FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Trajectory Generation for a Robot Manipulator using data from a 2D Industrial Laser

Diogo Filipe Moreira Meireles Gomes

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

Supervisor: Vítor Hugo Machado Oliveira Pinto

July 27, 2022

Abstract

Nowadays, the automation of factory floors is necessary for extensive manufacturing processes to achieve the ever-increasing competitiveness of current markets.

The technological advances applied to the digital platforms have led many businesses to automate their manufacturing processes, introducing robotic manipulators collaborating with human operators to achieve new productivity levels, manufacturing quality, and safety. However, regardless of the amount of optimization implemented, some quality problems may be introduced in production lines with many products being designed and produced.

This project proposes a solution for feature extraction that can be applied to automatic shape and position detection using a 2-Dimension (2D) industrial laser to extract 3-Dimension (3D) data where the movement of the item adds the third dimension through the laser's beam. The main goal is data acquisition and analysis. This analysis will later lead to the generation of trajectories for a robotic manipulator. The results of this application proved reliable given their small measurement error values of a maximum of 2 mm.

Keywords: Robot trajectory generation, 2D Laser Scanning, Point cloud, Data acquisition

Acknowledgments

Although the writing of this document is an individual work, it also represents the culmination of a five-year journey where I've been surrounded by many people who helped through all this time. As such, some acknowledgments are in order.

I want to begin by thanking my parents for being the most important people in this journey, supporting me throughout my entire educational journey, and helping me through challenging times.

Special thanks to my supervisor Vítor Pinto, and co-supervisor, José Gonçalves from Instituto Politécnico de Bragança with whom this project was developed with, for entrusting me with the confidence, necessary tools, and help to develop this project.

A big final thank you goes to all the friends I was lucky to have made through all these years who helped me whenever I needed them. In the course of these sometimes long felt five years, the times passed in this university seemed so short because of the amazing times I got to spend with them, and they are what made it all so special.

Diogo Gomes

"Valeu a pena? Tudo vale a pena
Se a alma não é pequena."

Fernando Pessoa

Contents

List of Figures

Abbreviations

FEUP	Faculdade de Engenharia da Universidade do Porto
IPB	Instituto Politécnico de Bragança
2D	2-Dimension
3D	3-Dimension
OLP	Offline Programming
AGV	Automated Guided Vehicles
LiDAR	Light Detection And Ranging
CAD	Computer-Aided Design
ASPRS	American Society for Photogrammetry and Remote Sensing
RGB	Red, Green, Blue
PCA	Principal Component Analysis
ROI	Region Of Interest
SVM	Support Vector Machine
SLAM	Simultaneous Localization And Mapping
RANSAC	RANdom SAmple Consensus
CVS	Computer Vision Station
MFC	Microsoft Foundation Class
DNN	Deep Neural Networks
CNN	Convolutional Neural Networks
RF	Random Forest
AFP	Automated Fiber Placement
VG	Visibility graph
VD	Voronoi Diagram
OCC	OPEN CASCADE
HRI	Human-Robot Interface
OCC CAD	OPEN CASCADE Computer-Aided Design
PID	Proportional–Integral–Derivative
ROS	Robot Operating System
PLC	Programmable Logic Controller
SPI	Serial Peripheral Interface

Chapter 1

Introduction

1.1 Context

In modern industry, one major issue is feature extraction and data processing. Since these highly influence product prices, their detection and correction have become a necessary subject to consider when providing a service or designing and creating a product. However, these tasks usually require very specialized and skilled operators, thus making it difficult to automate processes.

To create a more automated production, several businesses and factories have used some methods like the implementation of conveyors, robotic manipulators, and Automated Guided Vehicles (AGV) within their workspaces. For example, a conveyor can transport materials and products through the factory floor. It is a valuable way to move the product around the factory without an operator, enabling it to analyze and detect defects in each product on the line. Typically, lines have workers assigned to check and evaluate the quality of each item. However, with the growing concern about quality control, there is an increase in strategies that can detect defects of some products using technologically advanced solutions.

1.2 Motivation

In the ceramic industry, after production, the products have small amounts of excess material on the edges due to molding and casting, which require polishing. This operation is performed using a cleaning system consisting of a rotating sponge, and the process is done automatically in a machine if the plates are circular but must be done manually if the shape is not circular. This happens because the sponge is static, and the dishes are rotated and moved towards it. For non-circular parts, this process can not be used. The circular products are processed due to their shape, and the distance to the sponge is always known due to its constant radius. However, to improve and grow as a business and introduce a variety of new Stoneware Tableware, it is necessary to implement an automatic robotic solution to account for more possible product shapes.

The developed work is part of a larger project, identified as "STC 4.0 HP (New Generation of Stoneware Tableware in Ceramic 4.0 by High Pressure Casting Robot Work Cell) Project" [1], which aims to establish the production conditions that will allow GRESTEL to worldwide launch a range of innovative products in fine stoneware tableware, as a result of the combination of different Ceramics 4.0 technologies of the industrial value chain, with robotic finishing and artificial vision control, supported by new knowledge and technological developments at industry 4.0 level. This project aims to robotize an industrial process that, until recent times, would have been almost unthinkable. With the advent of collaborative robots and the advancement of fundamental research in industrial robotics, today, it is possible to approach the robotization of all forming processes, including high-pressure pressing, drying, and subsequent finishing.

1.3 Scientific Contributions

In the course of the developed work, it was expected that the features of ceramic items be acquired and processed to allow for the automation of the polishing process. Said features include the piece's center point, border height, and distance of the border to the center point according to its angle. This approach no longer requires human resources, which can be transferred to a more productive setting.

An industrial laser system is used to acquire data from each part and transmit it for processing in a local application. For this process, the detection of the part will be performed in a conveyor using a 2D laser scanner.

After detecting and identifying the part's shape, the piece's polishing can be performed in several different ways. However, in the presented approach, a robotic manipulator is used for the polishing process, combined with a polishing sponge designed explicitly for this task.

This project focuses on the detection of ceramic plates' shapes with the intent of center position and identifying the object's border's height as well as the distance between edges and the center point. Furthermore, a robotic collaborative manipulator equipped with a vacuum gripper will remove the ceramic plate from the conveyor and move it close to a cleaning system to perform the polishing of the finished product. This movement is programmed with the values extracted from the laser.

Thus, the presented project introduces an innovative technique of robot path generation due to its dynamic position and shape detection considering the acquired data, using point clouds and depth matrixes.

The robot's path is determined considering the features identified by the dataset and is comprised of different steps. Firstly, it grabs the piece by dynamically obtaining its localization, transports it to the expected polishing station, and returns the polished piece to its original location or a specific one.

1.4 Document Structure

This report introduces the project's context, motivation, and objectives. Followed by a brief introduction to each subject studied during this project and a bibliographical review containing several alternatives as possible solutions to the presented problems, focusing mainly on image acquisition and analysis. The equipment and setup used for testing are exposed in detail, and the overall system setup. This is followed by the work developed during the project, including algorithms and results. Finally, the following steps of this project are presented, given that this is a modular part of a more significant project requiring incorporation later.

Chapter 2

Bibliography Review

2.1 Image Acquisition and Analysis

In modern times, many automated industrial applications require image acquisition. A very commonly used technique is point clouds.

Point clouds are data sets that represent objects, shapes, or spaces. Each point represents the sampled surface's X, Y, and Z geometric coordinates. Point clouds are collections of individual points plotted in 3D space. These points allow the reconstruction of the sampled surface in a digital environment and study it following the requirements.

According to the information in [2], this approach is used in several different applications and categories, for example, in the construction industry, such as 3D model reconstruction and geometry quality inspection, as shown in figure 2.1.

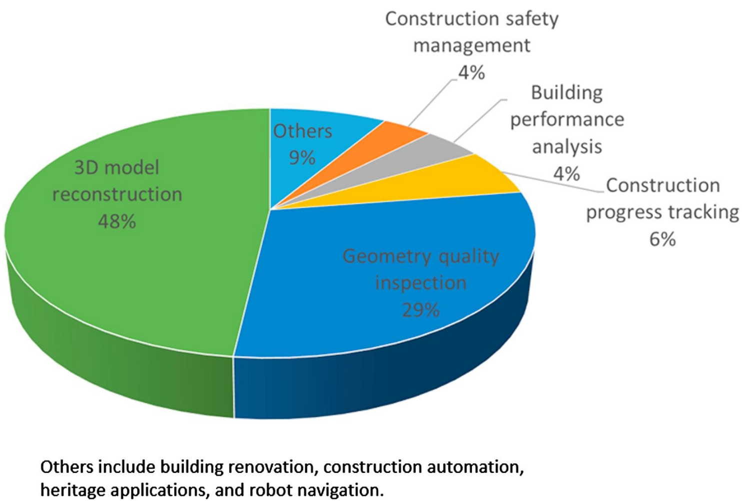

Figure 2.1: Point cloud applications in the construction industry
(Source: Adapted from [2])

A relevant topic for this work is the reconstruction of the geometric model, which consists of creating a point cloud of the object with the data retrieved from sensors.

Datasets are typically obtained using 3D laser scanners and LiDAR (Light Detection And Ranging) technologies. These can also be used in geographical applications and are capable of mapping 3D models of terrains or urban environments. A higher density of points collected leads to a higher level of detail in the finished 3D model.

To create a 3D model of the object, the dataset is registered, converted into a mesh, and analyzed in software such as CAD (Computer-Aided Design) ([3], [4]). Registering is done by aligning overlapping point clouds. For example, if provided with datasets of the same object from different angles and positions, the registration can produce a finished point set of higher density and detail, as is the example of Figure 2.2. Meshing consists in converting the point cloud data into triangles or polygons and representing the surfaces of the objects.

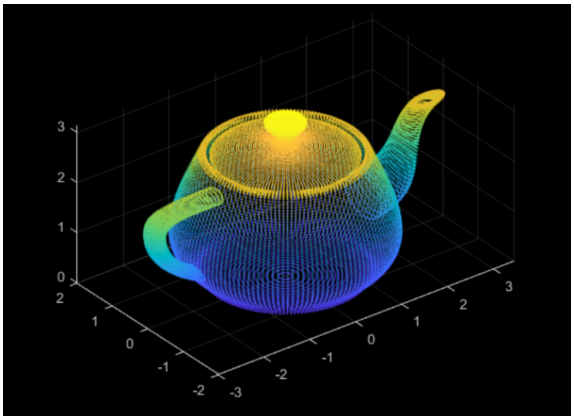

Figure 2.2: Example of a point cloud
(Source: Adapted from [5])

Point clouds are present in several industrial applications with different goals, but one of the biggest is focused on increased accuracy and is obtained using several other methods. The study presented by the authors of [6] provides a set of possibilities for point cloud data acquisition which includes:

- **3D Laser Scanning and LiDAR**

  Data points are collected using laser beams which allow measuring the distance of the point to the scanner by detecting the reflected beam from the object to be scanned.

  3D Laser scanning is a non-contact technology that captures the shape and surfaces of physical objects or spaces using linear laser light [7].

  Using a data processing software associated with the laser scanner, the user is presented with a point cloud created from datasets of the detected surface. The scanner works with pulses of light sent to the object's surface. The object reflects the light back to the scanner, which is captured by its built-in camera after a specific time. The time measured between sending and receiving allows the scanner to map the points in 3D space.

- **Photogrammetry**

  According to the American Society for Photogrammetry and Remote Sensing (ASPRS), Photogrammetry is a method of obtaining trustworthy information regarding physical objects and their surrounding environment using photographic images [8]. This technique uses multiple photographs taken from different angles of the target object. Later, these photos are overlapped, and already developed algorithms can estimate the relative locations of these images. The final result is a 3D point cloud of the object.

- **Videogrammetry**
  Videogrammetry is similar to photogrammetry, but instead of using several photos to generate the point cloud, it uses video streams as input. This method is also capable of a progressive reconstruction of the point cloud due to the sequential nature of the video frames. According to [9], there are several advantages to using this method in certain situations, which include:
  - 3D and non-contact measurements;
  - Large number of objects (targets);
  - Moving objects;
  - Good temporal resolution requirements;
  - Precise and reliable results requirements;
  - Fast recording and processing requirements (on-line solutions).
- **RGB-D Camera**
  An RGB-D camera is an RGB (red, blue, green) camera equipped with a depth sensor. The generated point cloud is colored with XYZ coordinates and RGB values.
- **Stereo Camera**
  A stereo camera has two or more lenses and a separate image sensor. 2D images of each lens obtain the 3D point cloud since one lens's relative position, and orientation to the other is known beforehand.

The analyzed alternatives provide a better understanding of the acquisition and processing of point cloud data. It becomes clear that 3D laser scanning technology is one of the best choices for implementations that require high levels of precision, and the preprocessing of the points data may be a necessary step to remove undesired values from the datasets and make the overall system more stable and capable of performing its designated tasks. A comparison between these methods is also performed in [6], and it is concluded that 3D laser scanning has a higher accuracy and measurement range than the other mentioned technologies.

One of the many applications of 3D laser scanning is the analysis of recovered archaeological finds such as ceramic items. The work presented in [10] provides an example of this application by classifying the friezes engraved on ceramic sherds. Sherds are broken pieces of ceramic material, mainly found on an archaeological sites. The purpose of this project is the automated classification of the engravings of the ceramic pieces. The ceramic sherds are scanned using a Next Engine 3D laser scanner with an accuracy of 0.127 mm. The generated point clouds have 268 000 points per square inch. The overall process is divided into two parts: the classification of sherds and the manual stamping performed by an archaeologist. The automated process takes about 7 minutes and 30 seconds to complete: 1 minute and 30 seconds per sherd for the classification and 6 minutes for the manual stamping, thus improving the time spent due to the classification system. The obtained 3D point cloud is filtered for outliers and projected onto a 2D plane as a depth map to reduce computational cost. The points are referenced to the scanner and defined by x,y, and z coordinates. The orientation of the sherd during scanning proves to be a difficulty when later on trying to analyze the point cloud. The authors propose a Principal Component Analysis (PCA) on the 3D point cloud. The final result is a point cloud aligned with the XY plane by a rigid transform. However, the 2D projection of the point cloud doesn't create a regular image sampling. The gaps and imperfections of the point cloud are corrected by applying dilatation and median filter, followed by a smoothing estimator along the image gradient, which considers the weighted average of the neighboring points. The final depth map is presented in a grayscale where each point's grey level is proportional to its z value. The 2D depth map is binarized to extract its binary pattern using the Niblack method([10]):

$$S(i,j) = \mu_w + k\sigma_w \tag{2.1}$$

- μ_w : mean grey level in the window
- σ_w : standard deviation of the grey level in the window
- k : constant value set at 0.2

Lastly, manual stamping requires selecting a region of interest (ROI) centered on the previously obtained binary pattern. Finally, the classification is performed by training a Support Vector Machine (SVM), a supervised learning method for classification and outliers detection. The final results provide a correct classification rate of 74% on the binary patterns extracted from the 3D laser scans, which, although a lesser value than the success rate of the manual stamping process, provides a much faster recognition time, which allows for a more efficient archiving of the pieces.

The work in [11] presents a way to build a 3D point cloud map with a mobile laser scanner. The setting is a residential area where the larger map is created by aggregating generated sub-maps of smaller areas. The 32-layer laser scanner used in this project was Velodyne HDL-32E. To correctly map the setting, the vehicle can identify its position and orientation using landmarks. The vehicle's position and orientation are determined by applying the normal distributions transform based on Simultaneous Localization And Mapping (SLAM) and generating the map's point cloud. The vehicle's pose and environment landmarks are recorded and mapped in a graph-based SLAM.

Sub-maps are merged by detecting overlapping points, calculating the vehicle's relative position, and matching the corresponding overlapping landmarks. The approach used in this work provided the authors with very effective results, making way for the attempt at mapping larger areas.

Considering the handling of point cloud data, the authors of [12] present a solution using a bandpass filter, a statistical filter, and a RANSAC (RANdom SAmple Consensus) algorithm. The bandpass filter allows selecting points within a defined range based on constraints for one field of the point's values. A statistical filter is used to clear the dataset of outliers by computing the neighbor distances in the input dataset. Also, a RANSAC algorithm is used to acquire points of interest for the analysis. This step is required because the previous filters cannot remove unnecessary points on the same plane as those of interest. The overall process is shown in figure 2.3.

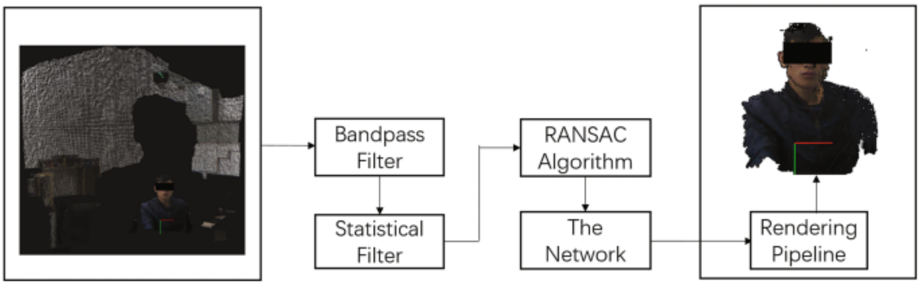

Figure 2.3: Preprocessing and transmission overall process
(Source: Adapted from [12])

The work in [13] exposes a prototype computer vision station (CSV) for real-time defect detection in ceramic tiles. The system was implemented to try and increase the defect detection rate, which was previously performed by human workers and was more likely to be influenced by each worker's factors like health conditions. The system's application is a Microsoft Foundation Class (MFC) GUI application, and the algorithms were implemented in C++ using OpenCV and Nvidia CUDA libraries.

Tiles are split into two classes: correct, presenting no visible defects; defected, showing imperfections on its surface, edge, or corners. Defect detection on edges and corners is performed using thresholding and contour tracing algorithms. Surface defects are detected with a statistical method, in which the image is divided into smaller pieces where the statistical measures are calculated. The flaw is detected by comparing the acquired values and the previously obtained measurement values template of each tile.

The final implementation was tested on 520 tiles divided into three different tests, and the results were compared to those from work performed by quality staff members. The final results show an accuracy of 98.65% of correctly classified tiles, presenting a satisfactory outcome for the image analysis algorithms used in this approach.

The authors of [14] present an automatic defect detection and management system that can detect and quantify defects in porcelain pieces based on machine learning and computer vision. The system was implemented using a convolutional neural network capable of analyzing images of each product and determining if there are any significant visible imperfections. The main goal of this work is to optimize the production flow and reduce production costs. The author describes deep learning as an emerging technique consisting of a composite neural network model capable of satisfying results in several applications such as voice, image, or video recognition and robotics. A neural network comprises input and output layers and one or multiple hidden layers between these extremities. Deep neural networks (DNN) have the advantage of being able to increase the complexity of their successive layers and representing non-linear functions. However, the difficulty in using Neural networks is correctly training them with the proper values. Convolutional neural networks (CNNs) are mainly used for image and speech recognition. The convolution kernel is a 2D structure consisting of multiple layers (figure 2.4), whose coefficients determine how each pixel's value is processed. The final value of the pixel is a weighted calculation considering its neighboring pixels' importance. This analysis allows for the extraction of features from the dataset.

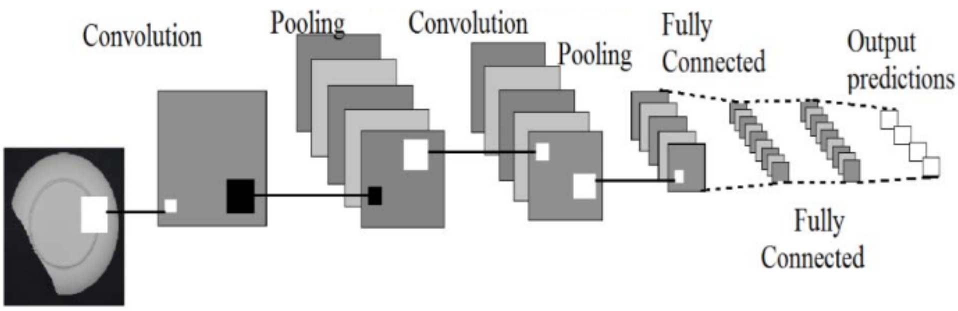

Figure 2.4: Layers in a convolutional neural network used for defect detection in plates images (Source: Adapted from [14])

The system was implemented on a conveyor in collaboration with a robotic manipulator responsible for removing any defective product from the conveyor belt, as shown in figure 2.5.

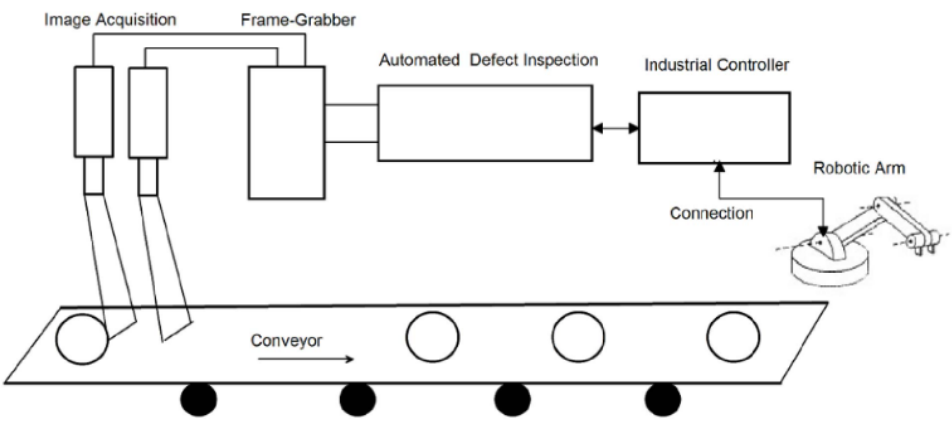

Figure 2.5: Defect detection system workflow
(Source: Adapted from [14])

The final results were achieved through the comparison of different algorithms. The one that provided the best results was Convolutional Neural Networks compared to others such as Support Vector Machines (SVM) and RandomForest (RF).

The work in [15] presents a defect detection technique based on 3D laser scanning for automated fiber placement. In the composite industry, materials such as carbon fiber require an Automated fiber placement (AFP) layup inspection to guarantee the quality and efficiency of the overall process. The analysis can be performed using a laser profile scanner to achieve a significantly detailed point cloud during the layup. This work exposes an algorithm called cross-sectional line-processing, which processes each line retrieved by the scanner and clusters the results together. The resulting point cloud is used for segmentation to recognize the visible layup defects and quantify them. The algorithm is executed in two steps:

1. **Feature extraction**
   Segmentation of single lines to find defects and layup features.

2. **Clustering**
   Obtain features clusters of the laser lines

The system was tested on a setup comprised of a laser profile sensor, a three-coordinate mobile platform, a computer, a calibration block, and fiber pieces for testing. The testing parts were handmade with pre-determined measurements to allow correct results analysis.

The final results of the implementation proved successful in recognizing defects in real-time and that the scanner can detect surface defects with high accuracy and precision. However, these results are not entirely correct as the algorithm can still mix the flaws due to their similar appearance. Testing showed that the proposed algorithm could process up to 200 laser profiles per second, and the correct recognition of defects can reach an accuracy of 78%.

The authors of [16] present a method to determine the best object grasping position and pose based on the item's point cloud. The 3D scanning technology used was a Keyence line laser sensor, which was mounted on top of a conveyor belt allowing for the read of the object passing below. This setup also included a robotic manipulator equipped with a two-finger gripper that was later used for testing the determined grasping positions. The laser scanner can give the user a point cloud of the scanned object. This approach uses the point cloud data to determine the object contour points which are determined by searching for extreme points of the object, which are later used to calculate the center lines of the X and Y-axis of the point cloud. The intersection of these lines represents the center point of the object. The next stage is the generation of the grasp cuboids, which is explained in the following steps:

1. The process begins by detecting the highest point of the dataset. The resulting point is used to generate a new point cloud model resulting from the intersection with the original model with a height of 40 mm below the highest detected point.

2. Following the previous step, the author can extract the contour point of the point cloud model, similarly to the previously explained technique, determining the model's center point

and generating the cuboids accordingly. The generated cuboids have a length, width, and height of 80 mm, 22 mm, and 40 mm, respectively. The result is four differently positioned cuboids with the same center point.

3. Finally, the algorithm calculates the contour points captured by the cuboids and generates their scores. Scores determine which of the four cuboids provides the optimal grasping solution. The scores are calculated using a weighted coefficient for four pre-determined rules, which take into account, for example, the smallest distance of contour points to the center and the minor average error between fitting line and points.

The overall process is described in figure 2.6.

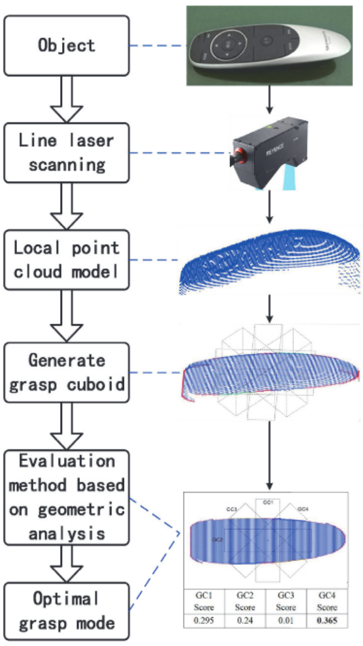

Figure 2.6: Grasping position process flow
(Source: Adapted from [16])

The final results show a very high level of grasping pose recognition accuracy of 99% from the set of 11 test objects, thus proving a high success rate of this approach.

The sensor industry has many laser equipment alternatives and the Gocator brand is present in several applications. In [17] an approach using Halcon Vision software is presented. The program works with a Gocator 2340A laser profile scanner and can show the 3D model of the analyzed surface and count the number of holes in the item. The program is exported to a C# file and used in Microsoft Visual Studio without the need for HALCON HDevelop, which was the environment in which it was initially designed.

In [18] the authors propose a solution for a path planning algorithm meant for a spray robot using 3D point clouds. The items for this application are furniture, and the image data is acquired through a gocator laser sensor. Image data is analyzed using the 3D point cloud model of the pieces, which is used for edge extraction. This information is later used for the correct positioning of the spray gun.

Analyzing the previous data acquisition applications, it is possible to choose a 3D laser sensor as an adequate technology for developing an image acquisition and processing system.

2.2 Robotic Manipulators

In recent years, the use of automation, more precisely robotic alternatives, has been incorporated in several industries with specific applications in each different field. Robotics is currently present in various applications, from health care to manufacturing. Many consider it essential to each company's competitiveness in today's market.

Robots have become an essential part of today's industry. In the context of industrial robots, a manipulator is defined as a sequence of rigid elements connected by joints and is characterized by its arm for maneuverability, a wrist that provides dexterity, and an end-effector that allows for the completion of specified tasks assigned to the robot [19] .

Regarding the ceramic industry, the automation robotic incorporation may be described as a slower process due to the fragility of the finished products and the precision required while handling such items.

Definitions of an industrial robot ([20]) describe it as an automatic manipulator programmable in three or more axes, either fixed or mobile in industrial applications. A robot is used in mainly three different industrial applications:

- **Transport** - Moving parts between work spaces
- **Additive Manufacturing Process** - Assembly, painting
- **Subtractive Manufacturing Process** - Drilling, grinding

An industrial robot is composed by the following components ([20]):

- **Manipulator**
  Considered the robot's main body and consisting of a base and arm, the manipulator is the mechanical and electrical device that provides the system with several degrees of freedom. The movement of the manipulator is directly dependent on its controller.
- **Controller**
  The robot's flexibility is given by its mechanical construction and how it is programmed—reprogramming the device provides the user with more versatility in executing different tasks. This component is responsible for generating the manipulator and tool's path and trajectory using the system's cartesian coordinates and joint's motion.
- **Tooling**
  A manipulator's end tool allows for performing particular tasks such as polishing, drilling, or welding. End tools, such as mechanical or vacuum grips, electromagnets, or hooks, can be of diverse types.

2.3 Path Planning and Trajectory Generation

Robot paths and trajectories are widely studied subjects. While these subjects share the same goal, which is the correct movement of the manipulator, there are some essential distinctions.

Path planning is introduced([19]) as a purely geometric matter and defined as the generation of a geometric path, with no mention of any specified time law. The goal is to find a collision-free motion between the starting and final positions. There are several different kinds of possible paths which can be simply geometrical or more dynamic, including obstacle avoidance or simultaneous work with other robots with significant degrees of freedom or humans. To better understand this subject, the author exposes some definitions such as configuration space (C-space), which is the entire available space; the space of free configurations (C-free), free of obstacles, and the obstacles' representation in the C-space (C-obs);

Path planning is divided into three main categories:

- **Roadmap techniques**
  In a roadmap algorithm, several nodes and links may have a physical meaning. Nodes can represent locations, and the links correspond to the path between these locations. The possible paths are mapped similarly to a graph, and the search for the best path can be reduced to research in graphs. There are several different research techniques, such as:
  - Visibility Graph (VG)
    In a two-dimensional C-space, nodes are vertices of the obstacles, and the links between nodes exist if they are visible. This way, there is a straight line between nodes and no barrier between them. The generated path is close to obstacles.
  - Voronoi Diagram (VD)
    Set of points that are equidistant from two or more obstacles. C-space is divided into regions, and each one contains only one object. Any point in each region is closer to that region's obstacle than any other. The generated path is far from the obstacles.
- **Cell decomposition algorithms**
  The C-free space of the robot is subdivided into several regions, and the path between configurations is inside the same cell. It is possible to represent the connections between cells based on their adjacency and define these in a graph. Following the previous examples, path planning is again based on graph-searching techniques.
- **Artificial potential methods**
  Artificial potential methods are different from the traditional approaches. The robot is considered a moving point in the C-space in this approach. The potential field determines the path by the final goal position, which generates attractive potential, and the obstacles which generate repulsive potential. The sum of these potentials achieves the final path. As such, the direction of the robot's motion can be determined regardless of its configuration.

Trajectory planning, on the other hand, is performed by a user-developed algorithm that takes the previously planned path and the physical, kinematic constraints of the manipulator and is defined by assigning a time law to the geometric path. As such, for each point in time, it determines where the robot must be positioned. The final output of the trajectory is given by the joint's route or the end-effector as a sequence of position, velocity, and acceleration values [19]. The user can define the end-effector's position and orientation at each time interval.

Usually, the classification of trajectories is based on previously decided upon criteria which may vary, but some of the most used and relevant are:

- **Minimum execution time**
  High productivity is an essential aspect of the automated factory floor. One crucial factor in productivity is the task execution time. The overall process becomes more efficient if an item can be produced in a smaller timeframe. This is a widespread and recurring criterion when evaluating trajectories.

- **Minimum energy (or actuator effort)**
  Like minimum execution time criteria, energy spent is essential in factory plants. This criterion aims to minimize the energy consumed or the actuator's efforts, thus complying with energy-saving requirements imposed by economic reasons or by scenarios where the actuator may be limited to a certain amount of supplied power and account for the actuator's maintenance and lifetime.

- **Minimum jerk**
  A commonly used option to obtain smooth profiles of the actuator's torques is to limit the jerk, defined as the time derivative of the acceleration. According to the author, minimization of this value provides the system with solid results: more minor errors during the tracking phase and reduction of resonance frequencies and stresses induced to the overall mechanism.

A popular solution for robot path generation is using the CAD model [21] created from a 3D point cloud of the object [22]. These data models can produce highly detailed results, so it will require posterior filtering to retrieve only the desired points and shape of the trajectory to be considered. Reducing the number of points to analyze also ensures that the trajectory calculation is faster. However, an adequate number of points must be kept so as not to lose valuable information about the object to maintain the quality of the overall process and ensure the correct handling of the finished product.

The process of path generation will go through a phase of shape detection, followed by desired face detection. In the example of Figure 2.7, an additional step is also performed, which refers to choosing the edges of the selected face. These solutions are mainly used with the working tool placed on the tip of the robotic manipulator.

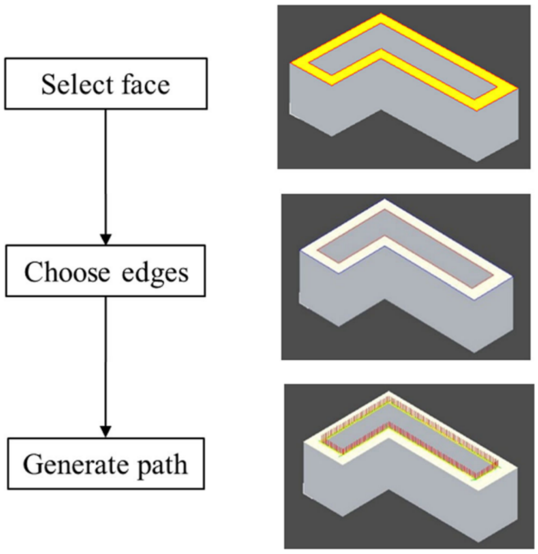

Figure 2.7: Steps to generate path
(Source: Adapted from [22])

Regarding CAD models, [23] illustrates a way to use OPEN CASCADE to plan a robot path using such models. The author presents an Automated Offline programming system (AOLP) that uses an offline platform combined with an autonomous object pose estimation method and a three-dimensional vision system. The robot's pose is later used in the OLP to generate, simulate and execute the robot's path and functions. Similar to other approaches, CAD designs are essential for the correct path generation. In this work, a depth sensor (Kinect) was selected for extracting three-dimensional information about the object and environment, which is used to locate the piece in the work cell, regardless of lighting conditions. The generated path is later executed on an industrial manipulator (Denso 6556), and the overall process architecture can be analyzed in figure 2.8.

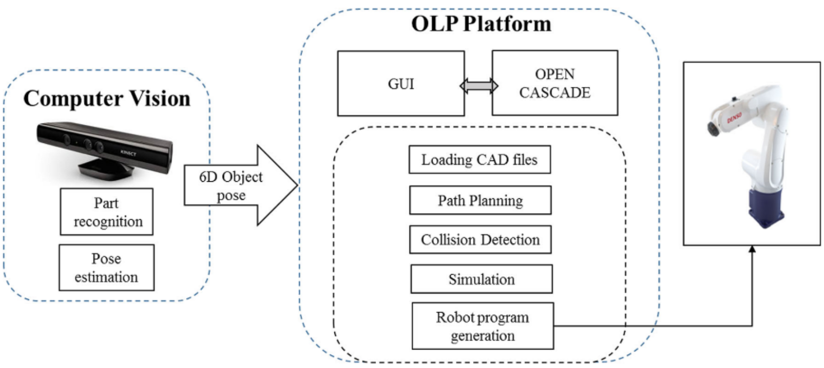

Figure 2.8: Process architecture
(Source: Adapted from [23])

This approach uses an offline programming (OLP) platform with OPEN CASCADE's open-source libraries and Microsoft Visual Studio. One of the advantages of using an OLP is that the user can represent, model the robot, extract the CAD model and simulate the program before programming the actual system with the calculated trajectory. The developed program works as an HRI (Human-Robot Interface). The process starts by extracting information from the CAD model using OCC CAD (OPEN CASCADE Computer-Aided Design) kernels to obtain the position values. As illustrated through Figure 2.9, the determination of the position values is initiated by the user's selection of the face/wire/edge on the model. After this process, points are created and represented on the curve. This approach also makes it possible to obtain the orientation of the workpiece, which is defined by a transformation matrix.

The work presented in [23] exposes some ideas that can be implemented, such as the use of an offline programming platform, the simulation of the robot's trajectory before programming the actual system, and the use of OPEN CASCADE open-source libraries as a way to analyze and use the CAD model.

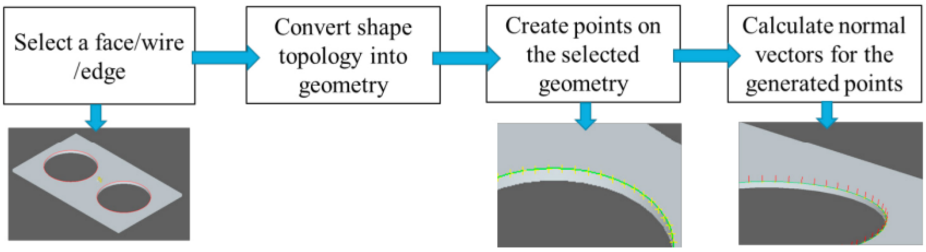

Figure 2.9: Steps to generate points
(Source: Adapted from [24])

The work presented in [25] uses a method that creates a path for 3D grinding using online measurement data of workpieces with complex surfaces. This approach uses a 3D laser scanner and point cloud data. Since the point cloud data is of very high density, the author proposed two different topologies:

- **Octree**
  Fast and convergent method in which each node has eight children [26], as shown in figure 2.10. If every internal node of the Octree contains exactly eight children, it is called a full Octree.
  Possible use of this technique presented in [27] implements an open-source (using a C++ library) framework, and its mapping can represent occupied free and unknown spaces. In addition, a method of compacting 3D models based on octrees is also proposed.

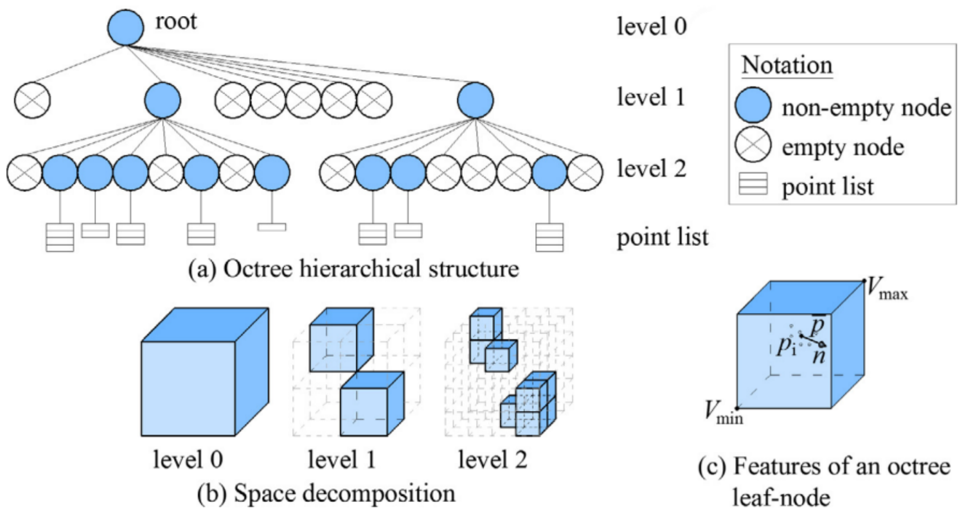

Figure 2.10: Example of octree decomposition
(Source: Adapted from [28])

- **K-d tree**

  A binary search tree, also known as a K-Dimensional tree, defines nodes as a K-Dimensional point in space [29]. It is a data structure capable of organizing points in a K-Dimensional space.

This data goes through pre-processing methods, such as filtering and smoothing. Later, the path is calculated using a series of planes intersecting with the target's surface. The lines formed by the intersection of the planes are used to define the robot tool's path. The path is composed of a series of contact points which are then generated, and the posture of the tool is calculated for each point using the surface's normal vectors. This study gives several alternatives to the previous approach by not using Octree or K-d tree topologies instead of a CAD model, allowing for a fast pre-processing of the point cloud data.

The work in [30] presents an interactive programming method for using robots in the ceramic industry. This work aims to develop a robotic solution to finish ceramic tableware parts per the quality requirements. One of the requirements was to have fully customizable trajectories and not only dependent on the object's shape. This work uses a vacuum grip to pick up the ceramic piece. The process is divided into five stages: part pick-up, grind, fettle vertically, horizontally, and unload. Similar to this dissertation's focus, the product is picked up and released to the top of a conveyor belt. The robot's path for each part was achieved using a programming table and Augmented Reality (AR). AR is a technology in which virtual elements are added to a natural scene. The table is equipped with an infrared vision system that detects an LED pen and a projector for the augmented scene and allows the definition of trajectories by the user. The working process allows for replicating the current polishing of the piece in the user's application using AR. The final solution provides an application capable of highly flexible programming. One important aspect is its outperformance of offline programming in terms of programming time and system tuning.

The authors of [31] describe a technique used for polishing/fettling ceramic pot handles using a robot system with a stepper motor, ball screw, and force sensor. As opposed to the previous example of the work presented in [30], the polishing tool is placed at the end of the robotic arm and is not an external system. The force actuation is controlled with a fuzzy PID (Proportional–Integral–Derivative) controller instead of a regular PID to achieve a better response. The setup is shown in figure 2.11.

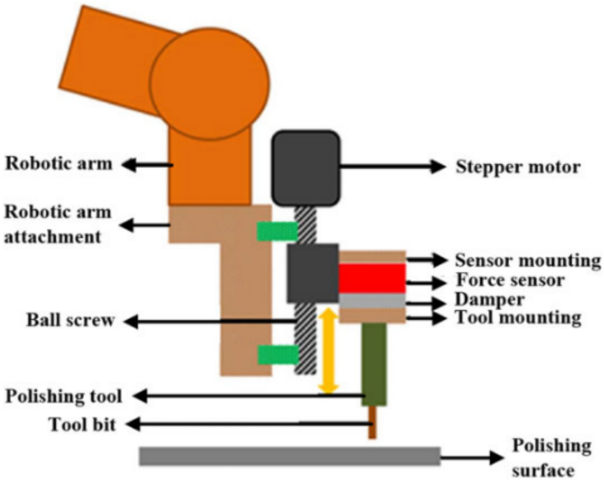

Figure 2.11: Pot handle polishing mechanism
(Source: Adapted from [31])

The polishing trajectories can be generated using CAD or computer-aided manufacturing data. The CAD files corresponding to each object are imported, and the polishing is controlled by a position and force control system. The main downside of this approach is the limitation to only objects with CAD files. Trajectory points are acquired with cameras which are used in different ways, such as extracting the features from a polished surface or recording the path of the tool in a manual process and replicating later on. The latter demands that the camera is fixed to register the tool's angles in the same plane.

This system was validated using a six-degrees-of-freedom Staubli TX 40L robotic arm. The polishing results were validated using a microscope to verify the molding defects of the ceramic pot, which achieved suitable and desired outcomes.

The work in [32] exposes a surface polishing solution with a force-controlled end effector and multi-step planning. For this application, the selected robot is an ABB IRB 1200 with a 7kg payload and 900 mm reach. The manipulator is equipped with a rotary gripper, and the polishing tool is stationary. Workpieces are picked and pushed against the abrasive tool. The force at which it is pushed against the polishing tool is determined by a force controller using a force sensor equipped on the manipulator. The final system was tested by polishing a watch bezel with curved surfaces resembling a ring's shape.

The force-controlled system is designed per the following requirements([32]):

- Fast and reliable
- Able to compensate against gravitational and polishing forces
- Low weight and compact
- Capable of supporting different tool weights

The overall system is described in figure 2.12

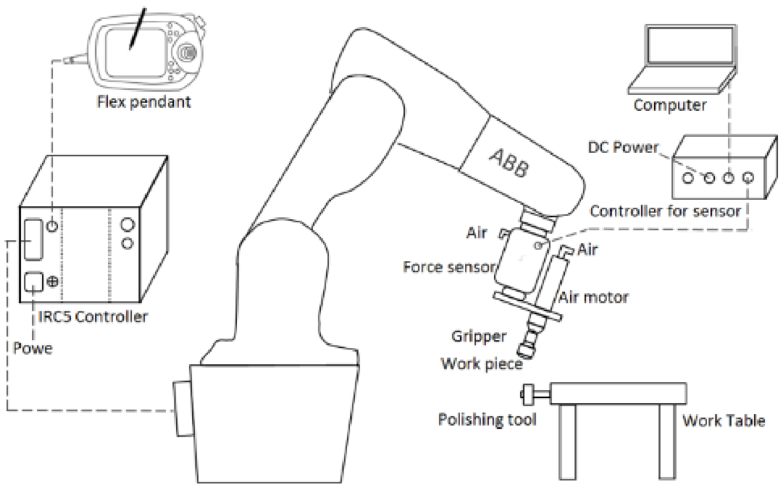

Figure 2.12: Robot polishing system schematic
(Source: Adapted from [32])

The end tools path is planned per the previously selected shape and size of the workpiece, not dynamically. Given the ring shape of the object, the authors chose to divide the process into two different steps: the interior ring and the outer surface. The final polishing process produced satisfactory results compared to the polishing performed by factory workers.

The work in [33] presents a solution for an automatic system capable of removing tableware from a commercial dishwasher. This work focuses on robotic manipulators in the food serving industry. The overall system concept, presented in figure 2.13 describes a 6 degrees of freedom robotic manipulator and a hand capable of grabbing and suction tasks. The tableware recognition, performed using a 3D camera, provides the system with the necessary data to plan the various functions.

The tableware used for testing this work was different-sized bowls with known diameters, weight, and build material, such as plastic or ceramic. Upon detecting each different bowl, the system can pick up and place the item on the dishwasher racks with two different orientations:

horizontal and vertical. This detection system, trained with sixty images using VGG Image Annotator, is capable of determining the robot's path and the final position of the item.

The piece's position is calculated using bounding boxes resulting from image analysis of each bowl. As such, the application can calculate the center of each bounding box and the bowl's location, considering the z coordinates obtained through the 3D camera for the final grasping position. The item's orientation was determined using the center points of each of the four sides of the bounding boxes. With the item's center and orientation, the robot can pick up the bowl and place it on the dishwasher rack. The final results proved the system successful. However, results also showed that significant light reflection could affect the accuracy of tableware recognition and is an essential factor to consider.

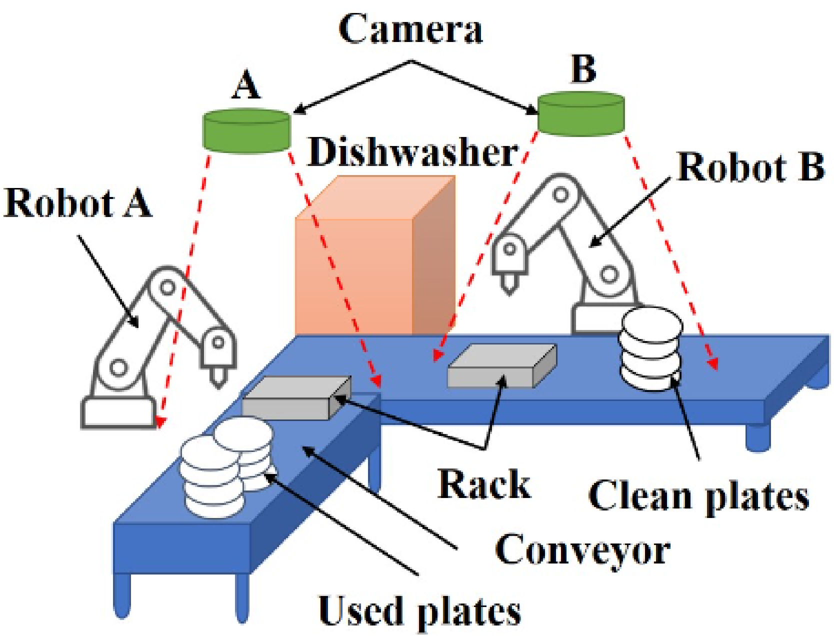

Figure 2.13: Robot polishing system schematic
(Source: Adapted from [33])

2.4 Robotic Simulator

Simulation is currently used to ensure the correct functioning of the user-designed applications for real-world hardware without the need for physical access to the robot.

The authors of [34] present an analysis of the reality gap between real-world manipulators and their corresponding simulation in virtual environments. A reality gap is formed by small or significant performance differences between the simulated robot and its physical counterpart. This behavioral deviation may lead to the underperformance of the user-developed application in real-world applications.

Advantages for the use of simulators include:

- No risk or damage to hardware
- Ability to conduct several simultaneous simulations
- Instant access to robots
- No human intervention is required, avoiding potential harm to the user

The simulation softwares used in this comparison were selected to provide the user with a standard programming language interface and access to the Robot Operating System (ROS). The final selection includes V-Rep, MuJoCo, and PyBullet. The simulators support the following physics engines: Bullet, ODE, Vortex, Newton, and Mujoco.

This work uses a Kinova Mico2 6DOF arm with an attached KG-3 gripper and a motion capture system to compare several simulators. All simulators use the same official Kinova URDF (Unified Robotics Description Format) file and the same usage scenarios:

- A simple joint motion
- Complex multi-movement task
- Simple interaction task for pushing a cubic object through a tabletop

The system compared the results of the simulations to a real-world application of these exact scenarios. The design captured results with a Qualisys motion capture system with 24 cameras. This setup allows accurate data regarding the robot's pose, position, and orientation as Euler angles. With this approach, the application can correctly capture a ground truth which serves as a term of comparison to the simulated results.

The final results conclude that the simulator with less accumulated euclidian error compared to the ground truth is PyBullet for testing one joint and the cube scenarios, and V-Rep with Newton engine for two joints motion.

The work exposed in [35] presents a systematic literature review of several simulators that can be applied in an educational context. This study compared and evaluated the capabilities of simulating a using realistic physics. Many simulators were considered, including Sim-Two, ROS (Robot Operating System); Gazebo; MATLAB/Simulink, V-REP, and Webots, among others. However, after analyzing the capabilities of these simulators and the physics engines they are built with, it was concluded that the ones that can be applied with ease to educational purposes are:

- **Gazebo**
  In [36] a method for using this simulator for robotic arms is exposed. This approach uses Parallel-Axis Theorem to estimate the moment of inertia of each of the robot's links and also opts for ROS to organize its tasks architecture. The simulator can analyze and compare the robot's response with different algorithms and target poses.

- **Webots**
  A study and analysis on how Webots can be used to simulate line following robots, a method of image preprocessing in-camera obtained images is presented in [37]. This approach uses different techniques such as dilation, erosion, and Gaussian filtering. Webots can simulate the robot's implementation, and software such as OpenCV and Python are used to design the line detection system and the robot's movement.

- **SimTwo**
  Sim-Two is exposed in [38] as a simulation environment capable of supporting a wide variety of robot types. In addition, it is also capable of simulating items such as conveyor belts.

- **V-REP**
  V-REP is introduced by [39] as a versatile and scalable simulation framework. The simulator can be used in an educational environment and industrial applications. However, as of 26/11/2019, V-REP was discontinued and replaced by CoppeliaSim [40]. This new simulator is 100% compatible with V-REP, evolves on its implemented features, and even adds new ones.

These options were selected in [35] due to continuous update support, support of multiple variations of robots, the ability to program a custom robot and use more than one physics engine, incorporated 3D vision of the robot/environment and finally, the capabilities of integrating third-party protocols.

However, a deeper analysis of the simulation requirements will need to be made to ensure that the chosen simulator meets said conditions. Such needs may include: preferred programming language, simulator precision, ability to mimic realistic environment, and overall software performance.

The comparison with real-world robots quantifies the accuracy of simulators.

Chapter 3

Prototype Description

3.1 Setup

The setup built using the previously mentioned components is shown in figure 3.1. The system was mounted at FEUP installations in the Department of Electrical and Computer Engineering and comprises of all the components except the robotic manipulator, which is currently placed at the Instituto Politécnico de Bragança's facilities.

Figure 3.1: Assembled setup

3.2 Components

3.2.1 2D Laser

Regarding image acquisition, a 2D laser (line profile sensor), designed for industrial applications, was chosen among several alternatives for this implementation and is identified as:

- Product: Gocator 2170 3D Laser Line Profile Sensor
- Model: 2170D-3R-R-01T
- Manufacturer: LMI TECHNOLOGIES INC.
- Laser Class: 3R
- Dimensions (mm): 49 x 75 x 272
- Overall Weight (kg): 1.3

Figure 3.2: Gocator 2170

3.2.2 Conveyor

Replicating the working conditions of an industrial assembly line can prove to be a difficult task in an educational environment. As such, a conveyor belt, shown in figure 3.3, was acquired and assembled for testing purposes. This component is an essential part of the prototype, given that it is responsible for the third dimension added to the 2D laser provided the movement of the belt. This product represents a smaller scale of the conveyors used in actual factory conditions and is identified as follows:

- Product: Electric Belt Conveyor Top-grade Conveyor 220v Powered Rubber Pvc Belt 59"x 7.8"
- Manufacturer: VEVOR
- Build: Stainless Steel
- Belt Dimensions (cm): 20 x 150
- Overall Weight (kg): 19

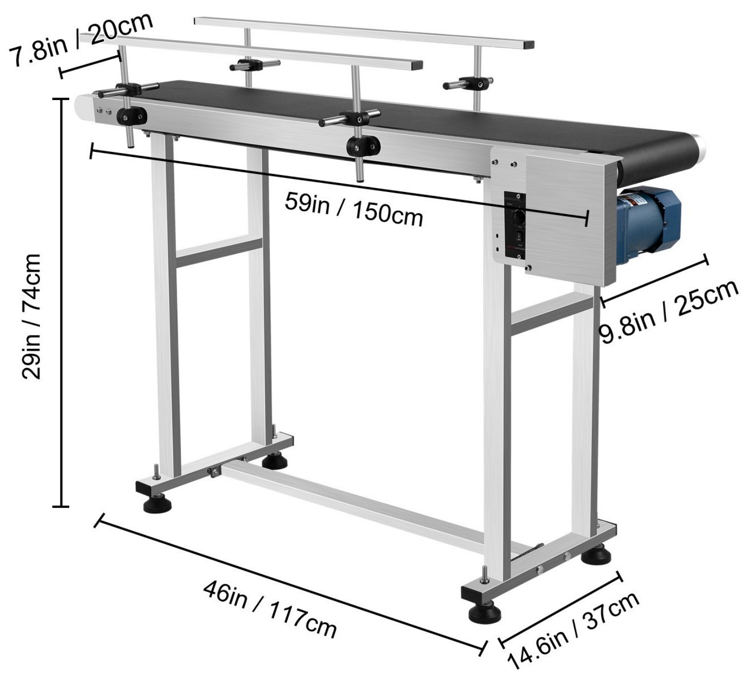

Figure 3.3: Conveyor (Source: Adapted from [41])

3.2.3 Magnetic Encoder

An absolute position encoder is used to determine the conveyor belt's movement speed in rotations per minute using the magnets position value and software timers.

- Product: Magnetic Rotary Encoder (14-Bit Angular Position Sensor)
- Model: AS5048A
- Manufacturer: ams
- Input Voltage (V): 5
- Communication standard : SPI (Serial Peripheral Interface)

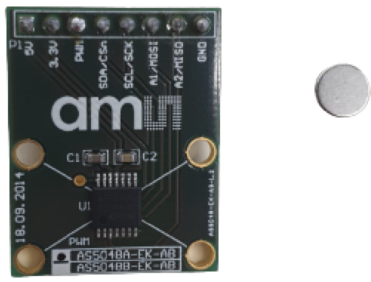

Figure 3.4: Magnetic Rotary Encoder (left) and Magnet (right)

3.2.4 M-DUINO PLC ARDUINO

A PLC (Programmable Logic Controller) was used to communicate with the magnetic rotary encoder. Using an Arduino board provided a more direct approach and implementation of the communications using SPI protocol.

- Product: M-DUINO PLC ARDUINO ETHERNET 38R I/Os ANALOG / DIGITAL PLUS
- Model: IS.MDUINO.38R+
- Manufacturer: INDUSTRIAL SHIELDS
- Input Voltage (V DC): 24 to 48

Figure 3.5: PLC
(Source: Adapted from [42])

3.2.5 Rotary Encoder

The incremental encoder is responsible for the belt's speed acquisition and can be connected to the 2D laser, allowing for the sensor's direct calibration. However, this item is only meant to be incorporated into the setup at a later stage.

- Product: Incremental AB 100ppr NPN 12-24Vcc Cable 0,5m
- Model: E6A2-CW5C 100P/R 0.5M
- Manufacturer: OMRON
- Input Voltage(V DC): +12 to +24
- Precision (pulse/rotation): 500

Figure 3.6: Rotary encoder
(Source: Adapted from [43])

3.2.6 Robotic Manipulator

At a later stage of this project, it is expected to incorporate the developed work with a real robotic manipulator capable of performing pick and place tasks. The acquired robot is shown in figure 3.7 and is identified as:

- Product: KR810
- Manufacturer: kassow robots
- Payload (kg): 10
- Degrees of freedom: 7
- Reach in all directions (mm): 850
- Joint speed (deg / sec): 225
- Overall Weight (kg): 23.5

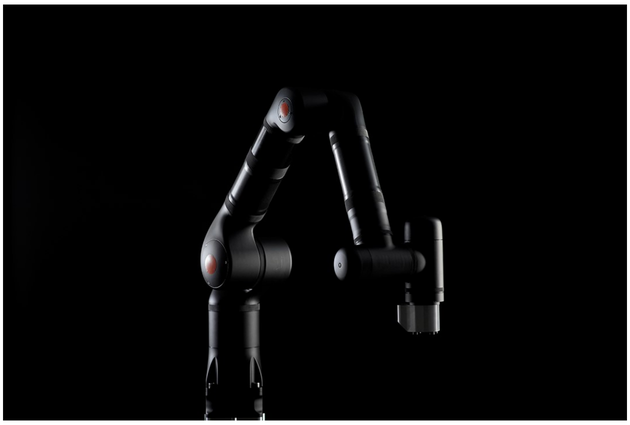

Figure 3.7: Robotic manipulator
(Source: Adapted from [44])

Chapter 4

Data Collection and Analysis

The main goal of this project is data acquisition and analysis. feature extraction is performed to support dynamic trajectory generation. The objective is to find the center value of the objects read through the laser scanner and determine the various distances to the center in function of the rotation angle and the height of its border values. This work involves communications with the laser sensor, its calibration, and developing different algorithms for each necessary task.

4.1 Experimental Setup

4.1.1 Laser calibration

The chosen laser is equipped with a web-supported interface through an Ethernet connection. This menu allows the user to perform several adjustments. Some of these adjustments include the selection of the active laser area. The dynamic laser area is the selection of data points received through the laser's camera. In some applications, only a part of the acquired region is useful to the system, as in this application. Given the conveyor's build, the laser is mounted slightly higher than its support, as presented in the profile of figure 4.1. The active area can be resized only to receive the data required for a particular application.

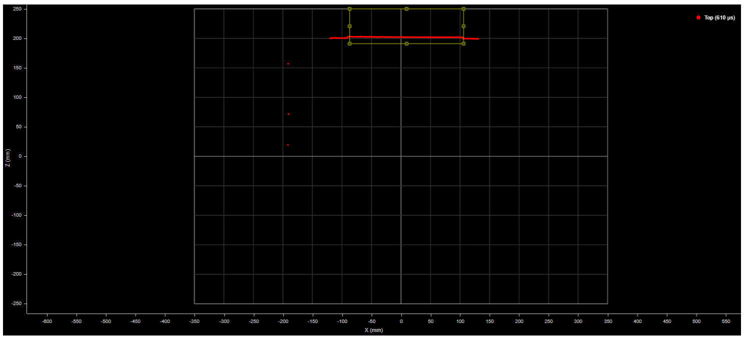

Figure 4.1: Active area selection

Given the conveyor belt's width, the final profile length is 59 mm. Another important aspect is that with the current setup, the height of the sensor allows for the read of objects with heights up to 50 mm. This value could be increased to 500 mm if the laser was placed in a higher position.

Following area selection, the profile points are visible through the interface and presented as shown in figure 4.2. An example of an object read is present in figure 4.3.

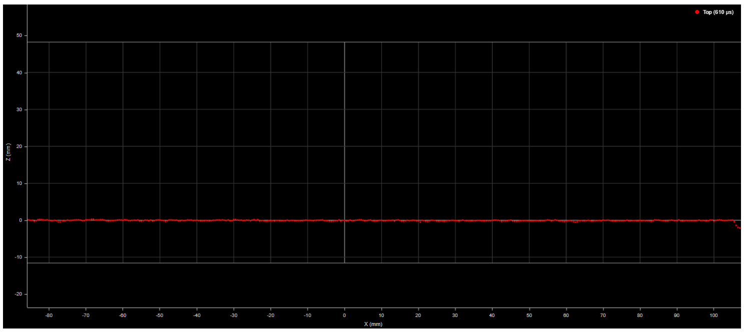

Figure 4.2: Base profile

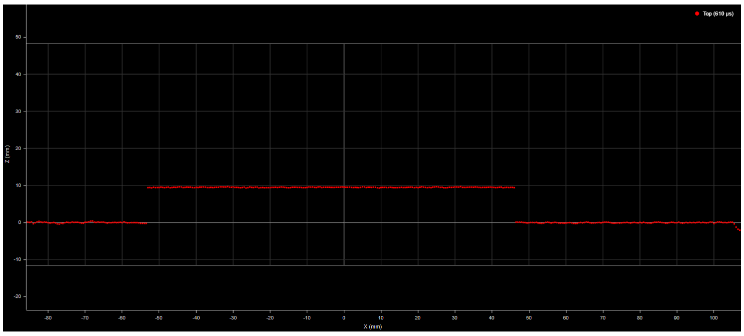

Figure 4.3: Object profile

The laser's capabilities were initially tested using the equipment's surface mode. This allows the user to instantly view the whole object's surface and shape on its built-in web interface as shown in figures 4.4 (two-dimensional view) and 4.5 (three-dimensional view).

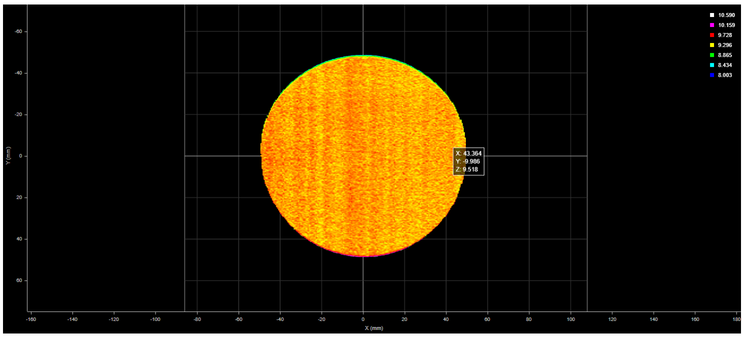

Figure 4.4: Object surface

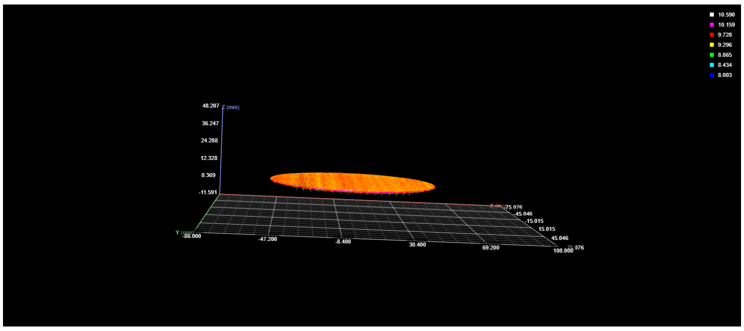

Figure 4.5: Object surface (3D view)

4.1.2 Connection

The laser device supports several forms of communication which which can be manually configured and selected using it's interface, as shown in figure 4.6:

- Ethernet
- Digital
- Analog
- Serial

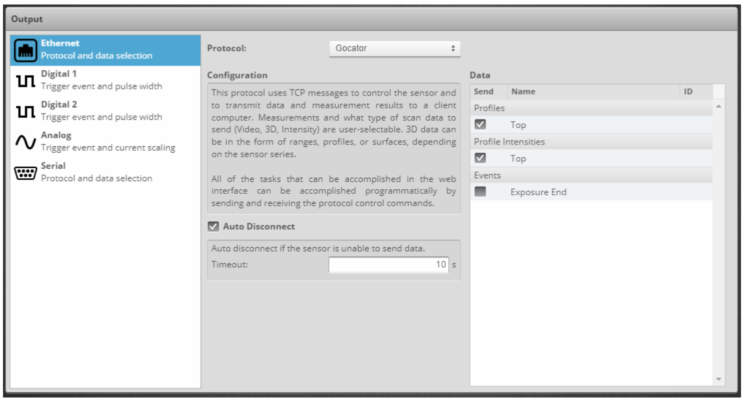

Figure 4.6: Laser output alternatives

One of the device's features is its custom IP address that the user can modify. During this project, only the default IP address 192.168.1.10 was considered.

During the development of this project, the Ethernet option was used due to an official Gocator GoSDK library provided by the manufacturer, LMI Technologies Inc. It can be used by an application written in C, C++, C#, or VB.NET. This library offers many functions for the user to communicate and retrieve information from the laser.

The developed application is programmed in C++ using the provided GoSDK C library with Microsoft Visual Studio.

4.1.3 Speed acquisition

As a first approach, a magnetic rotary encoder was used combined with a PLC. Magnetic encoders use a small diametrically magnetized (two-pole) standard magnet to provide the angular position. Using SPI communication protocol, an application was implemented capable of receiving the magnet's position in degrees and determining the amount turns of the conveyor wheel based on the last read value. The conveyor's belt speed in rotations per minute can be calculated using local program timers and a count of wheel turns in each time frame. The process's diagram of figure 4.7 presents the used implementation's algorithm.

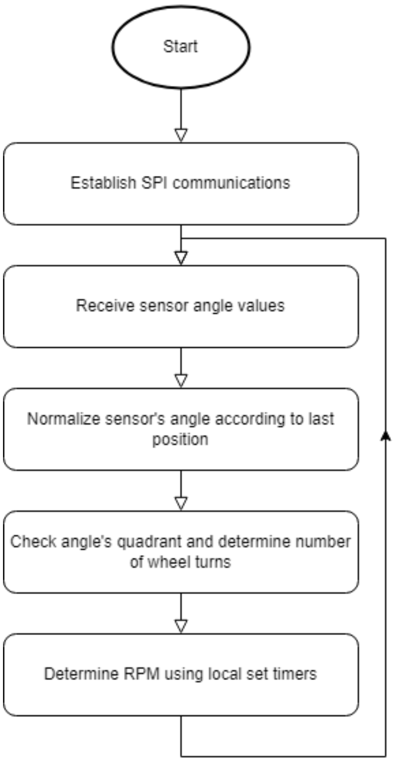

Figure 4.7: Magnetic encoder speed acquisition algorithm

After calculating the speed in rotations per minute, the conversion to mm/s can be performed using the radius of the conveyor wheel, and the value is sent to the laser sensor using the function GoTransform_SetSpeed(). However, this approach is not feasible because the value is sent to the laser and written in its Flash Memory. The problem is that the memory can only undergo a limited number of writes. As such, given that the primary use for this feature is the nearly constant update of the conveyor's travel speed to ensure the correct precision, we cannot use this algorithm in the final implementation and require an alternative.

A possible alternative is calibrating the travel speed using the calibration disk provided by LMI. The assembled conveyor and the mounted laser allows for suitable data acquisition. The user can manually control the conveyors' motor speed so the object can be transported through the laser beam at a constant rate. This feature allows for the correct calibration of the sensor using a calibration disk, shown in figure 4.8, or using the built-in capabilities of the encoder connection.

Figure 4.8: Calibration disk
(Radius/Width: 100 mm / 10mm)

4.1.4 Acquisition

By analyzing the features present in the laser, there are different possible operation modes:

- **Video**
  Outputs video images captured by the sensor.
- **Profile**
  Outputs profiles and profile measurements. These profiles are produced by processing video images captured by the sensor.
- **Surface**
  Outputs 3D point clouds and performs surface measurements.

Both the Profile and Surface modes can output a height map of the analyzed object. The height map is output through the laser with the object data read through the sensor. Each profile is read, and the whole surface is reconstructed according to the object's current moving speed.

The surface mode can generate a CSV file containing the object's data. On a first approach, it's possible to try and receive the complete object data set.

However, due to time constraints in obtaining and implementing the encoder in the setup, it was considered more suitable to individually receive the object's profiles and later reconstruct the entire object surface, thus leading to the use of Profile mode. This method provides a more feasible approach due to possible reading errors in the surface mode. It is possible to filter or discard incoming data by receiving each profile individually, and there is no need to rely on the CSV file generated by the sensor's software. Another reason for choosing Profile over Surface mode is the increase in the sensor's CPU usage. These high levels can cause delays in the sensor's communications system, specifically, drops in Ethernet output rates, and warning the user as shown in figure 4.9.

Figure 4.9: Ethernet drops warning

4.1.5 Calibration

The amount of valuable data sent output from the sensor, i.e., line profiles containing part of the object can vary with the item's length or the speed at which it is passing through the laser's light beam.

Given that the user can manually set the rotation speed of the conveyor, it was essential to fixate the belt's velocity. As such, the value of 30% of the conveyor's full speed was chosen for testing purposes.

The belt's speed is required to calibrate the laser sensor to use the surface mode correctly. The speed may be set by calibrating the laser using a calibration disk, a user-defined value, or the encoder ticks values. The user must also define the encoder's resolution to achieve a correct reading to use the encoder ticks. The user can choose this value with no particular constraint, and the laser can be calibrated for a speed of 150 mm/s as shown in figure 4.10.

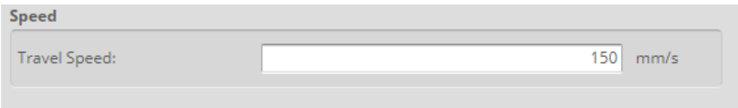

Figure 4.10: Gocator travel speed manual calibration

However, it is also important to calibrate the receiving end of the laser's output, i.e., the developed application. This operation can be performed by combining the value of local set timers with the laser's Ethernet output frequency. After several studies, three milliseconds were obtained as the approximated value between profiles reads and 1200 Hz for the laser's output to have a final correctly reconstructed surface. An important aspect is that while the sensor can work at a maximum frequency of 1600 Hz, a smaller value was chosen to prevent possible Ethernet output drops.

To correctly calibrate the receiving end of the application and ensure the overall consistency of the reconstructed surface, the Uniform spacing option of the laser is selected with a spacing between profile points of 0.5 mm. With this feature, for example, for a disk of a 10 cm diameter, the laser will transmit 200 relevant profile points, that is, object points where the height is larger than the previously defined threshold.

The developed application processes the received profiles, and the final result is a 2D depth matrix containing the x, y, and z values of each data point captured and filtered by the scanner, as shown in figure 4.11.

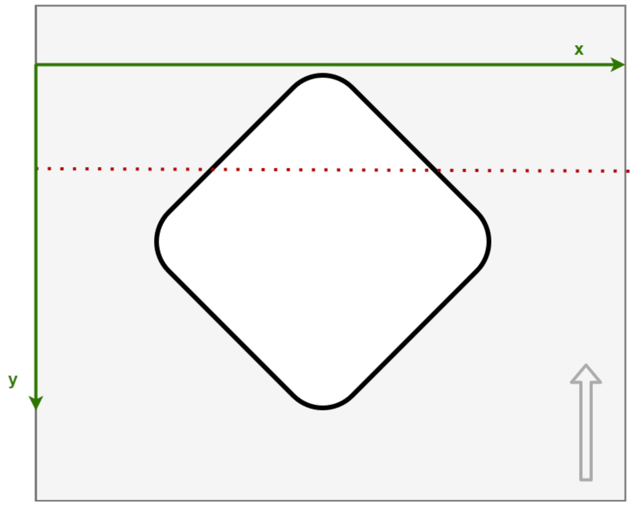

Figure 4.11: Example of object scan

Given that the profiles sent through the sensor contain the object and its surrounding, it is essential to isolate the piece's point cloud. This process is performed using a height threshold defined by the user regarding the minimum height of the ceramic pieces. In the studied scenarios, the minimum height of the items was 10 mm. As such, 9mm was chosen as the threshold height value for this application. The final result is a 2D depth matrix filled with zeros except for the points belonging to the object's surface, as shown in figure 4.12.

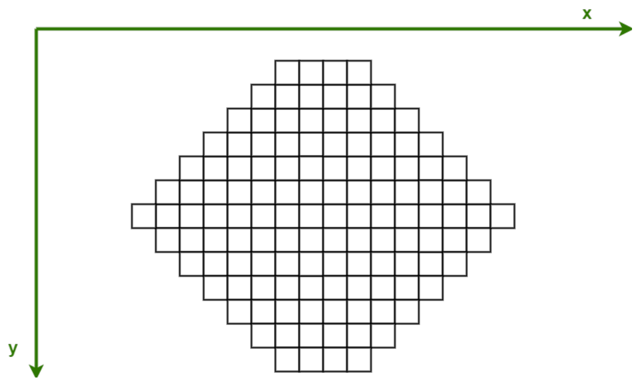

Figure 4.12: Example of filtered 2D matrix where the blank spaces represent the values of zero and the squares represent the object's surface points

4.2 Data Processing

4.2.1 Full Application Algorithm

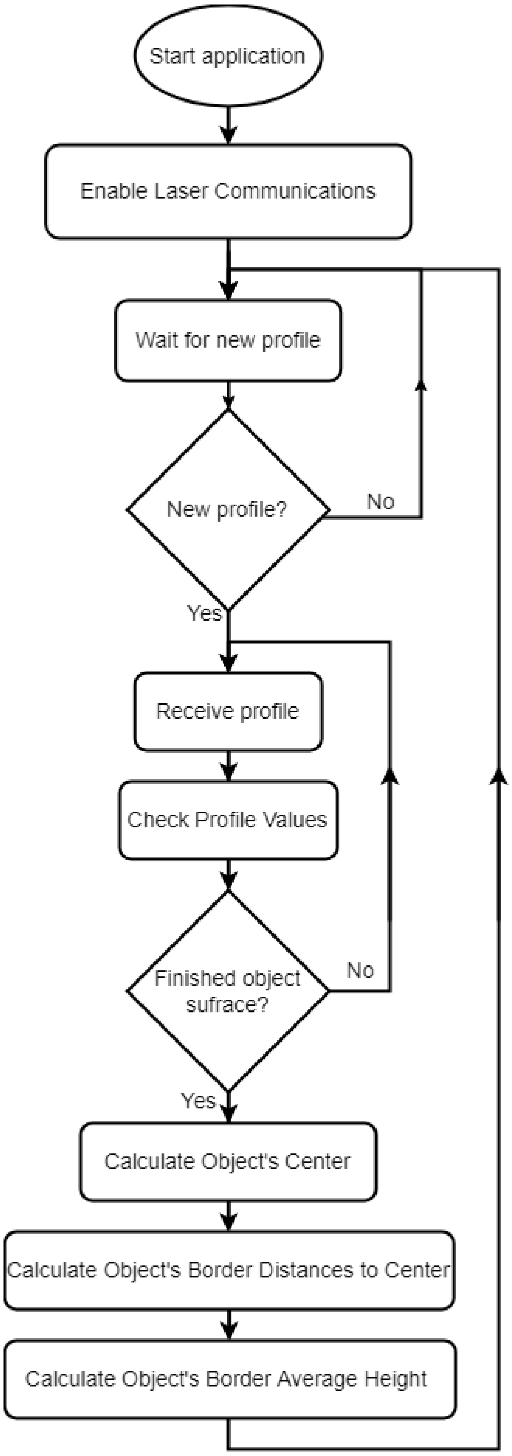

Figure 4.13: Full application algorithm

4.2.2 Library function

The steps and functions used to establish communications via Ethernet with the laser are described in figure 4.14

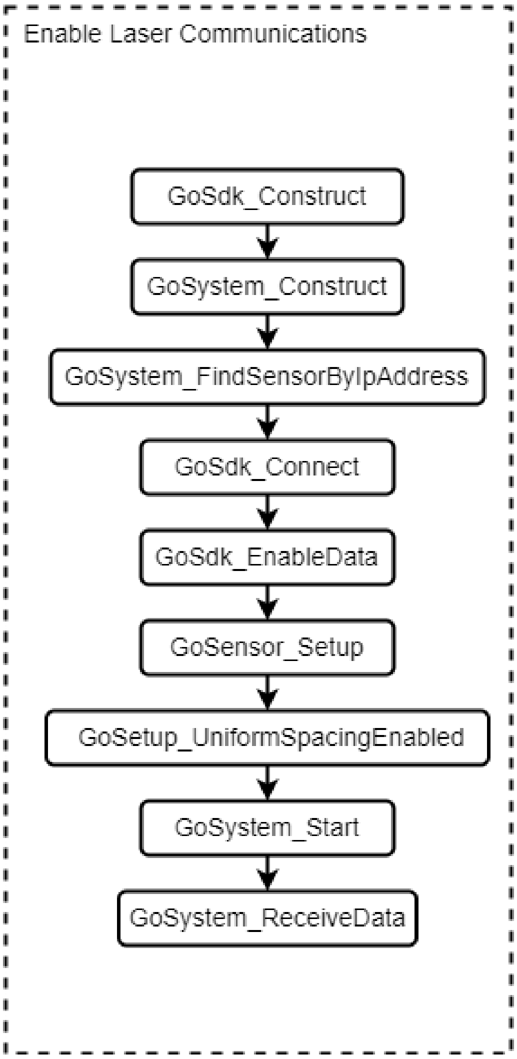

Figure 4.14: Library functions algorithm

The official Gocator GoSDK C language library provides the necessary functions for Ethernet communication. The list of used functions is presented in table 4.1.

Function	**Description**
GoSdk_Construct	Constructs the Gocator SDK library
GoSystem_Construct	Constructs a GoSystem object
GoSystem_FindSensorByIpAddress	Gets the sensor object with the specified IP address
GoSensor_Connect	Creates a connection to the sensor
GoSystem_EnableData	Establishes data connections to all connected sensors currently in the ready or running states.
GoSensor_Setup	Gets the GoSetup instance associated with the sensor.
GoSetup_UniformSpacingEnabled	Gets the user specified Uniform Spacing enabled state
GoSystem_Start	Starts all sensors that are currently in the ready state.
GoSystem_ReceiveData	Receives a set of sensor data messages.
GoDestroy	Frees the memory associated with a given kObject sourced class handle.

Table 4.1: Used GDK available functions and description

4.2.3 Developed algorithms

Firstly, the functions presented in section 4.2.2 were used to establish communications with the laser. Secondly, it's necessary to analyze and process the acquired data for several objectives, including determining the object's center, distance to its boundaries, and the height of its border, which proved more complex due to the possible varying value of the border distance to its center point. The value is ideally constant only for circular shapes. However, the application is meant to calculate several distances in rotational order even if the object's shape is non-circular.

4.2.4 Center Point

4.2.4.1 Algorithm

One of the objectives of this project is to determine the value of the center point of the ceramic piece. After acquiring the point cloud as a 2D depth matrix, the data model can be analyzed to obtain the required values.

A height threshold is used to remove profile outliers resulting in a point cloud filled with zeros except for the scan of the object above the height threshold, thus isolating the object in the model.

The usual method for finding the center of mass of an object uses the weighted values of each point of the point cloud model, which in this case is the object's height at each captured point. For a global scenario, the center of mass of an object can be calculated as shown in equation 4.1.

$$X = \sum_i \frac{x_i z_i}{M} \tag{4.1}$$

For this approach the test pieces are considered uniform. The 3D scan is projected onto a 2D plane, and each point's height is regarded as a unitary value for the calculations. The total projection of the object allows for the correct identification of the center point while not being interfered with by the possibly different heights of each point. As such, the following equations are used for the final calculations:

$$X = \sum_i \frac{x_i}{M} \tag{4.2}$$

$$Y = \sum_i \frac{y_i}{M} \tag{4.3}$$

where:

- x_i: x coordinate of the point i
- y_i: y coordinate of the point i
- z_i: z coordinate of the point i
- X: center value in the x-axis
- Y: center value in the y-axis
- M: number of points in the point cloud where the height is above the predefined threshold, i.e., the number of points where the z value is different than zero

The overall process is describe in figure 4.15.

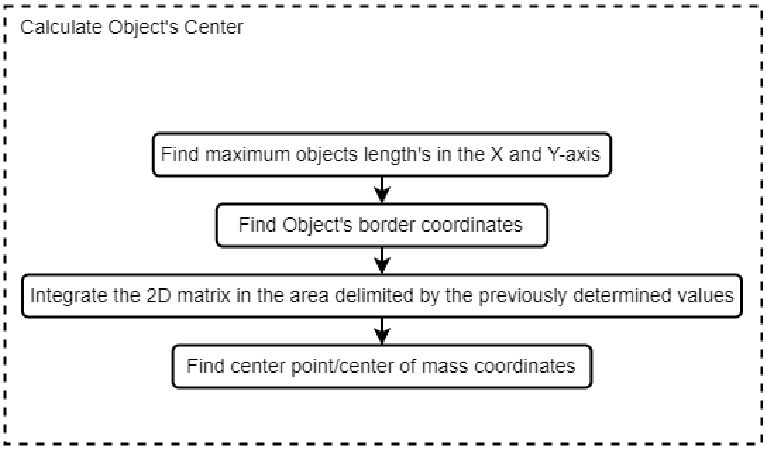

Figure 4.15: Center point algorithm

4.2.4.2 Results

To have a better understanding of the influence of the object's shape with the proposed algorithms, different shaped objects were tested. These were designed to test the algorithms' accuracy and precision in multiple situations and object orientation. Figure 4.18 represents a circular item in which the distance of the border to the center is constant. Opposite to the circular item, figures 4.16 and 4.17 represent test items with varying distances of the border to the center point.

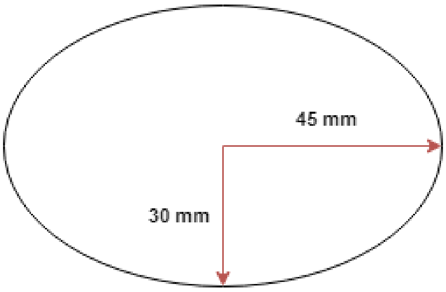

Figure 4.16: Oval

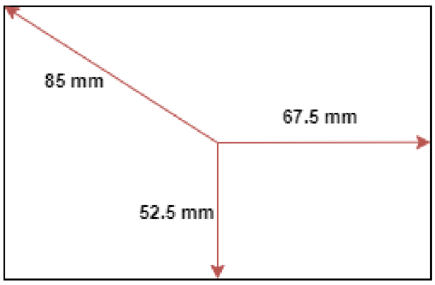

Figure 4.17: Rectangular

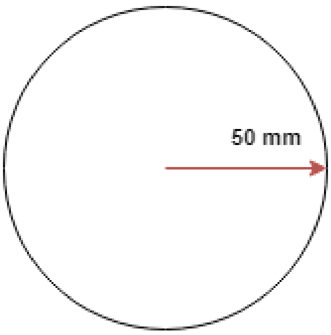

Figure 4.18: Circular

In order to achieve a reliable average result, a number of fifty experiments were performed for the read of each object shape and consequent calculations. Each experiment consisted in passing the item through the laser beam at a constant speed of approximately 150 mm/s.

The average errors for each of the X and Y-axis and each shape are presented in table 4.2.

Shape	**Average Error of X coordinate (mm)**	**Average Error of Y coordinate (mm)**	**Maximum Error (mm)**
Circular	0.34	0.78	**0.78**
Oval	0	1.02	**1.02**
Rectangular	0.108	0.781	**0.781**

Table 4.2: Maximum average error for center localization

An analysis of table 4.2 shows a higher level of uncertainty in the values captured in the Y-axis. The most significant cause of this problem is the correct calibration of the profile reads. As locally set timers and time intervals are currently being used for accounting for new profile reads and have no precise way to control the conveyor's speed except for its manual dial, higher error values may occur. One possible solution, which has not yet been implemented, is incorporating the rotary encoder, which provides a higher precision of the traveled distance of an object in the conveyor belt. This value allows for the precise Y-axis mapping and achieves lesser error values.

4.2.5 Distance to Center

4.2.5.1 Algorithm

An additional requirement is to determine the distance of the outer points to its center point. These values are used to assist in the rotation of the item against the polishing tool. The distance can be constant in the case of circular pieces but may also vary, for example, in a square-shaped plate.

The process starts by analyzing the 2D depth matrix and finding the object's border points of the top and bottom rows and the limits on the right and left sides. This sequential analysis of the model provides an array of points. However, some edges of the model may be subject to two or three analyses and are repeatedly considered as border values, as shown in figure 4.19. As such, the array is filtered to delete duplicate values.

A crucial aspect of these values is the necessity for them to be in the order of rotation to obtain the correct rotation of the ceramic piece. The array is ordered in a clockwise method as shown in the steps of figure 4.20:

- **Step 1**
  Analyze the first row and store the distance values from the lowest to the highest values of x.

- **Step 2**
  Find the greatest value of x, line by line, until reaching the bottom row and store the distance values in order.

- **Step 3**
  Analyze the bottom row and store the distance values from the greatest to the lowest values of x.

- **Step 4**
  Find the lowest value of x, line by line, until reaching the top row and store the distance values in order.

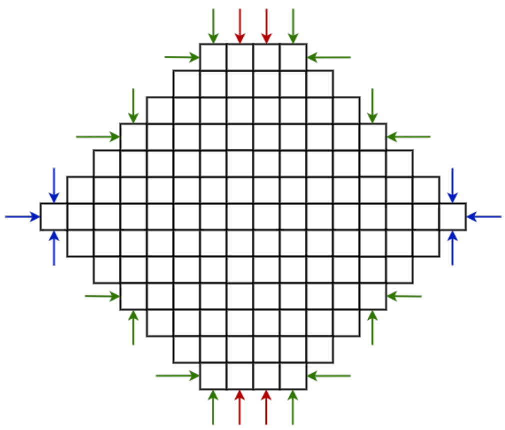

Figure 4.19: Border points acquisition

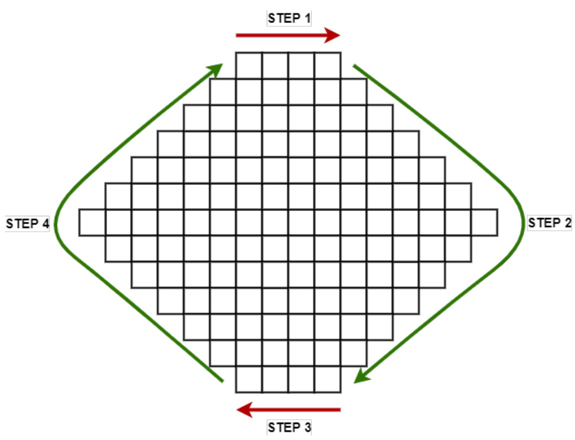

Figure 4.20: Steps to order values of distances to center

The overall process is describe in figure 4.21.

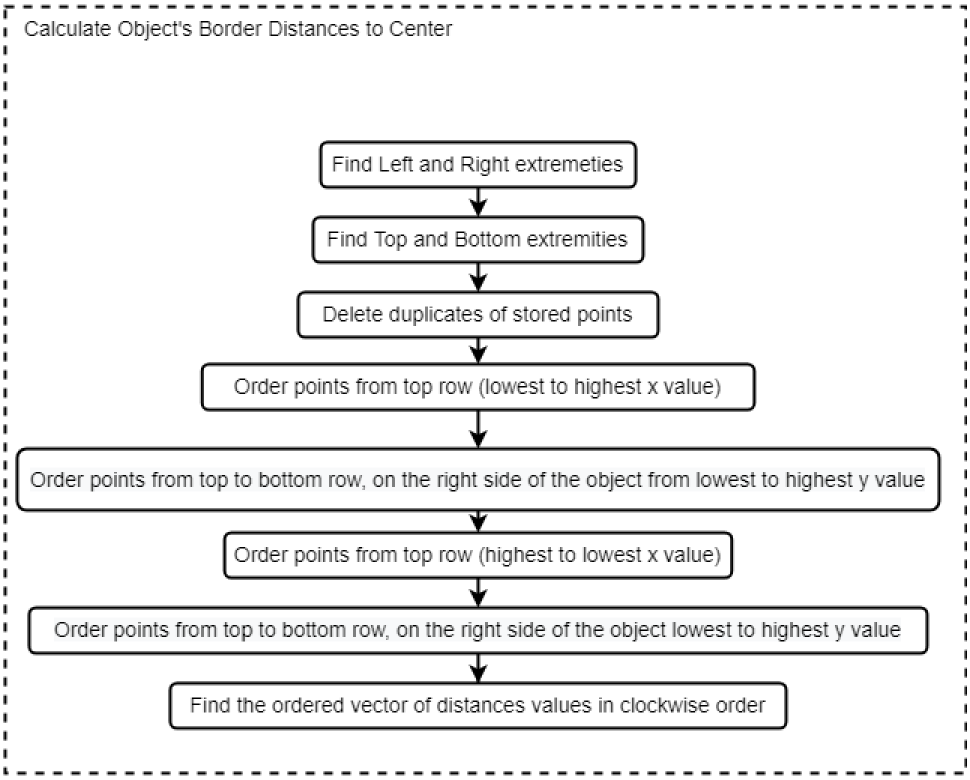

Figure 4.21: Center point algorithm

4.2.5.2 Results

Similar to the process in section 4.2.4.2, a set of different shaped pieces were considered for testing, and a dataset of 50 experiments was acquired for each different shape.

To test the algorithm proposed in section 4.2.5.1, it was considered adequate to evaluate the application's performance regarding the maximum and minimum expected values, which correspond to the top and bottom distances scheduled from the center of each object to its outer rim. For example,regarding the oval shape of figure 4.16, the minimum and maximum distances to the center are ideally 30 mm and 45 mm, respectively. Each top, bottom, and corresponding error is presented in table 4.3.

Shape	**Top ideal measurement**	**Bottom ideal measurement (mm)**	**Top value error (mm)**	**Bottom value error (mm)**	**Max error (mm)**
Circular	50	50	2.241	2.241	**2.241**
Oval	45	30	1.401	1.675	**1.675**
Rectangular	85	52.5	1.774	1.599	**1.774**

Table 4.3: Maximum Average Error for Distance to Center Calculation

The results show a maximum error of 2.241 mm, which is considered as a relatively low value due to the dimensions of the pieces. However, as mentioned in section 4.2.4.2, these results can be improved if provided the necessary implementation of the rotary encoder.

4.2.6 Border Height

4.2.6.1 Algorithm

The third requirement for this application is to determine the height of the object's outer borders to allow for the correct positioning of the robotic manipulator when next to the polishing tool.

The height value of each captured point is stored in the application. As such, considering the only necessary tasks to perform are to average the already filtered array of values obtained values, explained in section 4.2.5.1, and transmit the values to the robot for placement of the end tool. The overall process is described in figure 4.22.

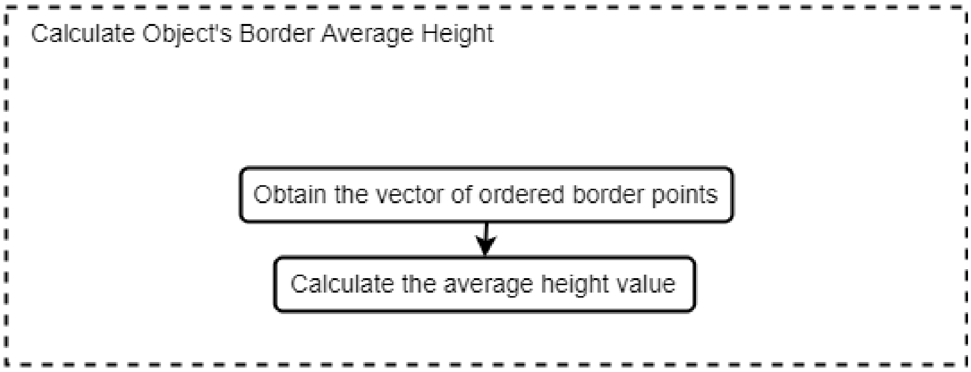

Figure 4.22: Border height algorithm

4.2.6.2 Results

Similar to the process in section 4.2.4.2, a set of different shaped pieces were considered for testing, and a dataset of 50 experiments was acquired for each different shape.

The heights expected and measured are presented in table 4.4, as well as the error of the values.

Shape	**Height (mm)**	**Average Measured Height (mm)**	**Maximum Error (mm)**
Circular	94.4	93.361	**1.039**
Oval	19	19.448	**0.48**
Rectangular	17	17.263	**0.263**

Table 4.4: Maximum Average error for height determination

Similar to the other algorithms, the proposed approach presents a relatively small error value, reaching maximum errors of approximately 1 mm, which is a very positive factor for the development of this work.

4.3 Trajectory Generation

4.3.1 Environment conditions

For initial testing, all the components were considered static, and that there were no unknown obstacles between the conveyor and the polishing tool. The result is a more direct approach to the trajectory of the manipulator. A general view of the proposed setup is presented in figure 4.23

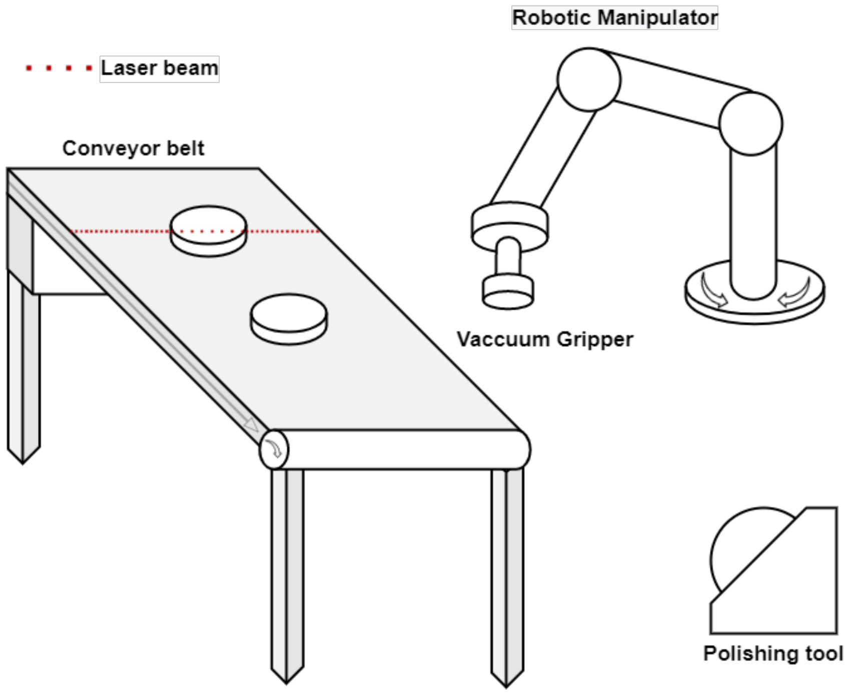

Figure 4.23: Full setup

4.3.2 Strategies

The main goal of this project is the data acquisition and trajectory generation of the end-tool of a robotic manipulator capable of moving a ceramic piece close to a polishing station. However, a more classic approach was used for the initial testing of the acquired data, considering the polishing bit at the tip of the end tool instead of a vacuum gripper.

For initial testing, the robot was tested with a predefined trajectory using its STEP FILE[45] option, which allows the program to import trajectories from CAD software and run it in the robot environment. However, the trajectory points can not be modified in the CAD approach. The alternative was to calculate and directly send the points to the robot using Modbus and a developed Lazarus application.

4.3.3 Robotic manipulator

The communication protocol used for controlling the robot is Modbus, a standard data communication protocol used in industrial automation, establishing a common language for information exchange between electronic devices. In this application, the protocol is used through an Ethernet cable and works in two different ways:

- **Modbus Serial**
  Based on a master/slave architecture. The master usually starts communication, and the answers are sent by the slave with the requested information or acknowledging a specific task.
  Modbus serial can work with RTU mode, with addresses represented in binary format or ASCII mode, where they are represented in ASCII format. The latter, however, consumes more resources from the network.

- **Modbus TCP**
  Based on a server/client architecture. The communication is started by the client requesting data from the server. In this application, the robot acts as the client and the computer as the server.
  Data is encapsulated in binary format using TCP frames and sent through an Ethernet cable.

4.3.4 Polishing tool

The polishing tool to be used with this prototype is currently being developed in IPB [46]. The finishing device is equipped with a sponge, DC motor, stepper motor, several measurement sensors, and its 3D printed outer casing. The assembled tool is shown in figure 4.24. The overall system is controlled with an Arduino UNO.

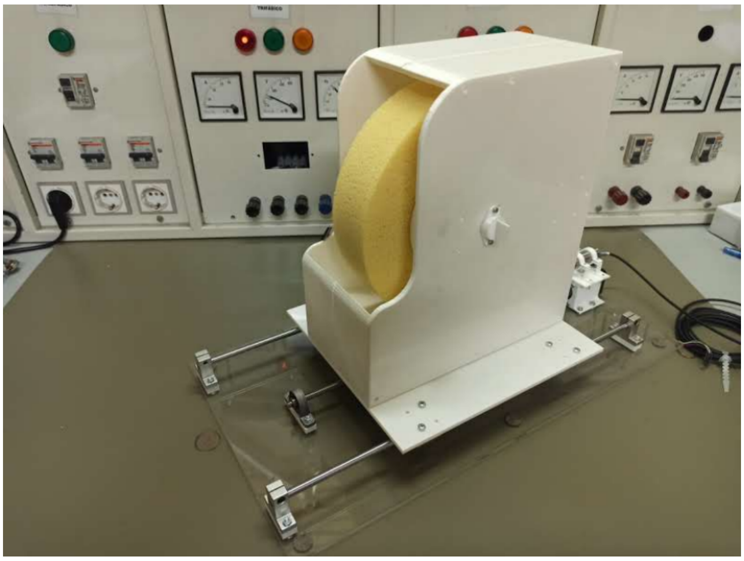

Figure 4.24: Full application algorithm

Chapter 5

Conclusion and Future work

Image acquisition can be accomplished in multiple ways. Each analyzed alternative has certain advantages, namely precision and accuracy. In this work, using a 2D laser scanner combined with a conveyor provided the necessary precision for the required performance of the system. It was considered a good approach for the required detail levels for this application. The results prove a minimal error regarding its expected values, which is a positive factor in favor of this implementation.

Future work in this subject will likely include the simulation of the robot's behavior with the output coordinates from the developed algorithm, with a simulation environment yet to be determined. This work is an important module of a larger project in collaboration with Instituto Politécnico de Bragança. The other modules are currently being studied and perfected, including the polishing tool with a force response system and the control and actuation of the robotic manipulator.

The next step in the project will include combining the algorithms presented in this dissertation with the other modules. Initial testing will only be performed with the robotic manipulator to ensure the correct localization of the ceramic's central point. Only then will the polishing module be incorporated with the information about the robot's path. However, tests are currently being performed using only the other two modules without the dynamic acquisition of data from the laser scanner.

As such, combining the whole system will likely prove to be a faster process. Future work may also include detecting surface defects and imperfections by comparing the acquired point cloud with that of a CAD model of the ceramic plates. However, this feature is subject to discussion to determine the necessary details and requirements.

References

[1] *STC 4.0 HP (New Generation of Stoneware Tableware in Ceramic 4.0 by High Pressure Casting Robot work cell) project*, Jan. 2021. [Online]. Available: https://cedri.ipb.pt/communication/news/stc-4.0-hp-%5C%28new-generation-of-stoneware-tableware-in-ceramic-4.0-by-high-pressure-casting-robot-work-cell%5C%29-project.

[2] Q. Wang and M.-K. Kim, "Applications of 3d point cloud data in the construction industry: A fifteen-year review from 2004 to 2018," *Advanced Engineering Informatics*, vol. 39, pp. 306–319, 2019, ISSN: 1474-0346. DOI: https://doi.org/10.1016/j.aei.2019.02.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034618304683.

[3] D. D. Nguyen, J. P. Ko, and J. W. Jeon, "Determination of 3d object pose in point cloud with cad model," in *2015 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, 2015, pp. 1–6. DOI: 10.1109/FCV.2015.7103725.

[4] P. Glueck, *All you need to know about 3d point clouds*, Mar. 2021. [Online]. Available: https://blog.hexagongeosystems.com/all-you-need-to-know-about-3d-point-clouds/.

[5] A. Bennett, *Point Cloud and 3D Image*, Mar. 2020. [Online]. Available: https://www.revopoint3d.com/point-cloud-and-3d-image/.

[6] Q. Wang, Y. Tan, and Z. Mei, "Computational methods of acquisition and processing of 3d point cloud data for construction applications," *Archives of Computational Methods in Engineering*, vol. 27, no. 2, pp. 479–499, 2020, ISSN: 1886-1784. DOI: 10.1007/s11831-019-09320-4. [Online]. Available: https://doi.org/10.1007/s11831-019-09320-4.

[7] P. Trebuňa, M. Mizerák, and L. Rosocha, "3d scaning – technology and reconstruction," *Acta Simulatio*, vol. 4, pp. 1–6, Jun. 2018. DOI: 10.22306/asim.v4i3.44.

[8] ASPRS, "Guidelines for procurement of professional aerial imagery, photogrammetry, lidar and related remote sensor-based geospatial mapping services," p. 4, [Online]. Available: http://www.asprs.org/wp-content/uploads/2010/12/ProcurementGuidelines_SubmittalforBoardApproval_081409.pdf.

[9] A. Gruen, "Fundamentals of videogrammetry — a review," *Human Movement Science*, vol. 16, no. 2, pp. 155–187, 1997, 3-D Analysis of Human Movement - II, ISSN: 0167-9457. DOI: https://doi.org/10.1016/S0167-9457(96)00048-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167945796000486.

[10] T. Debroutelle, A. Chetouani, S. Treuillet, M. Exbrayat, and S. Jesset, "Classification of friezes engraved on ceramic sherds from 3d scans," IEEE. DOI: 10.1109/isspit.2016.7886038. [Online]. Available: https://dx.doi.org/10.1109/isspit.2016.7886038.

[11] K. Morita, M. Hashimoto, and K. Takahashi, "Point-cloud mapping and merging using mobile laser scanner," IEEE. DOI: 10.1109/irc.2019.00078. [Online]. Available: https://dx.doi.org/10.1109/irc.2019.00078.

[12] Z. Wang, C. Yang, Z. Ju, Z. Li, and C.-Y. Su, "Preprocessing and transmission for 3d point cloud data," in *Intelligent Robotics and Applications*, Y. Huang, H. Wu, H. Liu, and Z. Yin, Eds., Springer International Publishing, pp. 438–449, ISBN: 978-3-319-65289-4.

[13] Z. Hocenski, T. Matic, and I. Vidovic, "Technology transfer of computer vision defect detection to ceramic tiles industry," IEEE. DOI: 10.1109/sst.2016.7765678. [Online]. Available: https://dx.doi.org/10.1109/sst.2016.7765678.

[14] A. Birlutiu, A. Burlacu, M. Kadar, and D. Onita, "Defect detection in porcelain industry based on deep learning techniques," IEEE. DOI: 10.1109/synasc.2017.00049. [Online]. Available: https://dx.doi.org/10.1109/synasc.2017.00049.

[15] Y. Tang, Q. Wang, H. Wang, J. Li, and Y. Ke, "A novel 3d laser scanning defect detection and measurement approach for automated fibre placement," *Measurement Science and Technology*, vol. 32, no. 7, p. 075 201, 2021, ISSN: 0957-0233 1361-6501. DOI: 10.1088/1361-6501/abda95. [Online]. Available: http://dx.doi.org/10.1088/1361-6501/abda95.

[16] H. Zeng, N. Huang, Y. Yang, and J. Li, "Object optimal grasping recognition method based on local point cloud model," IEEE. DOI: 10.1109/robio54168.2021.9739259. [Online]. Available: https://dx.doi.org/10.1109/robio54168.2021.9739259.

[17] J. Hänninen, "Using gocator laser profiling sensor with halcon machine vision software."

[18] J. Yang, M. Kong, and R. Wang, "A path planning algorithm of spray robot based on 3d point cloud," IEEE. DOI: 10.1109/cisce50729.2020.00078. [Online]. Available: https://dx.doi.org/10.1109/cisce50729.2020.00078.

[19] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path planning and trajectory planning algorithms: A general overview," in *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*, G. Carbone and F. Gomez-Bravo, Eds. Cham: Springer International Publishing, 2015, pp. 3–27, ISBN: 978-3-319-14705-5. DOI:

10.1007/978-3-319-14705-5_1. [Online]. Available: https://doi.org/10.1007/978-3-319-14705-5_1.

[20] B. Singh, N. Sellappan, and P. Kumaradhas, "Evolution of industrial robots and their applications."

[21] U. Berger, R. Lepratti, and M. May, "An approach for the automatic generation of robot paths from cad-data," in *2005 IEEE Conference on Emerging Technologies and Factory Automation*, vol. 1, 2005, 7 pp.–297. DOI: 10.1109/ETFA.2005.1612537.

[22] A. K. Bedaka, J. Vidal, and C.-Y. Lin, "Automatic robot path integration using three-dimensional vision and offline programming," *The International Journal of Advanced Manufacturing Technology*, vol. 102, no. 5, pp. 1935–1950, 2019, ISSN: 1433-3015. DOI: 10.1007/s00170-018-03282-w. [Online]. Available: https://doi.org/10.1007/s00170-018-03282-w.

[23] ——, "Automatic robot path integration using three-dimensional vision and offline programming," *The International Journal of Advanced Manufacturing Technology*, vol. 102, no. 5-8, pp. 1935–1950, 2019, ISSN: 0268-3768. DOI: 10.1007/s00170-018-03282-w. [Online]. Available: https://dx.doi.org/10.1007/s00170-018-03282-w.

[24] A. K. Bedaka and C.-Y. Lin, "Cad-based robot path planning and simulation using open cascade," *Procedia Computer Science*, vol. 133, pp. 779–785, 2018, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.07.119. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187705091831072X.

[25] G. Zhang, J. Wang, F. Cao, Y. Li, and X. Chen, "3d curvature grinding path planning based on point cloud data," in *2016 12th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pp. 1–6. DOI: 10.1109/MESA.2016.7587150.

[26] *Octree*. [Online]. Available: http://www.open3d.org/docs/latest/tutorial/geometry/octree.html.

[27] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013, ISSN: 1573-7527. DOI: 10.1007/s10514-012-9321-0. [Online]. Available: https://doi.org/10.1007/s10514-012-9321-0.

[28] A.-V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto, "Octree-based region growing for point cloud segmentation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 104, pp. 88–100, 2015, ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2015.01.011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271615000283.

[29] *K Dimensional Tree | Set 1 (Search and Insert)*. [Online]. Available: https://www.geeksforgeeks.org/k-dimensional-tree/.

[30] G. Veiga, P. Malaca, and R. Cancela, "Interactive industrial robot programming for the ceramic industry," *International Journal of Advanced Robotic Systems*, vol. 10, no. 10, p. 354, 2013, ISSN: 1729-8814. DOI: 10.5772/56753. [Online]. Available: https://dx.doi.org/10.5772/56753.

[31] Z. Yu and H. I. Lin, "Development of robotic polishing/fettling system on ceramic pots," *International Journal of Advanced Robotic Systems*, vol. 18, no. 3, p. 17 298 814 211 012 851, 2021, doi: 10.1177/17298814211012851, ISSN: 1729-8806. DOI: 10.1177/17298814211012851. [Online]. Available: https://doi.org/10.1177/17298814211012851.

[32] I. Mohsin, K. He, J. Cai, H. Chen, and R. Du, "Robotic polishing with force controlled end effector and multi-step path planning," IEEE. DOI: 10.1109/icinfa.2017.8078931. [Online]. Available: https://dx.doi.org/10.1109/icinfa.2017.8078931.

[33] Y. Fukuzawa, Z. Wang, Y. Mori, and S. Kawamura, "A robotic system capable of recognition, grasping, and suction for dishwashing automation," IEEE. DOI: 10.1109/m2vip49856.2021.9665169. [Online]. Available: https://dx.doi.org/10.1109/m2vip49856.2021.9665169.

[34] J. Collins, D. Howard, and J. Leitner, "Quantifying the reality gap in robotic manipulation tasks," IEEE. DOI: 10.1109/icra.2019.8793591. [Online]. Available: https://dx.doi.org/10.1109/icra.2019.8793591.

[35] C. Camargo, J. Gonçalves, M. Á. Conde, F. J. Rodríguez-Sedano, P. Costa, and F. J. García-Peñalvo, "Systematic literature review of realistic simulators applied in educational robotics context," *Sensors*, vol. 21, no. 12, 2021, ISSN: 1424-8220. DOI: 10.3390/s21124031.

[36] Z. Huang, F. Li, and L. Xu, "Modeling and simulation of 6 dof robotic arm based on gazebo," in *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, pp. 319–323, ISBN: 2251-2446. DOI: 10.1109/ICCAR49639.2020.9107989.

[37] A. Ma'arif, A. A. Nuryono, and Iswanto, "Vision-based line following robot in webots," in *2020 FORTEI-International Conference on Electrical Engineering (FORTEI-ICEE)*, pp. 24–28. DOI: 10.1109/FORTEI-ICEE50915.2020.9249943.

[38] P. Costa, J. Gonçalves, J. Lima, and P. Malheiros, "Simtwo realistic simulator: A tool for the development and validation of robot software," *Theory and Applications of Mathematics & Computer Science*, vol. 1, pp. 17–33, 2011.

[39] E. Rohmer, S. P. N. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1321–1326, ISBN: 2153-0866. DOI: 10.1109/IROS.2013.6696520.

[40] *CoppeliaSim from creators of V-REP*, Feb. 2022. [Online]. Available: https://coppeliarobotics.com/.

[41] *Electric Belt Conveyor Top-grade Conveyor 220v Powered Rubber Pvc Belt 59"x 7.8"*, Accessed: 2022-02-18. [Online]. Available: https://eur.vevor.com/belt-conveyor-c_10439/electric-belt-conveyor-top-grade-conveyor-220v-powered-rubber-pvc-belt-59-x-7-8--p_010464727863.

[42] *IS.MDUINO.38R+*. [Online]. Available: https://pt.farnell.com/industrial-shields/is-mduino-38r/ethernet-38r-i-o-analog-digital/dp/3515744.

[43] *E6A2-CW5C 100P/R 0.5M*. [Online]. Available: https://industrial.omron.pt/pt/products/E6A2-CW5C-100P-R-0-5M.

[44] *KR810*. [Online]. Available: https://www.kassowrobots.com/products/kr810/.

[45] *Kassow Robots: Software Manual*, Version: 1.2.1 EN01, Kassow Robots ApS, Oliefabriksvej 57 2770 Kastrup Denmark, p. 116.

[46] M. Alvarez, L. Brancalião, D. Gomes, *et al.*, "Prototyping and control of an automatic ceramic tableware finishing device," Springer, 2022.