFs devices').

Moreover, a 3D guide for Halcon program is attached as well in the folder.