UNIVERSITÀ
DI PAVIA

University of Pavia
DEPARTMENT OF ELECTRICAL, COMPUTER AND BIOMEDICAL ENGINEERING

PhD thesis in Electronics, Computer Science and Electrical engineering

---

Trajectory planning of collaborative robotic contact-based applications

PhD Candidate
Ahmed Magdy Ahmed Zaki

Supervisor:
Hermes Giberti

Co-supervisor:
Marco Carnevale

Academic year 2021/2022
PhD Cycle: XXXV

To my family...

Abstract

In the last years, due to the huge and fast development of the automation in general and the field of robotics in particular and the scarcity of the skilled technical manpower, the number of companies seeking to automate the steps of their production chains is continuously increasing. Different operations still struggling to be fully automated. Especially contact-based operations such as surface finishing, polishing, welding and material depositions still from the operations that are done manually by skilled operators. This is due to the high quality required that may be kept consistent, in case of manual execution of the process, if the human operator have enough experience that allows the adjustment of the huge number of the process parameters that may influence the process output. Several solutions have been introduced to automate contact-based operations based on the use of robotic manipulators. Despite that, the proposed solutions still lack characteristics like the ease of use and the flexibility to deal with variations of the process parameters. In this thesis, robotic solutions are proposed to automate the execution of robotic contact-based operations. Particularly, by developing solutions able to automatically generate the working trajectory to be followed by the robot end-effector during the operation execution. The proposed solutions are fully based on the data acquired by a low-cost 3D vision system and do not require previous knowledge of the workpiece, in the form of CAD model for example. In this way, inaccuracies or uncertainties about the workpiece form or positioning can not affect the performance of the robotic application. Low-cost 3D cameras used in this thesis have relatively higher errors estimating the depth values when compared to professional laser scanners. The error of the 3D camera used is less than 2% of the distance between it and the observed object, and for the laser scanner the error is in the range of few tenths of a millimeter. A feasibility study of the possibility to apply low-cost 3D cameras in applications that usually require high accuracy trajectory is done. The performance evaluation of the developed algorithm is shown, focusing on important aspects that may affect the performance of a contact-based robotic application when applied in industrial environment. The considered aspects are the computational time, accuracy of the generated trajectory and the contact force behavior that has to be exerted over the surface of the workpiece to guarantee a good performance of the task. Different robotic cell layouts are used based on the task required. Two main cases, changing the 3D camera position, are considered. In the first layout, a fixed position for the 3D camera is considered. While in the second, the 3D camera is attached to the robot end-effector to be able to scan the workpiece from all the sides. In the case of a moving camera, an algorithm for 3D model reconstruction is developed. That 3D model is what is elaborated to generate the working trajectory.

An adjustment procedure for positional error compensation and online adjustment of the trajectory during the execution of the contact-based task based on the use of force control is proposed. In the two cases, fixed camera and the moving one, a good contact force behavior, in terms of mean value, is obtained. Geometric errors, due to the depth estimation error of the used low-cost 3d camera, of the generated trajectory when compared to the ideal trajectory are compensated. In the case in which the 3D camera is attached to the robot end-effector, the accuracy of the proposed 3D reconstruction algorithms is based on the evaluation of the computational effort and the accuracy of the reconstructed 3D model with respect to the exact 3D CAD model of the workpiece. The 3D model is reconstructed in few seconds with an error lower than one millimeter with respect to the CAD model.

Sommario

Negli ultimi anni la crescita del settore della robotica e più in generale dell'automazione, insieme alla mancanza di manodopera tecnica qualificata, ha portato un crescente numero di aziende ad automatizzare i processi produttivi. Nel settore industriale vi sono alcune operazioni che presentano difficoltà ad essere interamente automatizzate. In particolare, le operazioni a contatto come la levigatura, la lucidatura, la saldatura e la deposizione di materiale vengono principalmente eseguite manualmente da operatori esperti. Ciò è dovuto alla necessità di mantenere costante l'elevata qualità di queste lavorazioni; infatti in caso di lavorazione manuale, è l l'operatore presenta sufficiente esperienza tale da consentire la regolazione dell'enorme numero di parametri di processo che possono influenzare del processo. Sono state introdotte diverse soluzioni per automatizzare le operazioni a contatto basate sull'uso di manipolatori robotici. Nonostante ciò, le soluzioni proposte mancano ancora di caratteristiche come la facilità d'uso e la flessibilità per far fronte alle variazioni dei parametri specifici del processo. In questa tesi vengono proposte soluzioni robotiche per automatizzare l'esecuzione di operazioni robotiche contact-based. In particolare, sviluppando soluzioni in grado di generare automaticamente la traiettoria di lavoro che l'utensile attaccato all'end-effector del robot deve seguire durante l'esecuzione dell'operazione. Le soluzioni proposte si basano interamente sui dati acquisiti da un sistema di visione 3D a basso costo e non richiedono una conoscenza preliminare del pezzo da considerare, ad esempio sotto forma di modello CAD. In questo modo, imprecisioni o incertezze sulla forma o sul posizionamento del pezzo non possono influire sulle prestazioni dell'applicazione robotica. Le fotocamere 3D a basso costo utilizzate in questa tesi presentano errori maggiori nella stima dei valori di profondità rispetto agli scanner laser professionali. L'errore della telecamera 3D utilizzata è inferiore al 2% della distanza tra essa e l'oggetto osservate, invece per lo scanner laser l'errore è dell'ordine di pochi decimi di millimetro. Viene condotto uno studio di fattibilità della possibilità di applicare telecamere 3D a basso costo in applicazioni che normalmente richiedono traiettorie di elevata precisione. Viene mostrata la valutazione delle prestazioni dell'algoritmo sviluppato, concentrandosi su aspetti importanti che possono influenzare le prestazioni di un'applicazione robotica contact-based in un contesto industriale. Gli aspetti considerati sono: il tempo di calcolo, l'accuratezza della traiettoria generata e il comportamento della forza di contatto che deve essere esercitata sulla superficie del pezzo per garantire una buona esecuzione del processo. Vengono utilizzati diversi layout di celle robotiche in base all'attività richiesta. Vengono inoltre presi in considerazione due casi principali, in cui viene modificata la posizione della telecamera 3D. Nel primo layout viene considerata una posizione fissa per la telecamera 3D. Mentre nel secondo, la telecamera

3D è collegata all'end-effector del robot per poter scansionare il pezzo da tutti i lati. Nel caso di una telecamera in movimento, viene sviluppato un algoritmo per la ricostruzione del modello 3D. Quel modello 3D è ciò che viene elaborato per generare la traiettoria di lavoro. Viene poi proposta una procedura di adeguamento per la compensazione dell'errore di posizione e la regolazione in linea della traiettoria durante l'esecuzione dell'attività di contatto basata sull'uso del controllo della forza. Nei due casi, quello con telecamera fissa e quello con telecamera mobile, si ottiene un buon comportamento della forza di contatto, in termini di valore medio. Gli errori geometrici dovuti all'errore di stima della profondità della fotocamera 3D a basso costo utilizzata, e gli errori della traiettoria generata rispetto alla traiettoria ideale vengono compensati. Nel caso in cui la telecamera 3D sia collegata all'end-effector del robot, l'accuratezza degli algoritmi di ricostruzione 3D proposti si basa sulla valutazione dello sforzo computazionale e sull'accuratezza del modello 3D ricostruito rispetto all'esatto modello CAD 3D del pezzo. Il modello 3D viene infine ricostruito in pochi secondi con un errore inferiore al millimetro rispetto al modello CAD.

Contents

List of Figures

List of Tables

Introduction

Since the beginning of the sixties, when the first industrial robots started to be installed in industrial plants, the manufacturing processes have changed considerably. With the development of industrial robots, it was possible to automate repetitive, heavy and not ergonomic tasks that are harmful for the human operators. Industrial robots are capable to execute highly precise tasks at high speeds. They also require to be programmed by programming experts that have to design accurately the task. To do so, they still need to be preprogrammed using offline simulations before applying the robotic system in real application. In this case, industrial robots are not capable to evaluate their performance while executing the task or to be able to adapt some task parameters to solve some performance drawbacks. Industrial robot and classical industrial robotic cells, are optimized for mass production and could help to increase production volumes and reduce waste and have many limitations when they have to work in a dynamic and unstructured work environments.
In recent years, with the mass adoption of the trends of Industry 4.0 and the recent Industry 5.0. More industrial processes became human centered with the focus of allowing the safe interaction between the robot and human operator in a way that exploits the human input to optimize the performance of the robotic cell and to execute tasks that would be not possible to fully automate without that human input. Also, a mandatory characteristic to satisfy the requirements of this new trend is the need of technologies that allow the task planning and programming in an easy and flexible way that can be done by operators that are not programming experts. Collaborative robots (Cobots) were introduced to answer to the new challenges, overcome some of the limitations of industrial robots, to be able to automate production in dynamic production environments and to interact safely with human operators.
The production in small and medium enterprises (SME) has been moving, in recent years, toward small volumes and high level of customization. In these production systems, many manufacturing steps are still carried out manually by experienced operators, mainly due to the lack of flexible, easy to use and fast to configure robotic tools, able to adapt to highly variable environments.
In small and medium enterprises, collaborative robotic cells provide production systems flexibility and allow safe interaction with human operators. In some applications, traditional trajectory planning techniques limit the capabilities of the cobots. The enhancement of cobot flexibility, simplification of tasks programming and the minimization of cobot setup time are from the main topics discussed in industry and in academia. In industry, cobot manufacturers introduced new simplified programming languages (e.g. block based or visual programming languages). In academia,

new paradigms such as task based programming and learning from demonstrations techniques were introduced for cobot programming.
Despite the introduction of the above-mentioned solutions, the design of complex tasks such as contact based robotic operations is still lacking flexibility and sometimes also accuracy. In contact-based robotic applications, trajectory planning requires a high level of accuracy, since the robot end-effector has to exert a proper contact force while following the desired trajectory and the work cells in these work environments are very dynamic, and the operator interacts often with the process.
Contact-based robotic applications in manufacturing systems are highly dependent on the knowledge of an exact digital model of the workpieces the robot tool must come into contact with. The trajectory generated using this digital model is usually dependent on the previous knowledge of the exact fixing position of the object in the working space that it is referred to the robot reference frame. The CAD models of the product "as planned" are usually available, particularly small and medium-sized enterprises (SMEs) often lack tools and procedures to capture digital models of the products "as manufactured". In addition, if product ranges shows high variabilities, this gap in the digital coverage of robotic manufacturing processes is often even more impacting. Other way to generate the working trajectory, that is more flexible but less accurate, is based on manual teaching procedure. It consists of dragging the robot while touching the workpiece to a subset of poses combining the working trajectory that a professional operator would follow by the tool center point. At each of these poses, the tool center point pose is saved to be reached together with the others to form the working trajectory.
In order to exploit collaborative robot to carry out contact based operations in such an environment, 3D vision sensors combined with force and torque sensors can be a feasible tool to generate the working trajectories. They allow developing flexible, fast and easy to configure tools that allow to reconstruct the digital model of unknown objects, to be then used to generate the trajectories needed for the robotic application.
The proposed solution in this thesis is aimed to introduce flexible tools that allow the generation of ad hoc trajectory for specific workpieces in a fast way without the need for advanced programming skills. These solutions would accelerate the cycle time needed for contact-based task planning and execution, that would as a consequences increase productivity. The proposed solution would have a direct impact on the production tetrahedron parameter improvement. The effect on these four parameters is summarized in the following way.

- Production cost is minimized since engineering effort for designing and drawing an exact CAD model of the product at each production step is not necessary anymore.
- Production quality is increased because the trajectory is generated based on the real as-built state of the workpiece and is adapted to compensate possible errors. In this way, a consistent overtime execution of task is achieved.
- Production time is minimized because there are no constrains of the fixing position of the workpiece. It is enough to place it in the robot reachable area. This would decrease considerably the setup time of the robotic cell.

- Production flexibility is optimized because of the ease of configuration of that can be done by operators also with or without programming experience. workpieces with huge variability of shape and geometrical features can be considered to apply the developed solutions.

In this thesis, 3D vision sensors and force sensors will be exploited to generate and execute the working trajectory to be followed by the robotic tool in a contact based operation in a flexible way without the need of previous knowledge of the workpiece or its digital model (e.g. CAD model).

Thesis structure

- Chapter 1: State of the art of the related work will be presented, focusing on trajectory planning techniques (traditional and more flexible solutions). Also, computer vision-based techniques for trajectory planning will be presented.
- Chapter 2: the hardware and software tools necessary to develop the proposed solutions will be introduced
- Chapter 3: an overview of the methods and essential technical background on which the developed solutions are based are introduced
- Chapter 4: CAD-based technique is used for trajectory planning together with feasibility study of using low-cost 3D vision system in trajectory planning in contact-based applications.
- Chapter 5: 3D reconstruction algorithm based on the color and depth elaboration and integration is proposed. The algorithm, named Odometry-based is based on the use of computer vision techniques to be able to integrate the images.
- Chapter 6: 3D reconstruction algorithm based on the use of robot poses while capturing the dataset of color and depth images is introduced. The algorithm, named robot poses-based, is based on the use of the known camera pose to be able to integrate the images content.
- Chapter 7: the conclusions are commented

Chapter 1

State of the art

Manufacturing industrial plants have been changing in recent years due to several factors such as the change of the market demand towards highly customized products instead of mass production and the introduction of new technologies with high performance and low prices.
The need for more customized products implies the necessity for machines, specifically robotic manipulators, that are able to adapt and execute the required production tasks within a short configuration time and applying the minimum possible engineering effort to design and program the task.
The introduction of collaborative robots, allowed to enhance the safety level of robotic manipulators and the flexibility of the robotic cells. This because of the possibility given to the human operator to share the working area with the robot. Cobots, beside of being safe, they are characterized by the ease of use and the possibility to be programmed without the need for high level of experience in programming. Cobots usually have native programming languages and graphical user interfaces that are simplified and intended to be used to design the task with the minimum engineering effort.
To enhance the flexibility of the robots, 3D vision sensors are used to monitor the surrounding environment of the robot. In recent years, with the rapid development of low-cost hardware with a high computational capacity and the introduction of optimized software, different low cost and highly performing 3D vision sensors have been introduced.
Thanks to the above-mentioned factors, new industrial scenarios and new opportunities to develop more advanced and at the same time flexible robotic solutions have been and are being introduced continuously. That is also the case for accurate robotic contact-based operations that require the generation of a precise path to be followed by the robot end-effector during the task execution.
In this chapter, the state of the art of the mentioned aspects and new technologies that are used for trajectory generation for contact-based operations will be analyzed. Also, some proposed techniques that exploiting these tools are also analyzed. In the end, limitation and new solutions of the current state of the art of the commented elements will be shown.

1.1 Introduction about trajectory planning

Trajectory planning and path planning are from the main research topics in the automation field in general and the robotic field in particular. When a robotic manipulator is used to automate a step in the production chain, the objective is to execute the task in the minimum time to decrease the total cycle time and increase the production volume. Pushing the machines or the robots to work at high speeds has to be done in a harmless way for the hardware components. The smoothness of the generated trajectory is an important aspect to keep in mind while generating a trajectory. This is done, for example, having trajectories with high degree of continuity. Particularly, limiting the acceleration discontinuity at the joints allows limiting the jerk to avoid excessive vibrations of the mechanical structure that may cause the wear of the robot (i.e. damage in the actuators). In specific tasks, rapid moves may degrade the accuracy in the geometric path following, decreasing the quality of the task execution [1].
The trajectory planning consists in the definition of the path to be followed in terms of starting, ending point and the way points. The generation of that set of points can be done in the operation space defining the several poses of the robot end-effector, or in the joint space defining the angles of each joint to make the robot end-effector reach the desired poses. Together with the geometric path to be followed, it is necessary to express a set of values for velocity and acceleration while considering the kinematic and dynamic constraints of the robot [2].
The geometric path is usually done in the operation space, defining a time sequence of the end-effector poses values. This because operation space is more intuitive for the users, and it is the same as the task space. Since the control signal of the robot controller is done in the joint space, the inverse kinematics inversion is necessary. A trajectory planning in the joint space can avoid the movement near to singularity, but has the disadvantage to be unpredictable in the operation space due to the non-linearity caused by the direct kinematics [3].
To execute a specific task using a robotic manipulator, it is necessary to generate an offline sequence of movements to be followed by the manipulator in a known environment.There are several ways to follow. Usually an optimal motion planning is used to find the trajectory optimizing a certain criterion [4]. The trajectory planning algorithms are named based on the function to be optimized. In literature, the most used algorithms are minimum execution time, minimum energy and minimum jerk. Besides some algorithms optimizing a single function, there are those optimizing more than one (e.g. time-energy optimal trajectory [5] and time-jerk [6]).
Minimum time optimal trajectory is a well discussed topic in literature due to the direct effect on production increase. This technique optimize the time but with some limitation of generating a smooth trajectory. Minimum energy trajectory consists of finding the trajectory minimizing the integral of the squared torque that measures the effort of the actuators or can be found in literature some methods minimizing the energy as the physical quantity measured in Joules. Since the torque is related to the current of an electric motor, the two terms to be optimized are correlated [1]. The third category of optimal trajectory generation is the minimum jerk method. The goal of this technique is to generate a smooth trajectory. This

is done by guaranteeing the continuity of the actuators' acceleration, that is done if the jerk has bounded values. Limiting the values of the jerk is an indirect way to have a continuous function and bound the variation rate of the joint torques [6]. This method has many benefits such as minimizing stress minimization to the manipulator structure and actuator and improve of the trajectory tracking error.
The concept of generating optimal criteria that optimize particular factors can be extended to optimize other functions related to the task that is executed by the manipulator. Such as optimizing material uniform distribution that is important for spray-painting. In [7], an optimal trajectory is found to maximize the spray quality. The cost function used is based on the spray thickness. In [8], an optimal painting trajectory is obtained, minimizing the paint thickness deviation with respect to ideal desired thickness value. Also, optimization can be done having a cost function based on some parameters important for maintaining a good quality for contact-based operations (e.g. polishing [9] and surface cleaning). These parameters are the coverage of all points of the workpiece surface and speed feed rate to have good quality for the machined object.

1.1.1 Trajectory programming

One of the advantages of industrial robots is the programmability or the capability to accept instructions that alter the state of a device or in particular the robot. The robot programming has been for long time the main topic of many research works due to the huge impact on simplifying the configuration of the robot and the decrease of downtime of the production chain. The difficulty of programming is one of the main problems to that avoiding the expansion of the use of industrial robots in small and medium enterprises.
Robot programming can be divided in online programming, offline programming and programming using Augmented Reality [10]. Online programming consists of moving the robot end-effector, manually or using the teach pendant, to the poses to be reached at every stage of the task execution. The end-effector poses together with the relevant robot configurations are recorded, and the program is developed to execute sequentially the movement commands to reach the recorded poses. The other way to execute online programming consists of using sensors to drive the robot movement to reach the poses necessary to execute the desired task. This technique is intuitive and does not require high programming skills. It presents the drawback of limited performance, since it the program quality is highly dependent on the operator skills. This technique is mostly used to execute tasks that require simple movements and considering workpieces with simple forms.
Offline programming tools (OLP) have gained huge interest to design and program robotic cells. In offline programming, the 3d model of the overall robotic cell can be added to generate a digital twin that replicate exactly the real robotic cell. In this way, it is possible to study the feasibility of using particular hardware devices and to optimize the robotic-cell footprint. Since the generated digital twin is identical to the real one, task programming can be done using only offline simulations without the need for the real robotic cell. This programming technique is cost-efficient for large production volumes.

More specific explanation of the available tools and techniques for online and offline programming will be discussed in the next section.

1.1.2 Software tools for trajectory planning and execution

Trajectory programming, online and offline, and execution processes are done using several software tools. There is a vast variety of not standardized software tools and simulators that are used in industry and in research work. Some tools are used to create the digital twin of the robotic cell and are mainly used for the cell layout design and study of the feasibility of using a certain hardware device. These tools are useful for feasibility study and hardware selection. They are also flexible to develop complex motion planning techniques.
Other tools, besides the creation of the robotic cell digital twin are used to program the sequence of commands that the robot has to execute to reach the operation desired goal. These tools are programmed in the robot native programming languages and are optimized to communicate the trajectory following script with the robot controller. Examples of these software tools are introduced as follows.

Robot Manufacturer simulators

Robot manufacturer simulators are the most used software tools to simulate the robot movements before applying them to the real robot. These simulators allow choosing a certain robot model from a library of the robot made by that manufacturer. The robot models include an exact digital twin of the robot, including the same kinematic and dynamic models. It is possible to add and simulate other hardware components such as grippers and cameras that are possible to integrate in the robot programming language and controller. These software simulators have the drawback of being limited to consider only the robotic manipulators made by the manufacturer. Usually, for every software a specific text based programming languages has to be used that make it is difficult to design modular robotic solutions that are not dependent on the robot model.
These software tools allow the programming of the robot and realtime visualization of robot state change and movements during the task execution. The digital twin of the robotic cell can be created, adding the CAD model of other static components such as working benches and objects handled during the task execution. In this way, collisions between the robot and the other components in the cell and the feasibility of using the robotic cell to execute the desired task can be evaluated.
In figure 1.1, manufacturer simulator to control Mitsubishi robots is shown. The figure shows a robotic cell for pick and place applications. The robotic cell is made by a Mitsubishi scara robot with a vacuum gripper attached to it.
In figure 1.2, the script for controlling the pick and place application shown in figure 1.1 is shown. The script is written in the programming language MELFA-BASIC V that is used to control Mitsubishi robots.

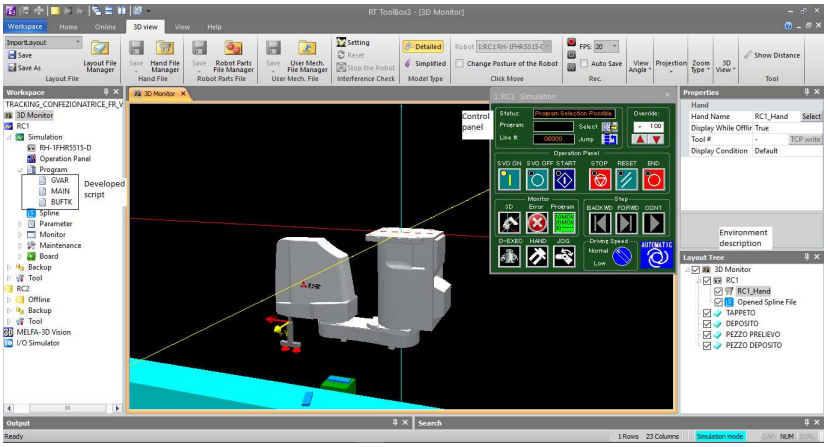
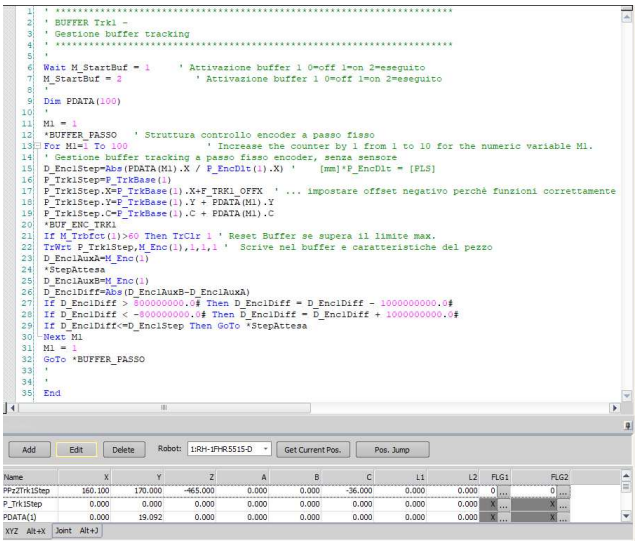


Figure 1.1: Mitsubishi robot simulator RT ToolBox3



```
' ********************************************************************
' BUFFER Trk1 -
' Gestione buffer tracking
' ********************************************************************
'
Wait M_StartBuf = 1     ' Attivazione buffer 1 0=off 1=on 2=eseguito
M_StartBuf = 2          ' Attivazione buffer 1 0=off 1=on 2=eseguito
'
Dim PDATA(100)
'
M1 = 1
*BUFFER_PASSO   ' Struttura controllo encoder a passo fisso
For M1=1 To 100             ' Increase the counter by 1 from 1 to 10 for the numeric variable M1.
' Gestione buffer tracking a passo fisso encoder, senza sensore
D_Encl1Step=Abs(PDATA(M1).X / P_EncDlt(1).X) '    [mm]*P_EncDlt = [PLS]
P_Trk1Step=P_TrkBase(1)
P_Trk1Step.X=P_TrkBase(1).X+F_TRK1_OFFX  ' ... impostare offset negativo perchè funzioni correttamente
P_Trk1Step.Y=P_TrkBase(1).Y + PDATA(M1).Y
P_Trk1Step.C=P_TrkBase(1).C + PDATA(M1).C
*BUF_ENC_TRK1
If M_TrbFot(1)>60 Then TrClr 1 ' Reset Buffer se supera il limite max.
TrWrt P_Trk1Step,M_Enc(1),1,1,1 '  Scrive nel buffer e caratteristiche del pezzo
D_Encl1AuxA=M_Enc(1)
*StepAttesa
D_Encl1AuxB=M_Enc(1)
D_Encl1Diff=Abs(D_Encl1AuxB-D_Encl1AuxA)
If D_Encl1Diff > 800000000.0# Then D_Encl1Diff = D_Encl1Diff - 1000000000.0#
If D_Encl1Diff < -800000000.0# Then D_Encl1Diff = D_Encl1Diff + 1000000000.0#
If D_Encl1Diff<=D_Encl1Step Then GoTo *StepAttesa
Next M1
M1 = 1
GoTo *BUFFER_PASSO
'
'
End
```

Figure 1.2: Mitsubishi robot programming language (MELFA-BASIC V)

Universal simulators

There is another category of robotic simulators that can be used to develop and design robotic solutions combining robots made by different manufacturers. These software tools provide a vast variety of robots to be used and not limited to a specific manufacturer. The control commands can be programmed in the most common programming languages such as Python, C and C++. Universal simulators are usually used to develop more complicated control algorithm with respect to the manufacturers simulators.
V-rep simulator is an open source universal simulator for fast algorithm development and fast prototyping. It is used for research and educational purposes. It provides several APIs that allow the programming in Python, C, C++, Matlab and Java. It provides physical engines to simulate real world physics. Sensors such as vision and proximity sensors can be simulated to develop all the functionality of the robotic cell. V-rep also allows the creation of robots by importing the 3d models of the various components. These simulators have the drawback of the complexity

and sometimes the impossibility to control the real robots. To be able to control a real robot, it is necessary to install specific APIs to communicate with the robot controller.
Figure 1.3 shows a robotic cell for pick and place applications using a parallel kinematics robot.

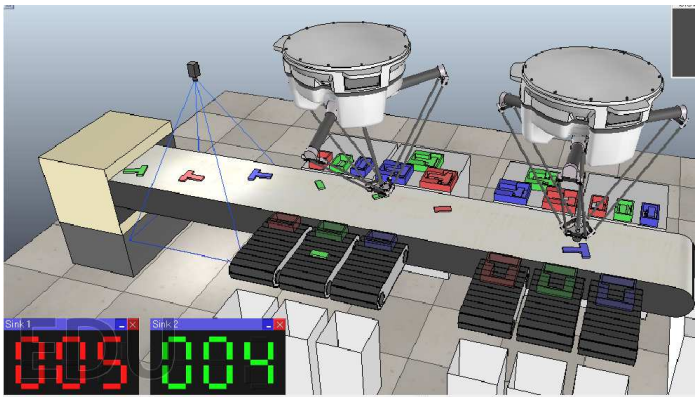


Figure 1.3: V-rep simulator

1.2 Traditional techniques for trajectory planning for contact-based operations

In this section, the focus is on the generation of the geometric path in the operation space to execute contact-based tasks. The geometric path is made by the definition of the sequence of poses to be followed by the end-effector including and initial, end and intermediate poses. These poses have to be accompanied by the sequence of values of the velocity and accelerations
In literature and industry, the starting point for path generation is the detailed knowledge of the workpiece and the operation considered. There are different techniques to generate the working path, that are introduced as follows.

1.2.1 Manual teaching

Primarily, the process of trajectory planning for high accuracy contact-based operations was done manually where an expert in robotic and in manufacturing engineering does extensive tests to define the optimal path, by manual teaching of every point of the geometric path, to be followed by the robotic tool to execute in the possible way the considered task. This technique is usually a trial and error process that require several tests to find a solution with an acceptable result. This makes it a very tedious, time-consuming process and expensive. Other important disadvantage that it is highly dependent on the level of skills of the engineer to exploit the robotic cell components to achieve a path satisfying specific criteria. To overcome these problems, there are different solutions that were introduced to automate the geometric path generation. These techniques try to optimize the performance of the robotic cell, keep a good performance consistency and execute operations over a

complicated form objects automatically without the need of time-consuming configuration. This is extremely useful when the production volume is highly customized, with many differences between a workpiece and the others. Since the path optimized for a specific workpiece is hardly adaptable to a different one also if the differences are limited.

1.2.2 CAD based techniques

The most used technique for trajectory or path generation based on the knowledge of the workpiece is based on the use of the digital model of it in form of a Computer Aided Design model (CAD). For more than two decades, many research works introduced techniques based on the elaboration of the CAD model [11, 12, 13]. Also, in recent years many solutions based on the use of CAD models still introduced [14]. This is due to the availability in most of industrial cases of the CAD model of the exact workpiece, since it is the starting point of the production chain in many production systems.
Trajectory or path planning based on CAD model are used in industry to automate several tasks that are usually done manually because they require high level of skilled operators. These tasks usually require performance consistency, so they have to be automated to guarantee always a good execution level.
When the CAD model is used for geometric path generation of contact-based operations, there are several relevant aspects to be considered. In [15], the authors provided a review for automatic path generation based on CAD elaboration. The considered application is a robotic painting applications. The introduced aspects could be generalized for different CAD-based path generation applications. The aspects to be considered when a CAD-based path generation used, are:

workpiece digital model

The first step in the path generation is the identification of the workpiece CAD model. There are several CAD models formats that are used in industry to create the digital model of a workpiece. The main two categories of models are the parametric models (e.g. STEP) and mesh models (e.g. STL).
Mesh models have a simple representation of the workpiece, representing it in simple triangles. This format has the advantage of being easy to analyze. This because of the availability for all the triangles very important values that can be used for the elaboration process. These values are the position, normal vectors and the area. The values allow calculating precisely the pose (position and orientation) of the robot end-effector. Other advantage is that a mesh model can be considered as a unique object that simplifies the elaboration. An example is shown in figure 1.4 of a workpiece in STL file format.
The disadvantages of this format is the lack of the information about material properties that is crucial for contact-based operations, since the robot comes in contact with the workpiece. Also, the lack of geometrical features such as edges may cause some difficulties to use this for applications such as welding or surface finishing. Other important limitation that a workpiece is usually represented with a very large number of triangles that makes the elaboration quite complex to do.

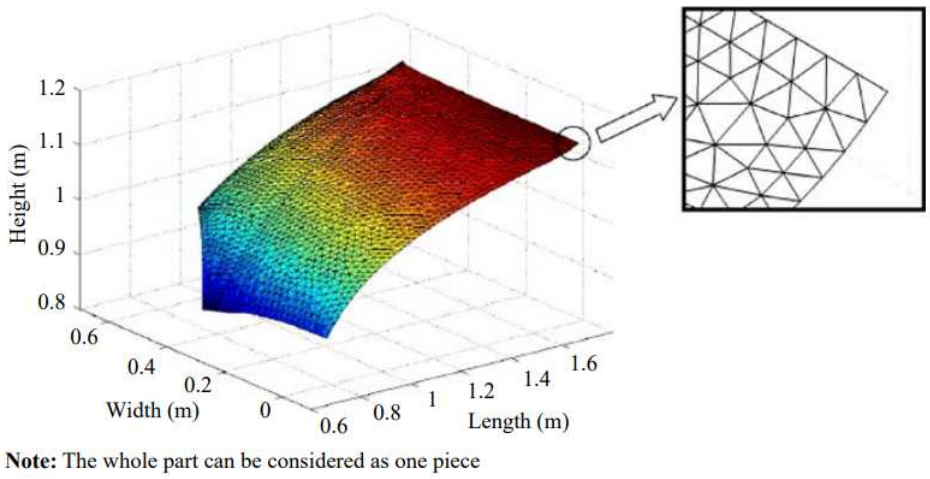


Figure 1.4: STL file format example [15]

Parametric models have some advantages and disadvantages as well. From the advantages, the information that uses for describing the workpiece, making it a full and precise description. In a parametric model, they are present in information like points, curves, lines and solid surfaces. All the information of the material used and features like edges are also described. Disadvantages of this format is the information included in it that may be more than the needed ones. The workpiece is represented by several small separate surfaces. An example is shown in figure 1.5 of a workpiece in step file format. The workpiece is represented in several parts.

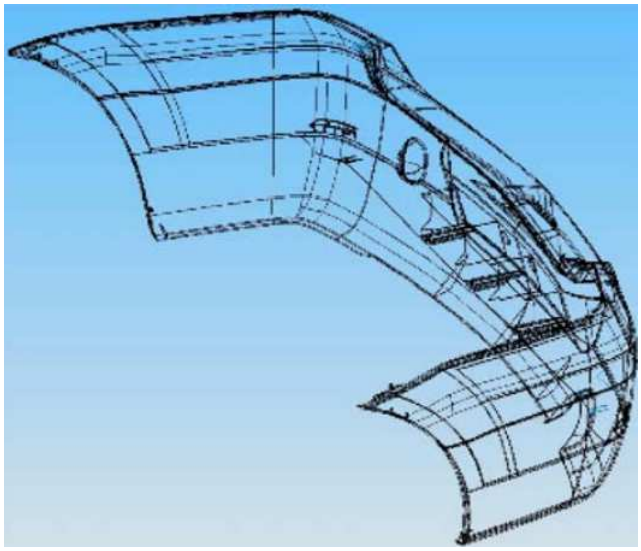


Figure 1.5: Step file format example [15]

Path pattern

Path following pattern is one of the most important factors that may influence the quality of execution of a defined process. Several path patterns were introduced for different purposes. Examples of these patters are raster and spiral patterns. The

pattern has to be synchronized with the functionality with the robotic tool and the movement speed.
In raster pattern, the behavior in the vertices in which the pattern following changes the direction, the tool has to adapt to not over deposit the material or remove more material than needed since it remains covering the same area for a relatively longer time than the linear parts of the pattern. In this case, a variable tool and robot speed is needed.

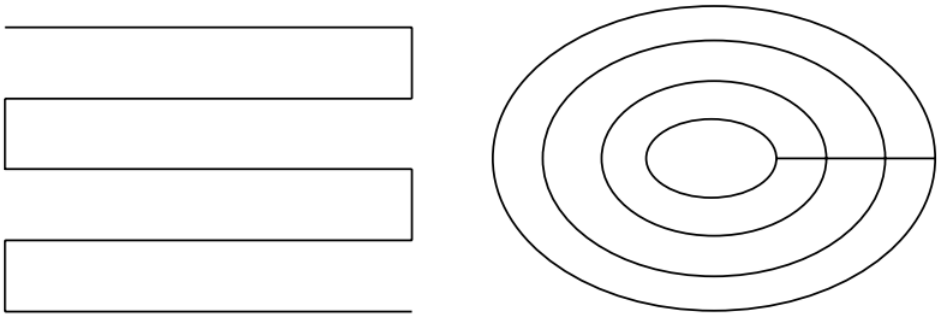


Figure 1.6: Working patterns. On left raster pattern. On right spiral pattern [15]

1.3 Advanced techniques for trajectory planning for contact-based operations

1.3.1 Vision and CAD

The CAD-based techniques have many advantages, but they have the limitation of being constrained to some aspects. These aspects are like the dependency on the 3D model assuming the perfect similarity to the actual workpiece considered in the robotic operation. The other main aspect is that these techniques are designed to execute the trajectory according to a predefined fixing position of the workpiece. That position has to be precisely defined with respect to the robot manipulator reference frame.
To overcome the dependency of the robotic cell performance to the accuracy of the definition of the workpiece position with respect to the robot, solutions were introduced to increase the flexibility of the robotic cell regarding that matter. These solutions, based on the use of vision sensors, detect the current workpiece pose and base on it adjust a predefined trajectory. That predefined trajectory is usually generated with a parametric geometric path that can be modified in correspondence to the pose obtained using the vision system.
In [16] a solution for trajectory planning for a welding application is introduced. The solution consists in generating the working trajectory in an offline way based on the workpiece CAD elaboration. Before the execution using the real robotic cell, a time of flight 3D vision system (Microsoft Kinect) is used to capture an image of the workspace of the robot. These 3D camera acquisition is elaborated to identify the current object pose. Then the CAD model is aligned to the 3D image to correct the previously generated path.

Another example is introduced in [17] for debburrig application of a Die-Cast workpieces. The solution consists of the correction of the error in workpiece positioning through capturing a 3D image of the workpiece surface and aligning it to the target surface. The 3D measurement is based on the use of laser triangulation technique for depth measurement [18]. The working trajectory for the deburring application is defined using manual teaching on a reference workpiece that is already debburred. Based on the measurement of the current object position, a correction in terms of position and orientation is applied.
Limitations are the necessity of data elaboration to be able to align the captured point cloud of the workpiece. These elaboration consists in filtering the point-cloud, removing the unnecessary data of irrelevant parts to the case considered. Other elaboration step is the point-cloud down-sampling to decrease the number of points elaborated that accelerates the elaboration process. Also, smoothing and surface estimation is important to be able to align the two models. This is due to the irregular surface of the captured 3D model because of sensor noises that make it difficult to be aligned to a CAD model with very smooth surface.

1.3.2 Learning from demonstration

Learning from demonstrations (LfD) is the approach in which a human (an expert) transfers the knowledge of executing a certain task to the robot by demonstrating the best way to do it. This method is used when the geometrical path and sometimes also the contact force are difficult to be planned using classical programming scripting and can not be represented as an optimization problem. For this kind of application, it may be simpler to demonstrate the way in which the task is executed [19]. Learning from demonstration does not require specifications of the sequence of movements and actions and low-level control actions (e.g. trajectory) of the steps to be followed to execute a task. In this way, not programming experts can design robotic tasks, since the robot is able to learn implicitly from the demonstrations of task constraints and requirements. The goal of LfD is to not repeat the exact tasks as they were executed by the expert in a constrained environment, but be able to find optimal solutions in unstructured environments.
There are many ways in which learning from demonstration models are classified. The first classification is based on the demonstrator technique. Three demonstrators are commonly used. The first is Kinesthetic teaching, in which the demonstrator moves the robot end-effector and follows a specific trajectory. The training data is collected by recording the state of the robot during the skill execution through the use of joint angle and torque measurements. The second demonstration technique is teleoperation LfD that can be used for learning trajectory and objects manipulation. The limitation of this method is the necessity to develop an input interface using external sensors (e.g. haptic sensors or joysticks. The final LfD demonstration technique is based on passive observation. This technique is based on the observation of the human expert while executing the task. The human movements are usually captured by using additional sensors fixed on the body of him.
Many applications were introduced to learn from demonstrations the working trajectory to execute in a specific way the considered skill. In these techniques, the

output trajectory is described by different parameters such as end-effector poses, forces and joint angles.
LfD, besides allowing non-expert robot programming, it is in general a data efficient technique. Trajectory learning algorithms [20, 21] can be learned from only 10 or less demonstration to learn the skill. Only one demonstration can be useful to teach the robot a high-level hierarchical task like moving car tier as proposed in [22]. Since the robot is intended to imitate the expert behavior, another important advantage of the LfD techniques is to be safe. The robot after learning the skill stays in the safe or the relevant region for task execution.
LfD has some limitations, that first one is the dependency on the demonstration interface to be used for the demonstration. The used interface limits the demonstrator comfort and the usability for specific robot. For example, Kinesthetic teaching is used for robotic manipulators, and it is hardly usable for humanoid robots. Other techniques such as visual demonstration LfD need a precise mapping between the demonstrator operation space and the robot control system. Also, the learning accuracy is highly dependent on external aspects, such as the quality of the demonstrations and use of noisy sensors.
The other important limitation of LfD is the need for labeled data. Despite the fact that LfD can use only a limited amount of data. This data has to be precisely annotated according to the design choices of the learning algorithm in defining the states, features, actions and the goals to reach.

1.3.3 Vision-based techniques

Using only vision sensors to generate the working trajectory of a robotic operation can be done by acquiring only images of the workspace of the robot and then elaborate them to generate the working trajectory. Solutions based on this idea do not need the exact CAD model of the workpiece, making them more flexible. Using only vision systems, to generate the robot trajectory consists of three main steps. In the first step, it is necessary to select the unknown workpiece from the captured images or point clouds. The second step is the one in which the area interested in the operation is selected. Finally, the last step is to refer the trajectory covering the selected area with respect to the robot reference frame and considering the robotic tool used for that task.
In welding applications, a solution is proposed in [23] for automatic trajectory planning. The solution consists of the use of a stereo depth vision sensor (Intel Realsense D415) is used to capture a point-cloud of the workpiece. That point-cloud is elaborated to detect the welding pattern, that in this case is a V-shaped groove. Once the groove is found, the trajectory to be followed by the welding torch is generated. The 3D camera is attached to the robot end-effector together with the welding torch. In this way, the robotic system can deal with more complicated welding cases. To detect the welding groove, the point-cloud is elaborated to find the surface profile variation of the workpiece. The welding groove is in correspondence with the area having a huge slope variation, and it is placed between two flat or smooth areas.
In [24], a similar solution based on the use of only the data collected using a vision sensor (Microsoft Kinect V2) is proposed for the generation for to be followed by

a glue spray gun in the footwear manufacturing. The developed solution is able to generate the path over the surface of 2D flat surfaces like leather and also the 3D objects like shoe soles. The gluing pattern is found by identifying the contour pattern and follow it or follow a path inside it.
The above-mentioned solutions rely on a single acquisition of the workpiece. That make it is possible to generate the working trajectory over only one face that is the one visible in the acquisition. To develop a more flexible solution that considers all the faces of the workpiece to generate the path, it is necessary to acquire more data about the object covering the desired faces. Then elaborate the acquired data for the trajectory generation. In the next chapter, different techniques for complete model reconstruction are introduced.

1.4 Collaborative robotics

Manufacturing plants, for several decades, are being designed to increase the production volume and are characterized by the presence of bulky and heavy machines working at a very high speed. That conditions made it necessary to introduce more specific safety guidance to guarantee the safety of human operators present in the manufacturing plant during the functioning of the machines.
Particularly, safety guidance when industrial robots are used includes the necessity to enclose them behind safety fences to avoid any kind of interaction between the robots and human operators and also other obstacles that are not previously considered during the programming of the robot and the task design. Several safety solutions are usually used to stop immediately the industrial robot as soon as a human or an obstacle enters the danger area in proximity of the robot. These safety solution include the use of laser beams or doors with limit switches.
To increase the flexibility of the industrial robots to work in dynamic environments while guaranteeing a high safety levels without the need for safety fences, laser scanners and the implementation of safety algorithms new robotic manipulators have been introduced. These robots are equipped by sensors and mechanical joints that limit the intensity of the impact of not expected contacts. These robots or precisely collaborative robots allow the close interaction with human operators safely. By using these robots, the cost of the robotic cell could significantly decrease because of the decrease of the cell footprint and the elimination of safety fences and sensors.
Collaborative robotics is a wide term that includes industrial applications where the machine, or more specifically a robotic manipulator, that works in a proximity with a human operator. The opportunity for a closer interaction between the robot and the human operator allows using the human input to enhance the flexibility of the machine and allows the execution of more complicated tasks that would have been not possible to design in case of a fully automated system [25]. The research in the field of collaborative robotics is an open topic for more than a decade [26].
Despite the growth of the number of collaborative robots deployed in industrial applications of 11% in 2019, the market share of the collaborative robots with respect to the overall robots of 4.8% in 2019 still quite small as mentioned in the World Robotics 2020 Report [27].
To guarantee the efficiency and usability of a collaborative robotic application, de-

vices used in the design of collaborative robotic solutions have to satisfy the requirements of safety and ease of interaction [28, 29, 30]. These requirements mean that robotic manipulator have to be designed for example with lightweight structure [31] and with compliant actuators. Also, the process to be automated in such a way that benefits from the collaboration with a human-centered design in terms of layout and the application itself. Other components of the collaborative applications have to be integrated to satisfy the requirements. These other components include extra sensors (e.g. vision sensors) and an efficient interface to exchange the information with the human operator.

1.4.1 Safety regulation

The safety conditions of collaborative robots when applied in industrial environments is regulated by some rules. These rules are defined under the section of collaborative safety features of the safety technical standards [32, 33] for industrial robotics. Specifically, in the technical standard ISO 12100, a collaborative application has to guarantee the robot safeguarding requirements. The term safeguarding does not consider only the definition of risk areas where the human can not access, but whatever safety measure to reduce the risk. These safeguarding safety features include:

- Hand guiding: industrial uses of this method are mostly for moving heavy loads at slow speeds. It is used in applications where human and robot share the workspace and have to cooperate for lifting and handling large objects [34]. The benefits of this technique is the reduction of the tact time and the workspace requirements. Hand guiding is different from walk-through programming that is a technique used for teaching the movements in manual mode[35]. Hand guiding does not have a limit on speed, but a suitable range is between 1000 and 1200 mm/s that is dependent on the movement speed of human arms speed while generating the movement trajectory.

- Speed and separation monitoring: this technique consists of the division of the area around the robot in ranges based on the distance with respect to the robot. If the human operator enters the different ranges, a change in the robot behavior has to be executed. For example, if the human operator enters the farthest range the robot starts to slow down its movement speed and as the human approaches the robot it has to decrease more the speed until complete stop of the movement when the human operator enters the closest area. This technique can be applied using vision systems or laser scanners for the detection to stop the robot or to apply other techniques such as speed and separation monitoring [36, 37]

- Power and force limitation: the goal of collaborative application is to eliminate the risk of injury of human operators. This can be done by decreasing a possible harmful impact of accidental collisions. So these collisions can create a painful sensation similar to the pain of any daily-life contact. The quantify process of the pain sensation and the possibility of a contact can create an

injury, it is not a straightforward and not intuitive. This measurement could be done by measuring the pressure on the body of human operator [38]. The pressure generated by the impact force over a certain surface in amplitude in a certain period of time has to be evaluated with a solution to reduce these value in case of the increase over a predefined threshold. Practical implementation of this collaborative feature consists of solving the problem from a design point of view by avoiding the use of pointed, sharp objects and the use of elastic material. These concerns are extended to the overall robotic cell components, not only the robot. Simulations and measurement devices can be used for quantifying the pressure [39].

- Safety-rated monitored speed: introduced with the 2020 version of the technical standards ISO 10218 and ISO/TS 15066. It is not a feature, but it is the definition of the state to which the robotic system is switched before the human exposure to a hazard. It is usually a stopping phase that from practical point of view could be associated to the robot workspace entering safeguarding devices or the stopping conditions when Speed and Separation monitoring technique is used.

1.4.2 Mechanical structure and control of collaborative robots

The main difference of collaborative robots when compared to industrial robots is the possibility to share their working space with human operators without creating harmful situation for them. Collaborative robots when used in dynamic environment, in case of collision or interaction with something, they have to limit the impact intensity. This goal is mainly achieved by introducing three components. The first is flexible mechanical design of robot joints. The second is the use of sensors to detect any anomalies in the robot behavior in case of collision or interaction. Finally, control strategy to be able to react properly to them to avoid harmful impact on the hardware components and to the surrounding environment.
To design a robot capable to absorb the interaction energy to limit the impact, it is necessary to design mechanisms with higher compliance. In collaborative robotics, the term compliance is usually linked to higher safety levels when an interaction happens. From the hardware design point of view, the compliance can be achieved adding visco-elastic components in the joint actuation with impedance control [28]. From the first examples of non-rigid actuators the series elastic actuators (SEA) that were introduced in [40], where mechanical passive compliant component are added in the connection between the actuator and the load. In [41], the series of the elastic components are replaced by a variable viscosity damper [42, 43]. The introduction of compliant elements may be a reason for low accuracy due to the impedance of the actuators. In [44], a variable stiffness transmission scheme is proposed to enhance flexibility and good performance. The control scheme is based on an optimal control based on minimum time requirement with safety related constraints.

1.4.3 Block based programming

To allow efficient collaboration between human and robots, new paradigms for simplification of programming and task planning are continuously introduced [45]. This due to the lack of easy tools and programming interfaces that allow the robot programming by users with limited programming knowledge [46].
From the introduced programming tools for cobots, the skill based programming [47]. Following this programming paradigm, a robotic application is made by several tasks followed in a specific order. To design each task, several robot skills are used. These skills are made by a primitives like robot movement command, gripper opening and also sensor related primitives like capture an image. Block based programming [48] is also a programming tool in which a graphic interface is used for robot task programming. Task design consists of the use of graphical interface in which every robot action is represented in the form of a block. To program an application, the user has to drag and drop the blocks in the desired order.
A more user-oriented programming have been introduced to allow the programming through the user natural language input [49, 50]. The user can use a vocal commands to create simple robot programs. Other user-oriented programming method is learning from demonstration and kinesthetic teaching, in which the user moves by dragging the robot to demonstrate the desired task [51, 52].
Recently, also robot manufacturers implemented some from the introduced techniques. These techniques include the robot block-based native programming environment like TM flow [53], shown in figure 1.7, for Techman collaborative robots programming. On the left part of the figure, the available block of the different functions are shown, and these block can be dragged and dropped to design the robot task. Other robot manufacturer provide a teach pendent with an intuitive graphical programming environment [54].

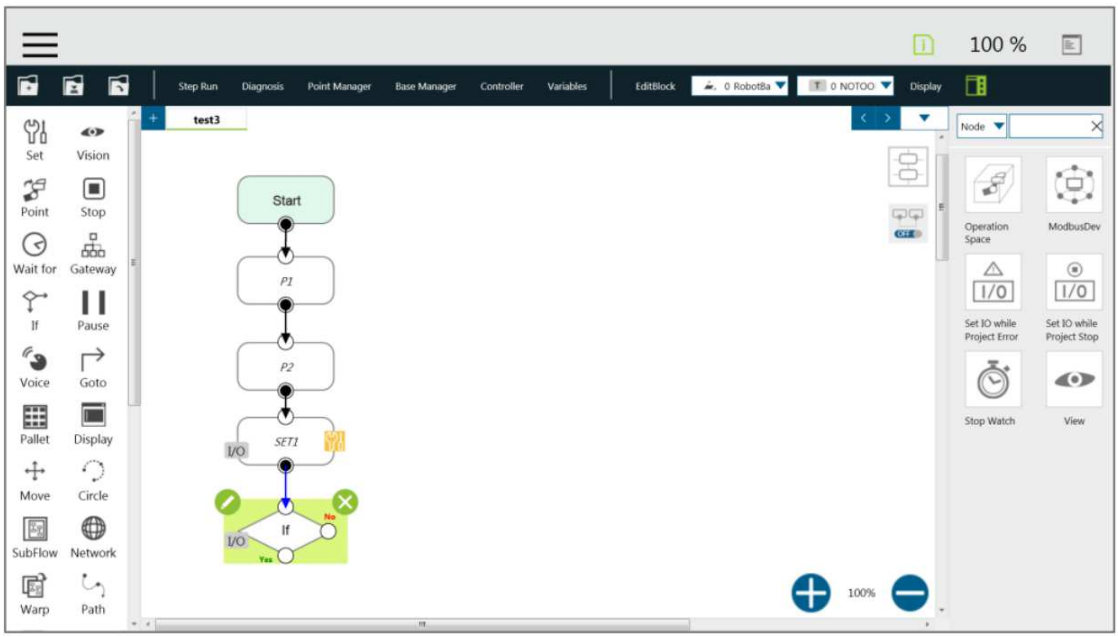


Figure 1.7: TMflow programming environment [53]

1.4.4 Collaborative plug and play concept

Dynamic changes of the production requirements in terms of work flow, production customization and volume created a huge demand for robotic equipment that is easy to reconfigure without or minimum reprogramming effort. The term "plug and play" is derived from the IT domain. It consists of the design of a component in such a way that allows to plug or unplug it in the minimum time possible. In a robotic system, a huge variety of components (e.g. robots and sensors) are used. These components have to interact with each other to achieve the desired goal. Configuration and setup processes are time-consuming and require huge engineering effort because of the lack of standardization of these components.
In recent years, with the increase of collaborative robotics deployment, and having programming simplification as one of the main characteristics of these robots. The term plug and play is being related to the design of collaborative robotic tools and end-effectors to extend the ease of use concept also for them and not only to the robot programming. These plug and play tools are intended to be installed, reconfigured and programmed without programming experience. In industry, examples of plug and play tools are available. Such as finger grippers, vacuum grippers, surface finishing tools and screw deriving systems [55].
This concept can be extended to the overall robotic systems to planning the production and the execution of it. In [56] a plug and produce solution is proposed to simplify the configuration of the production system. It consists of the development of a control framework over the robot operating system that allows the change and the configuration of hardware modules of the system.
Plug and produce solutions introduced to support the discovery, configuration and integration of the components pf a robotic cell. They still present limited configurations possibilities, limiting the capability to the robots to operate in dynamic environments. To solve that, solution for plug, plan and play solutions are proposed [57]. In these techniques, to enhance the flexibility of the robot to adapt in modular production systems aspects such as dynamic motion planning, simulation and collaborative robot human interaction schemes are used.

1.5 3D vision

Recently, thanks to the fast development of the vision sensors hardware with high resolution and the introduction of very efficient software, computer vision solutions are continuously being deployed to execute a big variety of tasks. These tasks are in different areas such as surveillance, face recognition, autonomous driving, navigation, inspection, quality monitoring, pose estimation and environment mapping. This is because the color images captured by a high resolution sensors contain enough data that can extracted using machine learning algorithms.
Color images lack a crucial element that is extremely useful when a computer vision solution for robotic application is developed. In the robotic field, depth value is important because it gives the robot a complete knowledge about the environment surrounding it. Not only by knowing the 2D coordinates of an object, but also how far it is. This information can is crucial in several ways in the robotics field, where

the robot has to come in contact with an object, or it has to avoid the collision with it [58]. Examples of applications where depth value is useful are object handling [59], pick and place applications [60], pose estimation [61] and collision avoidance [62].

1.5.1 Working principle

There are different techniques for the depth value measurements. These techniques are summarized in the following way:

1. Time of flight: in this technique, the sensor illuminates the scene by sending light signals generated by a solid-state laser or a LED. This light signal is a modulated signal, for example a square or sinusoidal wave, and invisible for human beings. By measuring the phase shift between the emitted signal and the measured one reflected from the scene, it is possible to calculate the distance of every point or pixel. In figure 1.8, the concept of time flight is summarized.

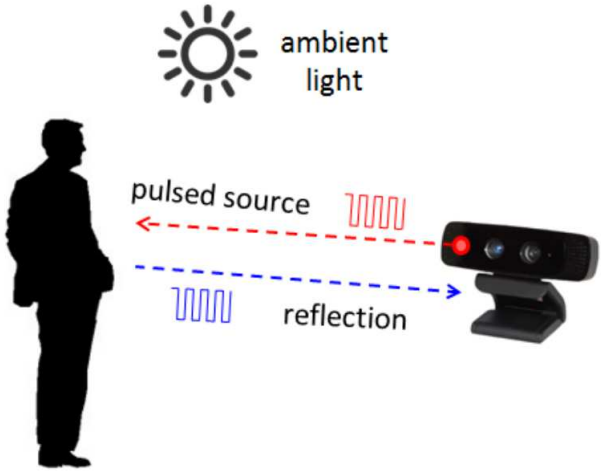


Figure 1.8: Time of flight working principle [63]

   The output is a pixel array where the calculate depth values are stored. In recent years, with the development of hardware with a good performance and low cost allowed to spread the use of the Microsoft Kinect [64] that is based on this technique.
   Lidars or Light Detection and Ranging is based on the use of time of flight concept to measure the depth values. A LIDAR is made by several emitters that are placed at equal distance and cover 360°around it. This technique has a low latency and longer range. The main use of these sensors is in the mobile robotics and particularly in self-driving cars.

2. Stereo depth: this technique mimic the behavior of human eyes. It consists of the use of two vision sensors displaced at a known distance between them. By capturing two images using the two sensors. The separation between the two

sensors used will cause a position disparity in the image for a specific observed pixel. For each pixel, in a comparison between its positions in both of the captured images, it is possible to calculate the actual depth.
In figure 1.9, it is shown how the depth value can be computed considering that the two sensors are placed in the points A and B at the distance X between them.

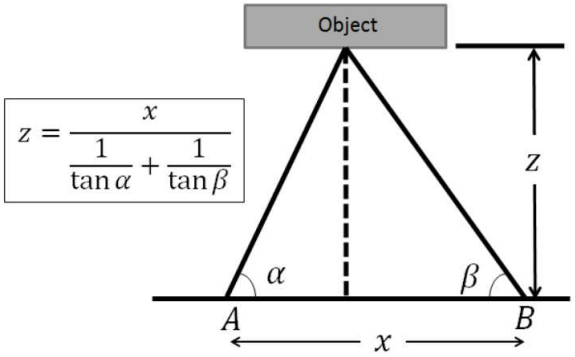


Figure 1.9: Stereo depth working principle [63]

The accuracy of the depth values is strictly dependent on finding the correspondent pixels in the two images. This can be done by using feature matching algorithm that require computationally intensive calculations. The result of these feature matching algorithm is dependent on the illumination condition and how many visual features the observed scene have. With more features such as change in color intensity, the depth measurement is more accurate.
The most used 3D cameras based on the use of this technique are the Intel Realsesne D400 [65] product family. These cameras have an acceptable resolution and a low cost.

3. Structured light: a structured light vision system is made by a projector and a receiver. The projector, projects a light pattern over the observed scene. The receiver detects and measures the projected pattern. The pattern, when projected over a surface with different depth values it becomes distorted and the receiver, based on that distortion, calculates the depth values.
In figure 1.10, the concept on which the structured light technique is shown.
The pattern characteristics can be programmed to satisfy some conditions, such as the object reflection or environmental illumination. For higher accuracy, several patterns have to measured that make it not suitable for dynamic scene since the components have to be static during the acquisition. Because of that, it has a lower frame rate with respect to the previously mentioned techniques. These devices have a very accurate spatial measurement (x and y) and use High Definition color cameras for capturing the texture of the scene. Production cost of pattern projectors are higher when compared to the hardware cost of the stereo vision or time of flight devices.

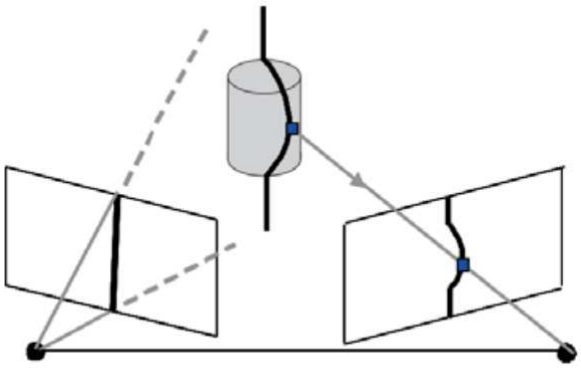


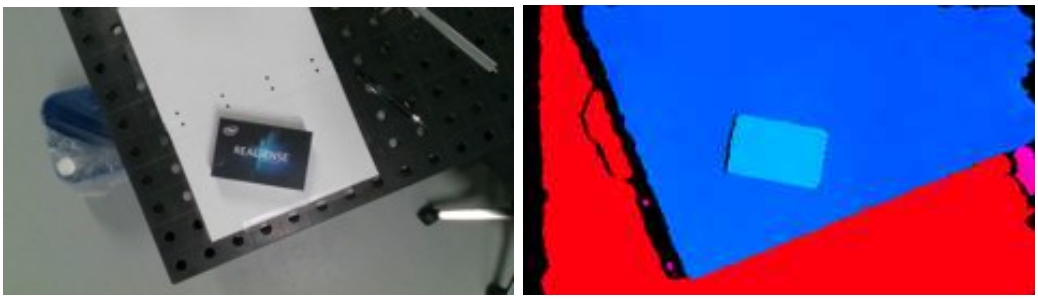
Figure 1.10: Structured light working principle [63]

1.5.2 Acquired data

The mentioned 3D vision techniques are used to calculate the depth values of every point of the observed scene. The acquired data can be represented in different formats. The most used formats are RGB-D images and point clouds representation.

RGBD images

3D vision devices are usually made by integrating several electronic components. In some devices, it is also integrated an RGB sensor to capture a color image of the observed scene. Two separated images can be saved of the observed scene color and depth. In the color images, every pixel is presented with three values (RGB) that represent a value for red, green and blue colors. In the depth image, for every pixel it is saved a value representing its distance from the camera origin.
In figure 1.11, an example of color and depth images captured at the same moment is shown. The depth images to be more intuitive for the users, color map representations are always used. In this color maps, depth values are represented by different colors based on their values. The two images are captured using different physical sensors having different characteristics (e.g. dimensions, focal length, principal points and referred to different origins). To use them in mutual way, it is necessary to execute an alignment procedure, as will be explained later.



(a) Color image *(b) Depth image*

Figure 1.11: Example of color and depth images taken by D415 3D camera

Point clouds

The other form of data to represent the data acquired by a 3D camera is the point-cloud representation. In which the two-dimensional representations of color and depth images are extended to the three-dimensional space.
Point clouds represent the information of a set of points in the space. Each point has coordinates values (x, y and z). In case of 3D vision, they may have also other values representing also the RGB or color values and normal vectors over.
In industry, point-cloud representation is widely used in many applications thanks to the number of information incorporated in it. In the field of mobile robotics and navigation, point clouds are used for understating of the surrounding environment to better interact with it [66, 67]. Point clouds are also used in the construction field to inspect and for quality monitoring of constructed structures [68]. In industrial robotics, point clouds are also used to execute tasks such as pose estimation [69] and to calculate the optimal grasping configuration of unknown objects [70].

1.6 Digital model reconstruction using vision sensors

In recent years, the interest of having as much as possible of information about the surrounding environment. In the case of industrial robotics, the interest increased to know more information about the workpiece to be handled by the robotic manipulator. There are some solutions proposed to integrate several partial information to have overall knowledge of it.
Acquiring different images of the workpiece from different perspectives and using reconstruction algorithms, allows the integration of these data of the different view points of the workpiece to reconstruct a complete 3D model of the workpiece.

1.6.1 2D vision sensors

2D images have the color, illumination and shadow information of the observed scene. Some algorithms based on Machine Learning and on the use of thousands of labelled data, that associate a set of color images and the relative 3D model, are able to reconstruct the 3D model of an object using only color images.
A machine learning model (Neural Reflectance Surfaces (NeRS)) has been proposed to reconstruct the 3D model of an object based only on the use of multi-view images [71]. The reconstructed model also includes information about the texture, color, illumination and shininess of the object. The solution has been applied to images of several objects from marketplaces, of which few view points are available. Input images are ranging between 8 and 10 images for every object considered during the training process.
In the robotics field, solution have been proposed solutions for 3D reconstruction based on multi-view color images [72]. The authors introduced a robotic solution for quality inspection of goods using the lowest possible number of images. In this solution, a camera that is attached to the robot end-effector is moved around the object to capture few images of it from few view-points. Using a machine learning

algorithm, these few images are used to reconstruct the 3D model of the object in the form of point-cloud. An evaluation procedure is done to detect if the point-cloud is complete based, for example, on the evaluation of holes in the point clouds. If the point-cloud is incomplete, the robot pose is calculated to capture and image covering the remaining part. Fewer solutions are proposed for the 3D reconstruction of the minimum possible number of images. In [73], a solution based on the use of only one colored image is proposed. The solution, by elaborating the image, estimates the information about the 3D surface and the color to be projected over that surface. By combining both the outputs, a 3D model is reconstructed.

1.6.2 3D vision sensors

In the field of computer vision, 3D model reconstruction using 3D images (RGBD) is a well known problem and many techniques have been proposed to solve it. Professional scanners are available in the market for many years. In recent years, consumer level solutions has have been developed thanks to the introduction of consumer level devices having low-cost and good performance (high resolution and high frame rate) such as Microsoft Kinect devices (V1, V2 and Azure) that are based on time of flight technology and the Intel Realsense stereo depth devices family.
The pipeline for 3D reconstruction using depth cameras and multi-view color and depth images consists in the following steps [74]:

1. Depth frame filtering: depth images usually contain a lot of noise due to missing depth or where the object is too close or too far to the camera, these parts are usually substituted by zero depth values. Zero depth values can be generated because of parts with shadows, surface reflections and geometric discontinuities. Other reasons for depth noise are related to the camera, such as depth inconsistency, where a static point can have different depth values over time. Also, the measured depth value may differ with a percentage of error proportional to the distance of the observed point with respect to the camera [75]. To solve these depth problems, a bilateral filter is used [76] to smooth the images while maintaining geometric features like edges.

2. Camera tracking: estimate the camera pose while taking every image to refer them to a common reference frame. The camera pose is important to be able to map every one of the observed points from being with respect to the camera frame to the world frame. This process is done by identifying a set of common points visible in the different frames and then apply 3D maps registration. Iterative closest points (ICP) algorithm are used for the registration. Using an optimization process, the homogeneous transformation matrix that if applied to the points from the second frame, the two frames will coincide. The matrix is the one to obtain the minimum positional error between the translated and rotated points of the second frame and the correspondent in the first one [77, 78].
   Other techniques that are possible to be used for the registration are based on the use of fiducial markers positioned on the workpiece or the workbench. The poses of the markers in the different view-points are used to calculate the

rigid body transformation between the multi-view point clouds.
Other registration technique is based on the use of surface features matching [79]. In these techniques, the surface features such as corners, edges and blob are used for the matching process. These features are calculated considering variations in colors and textures. When point clouds are considered, other features can be used like curvature, normal vectors direction of k-neighboring points and local geometries characteristics [80].

3. Depth images merge: The last step consists of merging the depth maps in a common model based on the calculated poses. Based on the data structure in which the data from depth frames are integrated, two techniques can be used. The first is volumetric representation [81] and the second is surfel representation [82]. In volumetric representation, a three-dimensional grid made of voxels is used to store the depth values from the different frames and represent the observed scene. The voxels have other attributes of the color. The other representation is Surfel representation that is usually used for surface rendering. In which, the observed surface is represented as discredited surfels (surface elements) in form of points or discs. It is a generalized version of point-cloud representation. Every surfel have attributes describing the related part from the observed surface such as space coordinates, normal vector, color and radius

Many applications are proposed for 3D reconstruction based on the use of 3D vision sensors. In[83], multi-view color and depth images are used to reconstruct a 3D model of an observed indoor scene. In [84], for detecting welding seams and edges based on the 3D reconstruction from multi-view color and depth images. The multi-view color images are elaborated to detect the welding seams and edges of a workpiece. When they are found, the welding seams, the edges are used together with all the information in color and depth images to reconstruct a point-cloud having the workpiece and the welding areas.

1.6.3 3D vision sensors and using robot poses

When the 3D camera is attached to the robot end-effector, the pose of the 3D camera to capture the images can be calculated. Also, the rigid body transformation between the poses of the multi-view images can be calculated since the scanning trajectory has been defined previously. In this way, the several point clouds can be registered together and referred to a common reference frame or the world reference frame.
In [85], a solution proposed for the 3D reconstruction of a workpiece using multi-view point clouds. The point clouds are capture using a structured light scanner that project a sinusoidal light pattern and a receiver measures the variations of the reflected pattern to reconstruct the point-cloud. The registration of the different point clouds can be done knowing the movement between a camera view and the others without using computer vision-based registration techniques like Iterative Closest Points (ICP).
The quality of this technique relies on the used tracking device that moves the 3D

camera to reach a defined pose and its accuracy. In the shown case, a 6-degrees industrial manipulators is used. That robot has a low repeatability error of $40\mu m$ and a high absolute accuracy error of almost $1mm$. Also, the output quality is based on the accuracy of the rigid body transformation that describe the geometric relation between the camera reference frame and the used robot reference frame. The output presented lower precision with respect to other techniques for the registration process. The other technique is based on the use of markers that are placed randomly over the surface of the workpiece and based on the detected position of those markers in every point-cloud the relative transformation matrices are found.
In [86], the authors introduced a similar solution based on the use of a priori calculated camera poses while capturing every one of a multi-view images. The goal of this work is to reconstruct a 3D model to estimate the grasping point of an unknown object. The results showed that the proposed method based on the use of volumetric representation and Truncated Signed Function (TSDF) for the description of the observed scene representation in a voxel grid. The obtained 3d model using this fusion technique is better than the one obtained by stitching several point clouds together. Using more images allowed to increase the accuracy of the reconstructed model, and 10 to 15 images are necessary to obtain a reasonable result. Since the goal of the work [86] is to evaluate the efficiency of the system to calculate the grasping pose, not enough data are presented for the evaluation of the accuracy to reconstruct a 3D model of a specific workpiece.

1.7 Limitation of the available methods and possible solutions

In this chapter, several aspects regarding the trajectory planning for a robotic contact-based operation are summarized. The discussed topics covered the following concepts. Collaborative robotics have been deployed in many manufacturing systems in recent years. A main characteristic of cobots besides high safety is the programming simplicity and decrease of configuration time. To not lose this main advantage, robotic tools for task design have to be simple to use as well as the robot itself.
Traditional techniques for trajectory generation like manual teaching and CAD-based methods are able to generate a very accurate geometric path, but they lack simplicity of use and require very high engineering effort and time-consuming. They require also some requirement about the high degree of similarity between the CAD model and the actual situation of the workpiece, and they require also a precise fixing position of the workpiece in the robot working space. In SMEs, when used for in collaborative robotic cells where production has low size and highly customized and working environment is extremely dynamic, these traditional methods are not suitable for the lack of flexibility, simplicity of use and require high engineering effort to design a task.
More flexible solutions were proposed for trajectory generation robotic surface following applications on unknown workpieces based only on the use of vision systems. The proposed solutions were limited to applications where a contact was not re-

quired for task execution, like glue deposition using a spray gun. Or the results were limited to evaluate the overlapping accuracy of the generated trajectory with the real trajectory without evaluating the performance executing the task.
In this work, we propose an easy-to-use pipeline for trajectory generation of robotic contact-based tasks in collaborative robotic cells. The solutions proposed are based only on the use of the data acquired by a low-cost vision systems (Intel Realsense D415 and D435 3D cameras) that can be fixed or moving by attaching it to the robot end-effector based on the requirement of the task. Force control is used for online compensation of errors in the form of the workpiece, positioning and errors related to the depth sensor noise.
Different aspects that allow a successful execution of the contact-based tasks like degree of similarity between the data captured by the vision system and the workpiece, positional error of the generated trajectory and contact force between the tool and the workpiece while executing the task are evaluated. The solution is intended to be used by an end user that is non-programming expert, regardless of the definition of some input parameters, and be automated from the design of the scanning procedure until the generation and the execution of the trajectory.
The following sections introduce the tools and the methods considered developing the proposed solutions. After that, the experimental results are shown.

Chapter 2

Software and hardware tools used in the development of robotic solutions

To automate the generation of the working trajectory of a robotic contact-based operation, where the digital model of the workpiece is not available, or it is different from its real state, it is necessary to reconstruct the 3D model of the workpiece and elaborate it to generate the trajectory. In this chapter, the hardware and software tools used to allow the development of the proposed solutions are introduced. Will be highlighted the technical specifications that may influence the performance of the developed solutions. The introduced tools include the devices used for data acquisition, software and hardware tools for the development of the proposed solutions and the added hardware and software for accuracy evaluation software.

2.1 Collaborative robots

The main component of the proposed solutions in this work are the robotic manipulators. The objective using the robotic manipulator is to scan the interested workpiece by moving the 3D camera around it. The second objective is to move the robotic tool attached to its end-effector to execute the considered contact-based robotic operation.
The choice of the other components of the proposed robotic solution for contact-based operations such as 3D camera model, feasible workpiece dimensions and robotic tool maximum weight are based on the manipulator used and its hardware specifications and physical limits, especially maximum payload and maximum reach.
In all the developed solutions and the experimental setups used in this work, two models of collaborative robots are used.

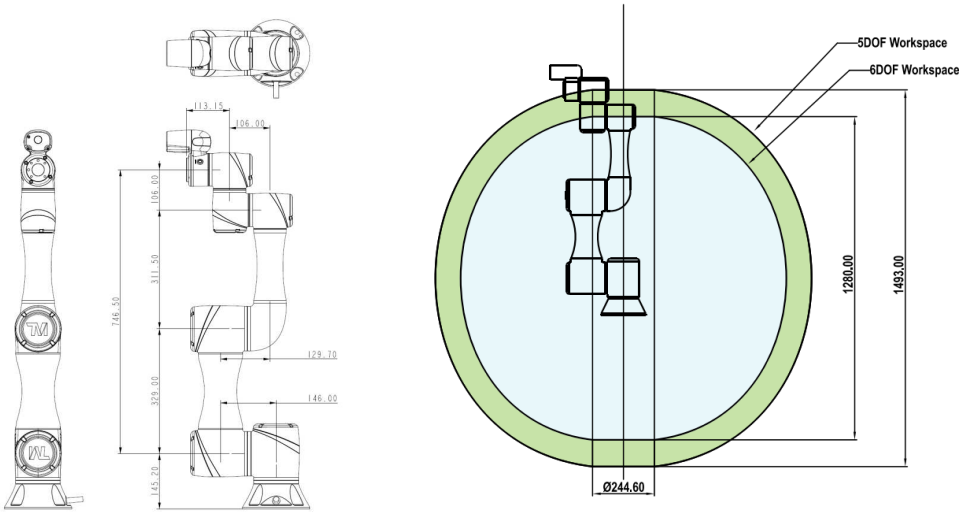
2.1.1 Techman robot TM5 - 700

Techman robot TM5 - 700, shown in figure 2.1, is a commercial six axes collaborative robot. It provides a simplified graphical block-based interface for robot programming

and a built-in RGB camera.

Figure 2.1: TM5 - 700 collaborative robot [87]

TM5 - 700 collaborative robot [87] has a maximum reach or the working spherical (radius) range from the base about 700 millimeters as shown in figures 2.2. It has also a maximum payload of 6 kilograms for closer poses with respect to the robot center of gravity, and decreases gradually toward 2.5 as the distance from the center of gravity of the poses increases more than 150 millimeters.



(a) TM5-700 dimensions *(b) TM5-700 reach*

Figure 2.2: Techman robot TM5-700 technical specifications [87]

2.1.2 UR5e

The second robot used is the Universal Robot UR5e commercial collaborative robot, shown in figure 2.3. UR5e collaborative robot [88] has a maximum reach of 850 millimeters, as shown in figure 2.4. It has a maximum payload of 5 kilograms for closer poses to the center of gravity, and decreases towards 3.2 kilograms as the

Figure 2.3: UR5e [88]

distance of the pose increases more than 350 millimeters with respect to the robot center of gravity.

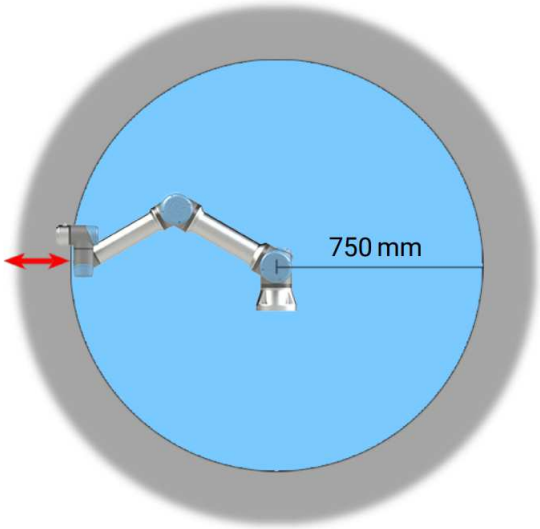


Figure 2.4: UR5e reach [88]

2.2 3D vision sensors

Adding external vision sensors to the collaborative robotic cell allows monitoring the robot surrounding area and to allows to better interact with the environment while executing the task. In robotic applications, where the robot has to manipulate, interact or touch a workpiece, 3D vision sensors provide the necessary depth and color information of the workpiece. This 3D information is elaborated to generate the working trajectory to be followed by the robotic tool used to execute the contact-based robotic operation.
A crucial component for the robotic solution proposed efficiency is the 3D vision

sensor. A suitable 3D vision sensor for the proposed robotic solutions has to be characterized by a low weight. This is necessary to allow its attachment temporarily with the robotic tool without exceeding the maximum payload of the robot. It is necessary to scan the workpiece from different point of views, satisfying a minimum distance with respect to the workpiece. To guarantee that, it is necessary to use a small dimension 3D camera to simplify its manipulation and keep it pointed toward the workpiece all during the scanning process.
D400 3D cameras present a wide range of low-cost products (price range between 300 and 400 Euro) that can be applied in different applications based on the performance required. These 3D cameras are based on the stereo depth working principal to calculate the depth of every pixel within the field of view of the camera. This technique exploits two vision sensors to capture two images from two point of views. Knowing previously the position of the two cameras with respect to each other and comparing the positions of the pixels in the two images, it is possible to calculate the distance of all the pixels.
The cameras are also equipped with a color vision sensor to acquire RGB images of the scene within the camera field of view. D435 stereo depth camera is shown in figure 2.5 as an example of the D400 3D cameras provided by Intel. The 3D sensors are highlighted to show the RGB color sensor and the two sensors used for the calculation of the depth information using stereo technique.

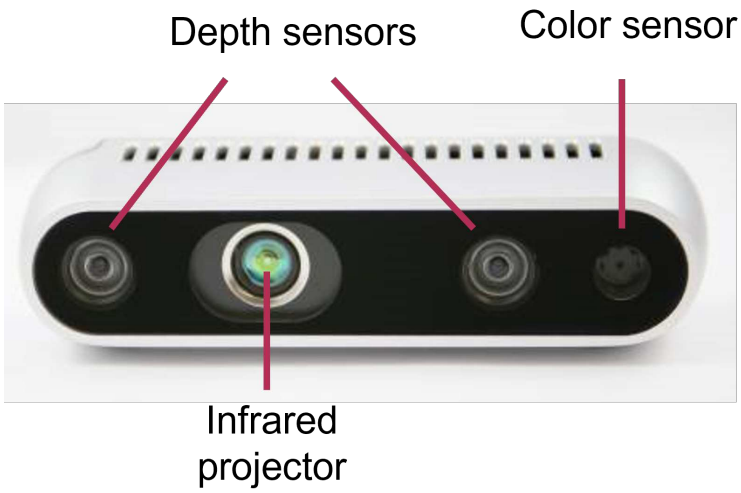


Figure 2.5: D435 3D camera color and depth sensors

To develop the proposed solutions, two 3D cameras from the Intel Realsense D400 3D devices' family are used. These two devices are the D435 and the D415 stereo depth 3D cameras. The two 3D cameras differ in the field of view and the shutter type.

2.2.1 D435 3D stereo depth camera

The D435 stereo depth camera [89] is a general purpose 3D camera for robotics applications. This camera has a wide field of view of 85 degrees that allows to cover bigger scenes that is useful in applications such as drone navigation and collision

avoidance. D435 shutter is based on the use of global technique that allows to capture depth images of high-speed moving camera without having blurring depth images.
The captured color and depth images resolution can be configured to reach a maximum resolution of 1280 x 720 megapixels. The resolution is inversely proportion to the minimum distance between the camera and the workpiece. At maximum resolution, the minimum distance is about 280 millimeters. The frame rate is up to 30 frames per second (fps) for higher resolution configuration and 90 fps for lower resolution configurations.
The D435 3D camera drawings are shown in the figure 2.6, it could be enclosed in a box with the dimensions of 25 x 25.5 x 90 millimeters.

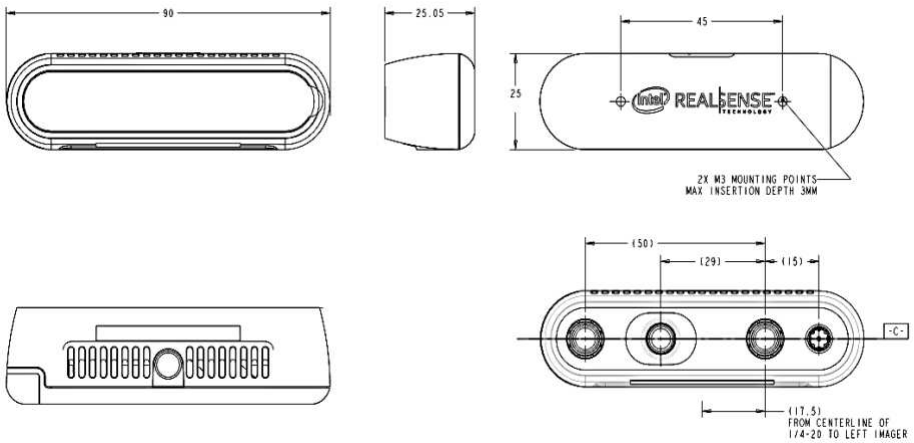


Figure 2.6: D435 3D camera dimensions [89]

2.2.2 D415 3D stereo depth camera

The D415 stereo depth 3D camera [89] has a smaller field of view with respect to the D435 one, about 65 degrees. This means a smaller observed scene, but means a higher pixel density and more granularity considering the same image portion captured by the two 3D cameras. This means from a performance evaluation point of view a higher accuracy since the same area is described by more pixels. For the 3D scanning applications, the D415 is the preferred one for having higher accuracy. The shutter technique used is rolling shutter that require a couple of frames to calculate the depth frame, that is irrelevant for a static objects like in the considered case.
The captured color and depth images resolution can be configured to reach a maximum resolution of 1280 x 720 megapixels. The resolution is inversely proportion to the minimum distance between the camera and the workpiece. At maximum resolution, the minimum distance is about 450 millimeters. The frame rate is up to 30 frames per second (fps) for higher resolution configuration and 90 fps for lower resolution configurations.
The D415 3D camera drawings are shown in the figure 2.7, it could be enclosed in a box with the dimensions of 23 x 20.05 x 99 millimeters.

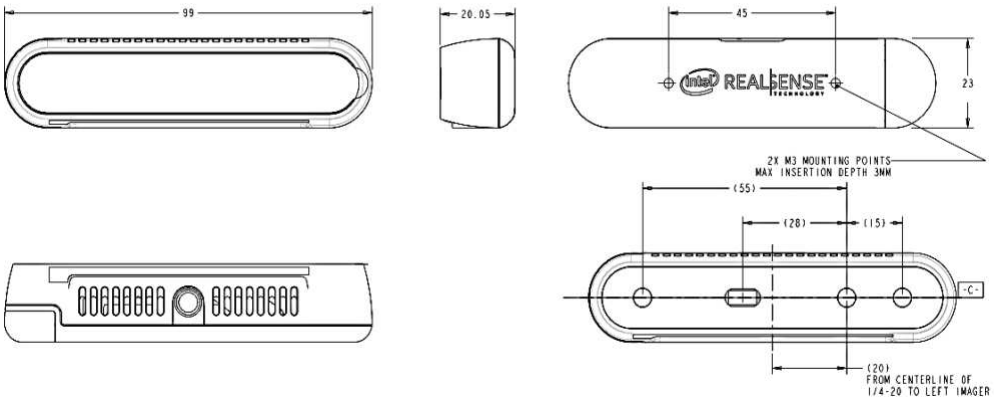


Figure 2.7: D415 3D camera dimensions [89]

2.2.3 Artec Eva scanner

Artec Eva scanner, shown in figure 2.8 is used during the accuracy evaluation process. In which, the reconstructed 3D model of the workpiece is compared to its CAD model. This scanner [90] has a 3D accuracy of 0.1 millimeter and a 3D precision of 0.2 millimeter. The price is about 19000 euro. Its dimensions are 262 $\times$ 158 $\times$ 63 millimeters and a weight of 0.9 kilogram.

Figure 2.8: Artec scanner [90]

Artec scanner is based on the structured light technique for capturing the depth values. This technique is based on the projection of a predefined pattern over the workpiece and, using different cameras, detect the projected pattern that is distorted due to the form variation of the workpiece. The distorted detected patterns are compared to calculate the depth values of the workpiece.
This scanner is more accurate than the Intel Realsense 3D cameras, but with a huge

cost difference and size that would limit the available robot work space if attached to the robot end-effector.

2.3 Force/torque sensors

After generating the working trajectory, we propose to integrate it with a force sensor and a force control system to measure the positional error and also control the contact force applied by the robotic tool during the execution of the contact-based robotic operation to keep it constant. In the experiments done, two sensors are used and are described as follows.

2.3.1 Onrobot HEX-E

Onrobot Hex-E force/torque shown in figure 2.9 is a commercial sensor made by Onrobot company. The sensor allows the application of robotic manipulators in applications requiring high level of sensitivity and accuracy like insertion and assembly tasks.
The sensor has the following technical specifications, nominal capacity of 200 N, signal noise of 0.15 N and a noise free resolution of 0.8 N.

Figure 2.9: Hex-E force/torque sensor [91]

The sensor has the dimensions of 50 x 71 x 93 millimeters that are shown in figure 2.10 and a weight of 0.347 kilograms. These values have to be considered when planning the trajectory, since the sensor limit the maximum reach of the robot, adding more length to the robotic tool used. The weight of the sensor limits the maximum weight of the robotic tool that can be used.
The wiring of the sensor is done as shown in figure 2.11. The sensor is directly connected to a dedicated compute box that is connected to the robot controller using high speed EtherCAT communication.

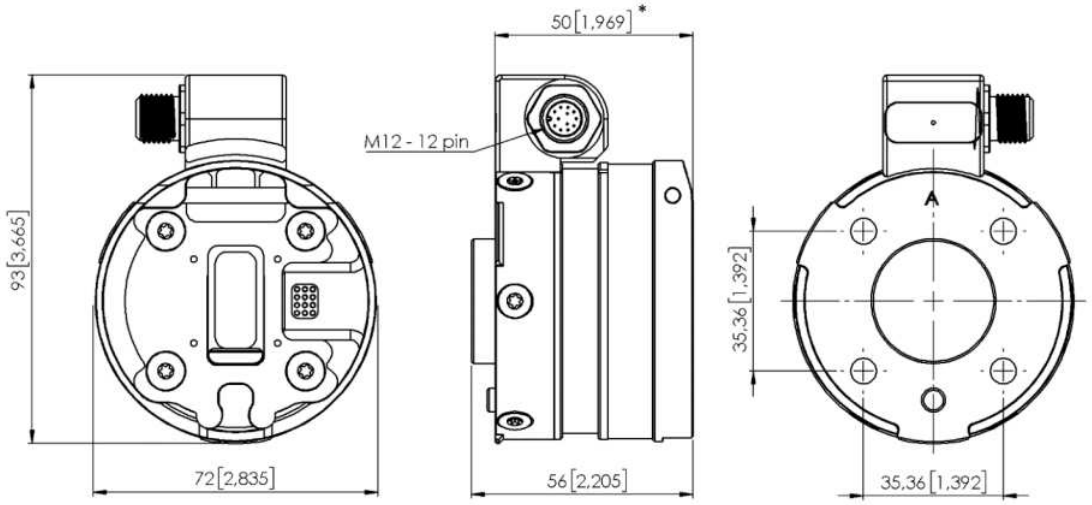


Figure 2.10: Hex-E force/torque sensor dimensions [91]

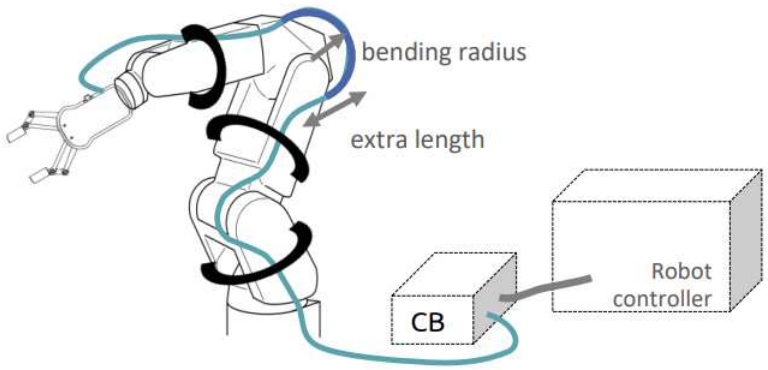


Figure 2.11: Hex-E force/torque sensor wiring [91]

2.3.2 UR5e built-in sensors

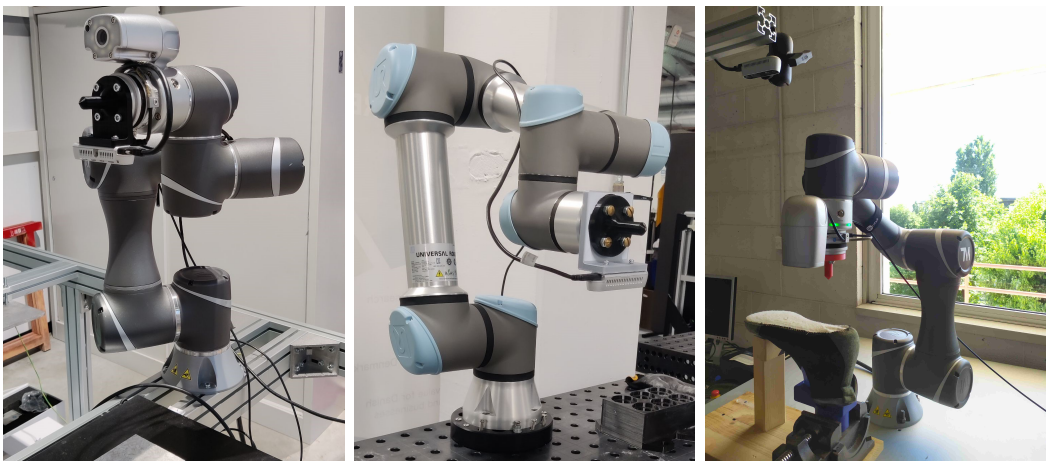
Universal Robot e-series are equipped with a built-in force sensor at its wrist. This functionality allows it to be applied in applications requiring more sensitivity while interacting with the surrounding environment. A cobot with a built-in sensor could be used in several force control-based applications without the need to add an extra and expensive force sensors. Applications in which it may be useful are like surface following, peg in hole and screw driving.
The force sensor [88, 92] in the UR5e cobot, has the following technical specifications. Range of 50 N, resolution of 2.5 N and an accuracy of 4 N. Accuracy means how close is the measured value to the real value. Resolution is the step with which the measured values change. The built-in sensor, has a lower performance with respect to external and dedicated force sensors, but in our case it is enough for the initial evaluation of the accuracy of the generated trajectory from the reconstructed 3D model.

2.4 Robotic tools

In this section, the robotic tools used during the different phases of the proposed work are introduced. Considered steps are the development of the algorithm, accuracy evaluation of the generated trajectory and performance evaluation executing the exact contact-based operation based on contact force behavior.

2.4.1 3D printed tester

Once the digital model of the workpiece is generated and elaborated to generate the working trajectory, a 3D printed tester is used for the accuracy and precision assessment of the trajectory.
Different testers, shown in the figure 2.12, are used based on the robot manipulator used. They are attached to the robot end-effector to mimic the real tool that will be used during the execution of the contact-based operation. The testers, having a simple form with the tool center point (TCP) aligned with the robot-end-effector with no tool with only an offset along the Z-axis. This allows applying the force control loop along the z-axis of the load cell.
When applying the force control along the z-axis, the accuracy evaluation of the generated trajectory can be done, comparing it with the measured feedback of the actually followed trajectory.



(a) *(b)* *(c)*

Figure 2.12: Example of testers used in the experiments

2.4.2 Glue deposition system

One of the contact-based robotic applications in which the developed solutions are applied is the glue deposition in footwear manufacturing. In this application, a Fused Deposition Modelling (FDM) extruder is used as a glue deposition system. The extruder, similar to 3D printing extruders, is made of a stepper motor that pushes the glue in the form of filament into the hot-end nozzle. This heats the glue to the melting point of 200°C.

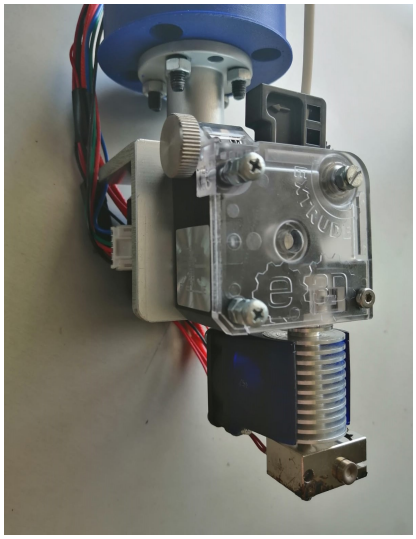


Figure 2.13: Glue deposition system

The stepper motor speed controls the quantity of the glue output. In the robotic glue deposition process. The stepper motor speed is synchronized with the robot movement linear speed. The extruder control is done using an Arduino microcontroller. To synchronize the operation of the extruder and the robot movement, TCP/IP communication is used to exchange the process parameters between the Arduino and the robot controller.

2.4.3 Onrobot sander

Onrobot sander, shown in figure 2.14, is a plug and pay robotic tool for high variety of surface finishing operations such as sanding, buffing, polishing and cleaning operations. The Onrobot sander can be easily integrated in the software programs provided by the manufacturers of the used robots, Techman robot and Universal Robot. It can be integrated with the Onrobot force sensor to control the force applied on the workpiece surface during the execution of the sanding task.
The sander has a rotation speed between 1000 rpm and, 10000 rpm. Its outer dimension, as shown in figure 2.15 are 87 x 123 x 214 millimeters and its weight is equal to 1.2 kilograms and the weight of the pad is equal to 0.1 kilograms.

Figure 2.14: Onrobot sander [55]

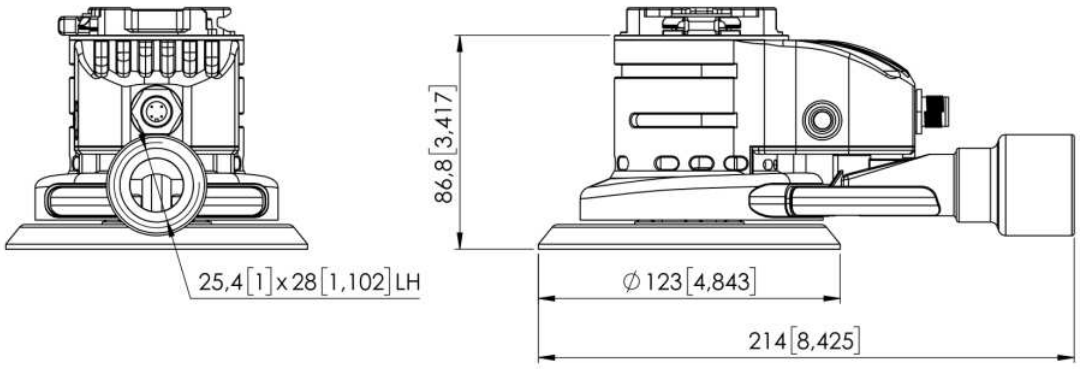


Figure 2.15: Onrobot sander dimension [55]

2.5 Software tools - open source libraries

The proposed solutions are developed mainly in Python programming language and exploit several Open-source libraries that are introduced as follows.

2.5.1 Open3D

Open3D [93] is an Open-source library for the rapid development of the of software that manages 3D data. The library provide the main features that can be used for 3D data processing such as color and depth (RGBD) images, point clouds and mesh models.
It can also be used for 3D surface alignment using local 3D registration algorithms such as Iterative closest point point-to-point and Iterative closest point point-to-plane. Also, global 3D registration are available like fast global registration and multi-way registration.
These mentioned features are the core of the developed 3D scene reconstruction

algorithm. These features are modified to be adapted to the robotic application considered, since they were initially introduced for the 3D reconstruction of indoor scenes where errors in the order of 10 or 20 millimeters are acceptable. Open3D can be used in Python or C++ programming languages.
The library has a built-in features that can easily integrate the color and depth images that are captured using the most used 3D vision sensors in robotic applications that are Intel Realsense D400 family for 3D stereo depth cameras and also Azure Kinect time of flight 3D camera.

2.5.2 Librealsense

Librealsense is a Software Development Kit (SDK) for Intel Realsense depth cameras (D400, L500 and SR300). It can be installed in different platforms such as Linux, Windows, macOS and Android. It allows changing camera configuration, changing parameters like sensors resolutions and frame rate. It is also used to manage the RGBD images streaming, applying post-processing algorithms to improve depth quality.
Built-in functions of this library are used in the developed robotic solutions. The main functions used are for camera configuration like setting images size and format. Other built-in function that is essential for the robotic application developed is the alignment function that is used to refer the depth and color images to the same physical origin since they are captured using different physical sensors.

2.5.3 Real-time Data Exchange library

The Real-time Data Exchange interface allows synchronizing external applications with the Universal Robot controller without interrupting any of its Real-time properties. The communication is done using TCP/IP communication protocol. This interface allows controlling and manipulate the robot changing the joint angles or the tool center point (TCP) pose, control of digital input and outputs and tools control. Current robot status can be read with the current state of robot registers. The output signals from RTDE interface are generated to a frequency of 125 Hz.

2.5.4 MeshLab

MeshLab [94], is an Open-Source software for processing and editing mesh models. It has several tools for cleaning, healing, inspecting, noise removal and rendering a mesh model. Some functionality for registration and alignment of two models are also available. MeshLab is available for Linux, windows and macOS. Output format can be in PLY, STL, OBJ or OFF.

In this chapter, all the tools used in the development of the proposed solutions are introduced. The tools technical specifications that may influence the performance of the developed solutions have been highlighted. In the next chapter, the methodologies to deploy these tool to reach the research objective will be shown.

Chapter 3

Methods

In this chapter will be introduced the methodologies and technical decisions made to develop the proposed solutions to automate the generation of the working trajectory of robotic contact-based operations considering unknown workpieces. The main objective of the thesis is to reconstruct the digital 3D model of the workpiece, so the main component used in the development of the proposed solution is the 3D camera.
The first topic to be covered is the camera fixing position and its relation to the robotic solution functionality. In a robotic cell, where the workpiece form is unknown, and its pose is dynamic, it is necessary to define the fixing pose of the 3D depth camera in such a way that guarantee the complete coverage of the robot workspace or the working area where the workpiece is likely to be found. For robotic contact-based operations, the most important aspect to be considered when choosing the camera fixing pose is to have in its field of view the entire workpiece or the face and the area interested by the contact-based operation. This is mandatory since the only way the robot can perceive the surrounding environment is through the use of the data acquired by the 3D depth camera. 3D cameras can be fixed in two ways, static or attached to the robot end-effector.
Secondly, camera calibration, camera preparation for data acquisition and data elaboration techniques are shown in the two different cases of fixed and moving camera. In the case of a moving camera, a 3D reconstruction algorithm is introduced, showing its three development phases. Once the digital model of the workpiece is reconstructed, it is introduced the developed algorithm for 3D model elaboration to extract and generate the trajectory.
Finally, methodologies for accuracy evaluation of the generated trajectory and the reconstructed 3D model are shown.

3.1 Static 3D camera

When the 3D camera is static, like the 3D camera shown in figure 3.1, it provides the images of the robot workspace and the workpiece from a single view point. The robotic cell in this case is designed to guarantee the optimal working conditions, exploiting all the components. Firstly, the 3D camera has to be pointed toward the robot workspace. To be satisfied also, the positioning of the workpiece in such a

way to show the area interested by the contact-based operation to the camera.

Figure 3.1: Static camera

The shown cell configuration, simplifies the integration of the 3D camera in the robotic cell since it is necessary to calculate for only one camera pose the geometric relation that relates it to the robot reference frame. This geometric relation is the homogeneous transformation matrix that transform the captured images from the camera reference frame to the robot base reference frame. The trajectory information extracted from the captured images are then sent as movement commands to the robot controller to execute the operation.
The limitation of this technique are related to the several working conditions to be satisfied. Specially, to position the workpiece in the camera field of view and to be fully reachable by the robot end-effector. When the camera is fixed, only one face of the workpiece is visible to the camera and can be considered during the contact-based operation. Other faces are invisible and can not be targeted. The other limitation is related to the necessity of repeating the calibration procedure every time the robotic cell lay out changes to calculate the new geometric relation between the camera reference frame and the robot reference frame.

3.1.1 Data acquisition and elaboration

The Intel Realsense D435 3D stereo depth camera is considered in the development of the robotic solution for the generation of the working trajectory in the robotic contact-based operation. The considered case is shown if the figure 3.2. The 3D camera is fixed on top of the robot workspace and pointed downward. The camera is used to capture continuously the workspace and to analyze the acquired images to detect the presence of the workpiece. When it is detected, the area considered in the contact-based operation is selected and the trajectory covering that area is generated to control the robot movements. The detailed steps to reach that goal are described as follows. The 3D camera, pointed toward the robot workspace, acquire

Figure 3.2: Static camera layout

color and depth images using different sensors.
The different sensors, exploited to capture color and depth images, have different positions and orientations and different intrinsic calibration parameters (e.g. width and height of video stream, principle point, focal length, lens distortion model, lens distortion coefficients), which makes it impossible to directly correlate the same pixel coordinates in the depth and RGB images. The texture mapping and alignment procedure is therefore exploited to assign to each depth point in the point cloud its relative RGB value recorded from the RGB sensor. This procedure is made in the following way:

1. De-project points of the depth frame from camera pixel coordinates (with the origin at the top left of the image) to 3D world coordinates (defined in meters with respect to the physical sensor center). This step is done using sensor intrinsic calibration parameters. The result is a point-cloud.

2. Apply to the resulting point-cloud the extrinsic matrix (made by rotation matrix R and translation vector t between depth sensor and RGB sensor) it is possible to calculate 3D coordinates of the points in the point-cloud with respect to RGB sensor reference frame (RGB sensor is becoming the new origin).

3. UV-map: project points from world coordinates to color sensor pixel coordinates (RGB sensor). This is done by using intrinsic parameters.

4. Alignment: To correlate pixels in the depth and color images, it is necessary to create an artificial view that shows how a color stream would be if it were captured from depth sensor perspective or vice versa. In this work, the depth frame is aligned to the color frame.

3.1.2 Trajectory generation

Once the coordinates of color and depth frames are defined with respect to the same reference frame, it is possible to elaborate their contained data simultaneously. The developed algorithm, done in Python programming language and using Open-source Open-CV and Librealsense libraries for image processing, consists in the following steps:
Remove background: the camera fixing position with respect to the working table on which the workpiece is positioned, the far pixels having depth values more than this threshold can be considered as background that are irrelevant for our case and can be removed by setting a distance threshold. Pixels in the background can be removed from both the depth and color images.
The remaining part of the images contains the workpiece, so a search algorithm can be developed to select only the part interested in the contact-based operation. This search algorithm can be designed based on the different cases. Examples of these search algorithms could search for parts having a particular color in the color image and from the depth frame obtain the related depth values. Also, the search algorithm could search for parts of the workpiece based on their positions, like parts closest to the camera or having the smallest depth values or parts closest or farthest from the robot base.

3.1.3 Camera calibration

After determining the pixels forming the part of the workpiece considered by the contact-based operation, it is necessary to transform the points in the camera coordinates from pixels to millimeters. This operation is done by using the camera parameters provided by the manufacturer and fetched using the dedicated function of Librealsense software developer kit (SDK). Then the path has to be defined with respect to the robot coordinate system R, since it was initially defined with respect to the camera reference frame C. In figure 3.3, the different reference frames of the camera and the robot base are shown.
A calibration procedure is introduced to calculate the homogeneous transformation TRC matrix to apply the required transformation as shown in the set-up shown in Figure 1. The pose of at least one point is calculated in both of the coordinate systems reference frames (camera and robot base) to generate the homogeneous transformation matrix (rotation matrix and the translation vector between the two reference frames). The point used in the calibration process is a point with different color placed on the shoe surface, whose pose with respect to the camera reference frame is detected by a color search based algorithm. To define the coordinates of the same black point with respect to the robot base reference frame, the robot is moved manually until a pointed tester used as the robot's end-effector reaches the point position.
The calibration procedure is commonly done through an optimization procedure on a more numerous subset of points and usually using a chessboard attached to the robot end-effector. In that alternative procedure, the calibration procedure consists in moving the chessboard in different poses. For each pose, save an image of the chessboard and the chessboard pose with respect to the robot base. Using Com-

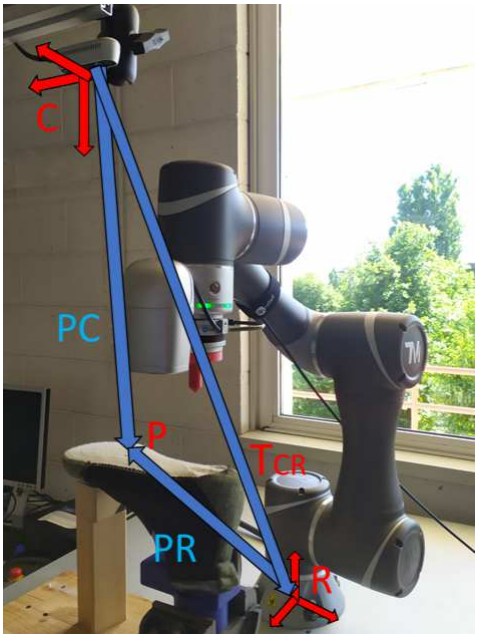


Figure 3.3: Static camera calibration

puter vision algorithms it is possible to calculate the position of every corner in every image with respect to the camera reference frame and as consequence calculate the chessboard pose with respect to the camera reference frame. knowing for every image the relation between camera and chessboard and the relation between the chessboard and the base, it is possible to find the optimal homogeneous matrix to relate the camera pose to the robot base reference frame, minimizing the errors in the different images.
For the considered case the first calibration procedure is the one used and combining the resulted trajectory with force control allowed to compensate for the possible errors as will be shown in the following 4.3 chapter.

3.2 Moving 3D camera

A more flexible solution to integrate the 3D camera in the robotic cell consists in attaching the 3D camera to the robot end-effector, as shown in the figure 3.4. In this way, thanks to the multiple view images, it is possible to cover the robot workspace or the area where the workpiece is most likely to be found. This allows also to move the 3D camera around the workpiece that means that several faces of the workpiece are visible and could be considered to apply the robotic contact-based operation.
This solution gives the robotic cell more flexibility since it removes the constraints on the positioning of the workpiece to point the interested face towards the 3D camera and allows working on multiple faces of the workpiece. Other advantage of this technique consists in the necessity to calibrate the camera to calculate the geometric relation between the camera frame and the robot base frame only once. The calculated homogeneous matrix can be used for referring all the images captured by the 3D camera in all the possible poses.

Figure 3.4: Moving camera

This technique represents some limitations like the necessity to design accurately the interface to attach the 3D camera to the robot end-effector to allow also the attachment, at the same time, of the robotic tool needed to execute the robotic contact-based operation. The robotic tools are mostly to be bulky, and the 3D camera has to have always a free field of view.

3.2.1 Camera configuration and images acquisition

The 3D camera when attached to the robot end-effector enhance the flexibility of the robot cell, allowing to scan the different faces of the workpiece and then generate the working trajectories to have contact with these faces.
The main goal of this section is to introduce the steps and methodologies followed to exploit the moving 3D camera to capture several images of the workpiece to generate an overall digital copy of it. This digital copy is used for to generate the working trajectory for the robotic contact-based operation.
To reach this goal, three development steps are followed. The first step is to integrate images captured by a manually moved 3D camera to reconstruct the 3D model of the workpiece, this solution is based on the estimation of camera movement using computer vision techniques. The second step consists of the integration of the developed algorithm in the first step in a robotic application. The final step consists of the optimization of the developed algorithm to improve its performance and computational time, exploiting the knowledge of the robot poses.
Sensor setup and data acquisition is the first step in the three development steps. The vision sensor used, Realsense D415 camera captures color and depth images using different sensors and allows configuring the image quality based on user's

preferences. In the sensor data sheet [89], the sensors can be configured to resolutions up to 1280x720 pixels for the depth sensors and 1920x1080 pixels for the color sensor. Sensors resolution have a significant effect on the output accuracy and optimal configuration parameters for the considered case as discussed in section 5.2.8. At 1280 x 720 pixels resolution configuration, the sensor can capture images at frame rate up to 30 frames per second (fps).
Color and depth images are captured by different sensors and could have different resolutions. So the first step is to transform the two kinds of images to have the same characteristics. The two images are aligned to have the same exact size and to be geometrically referenced to the same physical sensor origin. The alignment procedure, explained in details in our previous section and in [95], allows to align depth to color or vice versa.
To acquire the images that allow for the 3D construction of the object, all parts of the workpiece have to be captured in the different images. To do that, the acquisition movements are planned to guarantee the coverage of almost all the working place where the object or at least the interested area of the workpiece is mostly to be found and always within the 3D camera field of view.

3.2.2 3D reconstruction using manually moved 3D camera

The technique used, is based on matching the images, comparing the overlapped parts in them. To guarantee that, the frame rate or the number of the images that the camera can capture at every second and the camera movement have to related to have the time necessary to capture the images covering all the parts of the object without large movement between sequential images.
In this considered step of the algorithm development, the user moves the 3D stereo depth camera around the object to be scanned and capture RGB-D images of all the interested areas.
It is necessary to satisfy some requirements while executing the scanning process. The requirements are to guarantee that the scanned workpiece is within the camera field of view, and it is visible if possible in all the captured images. The other important requirement to satisfy is related to the 3D camera minimum distance with respect to the workpiece.
The color and depth images contain different information of the observed scene as they have been captured from different view angles. To create a unique 3D mesh model, it is necessary to put the images in order, refer each image to a common reference frame and then integrate the information contained in the images together to create the 3D model. This is done as follows:

Images matching and camera motion estimation

To calculate the motion between the camera poses capturing the RGB-D images of the observed scene, RGB-D odometry technique [96] is used. This technique, comparing two images captured by a moving camera of a static scene, determines the homogeneous transformation matrix that if applied to the second image maps it to match the first one.
RGB-D odometry technique used is the one introduced in [97]. It determines the

camera motion or the homogeneous transformation matrix by solving an optimization problem that maximizes pixels photo-consistency [98, 99] (if the same pixel is visible in two images it has to be represented by the same color and brightness in both of the images) [100] and minimizing the least square error of pixel positions.
Like most of Iterative Closest Point registration algorithms, the one used, needs an approximate initial value [101] of the transformation matrix. This initial value is usually obtained by a global coarse registration algorithms based on the point-cloud features. In our case, assuming small motion between the images and to have a big portion of the object always visible in all the images allows using as initial value of the homogeneous transformation matrix the identity matrix.
The camera reference frame while capturing the first RGB-D image of the dataset is considered as the common reference frame to which all the images have to be referred. The RGB-D Odometry algorithm calculates the camera motion between only two images, if the two images do not have enough overlapped pixels this may cause low accuracy calculating the motions.
Comparing all the images of the dataset to the first image, may cause errors in case of small or missing overlapped parts. To overcome this possible error, every image is compared to the consecutive image and to randomly picked images from the dataset. In this way, different motions are obtained for every image. In case of missing overlapped parts between a specific image and the first one, it is possible to construct the cumulative motions considering intermediate motions. These intermediate motions are chosen, maximizing pixel overlapping, to find the right sequence to be able to refer the considered image with the first one.

Volume integration

To create the 3D mesh model of the observed scene, the pose graph or camera motion obtained is used while integrating the information contained in every RGB-D images. The volume integration process is introduced in [102]. It consists in the integration of color and depth information of the observed pixels of each RGB-D image to construct a unique voxel grid representing the scene. The voxel grid is a three-dimensional representation of the information contained in the two-dimensional color and depth images. For every pixel, a related cell in the grid is identified based on the pixel position and depth. Then this cell is updated by its color information. After elaborating all the images to construct the grid, the full cells are those related to the observed scene while the pixels between the camera and the scene and those behind the scene remain empty. The following step is to calculate the so-called isosurface. The isosurface is the smooth and continuous surface interpolating the not empty voxels in the grid.
While integrating the several RGB-D images in one volume, it may happen that the voxels visible in several images have different positions. That is for example because of errors in the alignment or sensor's noise. To overcome this problem, Truncated Signed Distance Function (TSDF) technique [102] is used.
TSDF function is used for averaging the values of voxels positions from the different images based on two variables. The first is the signed distance function representing the distance of each point to the nearest surface along the line of sight to the sensor. The other parameter is the weight, if the voxel normal is along the line of sight pro-

jected from the sensor while capturing the RGB-D image, from the pose calculated in 5.2.3, its weight is maximum, and it decreases if the inclination between them increases. In this way, the voxel position value is dependent on the value in the RGB-D images in which it is clearly visible. In figure 3.5 it is shown an example for the integration of two images to construct a unique surface. The whole surface is not visible in both of the images. The average of depth values from the two images is done, giving more weight for the values from the image on the left for the points having the normal equal to the vector n1 and less weight for the values from the images in the middle. The same is done, giving more weight for the values in the image in the middle for the points with the normal vector n1. The reconstructed surface is the one on the left.

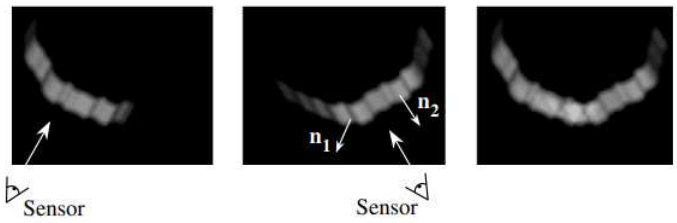


Figure 3.5: Weights based on the relation between sensor line of sight and normals [102]

The weighted TSDF functions for all the images are combined to construct the overall voxel grid. From the grid, the isosurface that is the continuous surface connecting the voxels together is estimated. In figure 3.6, an example of the truncated signed function calculated for two frames are shown in the left and the middle images. After integrating the values from the two frames, the isosurface for the reconstructed surface is shown with the dotted line in the image on the left side.

Figure 3.6: Isosurface estimation [102]

Fragments integration and scene construction

If the scene observed is big and the number of RGB-D images to integrate is high, it is necessary to partition the dataset in several parts. This allows for error minimization when integrating the images in a global space. For every partition of the dataset, a point-cloud is created to represent a fragment of the scene through RGB-D odometry 5.2.3 and volume integration in 5.2.4.
After the construction of the fragments covering several observed parts of the scene, it is necessary to refer all fragments to a global reference frame and to integrate them together to construct a complete 3D model of the observed scene as explained in [83]. The fragments are covering different parts of the scene, integrating them

considering only the spatial order of the poses calculated using RGB-D odometry may lead to registration error in the construction phase. A better way for integrating them is based on finding the optimal sequence, maximizing the overlapped areas between fragments. To do this, several registration processes between each fragment and the rest of fragments is done to find the best adjacent candidates (previous and following fragments).
Once all possible consequences (that may have many false positives) of the fragments are obtained, the candidate are considered to apply a dense global surface registration to find the configuration with best match.
The best fragments order and RGB-D odometry are used to find the best camera motion between the observation of every fragment. This information is then used to fuse all the color and depth information from the RGB-D dataset into a unique 3D model of the scene using TSDF volume integration.
The considered workpieces in this paper are relatively small and limited number of images are enough to cover all their faces with limited number of images (e.g. less than 150 images). It is not necessary to partition the dataset and construct several fragments.

3.2.3 Odometry-based 3D reconstruction using an automatically moved camera

To acquire the images that allow for the 3D construction of the object, all parts of the workpiece have to be captured in the different images. To do that, the acquisition movements are planned to guarantee the coverage of almost all the working place where the object or at least the interested area of the workpiece is mostly to be found and always within the 3D camera field of view.
The images are captured during the movement of the 3D camera, to guarantee stable and high quality images, the sensor's frame rate is decreased to 10 frames per second and the scanning trajectory is followed by a moderate speed. More in details explanation about the tests and parameters calculation are shown in section 5.2.2. In figure 5.6 the scanning trajectory is shown with the camera poses while capturing the images. It is also shown the output point cloud of the working table.
The acquisition movements start by capturing a from top image of the working table to capture global information of the object. Then the robot moves the camera to capture images of the different faces of the object. The test shown in the figure is done, acquiring 100 pairs of synchronized color and depth images of the working table in front of the robot.
Once the acquisition movements are executed and the data are acquired, the image matching procedure 5.2.3 and the integration procedure 5.2.4 mentioned before are followed to integrate the color and depth pairs of images to reconstruct the 3D model of the scene covered in the images.

3.2.4 Robot-based 3D reconstruction using an automatically moved camera

In the robot poses technique, the images are captured only with correspondence to a specific poses that cover specific points of the workpiece. The robot when arrives to the defined poses, it stops for 500 ms to allow the 3D camera to capture the RGB-D images. The minimum number of the images is 5 that allows to cover the workpiece from all the four sides and also with an extra top view image. Knowing exactly the pose of the robot end-effector and hence the pose of the camera while capturing a certain image, allows for the reconstruction of the motion and to calculate the homogeneous transformation matrices for each image to refer them to a common reference frame. In this way, the Odometry process for camera pose estimation is not necessary that decrease significantly the time necessary for the elaboration. The calculated homogeneous transformation matrices are then used to integrate the information using TSDF, as explained in volume integration 5.2.4.

3.2.5 Point cloud elaboration and trajectory generation

The result obtained from the integration process is the 3D model of the observed scene as a mesh model and a point cloud. In the developed algorithm, the point cloud representation is used.
Assuming that the object is placed in the working area that is reachable by the robot and referring the point cloud with respect to the camera origin pose, capturing the first image. The point cloud can be trimmed, removing far points from the camera representing unrelated background points (i.e. further than 1.5 meters).
In the remaining part of the point cloud, clusters of points forming a plane are found (neighboring points having depth in a defined range, i.e. 2 mm). The points forming the working table on which the workpiece is placed can be found, searching for the biggest cluster of the planes found. Removing these points allows removing the working table, leaving only the workpiece considered by the operation.
Different subroutines are developed to select the part of the workpiece considered by the operation. If the operation is targeting the center of the workpiece upper face, it is possible to select all the points of the object having the lowest depth values and from them select the central part defining offsets from the points forming the borders. In other cases, when the operation, targets the edges of the workpiece. For example, in polishing operations, the edges can be found elaborating the normals of the workpiece point-cloud and selecting the area having neighboring points with huge variation in the normals directions. Having developed the integration algorithm based on the elaboration of color and depth images, the 3D model of the scene preserves also the color information. This process also allows selecting parts of the object, searching for area having specific color. This is useful in case of surface finishing and painting.
The selected portion of the workpiece is made by high number of close points, before the generation of the working trajectory it is necessary to down sample the point cloud. This allows to remove close points and have a minimum distance between the consecutive points that helps the robot controller to interpolate better the movement commands and in the applications considered to have good contact force control.

As mentioned before, the point cloud is referred to the camera origin at the first image capturing position and so the points forming the selected part of the object. Knowing that initial pose with respect to the robot's end-effector, the homogeneous transformation matrix to transform the points coordinates and orientations from the camera origin to be with respect to the end-effector is found.
A final homogeneous transformation matrix is found to refer transform the points forming the trajectory from the end-effector to be referred to the tool center point (TCP) of the tool used to execute the contact-based robotic operation.
Based on the operation to be executed, the order in which the selected points are followed can be changed and parameters like movement speed and contact force have to be defined to generate the trajectory that guarantees a good execution performance.

3.3 Performance evaluation metrics

The output of the developed solutions can be divided in two categories. The first, is the 3D mesh model of the workpiece and the second output is the trajectory to be followed by the robotic contact-based tool. In this section, the evaluation metrics used to evaluate the accuracy of both of the outputs are described.

3.3.1 Accuracy of the generated 3D model

The main goal of this work is to generate the trajectory to be followed by the robot. The exact digital copy of the workpiece is not available and to have an accurate trajectory an exact 3D reconstructed model is needed to generate the trajectory.

Considered evaluation cases

- Comparison between the reconstructed 3D model and CAD model: the evaluation procedure consists of the comparison of the reconstructed 3D model with the exact CAD model of the workpiece. For the evaluation, workpiece of which the CAD model is available are considered.

- Comparison between the reconstructed 3D models using different reconstruction techniques: in the cases in which the robotic manipulator is used, two different techniques are proposed to reconstruct the 3D model of the workpiece. The first is the Odometry-based technique in section 3.2.3 that exploits computer vision and feature matching to estimate the camera pose to capture each image and use these poses for images integration in one 3D model. The second is the Robot-based technique explained in 3.2.4 uses directly the known poses of the robot and hence the poses of the 3D camera to capture each image. For accuracy assessment of both of the techniques, the two 3D model reconstructed by the two techniques are compared with the exact 3D CAD model of the workpiece.

- Comparison between the reconstructed 3D models and a reconstructed 3D model using a professional structured light scanner: to further evaluate the

accuracy of the reconstructed 3D model using the two techniques, a professional structured light-based scanner is used to reconstruct the 3D model of the workpiece. The reconstructed 3D models by the scanner and the 3D scanner are then compared to the 3D CAD model of the workpiece

Registration of two 3D models

Considering the three comparison cases previously explained, in which a comparison is done between two point clouds or between a point-cloud and CAD model. A good comparison accuracy depends on the ability to compare an exact point in the scanned point cloud to the exact relative point in the 3D model. To guarantee good comparison accuracy, the two point clouds have to be aligned.
Alignment of point clouds, known as point cloud registration process, is a well known problem in computer vision and widely used in the construction to compare and inspect buildings, roads and civil infrastructures using laser scanners or similar scanning techniques. 3D registration process consists in finding the transformation matrix that if applied to the second model align it to the first one and also refer both of the models with respect to the same reference coordinate system.
Different registration techniques can be applied that can be categorized in coarse and fine registrations [103]. Coarse registration are feature based techniques that match the features in the two point clouds. The most used strategies are classified in point-based, line-based and surface-based. These methods are very accurate but highly dependent on the similarities level of the two point clouds that is dependent on the sensor noise, scanning point of view, point clouds density and geometrical nonconformity due to fabrication process accuracy. To guarantee high accuracy, complicated feature extraction tools may be necessary to develop.
Fine registration techniques are based on approximate iterative processes to find the optimal rigid transformation matrix between the two point clouds. The most used technique is Iterative Closest Points ICP [101, 104] and its variations. These techniques are based on the minimization of positional errors of relative point sets selected from the two point clouds. This implies the necessity for a good initial guess of the rigid body transformation to avoid local minimum problems. Elaboration time is dependent on the point-cloud size.
In the case of 3D printed objects considered later in this thesis, using the above techniques leads to registration failure. This because of the low geometrical conformity of the printed object due to printing accuracy tolerance. Also, difficulties to identify features like lines or surface in the noisy scanned point-cloud. And finally, lack of initial transformation guess to be used for ICP techniques as the scanning process is done manually.
To overcome these limitations, we propose an easy and fast to apply registration technique to align the two point clouds based on more general features considering the overall point clouds. These features are the boundary box and the three-dimensional center of the object.

Hausdorff Distance for error measurement

To quantify the error between two 3D models. In this case, the comparison will be done between the reconstructed 3D model and CAD model. Hausdorff distance [105, 106] is used for the execution of this comparison and for error quantification. The Hausdorff Distance is a parameter usually used for measuring the degree of similarity between two overlapped and aligned images, point clouds or sampled 3D mesh models [107].
For two data sets $A = \{a_1, a_2, ..., a_n\}$ and $B = \{b_1, b_2, ..., b_n\}$, Hausdorff Distance $d_h(A, B)$ is given by evaluating for every point in A the distance with respect to the nearest point in B. The maximum of these distances, between correspondent nearest points, represent the Hausdorff distance, and it is given by 3.1. That mean that all points of the first point's set have distances with in the calculated $d_h(A, B)$. The maximum distance may be different based on the data set considered as a reference, so the maximum between $d_h(A, B)$ and $d_h(B, A)$ is considered as shown in 3.2.

$$d_h(A, B) = \max_{a \in A} \min_{b \in B} \|(a - b)\|^2 \tag{3.1}$$

$$D_h = max(d_h(A, B), (d_h(B, A)) \tag{3.2}$$

3.3.2 Accuracy of the generated trajectory

From the reconstructed 3D model, the area considered by the contact-based operation is selected. The reconstructed 3D model is transformed to a point-cloud for the trajectory generation. Putting the selected points in order or following a certain pattern allows defining the points order that is sent to the robot controller. For evaluating the accuracy of the generated trajectory, two aspects relevant to almost any robotic contact-based operation are evaluated.

Depth accuracy evaluation using touch stop technique

It is very important that the selected points from the reconstructed 3D model and the physical points that will be touched by the robot end-effector are perfectly coinciding. This to guarantee a good performance of executing the contact-based operation. In some contact-based operations like grinding or deburring, a rotating tool touches the workpiece at high speed and exerting high and reasonable contact force has to be positioned exactly over the area considered by the operation to avoid the damage to the workpiece.
The 3D camera used, Intel Realsense D415, have major measurement error in the calculation of the depth values of the pixels combining the image. This due to the use of two imagers with a relatively low resolution. The device depth error provided by the manufacturer in the datasheet [89] is lower than or equal to 2% of the distance between the sensor and the observed surface. Assuming that the minimum working distance at the resolution of 1280 x 720 pixels of 450 millimeters is the one used to capture an RGBD image. The error value would be lower than 9 millimeters, that is quite high for an accurate robotic task as the one considered in this work.
To evaluate the real depth values of the generated trajectory, the poses to be followed

by the robot considering the 3D printed tester introduced before are calculated. The robot is controlled to reach every single point pf the trajectory separately. Using force sensors, the built-in and the Onrobot hex-e force sensor based on the considered case, a touch stop function is used to reach the workpiece and stop as soon it touches it. The values measured using this method are compared to the points extracted from the 3D model to calculate the positional error between them.

Contact force evaluation following the generated trajectory

One of the most important key performance indices to evaluate the quality of a contact-based operation is the stability of the exerted contact force during the execution of the operation. This value is important to evaluate the consistency of the robotic solution to execute the task, keeping always the quality of the output workpiece as constant as possible.
The trajectory is calculated considering the 3D printed tester. In this case, the trajectory is followed entirely. Using the force sensor, built-in or the Onrobot sensor, The trajectory is followed by the robot in force control mode along the z-axis of the robot end-effector reference frame. In this mode, the exerted contact force applied perpendicularly over the surface of the workpiece can be controlled setting a specific set-point. For the application considered, a force value between 3 and 10 N are used. Recording the feedback values measured by the force control loop allows measuring and analyze the contact force behavior during the trajectory following process.
Measuring the feedback position of the 3D printed tester make it possible to further evaluate the depth values of the generated trajectory.

3.4 Force control for trajectory online adjustment

The force sensors and the different functionalities provided by the force control loop can be used for error compensation, besides the error and accuracy evaluation of the generated trajectory.
The errors measured comparing the generated trajectory and the actual followed one applying force control are generated to many factors. These factors are the errors due to the accuracy of the 3D camera calculating the depth. Positional error can also be caused by the low durability of the workpiece in case of wooden objects. Or due to manufacturing errors, especially in textile made workpieces such as shoes.
The calculated error is used for the adjustment of the generated trajectory. This is made following two ways. The first using the measured errors using force stop and to adjust the all the points separately. The second way is to save the feedback position of the 3d printed tester applying the desired contact force and then use it as the working trajectory.
Applying the force control to the adjusted trajectory allows applying further online adjustment of the trajectory that stabilize more the contact force values during the execution of the robotic contact-based operation.

3.5 Discussion

In this chapter, the methods and the developed algorithms are introduced to show how the tools introduced before can be used to be able to generate the trajectory to be followed by the robot-end effector to execute a contact-based operation. The proposed methods not require any digital model of the workpiece. Exploiting a low-cost 3D stereo depth camera, it is possible to reconstruct the 3D model of the workpiece and extract the trajectory following the area considered by the contact-based robotic operation. Using external force sensors allows evaluating the accuracy of the proposed methods to generate the working trajectory and to compensate offline and online for positional errors.
Using the proposed methods, a contact-based robotic operation can be automated without the need for time-consuming CAD elaboration procedure. This is necessary in small and medium enterprises, where the production volume is small and highly customizable, the CAD model of the actual built workpiece may be unavailable or not identical.
In SMEs, the collaborative working environment is highly dynamic, and the used tools need to be flexible and easy to configure for tasks automation. So, the proposed robotic solutions aim to simplify and automate the trajectory generation procedure without any human input. Setting some predefined parameters such as the scanning trajectory, the characteristics of the area considered by the contact-based operation and the contact force to be applied allows to automate entirely the process of trajectory generation and execution without workpiece digital model and knowledge of its exact fixing position.

Chapter 4

CAD-based trajectory planning and feasibility study of using low cost 3D camera in contact-based applications

In this chapter, a traditional technique for trajectory planning of robotic contact-based operations is used to identify the parameters and the requirements to obtain a good performing contact-based operations. The technique is based on the use of the digital model of the workpiece in the form of CAD model. The CAD model is elaborated to extract the trajectory to be followed by the robot end-effector during the execution of the contact-based operation.
A solution based on the use of a fixed 3D camera is introduced for trajectory planning of contact-based operations. A low-cost 3D camera is used for the trajectory generation. Hence, a feasibility study is done to evaluate the feasibility of using such cameras in applications that are always done using more accurate and much expensive laser scanners. The performance evaluation of this solution is done considering the requirements identified applying the technique based on the use of the workpiece CAD model.
The solution based on the use of CAD model and the one based on the use of the low-cost 3D camera are evaluated in an application of glue deposition to automate a step of the production chain of footwear manufacturing.

4.1 Industrial context

As many others, the footwear industry undergone the industrial revolutions and mechanization. The production, from being traditionally a handicraft process, assumed in the last century a different dimension. Despite the mass production has completely smashed the numbers and the required time for a shoe to be produced is incomparable with the craftsmanship made one, the latter resist by offering often a higher quality and a different attention to details. The craftsmanship's production is still a lot more flexible, in diversity of the product as well as in materials usage. The reported characteristics have split the market into the volume based one, of the

mass production, and on the other side the craft based one, which best fits in luxury sector as well as in the customized prototypes.
The Italian footwear is the most traditional manufacturing industry of the country and represents the most imposing European shoe exporter. In [108] have been reported three major prerequisites for the shoe industry to become a mass customized one, these required capabilities are:

- design of shoe lasts;
- adaptability of aesthetic design;
- manufacturing and information management.

Customization demands a series of improvements in all the phases of production. The manufacturing stage is a very important one to upgrade, without losing flexibility and agility. To this end, new machines and organization are needed in the parts' production processes as well as in the assembly phase. These aspects are going to be analyzed in the next section.
The production of a shoe begins with the supply of materials, including both row materials and semifinished components. Before start the manufacturing, the materials have to be inspected and modified in order to satisfy the quality requirements. Often, the upper part, the lower components and the rest of the necessary elements are manufactured separately since different constructive methods have to be used. Cutting, machining and pre-stitching operations are executed on the components before the final product is assembled. To produce a shoe requires a great deal of personal experience and workmanship, even though most of the process is assisted by more or less sophisticated machines. The completed lower and upper parts are united using different techniques, which depend on the type of shoe to be produced as well as the machinery and technology available to carry out the job. Another important factor is the producer's choice, since the quality and the production time constraints are characterized by the required standards. Usually, the upper is stretched over a last (a fixture that represent the shape of the foot, Fig. 4.2) and attached to the bottom through a process called lasting [109]. The assembling process or lasting, as reported by [110], can be summarized in six main steps:

1. Last and upper preparation:
   The upper initial design may be performed directly on the last or on a standard shape (the flattened upper) following the fashion designer drawings. This shape is lately scaled to provide all the foreseen sizes. The next step is the cutting out the leather, or any material is going to be used, manually or automatically. In many cases is still used a socket punch which requires templates, these have to be designed in an additional stage, but offer a great advantage in case of a big production. To avoid this time-consuming practice and save some flexibility, it is possible to rely on a machine similar to a plotter with several cutting heads. The actual process may be slower than the cut with the socket punch, but the flexibility is preserved, while the process is way faster than manually done. The technology to chose depends on the entity of the production, economic studies in every particular case are advisable. Additional operations could be required to finish the upper.

At this point the upper part is ready to be attached to the bottom and preparing operations on the insole are performed. The inner part of the sole is temporarily fastened to the last, manually or through use of semiautomatic machines, and the upper can be now modelled onto the same last to give the right shape to the shoe.

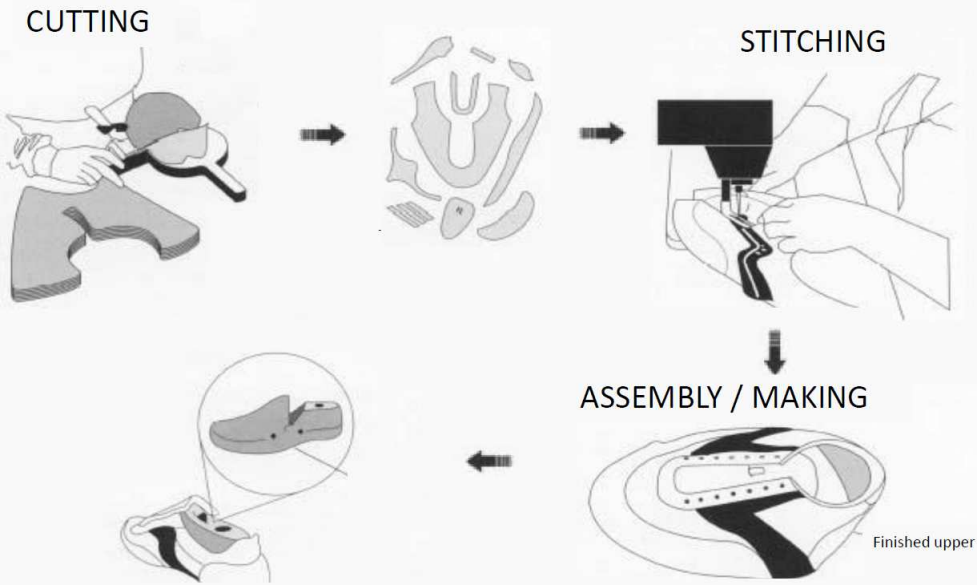


Figure 4.1: Phases of preparation of the upper

2. Assembling of the upper on the last:
   The upper part is modelled on the last and fastened to the insole through use of glue, nails or stitching. The exceeding material is cut away, and decorative operations may be done. A quality control is performed to ensure the correct coupling.

Figure 4.2: Shoe design and its components [110]

3. Heat treatment:
   To let the shoe assume the desired shape, guided by the last, it would be necessary to leave it rest for some time, which depends on the used material. Considering the necessity, it is possible to use a heat treatment to speed up the process.

4. Bottom and sole preparation:
   It would be better to have a rough surface onto which to apply the layer of glue, to this end, preparatory operations could be necessary both for the insole and for the bottom. These operations are usually performed through use of semiautomatic machines, what means an operator is required.

5. Sole fastening:
   This stage is a delicate one since it directly influences the quality of the product. The upper part is positioned onto the sole very carefully, and closing force is applied during the necessary period to couple the components. Even if the application of the force is performed using a machine, the initial phase of coupling is often done manually.

6. Last removal and finishing:
   Once all the previous operations have been completed, it is possible to remove the last, add the heel and perform the finishing operations. A quality check is executed on the final product before delivering it to the packaging department.

The whole manufacturing and assembling process has to be adjusted to the type and the model of the shoe is going to be produced. The rapidly changing environment, into which the factories are immersed, force the production to keep adapting to the volatile demand of the market. Therefore, it is needful to organize and manage

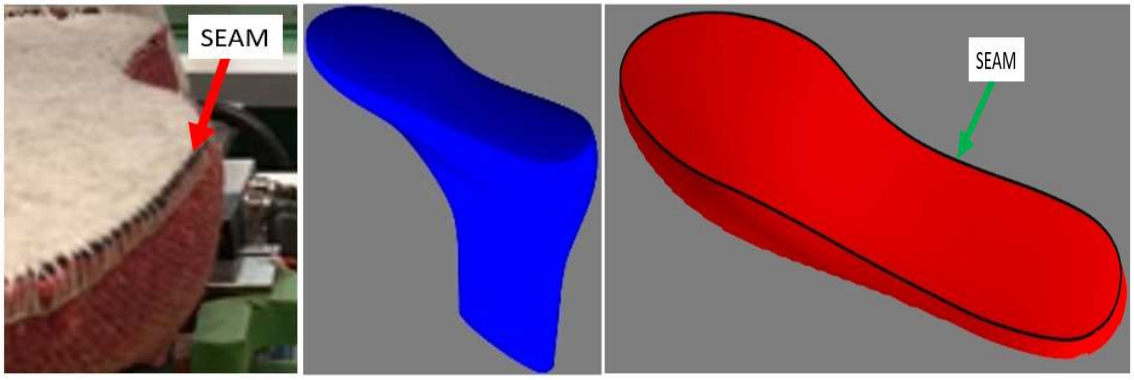
optimally the production policy in order to let the scheduling be able to respond satisfactory.

In [111] an analysis of the manufacturing factory is conducted through use of simulations of the processes to recognize the optimal scheduling and discover bottlenecks. To increase the production rate is taken in account the variety of shoes models. Flow time and cycle time were taken in account and a bottleneck was recognized in the station "gluing uppers and sole" due to the operation and waiting time of the latter. The previously taken in account considerations about the direction the industry is taking, more flexibility is required, according to [108, 112, 113].
This section is aimed to recognize the limitations of the cementing process and provide solutions focused onto improve the production through use of new available technologies.

4.2 CAD-based trajectory planning

For complicated workpieces, the geometric path identification is crucial for the success of the task. The techniques used for that are divided based on the workpiece digital model used for the process.
Mesh models are available in almost all the CAD design software kits. They are the most used digital format for layered manufacturing and additive manufacturing. In additive manufacturing, the whole object in form of STL format for example is sliced in layers and the layers are divided in points. The number of the overall points to be followed is usually huge, and the material deposition is done point by point, so it is important to minimize the cycle time and to optimize the coverage of the surface without skipping any point.
An example of the automatic tool path generation process in contact-based operation like glue deposition in footwear manufacturing [114, 115] based on the use of mesh models is shown. The process consists of the development of a slicing software to generate the geometric path to be followed.
The slicing software performs its calculation on an input CAD model of the shoe upper geometry, in the format of an STL file, which must be available from the shoe manufacturer. Both ASCII or binary STL files can be directly loaded in a Matlab procedure (*stlread* function), or Python procedure (*numpy-stl* library, *PyPi* repository), which return the mesh in terms of connectivity list and vertices. Normal vectors can be also obtained from the built-in functions, or directly computed from the definition of each triangle of the mesh. The loaded data then undergo a preprocessing routine to repair possible defects in the model (e.g. gaps, mixed normal, overlaps), and to reduce the size of the stored variables by eliminating repetitions of the same points (i.e. shared ordinates) defining the mesh. As an example, for a starting geometry of 104338 triangles defining a mesh of a shoe upper (corresponding to 313014 points), only 52169 points unequivocally describe the related geometry. As a rule of thumb, the point to describe the desired geometry are reduced by one sixth.
A further viable mean to reduce the computational costs of the path generation consists in cutting the geometry under analysis. In the case of the shoe upper, only the portion of the upper where the glue is to be deposited can be selected. This

region is identified as the area of interaction between the upper and the sole in the final shoe. A graphical representation of the input and output of this step can be observed in Fig.4.3, representing the original picture, the model of the shoe upper and the selected portion where the glue has to be placed. The figure also represents the position of the seam.



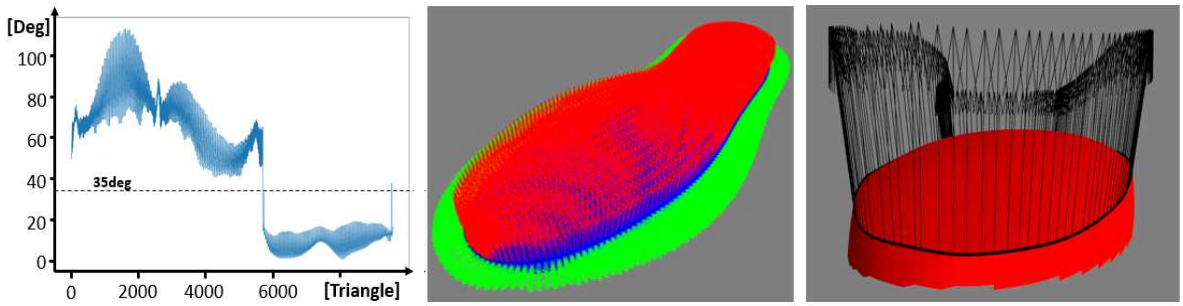
(a) *(b)* *(c)*

Figure 4.3: a) Shoe upper. b) STL file of shoe upper. c) Reference glue deposition trace.

The slicing algorithm uses the position of the seam as a reference contour for the evaluation of the trajectories where the glue has to be deposited, the seam position being directly computed from the STL file in the slicing software.
As it can be observed in the upper shape reported in Fig.4.3, the seam correspond to an abrupt variation in the vector normal to the surface. Its points can be therefore found by identifying the mesh triangles where a sudden variation of normal vector n_i direction occurs.
For each triangle, the angular distance between the normal vector n_i and the unit vector defining the vertical reference direction n_{ideal} can then be computed using the inner product

$$\theta_i = \arccos(n_i \cdot n_{ideal})$$

Fig.4.4a represents the trend of the θ angle obtained for the shoe upper considered in the present work. Two groups of angles can be identified, one in the range from 50^o to 100^o, corresponding to the side surface of the shoe upper, and a second one in the range from 0 to 20^o, corresponding to the top surface of the shoe upper. A threshold value to discern the two groups of points can be defined (e.g.35^o) to separate the mesh triangles belonging to the top surface (red normal vectors in Fig.4.4b) from those belonging to the side surface (green normal vectors in Fig.4.4b).



(a) *(b)* *(c)*

Figure 4.4: Seam evaluation algorithm.

The actual points defining the seam contour can then be identified by intersecting the model of the shoe upper with a suitable cutting surface (defined by a list of points and relative connectivity list). The generation of this cutting surface starts with the projection of the triangles belonging to the top surface cluster (characterized by the red arrows in Fig.4.4b) onto a plane parallel to the X-Y plane (whose normal vector coincides with n_{ideal}). This plane corresponds to the lower base of the cutting surface, and it is defined at a height Z_{lower} equal to:

$$Z_{lower} = min(P_z) - \epsilon$$

having indicated with P_z the vector of the Z coordinates of all the points belonging to the shoe upper and being ϵ a safety margin used to guarantee the intersection. Then an alpha-shapes based algorithm [116] is exploited to get only the boundary points (i.e. the seam) that functionally serves as base for the cutting surface. In order to characterize the cutting mesh, the base is extruded along the n_{ideal} direction by creating a duplicate of these points on a plane at Z equal to:

$$Z_{top} = max(P_z) + \epsilon$$

The connectivity list is reconstructed using a counter j belonging to [1, Np), where Np is the number of points constituting the lower contour. For each j, two triangles are demarcated:

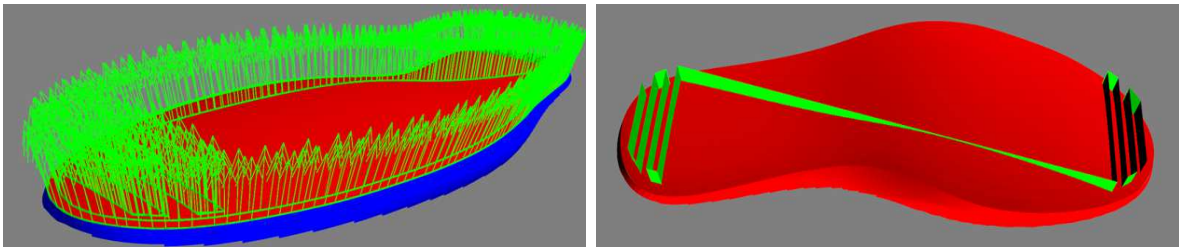
$$Triangle_1 = [\ Lower(j),\ Lower(j+1),\ Top(j)\ ]$$

$$Triangle_2 = [\ Lower(j+1),\ Top(j+1),\ Top(j)\ ]$$

where the *Lower* and *Top* labels indicate the points belonging to the lower and upper base respectively.
By intersecting the shoe model with the cutting surface just generated, the actual profile of the seam is computed. The final seam line is represented in Fig.4.4c, together with the corresponding unit vectors normal to the surface.

Once the seam line has been identified, the desired path for glue deposition is computed similarly, by exploiting the intersection between suitably defined surfaces and the STL mesh of the shoe upper. The seam contour is scaled inward to generate multiple concentric perimeters on the top surface of the shoe upper (using

algorithms derived from computer aided machining, such as [117]). Each scaled contour is projected on the X-Y plane, and a new surface is extruded in the n_{ideal} direction to identify the actual intersection with the top surface of the upper. This operation is repeated for the desired number of perimeters. The final path is composed of different lines, perimeters and infills defined as the coordinates of points on the shoe upper surface and the corresponding unit vectors normal to the surface. An example is reported in Fig.4.5a.
Fig.4.5a also report the presence of infills, visible in the toe area. Infills are generated by identifying the intersection between additional surfaces, generated by using user-defined primitives with a selected patterns (e.g. lines generating parallel planes), and the shoe upper surface. For instance, parallel lines can be used to define the path for glue deposition in the correspondence of the tip and heel of the top surface, like represented in Fig.4.5b.



(a) *(b)*

Figure 4.5: a) Example of the final tool path constituted by two perimeters and an infill. b) Depiction of the surface used to get the infill.

The generation of the path on the lateral surface of the shoe upper still relies on the seam contour, but this time the slicing surface changes. The cutting plane is obtained by lowering the seam along the n_{ideal} direction by the requested quantity and by generating two child curves inward and outward, so that the surface between these two boundaries intersect the portion of the upper.

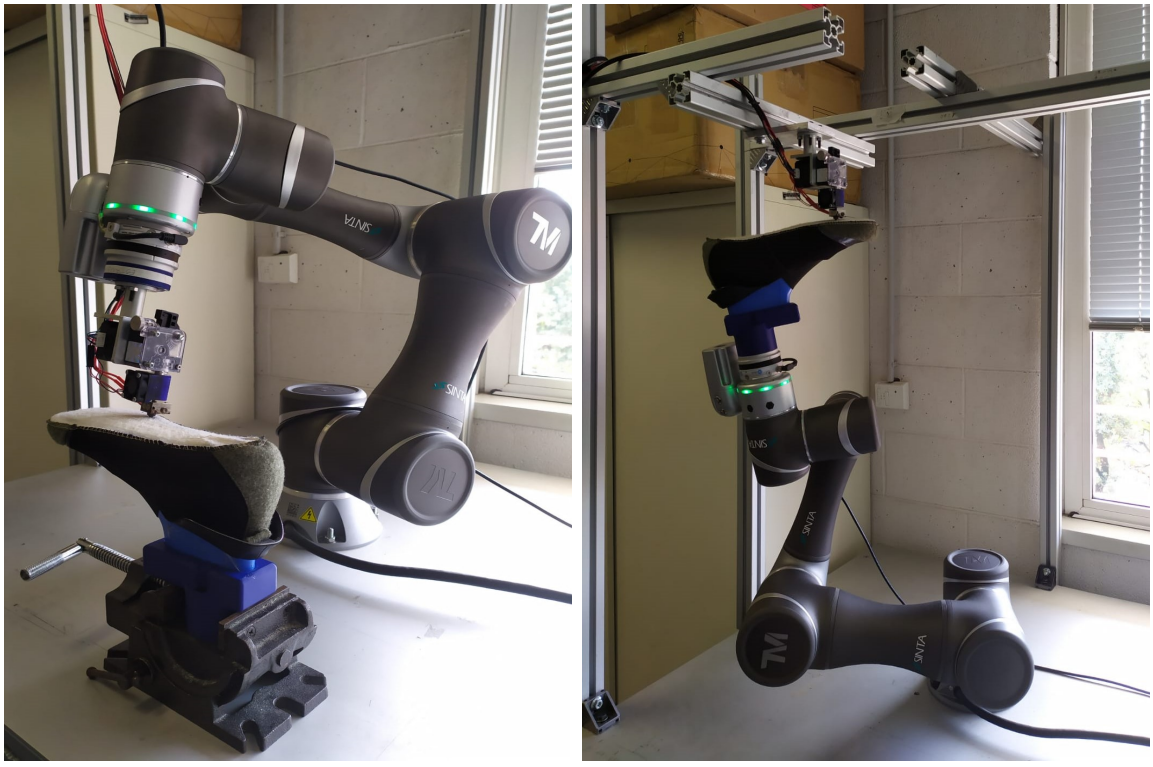
4.2.1 Experimental setup

The robotic cells proposed for the automated deposition process are composed of two main subsystems:

- A system for deposition of molten material, consisting of an extrusion system to be commanded synchronously with the robot's movement, to provide the exact quantity of adhesive.
- A 6 d.o.f. manipulator, which fulfills the task of positioning the system for material deposition relatively to the shoe upper.

Figure 4.6 represents the two alternative configurations proposed for the usage of the robot manipulator, both of them aimed at governing the relative position and orientation between the glue supply system and the 3D-shaped surface of the shoe. A former configuration (figure 4.6a) relies on a mobile extruder, and consequently

on an orientable nozzle, directly connected to the robot wrist, with the shoe being steadily grounded into the operational space. Conversely, the second configuration is based on a fixed nozzle, grounded into the operational space, with the robot wrist holding and handling the last and the shoe upper worn onto it. This layout is represented in figure 4.6b. The extruder is hold by a supporting structure, whereas the last is clamped by a fastener linked to the robot wrist. During the adhesive deposition, the tool supplies a suitable flow of glue at a chosen rate, in order to deposit the required adhesive film over the curved surface. The material has to be deposited on both the bottom and the side part of the shoe upper.
Figure 4.6 represents the cell layouts exploiting the collaborative Techman Robot TM5-700, even if also the industrial robot Mitsubishi RV-2F-Q has been considered in the present analysis. The extrusion system, specifically designed to facilitate the



(a) *(b)*

Figure 4.6: Experimental test layouts: a) Mobile extruder; b) Mobile shoe last

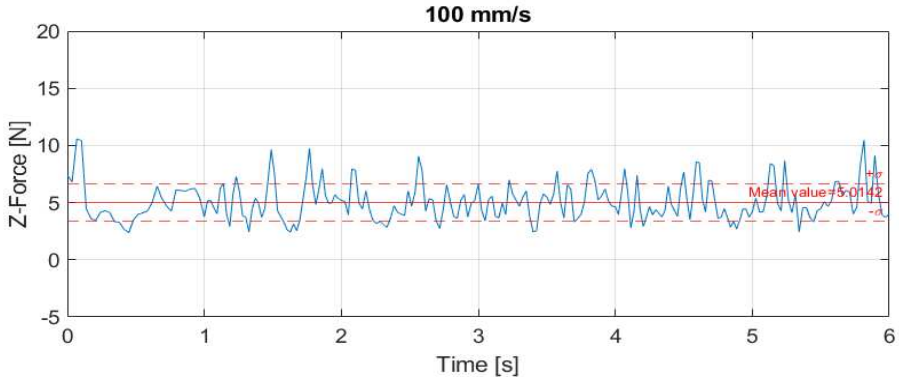
filament provision, can be exploited in both the configurations proposed, being either mounted on the wrist of the robot or on a fixed frame which would keep it in a set location. It was specifically designed and assembled with customized components, to enable the possibility to interchange any item, code and robot during the tuning and the experimental phases. It is composed of a stepper motor (NEMA 17, 400 steps per Revolution, 0.34 Nm stall torque) to control the filament feeding to a hot-end, at the desired flow rate. The adhesive adopted is a prototype product produced by a third-party Company. It is in the form of a filament with diameter equal to 3 mm, and has to wet the shoe upper surface in order to guarantee a stable and durable bond. The extruder motor is coupled with a speed reducer (ratio 1/3) that directly

drives the filament to the hot-end at a feeding speed of around 2 mm/s. The hot end is able to guarantee the temperatures required by the specific glue adopted, in the range $285 - 295^oC$. The nozzle diameter is equal to 0.8 mm.

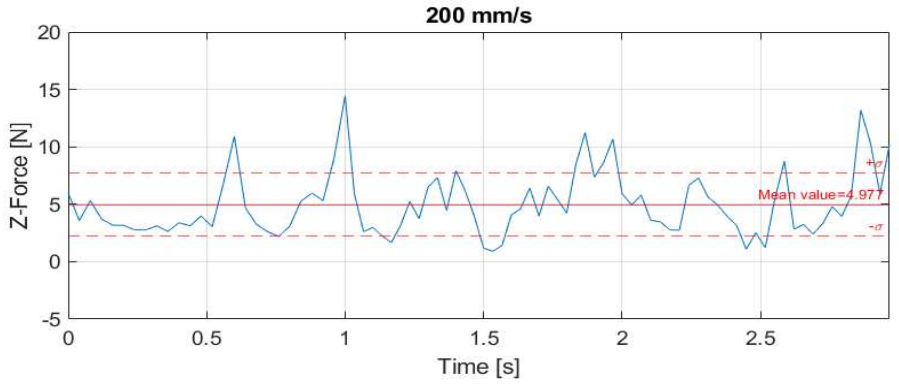
4.2.2 Experimental tests and results

In the case of the cell layout with mobile extruder, the force measured is affected by the inertia of the extruder (around 0.3 kg), which should therefore be compensated to get the actual contact force between the extruder and the shoe upper surface. The efficacy of the control for different linear speeds was assessed by adopting a low weight feeler (i.e. $0.05[kg]$) built with ABS through FDM process, so that the contact force can be directly assimilated to the force measured by the load cell.
In the case of Mitsubishi industrial robot, the final parameters adopted are $Gain = 20[\mu m/N]$ and damping coefficient $= 0.1[N/(mm/s)]$ [115].

Figure 4.7 reports the contact force results achieved with the mobile extruder in the case of the collaborative TM robot, at the speeds of 100 mm/s (figure 4.7a) and 200 mm/s (figure 4.8b). The experimental results showed a full capability of



(a)

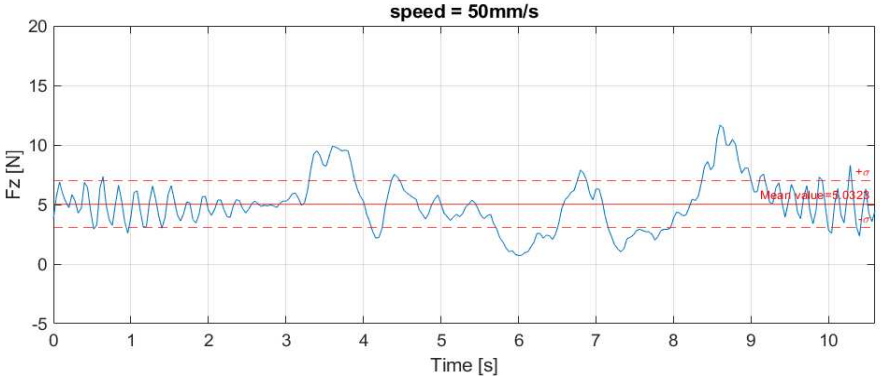


(b)

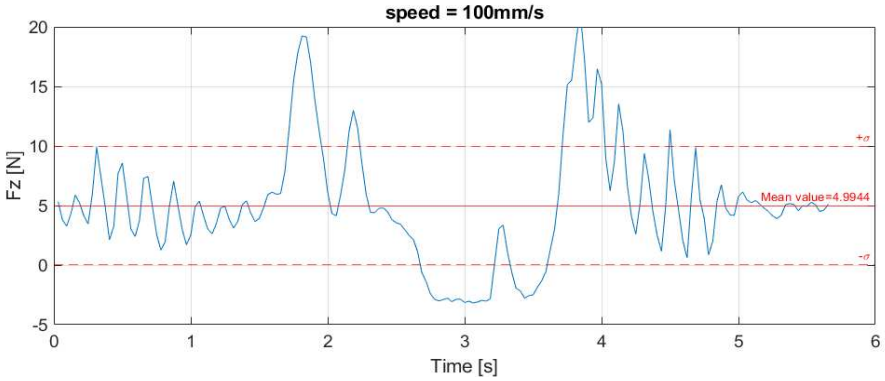
Figure 4.7: Contact force results for the cell layout with mobile extruder, TM Robot. (a) 100 mm/s. (b) 200 mm/s.

the system to get the linear speed of 200 mm/s, whereas, when dealing with the capability of maintaining the proper contact force, the results show an increase of

the standard deviation of the contact force starting from the speed of 100 mm/s.
In the case of the test cell with mobile shoe upper and fixed extruder, in which the shoe last is directly attached to the robot wrist, the control performance is expected to be intrinsically worse, due to the higher inertia attached to the robot wrist and to the fact that the moments applied to the robot wrist rapidly vary as a consequence of the relative position between the extruder tip and the shoe upper surface: with reference to the setup represented in figure 4.6b, indeed, for a given contact force between the extruder and the shoe upper surface, lower moments are generated when the extruder is in contact on the heel rather than when the contact is on the tip.
Figure 4.8a and 4.8b report as an example the results achieved with TM robot at the speed of 50 mm/s and 100 mm/s respectively. In Figure 4.8b two detachment are observed around 3 s, corresponding to the zone of the sharp curve of the shoe upper toe.
It can be therefore concluded that the application with the mobile shoe should run at lower speed compared to the case with mobile extruder, which would imply longer glue deposition time for each shoe. However, this drawback would not make the overall cycle time achievable with this cell layout necessarily longer, thanks to other pros this cell configuration leads to. When the shoe upper is directly hold by the robot, indeed, right after the glue deposition phase it could be directly driven to the press for sole fastening, without the need to re-grip the workpiece.
Moreover, when the shoe upper is moved by the robot and the hot extruder is fixed, the robotic cell could more easily be upgraded to become a collaborative robotic cell [118, 119], thanks to the fact that this solution shows no hot-mobile parts in the operational space, thus enabling a higher degree of security for the operators.
The use of a collaborative robots would open to relevant advantages in terms of improvement of the organization of manufacturing industries, allowing the introduction of robot in craft enterprises without the need of major changes in the layout of the production floor. This would be particularly relevant in an environment like the one typical of small and medium-sized shoe manufacturing enterprises, where the shoe production process still involves human labor to a large extent.



(a)



(b)

Figure 4.8: Contact force results for the cell layout with mobile shoe upper and fixed extruder. TM Robot. a) Speed 50 mm/s.
b) Speed 100 mm/s.

4.2.3 Discussion

Elaborating the CAD model of the workpiece considered in the contact-based operation allows generating the trajectory to be followed by the robot end-effector. To evaluate the performance of the trajectory and the overall of the robotic solutions, considering the two robotic cells layouts to execute the contact-based operation, contact force between the tool center point and the workpiece is considered as an evaluation parameter. The performance is dependent on the stability of the contact force behavior. The performance is also dependent on the amplitude of the error between the generated trajectory and the real one to be followed. If the error is acceptable, the force control loop will compensate it by applying online adjustments to the generated trajectory to keep the contact force the same as the force set point. The error amplitude is calculated comparing the trajectory generated, and the one actually followed applying the force control.
The solution based on the use of the CAD model allows generating the trajectory that has the performance depending on the degree of similarity of the workpiece and the CAD model. If the workpiece differs with respect to the model or there are inaccuracies in its form or positioning, the accuracy of the trajectory will be highly affected.

To overcome the aforementioned limitations of the CAD-based technique, different solutions based on the data acquired by a 3D vision systems could be feasible. In the next section, a solution for trajectory planning is introduced to evaluate the feasibility of a low cost 3D camera in the considered task.

4.3 Trajectory planning using a fixed 3D camera

An Intel Realsense D435 3D stereo depth camera is fixed to have in its field of view the robot workspace or the area reachable by the robotic arm used in a contact-based operation. In this robotic solution, the 3D camera is fixed in such a way to be pointed toward the object and to acquire a from top images of it. The robotic cell design is done in such a way since it is necessary to execute a contact-based operation over the upper face of the workpiece. The 3D stereo depth camera acquires color and depth images of the workpiece from a single and fixed point of view.
As a testing application of the accuracy of the developed solution based on the use of a fixed 3D camera, it is considered the application of glue deposition in footwear manufacturing. In this application, the robot arm moves the attached glue deposition system towards the upper face of the shoe and start following the glue deposition trajectory while exerting a predefined contact force.
In figure 4.9, a summary of the pipeline of the developed robotic solution for the trajectory generation is shown. The steps start from the acquisition of color and depth image of the workpiece until executing the trajectory exerting the desired contact force over the workpiece. The experimental setup used and the detailed description of the necessary steps are shown as follows.

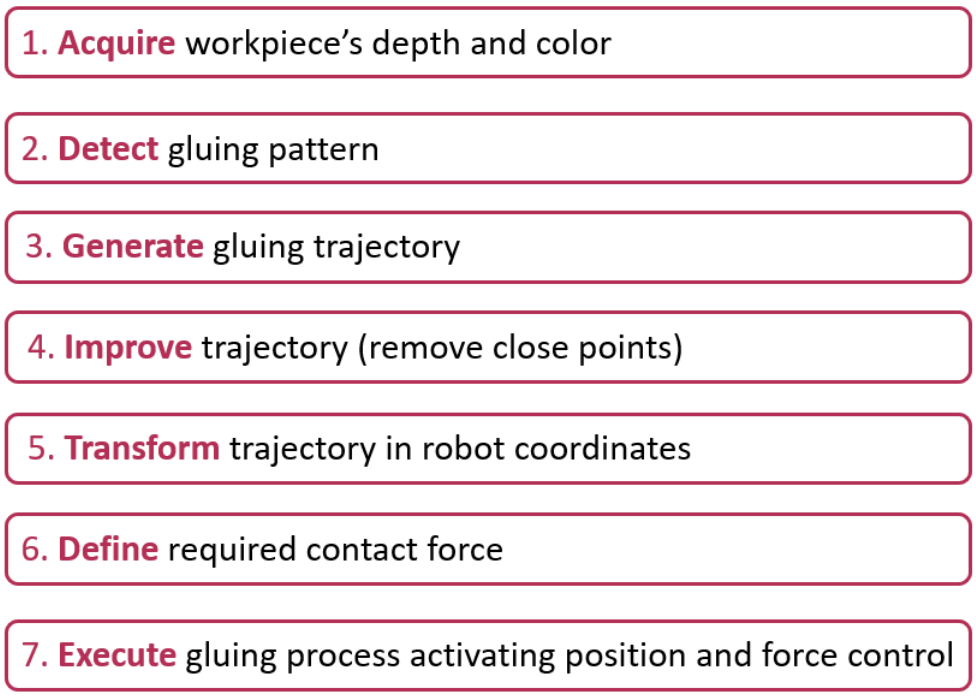


Figure 4.9: Pipeline of trajectory generation using a fixed 3D camera

4.3.1 Experimental setup

The experimental setup used to test the feasibility of generating a precise working trajectory is the one shown in figure 4.10. The setup is made of the Techman Robot TM5 that is a 6-axis collaborative robot, D435 depth camera, Hex-E Onrobot

force/torque sensor and a 3D printing Fused deposition modelling (FDM) extruder for glue deposition.

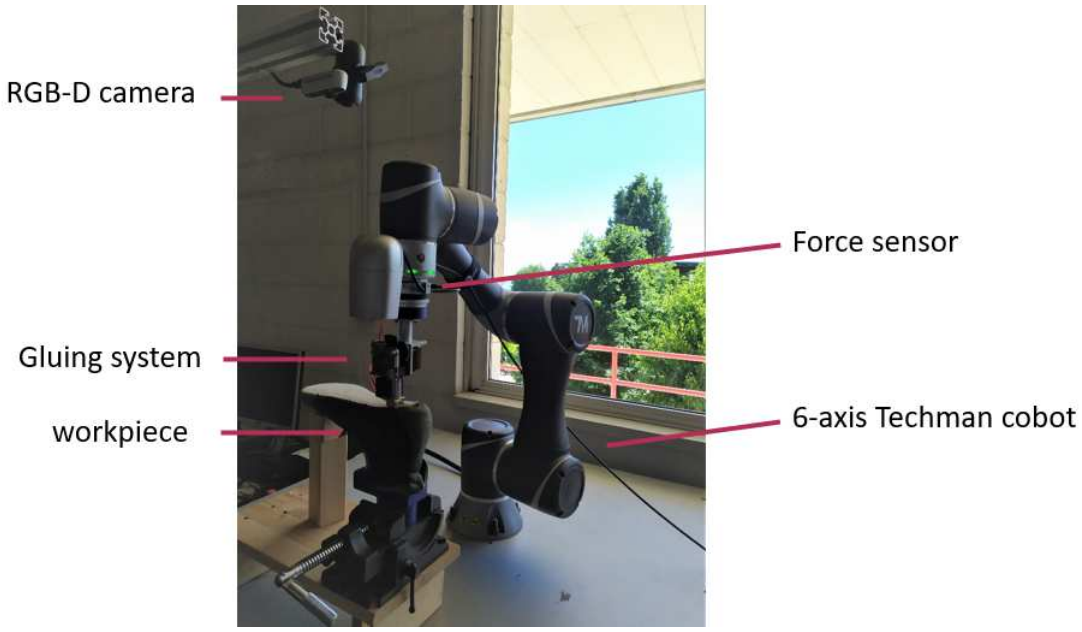


Figure 4.10: Static camera case layout

The 3D camera has an optimal working range between 0.3 and 3 meters. It is fixed in such a way to have the closest part of the shoe at the distance between 0.3 and 0.4 meters. Considering this configuration, the closest object to the 3D camera is the workpiece and the farthest part is the floor under the working table at the distance of 1.70 meters. The measured depth error of the 3D camera used is lower than or equal to two percent of the distance with respect to the workpiece. That means an expected error between 6 and 8 millimeters.

4.3.2 3D camera configuration and data acquisition

The 3D camera parameters are configured to capture 30 Frames per second (fps) of the working table at the resolution of 1280 x 720 pixels for both of the color and depth sensors. The first step in the image elaboration is to execute the alignment procedure, explained in section 3.1.1, to transform both of the images to have the same physical characteristics (to have the same size and to be referred to the same exact physical sensor origin). This alignment process can be executed to refer the depth image to the color image or vice versa.
Then, a search algorithm is developed to capture continuously color and depth images of the working table. Evaluating depth images, objects having a depth value between 0.3 and 0.4 meters and placed on the working table within the robot workspace can be detected. A depth image captured of the robot workspace is shown in figure 4.11. The depth map contain the depth values of each pixel, combining the observed scene. In the figure, the depth values are represented as a depth map with pixels that have their color depending on their distance from the camera origin. In this depth color map configuration, the closest pixels to the camera are represented

by darker blue color and with the increase of the depth value the pixels tend to be represented with a clearer blue color.

Figure 4.11: Shoe depth image color map

Evaluating the depth image, if the object is detected for several seconds, the presence of the workpiece is confirmed and trajectory generation algorithm starts to define the trajectory to be followed by the glue deposition system based on the current pose of the workpiece.

4.3.3 Images elaboration and trajectory generation

After the confirmation of the presence of the workpiece within the robot workspace, the trajectory planning algorithm is used.
The figure 4.12 shows the shoe considered by the contact-based operation. In this application, it is necessary to deposit the glue in correspondence to the outer stitches seam connecting the white upper part shown in figure with the rest of the shoe.

Figure 4.12: Shoe color image

The stitching seem is the outer contour of the upper white area of the shoe. To find it, a searching algorithm is developed to detect the group of neighboring pixels

having white color. It may happen to find several spots in the color image that have a white color. To overcome this limit, all the spots having white color are detected. For every spot, the area is calculated and only the biggest white spot is considered. Once the pixels making the white spot are selected, it is possible to calculate the outer pixels forming the contour of this white spot. Having aligned the color and depth images, the related depth values of the stitching seem are fetched from the depth image to have the three-dimensional coordinates of the gluing pattern. The detected gluing pattern is shown in figure 4.13, where the outer stitching seem is highlighted by red color.

Figure 4.13: Detected gluing pattern

The gluing pattern detected at this stage is in pixel unit. To be able to control the movement of the robot to follow that trajectory, it is necessary to transform it in millimeters. This process is done using a built-in function of the Librealsense SDK (Software Development Kit). The used function transforms the pixel coordinates to real world coordinates in millimeters, exploiting the intrinsic parameters of the 3D camera (e.g. focal length, coordinates of principal points).
After the transformation from pixel to millimeters, a filtering procedure has to be applied to the generated trajectory. This is due to the high pixel density in every millimeter that causes to have consecutive points having a distance between them with a very small values. Using this small displacement between a point and the following one may limit the performance of the controller interpolating the trajectory and thus the performance of the force control loop. To overcome this problem, the points are filtered to have a minimum distance between consecutive points of 5 millimeters.

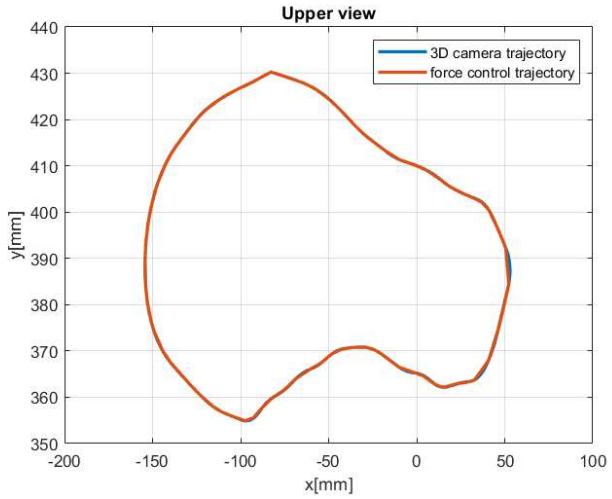
4.3.4 Experimental tests and results in glue deposition

To guarantee the good execution of the glue deposition operation, a good contact force exerted on the shoe is 5 N at the robot speed of 100 mm/s. This to guarantee

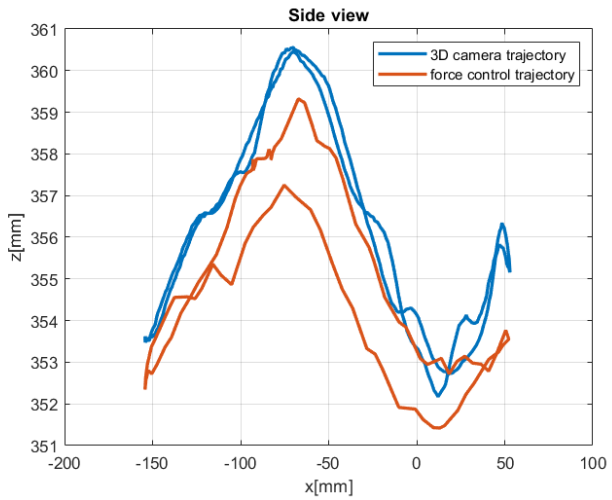
the consistency of a good glue deposition over the time. The overall evaluation of the accuracy of the generated trajectory can be represented in terms of the amplitude of the positional error of the generated trajectory and the real trajectory followed applying the force control loop. The overall accuracy is evaluated also on the contact force values measured during the execution of the trajectory applying force control with a set-point of 5 N.

Error measurements

Applying force control loop with a set-point equal to 5 N. Figure 4.14 shows the comparison between the generated trajectory in blue color and the feedback trajectory in orange color. Figure 4.14a on the top, shows the comparison between the



(a) Upper view



(b) Side view

Figure 4.14: Comparison between generated and the actually followed trajectories

two trajectories viewed from upper point of view. In the figure, the two trajectories are coinciding. Figure 4.14b on the lower side shows the comparison between the

two trajectories from side point of view. This comparison shows differences between the two trajectories, with correspondence of the area having a major slope and inclination. The error has a mean value of 1.3 millimeters, a maximum value of 3 millimeters that is less than the depth values provided by the manufacturer.

Contact force evaluation

The goal of every industrial application is to be executed in the shortest time possible, increasing the movement speed of the robot as much as possible without lose of performance and without adding any safety hazards. The contact force evaluation is done considering four different movement speeds that are 25 mm/s, 50 mm/s, 75 mm/s and 100 mm/s.
In figure 4.15, the measured feedback of the contact force values are shown considering the different speeds. Shown also the mean values of every graph with black color and the standard deviation in red color.

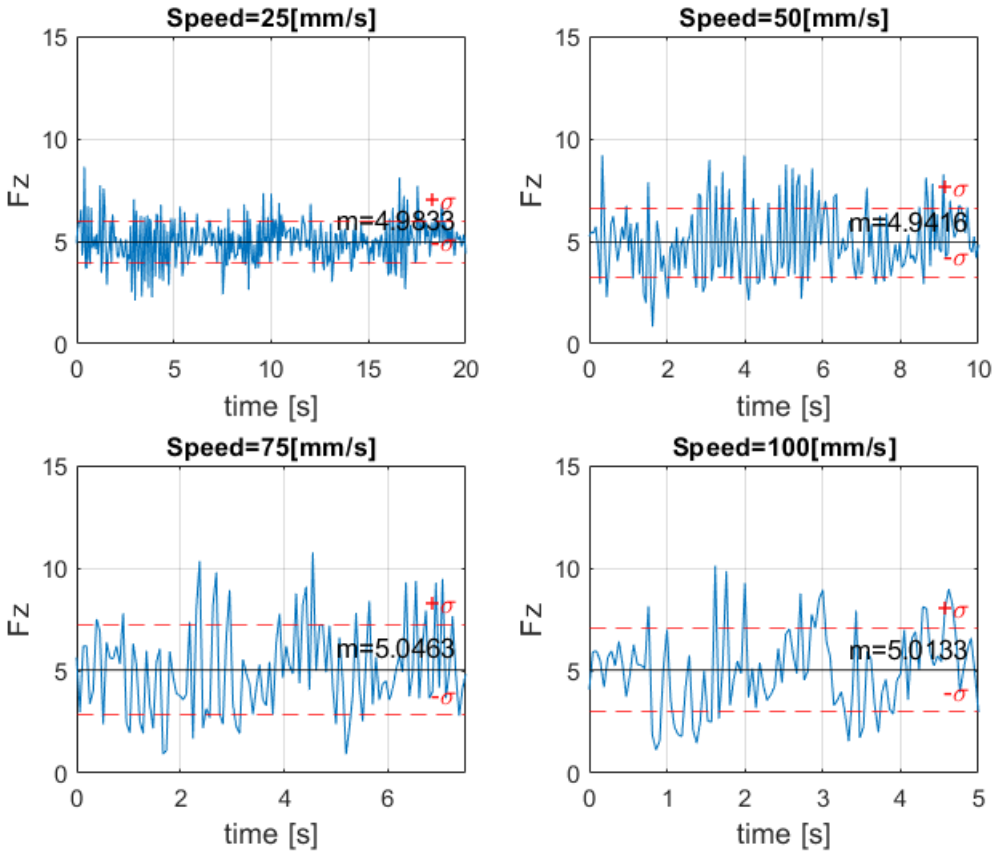


Figure 4.15: Contact force at different speeds: 25 mm/s, 50 mm/s, 75 mm/s and 100 mm/s

At low speed (25 mm/s) the contact force mean value is equal to 4.98 N and the standard deviation to 1.02 N. When Increasing the speed to 100 mm/s the mean value is still close to the 5 N target (5.01 N) and the force standard deviation gets equal to 2.03 N. These force values are comparable to those reported in [114], in

which the TCP trajectories were defined on the basis of CAD model of the shoe upper.
The mean values of the different tests are approximately equal to the set-point of the control loop of 5 N. It is possible to notice that the standard deviation increases with the increase of the movement speed. This means that the force control loop performance increases in the positional error compensation as the speed increases. This is despite the constant mean value obtained.

Parameter tuning

An important parameter that influences the performance of the force control loop is the filtering distance. The minimum allowed distance between the consecutive points because close points limit the performance of robot controller to interpolate better the trajectory and also decrease the performance of the force control loop.
Figure 4.16 shows the comparison of the feedback of the contact force following two trajectories with different minimum distances between consecutive points of the trajectories. The contact force with a minimum distance of 5 mm is shown in blue color, and the contact force considering a minimum distance of 10 mm is shown in red color. In bright blue and bright red, the corresponding mean values are shown together with the standard deviations.

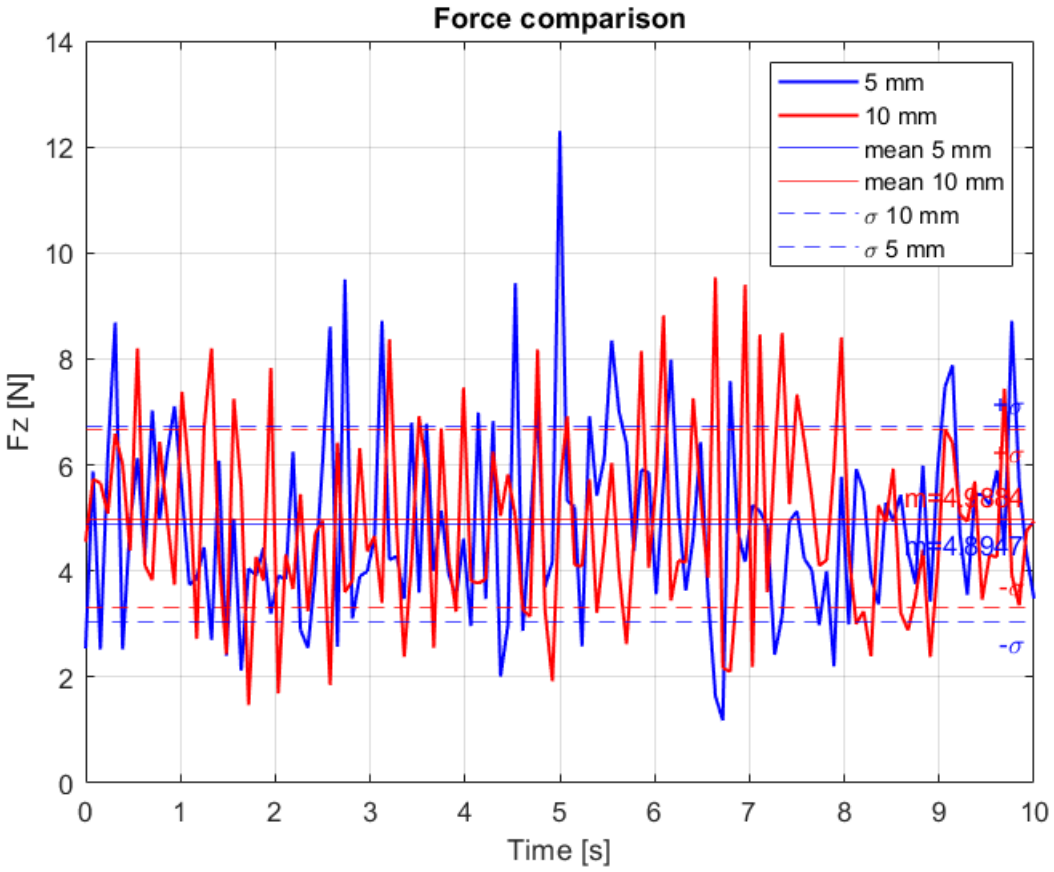


Figure 4.16: Contact force comparison changing the minimum distance

It is possible to notice that the increase of the filtering distance allows a reduction of the force peak value from 12.3 N to 9.54 N, as well as a lower standard deviation (from 1.84 to 1.8).

4.4 Discussion and limitations

In this chapter, a state-of-the-art methodology based on the use of the workpiece CAD model has been developed. The performance of that methodology is used as a reference to assess the performance of the solutions based on the use of the vision systems.
The trajectory planning procedure based on the use of a workpiece CAD model is introduced together with the analysis of the requirements that a good performing contact-based operation in general, or specifically glue deposition application, must satisfy. These requirements are the ability to maintain the contact-force constant and be able to compensate for possible error in the generated trajectory.
The solution for trajectory planning of glue deposition applications based on the use of the data acquired by an economic 3D stereo depth camera (Realsense D435) is analyzed. A force control loop is used to maintain an exact contact force to guarantee the right execution of the robotic operation and overcome inaccuracies of the trajectory generated. The system obtained good performance: stable mean value of contact force and error of generated trajectory of 1.3 mm. The obtained system performance is similar to those obtained using accurate CAD model.
The solution based on the use of only the data acquired by the 3D camera obtained a good performance executing the contact-based operation, comparing it to the traditional technique that is based on the use of the CAD model. Hence, the low cost 3D camera is feasible to be used in the trajectory generation of contact-based operations that require accurate trajectory.
This solution has the limitation of generating the working trajectory only over the face of the workpiece in the field of view of the 3D camera and visible in the acquired image. To overcome this limitation and be able to generate the trajectory over the different lateral faces of the workpiece, the 3D camera could be moved around the object to acquire images that cover all its lateral faces. This process is done attaching the 3D camera to the robot end-effector instead of having it fixed in a single position.

Chapter 5

Odometry-based 3D model reconstruction and robot trajectory planning

In this chapter, the developed solutions to reach the goal of the thesis of developing a simple tool for trajectory generation of robotic contact based operations are shown. The developed solutions provide a wide range of applications that are mostly to be found in industry, specially in SMEs where the working environment is dynamic and the position and exact 3D model of workpieces is not predefined. The solutions are based on the use of 3D vision sensors to acquire one or more color and depth images of the workpiece. These images are elaborated to extract the information necessary to generate the working trajectory to be followed by the robot end-effector while executing the contact-based operation.
The different solutions are described in details, highlighting the layout of the experimental setup and the configuration of the hardware and the software tools. It is shown also for all the solutions discussed, the accuracy assessment and the experimental results. Based on the different cases, different key performance indices are shown for the specific evaluation procedure.
The solutions in this chapter are based on the 3D reconstruction based on the integration of color and depth images captured by a moving 3D camera. Firstly, the general 3D reconstruction algorithm is shown using a manually moved camera. Then, the general algorithm is used in the case in which the 3D camera is moved automatically by the robotic manipulator and highlighting the necessary assumptions to apply it in robotic applications. In the end, the developed algorithm is applied in application in which the 3d reconstruction of objects is necessary. This application is the quality monitoring of the 3D concrete printing.

5.1 3D reconstruction based on manually moved 3D camera

A fixed camera allows covering only one face of the workpiece. A moving camera could be used to cover, besides the top face, also lateral faces of the workpiece,

making it possible to generate the working trajectory over them. To develop the robotic solution able to capture images of the workpiece from different faces and integrate them in a unique 3D model having an overall knowledge of the workpiece. The first step to is to develop an algorithm that uses images that are captured by a manually moving camera as explained in this section, and then integrate the algorithm in the robotic cell as explained in the next section.
A 3D reconstruction algorithm is based on a computer vision techniques to estimate the camera motion and the pose of the camera capturing each image. Then the images are integrated together to generate the 3D model of the workpiece based on their acquisition pose.

5.1.1 3D camera configuration and data acquisition

In this application, the D415 stereo depth camera is used to capture synchronized color and depth images of the considered workpiece. The resolution of the color and depth sensors are set to 1280 x 720 pixels and frame rate of 30 frames per second. This frame rate is suitable since the camera will be moved at moderate speed by a human operator.
Figure 5.1, shows a representation of how the 3D camera is moved around the workpiece.

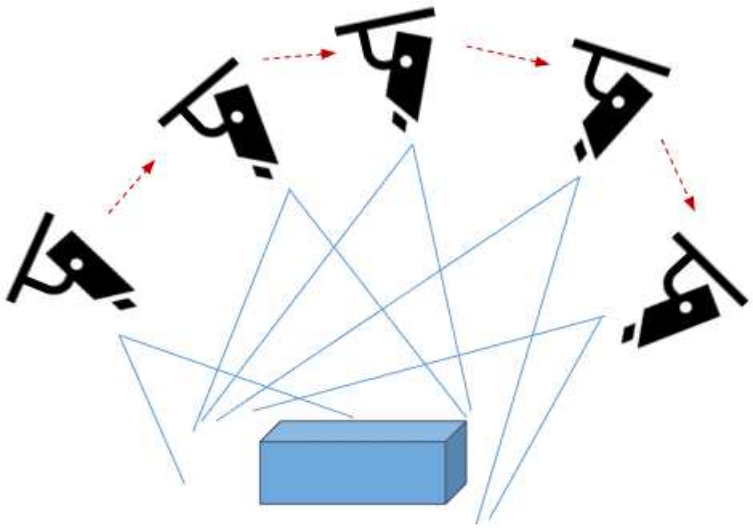


Figure 5.1: Manually moved 3D camera data acquisition

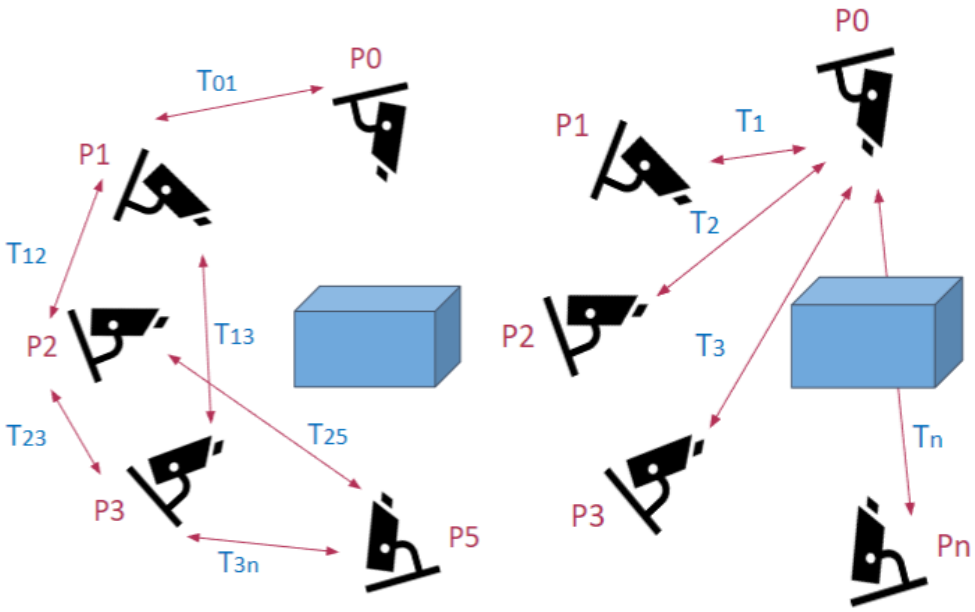
To be considered that, the camera needs to be moved in a stable way to avoid the capture of blurring images. It is important to satisfy the minimum distance requirement to maintain with respect to the scanned workpiece that is for the used camera D415 and at the resolution chosen is equal to 450 millimeters. The other aspect to be considered is to have the whole workpiece or at least the part of it considered by the scanning visible in all the captured images and to be within the camera field of view.

5.1.2 Developed algorithm

In this section, the software pipeline is introduced, starting from elaboration of the RGB-D images toward the creation of the 3D model of the object.

RGB-D odometry

The RGB-D images, are captured by a moving camera. That means that every image is referred to the camera coordinate system at that moment. The first step in the developed algorithm is the estimation of the camera pose of each image using RGB-D odometry technique introduced in [98].
Figure 5.2 shows a summary of the procedure to calculate the pose of every image with respect to the first image that is used as a common reference frame of the images and the 3D model to be reconstructed.



(a) First step: matching between randomly picked and consecutive images

(b) Second step: calculate the relation to refer images to the first one

Figure 5.2: Matching procedure to calculate camera poses

The estimation of the pose between two images, consists in the calculation of the homogeneous transformation matrix. That if applied to one image, it matches it to the other one like it has been captured from the same position and same camera orientation.
Having several RGB-D images, the matching process is done by matching every image and randomly picked images from the data set. In this way, for every image, several camera motions are estimated with respect to the matched images. This is show in figure 5.2a where several homogeneous transformation matrices are estimated to refer the several images to other images of the data-set.
The calculated camera motions matrices, are used to refer all the images with respect to a common coordinate system. The common reference frame used is the

frame of the first image. The referring process consists in the use of the camera motions found for every image to find the optimal sequence of camera poses, having between every two images in sequence the smallest camera motion and the biggest overlapped area. The output of this process is shown in figure 5.2b where the several transformation matrices for the different images are calculated.

Integration

The creation of a unique volume of the observed scene is done by using Truncated Distance Signed Distance Function technique (TSD) [102]. This technique consists in the initialization of an empty voxel grid that is referred to the above-mentioned common reference frame. The empty grid is then filled progressively with the information fetched from the single RGB-D images.
For pixels that are visible in several images and their values do not coincide, a weighted average of the values is done to obtain their depth. This process is done, giving higher weights to the images where the line of sight of the camera is parallel to the pixel's normal. Smaller weights are given with the increase of the deviation between the normal and the line of sight.
Once all the RGB-D are analyzed, the voxel values contained in the grid are considered to find the so called isosurface (a smooth and continuous surface given by averaging neighboring voxel values) representing the observed objects surfaces. The final results of this process is a mesh model of the observed scene that can be elaborated to extract the object we interested in.

5.2 Trajectory planning based on the use of data acquired by a moving 3D camera with a robot

Attaching the 3D camera to the robot end-effector and using the algorithm developed in the previous section, allows for the reconstruction of the 3D model of the workpiece considered by the contact-based robotic operation.
The developed robotic solution is introduced, focusing on explaining the algorithm pipeline to be able to acquire the RGB-D images and integrate them to obtain the 3D model. The algorithm summary is shown in figure 5.3.

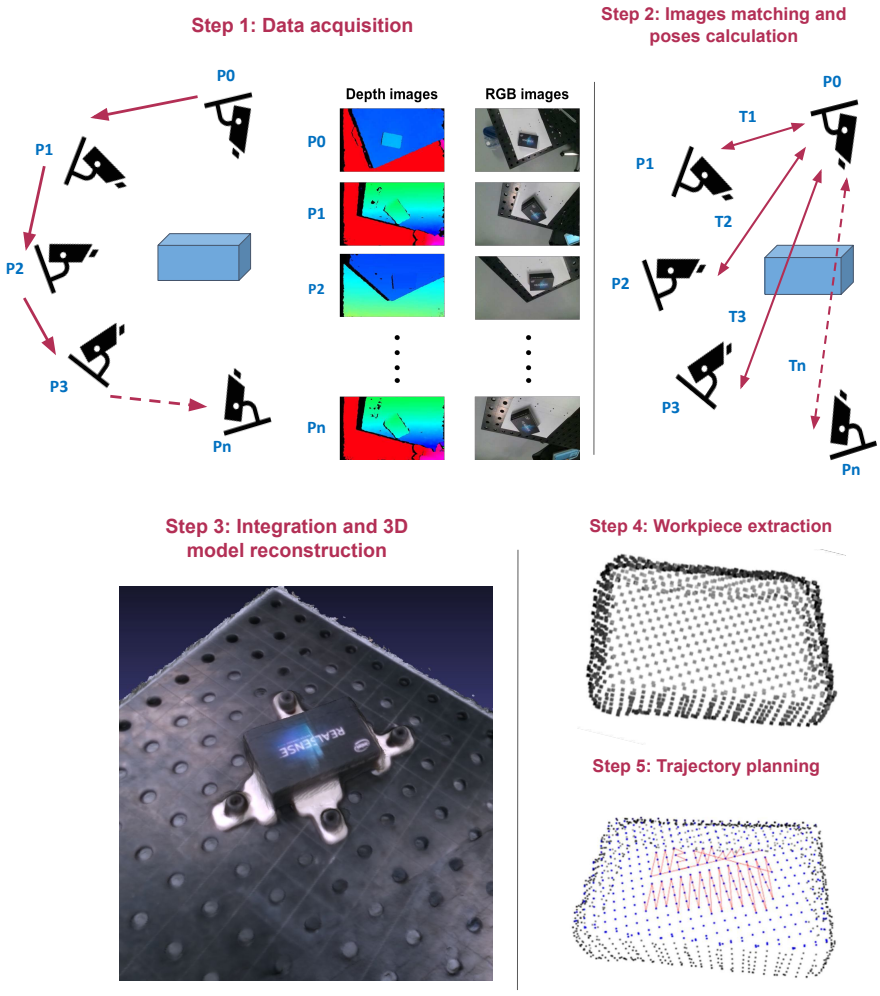


Figure 5.3: pipeline

In step 1, consists in moving the camera following the scanning trajectory and to capture RGB-D images of the working space. In step 2, images captured are analyzed to estimate the camera poses while capturing every image. In step 3, information contained in RGB-D images are integrated in to construct a 3D model of the scene. In step 4, extraction procedure of the interested workpiece is shown. Finally, in step 5 the area considered by the operation are considered to generate the working

trajectory.
To develop the algorithm described, the system setup shown in figure 6.2 is used. The system is made by a 6 axis UR5e collaborative robot [88], D415 Realsense stereo depth camera [89] and a 3D printed tool used for the 3D model accuracy evaluation as will be discussed.

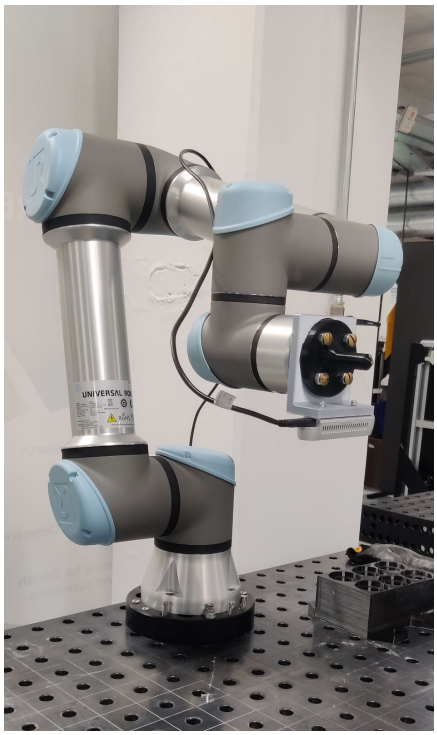


Figure 5.4: Setup

5.2.1 Developed algorithm - software

To control the robot movement commands to follow the defined trajectory that points the 3D camera towards the working area of the robot from different points of view and to simultaneously control the 3D camera to capture the necessary color and depth images, a software is developed in Python programming language. The software code is based on the use of Open-source libraries that are Librealsense, Open3D, OpenCV and Real-time Data Exchange (RTDE).
Figure 5.5, shows the flowchart summarizing the developed code. The software code consists of three steps: Data acquisition, point-cloud reconstruction and elaboration and finally trajectory following.
The first part in explained in section5.2.2, and it consists in the execution of two synchronized function that starts at the same moment. These two function are to start the following of the acquisition trajectory and to start the acquisition of the color and depth pairs of images.
The second step is responsible for the integration of the images to reconstruct the 3D model. This step is similar to the algorithm explained in section 5.2.5.

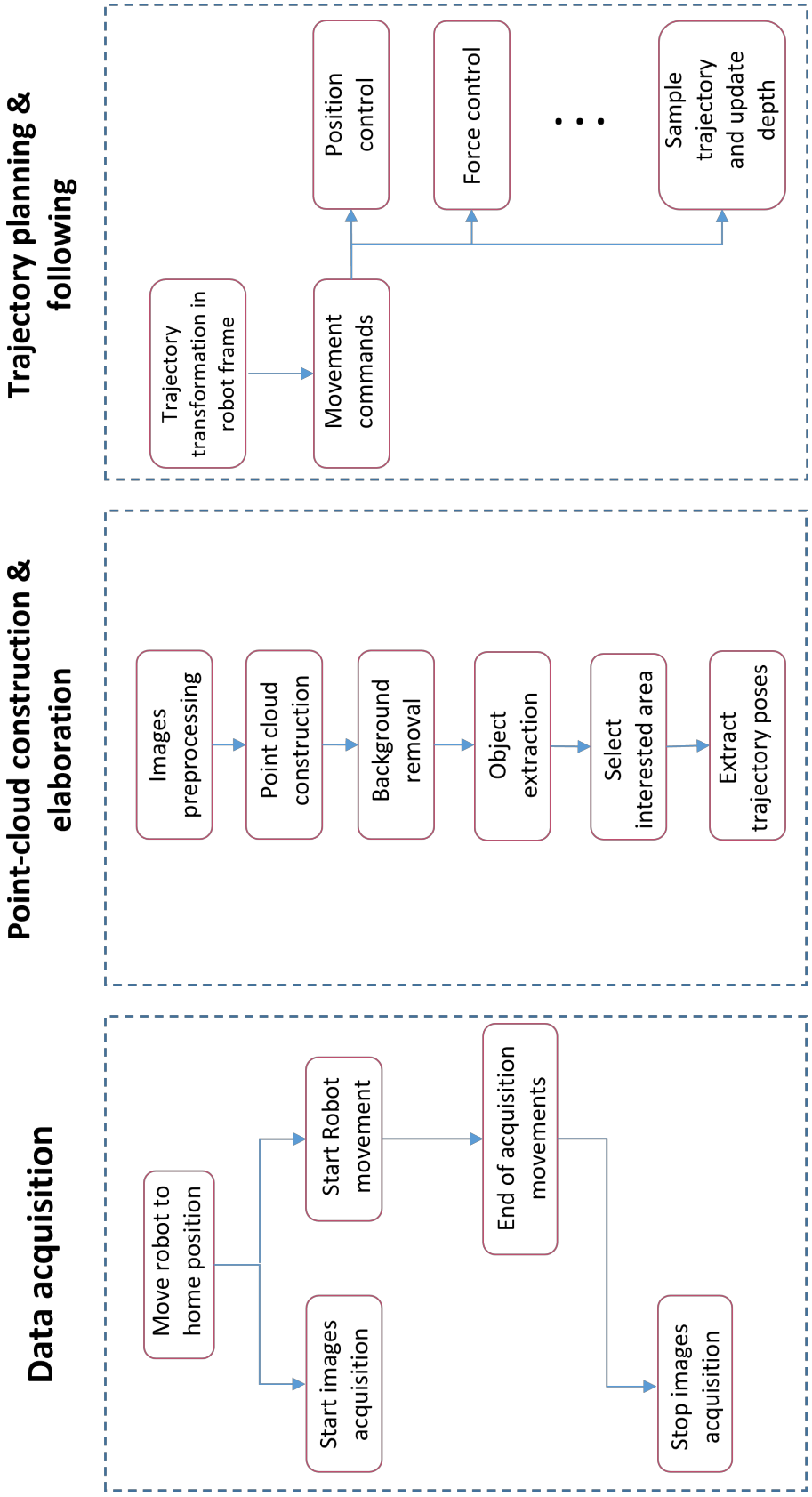


Figure 5.5: Developed software flowchart

5.2.2 Images acquisition and preparation

Sensor setup and data acquisition is the first step in the algorithm developed. The vision sensor used, Realsense D415 camera captures color and depth images using different sensors and allows configuring the image quality based on user's preferences. In the sensor data sheet [89], the sensors can be configured to resolutions up to 1280x720 pixels for the depth sensors and 1920x1080 pixels for the color sensor. Sensors resolution have a significant effect on the output accuracy and optimal configuration parameters for the considered case is discussed in section 5.2.8. At these high resolution configuration, the sensor can capture images at frame rate up to 30 frames per second (fps).
Color and depth images are captured by different sensors and could have different resolutions. So the first step is to transform the two kinds of images to have the same characteristics. The two images are aligned to have the same exact size and to be geometrically referenced to the same physical sensor origin. The alignment procedure explained in details in the published work [95] and allows to align depth to color or vice versa.
To acquire the images that allow for the 3D construction of an object, all parts/sides of the workpiece have to be covered and be visible in the multiview images. The scanning trajectory planning can be automated to optimize the visibility of the working area where the workpiece is placed. This process can be considered as a 3D multi-goal path planning problem. Where the multiview 3D camera poses are determined to optimize the coverage of certain object. In [120], a solution is proposed to find the optimal path to guarantee the visibility of a given set of objects.
In this work, the scanning trajectory is planned manually, making the following assumption. Supposing that the workpieces are always fixed inside a predefined area within the robot workspace, the scanning trajectory is planned in such a way that the 3D camera is moved and rotated accordingly toward different view-points that cover the working area. In these viewpoints, the 3D camera should always be pointed toward the working area. The scanning trajectory is made following acquisition movements that start by capturing a from top images of the working area, then the robot moves the camera to capture images of the different sides. That trajectory is approximately a half of a rectangle where three sides of the working area are covered (the side facing the robot base and the two lateral sides). The outer face is not reachable by the robotic setup used. A robot arm with higher reach can be used to cover it.
The explained scanning movements are feasible to scan solid objects with simple geometry. For objects with more complicated geometries or having undercuts, the scanning movements can be modified to cover also the undercuts by adding camera views that allow the coverage of these areas.
The 3D camera is synchronized to capture the multiview images of the working area while following the scanning trajectory. The total length of the scanning trajectory is divided in steps to define where the 3D camera captures an image of the work area. The step magnitude between an image and the following one, in millimeters, influences the total number of images and also the overlapped parts between them. Very small steps may lead to excessive number of images covering similar parts of the object. While very big steps may cause a partial coverage of the object.

Other factors to be considered to plan the scanning trajectory are the robot movement speed, scanning time and the 3D camera frame rate. The relationship between all the mentioned factors can be described by the following system of equations 5.1.

$$\begin{cases} n = \frac{l}{\Delta l} \\ V_r = \frac{l}{t} \\ fps = \frac{n}{t} \end{cases} \tag{5.1}$$

Where n is the number of total RGB-D images in the dataset, l is the length of the scanning trajectory, Δl is the scanning step, V_r is the robot movement linear speed, t is the scanning time and fps is the frame rate of the 3D camera.
The RGB-D elaboration algorithm is based on the technique provided in [97] where the authors suggest having a maximum number of images of 100 images. Knowing the length of the scanning trajectory and choosing a scanning time to satisfy production requirements allows calculating the other variables that are robot linear speed and camera frame rate. Figure 5.6 shows a practical example of a scanning trajectory, in which a scanning trajectory length of $l = 1000\ mm$ and dataset size equal to $n = 100$ images are used. The figure shows the coordinate systems representing the camera pose at each RGB-D image acquired. The acquisition movements start by capturing a from top image of the working table to capture global information of the object. Then the robot moves the camera to capture images of the different faces of the object. The test shown in the figure is done, acquiring 100 pairs of synchronized color and depth images of the working table in front of the robot.
From the first equation of 5.1 the scanning step is $\Delta l = 10\ mm/frame$. Supposing that a feasible scanning time is $t = 10\ s$, from the second equation of 5.1 the robot speed is $V_r = 100\ mm/s$. From the third equation of 5.1 the 3D camera frame rate is $fps = 10\ frames\ per\ seconds$.

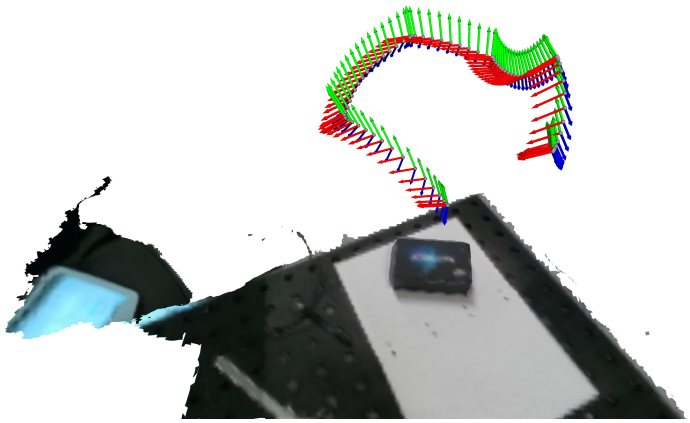


Figure 5.6: Acquisition trajectory to capture multi-view RGBD images of a box

5.2.3 Images matching and camera pose estimation

The color and depth images contain different information about the observed scene, as they have been captured from different view angles. To create a unique 3D model,

it is necessary to refer each image to a common reference frame, and then integrate the information contained in the images together to create the 3D model. To calculate the motion between the camera poses capturing the RGB-D images of the observed scene, the RGB-D odometry technique [96] is used. This technique, comparing two images captured by a moving camera of a static scene, determines the homogeneous transformation matrix that if applied to the second image maps it to match the first one. In our previous paper [121] the information of the end-effector poses from which each image is captured has been exploited to integrate all the images to build the 3D model. In the present paper, we discuss more in deep the possibility to reconstruct the camera poses by exploiting the odometry technique, which might be a useful alternative in case the poses from which the image is taken are not available (e.g. manual free scanning).
The RGB-D odometry technique [97] determines the camera motion or the homogeneous transformation matrix by solving an optimization problem. The objective function used in the optimization problem is shown in equation 5.2.

$$E(T) = (1 - \sigma)E_p(T) + \sigma E_g(T) \tag{5.2}$$

Where $E(T)$ is the objective function that is calculated considering two terms. The first term $E_p(T)$, to be maximized, considers pixels photo-consistency [98, 99] (if the same pixel is visible in two images it has to be represented by the same color and brightness in both of the images) [100]. It is represented as squared differences of pixel intensities. The second term $E_p(T)$ term, to be minimized, is a geometric objective function that measures the error of pixel positions. T is the homogeneous transformation matrix to transform an image to the coordinate system of the other image. The result of the optimization problem is the optimal T. $\sigma \in [0, 1]$ is a weighting term to balance the two terms.
Like most of Iterative Closest Point registration algorithms, the one used, needs an approximated initial value [101] of the transformation matrix. This initial value is usually obtained by a global coarse registration algorithms based on the point cloud features. Controlling the step length, Δl, in equation 5.1 allows having small motion between consecutive images and having a large overlapped portion. A feasible initial value of the homogeneous transformation matrix, the identity matrix that is modified at every iteration to find the best solution minimizing the error of pixels color and position.
The camera reference frame, corresponding to the pose from which the first RGB-D image is captured, is considered the common reference frame to which all images must be referred. The RGB-D Odometry algorithm calculates camera motion between only two images. If the two images do not have enough overlapped pixels, this may cause low accuracy calculating the motions. To match all images of the dataset, each image is compared to the following one, to find the matching homogeneous transformation matrix between them. The calculated individual matrices, representing camera motions between consecutive images, are finally used to calculate for every image a homogeneous transformation matrix of the cumulative motions to refer it to the first image.
The procedure is explained in Algorithm 1. The procedure input is the dataset of

RGB-D pairs of images of the working area. The output is an array of homogeneous transformation matrices that aligns every image to the first image of the dataset. H_{aux} is computed considering the objective function in equation 5.2 and applying the optimization procedure from [97]. H_k is the homogeneous transformation matrix that transforms the k_{th} RGB-D image to the coordinate system of the first image. It is given by the cumulative motion of the camera, starting from the first image until arriving to the camera pose from which the k_{th} image has been taken.

Algorithm 1 RGB-D Odometry

Require: Im RGB-D images dataset
Ensure: H Array of homogeneous transformation matrices
1: $n \leftarrow length(Im)$
2: $H \leftarrow []$
3: $H.Append(Identity\ matrix[4x4])$
4: **for** $k \leftarrow 1$ to $n-1$ **do**
5: First image $\leftarrow Im(k)$
6: Second image $\leftarrow Im(k+1)$
7: Compute H_{aux} from 5.2 considering: first image, second image and Identity matrix [4x4]
8: $H_k \leftarrow H[0] * \cdots * H[k-1] * H_{aux}$
9: $H.Append(H_k)$
10: **end for**

5.2.4 Volume integration for 3D model construction

All the multiview color and depth images of the working area are integrated together to create a unique 3D reconstruction representing the observed scene. The volume integration process, introduced in [102, 122], is based on the generation of a voxel grid representing the scene, in which color and depth information of each pixel of each RGB-D image are integrated and referred to a global reference frame. The voxel grid is therefore a three-dimensional representation of the all the information contained in the two-dimensional color and depth images. From the voxel grid, after applying the truncated signed distance function (TSDF), the isosurface representing the scene is found. The isosurface is the smooth and continuous surface interpolating the not empty voxels in the grid.
The procedure consists of the following steps. A dataset of $(1, \ldots, n)$ RGB-D images is discredized in a voxel grid. Then, calculate the signed distance functions $(s_1(x), \ldots, s_n(x))$ of each voxel x in the i_{th} RGB-D image. These values represent the distance that a voxel has with respect to the nearest surface along the camera line of sight. Voxels between the observed surface and the camera origin have a positive value that increases as it goes closer to the camera. Voxels of the observed surface have null values, while not visible points have a negative distance.
The depth measurements are subject to noise and two depth measurements of the same point, using the same 3D camera and from the same point of view, may have different values. To estimate better the observed surface depth measurement, the signed distance values from the different images are averaged to calculate a cumu-

lative signed distance function $S(x)$. To be considered also that the same point, in two images that are taken from different view angles, may have different values. More weight has to be given to the depth value in the image that cover better that point. Weight values $(w_1(x), \ldots, w_n(x))$ of each voxel x in the i_{th} RGB-D image. This weight measures the degree of certainty of the depth measurement of a point. Weight value assignment is dependent on the orientation of the surface normal vector and the viewing angle. If the camera line of sight is co-directional, the weight is the highest and decreases with the increase of the angle between them. This relation can be represented as the dot product between the two vectors.
The weighted integration of the all RGB-D images is given by the following equations 5.3 and 5.4. Where $S(x)$ is the global-signed distance value for every voxel of the integrated scene and $W(x)$ is the global weight for every voxel. Discretizing the functions $S(x)$ and $W(x)$ in a voxel grid allows calculating the zero-crossing or the isosurface having $S(x) = 0$ that describe the observed scene.

$$S(x) = \frac{\sum_{i=1,\ldots,n} s_i(x) w_i(x)}{\sum_{i=1,\ldots,n} w_i(x)} \tag{5.3}$$

$$W(x) = \sum_{i=1,\ldots,n} w_i(x) \tag{5.4}$$

From practical test, the grid voxel size can affect the accuracy of the reconstructed 3D model. In figure 5.16 a comparison between voxel sizes and a color image of an object is shown. In figure 5.16c, a color image from the dataset used for the 3D reconstruction is shown. In figure 5.16a, the point cloud considering big voxel size is shown, while in figure 5.16b the point cloud using a smaller voxel is shown. The smaller voxel size allows reconstructing a 3D model with higher level of details and hence more degree of similarity with respect to the real object. The effect of the voxel size on the accuracy of the generated trajectory will be shown in the subsection 5.2.8.

(a) Colored image *(b) Voxel size 5 mm* *(c) Voxel size 1.3 mm*

Figure 5.7: Voxel size effect on points density

5.2.5 3D model elaboration

The result obtained from the integration process is the 3D model of the observed scene as a mesh model and a point cloud. In the developed algorithm, the point cloud representation is used.
The information contained in the point cloud for every point is the coordinates with respect to the reference frame (i.e. the one corresponding to the camera at the first image), the color in form of RGB values, and the normal vector. In the considered setup, the xy-plane is parallel to the workbench where the robot and the workpiece are placed. The point cloud z coordinates represents the distance of each point with respect to the workbench, higher z values corresponding to more distant points. These values have been exploited to develop the searching algorithm, aimed at identifying and selecting the area of the workpiece to be considered in the contact-based operation. The procedure is described in the algorithm 2.

Algorithm 2 3D model elaboration and trajectory generation

Require: Pc Reconstructed 3D model as point cloud
Ensure: Generation and execution of working trajectory
$Pc_{filtered} \leftarrow []$, $Plane \leftarrow []$, $Objects \leftarrow []$, $Workpiece \leftarrow []$, $Trajectory \leftarrow []$
$n_{points} \leftarrow length(Pc)$
for $i \leftarrow 1$ to n_{points} **do**
    $point\ depth \leftarrow pc[i, z]$
    if $point\ depth < threshold$ **then**
        $Pc_{filtered}.Append(pc[i, :])$
    end if
end for
$Plane \leftarrow$ Plane search to remove points of the working bench
$Objects \leftarrow Pc_{filtered} \notin Plane$
$Object \leftarrow$ points satisfying the conditions 5.5
Calculate C_x and C_y of the centroid of $Object$ using equation 5.6
$n \leftarrow length(Object)$
for $i \leftarrow 1$ to n **do**
    $dx \leftarrow offset_x$, $dy \leftarrow offset_y$
    if $(dx - C_x < Object[i, x] < dx + C_x) \wedge (dy - C_y < Object[i, y] < dy + C_y)$ **then**
        $Trajectory.Append(Object[i, :])$
    end if
end for
Transform points
Execute trajectory

After the reconstructed 3D model is built, combining the color and depth images of the scene covered by the camera during the scanning process, a filtering process has to be applied to remove irrelevant points in the reconstructed 3D model. The point cloud can be trimmed by removing points far from the camera representing unrelated background points. In algorithm 2, the filtering process consists in the

evaluation of the depth value or the distance that a point have with respect to the working table on which the robot is placed. If the distance is greater than a predefined threshold, the point is excluded from the point cloud.
Assuming that the object is placed in the working area that is reachable by the robot and referring the point cloud with respect to the camera origin pose, capturing the first image. The point cloud can be trimmed, removing far points from the camera representing unrelated background points (i.e. further than 1.5 meters).
In order to find the workpiece in the remaining part of the point cloud, a plane segmentation process is applied to find the working table on which the workpiece is placed. Clusters of points forming a plane are found (neighboring points having depth in a defined range, i.e. 2 mm, this value considering inaccuracies of the 3D camera in calculating the depth of the flat surface of the working table). The points forming the working table can be found by searching for the biggest cluster of the planes found. Removing these points allows removing the working table from the overall point cloud, leaving only the workpiece. Plane segmentation is done by using a RANSAC search algorithm. RANSAC algorithm is an iterative algorithm that samples a subset of the dataset and use them to calculate a fitting model (in the considered case a plane). All points of the dataset, are then checked to evaluate if they fit in the model. If the number of points that fit in the model is lower than a defined threshold, the process is repeated sampling other points and calculating a new plane model. This is repeated until finding the model that fits for most of the points in the dataset. To configure the plane search algorithm, three parameters are defined. Distance error, that is the maximum distance that a point have to be considered as a part of the plane. The number of points that are randomly sampled from the dataset to estimate the plane. Finally, the number of iteration of the algorithm to sample points, estimate and verify the plane. Values used are: $Error = 2\ mm$, $points = 3$ and $iteration = 1000$.
The result are the plane equation coefficients a, b, c and d satisfying, for every point of the plane having the coordinates x_p, y_p and z_p, the plane equation $ax_p + by_p + cz_p + d = 0$.
In figure 5.8, the output of the plane segmentation process is shown. The plane is highlighted by a red color. Removing these points allows removing the working table, leaving only the workpiece considered by the operation.
Once the plane equation is found, the points forming it are selected and excluded from the point cloud to leave only the workpiece points. It may happen to find extra points due to the noise of the sensor. An example of these irrelevant points is shown in figure 5.9a. To be able to remove these points, a clustering process is done to put in several groups the neighboring points and most probably are forming an object. The output of this process is a list of the groups and the number of points of every group or object. In figure 5.9b, several groups or objects are colored in different colors for the sake of visibility. By selecting only the biggest cluster and removing the rest of the points, it is possible to select only the workpiece. Knowing the dimension of the working area where the workpiece is placed and how it is positioned with respect to the camera coordinates and hence the origin of the point cloud, the coordinates of the remaining points of the point cloud are checked to verify if they are found within the working area. Upper and lower bounds values

Figure 5.8: Plane segmentation

(a) workpiece with noise *(b) Clustered objects*

Figure 5.9: Presence of noise and application of clustering process

are used for that check. Points out of the bounds of the working area are excluded from the point cloud. The considered points are those satisfying the conditions:

$$\begin{cases} x_{low\ bound} < x_p < x_{upper\ bound} \\ y_{low\ bound} < y_p < y_{upper\ bound} \end{cases} \tag{5.5}$$

Where x_p and y_p are the point coordinates and $x_{low\ bound}$, $x_{high\ bound}$, $y_{lowbound}$, $y_{high\ bound}$ are the boundaries of the working are in the camera coordinates.

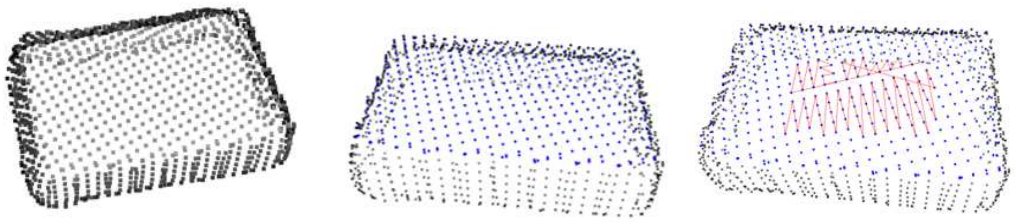
5.2.6 Trajectory generation

Different subroutines have been developed to select the part of the workpiece to be covered in the operation. An example of selecting a specific part of the workpiece could be when a central area should be covered. An example for this case is shown in figure 5.10. To select that area, the coordinates of the points found after outlier removals are used. The first step is to calculate the centroid of the set of points considering x and y coordinates in the following way. For a set of points $P_1, \ldots, P_n$

with the coordinates $((x_1, \ldots, x_n), (y_1, \ldots, y_n))$ the centroid is given by the equation 5.6.

$$centroid = (\frac{x_1 + x_2 + \ldots + x_n}{n}, \frac{y_1 + y_2 + \ldots + y_n}{n}) \tag{5.6}$$

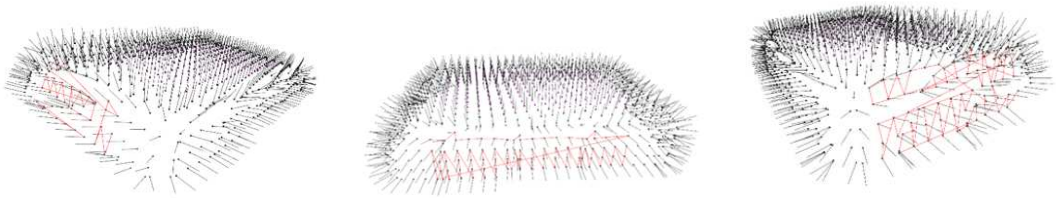
After calculating the centroid of the points, offsets on the x and y-axes are used to define the area dimension to be selected. The points within the limits of that area are selected, and the rest are excluded.



(a) workpiece *(b) Upper face selection* *(c) Center of upper face*

Figure 5.10: Example of searching routine: search for center of upper face

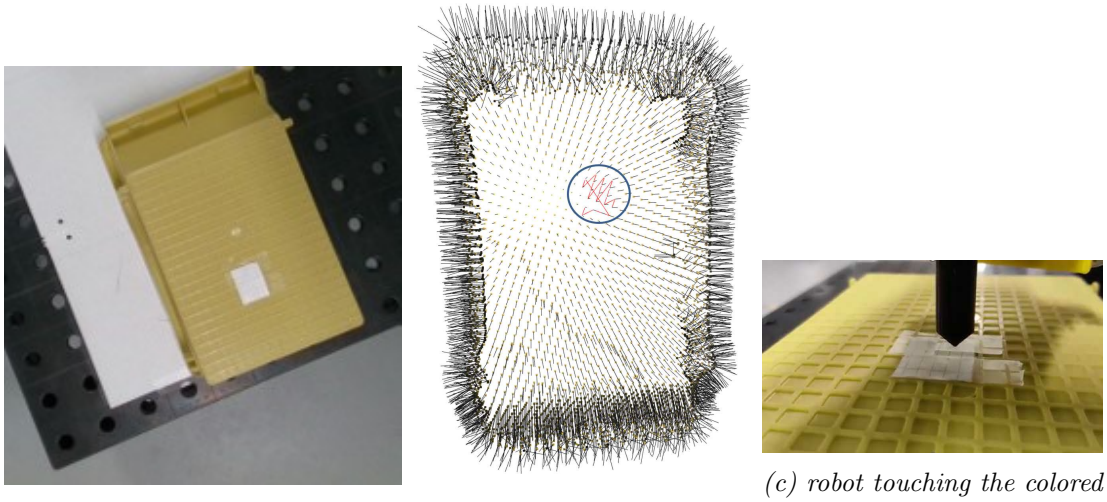
In other cases, when the operation, targets the edges of the workpiece. For example, in polishing operations, the edges can be found elaborating the normals of the workpiece point cloud and selecting the area having neighboring points with huge variation in the normals directions [114]. Having developed the integration algorithm based on the elaboration of color and depth images, the 3D model of the scene preserves also the color information. This process also allows selecting parts of the object, searching for area having a specific color different from the rest of the workpiece. This is useful in case of surface finishing.
The selected portion of the workpiece is made by high number of close points, before the generation of the working trajectory it is necessary to down sample the point cloud. This allows to remove close points and have a minimum distance between the consecutive points that helps the robot controller to interpolate better the movement commands and in the applications considered to have good contact force control.
Knowing the camera pose with respect to the robot end-effector, the homogeneous transformation matrix to transform the coordinates and orientations from the camera reference frame to the end-effector one is found.
Based on the operation to be executed, the order in which the selected points are followed can be changed and parameters like movement speed and contact force have to be defined to generate the trajectory that guarantees a good execution performance.
In other cases when the operation may target a certain one of the lateral faces of the workpiece. An example is shown in figure 5.11. On the left face of the workpiece is selected, in the middle the front face is selected and on the right the trajectory covering the right part is selected. It is possible to note that the normal vectors are available for every point. Comparing the inclination variation in the neighboring points, it is possible to detect also the edges of the point-cloud. That is useful for polishing operations.
Having developed the integration algorithm based on the elaboration of color and



(a) workpiece left face *(b) workpiece front face* *(c) workpiece right face*

Figure 5.11: Example of searching routine: workpiece lateral faces selection

depth images, the 3D model of the scene preserves also the color information. This process also allows selecting parts of the object, searching for area having specific color. This is useful in case of surface finishing and cleaning. In figure 5.12, an example of a workpiece with a colored spot is shown. In the left figure the workpiece is shown, in the middle, the generated trajectory covering the colored spot is shown and in the image on the right the robot end-effector is shown while following the trajectory.



(a) workpiece with white spot *(b) generated trajectory* *(c) robot touching the colored spot*

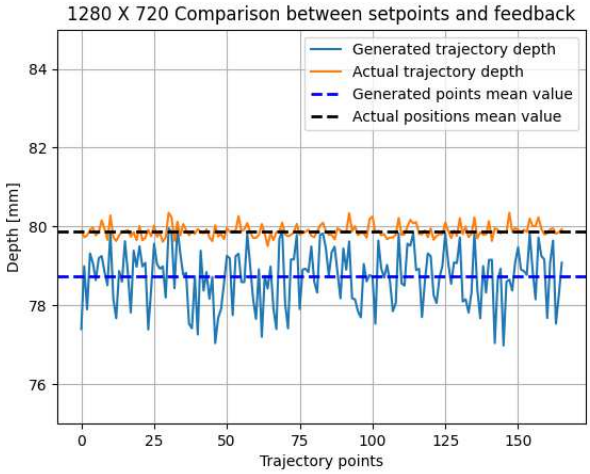
Figure 5.12: Example of searching routine: workpiece with colored spot

The selected portion of the workpiece is made by high number of close points, before the generation of the working trajectory it is necessary to down sample the point cloud. This allows to remove close points and have a minimum distance between the consecutive points that helps the robot controller to interpolate better the movement commands and in the applications considered to have good contact force control.
As mentioned before, the point cloud is referred to the camera origin at the first image capturing position and so the points forming the selected part of the object. Knowing that initial pose with respect to the robot's end-effector, the homogeneous transformation matrix to transform the points coordinates and orientations from the camera origin to be with respect to the end-effector is found.
A final homogeneous transformation matrix is found to refer transform the points forming the trajectory from the end-effector to be referred to the tool center point

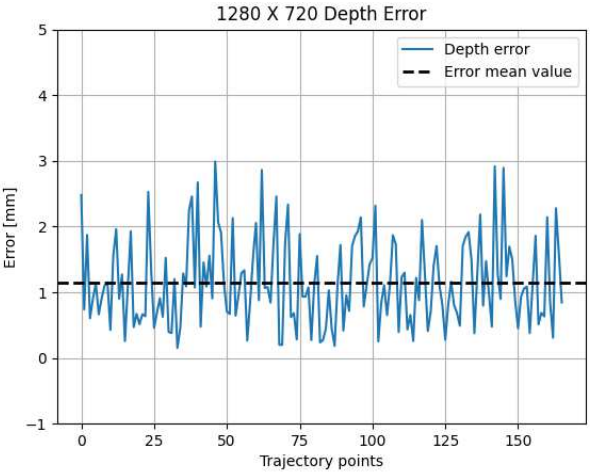
(TCP) of the tool used to execute the contact-based robotic operation.
Based on the operation to be executed, the order in which the selected points are followed can be changed and parameters like movement speed and contact force have to be defined to generate the trajectory that guarantees a good execution performance.

5.2.7 Experimental results and application oriented parameter tuning

In this section, a set of tests and experiments are shown to evaluate the performance of the developed algorithm. We consider cases that are mostly to be found when contact-based robotic operations have to be executed over an object without any previous knowledge. The cases considered are operations on plane objects, operations on curved objects, operations where it is necessary to apply proper contact force to guarantee the right execution of it, etc. Also, we consider the effect that some parameters could have on the accuracy of the output of the developed algorithm, such as camera resolution to capture the initial RGB-D images and voxel size considered during the images' integration.
The setup used during the experiments is the one shown in 6.2. The 3D printed robotic tool shown is used in the accuracy evaluation procedure by moving it toward the object following the generated trajectory. When the tool tip touches the object, the actual position is compared to the extracted, from the generated 3D model of the object, position to calculate the error. This is done exploiting the Universal Robot UR5e used [88] built-in force-torque sensor.
Considering the case in which a robotic contact-based operation has to be executed over the center of the upper face of the box shaped workpiece in figure 5.3. It is necessary to select from the workpiece point-cloud the interested area and generate the related trajectory. The working trajectory generated is the one highlighted by red color in the lower right image. The tool used that in this case is the 3D-printed tester shown in figure 6.2.
Applying force control to reach every point of the trajectory separately allows the evaluation of the accuracy of its position estimation. Figure 5.13 shows a comparison between the generated trajectory points and the tool tip feedback position obtained when it reaches the workpiece's surface. The values indicate the height of the object at every point reached. The error represents a mean value of about 1.1 mm. Maximum value of 3 mm and minimum value of 0.16 mm.



(a) Depth comparison



(b) Error

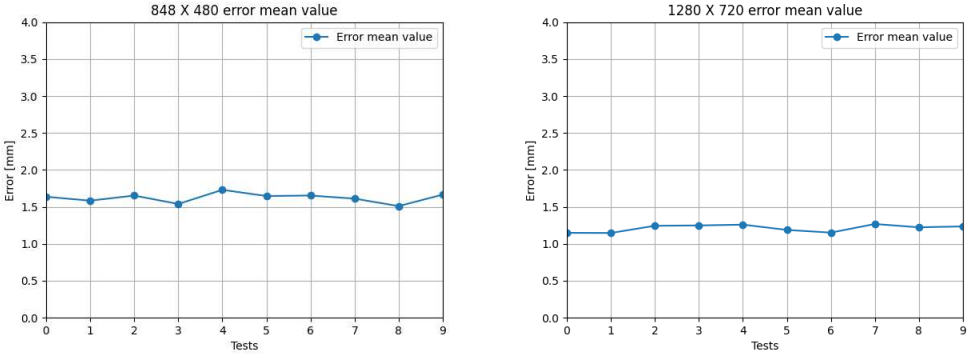
Figure 5.13: Plane surface depth accuracy evaluation

5.2.8 Parameters tuning

Camera acquisition accuracy

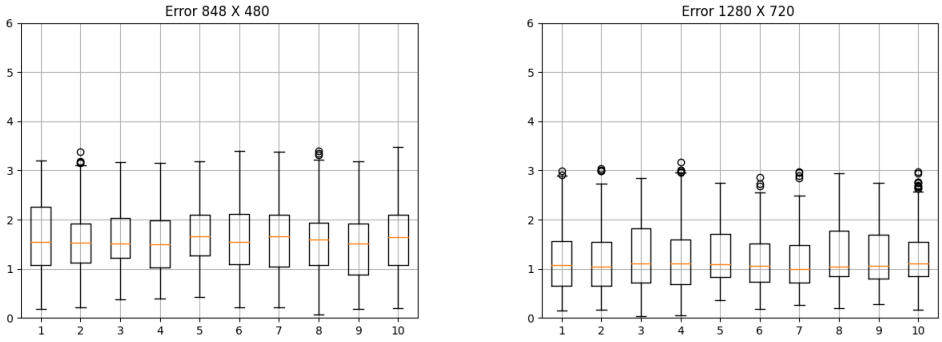
The Intel Realsense D415 has several resolution configurations for color and depth sensors to be set by the user before the acquisition procedure. In the sensor data sheet [89], it is shown that for obtaining the best resolution of the depth sensor, it is recommended to use 1280x720 pixels.
Changing the resolution settings, has a direct effect on the minimum allowed distance between the depth sensor and the observed scene. This distance is the one at which the depth processor starts to measure the depth. For the maximum resolution of 1280x720 pixels, the minimum distance is 450 mm between the sensor and the scene. Decreasing the resolution to 848x480 pixels, the minimum distance decreases to 310 mm.
In the application considered, where the D415 depth camera is moved by the robot

arm to capture the images of an object, a required minimum distance influences the maximum height that the scanned workpiece can have since the height of the extended robot arm is limited.
Tests are carried out to compare the resolution configurations used to capture the multi-view images. The resolutions considered are 1280x720 and 848x480. For each, 10 different acquisitions are made and the output of every test is used to calculate the error average value. The error is calculated for every point of the trajectory and then for each full trajectory the average error is calculated as shown in 5.13b for plane surface case.
Results are shown in figure 5.14 where errors mean values are shown for each acquisition. The error mean values for lower resolution is fluctuating between 1.5 mm and 1.72 mm. While for higher resolution, the error is between 1.15 mm and 1.3 mm. In the figure 5.15, the analysis of trajectories of every acquisition is represented showing the data distribution, quartiles, median, maximum and minimum. Increasing the resolution improve the accuracy of the generated trajectory, but with the trade-off of decreasing the maximum height of the workpiece or imply the necessity to use a robot with bigger reach.



(a) Lower resolution (848x480) *(b) Higher resolution (1280x720)*

Figure 5.14: Error mean value changing camera resolution

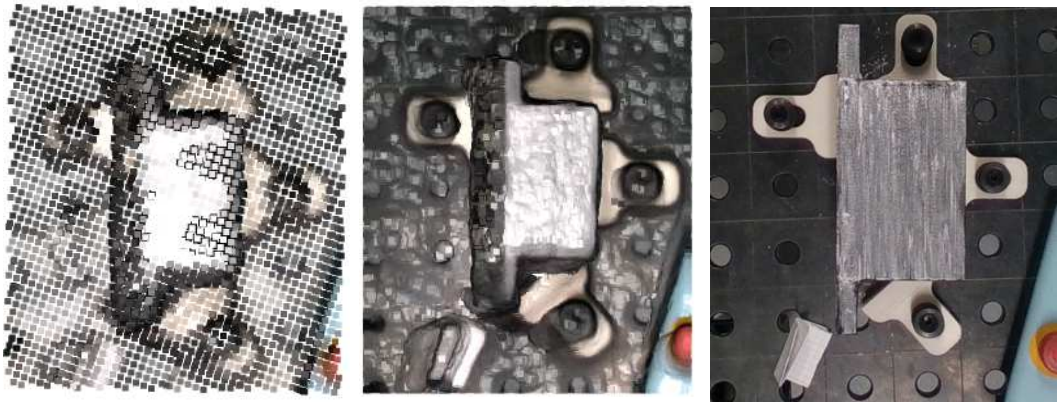


(a) Lower resolution (848x480) *(b) Higher resolution (1280x720)*

Figure 5.15: Statistical analysis summary

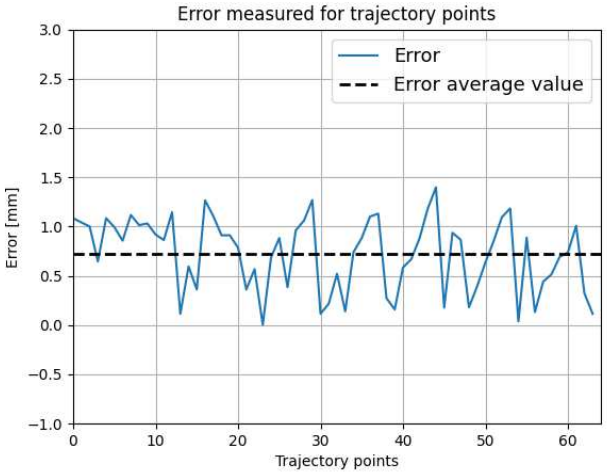
Voxel size effect on Integrated volume using TSDF

The integrated surface of the observed scene is generated transforming the depth values stored in a voxel grid in an isosurface as explained in section 5.2.4. The accuracy of that surface is dependent on the dimension of the voxels forming the grid, as voxels values are obtained by averaging depth values from the depth images. A bigger voxel dimension means that depth values contained in a bigger range are averaged to calculate a single value. Instead, smaller size leads to consider limited depth range and hence more accurate approximation.
In figure 5.16 a comparison between voxel sizes and a color image of an object is shown and in figure 5.17 the related errors are shown. For bigger voxel size of 5 mm, higher errors up to 1.35 mm are obtained. While for smaller voxel size of 1.3 mm lower errors up to 0.8 mm are obtained. Decreasing more the voxel size may lead to registration errors.

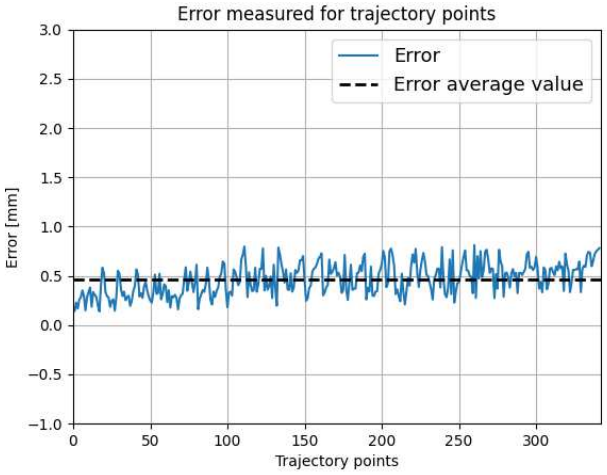


(a) Voxel size 5 mm *(b) Voxel size 1.3 mm* *(c) Colored image*

Figure 5.16: Voxel size effect on points density



(a) Voxel size 5 mm



(b) Voxel size 1.3 mm

Figure 5.17: Voxel size effect on object height error

5.2.9 Performance evaluation

In this section, the performance of the generated mesh model is evaluated, considering cases that are mostly to be found when contact-based operations are executed.

Depth evaluation plane shape workpiece

The object shown in figure 5.16c is considered as an example of objects having a plane surface on which the contact-based operation has to be executed. The test consist in reaching separately the points consisting the trajectory covering the central area.
The object has a known height of 173 mm. In figure 5.18, the evaluation results are shown. The blue line represents the height extracted from the reconstructed 3D model. The orange line represents the real height calculated from the measured

robot TCP position, reaching every point of the trajectory. The measured error shown in figure 5.17b has a maximum value of 0.8 mm

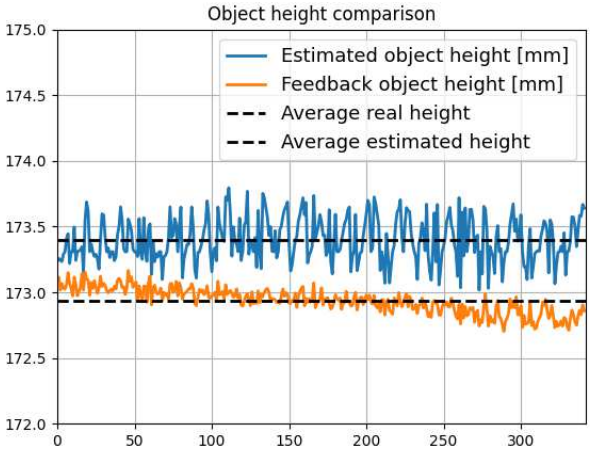
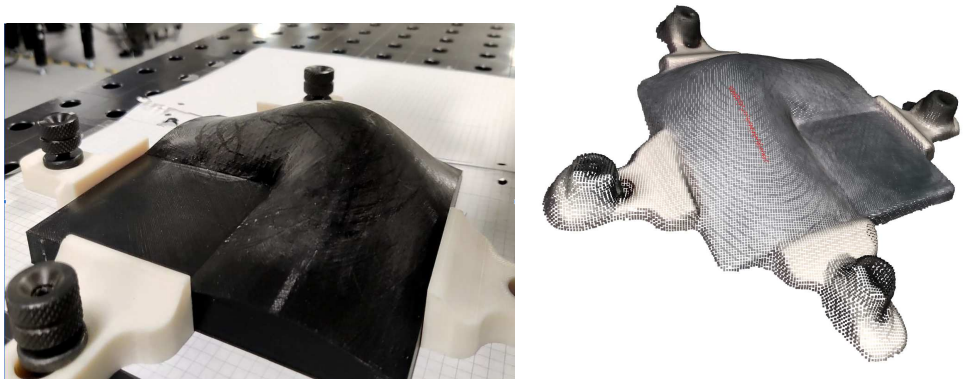


Figure 5.18: Height comparison of a plane object

Depth evaluation curved shape workpiece

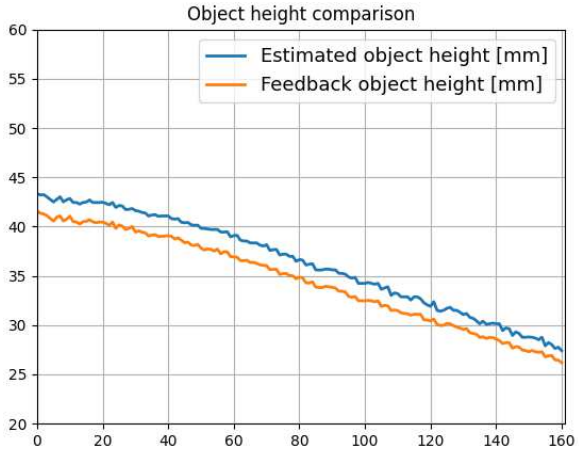
Other type of object that are mostly to be found in the considered cases are the curved objects. The curved object considered for the tests is the one shown in figure 5.19a. In figure 5.19b, the reconstructed 3D model of the object is shown. The points forming the testing trajectory are highlighted by the red color.



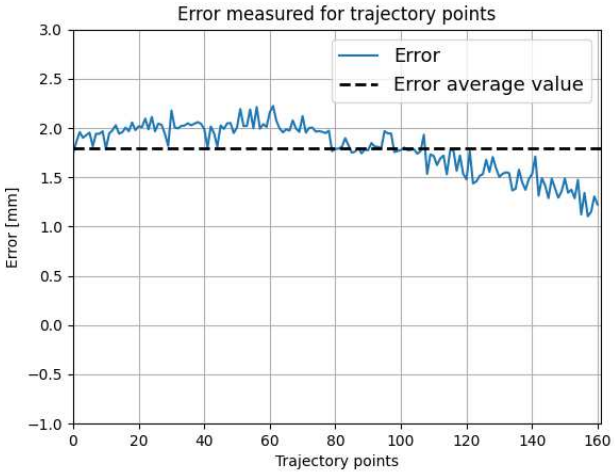
(a) curved object *(b) curved surface trajectory*

Figure 5.19: curved object and trajectory generated

The tests results are shown in figure 5.20. Where the figure 5.20a shows the comparison between the estimated height and the real measured height.
In figure 5.20b, the error is shown, and it has a mean value of 1.85 mm and minimum value of 1.2 mm and maximum value of 2.2 mm.
Comparing with the values of the plane surface, the reconstructed 3D model of the curved object has lower accuracy. The maximum error is obtained in the areas of maximum slope.



(a) Curved height comparison

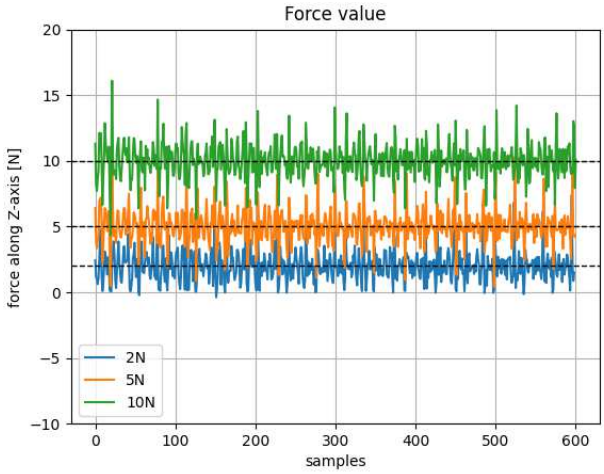


(b) Curved error

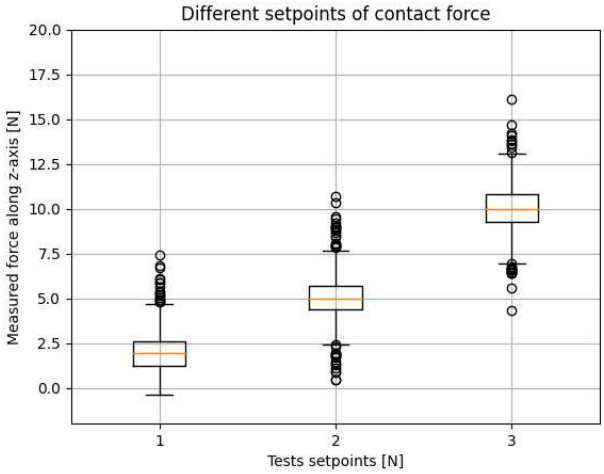
Figure 5.20: Curved object results

Contact force behavior evaluation

In the most of contact-based operations, the contact force applied by the tool over the workpiece surface is very important to guarantee the execution performance. For example polishing and surface treatment applications for example, constant contact is mandatory to have a consistent product quality.
In this subsection, tests are executed following the generated trajectory, applying different contact for forces on an object having a plane surface.
In figure 5.21 the contact force feedback is shown. The same working trajectory is followed applying different contact force set-points (2, 5, and 10 N).
In figure 5.21b the force values analysis is shown. The figure shows that the mean value in the three cases is equal to the set-point.



(a) Contact force readings



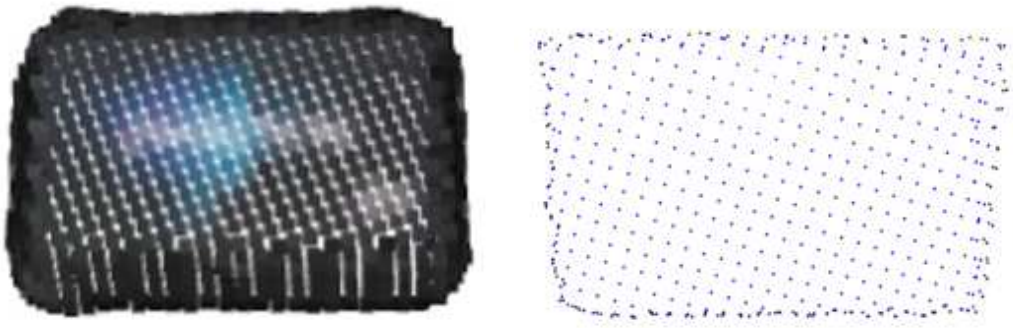
(b) contact force statistical summary

Figure 5.21: Contact force analysis applying different values

Area evaluation

Together with the depth values evaluation, it is important to evaluate the approximation efficiency of the generated 3D model. In figure 5.22, the box considered for this test is shown. The test consists in the isolation of the points constituting the upper face of the box and the area is compared with as shown in the figure 5.22b. Then the polygon connecting the outer points is found. The area enclosed within the polygon is found and compared to the area measured physically of the real box.

The figure 5.23 shows the output polygon found that surrounds the points forming the upper face of the scanned box. The measured area of the real box is equal to 134.8 mm^2 and the calculated area from the 3D reconstructed model is equal to 126.8 mm^2. That means that the 3D reconstructed model is underestimated than the real one by 6% in the area estimated.



(a) 3D reconstructed model of a box *(b) Upper face points*

Figure 5.22: 3D reconstructed model and selected upper face points

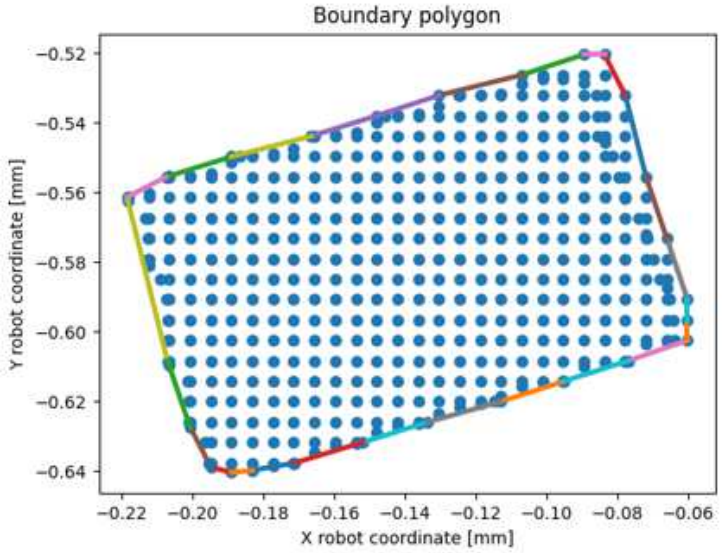


Figure 5.23: Outer polygon in which the points are enclosed

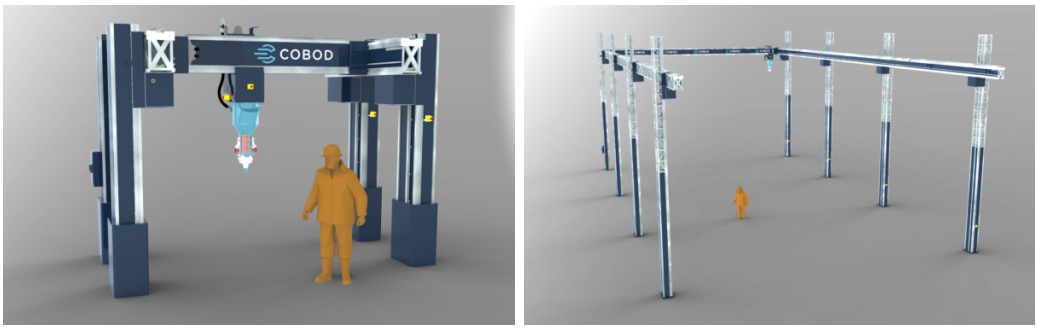
5.3 Quality monitoring in 3D concrete printing

The developed algorithm, could be applied in applications different from the generation of the working trajectory of robotic manipulators. It could be applied for quality monitoring applications to compare the actual state of a product "as built" with respect to its designed state. This design state can be in the form of a digital model (e.g. CAD model). These applications are extremely important in many fields, such as the field of 3D concrete printing.

5.3.1 Industrial context

The field of 3D concrete printing has been a highly growing field in recent years. Many researches have focused on technological aspects related to the optimization of the printing process, like the construction of the printing machine itself that could be a gantry system or robotic manipulators. Other investigated problems are the optimal material mix, finding printing parameters, evaluation of mechanical characteristics of the printed parts at their wet and hardened states [123, 124, 125].
In the current research on 3DCP, monitoring and evaluating the printing quality is a topic of high interest, because of the lack of systematic approaches to be followed. On site, the printing quality is highly dependent on many factors. The most important of them is the operator experience to set up the machine and dialing the right fundamental printing parameters - i.e. the movement speed and material flow rate, which accurately match the used material mix and the current environmental conditions such as temperature and humidity. Additionally, the printing quality can be affected by the basic machine accuracy and fluctuations in the quality of the extruded material and also external factors.
Quality monitoring of 3D concrete printing applications is necessary to guarantee the construction of the desired object with the required technical specification. The quality monitoring can be done at different productions steps. During the printing process, continuous monitoring is useful to adjust online the printing parameters to solve possible detected defects.
Post-printing quality monitoring is necessary to evaluate the mechanical characteristics of the print. The presence of holes, cracks due to material shrinkage, under filling or flow instability can highly affect the strength of the object, hence the maximum load that it can support.
Instead, the presence of extra material, material leakage, first layers tearing or layers closure pattern can affect the object geometrical accuracy. The printed object geometrical conformity is a crucial requirement in cases in which the printed objects have to be assembled, integrated in bigger structure or interface with other components.
Experiments done in this section are done in collaboration with COBOD Inernational A/S. It is a company providing innovative solution in the construction field, combining 3D printing and robotics technologies to automate the construction process. Printing solution could be based on the use of robotic manipulators or gantry systems.
The main product is BOD2 shown in figure 5.25 that is a modular gantry system

that can have different sizes based on the size and shape of the building needed to construct. The smallest version, BOD2 2-2-2 has a print area of 4.52 x 4.55 x 3.09 m. The biggest is BOD2 5-10-4 that has the print area about 12.10 x 24.75 x 8.14 m.



(a) Smallest printer BOD2 2-2-2 *(b) Biggest printer BOD2 5-10-4*

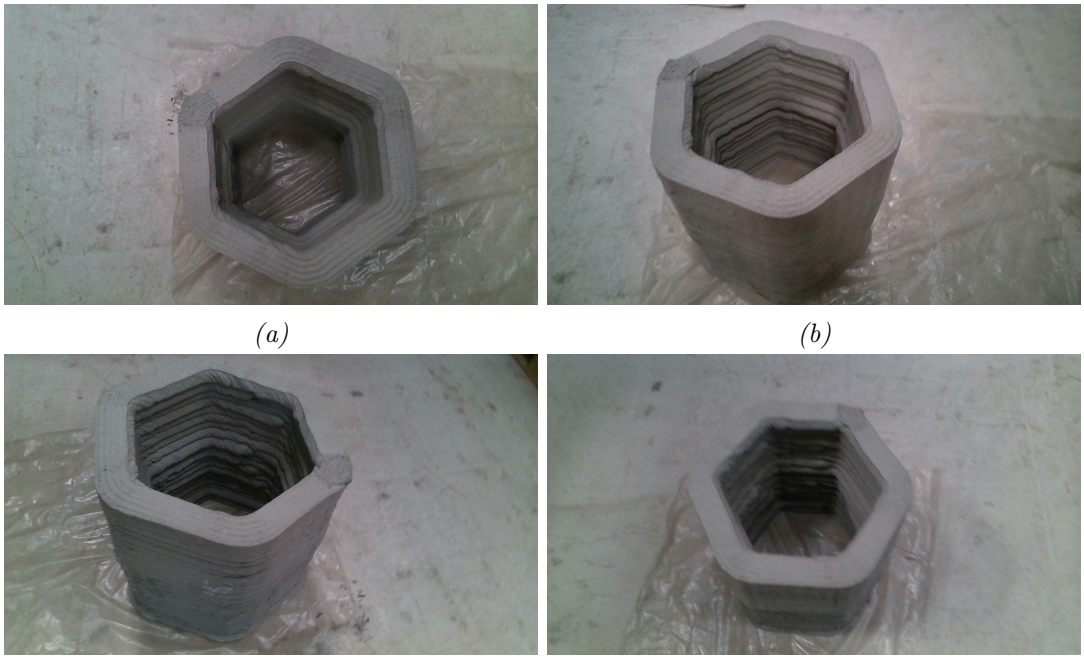
Figure 5.24: BOD2 models `https://cobod.com/bod2/`

Figure 5.25: COBOD printing process `https://www.linkedin.com/feed/update/urn:li:activity:6767460887042301953/`

5.3.2 Concerns applying Odometry-based 3D reconstruction technique in 3D concrete printing

The quality evaluation of a 3DCP object, that may not coincide perfectly with the starting CAD model used for generating the printing trajectory, could be done using the explained algorithm for the 3D model reconstruction of the object at his dry

state.
The quality evaluation procedure, consists in the reconstruction of the 3D model of the observed scene containing the printed object. The mesh model is then cropped to extract only the mesh of the object. The extracted mesh is then compared to the initial CAD model of the object to evaluate its quality to find the zones having similarities and zones having differences. The process in details is explained in the following way.
The acquisition procedure consists of the movement of the 3D camera around the workpiece or the 3DCP object considered acquiring synchronized pairs of color and depth images of it.
In figure 5.26 examples of the captured color images of a hexagonal shaped 3D concrete printed object are shown. Only four images are shown but to have a good performance of the algorithm and to be able to construct a 3D model describing the object from a 360°view, more images have to be captured of the object covering all the interested areas by the scanning. In this example 75 pairs of color and depth images are used for the reconstruction of the 3D model.



(a) *(b)*

(c) *(d)*

Figure 5.26: Example of color images taken of a 3D printed object

In figure 5.27, the reconstructed 3D model of the hexagonal shaped object is shown. The developed algorithm to integrate all the images available in the data-set to reconstruct the 3D model.
If the images cover also the floor under the object or other objects, these elements will be present in the reconstructed model as well. In the shown example, a portion of the ground is visible.

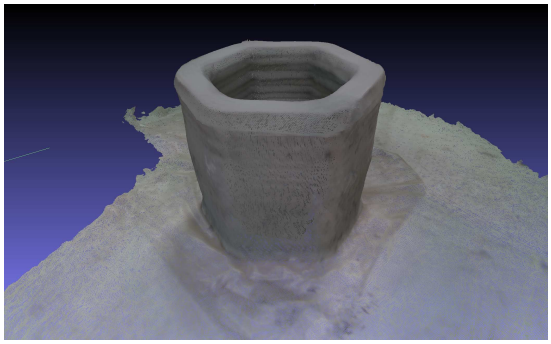


Figure 5.27: 3D reconstructed model of the hexagonal shaped object

5.3.3 Object extraction

To extract only the object that we are interested in, it is necessary to make some assumptions about the scanning and the printing processes. The first assumption is that the object is positioned, in most of the RGB-D images, in the center of the covered area. The second assumption is that the printing process happened over a flat surface.
With the first assumption, the search algorithm could consider only the center of the constructed 3D model of the scene. That means only a part of the model has to be analyzed. The second assumption allows considering that in the central area could be found the interested object and the floor on which it was printed. Searching in the model for a pane surface (points having the same height and connected together), it is possible to find all points combining the floor. Removing these points, it is possible to obtain the points of the object.

5.3.4 Registration of CAD and scanned point-cloud

To evaluate the reconstructed 3D model accuracy of the 3DCP object, it is compared with the CAD model. The comparison accuracy depends on the ability to compare an exact point in the scanned point cloud to the exact relative point in the 3D model. To guarantee good comparison accuracy, the two point clouds have to be aligned.
Alignment of point clouds, known as point cloud registration process, is a well known problem in computer vision and widely used in the construction to compare and inspect buildings, roads and civil infrastructures using laser scanners or similar scanning techniques. 3D registration process consists in finding the transformation matrix that if applied to the second model align it to the first one and also refer both of the models with respect to the same reference coordinate system.
Different registration techniques can be applied that can be categorized in coarse and fine registrations [103]. Coarse registration are feature based techniques that match the features in the two point clouds. The most used strategies are classified in point-based, line-based and surface-based. These methods are very accurate but highly dependent on the similarities level of the two point clouds that is dependent on the sensor noise, scanning point of view, point clouds density and geometrical nonconformity due to fabrication process accuracy. To guarantee high accuracy,

complicated feature extraction tools may be necessary to develop.
Fine registration techniques are based on approximate iterative processes to find the optimal rigid transformation matrix between the two point clouds. The most used technique is Iterative Closest Points ICP [101, 104] and its variations. These techniques are based on the minimization of positional errors of relative point sets selected from the two point clouds. This implies the necessity for a good initial guess of the rigid body transformation to avoid local minimum problems. Elaboration time is dependent on the point-cloud size.
In the case of 3DCP objects, using the above techniques leads to registration failure. This because of the low geometrical conformity of the printed object due to printing accuracy tolerance. Also, difficulties to identify features like lines or surface in the noisy scanned point-cloud. And finally, lack of initial transformation guess to be used for ICP techniques as the scanning process is done manually.
To overcome these limitations, we propose an easy and fast to apply registration technique to align the two point clouds based on more general features considering the overall point clouds. These features are the boundary box and the three-dimensional center of the object.

5.3.5 Experimental test and result in quality monitoring

The developed algorithm is applied over different kind of 3D printed objects, as is shown in the following paragraphs.

Pillar

Pillars or column are from the main construction components in all the buildings. They are vertical supports and intended to be load-bearing components to distribute ceiling weight to the floor or the structure below.
Pillars prefabrication could decrease meaningfully the construction time. As they are active construction components, an accurate quality monitoring process has to be carried on to guarantee high load resistance and geometrical conformity has to be evaluated to be able to integrate them with other construction components. Pillars may have also added details for decorative purposes, making them difficult when digital-based quality monitoring are used.
The developed solution is applied to construct the 3D model of the pillar shown in figure 5.28.
The pillar has a length about two meters and an extension of reinforcement bars about 30 centimeters from both ends. Due to the maximum and minimum distances of the 3D stereo camera used with respect to the scanned object and due to the length of the pillar, it was not possible to capture any image covering all the parts of the pillar but different parts of it.
Using RGB-D Odometry technique explained in section 5.1.2 a set of 100 color and depth images like those shown in figure 5.28 are integrated to reconstruct the 3D model shown in figure 5.29.

Figure 5.28: Input images of a pillar

Figure 5.29: Output images 3D model of a pillar

Full 3D model reconstruction

For relatively smaller objects it is necessary to have a 3D reconstructed model covering them from all the possible angles of views as they will be assembled with other components.
Several scans were done to capture images that cover the 3DCP objects from all possible point of vies to be able to construct the full 3D model. The test object used is the elliptical cylinder shown in figure 5.30. Upper side images are the capture color images and lower side images are showing the constructed 3D model from two point of views. The dimensions are semi-axes of 0.4 and 1.5 meters and height of 0.6 meters. Print defects like cracks due to layer closure pattern are clearly visible in the last layers.
The developed 3D reconstruction pipeline is applied also to scan the ring shaped 3DCP object shown in the sub-figure 5.31b and the hexagon shaped 3DCP object shown in figure 5.27. On the left side four color images are shown and on the right side the reconstructed 3D model is shown.

5.3.6 Error analysis

Hausdorff Distance is used to compare the 3D reconstructed model of the print's actual result and the ground truth or the 3D CAD model of the object used to generate the trajectory of the 3DCP.
Hausdorff distance measures the magnitude of the biggest printing defect causing low resemblance between the printed object and the CAD model. To calculate Hausdorff distance, it is necessary to calculate all the distances between relative points in the two point clouds and find the maximum of them.
To calculate the Hausdorff Distance, MeshLab [94] is used. MeshLab is an open

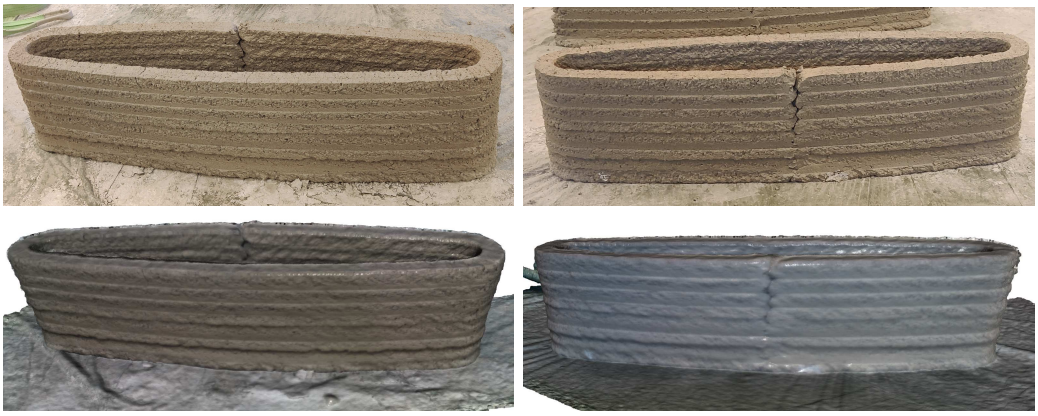
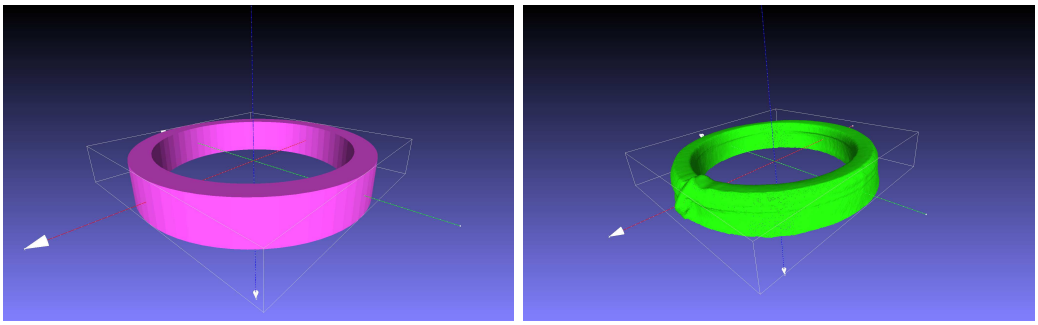


Figure 5.30: 3D printed object 3D reconstruction

source 3D model elaboration software. It allows calculating, between two aligned and down-sampled point clouds, maximum distance (Hausdorff distance), minimum distance, distance mean value and Root Mean Square error.
The object considered for the error analysis is the cylinder shaped printed object shown in figure 5.31. The sub-figure 5.31a represents the CAD model or the ground truth to which the reconstructed 3D model shown in the sub-figure 5.31b is compared.
The considered object presents a visible printing defect due to the printing layer closure. It is shown in the down left side of the sub-figure 5.31. Results obtained comparing the total of, 159039 relative points are: Hausdorff Distance of 0.039 meters in correspondence with the mentioned defect. A calculated null distance for most of the points of the upper face, hence high printing accuracy. Distances' mean value of 0.0057 meters and a Root Mean Square of 0.0073 meters.
In figure 5.32, all distances between the relative points of the two compared point clouds are projected on the ground truth CAD model. The distances magnitudes are represented using a color scale having red minimum values and blue maximum values. On the left side of the image, a histogram is shown to represent all the distances and their magnitude. Almost all the points have errors lower than 0.019 meters, and very few points have higher error values.



(a) CAD model *(b) Constructed 3D model*

Figure 5.31: Comparison between CAD model and output 3D model of the actual print

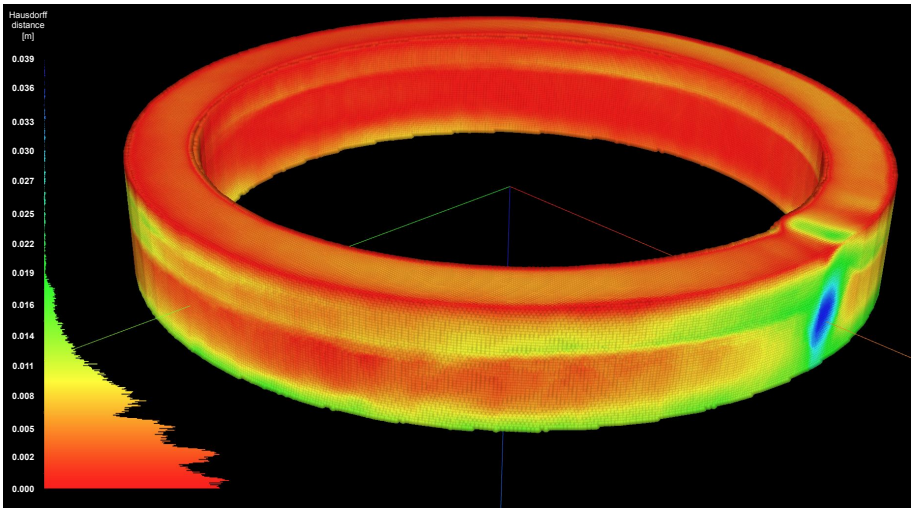


Figure 5.32: Hausdorff distance comparison applied to the ring case

5.4 Discussion and limitations

In this chapter, we propose a framework for trajectory planning for robotic contact-based operations. The solution is based on the integration of multi-view RGB-D images of the workpiece captured using a low cost 3D camera. The images are integrated to generate the 3D model without previous knowledge of the workpiece. Experimental results show how to tune relevant parameters that may improve the accuracy of the reconstructed the 3D model and hence the trajectory. Tests done show also that the proposed solution could be applied over planar and curved workpieces. Applying contact force control to keep constant the force applied between the tool and the object surface allowed to compensate the positional errors between the generated and the actual trajectory. In industry, having constant contact force following the trajectory allows for having a consistent performance of the robotic system executing the contact-based operation.
The developed algorithm, has been used in a different application where the 3D reconstruction of objects at their produced state is necessary to be compared to the initial 3D CAD model. The considered application is the 3D reconstruction of a 3D concrete printed objects to evaluate the accuracy with respect to the CAD model.
The introduced 3D reconstruction technique named Odometry-based technique, requires, for obtaining good reconstruction results, the necessity to have visible almost every part of the workpiece in several images. Many RGBD images (more than 50 images) are also required for the reconstruction. The number of images is proportional with the computational time. The elaboration time, considering a data set of 100 images, is around 15 minutes. The longest step of the developed algorithm is one in which the camera poses are estimated. This is done comparing the pixels positions in the several images. To overcome this limitation of having this time-consuming process, different algorithm for the 3D reconstruction is introduced in the next chapter.

Chapter 6

Robot poses-based 3D model reconstruction and robot trajectory planning

In this chapter, a different solution for the reconstruction of the 3D model of a workpiece is introduced, together with the steps necessary for the trajectory generation in contact-based robotic operations. This technique, named robot poses-based 3D reconstruction, is proposed to optimize the performance of the previously introduced Odometry-based technique. The optimization, considers the decrease of the number of images necessary for the reconstruction and hence the elaboration time. The performance of the developed algorithm is evaluated in the generation of the working trajectory to be followed in a contact-based robotic application.
A comparison between the two 3D reconstruction algorithms is done to evaluate the performance and quantify the differences between them, considering parameters that may influence their usage in industrial applications. The comparison is done considering as Key performance indices (KPIs), acquisition time, elaboration time and accuracy of the reconstructed 3D model when compared to the CAD model of the workpiece.

6.1 Robot poses-based 3D reconstruction

An essential step in the developed algorithm in the previous section is the estimation of camera poses while acquiring each image. The poses values are necessary to be able to integrate the poses in a single volume representing the observed scene. When the 3D camera is attached to the robot end-effector, it is possible to calculate the pose of the end-effectot and hence the 3D camera pose while capturing every image. In this way, the integration process can be done in shorter time and using fewer data.
To scan the workpieces, the robot moves the 3D camera following a predefined trajectory, orienting the camera towards the workpiece in all the images. The camera start acquiring from the top image and then moves to cover all the four faces of the object. The trajectory chosen is the same for the two developed techniques, but the acquisition sequence and timing are different in the two cases.

To control the robot movement commands to follow the defined trajectory that points the 3D camera towards the working area of the robot from different points of view and to simultaneously control the 3D camera to capture the necessary color and depth images, a software is developed in Python programming language. The software code is based on the use of Open-source libraries that are Librealsense, Open3D, OpenCV and TMflow for cobot programming.
The developed software consists in acquisition phase, elaboration phase and execution phase. The summary of the developed software is explained in the flowchart shown in figure 6.1. In the acquisition phase, the robot has to move consequently between the predefined scanning poses. These scanning poses are designed to have always the 3D camera pointed towards the workbench and to scan it from the top and from the four sides. In this way it most probably to scan the workpiece placed on the workbench and in the robot reachable area. At every scanning position, as soon as the robot reached the position, the 3D camera is switched on. After a waiting period of time to guarantee the complete stop of the robot and that the camera is switched in, a color and depth images are taken of the workpiece. This process is repeated for all the scanning poses.
In the elaboration phase, knowing already the poses of the 3D camera at the moment of capturing each image. It is possible to run the integration process based on the volume integration and on the truncated distance function explained in 5.2.4. Once the 3D model is reconstructed, the elaboration phase and the trajectory planning and execution are similar to the case based on the Odometry technique.

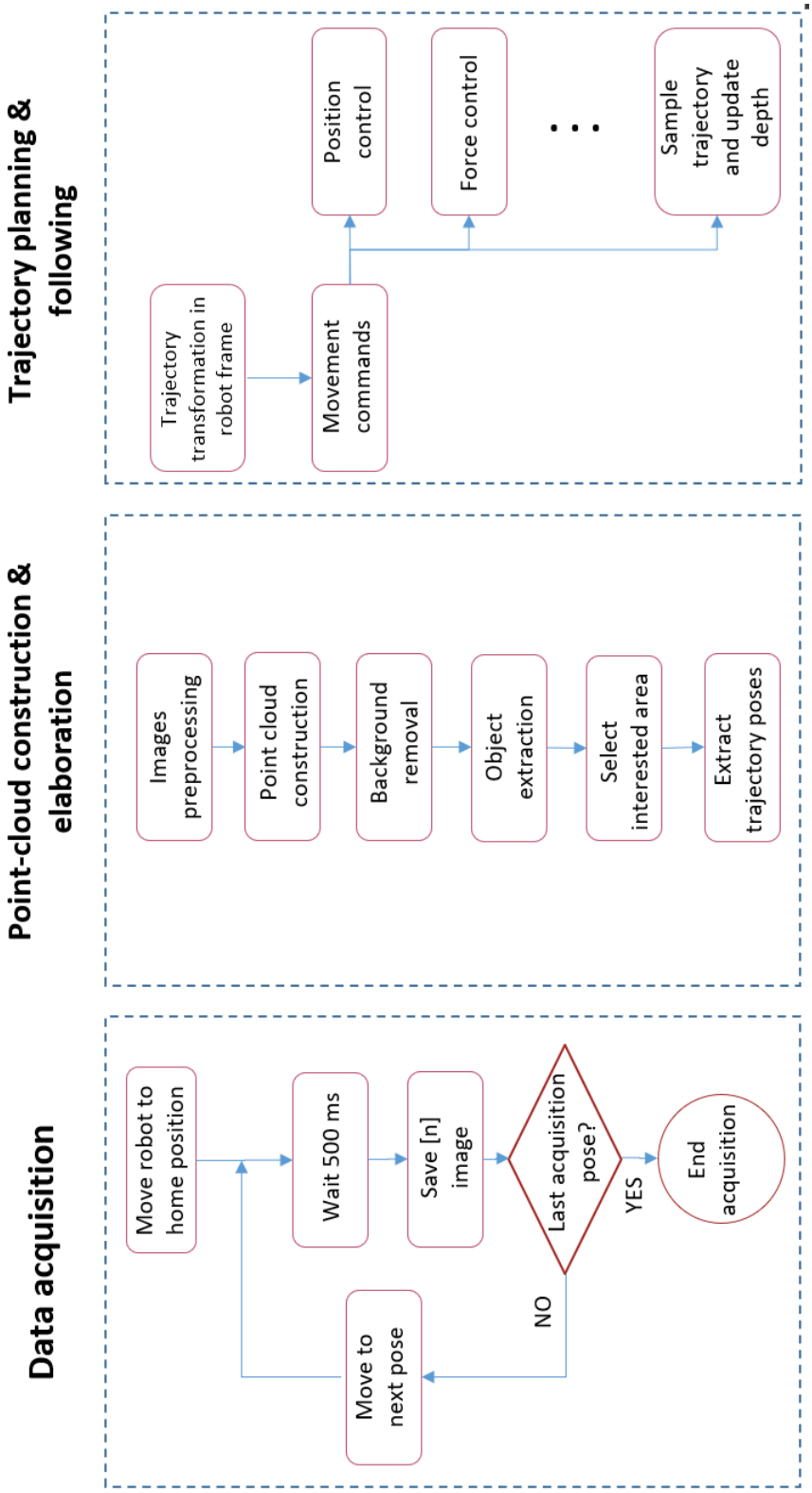


Figure 6.1: Developed software flowchart

6.1.1 Experimental setup and acquisition poses definition

To capture the dataset of RGB-D images, the robotic setup shown in figure 6.2 is used. The setup consists of a Techman TM5 6-axis collaborative robot. Intel Realsense D415 stereo depth camera is attached to the robot end-effector.

Figure 6.2: Setup

The 3D camera uses different sensors to acquire color and depth images of the scene in its field of view. The resolution of the sensors can be configured up to 720x1280 pixels and a frame rate up to 30 frames per second. Based on the resolution chosen, a minimum distance has to be respected between the camera and the observed scene to guarantee the measurement of the depth. The configuration used is the highest resolution of 720x1280 pixels that implies a minimum distance of 450 mm from the workpiece. The two algorithms are developed in Python programming language and using Open source libraries that are Librealsense for images acquisition and elaboration and Open3D for 3D data elaboration [93].
To evaluate the quality of the results obtained using this method, two different tests are considered. The first is to evaluate the quality of the reconstructed 3D model. The second test is to evaluate the accuracy of the generated trajectory. For the evaluation, a wooden object used for the fabrication of sailing boat is used as example for workpiece that can be found in Small and Medium Enterprises. As a final production step, the wooden object has to be exposed to a surface finishing process using a sanding tool 2.4.3. The considered workpiece is the one shown in figure 6.3.

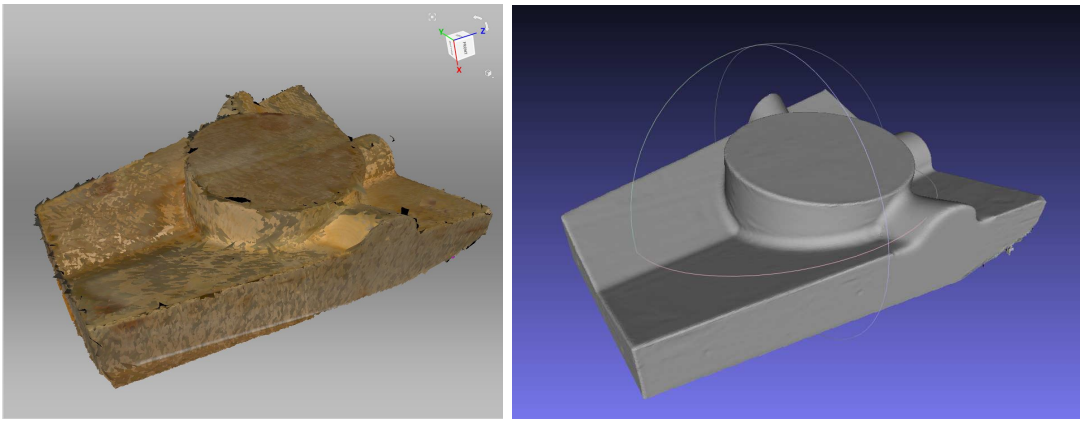
6.1.2 Accuracy evaluation of 3D reconstructed model

An accurate model with high degree of similarity, with respect to the actual state of the workpiece, is essential for being able to generate an accurate working trajectory.

Figure 6.3: Wooden item

In this section, a professional structured light scanner (Artec Eva scanner 2.2.3) is used to scan and construct a 3D model of the workpiece.
The evaluation is done by comparing two 3D reconstructed models with the CAD model of the object. The first is made using the proposed method based on the robot poses knowledge. The second model is made by using the Artec scanner and using the professional software Artec studio for images integration.
Figure 6.4 shows the output 3D model using the proposed method based on the use of the known camera poses.
Using the professional scanner allows integrating the multi-view color and depth image to reconstruct the 3D model of the workpiece. The output model is always affected by the sensor noise and have a lot of inaccuracies describing the surface. That make it not suitable for the accurate comparison done. The Artec studio has a built-in tool to filter and remove these inaccuracies. In figure 6.5. The output model is shown in figure 6.5a, and it is possible to note several areas affected by the sensor noise, especially in the edges. In figure 6.5b, the model after applying the tool is shown. That model is the one used for the comparison with the CAD model of the workpiece.
Using Hausdorff Distance technique, a comparison between everyone of the reconstructed 3D model and the CAD model is done. Hausdorff distance function calculate the error or the distance between every point in the reconstructed 3D model and the relevant one in the CAD model. The result can be represented visually using a color representation and a histogram. The color map represents the low error by red color and for increments in the error values the color moves to yellow, green and blue for maximum values. On the left of the images, a histogram is shown to represent the number of points for each color and error value.
Figure 6.6, shows the comparison accuracy. On the left the robot-based technique and on the left the professional scanner. Considering the points having an error up

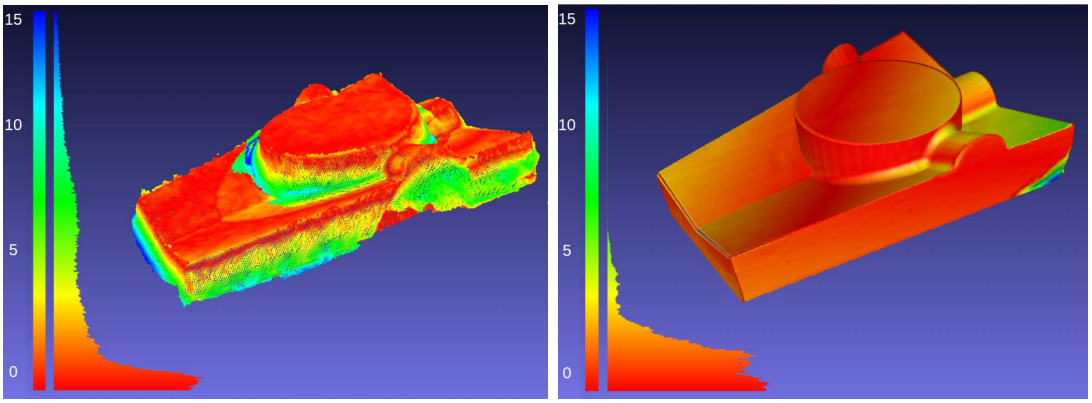
Figure 6.4: Robot-based 3D reconstruction output



(a) Output with noise *(b) Filtered mesh model*

Figure 6.5: Output model using Artec scanner before and after filtering

to 15 mm, Robot-based error has a mean value of 3.890 millimeters, a minimum value of zero millimeters and a root-mean-square error of 5.37 millimeters. While the model, reconstructed using the scanner, has a mean value of 0.987 millimeters, a minimum value of zero millimeters and a root-mean-square error of 1.31 millimeters. Despite the differences in the statistical analysis, the histograms show that in the two models, most of the points have error values lower than 2 millimeters.
The professional scanner and the professional software for 3D reconstruction have a better performance. The difference in the performance is reasonable to the accuracy values of the hardware devices used (low cost Realsense D415 camera and the high-end structured light scanner).

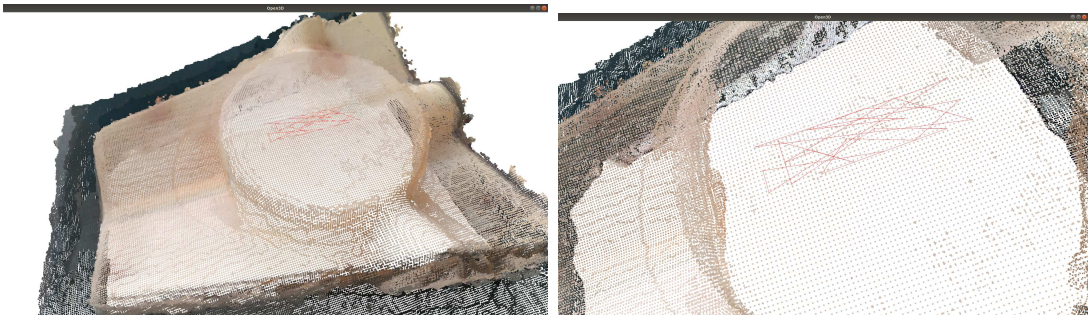


(a) *(b)*

Figure 6.6: Wooden object analysis. a)Robot-based reconstruction accuracy, b)Scanner-based reconstruction accuracy

6.1.3 Accuracy evaluation of the working trajectory

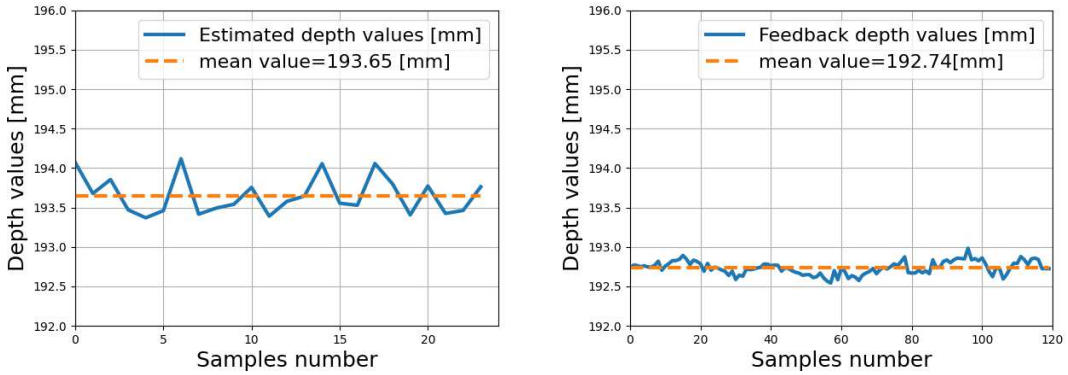
For a contact-based operation, the accuracy of the generated trajectory is crucial for its success. Considering the 3D reconstructed model using the robot-based technique, a trajectory following the upper surface of the cylinder part is generated to be used during the accuracy evaluation process. The used trajectory is the one shown in the figure 6.7. It is highlighted by a red color.



(a) *(b)*

Figure 6.7: Trajectory generated

The evaluation consists in following the generated trajectory applying force control loop to control the contact force between the tool center point (TCP) and the workpiece along the z-axis. The contact force is set at 5 N. Applying the force control allows adjusting the errors in the depth and by measuring the feedback position of the TCP it is possible to compare the estimated value and the real ones.
In figure 6.8b, the estimated trajectory and the real one followed by applying the force control room is shown. Comparing the mean values of the two trajectories, it is possible to calculate the error of the generated trajectory. The mean value of the generated trajectory is 193.60 millimeters, while the mean value of the actual trajectory is equal to 192.74 millimeters. The error is 0.86 millimeters.
The force value during the following of the generated trajectory is shown in the figure 6.9. The set point is equal to 5 N while the mean value of the force feedback



(a) *(b)*

Figure 6.8: Trajectory comparison between: a)Estimated trajectory and b)Real trajectory

recorded is 4.70 N. That make it slightly lower than the desired value by 0.3 N.

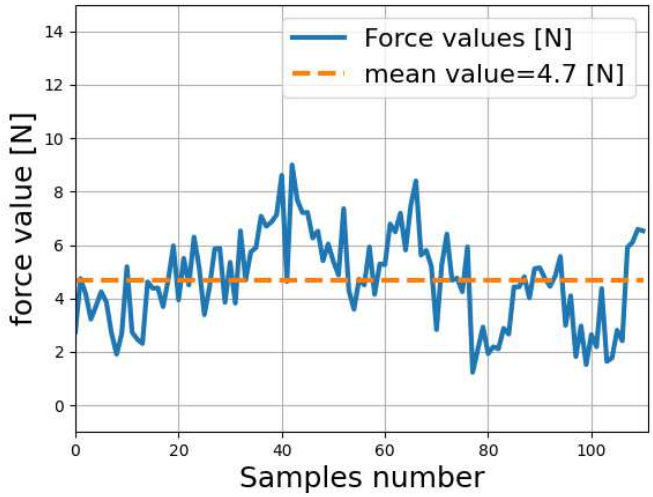


Figure 6.9: Contact force values

6.1.4 Discussion

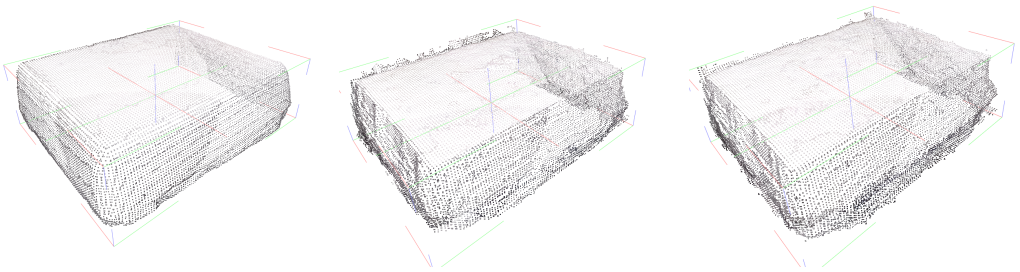
Using the robot poses values, and hence the camera poses, during the acquisition of the allowed to determine the camera movement between the pose at every image to be able to transform it with respect to a common reference frame. These frames are integrated together to reconstruct the 3D model of the observed scene.
The reconstructed 3D model is compared to the CAD model of the workpiece and to the 3D model reconstructed using a professional structured light scanner. The reconstructed models obtained similar results and had for most of the points an error lower than 2 millimeters. An evaluation of the accuracy of the generated trajectory is executed, and the generated trajectory has a mean value error lower than 1 millimeter.

6.2 Comparison between Odometry-based and robot poses-based techniques

In this section, the accuracy of the two algorithms introduced in 5.2 and 6.1 are evaluated, comparing the 3D models reconstructed of both of them with respect to the ground truth or the known CAD model. The robotic cell setup is the one used in 6.1.
The evaluation is based on three Key Performance Indices (KPI) that are acquisition time, elaboration time and geometrical accuracy with respect to the CAD model.
The object considered for the evaluation process is a metal cube of the dimensions 127 mm * 99 mm * 40 mm that is shown in the figure 6.10. This kind of object is mostly to be found in manufacturing systems on which it is necessary to apply a surface finishing procedure.

Figure 6.10: Metal box

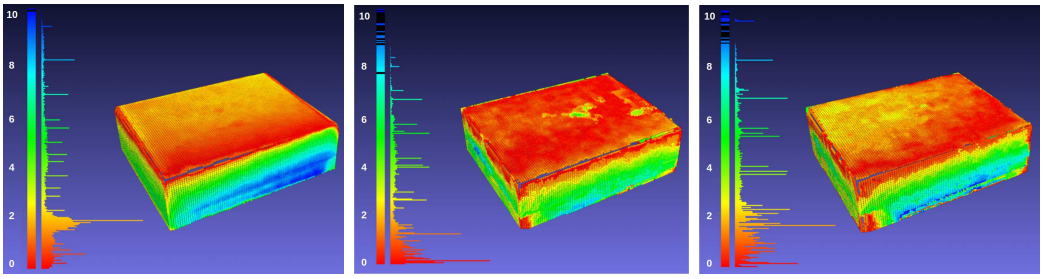
The first comparison is done between the 3D model reconstructed using RGB-D Odometry-based algorithm. The algorithm integrate 100 pairs of RGB-D images captured of the static workpiece by a moving camera attached to the robot end-effector.
The second comparison is done comparing the 3D reconstructed model of the box using the poses-based algorithm with the 3D CAD model of the workpiece. The pose-based technique is based on the use of minimum number of images and hence minimum elaboration time and needed resources. To be able to cover all the faces of the workpiece, the minimum possible number of images is five. That are covering its top face and other four images covering all the four faces.
We consider also other scanning procedure using eight images of the workpiece. In which, beside the minimum possible number of five images, we add extra three images covering the corners of the metal box. In this way, all the faces are visible in two images to limit the dependency on possible acquisition noise and reflection of the metal object.
Figure 6.11 shows the obtained 3D models applying the different 3D reconstruction algorithms. In figure 6.11a the Odometry-based algorithm result is shown. In figures 6.11b and 6.11c shows the results of the poses-based algorithm with five and eight images.
To quantitatively evaluate the geometrical errors between the models, we use Haus-



(a) *(b)* *(c)*

Figure 6.11: Reconstructed 3D models. a)Odometry-based, b)Poses-based 5 images and c)Poses-based 8 images

dorff distance to measure the distance between all the points of the point-cloud of the constructed 3D model and the relative point in the point-cloud of the known CAD model of the metal box. To apply the Hausdorff Distance algorithm, it is necessary to align both of the 3D models and to overlap them. The reconstructed 3D model is always not identical to the 3D CAD model because of 3D camera acquisition accuracy and presence of extra light or object reflections. Applying Iterative Closest point (ICP) automatic registration algorithm for the alignment may fail. To overcome this problem, an alignment procedure based on the alignment of the boundary boxes of the two 3D models is used.
The result of applying the Hausdorff Distance algorithm to two point clouds is represented graphically in a color-map. The color-maps for the three reconstruction cases are shown in figure 6.12. In the map, the points that have small errors are represented by a red color. With the increase of the error, the representation color transforms toward yellow and turns blue for maximum error values. The represented color-map is between 0 mm for red color and 10 mm for blue color. The histogram shown on the left of each image shows the number of points having a certain error level. The error color map can be projected on the point-cloud of the point-cloud of the CAD model. In this way, a significant graphical representation is obtained to visualize the areas where the two point clouds coincide and also where they mismatch. For all the three models, most of the points have error lower than 2 mm. The upper face is the face that is having the minimum error and coincides approximately with the real CAD model. Lateral faces present higher errors. In the reconstructed 3D model using the Odometry-based technique, in figure 6.12a, the right lateral face is overestimated in correspondence to the bigger blue area. When Poses-based reconstruction is used, edges are reconstructed having more realistic values closer to the real value of the real CAD model and this is shown in correspondence with bigger red areas covering the edges and the corners.
Despite the fact that the 3D model reconstructed using Odometry-based technique seems smoother than those reconstructed using Poses-based approach as shown in figure 6.11. The error analysis using Hausdorff Distance showed that Poses-based approach is more realistic.



(a) *(b)* *(c)*

Figure 6.12: Error representation using Hausdorff Distance. a)Odometry-based, b)Poses-based 5 images and c)Poses-based 8 images

	Acquisition time [s]	Elaboration time [s]	Measurement accuracy [mm]		
			min	mean	max
Odometry-based (100 images)	13.25	864.00	0.0009	3.1588	10.4879
Poses-based (5 images)	10.12	5.70	0.0002	2.4448	10.3795
Poses-based (8 images)	14.15	10.02	0.0001	2.5834	9.1436

Table 6.1: Result comparison

The time necessary to use the different techniques is an important evaluation KPI for industrial applications to decrease the cycle time of the 3D reconstruction process. We consider different periods of time for our evaluation. The first is the acquisition time or the time necessary to capture the required data that allow the 3D reconstruction. For the two techniques, the acquisition trajectory followed by the robot is the same. In Odometry-based technique, the robot follows all the trajectory and the camera acquires several images of the object during the movement. In Poses-based technique, the same trajectory is followed by the robot with the stop periods in predefined positions that allow to capture images that cover the workpiece from the interested faces.
Elaboration time is also used for the evaluation. This is the period of time needed to elaborate the images and to integrate all the information included in them to reconstruct the final 3D model. In Odometry-based technique, higher number of images are used and long optimization procedure is used to estimate the camera poses while capturing the images' dataset. Hence, its elaboration time increases significantly with the increase of the dataset dimension. Poses-based algorithm integrates directly the information from the images to reconstruct the 3D model, since the poses to capture the images are already known.
Table 6.1 shows a numerical comparison of the three cases considered. For each case, acquisition time, elaboration time and measurements accuracy are shown.

Acquisition time for all the cases presents similar values. The highest value is in the case of eight images, due to the necessity to stop the robot to capture each image for 500 ms to allow the acquisition of an RGB-D stable image. Elaboration

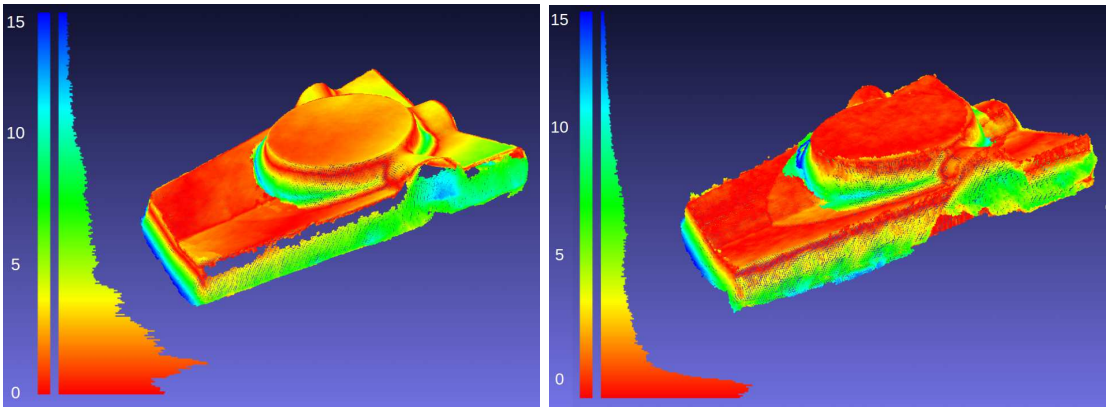
time decreases significantly using the poses-based method, around 10 seconds, with respect to the Odometry-based case where the elaboration time is more than 14 minutes.
The table also shows the accuracy evaluation summary using Hausdorff Distance technique. Mean value of geometrical error decreases from 3.15 mm in the case of Odometry-based method to approximately 2.5 mm in case of Poses-based method. Adding more images to cover all the faces in at least two images allows decreasing the maximum error from 10.37 mm to 9.14 mm.

6.2.1 Application to an actual component in manufacturing industry

In this section, the two techniques are used to reconstruct the 3D model of a wooden object. The considered object, with respect to the metal box, represents more complicated features in terms of curves, edges and irregular color distribution. The object considered is the one shown in figure 6.13, used as a mold for sailboat fabrication.

Figure 6.13: Wooden item

The resulting 3D models are compared to the 3D CAD model of the object to evaluate their accuracy. Hausdorff Distance technique is used to quantify the geometrical errors. The results are shown in figure 6.14.
The color maps show the error for the Odometry-based technique in figure 6.14a and for poses-based in figure 6.14b. The error values represented are between zero and 15 mm. The histogram on the left of each color-map shows the quantity of the points at every error level. Using both of the techniques allows having a high number of points with error lower than 5 mm and very few points with error higher than 10 mm.
Considering the points having an error up to 15 mm, Odometry-based error a mean value of 4.212 mm while for Poses-based algorithm the mean value is 3.890 mm.
Hausdorff analysis, shown in figure 6.14, shows that the two models are similar in the most of the workpiece. Odometry-based technique uses average depth values of



(a) *(b)*

Figure 6.14: Wooden object analysis. a)Odometry-based reconstruction accuracy, b)Poses-based reconstruction accuracy

many acquisitions to calculate the final value for every voxel. This allows to obtain a smoother flat surface and smoother and overestimated edges. For the areas with lower visibility it tends to overestimate them as shown in correspondence of the blue and yellow areas.

6.3 Discussion and limitations

The proposed technique, robot poses-based 3D reconstruction, allows generating a 3D model of the workpiece using few synchronized pairs of color and depth images covering the workpiece. Knowing the end-effector pose at every view point, allows calculating the camera pose. These camera poses are used to integrate the RGBD images content in a unique 3D model describing the scene visible in the images.
The accuracy of the algorithm, is evaluated considering the accuracy of the 3D reconstructed model, comparing it to the 3D model reconstructed using a professional laser scanner. In the comparison results, errors between the relative points of the two models are lower than 2 mm. The accuracy of the generated trajectory is also evaluated, by controlling the robot to follow the workpiece surface. Applying force control while following the generated trajectory, allowed to obtain contact force of a mean value about 4.7 N while a set point of 5 N is considered for the force control loop. The control loop compensated the error of the generated trajectory and adjusted it to keep contact while following the trajectory. The difference between the mean value of the generated and actual trajectory is lower than 1 mm.
Comparing the performance of the two techniques for 3D reconstruction, the acquisition time decreases from 13.25 to 10.12 seconds. While the elaboration time decreases significantly from 864 to only 5.7 seconds. Also, the accuracy of the generated 3D model, improved in terms of error mean value that decreased from 3.15 to 2.44 mm.

Chapter 7

Conclusions and future work

In small and medium enterprises, the production has been moving towards small volume and highly customized production. Some production steps such as contact-based operations still done manually by expert human operators. The full automation of these tasks, in the modern dynamic industrial environments, require the introduction of flexible tools that allow the task configuration and design within the minimum possible time with a limited or without the need for programming experience to be done by operators. In this thesis, different robotic solutions are proposed to automate constact-based operations to satisfy flexibility and high performance requirements regardless of high product customization.
Trajectory planning in contact-based robotic operations is done mostly using techniques based on the elaboration of a CAD model of the workpiece. These techniques allow generating accurate trajectories but have the drawback of being limited to the degree of similarity between the CAD model and the actual state of the built product that in many cases differ. In case of huge product customization and variations, the use of CAD-based techniques would slow down the production since the trajectory generation is based on CAD model that is not always available.
The proposed solutions for trajectory generation of contact-based robotic operation use only the data acquired by a low cost 3D camera (Intel Realsense D415 and D435). Different important aspects are discussed starting from the cell layout design, focusing on the effect on the 3D camera positioning and the ability to execute certain contact-based task.
An algorithm based on the use of a fixed 3D camera is introduced, and its performance was tested in a glue deposition application in footwear manufacturing. Other cell layout in which a 3D camera is attached to the robot end-effector is introduced and analyzed. This second layout allows scanning the workpiece from different view points. In this way applications over lateral faces can be executed in contrast with the other cell layout in which contact-based operations can be only executed on the visible face to the 3D camera.
To have a complete knowledge of the workpiece that allows to optimize the trajectory generation process, the 3D model of the unknown workpiece considered for the robotic application can be reconstructed by integrating the color and depth images acquired of it from different perspectives. An algorithm for the 3D reconstruction using the data acquired by a manually moved 3D camera is developed. The perfor-

mance of this algorithm is evaluated, applying it in the field of 3D concrete printing for quality monitoring of the printed objects.
The developed algorithm is based on the use of feature matching between the colored images to estimate the camera motion or the camera pose while acquiring each image. These poses are used to transform all the captured images to be with respect to a common reference frame. After that, a merge process is executed to integrate all color and depth information to generate a unique 3D model representing the observed scene.
To generate the trajectory for robotic contact-based operations, the developed algorithm for the 3D reconstruction is modified to be used in a robotic application. The new reconstruction algorithm is based as well on computer vision techniques, and called Odometry-based technique, to estimate the camera pose and then merge the images in a common volume. Once the 3D model is reconstructed, a developed searching algorithm is used to isolate only the workpiece from the scene and then select the area considered in the operation based on the user input. This robotic solution is evaluated based on the performance of factors that affect the accuracy of a contact-based operation, such as spatial error between the estimated trajectory and the real trajectory with respect to the real position of the workpiece. The other important criterion is the ability to keep the exerted contact force as constant as possible following the generated trajectory.
The developed algorithm is optimized to decrease the computational time to just 10% of the time needed for the reconstruction using the Odometry-based technique. The new version of the 3D reconstruction, called Robot poses-based, algorithm is based on the of fewer color and depth images that are captured from a predefined set of poses in which the 3D camera is pointed differently to cover a face of the workpiece. For example, to cover almost all the workpiece, at least five images are needed (one from the top and four covering the four faces of the workpiece). Knowing exactly the robot end-effector pose while capturing every image, make it easy to calculate the pose of the 3D camera as well. These calculated poses are used to calculate the homogeneous transformation matrices that can refer all points of each image to be with respect to a common reference frame. An integration algorithm is used to merge all the data from the different images in a common volume. The same evaluation process is done as the other case to evaluate the accuracy of the reconstructed 3D model, spatial error between the generated trajectory and the real one and contact force behavior. Finally, a comparison between the Odometry-based and the robot poses-based 3D reconstruction techniques are compared using key-performance indices (KPI) such as acquisition time, computational time and accuracy of the 3D reconstruction model with respect to the CAD model of the workpiece.
To evaluate the performance of the proposed solutions, a state-of-the-art solution for trajectory generation in robotic contact-based applications, based on the use of the CAD model of the workpiece, has been developed. The performance of the proposed solutions based on the use of only vision systems have been compared to the performance of the CAD-based solution considering trajectory accuracy and contact-force behavior following it.
In industry and in literature, the 3D reconstruction of workpieces is done by exploit-

ing high-end laser scanners. To assess the accuracy of the low-cost 3D vision system used in this thesis, and the accuracy of the proposed algorithms for the 3D reconstruction, a professional laser scanner combined with a commercial software for 3D reconstruction have been used to reconstruct the 3D model of the workpiece allowing the comparison between the proposed method and the state-of-the-art methods.
The generated trajectories based on the use of the developed 3D models obtained good performance of errors with reasonably low mean value. The positional errors are due to the low resolution and the noise of the 3D sensor and the soft material of some workpieces. Integrating the force control while following the generated trajectory allowed to compensate online and keep a constant contact force with a mean value equal to the desired set point.
The proposed solutions allow the automation of contact-based operations in small and medium enterprises. These solutions provide more flexibility with respect to traditional techniques since they do not require the presence of the CAD model of the workpiece and can be used to execute the tasks on unknown workpieces. Using low-cost 3D cameras in the development of the introduced solutions aims to enhance the automation of more and more production steps in SMEs by providing high performant solutions at low-cost. The use of only the data acquired by the 3D camera and exploiting the robot poses to accelerate the 3D reconstruction of the workpiece, would accelerate the cycle time of the contact-based operations and hence increase the production without influencing the production customization.

7.1 Future work

- Integrate the proposed solutions for flexible trajectory planning with techniques that enhance the safety of the robotic cell. This is done to design collaborative robotic solutions, that allow the co-presence of the robot and the human operator in the same workspace. This situation allows the production of highly customized products by exploiting the skilled human operator input. That input allows executing also online adjustments to overcome errors or to improve the performance.

- Optimization of the developed algorithm for the 3D reconstruction to decrease the computational time and reach a near real-time performance. Fast reconstruction of 3D models can accelerate the process cycle time for contact-based operations and also for other applications such as collision avoidance.

- Application of the developed algorithm for the 3D reconstruction on data acquired by high-end 3D camera. This can help to evaluate the trade-off between the camera low price and high accuracy

- Implementation of post-processing algorithms to fix the reconstructed 3D model in terms of hole filling and surface smoothing to overcome inaccuracies to the resolution of the depth sensor used

- Development of graphical interface intended to be used by non-expert in programming following the concept of plug and play for every collaborative robotic

tool. This would allow operators to design a contact-based operation without the need for programming experience.

7.2 List of peer-reviewed publications

1. Ahmed Magdy Ahmed Zaki, Marco Carnevale, Hermes Giberti, Christian Schlette, Digital model reconstruction through 3D Stereo Depth camera: a faster method exploiting robot poses, Procedia Computer Science, Volume 217, 2023, Pages 1542-1549, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2022.12.354.

2. Alberto Da Rold, Marco Furiato, Ahmed Magdy Ahmed Zaki, Marco Carnevale, Hermes Giberti, Deep learning-based robotic sorter for flexible production, Procedia Computer Science, Volume 217, 2023, Pages 1579-1588, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2022.12.358.

3. Ahmed Magdy Ahmed Zaki, Ahmed Mohsen Mohamed Fathy, Marco Carnevale, Hermes Giberti, Application of Realtime Robotics platform to execute unstructured industrial tasks involving industrial robots, cobots, and human operators, Procedia Computer Science, Volume 200, 2022, Pages 1359-1367, ISSN 1877-0509, `https://doi.org/10.1016/j.procs.2022.01.337`.

4. A. M. A. Zaki, Y. Dmytriyev, K. Castelli, M. Carnevale and H. Giberti, "Trajectory Planning for Contact-Based Robotic Applications by Use of a 3D Stereo Depth Camera," 2021 3rd International Conference on Robotics and Computer Vision (ICRCV), 2021, pp. 68-72, doi: `https://doi.org/10.1109/ICRCV52986.2021.9546958`

5. Y. Dmytriyev, A. M. A. Zaki, M. Carnevale, F. Insero and H. Giberti, "Brain computer interface for human-cobot interaction in industrial applications," 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2021, pp. 1-6, doi: `https://doi.org/10.1109/HORA52670.2021.9461383`

6. K. Castelli, A. M. A. Zaki, A. Y. Balakrishnappa, M. Carnevale and H. Giberti, "A path planning method for robotic Additive Manufacturing," 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2021, pp. 1-5, doi: `https://doi.org/10.1109/HORA52670.2021.9461380`

7. Zaki, A.M.A., Carnevale, M., Castelli, K., Giberti, H. (2021). A Simplified Approach to Size Multi-robot System for Pick and Place Applications. In: Niola, V., Gasparetto, A. (eds) Advances in Italian Mechanism Science. IFToMM ITALY 2020. Mechanisms and Machine Science, vol 91. Springer, Cham. `https://doi.org/10.1007/978-3-030-55807-9_38`

8. Carnevale, M., Castelli, K., Zaki, A.M.A., Giberti, H., Reina, C. (2021). Automation of Glue Deposition on Shoe Uppers by Means of Industrial Robots and Force Control. In: Niola, V., Gasparetto, A. (eds) Advances in Italian Mechanism Science. IFToMM ITALY 2020. Mechanisms and Machine Science, vol 91. Springer, Cham. `https://doi.org/10.1007/978-3-030-55807-9_39`

9. Castelli, Kevin, Ahmed Magdy Ahmed Zaki, Yevheniy Dmytriyev, Marco Carnevale, and Hermes Giberti. 2021. "A Feasibility Study of a Robotic Approach for the Gluing Process in the Footwear Industry" Robotics 10, no. 1: 6. https://doi.org/10.3390/robotics10010006

10. Castelli, Kevin, Ahmed Magdy Ahmed Zaki, and Hermes Giberti. 2019. "Development of a Practical Tool for Designing Multi-Robot Systems in Pick-and-Place Applications" Robotics 8, no. 3: 71. https://doi.org/10.3390/robotics8030071

7.3 List of submitted papers: The International Journal of Advanced Manufacturing Technology

1. Ahmed Magdy Ahmed Zaki, Marco Carnevale, Christian Schlette, Hermes Giberti, "Model free trajectory planning for contact based robotic applications". Abstract: Contact based operations such as surface finishing, polishing, deburring and material deposition are mandatory in the fabrication of numerous products. Many of these operations require a high level of accuracy to maintain consistent performance over time. In production systems characterized by small batches and high customization level, these operations are required to be flexible to adapt to different products within the shortest possible time and with the minimum effort for system setup. The generation of robot trajectories through digital models is a useful tool in this regard, even if in such industrial context an accurate digital model of the work piece is not always available for many reasons. This paper proposes a robotic solution to generate the robot working trajectory for contact-based operations over the external surface of unknown objects of which a digital model is not available. The solution is based on the 3D reconstruction of the object through the integration of multi-view pairs of color and depth images of the workpiece. The performance of the solution is evaluated by evaluating its accuracy in cases that are likely to be found in industrial contact-based operations, such as the application of a constant contact force and the operation over a planar or curved surface object.

Bibliography

[1] Alessandro Gasparetto et al. "Path planning and trajectory planning algorithms: A general overview". In: *Motion and operation planning of robotic systems* (2015), pp. 3–27.

[2] Bruno Siciliano et al. *Robotics. Modelling, Planning and Control*. London: Springer, 2009.

[3] Alessandro Gasparetto et al. "Trajectory planning in robotics". In: *Mathematics in Computer Science* 6.3 (2012), pp. 269–279.

[4] Atef A Ata. "Optimal trajectory planning of manipulators: a review". In: *Journal of Engineering Science and technology* 2.1 (2007), pp. 32–54.

[5] John Gregory, Alberto Olivares, and Ernesto Staffetti. "Energy-optimal trajectory planning for robot manipulators with holonomic constraints". In: *Systems and Control Letters* 61.2 (2012), pp. 279–291.

[6] Alessandro Gasparetto and Vanni Zanotto. "A technique for time-jerk optimal planning of robot trajectories". In: *Robotics and Computer-Integrated Manufacturing* 24.3 (2008), pp. 415–426. ISSN: 0736-5845. DOI: https://doi.org/10.1016/j.rcim.2007.04.001. URL: https://www.sciencedirect.com/science/article/pii/S0736584507000543.

[7] Kiyang Park and Doyoung Jeon. "Optimization of tool path pitch of spray painting robots for automotive painting quality". In: *International Journal of Control, Automation and Systems* 16.6 (2018), pp. 2832–2838.

[8] Daniel Gleeson et al. "Robot spray painting trajectory optimization". In: *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. 2020, pp. 1135–1140. DOI: 10.1109/CASE48305.2020.9216983.

[9] Jiang Liu et al. "Industrial robot path planning for polishing applications". In: *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2016, pp. 1764–1769. DOI: 10.1109/ROBIO.2016.7866584.

[10] Zengxi Pan et al. "Recent progress on programming methods for industrial robots". In: *Robotics and Computer-Integrated Manufacturing* 28.2 (2012), pp. 87–94.

[11] Heping Chen et al. "Automated robot trajectory planning for spray painting of free-form surfaces in automotive manufacturing". In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*. Vol. 1. IEEE. 2002, pp. 450–455.

[12] Steven Dubowsky, M Norris, and Zvi Shiller. "Time optimal trajectory planning for robotic manipulators with obstacle avoidance: a CAD approach". In: *Proceedings. 1986 IEEE International Conference on Robotics and Automation*. Vol. 3. IEEE. 1986, pp. 1906–1912.

[13] S-H Suh, I-K Woo, and S-K Noh. "Development of an automatic trajectory planning system (ATPS) for spray painting robots". In: *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. IEEE. 1991, pp. 1948–1955.

[14] Haichu Chen et al. "The trajectory planning system for spraying robot based on k-means clustering and NURBS curve optimization". In: *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*. 2020, pp. 5356–5361. DOI: 10.1109/IECON43393.2020.9255172.

[15] Heping Chen, Thomas Fuhlbrigge, and Xiongzi Li. "A review of CAD-based robot path planning for spray painting". In: *Industrial Robot: An International Journal* (2009).

[16] Eirik B. Njaastad and Olav Egeland. "Automatic Touch-Up of Welding Paths Using 3D Vision". In: *IFAC-PapersOnLine* 49.31 (2016). 12th IFAC Workshop on Intelligent Manufacturing Systems IMS 2016, pp. 73–78. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2016.12.164. URL: https://www.sciencedirect.com/science/article/pii/S240589631632835X.

[17] Hubert Kosler et al. "Adaptive Robotic Deburring of Die-Cast Parts with Position and Orientation Measurements Using a 3D Laser-Triangulation Sensor". In: *Strojniški vestnik - Journal of Mechanical Engineering* 62.4 (2016), pp. 207–212. ISSN: 0039-2480. DOI: 10.5545/sv-jme.2015.3227. URL: https://www.sv-jme.eu/article/adaptive-robotic-deburring-of-die-cast-parts-with-position-and-orientation-measurements-using-a-3d-laser-triangulation-sensor/.

[18] Rainer G Dorsch, Gerd Häusler, and Jürgen M Herrmann. "Laser triangulation: fundamental uncertainty in distance measurement". In: *Applied optics* 33.7 (1994), pp. 1306–1314.

[19] Harish Ravichandar et al. "Recent advances in robot learning from demonstration". In: *Annual review of control, robotics, and autonomous systems* 3 (2020), pp. 297–330.

[20] Seyed Mohammad Khansari-Zadeh and Aude Billard. "Learning Stable Nonlinear Dynamical Systems With Gaussian Mixture Models". In: *IEEE Transactions on Robotics* 27 (2011), pp. 943–957.

[21] "Skill Acquisition via Automated Multi-Coordinate Cost Balancing". In: (2019), pp. 7776–7782. DOI: 10.1109/ICRA.2019.8793762.

[22] Anahita Mohseni-Kabir et al. "Interactive Hierarchical Task Learning from a Single Demonstration". In: *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*. HRI '15. Portland, Oregon, USA: Association for Computing Machinery, 2015, 205–212. ISBN: 9781450328838. DOI: 10.1145/2696454.2696474. URL: https://doi.org/10.1145/2696454.2696474.

[23] Rui Peng et al. "A Point Cloud-Based Method for Automatic Groove Detection and Trajectory Generation of Robotic Arc Welding Tasks". In: *2020 17th International Conference on Ubiquitous Robots (UR)*. 2020, pp. 380–386. DOI: 10.1109/UR49135.2020.9144861.

[24] Stefano Pagano, Riccardo Russo, and Sergio Savino. "A vision guided robotic system for flexible gluing process in the footwear industry". In: *Robotics and Computer-Integrated Manufacturing* 65 (2020), p. 101965. ISSN: 0736-5845. DOI: https://doi.org/10.1016/j.rcim.2020.101965. URL: https://www.sciencedirect.com/science/article/pii/S0736584519301097.

[25] Arturo Realyvásquez-Vargas et al. "Introduction and configuration of a collaborative robot in an assembly task as a means to decrease occupational risks and increase efficiency in a manufacturing company". In: *Robotics and Computer-Integrated Manufacturing* 57 (2019), pp. 315–328. ISSN: 0736-5845. DOI: https://doi.org/10.1016/j.rcim.2018.12.015. URL: https://www.sciencedirect.com/science/article/pii/S0736584518302990.

[26] Abdelfetah Hentout et al. "Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017". In: *Advanced Robotics* 33.15-16 (2019), pp. 764–799. DOI: 10.1080/01691864.2019.1636714. eprint: https://doi.org/10.1080/01691864.2019.1636714. URL: https://doi.org/10.1080/01691864.2019.1636714.

[27] International Federation of Robotics. *World Robotics 2020 Report*. Tech. rep. International Federation of Robotics, September 24, 2020.

[28] Federico Vicentini. "Collaborative robotics: a survey". In: *Journal of Mechanical Design* 143.4 (2021).

[29] José Saenz et al. "Survey of methods for design of collaborative robotics applications-why safety is a barrier to more widespread robotics uptake". In: *Proceedings of the 2018 4th International Conference on Mechatronics and Robotics Engineering*. 2018, pp. 95–101.

[30] Valeria Villani et al. "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications". In: *Mechatronics* 55 (2018), pp. 248–266. ISSN: 0957-4158. DOI: https://doi.org/10.1016/j.mechatronics.2018.02.009. URL: https://www.sciencedirect.com/science/article/pii/S0957415818300321.

[31] G. Hirzinger et al. "On a new generation of torque controlled light-weight robots". In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*. Vol. 4. 2001, 3356–3363 vol.4. DOI: 10.1109/ROBOT.2001.933136.

[32] "Robots and Robotic Devices—Safety Requirements for Industrial Robots—Part 2: Robot Systems and Integration." In: ISO 10218-2 (2011).

[33] "Robots and Robotic Devices—Collaborative Robots." In: ISO/TS 15066 (2016).

[34] Sten Grahn et al. "Potential advantages using large anthropomorphic robots in human-robot collaborative, hand guided assembly". In: *Procedia CIRP* 44 (2016), pp. 281–286.

[35] Federica Ferraguti et al. "Walk-through programming for industrial applications". In: *Procedia Manufacturing* 11 (2017), pp. 31–38.

[36] Jeremy A. Marvel. "Performance Metrics of Speed and Separation Monitoring in Shared Workspaces". In: *IEEE Transactions on Automation Science and Engineering* 10.2 (2013), pp. 405–414. DOI: 10.1109/TASE.2013.2237904.

[37] Jeremy A Marvel and Rick Norcross. "Implementing speed and separation monitoring in collaborative robot workcells". In: *Robotics and computer-integrated manufacturing* 44 (2017), pp. 144–155.

[38] Michael Melia et al. "Pressure pain thresholds: subject factors and the meaning of peak pressures". In: *European Journal of Pain* 23.1 (2019), pp. 167–182.

[39] Bhanoday Vemula, Björn Matthias, and Aftab Ahmad. "A design metric for safety assessment of industrial robot design suitable for power-and force-limited collaborative operation". In: *International journal of intelligent robotics and applications* 2.2 (2018), pp. 226–234.

[40] G.A. Pratt and M.M. Williamson. "Series elastic actuators". In: *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*. Vol. 1. 1995, 399–406 vol.1. DOI: 10.1109/IROS.1995.525827.

[41] Brian T. Knox and James P. Schmiedeler. "A Unidirectional Series-Elastic Actuator Design Using a Spiral Torsion Spring". In: *Journal of Mechanical Design* 131.12 (Nov. 2009). 125001. ISSN: 1050-0472. DOI: 10.1115/1.4000252. eprint: https://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/131/12/125001/5601531/125001_1.pdf. URL: https://doi.org/10.1115/1.4000252.

[42] Alex S. Shafer and Mehrdad R. Kermani. "On the Feasibility and Suitability of MR Fluid Clutches in Human-Friendly Manipulators". In: *IEEE/ASME Transactions on Mechatronics* 16.6 (2011), pp. 1073–1082. DOI: 10.1109/TMECH.2010.2074210.

[43] Chee-Meng Chew, Geok-Soon Hong, and Wei Zhou. "Series damper actuator: a novel force/torque control actuator". In: *4th IEEE/RAS International Conference on Humanoid Robots, 2004.* Vol. 2. 2004, 533–546 Vol. 2. DOI: 10.1109/ICHR.2004.1442669.

[44] A. Bicchi and G. Tonietti. "Fast and "soft-arm" tactics [robot arm design]". In: *IEEE Robotics and Automation Magazine* 11.2 (2004), pp. 22–33. DOI: 10.1109/MRA.2004.1310939.

[45] Shirine El Zaatari et al. “Cobot programming for collaborative industrial tasks: An overview”. In: *Robotics and Autonomous Systems* 116 (2019), pp. 162–180. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2019.03.003. URL: https://www.sciencedirect.com/science/article/pii/S092188901830602X.

[46] Justin Huang and Maya Cakmak. “Code3: A System for End-to-End Programming of Mobile Manipulator Robots for Novices and Experts”. In: *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. HRI ’17. Vienna, Austria: Association for Computing Machinery, 2017, 453–462. ISBN: 9781450343367. DOI: 10.1145/2909824.3020215. URL: https://doi.org/10.1145/2909824.3020215.

[47] Casper Schou et al. “Skill-based instruction of collaborative robots in industrial settings”. In: *Robotics and Computer-Integrated Manufacturing* 53 (2018), pp. 72–80. ISSN: 0736-5845. DOI: https://doi.org/10.1016/j.rcim.2018.03.008. URL: https://www.sciencedirect.com/science/article/pii/S0736584516301910.

[48] David Weintrop et al. “Blockly goes to work: Block-based programming for industrial robots”. In: *2017 IEEE Blocks and Beyond Workshop (B&B)*. 2017, pp. 29–36. DOI: 10.1109/BLOCKS.2017.8120406.

[49] Daniela Fogli et al. “A hybrid approach to user-oriented programming of collaborative robots”. In: *Robotics and Computer-Integrated Manufacturing* 73 (2022), p. 102234. ISSN: 0736-5845. DOI: https://doi.org/10.1016/j.rcim.2021.102234. URL: https://www.sciencedirect.com/science/article/pii/S073658452100106X.

[50] Sara Beschi, Daniela Fogli, and Fabio Tampalini. “CAPIRCI: A Multi-modal System for Collaborative Robot Programming”. In: *End-User Development*. Ed. by Alessio Malizia et al. Cham: Springer International Publishing, 2019, pp. 51–66. ISBN: 978-3-030-24781-2.

[51] Scott Niekum et al. “Learning and generalization of complex tasks from unstructured demonstrations”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 5239–5246.

[52] Gopika Ajaykumar, Maia Stiber, and Chien-Ming Huang. “Designing user-centric programming aids for kinesthetic teaching of collaborative robots”. In: *Robotics and Autonomous Systems* 145 (2021), p. 103845. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2021.103845. URL: https://www.sciencedirect.com/science/article/pii/S0921889021001305.

[53] *Software Manual TMflow*. Software Version: 1.68. Original instructions. Omron.

[54] *Original-Betriebsanleitung*. Version 1.2. Für Installations-, Bedienungs- und Instandhaltungspersonal immer beim Produkt aufbewahren. Horst. 2.03.2020.

[55] *Onrobot sander Datasheet*. v1.0. Onrobot.

[56] Casper Schou and Ole Madsen. "A plug and produce framework for industrial collaborative robots". In: *International journal of advanced robotic systems* 14.4 (2017), p. 1729881417717472.

[57] Michael Wojtynek, Jochen Jakob Steil, and Sebastian Wrede. "Plug, plan and produce as enabler for easy workcell setup and collaborative robot programming in smart factories". In: *KI-Künstliche Intelligenz* 33.2 (2019), pp. 151–161.

[58] Yu He and Shengyong Chen. "Advances in sensing and processing methods for three-dimensional robot vision". In: *International Journal of Advanced Robotic Systems* 15.2 (2018), p. 1729881418760623. DOI: 10.1177/1729881418760623. eprint: https://doi.org/10.1177/1729881418760623. URL: https://doi.org/10.1177/1729881418760623.

[59] Colin Rennie et al. "A dataset for improved rgbd-based object detection and pose estimation for warehouse pick-and-place". In: *IEEE Robotics and Automation Letters* 1.2 (2016), pp. 1179–1185.

[60] Andy Zeng et al. "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching". In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 3750–3757.

[61] Colin Rennie et al. "A dataset for improved rgbd-based object detection and pose estimation for warehouse pick-and-place". In: *IEEE Robotics and Automation Letters* 1.2 (2016), pp. 1179–1185.

[62] Ahmed Magdy Ahmed Zaki et al. "Application of Realtime Robotics platform to execute unstructured industrial tasks involving industrial robots, cobots, and human operators". In: *Procedia Computer Science* 200 (2022). 3rd International Conference on Industry 4.0 and Smart Manufacturing, pp. 1359–1367. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2022.01.337. URL: https://www.sciencedirect.com/science/article/pii/S1877050922003465.

[63] Larry Li et al. "Time-of-flight camera–an introduction". In: *Technical white paper* SLOA190B (2014).

[64] Oliver Wasenmüller and Didier Stricker. "Comparison of kinect v1 and v2 depth images in terms of accuracy and precision". In: *Asian Conference on Computer Vision*. Springer. 2016, pp. 34–45.

[65] Monica Carfagni et al. "Metrological and critical characterization of the Intel D415 stereo depth camera". In: *Sensors* 19.3 (2019), p. 489.

[66] François Pomerleau, Francis Colas, Roland Siegwart, et al. "A review of point cloud registration algorithms for mobile robotics". In: *Foundations and Trends® in Robotics* 4.1 (2015), pp. 1–104.

[67] Anh Nguyen and Bac Le. "3D point cloud segmentation: A survey". In: *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. 2013, pp. 225–230. DOI: 10.1109/RAM.2013.6758588.

[68] Qian Wang and Min-Koo Kim. "Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018". In: *Advanced Engineering Informatics* 39 (2019), pp. 306–319.

[69] Aitor Aldoma et al. "Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation". In: *IEEE Robotics and Automation Magazine* 19.3 (2012), pp. 80–91.

[70] Hongzhuo Liang et al. "Pointnetgpd: Detecting grasp configurations from point sets". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 3629–3635.

[71] Jason Zhang et al. "NeRS: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 29835–29847.

[72] Selim Engin et al. "Higher order function networks for view planning and multi-view reconstruction". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 11486–11492.

[73] Yongbin Sun et al. "Im2Avatar: Colorful 3D Reconstruction from a Single Image". In: *CoRR* abs/1804.06375 (2018). arXiv: 1804.06375. URL: http://arxiv.org/abs/1804.06375.

[74] Hantong Xu, Jiamin Xu, and Weiwei Xu. "Survey of 3D modeling using depth cameras". In: *Virtual Reality and Intelligent Hardware* 1.5 (2019), pp. 483–499.

[75] Tanwi Mallick, Partha Pratim Das, and Arun Kumar Majumdar. "Characterizations of Noise in Kinect Depth Images: A Review". In: *IEEE Sensors Journal* 14.6 (2014), pp. 1731–1740. DOI: 10.1109/JSEN.2014.2309987.

[76] C. Tomasi and R. Manduchi. "Bilateral filtering for gray and color images". In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. 1998, pp. 839–846. DOI: 10.1109/ICCV.1998.710815.

[77] P.J. Besl and Neil D. McKay. "A method for registration of 3-D shapes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), pp. 239–256. DOI: 10.1109/34.121791.

[78] Y. Chen and G. Medioni. "Object modeling by registration of multiple range images". In: *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. 1991, 2724–2729 vol.3. DOI: 10.1109/ROBOT.1991.132043.

[79] Radu Bogdan Rusu. "Semantic 3D object maps for everyday manipulation in human living environments". In: *KI-Künstliche Intelligenz* 24.4 (2010), pp. 345–348.

[80] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. "Fast point feature histograms (FPFH) for 3D registration". In: *2009 IEEE international conference on robotics and automation*. IEEE. 2009, pp. 3212–3217.

[81] Brian Curless and Marc Levoy. "A Volumetric Method for Building Complex Models from Range Images". In: *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '96. New York, NY, USA: Association for Computing Machinery, 1996, 303–312. ISBN: 0897917464. DOI: 10.1145/237170.237269. URL: https://doi.org/10.1145/237170.237269.

[82] Jörg Stückler and Sven Behnke. "Multi-resolution surfel maps for efficient dense 3D modeling and tracking". In: *Journal of Visual Communication and Image Representation* 25.1 (2014). Visual Understanding and Applications with RGB-D Cameras, pp. 137–147. ISSN: 1047-3203. DOI: https://doi.org/10.1016/j.jvcir.2013.02.008. URL: https://www.sciencedirect.com/science/article/pii/S104732031300028X.

[83] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. "Robust reconstruction of indoor scenes". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 5556–5565.

[84] Jiwan Kim et al. "Multiple weld seam extraction from RGB-depth images for automatic robotic welding via point cloud registration". In: *Multimedia Tools and Applications* 80.6 (2021), pp. 9703–9719.

[85] Madhusudhan Rao R, Deepa Radhakrishna, and Usha S. "Development of a Robot-mounted 3D Scanner and Multi-view Registration Techniques for Industrial Applications". In: *Procedia Computer Science* 133 (2018). International Conference on Robotics and Smart Manufacturing (RoSMa2018), pp. 256–267. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.07.032. URL: https://www.sciencedirect.com/science/article/pii/S1877050918309773.

[86] Daniele De Gregorio, Federico Tombari, and Luigi Di Stefano. "RobotFusion: Grasping with a robotic manipulator via multi-view reconstruction". In: *European Conference on Computer Vision*. Springer. 2016, pp. 634–647.

[87] *Manual TM5 Series - Regular Payload Series Hardware Installation*. I623-E-02. Hardware 3.1 or earlier. Techman robots. 2020.

[88] *Universal Robots e-Series User Manual UR5e*. Version 5.7. Original instructions (en) US Version. Universal Robots.

[89] *Intel® RealSense Product Family D400 Series*. 337029-012. Revision 012. Intel® RealSense. 2022.

[90] *Artec Eva brochure*. EHD-B-001-10/2020-NOPE-EN. Artec.

[91] *Onrobot HEX-E/H QC force/torque sensor Datasheet*. v1.1. Onrobot.

[92] *Universal Robots e-Series brochure*. en/0618. Universal Robots.

[93] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. "Open3D: A Modern Library for 3D Data Processing". In: *arXiv:1801.09847* (2018).

[94] Paolo Cignoni, Massimiliano Corsini, and Guido Ranzuglia. "MeshLab: an Open-Source 3D Mesh Processing System." In: *ERCIM News* 2008.73 (2008). URL: http://dblp.uni-trier.de/db/journals/ercim/ercim2008.html#CignoniCR08.

[95] Ahmed Magdy Ahmed Zaki et al. "Trajectory Planning for Contact-Based Robotic Applications by Use of a 3D Stereo Depth Camera". In: *2021 3rd International Conference on Robotics and Computer Vision (ICRCV)*. 2021, pp. 68–72. DOI: 10.1109/ICRCV52986.2021.9546958.

[96] N. Ganganath and H. Leung. "Mobile robot localization using odometry and kinect sensor". In: *2012 IEEE International Conference on Emerging Signal Processing Applications*. 2012, pp. 91–94. DOI: 10.1109/ESPA.2012.6152453.

[97] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. "Colored Point Cloud Registration Revisited". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 143–152. DOI: 10.1109/ICCV.2017.25.

[98] Frank Steinbrücker, Jürgen Sturm, and Daniel Cremers. "Real-time visual odometry from dense RGB-D images". In: *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. 2011, pp. 719–722. DOI: 10.1109/ICCVW.2011.6130321.

[99] Christian Kerl, Jürgen Sturm, and Daniel Cremers. "Robust odometry estimation for RGB-D cameras". In: *2013 IEEE International Conference on Robotics and Automation*. 2013, pp. 3748–3754. DOI: 10.1109/ICRA.2013.6631104.

[100] Jianfeng Yin and J.R. Cooperstock. "A new photo consistency test for voxel coloring". In: *The 2nd Canadian Conference on Computer and Robot Vision (CRV'05)*. 2005, pp. 566–570. DOI: 10.1109/CRV.2005.9.

[101] Jiaolong Yang et al. "Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.11 (2016), pp. 2241–2254. DOI: 10.1109/TPAMI.2015.2513405.

[102] Brian Curless and Marc Levoy. "A Volumetric Method for Building Complex Models from Range Images". In: *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '96. New York, NY, USA: Association for Computing Machinery, 1996, 303–312. ISBN: 0897917464. DOI: 10.1145/237170.237269. URL: https://doi.org/10.1145/237170.237269.

[103] Liang Cheng et al. "Registration of Laser Scanning Point Clouds: A Review". In: *Sensors* 18.5 (2018). ISSN: 1424-8220. DOI: 10.3390/s18051641. URL: https://www.mdpi.com/1424-8220/18/5/1641.

[104] Hao Men, Biruk Gebre, and Kishore Pochiraju. "Color point cloud registration with 4D ICP algorithm". In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 1511–1516. DOI: 10.1109/ICRA.2011.5980407.

[105] N. Aspert, D. Santa-Cruz, and T. Ebrahimi. "MESH: measuring errors between surfaces using the Hausdorff distance". In: *Proceedings. IEEE International Conference on Multimedia and Expo*. Vol. 1. 2002, 705–708 vol.1. DOI: 10.1109/ICME.2002.1035879.

[106] M.-P. Dubuisson and A.K. Jain. "A modified Hausdorff distance for object matching". In: *Proceedings of 12th International Conference on Pattern Recognition*. Vol. 1. 1994, 566–568 vol.1. DOI: 10.1109/ICPR.1994.576361.

[107] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. "Comparing images using the Hausdorff distance". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.9 (1993), pp. 850–863. DOI: 10.1109/34.232073.

[108] Claudia Redaelli, Marzio Sorlini, and Claudio R Boër. "A laboratory for industrial research on mass customisation in the footwear industry". In: *International Journal of Mass Customisation* 1.4 (2006), pp. 492–506.

[109] Theodoros Staikos and Shahin Rahimifard. "An End-of-Life Decision Support Tool for Product Recovery Considerations in the Footwear Industry". In: *Int. J. Computer Integrated Manufacturing* 20 (Sept. 2007), pp. 602–615. DOI: 10.1080/09511920701416549.

[110] Giovanni Danese et al. "A Novel Standard for Footwear Industrial Machineries". In: *IEEE Trans. Industrial Informatics* 7 (Nov. 2011), pp. 713–722. DOI: 10.1109/TII.2011.2166789.

[111] Muhammed Selman Eryilmaz. "Analysis of shoe manufacturing factory by simulation of production processes". In: *Southeast Europe Journal of Soft Computing* 1.1 (2012).

[112] Claudio Roberto Boër and Sergio Dulio. *Mass customization and footwear: myth, salvation or reality?: a comprehensive analysis of the adoption of the mass customization paradigm in footwear, from the perspective of the EUROShoE (Extended User Oriented Shoe Enterprise) Research Project*. Springer Science & Business Media, 2007.

[113] Robert Freund. "Change management for Mass Customization–Changing minds through representational redescription". In: *13th Scientific Conference on INDUSTRIAL SYSTEMS, IS2005, Herceg Novi, Montenegro*. 2005.

[114] Kevin Castelli et al. "A Feasibility Study of a Robotic Approach for the Gluing Process in the Footwear Industry". In: *Robotics* 10.1 (2021). ISSN: 2218-6581. DOI: 10.3390/robotics10010006. URL: https://www.mdpi.com/2218-6581/10/1/6.

[115] Marco Carnevale et al. "Automation of Glue Deposition on Shoe Uppers by Means of Industrial Robots and Force Control". In: *The International Conference of IFToMM ITALY*. Springer. 2020, pp. 344–352.

[116] Herbert Edelsbrunner. "Alpha Shapes—a Survey". In: *Tessellations in the Sciences* (Jan. 2010).

[117] Xu-Zheng Liu et al. "An offset algorithm for polyline curves". In: *Computers in Industry* 58 (Apr. 2007), pp. 240–254. DOI: 10.1016/j.compind.2006.06.002.

[118] Iñaki Maurtua, Aitor Ibarguren, and Alberto Tellaeche. "Robotic solutions for footwear industry". In: *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*. IEEE. 2012, pp. 1–4. DOI: https://doi.org/10.1109/ETFA.2012.6489780.

[119] Iñaki Maurtua, Aitor Ibarguren, and Alberto Tellaeche. "Robotics for the benefit of footwear industry". In: *International Conference on Intelligent Robotics and Applications*. Springer. 2012, pp. 235–244. DOI: https://doi.org/10.1007/978-3-642-33515-0_24.

[120] Petr Janousek and Jan Faigl. "Speeding up coverage queries in 3D multi-goal path planning". In: *2013 IEEE International Conference on Robotics and Automation*. IEEE. 2013, pp. 5082–5087.

[121] Ahmed Magdy Ahmed Zaki et al. "Digital model reconstruction through 3D Stereo Depth camera: a faster method exploiting robot poses". In: *Procedia Computer Science* 217 (2023). 4th International Conference on Industry 4.0 and Smart Manufacturing, pp. 1542–1549. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2022.12.354. URL: https://www.sciencedirect.com/science/article/pii/S1877050922024395.

[122] Richard A. Newcombe et al. "KinectFusion: Real-time dense surface mapping and tracking". In: *2011 10th IEEE International Symposium on Mixed and Augmented Reality* (2011), pp. 127–136.

[123] Shaodan Hou et al. "A review of 3D printed concrete: Performance requirements, testing measurements and mix design". In: *Construction and Building Materials* 273 (2021), p. 121745. ISSN: 0950-0618. DOI: https://doi.org/10.1016/j.conbuildmat.2020.121745. URL: https://www.sciencedirect.com/science/article/pii/S0950061820337491.

[124] Hamad Al Jassmi, Fady Al Najjar, and Abdel-Hamid Ismail Mourad. "Large-Scale 3D Printing: The Way Forward". In: *IOP Conference Series: Materials Science and Engineering* 324 (2018), p. 012088. DOI: 10.1088/1757-899x/324/1/012088. URL: https://doi.org/10.1088/1757-899x/324/1/012088.

[125] Jingchuan Zhang et al. "A review of the current progress and application of 3D printed concrete". In: *Composites Part A: Applied Science and Manufacturing* 125 (2019), p. 105533. ISSN: 1359-835X. DOI: https://doi.org/10.1016/j.compositesa.2019.105533. URL: https://www.sciencedirect.com/science/article/pii/S1359835X19302829.