Chair of Computational Modeling and Simulation
TUM School of Engineering and Design
Technical University of Munich

Integration of laser profiler feedback into FIM-based additive manufacturing in construction

Scientific work to obtain the degree

Master of Science (M.Sc.) in Computational Mechanics

at the TUM School of Engineering and Design
of the Technical University of Munich.

Supervised by Prof. Dr-Ing. André Borrmann
M.Sc. Martin Slepicka
Chair of Computational Modeling and Simulation

Submitted by Simon Espinosa Leon [redacted]
[redacted]
[redacted]
e-Mail: simon.espinosa@tum.de

Submitted on September 2023

Preface

A very special thank you to Martin Slepicka, my dedicated supervisor, whose unwavering support and invaluable guidance played a pivotal role in shaping this thesis. Martin's profound knowledge of additive manufacturing within the construction industry and his genuine interest in the subject matter were a constant source of inspiration throughout this journey. I sincerely thank his boundless patience and persistent support, which contributed immensely to making this work the best it could be. His continuous suggestions proved invaluable, particularly during moments when I found myself stuck with the complexities of this study.

I am immensely grateful to my dear friends, Jalal and Ahmed, whose presence and support provided me with the strength and resilience needed to overcome the challenges that arose. Following the multiple crashes of my virtual machine that resulted in weeks of lost work, Ahmed's assistance with Docker was nothing short of a lifesaver. And their willingness to lend an ear to my confusing questions. Thank you.

To Sofija, who stood by my side throughout this endeavor, offering a sympathetic ear during my moments of struggle. She understood my concept of time, especially when I claimed I would leave the lab in just five minutes, only to lose track of time and arrive hours later.

Last but certainly not least, I want to express my most profound appreciation to my family for their constant and unconditional support. From day one, they have shown a genuine interest in my academic pursuits and research, and their encouragement has been a continuous source of motivation. Their belief in me has been the cornerstone upon which this thesis has been built.

Abstract

This research goes into the innovative application of laser scanning technology within the domain of Three dimensional (3D) printing in the construction industry. The central focus lies in the seamless integration of various cutting-edge technologies, including Building Information Modeling (BIM), Additive Manufacturing (AM), and Digital Twining.To actualize this exploration, a prototype is generated through the conversion of a digital model into a physical object. Throughout the process of printing and drying, detailed measurements are captured at multiple stages and stored as Point Cloud (PC) data. The setup involves a pair of Universal Robots (UR) working independently, an extruder, and a Gocator laser profiler.

In essence, the study seeks to find insights concerning the material required to post-process the exterior wall textures. Additionally, it aims to revolutionize operational planning and real-time "as-built" data generation through the detailed mapping of shape and height deviations. It is crucial to note that the current study is confined to a linear scanning trajectory along the x-plane, executed at a consistent speed. Furthermore, the scope primarily encompasses planar prints oriented along the x-axis. Finally, this research displays a list of recommendations for the further development of this study to overcome limitations and achieve the potential the technology has to offer.

Zusammenfassung

Diese Forschung widmet sich der innovativen Anwendung der Laserscan-Technologie im Bereich des 3D-Drucks in der Baubranche. Der zentrale Fokus liegt auf der nahtlosen Integration verschiedener hochmoderner Technologien, einschließlich Building Information Modeling (BIM), Additive Manufacturing (AM) und Digital Twin (DT). Um diese Erkundung zu realisieren, wird ein Prototyp durch die Umwandlung eines digitalen Modells in ein physisches Objekt generiert. Während des komplexen Prozesses des Druckens und Trocknens werden detaillierte Messungen in mehreren Stadien in Point Cloud (PC) Daten erfasst. Das Setup umfasst ein Paar von Universal Robots (UR), die unabhängig voneinander arbeiten, einen Extruder und einen Gocator-Laserscanner.

Im Wesentlichen zielt die Studie darauf ab, wertvolle Erkenntnisse über das benötigte Material zur Nachbearbeitung der Strukturen von Außenwänden zu gewinnen. Darüber hinaus soll sie die Betriebsplanung revolutionieren und die Echtzeitgenerierung von "as-Built"-Daten durch die detaillierte Kartierung von Form- und Höhenabweichungen vorantreiben. Es ist wichtig zu beachten, dass die aktuelle Studie auf eine lineare Scanning-Trajektorie entlang der x-Ebene beschränkt ist, die mit einer konstanten Geschwindigkeit ausgeführt wird. Darüber hinaus umfasst der Umfang hauptsächlich planare Drucke, die entlang der x-Achse ausgerichtet sind. Schließlich präsentiert diese Forschung eine Liste von Empfehlungen zur weiteren Entwicklung dieser Studie, um Einschränkungen zu überwinden und das Potenzial der Technologie auszuschöpfen.

Contents

Acronyms

2D	Two dimensional
3D	Three dimensional
AM	Additive Manufacturing
BIM	Building Information Modeling
CFD	Computational Fluid Dynamics
dpi	dots per inch
DT	Digital Twin
FEM	Finite Element Methods
FIM	Fabrication Information Modeling
FOV	Field of View
GUI	Graphical User Interface
IFC	Industry Foundation Classes
IoT	Internet of Things
NURBS	Non-uniform rational B-spline
OOP	Object Oriented Programming
OpenCV	Open Source Computer Vision
PC	Point Cloud
PCL	Point cloud library
PCL	Point cloud library
ROI	Region of Interest
RTDE	Real Time Data Exchnage
UR	Universal Robots

Chapter 1

Introduction

Building Information Modeling (BIM), Fabrication Information Modeling (FIM), Additive Manufacturing (AM), and Digital Twining are key technological innovations that have significantly impacted the construction industry, shaping the way buildings and infrastructure are designed, constructed, and maintained in a more productive way. Each of these concepts plays a crucial role in the realization of the fourth industrial revolution, "Industrie 4.0", in the construction sector.

Each of these technologies has its individual contributions and interconnected roles in transforming the construction landscape. BIM is a digital representation of the physical and functional characteristics of a building, and it serves as an intelligent, 3D model with the most valuable information about the project, this information is usually stored in Industry Foundation Classes (IFC).

AM represents a modern approach to manufacturing, replacing conventional methods with advanced technological techniques. It empowers the fabrication of intricate objects previously unattainable through traditional means, while also significantly enhancing material efficiency, thereby reducing waste. FIM is an intermediate layer between BIM and AM to handle the fabrication information as illustrated in Figure 1.1. Finally, DT keeps both the physical object and the digital model mutually synchronized in real-time.

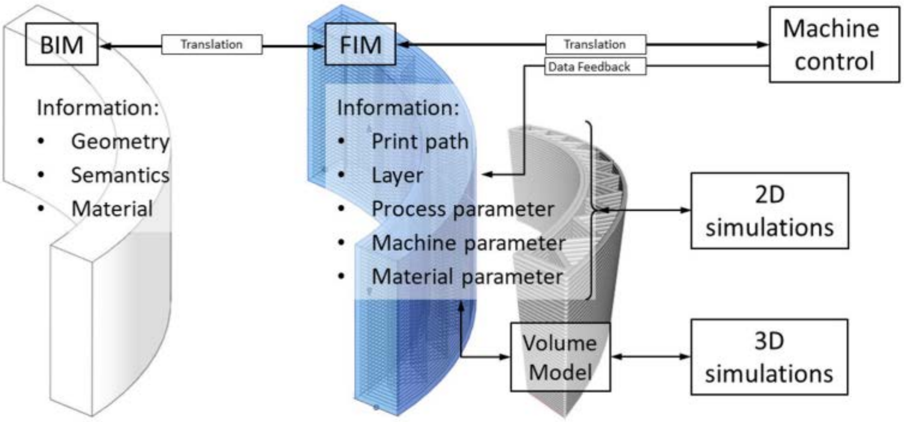


Figure 1.1: FIM in the fabrication process (SLEPICKA, VILGERTSHOFER, et al., 2022)

As pressure increases to implement technology in the construction industry, DT and robotics emerge to update BIM models and automate processes, respectively. Robots are being used in almost all industries to enhance productivity and improve efficiency. In the construction industry, one of the uses of robotics is for 3D concrete printing, among other applications. This study focuses on using AM in the construction industry, in particular, on enabling digital twining methods for AM. Digital Twins, on the other hand, enable real-time monitoring that improves decision-making (TUHAISE et al., 2023).

AM is one of the key pillars of "Industrie 4.0". Industrie 4.0 is the new type of industrialization, as explained by KAGERMANN et al., 2013 at the Hannover Fair in 2011, where different technologies are brought to manufacturing to increase the level of digitalization. Here, manufacturing will be smart, dynamic, and customized; changes can be made at the last minute in production and have full transparency in the manufacturing cycle as explained in chapter 2.

The construction industry is joining this revolution from the planning phase through its building phase, considering the end user. Quality control is key during construction 4.0 and is taken care of by sensors and thorough planning, as will be discussed during this study. (OZTEMEL and GURSEV, 2020)

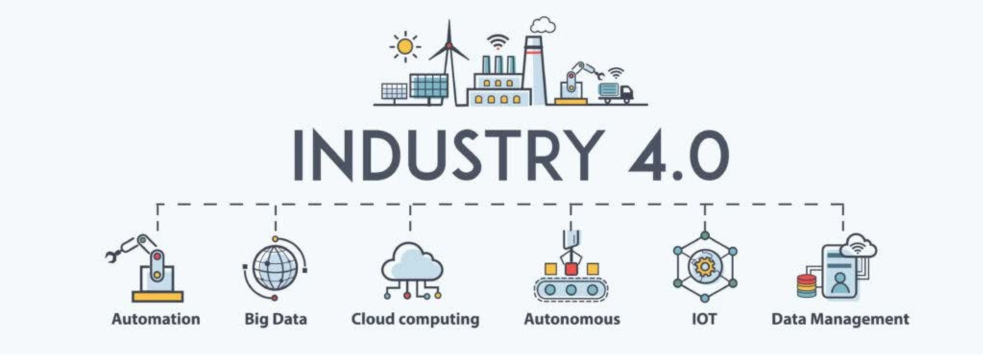


Figure 1.2: Technologies involved in Industry 4.0 (“What is industry 4.0?”, 2021)

Moreover, significant advancements in robotics have occurred across various industries, leading to the emergence of collaborative robots that can work together and interact. Collaborative robots, also known as cobots, are designed to operate alongside humans and perform tasks either in tandem or independently. Figure 1.3 exemplifies two mobile robots engaged in the same task but in different locations, as studied by DÖRFLER et al., 2022. Additionally, collaborative robots can effectively handle diverse tasks simultaneously within the same workspace, as is the case in this particular study. The integration of collaborative robots with other Industry 4.0 technologies enhances automation, efficiency, and safety in construction and various other sectors.

Figure 1.3: Collaborative robots supervised by people

Purpose statement

This research explores the use of laser scanning technology to retrieve information from the 3D printed structure and store that information in the FIM. The Thesis aims to provide the FIM with information on the uppermost layer during different scan stages and also on the outer walls. The information that is analyzed is the layer height deviation from the planned layer height, the printed shape deviation from the designed shape, and the material required to post-process the outer walls to make it flat.

Significance

Construction is a "highly data-intensive industry", and BIM, a digital design methodology incorporating tools and design strategies for different domain specialists, was expected to fulfill the data requirements (MUTESI and KYAKULA, 2011). However, for the use of AM in construction, BIM methodology needed to be extended by an intermediate layer, this layer would need to translate the 3D information into discrete Two dimensional (2D) information to be able to do the layer by layer manufacturing. FIM is this intermediate layer, it handles the fabrication information in discrete layers, from geometrical information to material and machine parameters.

A vital component that is still missing in the FIM framework is an automated data feedback mechanism for the as-built component update. Currently, FIM only considers what is needed to be built according to the BIM specifications, but it doesn't contemplate an action plan on how to react if the manufactured object is not within the specifications.

The proposed as-built update mechanism serves as a communication bridge between digital and physical object, facilitating efficient information exchange throughout the manufacturing stage of the construction project. It enables real-time feedback and quality assurance during the printing phase, enabling improvements on-the-go. Moreover, users gain insights into deviations between the physical and digital models, that could aid even later on, in operational planning, structural maintenance, and future renovation.

Incorporating information from the outer surface of printed objects provides valuable insights into material requirements, leading to better planning, reduced waste, and increased efficiency. Understanding the role of structural prints during the operational phase enhances decision-making for long-term maintenance and future adaptability.

Digital twinning aims to synchronize digital models with real-world components, offering numerous applications and storage potential for future use cases. It is crucial to collect and store as much information as possible to exploit the full potential of DT. Emphasizing the importance of research in this field enables timely utilization of available information and exploration of new possibilities.

Scope and limitations

This research seeks to find different usages for the scanned data provided by the Gocator laser profile scanner mounted on a UR. Once the relevant information is extracted from the raw data, it is stored effectively in the FIM. During this research, the best way to perform the scanning process is not studied, nor the communication between UR robots is considered. For the study, one robot is moved using a Real Time Data Exchnage (RTDE) module that has already been implemented, while the other was moved with the URcap that allows it to move and trigger the scanning process as seen in Figure 1.4. An automated scan planning is not part of the scope of this work either.

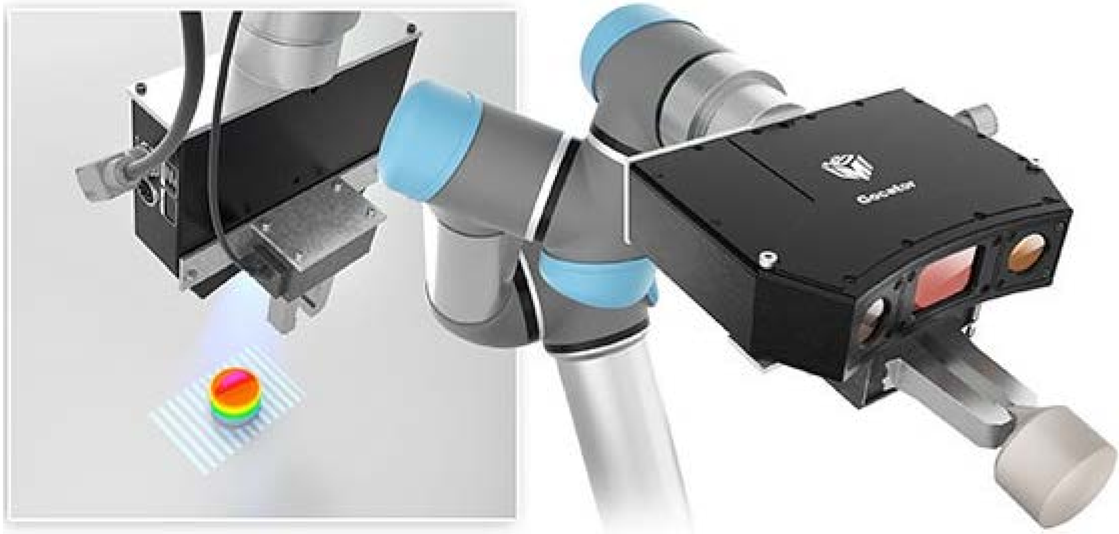


Figure 1.4: UR with a Gocator scanner

This study is limited to scanning in a straight line on the x-plane at a constant speed due to the Gocator URcap, which is used to move and trigger the scans. For the scope of the thesis, a top-down approach was chosen. First, the use cases were defined, and accordingly, the 3D scans were performed.

Research question

This research aims to implement several tools to integrate features into the FIM to reach the Construction 4.0 goals. The tools that have been implemented are detailed in chapter 5, and seek to answer the following question;

> ***How can scan data of the printing process be stored in a FIM to understand the shape and height deviation and measure the required material to post-process the component?***

The proposed solution entailed conducting a series of scans from the top of the printed structure, which were then compared with the FIM model. This comparison served to measure any deviations between the physical structure and the digital model. The deviations were then transformed into a heat map image, providing a visual representation of areas where the as-built object deviated from the as-designed model.

In addition to the top scans, a lateral scan is also performed to assess the intra-layer gap and identify the amount of material required to achieve a flat outer wall surface. The source of these gaps is better explained later, but in essence, are due to the resolution of the print and the round exterior of each filament.

Overview of thesis structure

This study is divided into seven main chapters. Introduction, theoretical background, methodology, hardware and software, implementation, results, and some conclusions.

The theoretical background goes through some state-of-the-art, the most important definitions, and a theoretical framework that is essential to understand best what has been done in this specific research. Here it also explains in more detail the limitations of the study. Then the hardware used, the different Python libraries, and other used software are also described here.

The implementation describes the different classes and the code structure. This is essential for future implementations besides the tools described here. The implementation is tested on a benchmark experiment to analyze the results, and finally, conclusions and recommendations are presented.

This study uses an IFC file as explained by Slepicka, Vilgertshofer, et al., 2022 to produce a physical model using a RTDE as described by Slepicka et al., 2023. These steps are those seen in Figure 1.5 where information goes from the IFC to the path plan, the path plan is translated into robot joint positions sent to the UR, and it is extruded to end up with an actual physical model. The physical was then scanned and analyzed.

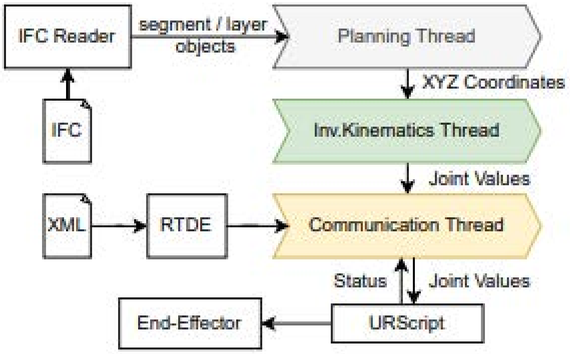


Figure 1.5: IFC to AM flowchart (SLEPICKA et al., 2023)

Chapter 2

Theoretical Background

This chapter serves as an introduction to the theoretical background of the study and provides an overview of the state of the art in the field. It also defines key terms and concepts necessary to understand the research better.

> *"Industry 4.0 refers to the integration of a multiplicity of technologies and agents for the common goal of improving the efficiency and responsiveness of a production system. This integration has the potential to revolutionize how business is planned and conducted."* (AHUETT-GARZA & KURFESS, 2018)

As mentioned in **??**, Industry 4.0 represents the fourth industrial revolution, integrating various cutting-edge technologies such as Big Data, Internet of Things (IoT), AM, Machine Learning, Robotics, and DT, among others. The primary objective of "Industrie 4.0" is to enhance digitalization and automation in industries. Figure 2.1 illustrates how different technologies leverage sensors to enable a higher level of automation.

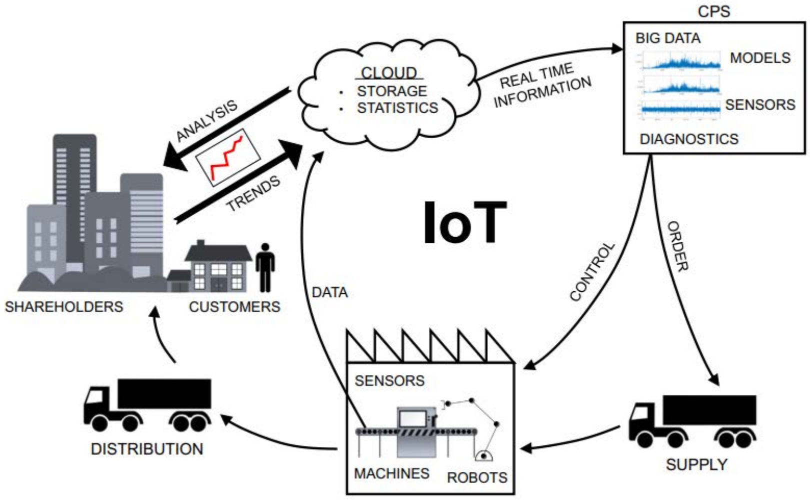


Figure 2.1: Habilitating technologies and elements of Industry 4.0 (AHUETT-GARZA & KURFESS, 2018)

This concept of "Industrie 4.0" has become a vision and a goal across industries. In the construction industry, achieving Industry 4.0's objectives will be partially accomplished through the adoption of AM. In this context, FIM will play a crucial role. FIM bridges the gap between planning and execution, aligning with the Industry 4.0 vision. It also has the potential to leverage real-time information to provide immediate feedback into the manufacturing process, allowing for the creation of instant solutions.

2.1 Additive Manufacturing (AM)

AM is an alternative process to conventional manufacturing by automating processes and using 3D model data. It enables the creation of complex shapes and structures, reducing material waste and offering design freedom. AM is a "computer-controlled process that creates three-dimensional objects by depositing material, usually in layers" as explained by CORRADINI and SILVESTRI, 2022. 3D printing and automated bricklaying are examples of AM. Initially, AM was for small-scale prototyping and has since grown, making it possible to manufacture real-size prototypes and custom components.

Numerous variations of AM processes exist, encompassing material extrusion, powder bed fusion, jetting, and other techniques. AM accommodates a wide range of materials, spanning ceramics, polymers, metals, and even edible substances. The choice of AM method depends on both the intended material and the desired geometry. Among the various AM techniques, material extrusion is the most prevalent. For instance, in construction, shotcrete, as observed in Figure 2.2 represents an example where the material is precisely sprayed onto a designated surface layer by layer to obtain a desired geometry.

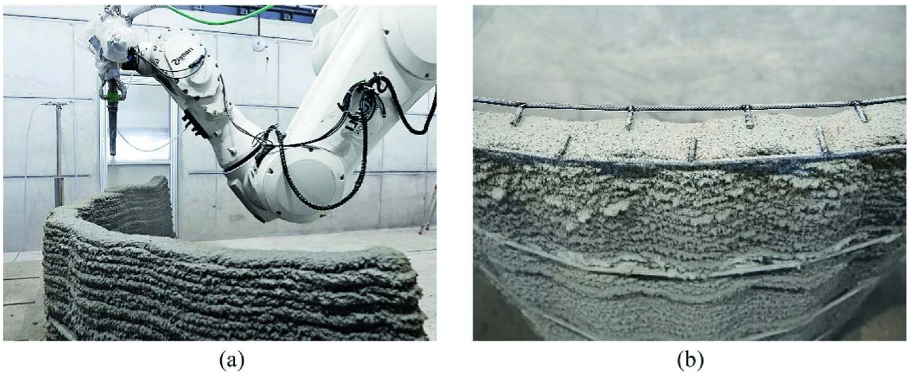


Figure 2.2: Shotcrete for wall building

For this study, only filament material extrusion is considered as it is the most common process. This process works as the name suggests by extruding material through a nozzle as a filament at a given location at a specific speed layer after layer to create physical objects as seen in Figure 2.3.

2.1.1 AM in the construction industry

AM promises to improve efficiency while improving sustainability and reducing costs. Companies worldwide have introduced industrial 3D printing as a method of building structures. From Denmark with COBOD to the United States with ICON. Research and development are still ongoing from both the companies' side and from the academia. Both race to determine the best and most effective way to use AM technology to address the labor shortage, housing shortage, and sustainability goals.

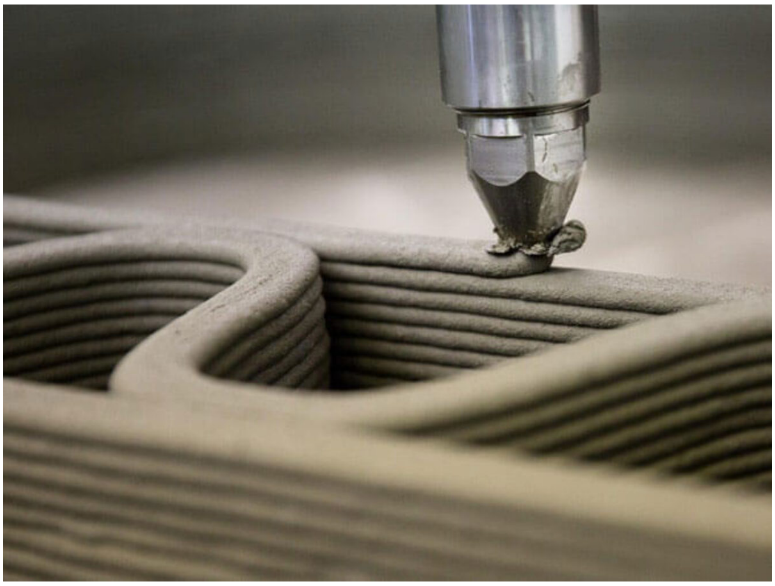


Figure 2.3: Extrusion based 3D printing

2.1.2 State of the art

Other studies have investigated other AM methods, such as shotcrete and the combination of formative and subtractive post-processing techniques to improve surface quality (HACK & KLOFT, 2020) as observed in Figure 2.4. A surface finish is implemented by moving material and/or adding additional material and distributing it with rotational disks and a robotic arm. This was a different approach than that studied before by SERDECZNY et al., 2019, where the surface finish was improved by a decrease in layer thickness, printing speed, or nozzle diameter and not by post-processing techniques.

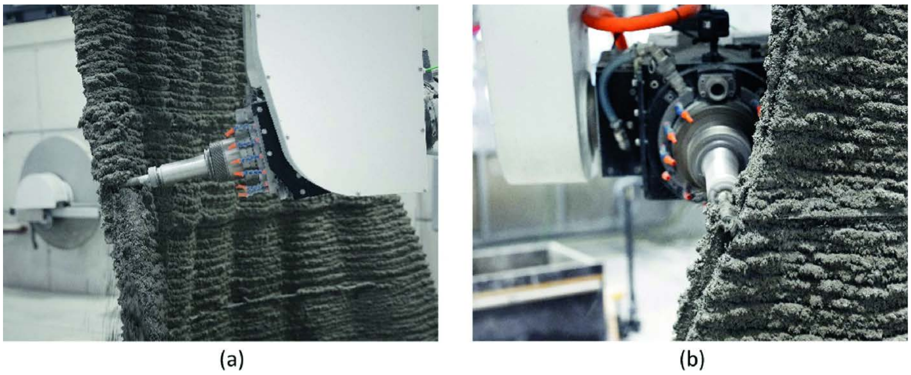


Figure 2.4: Material removal for surface finish (HACK & KLOFT, 2020)

In addition to the surface finish, literature has studied multilayer prints to understand the benefits and weaknesses of wet on wet, wet on dry, and all the printing states in between. Studies have been carried out with simulations (SERDECZNY et al., 2019, MOLLAH, SERDECZNY, et al., 2022, REINOLD et al., 2022 MOLLAH, COMMINAL, et al., 2022) and experiments (BUSWELL et al., 2018) to explore the relationship between layers and sublayer deformation while using fresh or hardened paste when extruding the subsequent layer. This relationship is significant due to the risk of plastic collapse when 3D printing with a viscoelastic material KRUGER et al., 2020.

Other studies have done detailed research on the effect of extrusion, the movement speed, and the distance between the nozzle and the previous layer, among other factors, to understand better how each variable affects the print. These studies have been done mainly at the *Technical University of Denmark* and include CORRADINI and SILVESTRI, 2022, COMMINAL et al., 2020a, COMMINAL et al., 2020b, among many others. One of the key factors that was taken into account but is outside of the scope of this study is the curvature of the print in the corners and turning points. It is crucial during printing, as explained in COMMINAL et al., 2019 and seen in Figure 2.5.

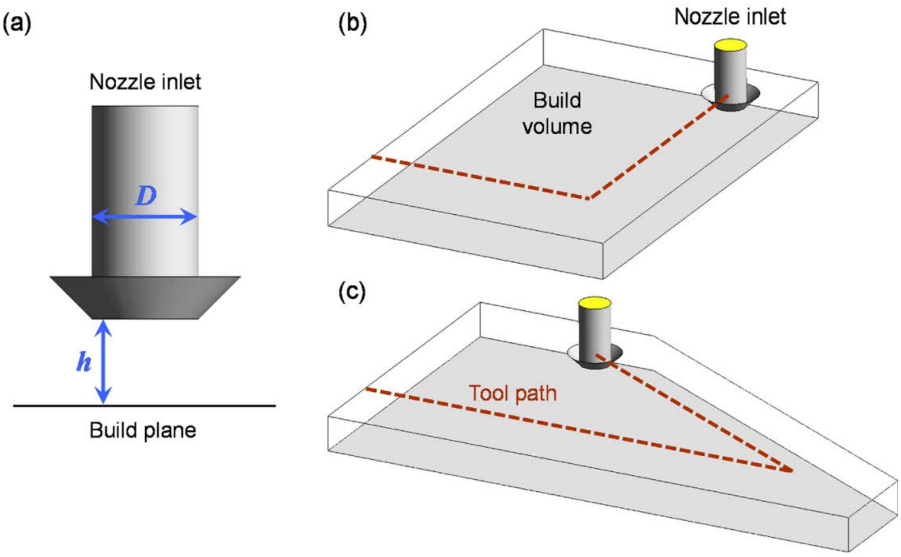


Figure 2.5: 3D printing corners (HACK & KLOFT, 2020)

2.2 Fabrication Information Modeling (FIM)

FIM is an intermediate stage between digital design and automated fabrication. The aim is to bridge the gap between BIM and the automated construction. In addition, it will also fill the gap between BIM and the finished structure. This is of utmost importance as the as-built object is usually not identical to the as-designed model, and for facility managers or for future works, it is necessary to understand to what extent the final construction deviates from the planned model.

Furthermore, FIM is considering the way a structure will be built, different from BIM that contains the design, FIM has a detailed path and coordinates where the material will be extruded; more importantly, it has information on when and the specific stages of construction. For FIM, the design for construction is the most critical process.

The FIM data structure consists of several key elements. First, the components that represent individual parts, these components have boundary representation. Such components are then discretized into the second key element, the layers. Layers contain details such as surface information and path. Finally, paths are represented as composite curves, which are a combination of simple and complex curves. Overall, the FIM data structure is designed to provide a detailed and structured representation of how components are fabricated, assembled, and processed. SLEPICKA, VILGERTSHOFER, et al., 2022

Industry Foundation Classes (IFC)

IFC is designed to represent building and construction data in a neutral and interoperable manner. The data structure follows a hierarchical organization and has the following key features.

- Entity-Relationship Model: IFC is based on an entity-relationship model. Entities represent real-world elements, such as walls or doors, while relationships define how these entities interact with each other.
- Extensibility: IFC is extensible, which means it can be customized to meet specific needs for any project.
- Geometry Representation: IFC allows the representation of 3D geometry for building elements and their visualization. BORRMANN et al., 2018

2.3 Point Cloud (PC)

A PC is a collection of discrete 3D points that collectively represent a physical object in space. Each point is defined by its x, y, and z coordinates. These points are often unorganized within the cloud, meaning they are not arranged in any specific order or pattern. Additionally, each point in the cloud can contain additional information, such as a normal vector, color, or other relevant data.

The visualization of a typical point cloud is shown in Figure 2.6. The level of detail in a point cloud is directly related to the number of points it contains, where higher resolution is achieved with more points. Researchers have extensively studied techniques for segmenting, downsampling, and storing such point cloud data, but these topics are beyond the scope of this research.

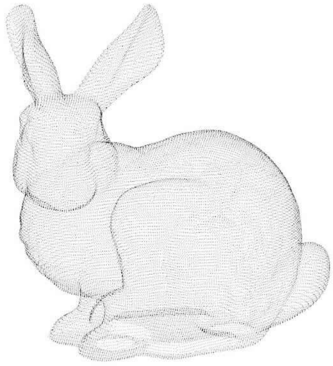


Figure 2.6: Stanford Bunny point cloud

Point clouds serve as valuable representations of physical objects, and they find extensive use in various fields, including computer vision, robotics, 3D modeling, and more. Their ability to store detailed spatial information makes them crucial in tasks such as object recognition, reconstruction, and localization.

2.4 Laser Scanning

Multiple technologies can capture point clouds, eg. stereo RGB cameras or laser profilers. In this thesis, the focus lies on laser profilers.

The sensor used works by projecting a laser line as seen in Figure 2.7 and capturing the reflection to create a 2D profile of the object in sight. Additionally, as the sensor moves, it starts to generate a surface out of the accumulated line profiles. The output is a list of 3D coordinates that generate a surface, and more information on the sensor is in section 4.1.

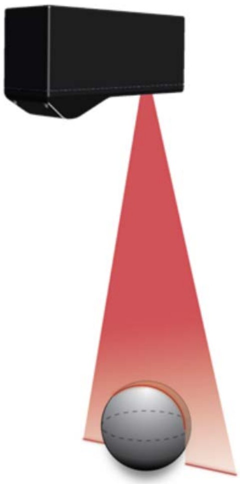


Figure 2.7: Line scanning

The result may be observed in Figure 2.8, where the sphere is scanned and a circle is obtained as well as the table. It is important to point out that only the reflected data may be obtained, and hence the sides of the sphere are not part of the output. For this reason, the scanning path and orientation are crucial for the result.

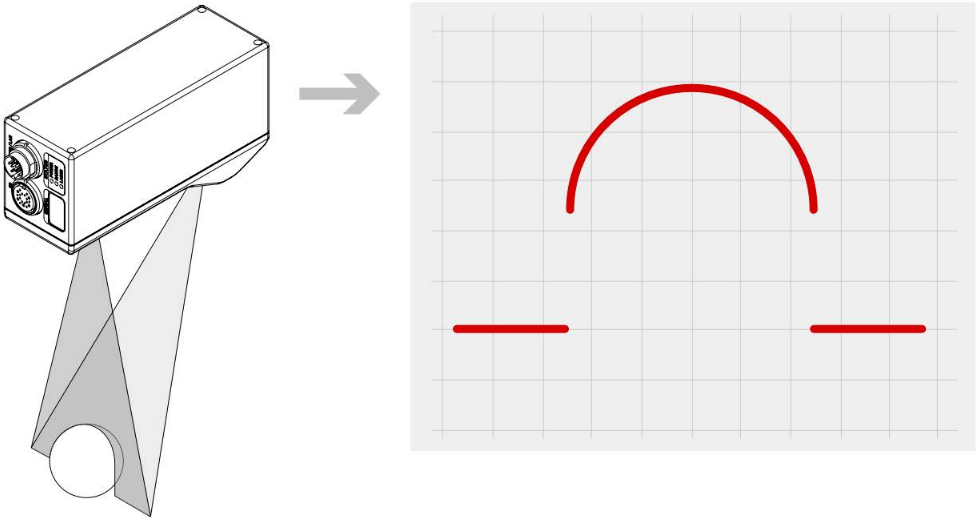


Figure 2.8: Line scan and result

Additionally, Figure 2.9 shows how the surface is composed of multiple consecutive profile scans. The figure shows the target moving but in the case of the study, the target is static and the sensor is moving, this is technically the same. The output of each scanned profile is a 3D PC, it uses the given speed and frequency of the sensor to detect the displacement and the spacing between profiles and give a fully accurate PC surface.

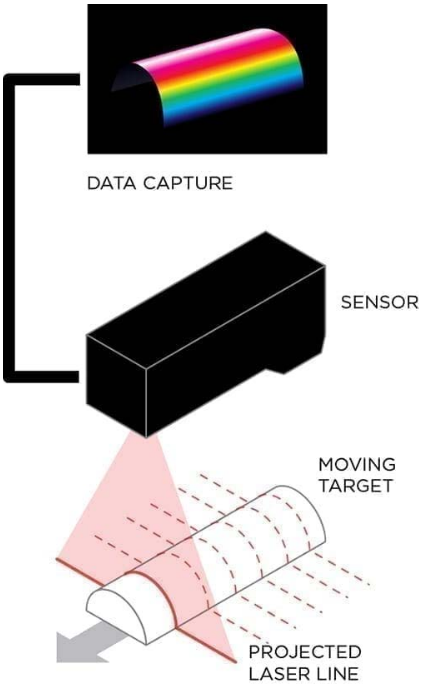


Figure 2.9: Surface generation from laser profile

2.5 Digital Model, Digital Shadow and Digital Twin (DT)

In many industries, the term *digital twin* is used, and its definition can vary depending on the specific context. However, for the purposes of this study, a digital twin is defined as an automated, bidirectional synchronization between a physical object and its digital counterpart. This concept is illustrated in Figure 2.10, which depicts the data flow and the various technologies utilized in each step.

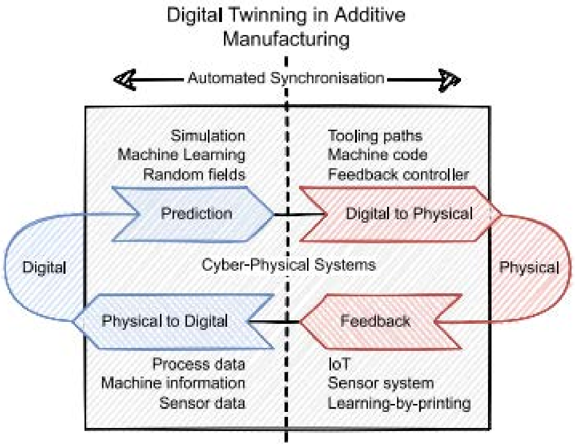


Figure 2.10: Circular flow of information for a Digital Twin in Additive Manufacturing (SLEPICKA, MAWAS, et al., 2022)

In this study, the concept of a digital twin is explored in the context AM in the construction industry. By employing automated synchronization, the digital model can identify variations in the physical model and trigger a response with feedback information to generate improvements and keep both objects updated. The automated response has not yet been implemented.

In addition to the concept of a DT, two other terms are utilized in a similar context: Digital Model and Digital Shadow, as illustrated in Figure 2.11.

A Digital Model involves manual data updates in both directions between the physical object and its digital counterpart. This is typical for BIM models after construction, where changes are manually recorded and incorporated into the digital representation.

On the other hand, a Digital Shadow exhibits an automatic data flow that updates the digital object with information from the physical object. However, there is no automated feedback to update the physical object, resulting in a one-way manual data flow from the digital object to the physical one. This scenario is common in automation, where the digital model can update using sensors but lacks the capability to automatically trigger a response in the physical object.

Finally, to achieve a fully automated data flow from the digital model to the physical model and vice versa, AM is utilized to make the required updates necessary for a complete Digital Twin implementation.

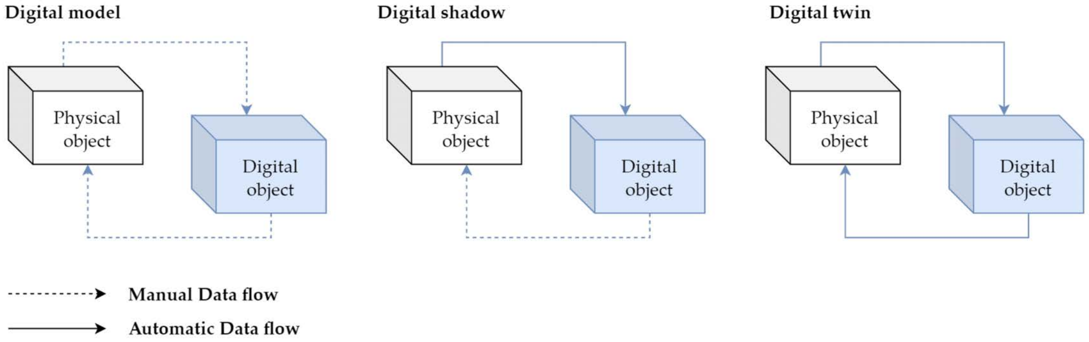


Figure 2.11: Data flow comparison between DM, DS and DT (Balla et al., 2023)

The state-of-the-art has used sensors to capture the as-built state of viscoelastic materials for AM. These studies have been primarily performed to create and test computational models for Computational Fluid Dynamics (CFD) and Finite Element Methods (FEM) to simulate how the extrusion and settlement of the layers will behave during the printing and drying process (Comminal et al., 2019 and Reinold et al., 2022). The motivation for these studies is; first, the lack of innovation in the construction industry compared to different industries that have used AM for years (Slepicka, Vilgertshofer, et al., 2022) and the second reason is due to the requirements of "Industrie 4.0" that promote the long term development for the manufacturing sector (Corradini & Silvestri, 2022).

2.6 Theoretical framework

The theoretical framework of this study arose due to the continuous development of FIM tools and the need to store more information about the as-built object within the digital model. For such reason, a 3D printer is used with clay to study on a small scale what would then be done with a concrete 3D printer. In order to use FIM as a DT as described in section 2.2, both the digital and its physical counterpart must be synchronized. To do so, an automated scanning process is introduced, and post-processing for the data implemented.

The research seeks to investigate the best way to capture the "as-designed" and the "as-built" deviations in the FIM for future use. The study includes accurate measurements done with a profile laser scanner on a fresh print with known dimensions and a known FIM history where the planned information is compared to the printed measurements. The data is then analyzed using a Python implementation and stored as images of the as-built deviation. The information is stored as a color map that can quantitatively and qualitatively indicate such deviations in shape and height maps, respectively. Additionally, for sustainable and economic reasons, a volumetric study is done to find the amount of material required to post-process the built object to have a flat outer wall.

Chapter 3

Method

This chapter states the methodology used during the study. The technology used and its limitations.

The study is based on the measurements from a physical sample that has been freshly printed from a digital model. Measurements are taken in a fresh and dry state. During the printing process, four measurements were performed from the top and one additional measurement on each side once the full print is finished. Four days later, three measurements are performed on the dry specimen. One from the top comparable to the last scan from the top of the fresh specimen, and the two lateral scans comparable respectively with their fresh counterparts.

Information during the printing process is essential as it provides insight into the progress and state of the print at different stages. For such reason, the four measurements done after 14, 17, 20, and 21 layers respectively, are part of the study. Additionally, the dry data also provides insights into the drying process and how the implementation is able to measure the end result.

This research combines quantitative and qualitative results. The reason for this is that some variables, as explained in more detail in section 3.4, are easy to measure, but more importantly, the measured variable makes sense as it gives some indication of the state of the print.

On the other hand, other variables are more important in a visual qualitative aspect as they are meant to show the lack of or excess of material. The fact that they are not quantified does not reduce their importance as it is mostly to guide the user into understanding the general quality and state of the physical model.

3.1 Setup

For this study, the setup consists of four main systems. Two UR robots, one extruder, and one laser scanner. The Gocator laser line scanner is attached to a small UR robot with an additional longitudinal degree of freedom as seen in Figure 3.1.

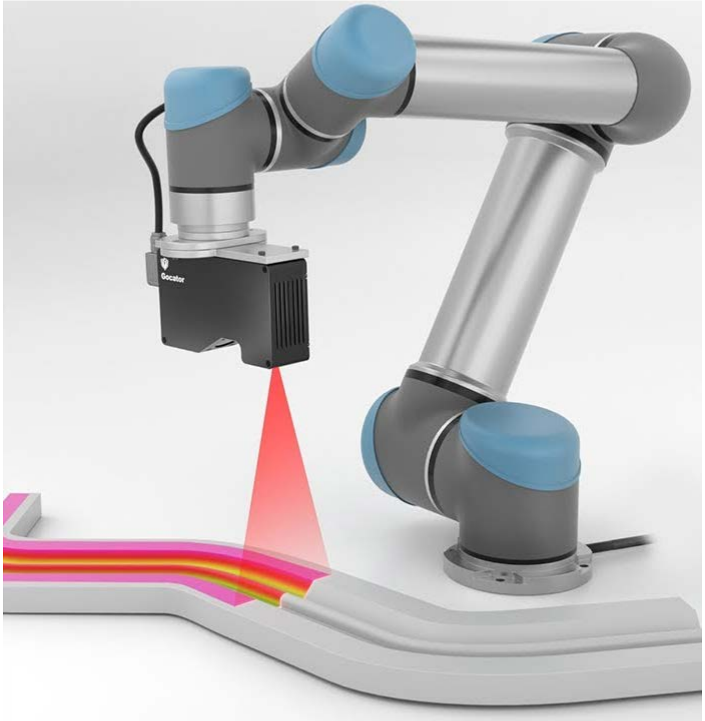


Figure 3.1: UR robots and extrusion setup

The extruder is attached to the larger UR robot. In Figure 3.2, the robotic arm with an extruder can be observed. The specifications of the extruder and its attachments are outside of the scope of this study. Still, it is essential to mention that the extrusion speed and extrusion angle can be adjusted in real-time, just as the position of the nozzle.

Figure 3.2 shows the full picture of the setup where the two robots can be seen, the longitudinal axis on the back with the small robot attached to it and the Gocator scanner on the robotic arm. On the opposite corner, the larger robot has the extruder attached to it. Both robots are securely placed on a level table.

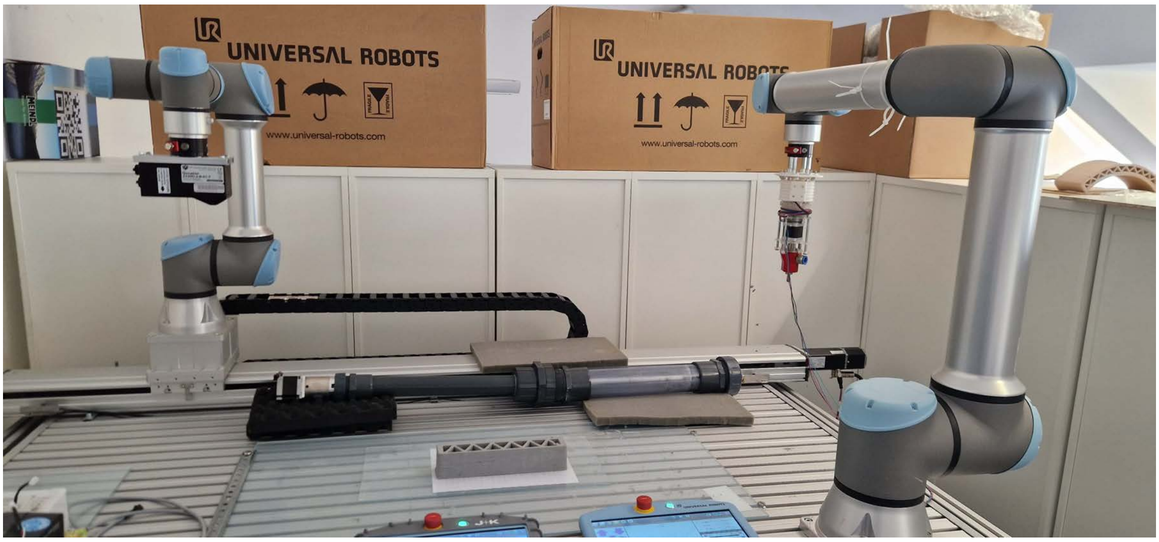


Figure 3.2: Full setup at home position

3.2 Benchmark experiment setup

For this research, a 30cm long and 6cm wide sample was chosen. It is extruded with a 6mm nozzle for 21 layers with a 3mm layer height. The expected print should be as depicted in Figure 3.3. Such structure is printed with the setup of the first robot as explained in chapter 4, but the general idea is that it follows a given path illustrated as lines in Figure 3.4 while extruding material.

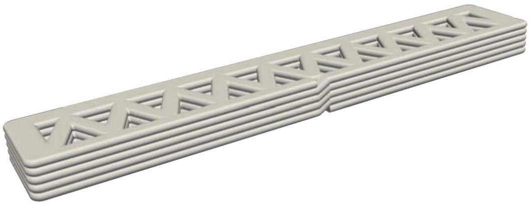


Figure 3.3: Designed structure

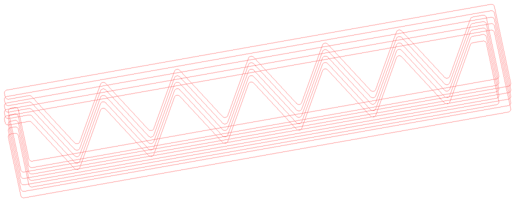


Figure 3.4: Planned path for printing

The sample features an internal infill designed in a zigzag shape. This serves two purposes: ensuring operational structural integrity and offering stability during the printing process to prevent a slender vertical wall from collapsing while still keeping a low weight. The sample material consists of standard gray clay, manually mixed and loaded into an extruder cartridge as seen in Figure 3.5. However, it is important to note that this study was limited to the current design and material, and as such, the design and the material were not analyzed or described in more detail.

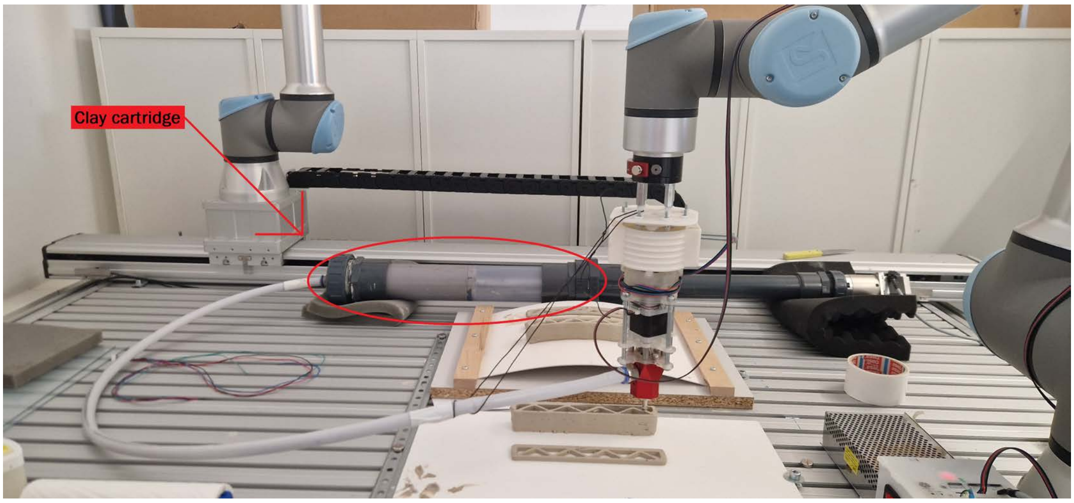


Figure 3.5: Clay printing cartridge

3.3 Data acquisition

For data acquisition, a Gocator Line Profile Sensor 2330 is employed in conjunction with a UR robot. The Gocator sensor is securely attached to the robot, and the scanning process is executed by moving the sensor from the start to the finish point using the URCap. The connection between the Gocator and the UR robot is established through Dynamic Host Configuration Protocol (DHCP).

During the scan, the Gocator sensor moves at a speed of 60mm/s, accompanied by a high acceleration of 1000mm/s. This significant acceleration ensures that the scan is performed at a constant speed. It is crucial for the robot to reach maximum speed before reaching the scanning region to maintain uniform scanning performance. This specific requirement ensures accurate and reliable data acquisition throughout the scanning process.

The resolution of the scanner is so high that it is able to accurately capture air bubbles in a 3D-printed clay structure. This is illustrated in Figure 3.6 and it is one of the main reasons such technology was chosen for the data acquisition in this research.

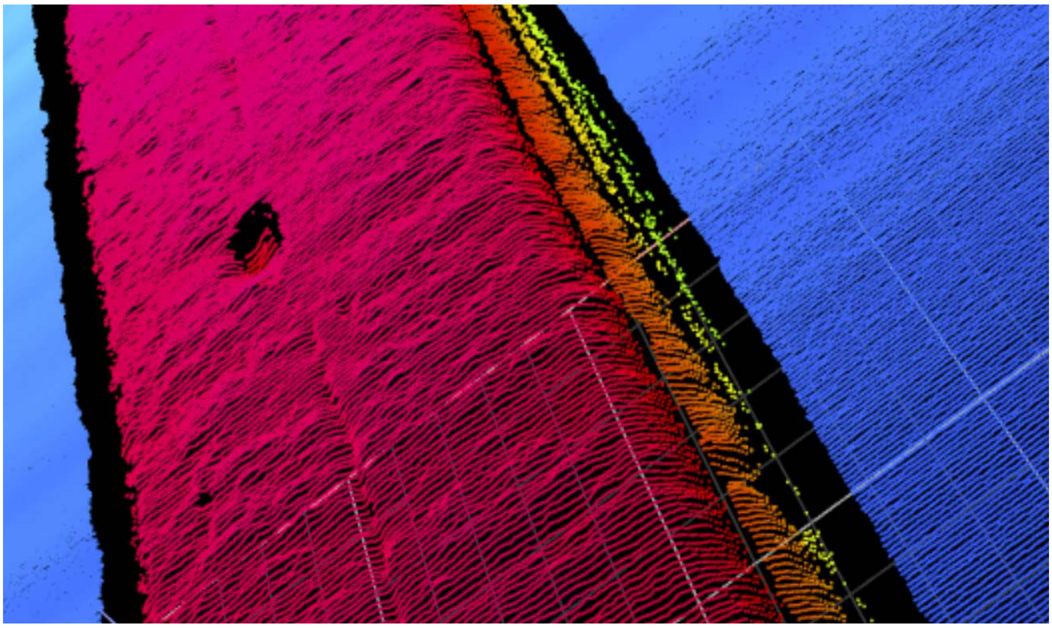


Figure 3.6: Air bubble in 3D filament

3.4 Variables and Measures

As mentioned above, three key measurements are made. A shape deviation, height deviation, and finally, a wall roughness. All measurements were derived from two independent scans. A scan of the last layer from above was used for the two first measurements, and a scan at a constant distance from the printing wall was used for the wall roughness.

3.4.1 Shape deviation

The shape deviation is a qualitative comparison of the printed shape compared to the designed shape. The printed shape is measured as explained in section 6.2, it is centered using the x and y edges and overlapped with the designed shape. The as-designed shape is initially a curve imported from the IFC and plotted with a line width to simulate the filament width. This is expected to have a perfect overlay, where the as-designed and as-built lines are identical, and the plotted line is completely on top of the filament.

Figure 3.7 shows the path planning for every layer that has been gradually extruded. This was then plotted with a thick line, such thickness is calculated with the dots per inch (dpi) to accurately represent the correct filament width. Once this is done, Figure 3.8 shows the as-designed plot, this is the expected result used to compare the as-built scan. Then, Figure 3.9 shows an example of an arbitrary scan of a similar structure. This one is an as-built object and was compared to the as-designed model to qualitatively check for differences, as already explained.

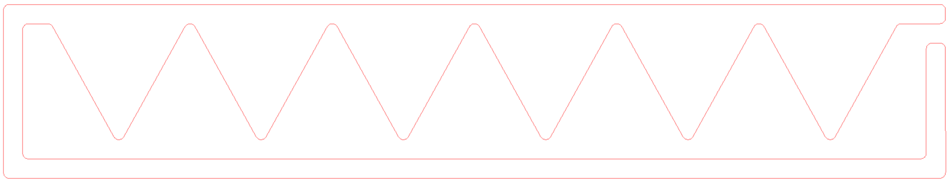


Figure 3.7: As designed path

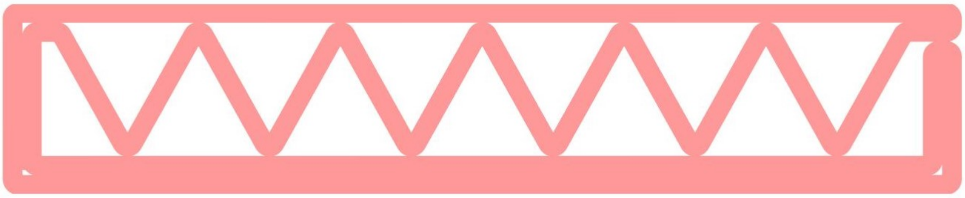


Figure 3.8: As designed plot respective line width

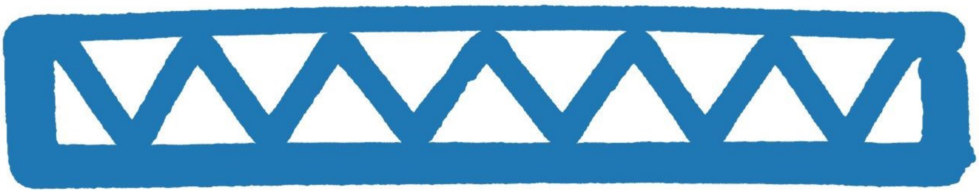


Figure 3.9: As printed scan from above

3.4.2 Height deviation

Here a plane at the expected layer height is calculated as the current layer times the individual layer heights. To generate the height deviation map, the scanner calculates the z difference between the expected and measured surface. Two color scales, a cold and a warm scale, are used to easily identify values that are below or above the expected height, respectively, as seen in Figure 3.10.

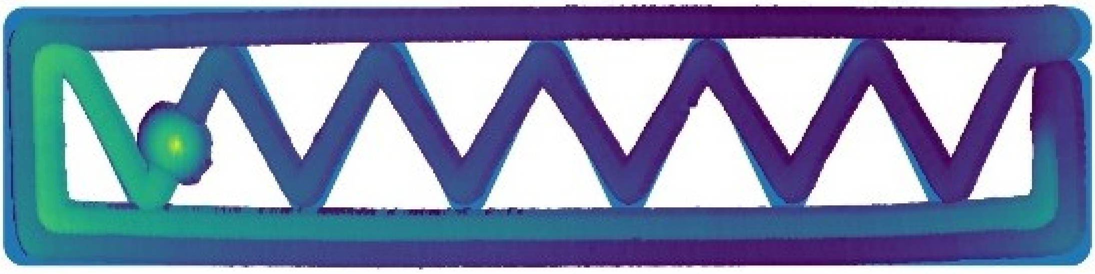


Figure 3.10: Height deviation example

3.4.3 Post-processing volume required

To measure the required volume accurately, the data was transformed by translating it across the depth = 0 axis. This translation ensures that the integral represents the required infill area for each profile. The integral of the scanned profile in Figure 3.11 illustrates the material that has already been extruded. To calculate the infill area needed, Figure 3.12 is used, which presents the same information but on the opposite side of the 0 depth axis. This translation causes the integral of this curve to represent the infill area required. It is essential to note that during the translation, the critical point is consistently maintained at a 2mm distance from the zero-depth axis. Consequently, the minimum infill at any given point is always 2mm.

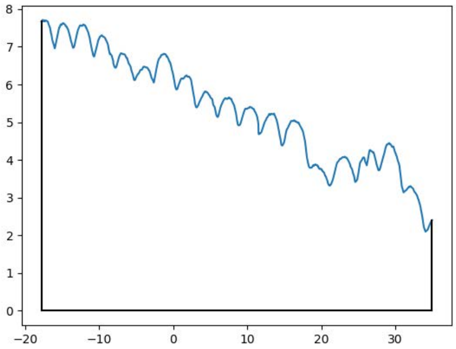


Figure 3.11: Scanned profile

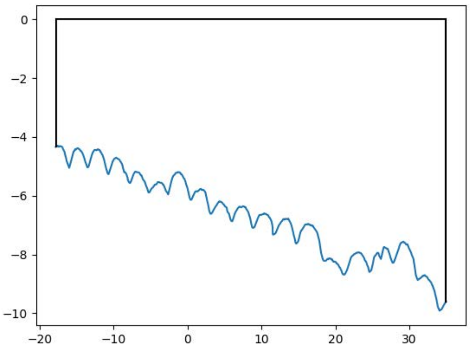


Figure 3.12: Translated profile

The scan data is expected to be as depicted in Figure 3.13. This is the combination of multiple scan profiles as explained in Figure 2.9. The required volume is the average of the area of each scanned profile as shown in Figure 3.12 and more clearly in gray in the schematic in Figure 3.14 multiplied by the total wall length.

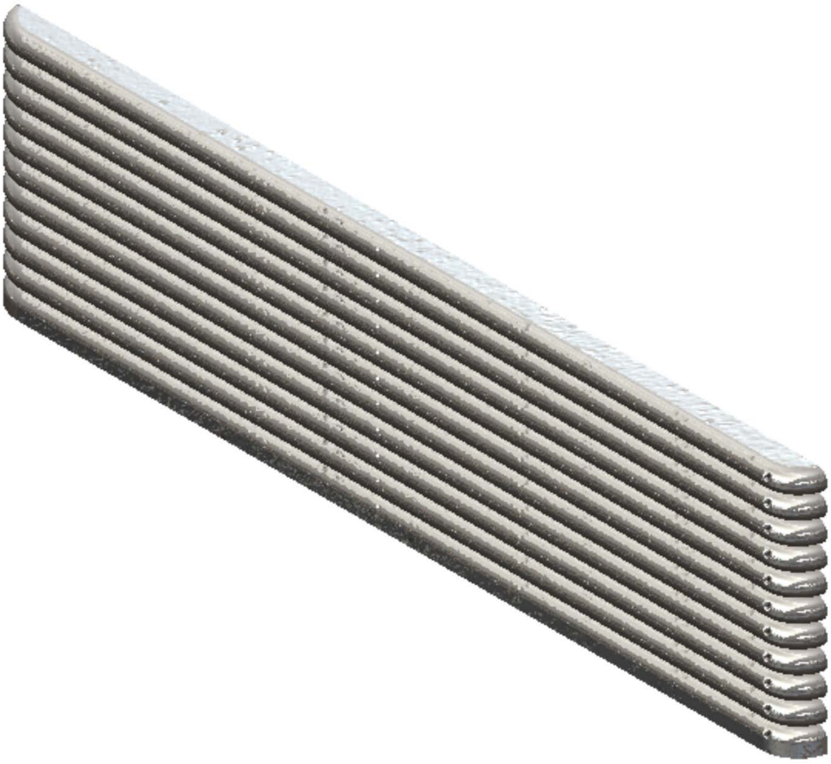


Figure 3.13: Lateral scanned volume

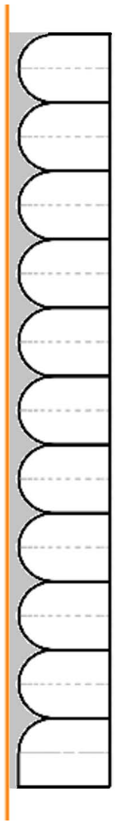


Figure 3.14: Profile area measured

3.5 B-Splines approach

Initially, the plan was to utilize B-Splines and Non-uniform rational B-spline (NURBS) to store the lateral scan information, offering significant storage savings and enabling efficient in/out testing to determine if points were inside or outside the surface. However, while this approach seemed promising in theory, practical implementation encountered several challenges.

The first issue arose with storing the top scan data, as NURBS surfaces cannot handle openings in the surfaces. This would have required using NURBS patches for each section, introducing complexity and alignment problems.

Furthermore, dealing with the massive amount of point cloud data led to a considerable computational cost when calculating the NURBS surfaces. The iterative nature of B-Spline generation in two directions contributed to this issue. To address this, the Point Cloud Library (PCL) was employed for NURBS generation, but it was only compatible with Ubuntu 22 and had complex compilation requirements. To ease integration with other libraries, a Docker environment was created, adding further complexity to the setup.

Due to these challenges, the decision was made to forgo the use of NURBS surfaces and PCL. Instead, the implementation was shifted to Python, as explained in chapter 5. This change allowed for a more streamlined and feasible approach to tackle the lateral scan data storage and manipulation tasks.

3.6 Positioning

Since the two robots do not collaborate intelligently but operate independently, the positioning is done by co-aligning the planned and scanned models using software. Nonetheless, it is important to point out that the UR10e is positioned at a corner. This allows better reach for both robots as the UR5e has a linear axis and can move along the length of the table. Additionally, it is important to position the robots in a way where the arms are never fully extended to avoid singularity issues [1]. As this study does not dive into the printing process, it does not go further into the positioning of the UR10e except for the fact that it prints in the direction parallel to the linear axis of the UR5e. This is done for two reasons. First, as has been previously mentioned, the scanning is done in that direction when using the URcap. The second reason is that it gave the UR5e a better reach, as it can most optimally utilize the linear axis as the arm's reach is more limited due to its size.

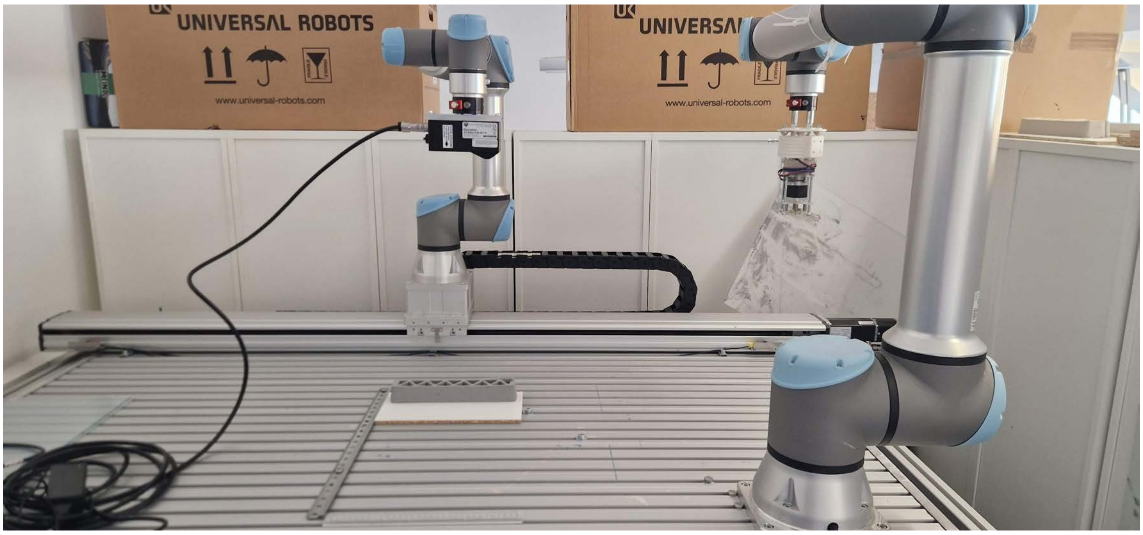


Figure 3.15: General scanning and printing position

For the lateral scan, the scanner is oriented at approximately 60° from the table Figure 3.17. This has been found to be an angle where the filament's visibility on the wall's exterior side is maximized, allowing the full ripple to be observed layer after layer. For this scan, it is important not to align the camera using the default Gocator function, as it eliminates points due to uniqueness, as explained in section 3.8. The scanning process is done at 60mm/s and in a straight line for 1.2 times the actual length of the print allowing the scanner to reach a constant speed when scanning the object and measuring the reference plane. The angle used for the transformation matrix is the angle measured by the scanner of the reference plane seen in Figure 3.16.

[1] When a robotic arm is fully extended, the next position can be reached with several configurations, this generates an error as the robot is not instructed to take one specific configuration leading to a singularity.

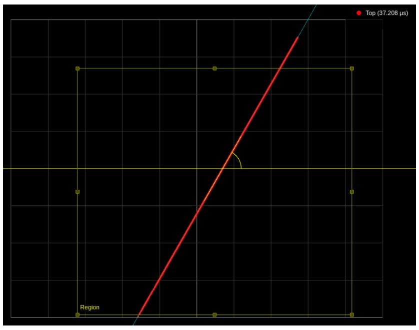


Figure 3.16: Angle of the scan

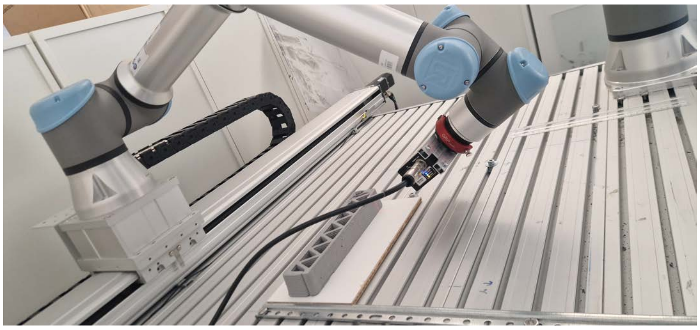


Figure 3.17: Lateral position

For the upper layer scan, the laser scanner is centered on top of the print at around 10cm and then moved towards an edge to start the scan as seen in Figure 3.18. The scanning process is done as explained above, but two times. An initial scan of the table is first done perpendicular to the print to check that the table is flat and/or the robot is mounted leveled horizontally; if any deviation is measured, it is used to offset the following scan. The results are stored for further processing.

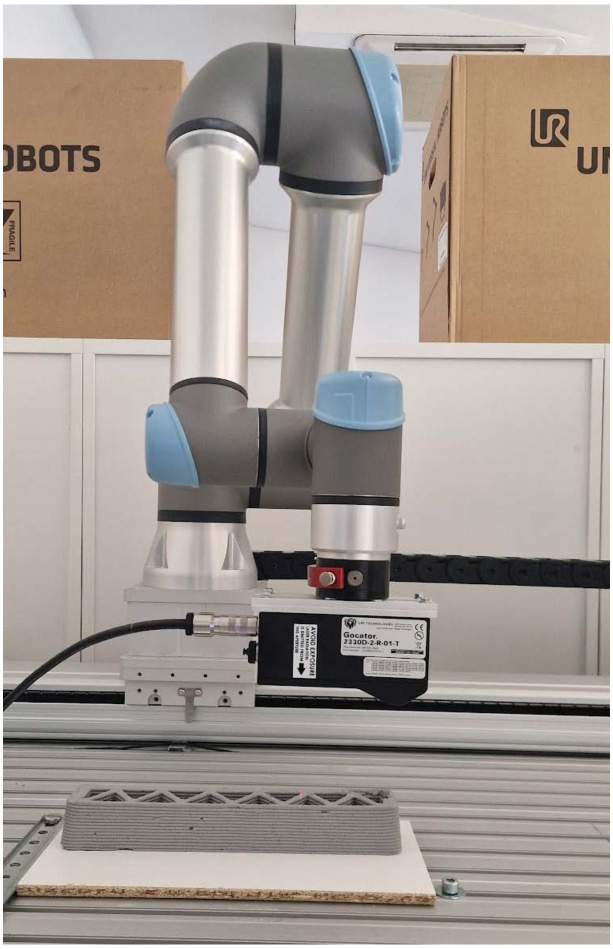


Figure 3.18: Top position

3.7 IFC Texture approach

Initially, it was planned to store the results in the IFC file as a texture. This would allow all the information relating to the as-built and as-designed in one place. Additionally, it would integrate the results and allow for easy visualization. The implementation was planned around the *IfcImageTexture* format, which allows a 2D image representation mapped onto a surface. After different implementation attempts, this was not possible, resulting in an external storage option as it has been implemented and described here.

Figure 3.19 show how the IFC tree works, where the *IfcImageTexture* would be a *IfcStyledItem* within a layer. This is extremely important as it depicts where the information was planned to be stored and how to visualize it afterward. As the results are stored in textures, it is possible to store multiple results in one single layer and visualize them independently, allowing different information to be stored and represented in the same location.

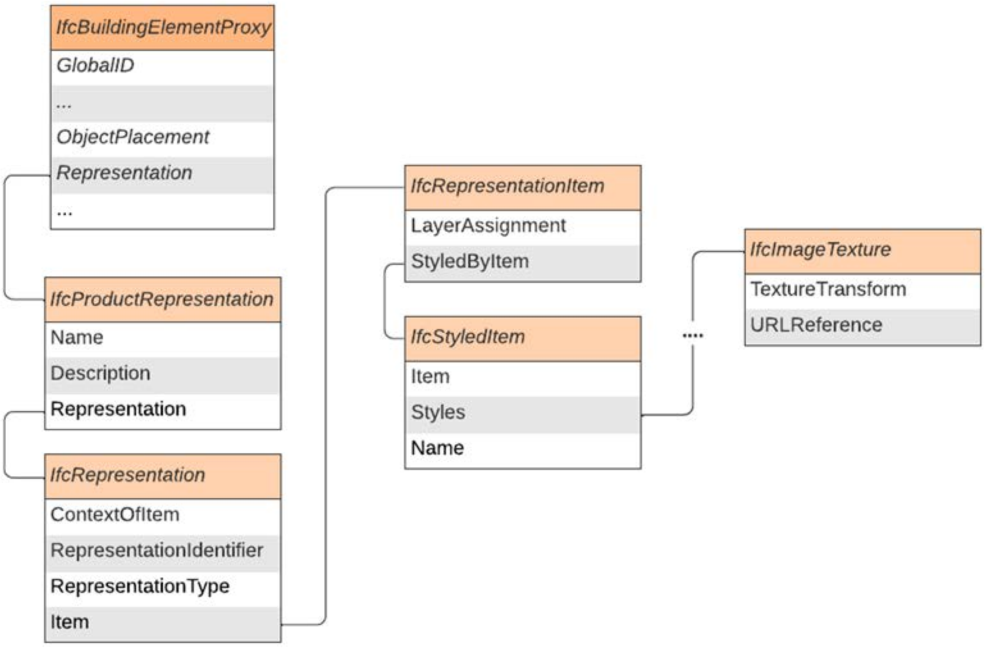


Figure 3.19: IFC for as-build information (Theory)

3.8 Limitations

As explained in section 3.3, the laser scanner only registers the projected view in which the laser hits the physical object. It has occlusions depending on the respective scanning angle and the target surface. Additionally, because the printing process stops during the scanning process, this creates a layer cut, and, as a result of the current RTDE implementation, the transition is not printed once the printing process is resumed from the start of the next layer.

An additional limitation is when it comes to the referencing between robots. This has been done by assuming the printed part and the designed model have the same center. This is not necessarily true and would be a potential error source when overlapping the models. To overcome this, a high-level transformation would be necessary to have the tip of the extruding robot and the scanner on the same coordinate reference system. In addition, the width of the planned layer is done using dpi when plotting the thickness of the planned path. Therefore, some errors can be induced here, especially at the turning points.

Finally, a limitation is that the laser scanner has a unique z coordinate for every pair of x and y, see also section 3.6. This means that if the scan is made at an angle, the point cloud is also at an angle, as expected. If an alignment is done within the Gocator framework, the new PC is again unique in z, and information is lost. This can be seen in Figure 3.20 and Figure 3.21. Therefore, it is best to do the alignment during the postprocessing. An angle is then measured with the Gocator to know how to construct the transformation matrix; this is explained in chapter 5.

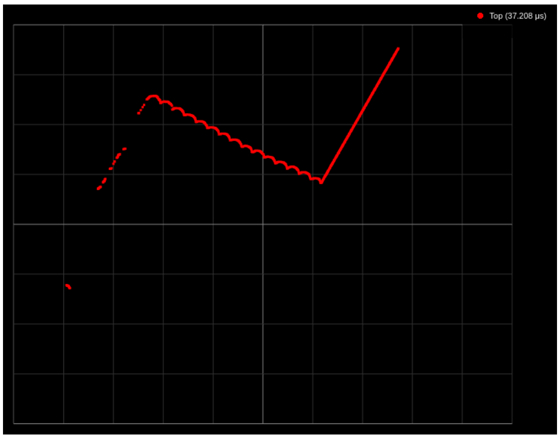


Figure 3.20: Wall scan at an angle

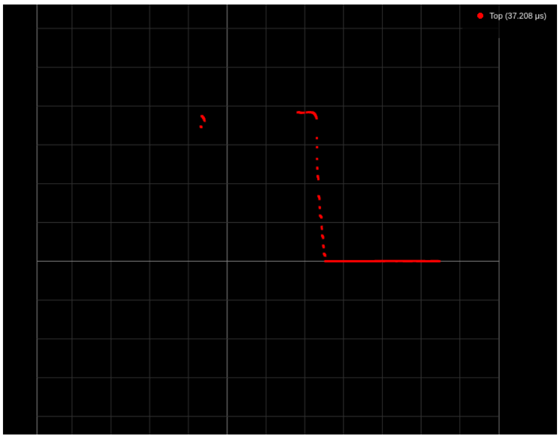


Figure 3.21: Wall scan aligned

Chapter 4

Hardware and Software

This section will go through the experimental setup. It will explain how the two robots work with their respective adjustments, how they are positioned with respect to each other, and it will slightly explain how they move. Furthermore, it will also explain the software used and some additional technical information about the robots, the sensors, and the extruder.

4.1 Harware

The hardware here is a key element going forward as it is crucial for understanding the robotic arms' reach and the laser scanner's scanning capabilities. As previously mentioned in section 3.1, the extruder will be neglected, as other studies have already researched the importance of extrusion speed and other variables directly related to the print quality, as explained by MOLLAH et al., 2021.

4.1.1 Universal Robots

Universal Robots are one of the market's largest commercial robotic arm robots. They are known for their flexibility as they are used in various industries. They are designed with collaboration in mind, as well as with a very high level of safety and a user-friendly environment. For the study, the two robots used are the UR 10e and the UR 5e. In Figure 4.1 the size difference between the robots is clearly visible. This is important because we use two different versions and the reach is proportional to the size.

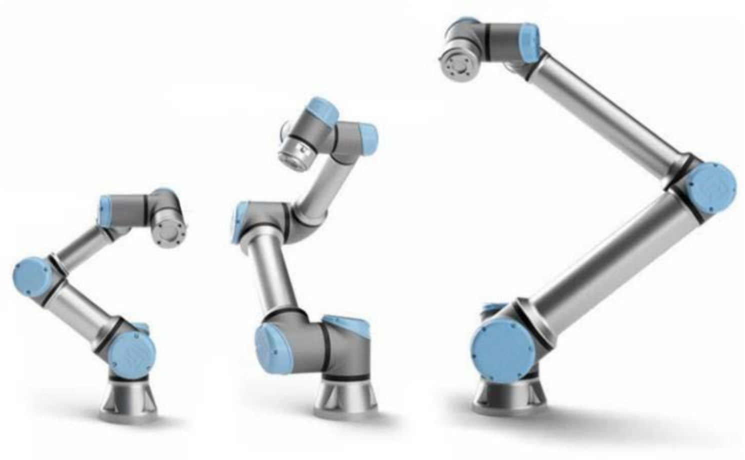


Figure 4.1: UR series e

In Table 4.1 the main specs of the two robots are displayed. It is clear that the extruding robot (10e) has a greater reach, nevertheless, as explained in section 3.1, the smaller robot (5e) is attached to a longitudinal axis as also seen in Figure 3.2, which allows the scanner to have a greater reach.

Table 4.1: UR main specs compared

Model	Reach (mm)	Payload (kg)	Weight (kg)	Footprint (mm)
UR 5e	850	5	18.4	149
UR 10e	1300	12.5	33.5	190

A requirement when working with collaborative schemes is that the robots should not crash at any time. This is ensured with individual path planning and mutually exclusive operations, and in the future a real-time position monitoring between systems. This has not been implemented as it is outside the scope of the study, but the theory states that the robots should know where the other is at all times and have the respective transformation matrix to transform between coordinate systems. Robots must also know about their attachments and other objects to avoid unexpected collisions.

4.1.2 Gocator laser scanner

Table 4.2 presents the key elements for the Gocator laser profiler, a high-resolution scanning device, while Figure 4.2 provides a clear visual representation of the meaning of each variable. Understanding these elements is crucial as they determine the scanning range and, more importantly, the quality of the scan. Being within the specified range is of utmost importance to maximize the utilization of this advanced scanning device and ensure accurate and high-quality data capture.

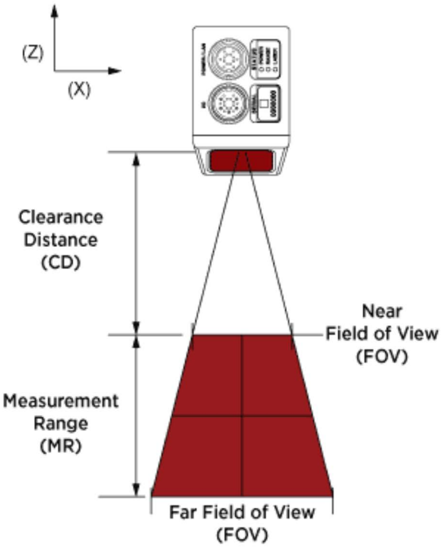


Figure 4.2: Gocator specs

Table 4.2: Gocator 2330 main specs

Model	2330
Resolution X (mm)	0.044
Resolution Z (mm)	0.006
Field of View (mm)	47 - 85
Clearance Distance (mm)	90
Measurement Range (mm)	80

As indicated in Table 4.2, the Gocator laser profiler has a depth resolution of 0.006mm and a longitudinal resolution of 0.044mm. It is capable of generating up to 1280 points per profile. These features enable precise and high-density point cloud data capture.

The Gocator scans by using the URcap. It requires a starting position, a speed, and a distance to scan. With this information, the Gocator generates a PC of the scan that is then exported and used as an input for all the calculations explained in chapter 5.

4.2 Software

Here, the integration of technologies and libraries is described. The functions and/or methods of the different libraries used are also justified.

Robotics and Real Time Data Exchnage (RTDE) Repository

The Real Time Data Exchnage (RTDE) repository serves a crucial role in this study, facilitating the movement of the printing robot. The repository effectively communicates with the IFC to extract print information and handle various tasks, including robot path planning, layer transitions, and control, as explained by SLEPICKA et al., 2023. Although the study does not focus on the RTDE and only uses it for printing, it offers the potential for more extensive applications in the future development of scanning and for adjustments while printing.

IfcOpenShell

This open-source library allows the interaction between Python and IFC files. It is the primary library used for the RTDE explained above, and the as-designed information for each layer is extracted from the IFC file also using this library.

Gocator web app

As previously mentioned, the Gocator scanner employed in this research is characterized as a smart scanner. The web interface allows users to perform measurements and incorporate different smart tools into the sensor for receiving feedback or triggering events. In this study, a line tool is used to measure the precise angles for the lateral scan. To ensure versatility and independence from the sensor, all processing is conducted externally, and the code can be extrapolated to other sensors if required. Limitations related to data loss induced by alignment in the Gocator interface have been discussed previously in section 3.8.

Gocator URCap

The Gocator URcap is instrumental in controlling the Gocator scanner and coordinating its actions with the UR. The URcap is loaded into the Universal Robots (UR) teach pendant. It provides a user-friendly Graphical User Interface (GUI) for convenient programming. In this study, two essential functions are utilized: *Gocator Connect* establishes the IP connection with the Gocator and transmits information to the sensor, while *Gocator Scan* instructs the robot on its speed, acceleration, and distance to move. These functions enable accurate surface generation and control of the scanning process. These functions don't allow to modify the scanning path, as it has already been stated.

Python Libraries

The implementation has been done in Python, and different libraries have been used to fulfill post-processing requirements.

Numpy

Numpy is a powerful Python library utilized in this study for numeric calculations and array manipulations. It leverages efficient C implementation to enhance performance and data management. Numpy's extensive capabilities enable various essential tasks in this research, including storing point cloud data in arrays, identifying points of interest, modifying the data structure, and conducting vector operations, among many other functions. By utilizing Numpy, the study achieved increased efficiency and reduced the complexity of data handling and processing, resulting in an effective data analysis and manipulation.

Matplot

Matplotlib is a versatile Python library employed in this study for data visualization and image generation. It is widely recognized and utilized for its extensive functionalities, making it an ideal choice to effectively communicate data and results. Matplotlib integrates seamlessly with Numpy, allowing for straightforward data handling and plot generation using Numpy arrays. This synergy between Numpy and Matplotlib streamlines the process of converting numerical data into informative visual representations. Additionally, it allows plots that were used in this study to be stored as image files.

OpenCV

The Open Source Computer Vision (OpenCV) library, a C++ library with a Python binding through the cv2 module, is a valuable tool utilized in this study. It facilitates feature recognition, camera calibration, image processing, and various other functionalities. Specifically, OpenCV was utilized to compare the as-designed image with the as-built image of the last layer, enabling accurate assessments and verification. The functions used were *absdiff* and *bitwise_not*, the first does a pixel-to-pixel comparison while the second one flips white and black pixels to improve visualization.

Chapter 5

Implementation

This chapter provides a detailed overview of the research implementation process, focusing on three main sections: design, post-processing, and structure.

The general approach will be to perform a scan of the specimen, post-process the as-built data, afterwards, the as-designed data is read from the IFC file, and finally the deviations are calculated.

5.1 Structure

The code is designed with a focus on Object Oriented Programming (OOP), incorporating five essential classes. The first class effectively manages all the coordinate information of the PC and takes charge of reading the source file. The second class manages the information of the designed path, including the layer information, desired height, and scale. The third class is dedicated to providing various utilities for processing the PC data. Lastly, the remaining two classes are tailored to specific tasks based on the scanned source, each serving its unique purpose.

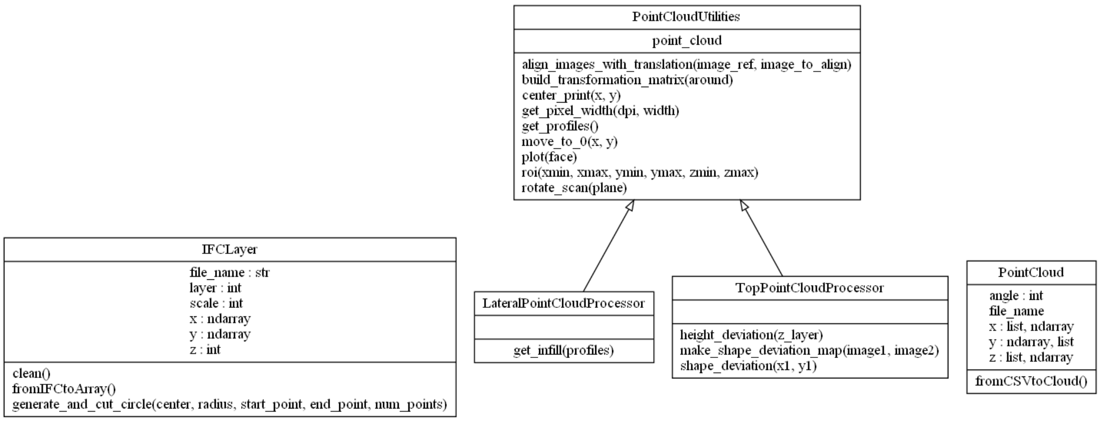


Figure 5.1: Code structure in UML diagram

The class *PointCloud* stores the most up-to-date information on the point cloud. It is initialized with the raw scanned data that is read from a CSV file Table 5.1 and stored in three individual arrays, one for each coordinate system. Here, only the valid information is stored, and all three vectors have the same size. The *IFCLayer* class extracts the information from the IFC file and stores the path as coordinates.

The third class, *PointCloudUtilities*, serves as a comprehensive collection of functions responsible for updating the point cloud data, generating plots, and performing data classification, among other tasks. Organized as a set of modular functions, it efficiently utilizes and updates the x, y, and z coordinates from PointCloud as required. As discussed in this chapter, post-processing of the point cloud data involves filtering, translation, rotation, and other essential operations.

The *PointCloudUtilities* class plays a crucial role in maintaining clean and structured code. It serves as the central hub for development and is called upon based on the scanned data source from either *LateralPointCloudProcessor* or *TopPointCloudProcessor* class.

5.2 Information from the Digital Design

The as-designed data is in an IFC file with a great level of detail originating from a BIM model as explained in section 2.2. For the sake of this study, only two pieces of information are required. First, the layer height, this information is in the IFC, but as it remains constant, it is given as an input. The second is the positional information, stored in the IFC file as curve geometries. Such geometries should be similar to that in Figure 5.2 with noticeable variations depending on the component's geometry.

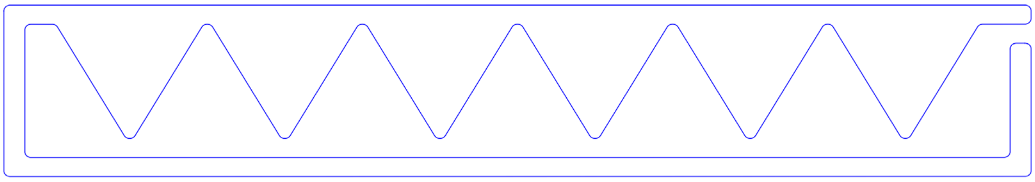


Figure 5.2: Layer information from design

One layer is selected for this because the IFC file contains a specific number of layers depending on the component's height. In the scope of this thesis, only simple examples are evaluated, they are designed uniformly in the z-direction, with no overhang or geometry changes, and one layer has all the information as it is replicated with an additional offset, as seen in Figure 5.3, nevertheless, the implementation allows to extract any other layer information at any time. The information of the selected layer is then extracted and discretized into x, y, and z coordinates. These coordinates are then stored in the *IFCLayer* class.

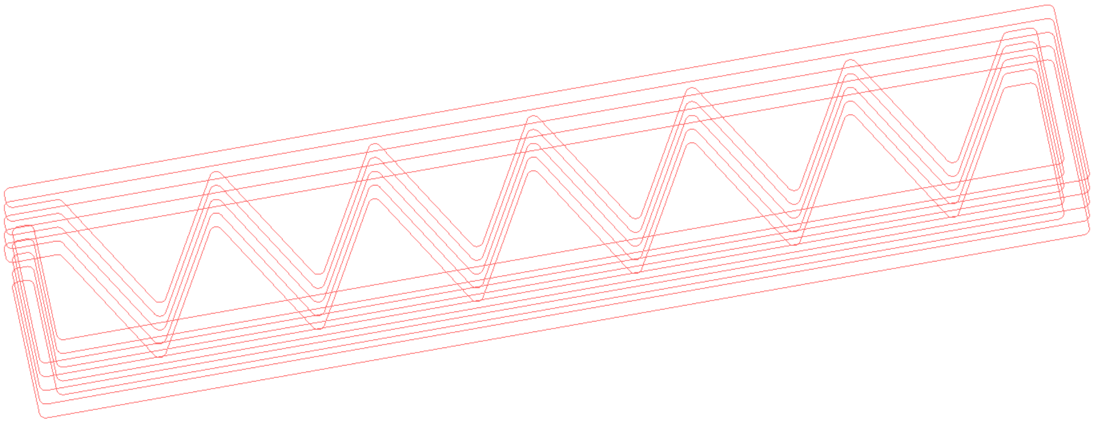


Figure 5.3: Multilayer information

The IFC data is read, and the geometry is extracted. Then the width and height to complete all the input parameters to store in the *IFCLayer*.

As illustrated in Figure 5.2 each layer contains straight lines called *IfcPolyline* and curves called *IfcTrimmedCurve*. The *IfcPolyline* consists of two points, the start, and the end point while the *IfcTrimmedCurve* consists of an *IfcCircle* with a center and a radius and two *IfcTrimmingSelect*, which are represented by *IfcCartesianPoint* for trimming the circle.

Figure 5.4 illustrates the designed layer with the direction of the extrusion. The start is the red circle, and the end is the red square. A layer transition is extruded between layers from the end of one layer to the start of the next. The start and end point can be selected during the planning process as it was implemented by the RTDE module described in section 4.2 that controls the printing process.

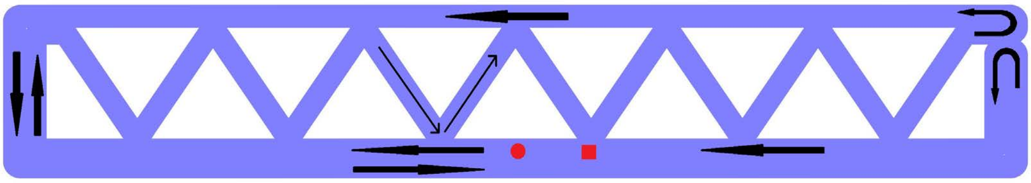


Figure 5.4: Direction and designed layer

Algorithm 5.1 shows the partial implementation to extract the path information from the *IfcTrimmedCurve*. Here the challenge was to properly extract the information to make sure the path planned in the IFC is the same as the as-designed generated. For that, it was identified that all the curves were partial circles and, specifically, all smaller than half a circle, line 9 and line 17 check for the smallest circle while line 5 checks for the correct direction of the circle.

Figure 5.5 shows and example of a circle, it has the center (c) and two additional points (A), (B). The shorter colored region represents the desired arc while the longer black region represents the rest of the circle. The full circle is discretized initially with *num_points* points and then cut to keep the short region with sufficient points to maintain the resolution. The points are organized clockwise or counterclockwise depending on the respective direction of the previous and following points to the arc.

Both for the straight segments and for the curve, the x and y coordinates are stored in the *IFCLayer* class as an *numpy.array* along with the scale information, the layer information, and the current z position. That z position is not currently used, as both the scanner and extruder have different coordinate systems.

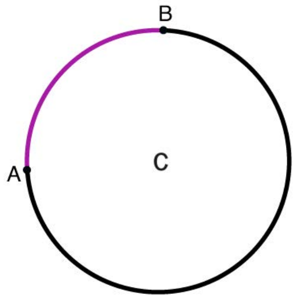


Figure 5.5: Generate and cut circle

Algorithm 5.1: Generate and cut circle

```
def generate_and_cut_circle(self, center, radius, start_point, end_point, num_points=100):
    -> Create a circle using center and angle ( sin, cos) and store in coordinates with num_points
    -> Find the index of the start and end trimming coordinates (start_idx and end_idx)

    if end_idx < start_idx:
        cut_points3 = coordinates[end_idx + 1:start_idx]
        cut_points1 = cut_points3[::-1]
        cut_points2 = np.concatenate((coordinates[start_idx:], coordinates[:end_idx + 1]))
        if len(cut_points1) > len(cut_points2):
            cut_points = cut_points2
        else:
            cut_points = cut_points1
    else:
        cut_points1 = coordinates[start_idx:end_idx + 1]
        cut_points3 = np.concatenate((coordinates[end_idx + 1:], coordinates[:start_idx]))
        cut_points2 = cut_points3[::-1]
        if len(cut_points1) > len(cut_points2):
            cut_points = cut_points2
        else:
            cut_points = cut_points1
    ...
```

5.3 Post-processing

The data is first organized to allow its use for the different analyses done. For this, it is necessary to first reformat the data into arrays, and then filter out undesired data points. For post-processing, a series of functions have been implemented. Some are general functions used during the top and lateral scans, and others are more specific.

The function *fromCSVtoCloud* in the *PointCloud* class holds significant importance as it reads the PC CSV data exported from the Gocator. This task proved to be challenging due to the data storage format from Gocator, as highlighted in Table 5.1. The unique-Z problem, which is explained in section 3.8, becomes evident when observing the example of how the data is stored. For a specific X_i and Y_j, only one Z value can be stored, resulting in empty locations where Z measurements are missing. The consecutive X and Y values with a fixed spacing in each direction contribute to this peculiar data storage pattern.

Table 5.1: Gocator output CSV format

X/Y	X$_1$	X$_2$	X$_3$	…	X$_n$
Y$_1$	Z	Z	Z		Z
Y$_2$	Z	Z	Z		Z
Y$_3$	Z	Z	Z		Z
⋮					
Y$_m$	Z	Z	Z		Z

To properly store the above data as regular (x,y,z) coordinates, a loop is done within one row/profile to find the valid height values for each column and then pass them to the following row. This is seen in Algorithm 5.2 where a double loop is implemented to iterate over all rows and columns, followed by two data integrity checks to eliminate empty or mistaken data. Finally, each coordinate is appended to independent (`x, y, z`) arrays.

Algorithm 5.2: Function extract to store PC into arrays from a CSV file

```
def fromCSVtoCloud(self):
    ...
    cloud = np.genfromtxt(filename, delimiter=',', skip_header=counter,
        skip_footer=12)
    rows, columns = cloud.shape
    x = []
    y = []
    z = []
    for row in range(1, rows):
        for column in range(1, columns):
            if ~np.isnan(cloud[row, column]):
                if cloud[row, column] > -np.inf:
                    x.append(cloud[row, 0])
                    y.append(cloud[0, column])
                    z.append(cloud[row, column])
    ...
```

In that sense, all the information is stored in consecutive points for each profile scan. This information structure is more beneficial as it has no uniqueness or empty values, but it stores information less efficiently. The fact that the points are stored in a consecutive x was beneficial for the infill calculations in subsection 5.3.2 as it does profile integration, and once the x value changes, the new profile starts. This benefit is only valid when the relative movement between the object and the camera is along the x-axis. As described in Figure 1, that is the case for this study.

Region of Interest (ROI)

The function *roi* in the *PointCloudUtilities* class below is used to trim the point cloud into a region of interest. This is used to eliminate outliers based on their position. Here, essentially what is done is to pass the limits on all three axes and keep only the values encapsulated within the limits. The usefulness is evident as the laser scanner records data outside the required scope, and it is necessary to filter it out for two reasons. First, handling fewer data makes the processing faster; second, and more importantly, only storing valuable information and doing calculations on an already filtered cloud. Figure 5.6 show the lateral scan from an end and Figure 5.7 shows the same scan but with the data trimmed.

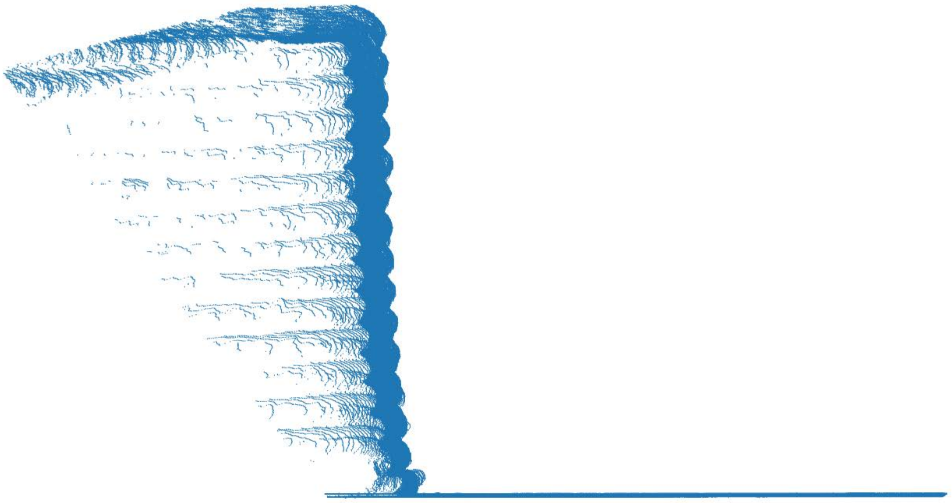


Figure 5.6: Lateral scan

Figure 5.7: Lateral scan trimmed

One of the main challenges when post-processing the data is to filter it. A very aggressive filter will lead to important information being lost, while a very lenient filter will imply some noise or undesired data will not be removed. In both cases, the results are affected as it includes fake data. Due to the high resolution of the sensor, no significant noise is expected, only scanned information at undesired locations. That led to a position-based filter.

A rectangular prism is a position-based filter, this prism is illustrated in Figure 5.8. It is clear how data points are eliminated based solely on their position. This prism has been manually positioned for all of its use cases in this study.

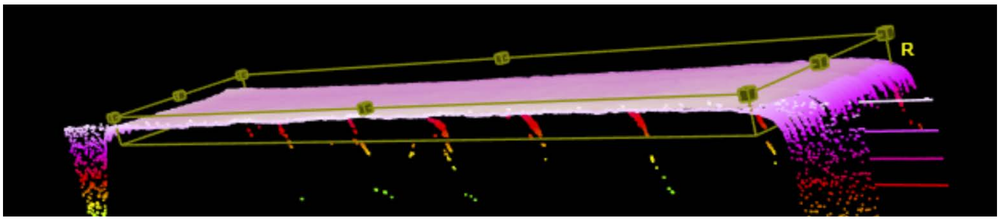


Figure 5.8: ROI illustration

Algorithm 5.3 shows an extract from the ROI function. This algorithm has been implemented for the case where the scan data and the scanner are aligned on one axis as in this study, described in Figure 1. Here it is possible to filter with 1 to 6 edges of the prism as observed in line 2 where in the case an edge is not provided for the filter, the maximum or minimum is used respectively to avoid modifying the data set for that region. Then, a loop is introduced in line 4 where an iteration is done for every point in the cloud to filter based on the required restrictions generating the 6 planes of the rectangular prism from Figure 5.8.

Algorithm 5.3: Region of Interest

```
1  def roi(self, xmin=None, xmax=None, ymin=None, ymax=None, zmin=None,
       zmax=None):
2      if None => min and max calculated from vectors
3      ...
4      for i in range(len(x)):
5          if xmax > x[i] > xmin:
6              if ymax > y[i] > ymin:
7                  if zmax > z[i] > zmin:
8                      xnew.append(x[i])
9                      ynew.append(y[i])
10                     znew.append(z[i])
11     ...
```

5.3.1 *TopPointCloudProcessor* class

As mentioned in section 3.4, the data from the top scan is used for two different calculations, the shape deviation and the height deviation. These two use the same input file, but one focuses on the (x,y) of each point while the other focuses on the z coordinate of the point and uses the (x,y) only to position it. Figure 5.9 shows a picture from above of a print. This is useful to understand the two comparisons that have been done.

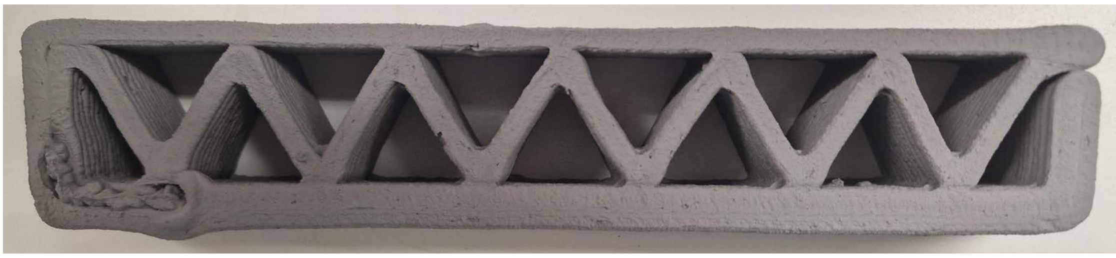


Figure 5.9: Top view

height_deviation **function**

Knowing what layer or height the last layer should be at is essential to perform the height deviation correctly. This is easily calculated from the number of printed layers and the layer height[1]. A ROI is then selected with this information to avoid measuring lower layers with the function described in Algorithm 5.3. The minimum height should be no more than one layer below the scanned one. Then, the current height is subtracted from every point in the point cloud as seen in line 3 in Algorithm 5.4, ensuring that the 0 level is the expected value, anything below has a negative value equivalent to the missing space, and the same to anything positive is additional material.

Algorithm 5.4: Height deviation extract

```
1 def height_deviation(self, z_layer):
2     ...
3     z -= z_layer
4     ..
5     plt.scatter(x, y, c=z, cmap='RdBu', vmin=-max(abs(min_z), abs(max_z
          )), vmax=max(abs(min_z), abs(max_z)))
6     ...
```

The results are in the form of a heat map represented as a matrix. The heatmap shows the deviation in two different scales. A cold scale has the over extrusion while the under extrusion is observed on a warm scale observed in line 5. The scale is centered at 0, so it is not necessarily equidistant. The matrix with all the information is stored in a file as it is useful for the user and in the future to store it in the IFC file layer.

shape_deviation **function**

The shape deviation consists of two images. The as-designed is done primarily with the IFC file information and a filament width calculation for the plotting. The as-built is from the data captured from the laser scanning.

[1] For this study the scan was done parallel to the top surface and it is a plant print. If the print is not planar or the scan is not done parallel to the surface, the calculations will increase in complexity.

The first step and primary source of potential error is aligning these two data sets[2]. The data is positioned by finding the geometric center. This is calculated as the middle in the x range and the same for y as observed in line 7 and line 8 in Algorithm 5.5 respectively. Then, in line 10 and line 11 the data is translated in x and y respectively.

Algorithm 5.5: Move print for alignment

```
1  def center_print(self, x=None, y=None):
2      ...
3      x_max = max(x_new)
4      y_max = max(y_new)
5      x_min = min(x_new)
6      y_min = min(y_new)
7      x_mean = (x_max + x_min) / 2
8      y_mean = (y_max + y_min) / 2
9      ...
10     x -= x_mean
11     y -= y_mean
```

This is trivial for the as-designed data as the information is perfect, on the other hand, for the measured data, if there is a significant value outside of the expected range, it causes the center to be shifted towards the error, generating alignment problems. Figure 5.10 shows how this problem is simplified where in red are the limits of the print and the circle is a printing error, as observed this is only a problem with over-extrusion. The green lines indicate the expected center, while the blue lines show the center due to the error.

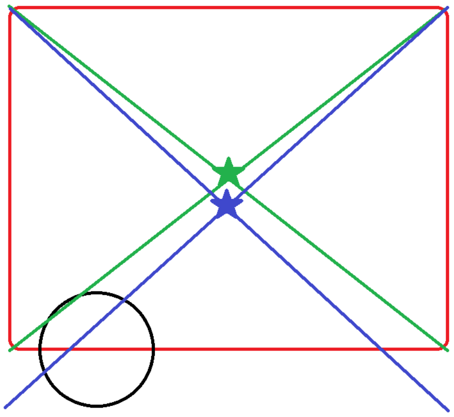


Figure 5.10: Center shift due to error

Other translations were tested like moving the prints to the minimum x and y, which is done by modifying line 10 and line 11 with the minimum x and y respectively. This method for alignment would not be suitable for a dry print as the top layer has a significant shrinkage and that method would show the shrinkage towards one side only.

Currently, the source of the error has been identified, but no practical solutions have been implemented to address it. Several potential approaches have been considered, including

[2] A better way would be to use the process data of the print, but this is, at the moment of writing the thesis, not available.

the use of a reference point, where a known coordinate is extruded for accurate positioning of the scan. OpenCV alignments were also considered but didn't prove effective either. Another option is a multi-robot calibration to align both robots to the same coordinate system. However, this solution is more complex, as the RTDE implementation does not have feedback for the processed data available at the moment of this study.

The laser scanner currently places the scan data in local coordinates. Nevertheless, since the starting point is known, it may be possible to translate all subsequent scan points with respect to that point to obtain coordinates in the global system. Despite these potential solutions, none have been thoroughly tested or implemented at this stage. Further examination and validation are required to determine the most effective approach.

The current implementation has the limitation that it requires information at the edges. For example, Figure 5.11 shows a layer that was stopped slightly after it started, it has very limited information, and the alignment does not represent the expected positions.

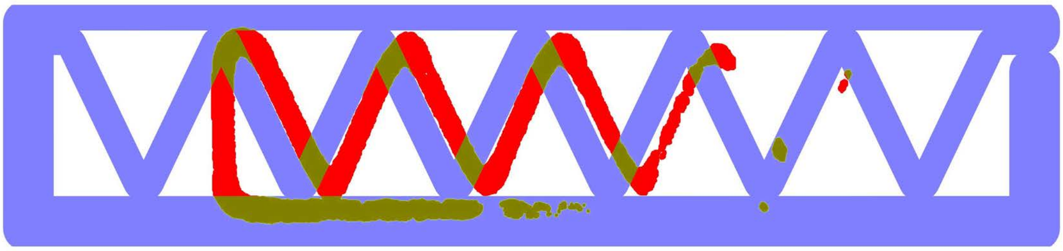


Figure 5.11: Center alignment with limited data

As-built shape

A ROI is created and includes information on the last two layers. This filters information by depth, removing data points based on their z value. This is a straightforward filter but a very effective one. The shape deviation is a shape comparison, so the z value is unimportant. Nevertheless, only the shape of the last layer is essential for the comparison. If the data records interior points from the previous layer due to the perspective distortion, it would seem like a shape variation when placed in 2D. Also, any error in prior layers could be interpreted as errors in the shape of the current layer. It is crucial not to over-filter by removing too many layers, as this could filter out edges or parts from the current layer. After that height filter, the Z coordinates are no longer considered, as the study is exclusively in two dimensions.

As-designed shape

As mentioned in section 3.2, the as-designed shape is extracted directly from the IFC file for the given layer, it consists of one layer of information as illustrated in Figure 3.7. This information is then plotted with a specific line width to simulate a printed filament. Here only the shape matters, so no information about the start or end points is required. The only nontrivial part for the as-designed shape would be the line width. The line width has to approximate the filament as it is used for a direct comparison. To do so, the dpi is used to calculate the most appropriate line width for the plot. It needs to be a dynamic number that depends on the plot and print size, as the line width relies on the plot size.

The calculation is as observed in Algorithm 5.6. The aspect ratio referred to in line 4 is used to set the height and width of the plot and is used for all plots. Finally, the layer is modified into the one observed in Figure 3.8

Algorithm 5.6: line width calculations

```
1       ...
2       width_in = width / 25.4
3       width_pixels = width_in * dpi
4       line_width = width_pixel * aspect_ratio
5       ...
```

Shape comparison

To make the comparison, two functions are implemented *shape_deviation* and *make_shape _deviation_map*. One is quantitative, and the second one is qualitative. The as-designed line plot is placed on top of the as-built scatter plot with the center alignment method for the quantitative function. They are plotted with different colors, and the as-designed is set with some transparency to observe the two properly. Here, the result is a 2D plot with three colors as seen in Figure 5.13. The light blue is the designed shape, the red is the built shape, and the overlap is in dark red.

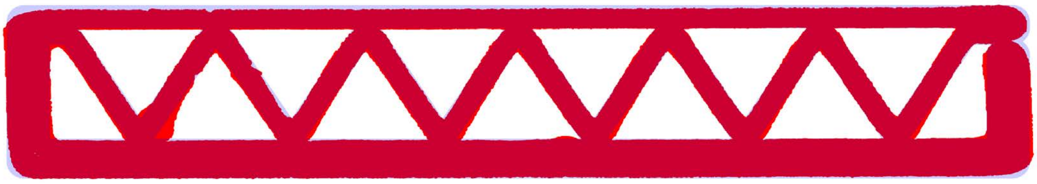


Figure 5.12: Combined shape deviation plot

The second function uses the same idea of overlapping the information, but in this instance, it uses an image of the line and scatter plot. These two images are overlaid and compared using OpenCV. This is important as the library allows a high level of manipulation and analysis of the individual images and the resultant one. In this case, also three colors are the output as observed in Figure 5.13.

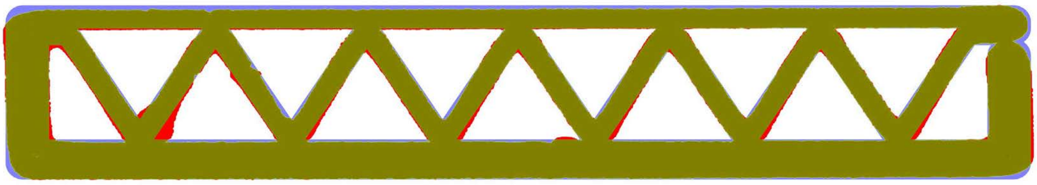


Figure 5.13: 3 color shape map

For both methods, the as-designed color represents the under-extrusion, the as-built color represents the over-extrusion, and the overlap represents the material in the correct position.

5.3.2 Lateral scan

For the lateral scan, the sensor is placed from the table at around 60° for one side and -60° for the other as mentioned in section 3.1. This avoids collisions with the table while scanning despite complicating post-processing. The complications are divided into four sections; first is the accumulation of errors due to the transformation matrix. For this reason, all the analysis is done at an angle and in local coordinates. The second and most relevant problem is the unwanted data measured during the scan. Third, the loss of information due to the limited field of vision. The final problem was associated with the visualization part, as the view is complicated to understand due to the angle, and it is discussed further on.

Figure 5.14 shows the raw data after it has been scanned. It can be observed that a significant amount of data is from the table. Additionally, enclosed in red, plenty of undesired data is visible in the raw data. The undesired data includes data from the inner structure, sides, and table. All this information has to be filtered out by the ROI and adjusted to keep only the desired information.

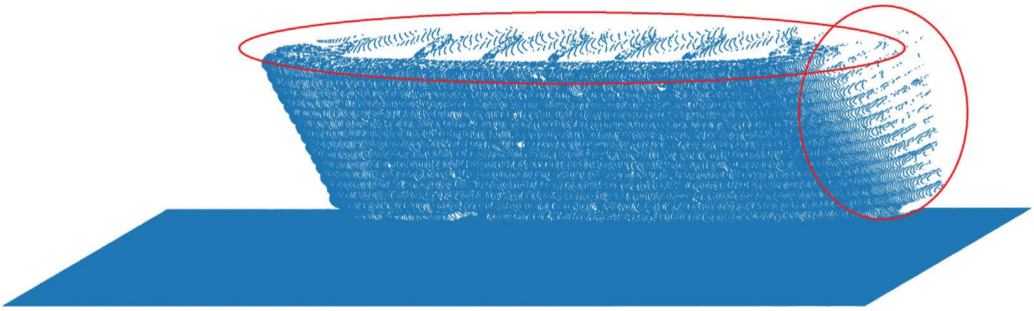


Figure 5.14: Lateral scan raw data

Problem 1: Transformation Matrix

Even though only one transformation matrix is needed since the scanning angle only changes the yz-plane, a function with all three transformation matrices was implemented. This can be useful if the table is not flat or if the scanning process is not parallel to the surface. Below, all three matrices are shown, and it is clear how each matrix only rotates over one axis as observed in Figure 5.15. This makes it more flexible, as an arbitrary rotation can be represented by the multiplication of the matrices of its Euler angles.

$$\boldsymbol{R_x}(\theta\,) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{R_y}(\theta\,) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 0 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{R_z}(\theta\,) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

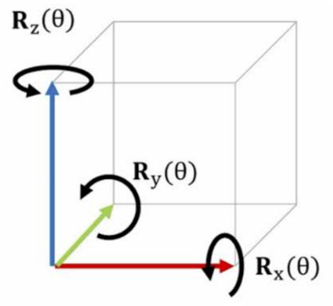


Figure 5.15: Rotation matrix and illustration

Problem 2: Unwanted data and filtering

The unwanted information is much higher when doing a scan at an angle than when doing it perpendicular to the specimen. This includes the printing platform and information on the top layer and edges of the previous layer. It contains information that can not just be filtered out with a plane, as before. A rectangular prism (ROI) is required to eliminate information based on the position as explained in section 5.3-ROI.

The rectangular prism operation is applied to all the data. The decision to use filters was influenced by the data being already in the form of numerical arrays rather than the conventional PC format, where utilizing various specialized PC libraries would be possible. Additionally, it was done to eliminate the need for segmentation as the lateral scan includes the curvature that connects to the other edges of the component.

Problem 3: Data loss

After the information is filtered, some data is inevitably lost. This should not be a big problem as the density is so high some points do not generate global issues. On the other hand, as previously mentioned, the angle causes occlusions in the field of view. The laser scanner only records what it can see, and some data is not measured when the sensor is placed at an angle. Figure 5.16 clearly shows in red how some data is lost, it is due to the limited field of vision of the sensor.

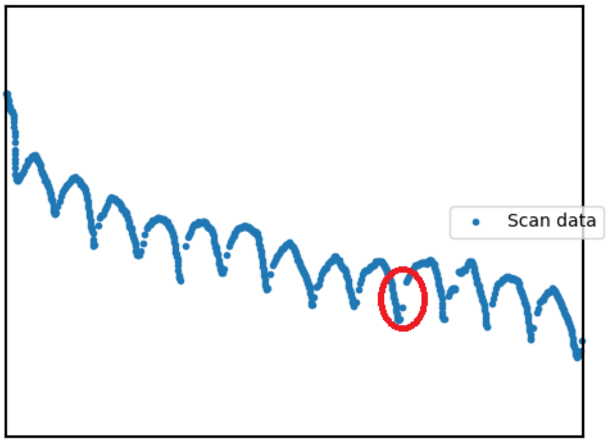


Figure 5.16: Scan profile with missing data

Integration

Algorithm 5.7 shows how the integration is performed. First, the upper limit is found as the idea is not to remove material. The infill has to cover the whole wall to the point of highest extrusion. All the data is moved across the 0-line until that critical point is 2mm across, as explained in subsection 3.4.3. This ensures that everything has at least a 2mm infill to the flat wall, as illustrated in the difference between Figure 3.11 and Figure 3.12. The 2mm is an arbitrary number used for this study.

In line 2, the surface data is split into the scanned profiles data. This is done as each profile data has the same x-coordinate. This is a property of the selected sensor, as previously explained, the sensor takes profile scans and generates a surface from the movement, nevertheless, in this case, it is more beneficial to work with individual profiles.

Algorithm 5.7: Lateral infill calculation of missing volume

```
1 -> move all profiles across the 0-depth line
2 -> surface into profiles
3 -> iterate over each profile
4     -> linear interpolation
5     -> trapezoidal integration for all interpolation functions
6 -> average infill and other statistical data
```

The individual profiles are found using *np.diff* where discrete differences are found along a given axis. In this case is along the x-axis as seen in line 2 of Algorithm 5.8. This was possible because the scanner has an x value for each profile, and the y and z values are measured. Hence, one profile has the same x-value for all of its points. Finally, as seen in Algorithm 5.8, the indices for each profile are identified and stored for future use.

Algorithm 5.8: Profiles index calculation from surface scan

```
def get_profiles(self):
    diff_x = np.diff(self.point_cloud.x)
    profile_indices = np.where(diff_x != 0)[0] + 1
    profile_indices = np.insert(profile_indices, 0, 0)
    profile_indices = np.append(profile_indices, len(self.point_cloud.x
        ) - 1)
    return profile_indices
```

Then, for each profile, a linear interpolation is performed between 2 adjacent points in line 4 of Algorithm 5.7, displayed in Figure 5.17. The interpolation helps overcome the limited field of view that was previously mentioned, as this mostly happens in the layer boundaries where the printed information can be assumed as a straight line as it doesn't involve a significant infill area.

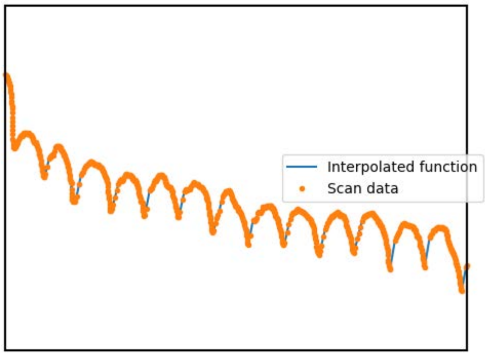


Figure 5.17: Linear interpolation of one scan profile

A trapezoidal integration is done for every profile, as seen in Figure 3.14. As it is linear between adjacent points, the numerical integration method selected does not incur additional errors. The integration area may be observed in Figure 5.18 and Figure 5.19 for opposite sides of the printed object in gray.

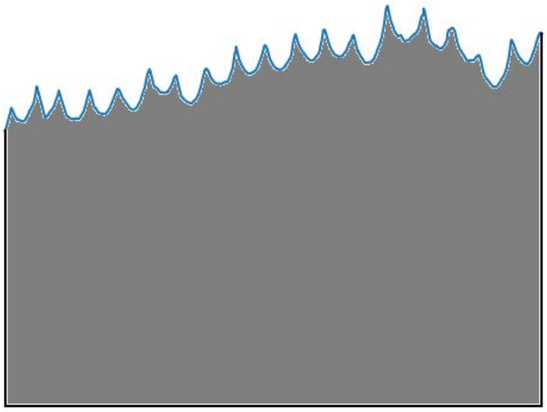


Figure 5.18: Profile integration for side 1

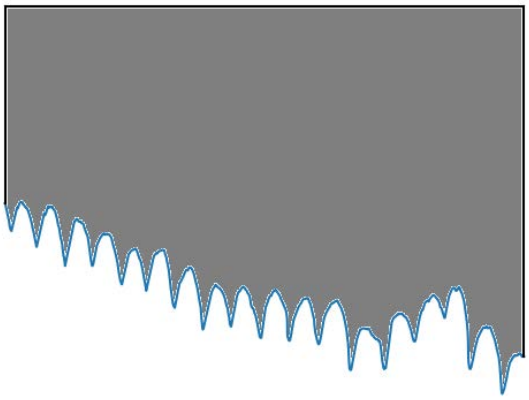


Figure 5.19: Profile integration for side 2

Problem 4: Visualization

All visualization is done in 2D. This is to make it faster and simpler to observe. It is implemented so that it can be observed from the top, the side, or the front. Additionally, rotations can be done with the previously explained function to obtain a view at an angle. The deviation maps are very useful as they represent the PC data. This means that the maps of all the sides are an effective way of storing the deviation data.

Additionally, it is crucial for the debugging process and for adequately understanding the results. For such reason, it is done with a color scale when looked at from the front to have a better dimensional understanding as observed in Figure 5.20.

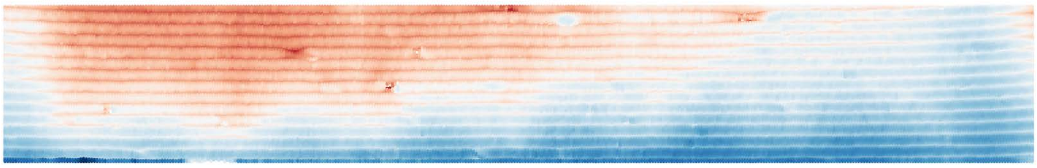
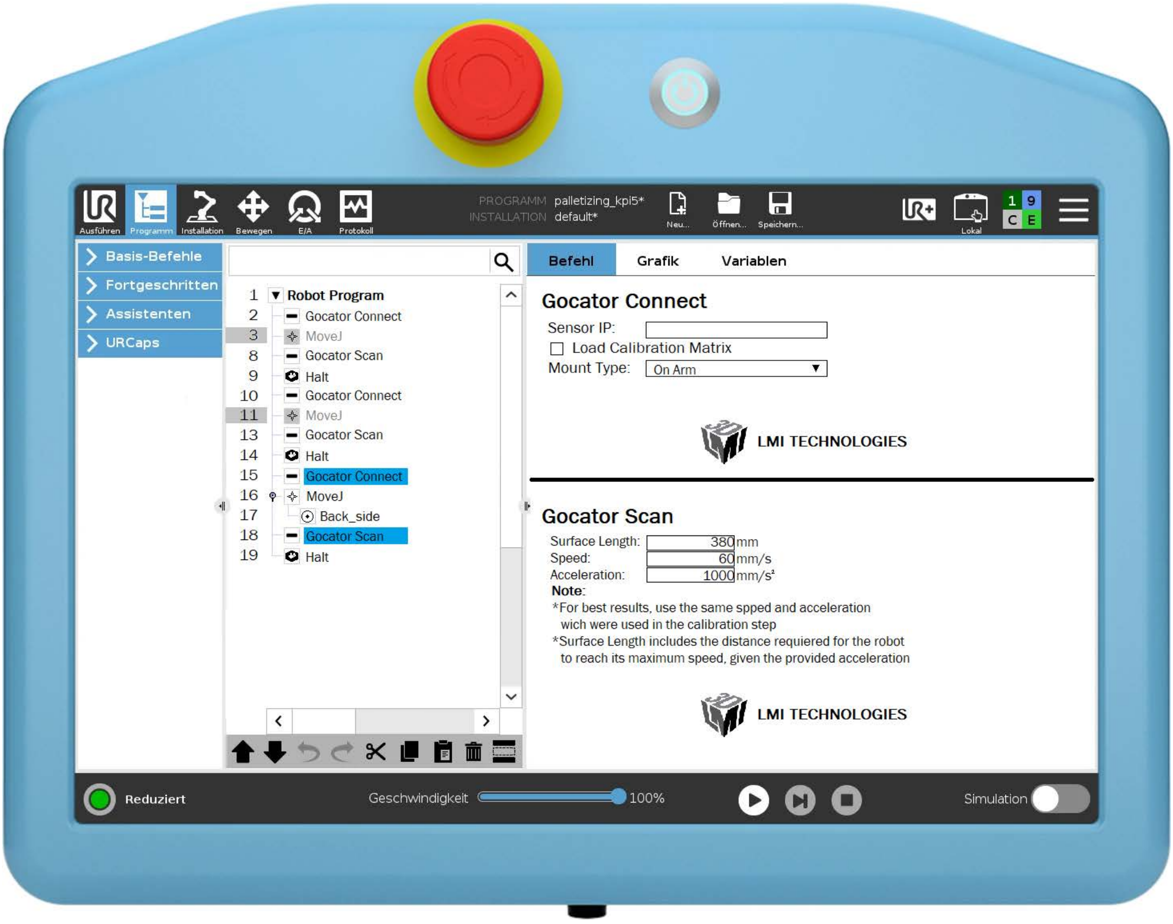


Figure 5.20: Colored front view

5.4 Scanning with URCaps

A concise program is developed using teach pendant programming during the scanning process. This program leverages functions from the Gocator URCap, which are illustrated in Figure 5.21. Initially, the program establishes a connection to the Gocator by utilizing the IP address. Subsequently, the scanning process occurs at a specified speed and covers a defined distance. To ensure easy scan replication, waypoints are integrated within the *MoveJ* functions, capturing the starting position. This feature proves highly valuable in achieving replicability.

Three sets of scans are observed, each commencing with a *Gocator Connect* and concluding with a *Halt*. This is due to the variation in initial positions. The first set is designated for the top scans and comprises four concealed waypoints, each possessing a distinct Z value. These waypoints facilitate the scanning of different layers during the printing process. The remaining two sets pertain to each side respectively.



PROGRAMM palletizing_kpi5*
INSTALLATION default*

```
1  ▼ Robot Program
2      Gocator Connect
3      MoveJ
8      Gocator Scan
9      Halt
10     Gocator Connect
11     MoveJ
13     Gocator Scan
14     Halt
15     Gocator Connect
16     MoveJ
17       Back_side
18     Gocator Scan
19     Halt
```

Gocator Connect

Sensor IP:

□ Load Calibration Matrix

Mount Type: On Arm ▼

Gocator Scan

Surface Length: 380 mm
Speed: 60 mm/s
Acceleration: 1000 mm/s²

Note:
*For best results, use the same spped and acceleration wich were used in the calibration step
*Surface Length includes the distance requiered for the robot to reach its maximum speed, given the provided acceleration

Figure 5.21: UR pendant programming

Chapter 6

Results

This chapter will review the results gathered from the top and the lateral scans. It will describe how the implementation has behaved in fresh and dry states and how all the valuable information is stored. Additionally, the effects of the limitations discussed in previous chapters will be discussed.

The following results have been chosen for discussion due to their similarity in information to the rest of the results. It's worth noting that a complete set of results can be found in Appendix A for further reference.

6.1 Raw data

The initial scan was conducted after completing 14 layers, with the unprocessed scan depicted in Figure 6.1. This image distinctly highlights two primary areas of interest clearly visible to the naked eye. The first focal point is located at the upper part of the image, where there is a noticeable tear in the extruded filament due to the presence of an air bubble in the mixture. The second point of interest is the layer transition area. The effect of a premature stop and an inappropriate retreat mechanism is observed. The early stop meant the layer was not finished, causing the lack of material, while the retreat mechanism showed excess material as the nozzle moved away with a vertical movement.

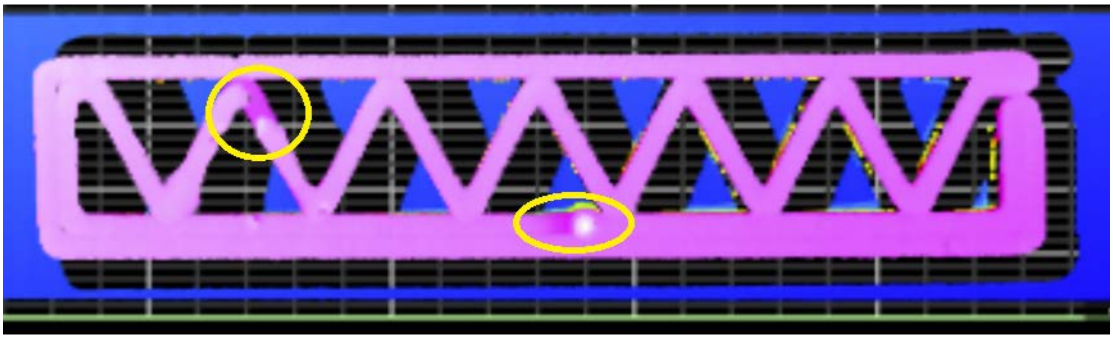


Figure 6.1: Top scan points of interest

In Figure 6.2 the dry scan data from Figure 6.1 is presented from a lateral perspective. Here, it becomes evident that the shrinkage process is not uniform across the entire object. Specifically, the center of the print experiences a more pronounced reduction in height than the edges.

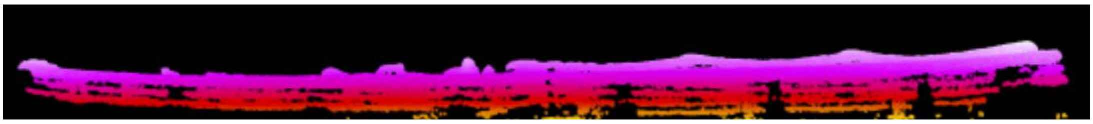


Figure 6.2: Top scan curvature

In Figure 6.3, it may be observed that there is also a critical, significant point among all the undesired data. That over-extrusion, more clearly depicted in Figure 6.4, will govern the outer wall post-processing as it is the furthest point outwards from the wall.

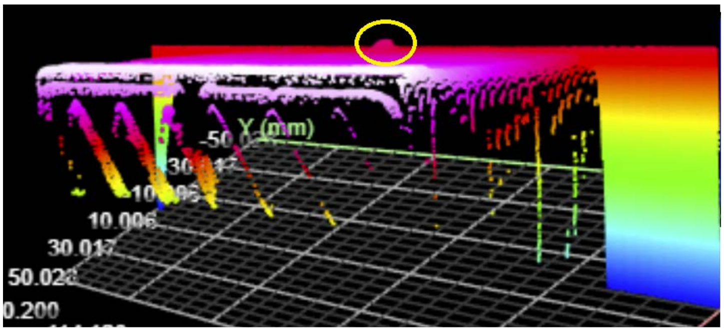


Figure 6.3: Lateral scan raw data

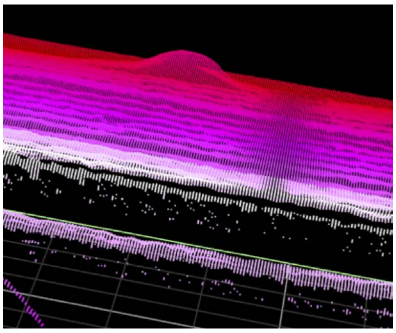


Figure 6.4: Close up to critical lateral point

6.2 Top Scan

Five results have been taken from the top of the benchmark sample. Four of these were obtained while the printing process was ongoing in a fresh state, whereas the remaining results were acquired once the sample had transitioned to a dry state. In this context, a detailed exploration of the outcomes pertaining to the final layer and an intermediary level is conducted and presented. A visual representation of the completed structure is visually represented in Figure 6.5. This illustration highlights the absence of a layer transition and a noticeable limitation in layer extrusion. Furthermore, Figure 6.6 provides a view of the air bubble initially depicted in Figure 6.1. Lastly, Figure 6.7 showcases the internal structure that lacks contact with the outer layer during an intermediate layer that was not scanned.

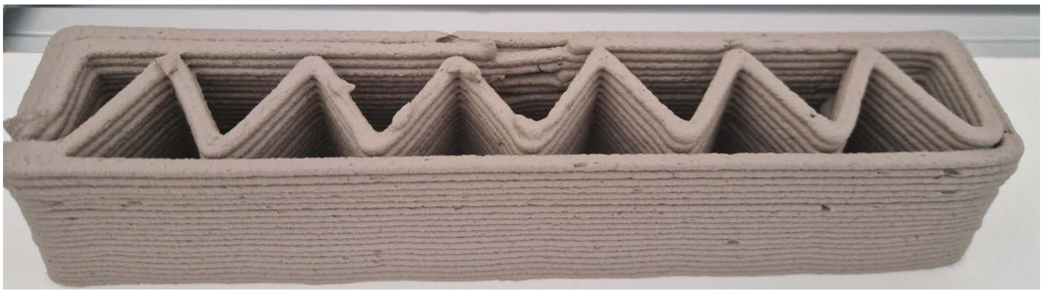


Figure 6.5: Last layer state

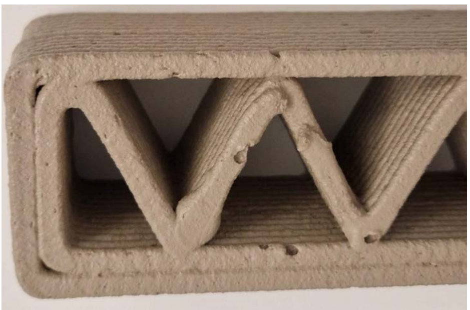


Figure 6.6: Intermediate layer with air bubble

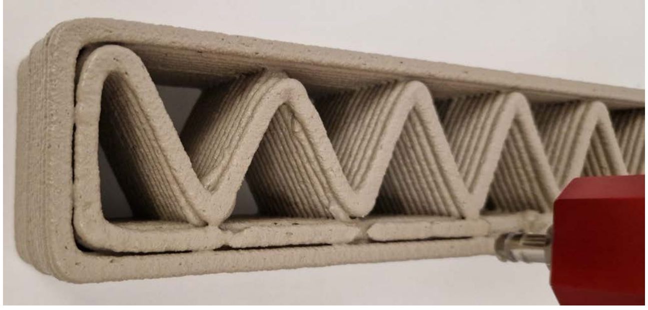


Figure 6.7: Intermediate layer with no internal contact

As explained in subsection 3.4.1, Figure 3.8 shows the designed and expected shape for every given layer. This is used as the baseline for the qualitative comparison presented here.

6.2.1 Layer 14

Figure 6.8 shows the scan data for layer 14, the as-built information for the expected layer height with an additional 3mm tolerance. It is observed that the layer shape is as expected, with some minor defects. For a full comparison, such scan was compared with the as-designed layer in Figure 6.9 for the shape deviation and in a height deviation map in Figure 6.10. The raw data is observed in Figure 6.1.

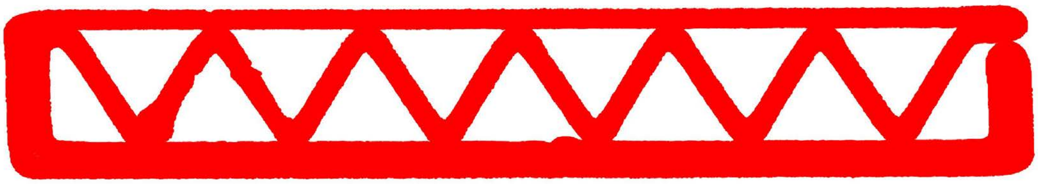


Figure 6.8: Layer 14 as-built scan fresh

It's evident in Figure 6.9 that in this specific layer, the occurrence of over-extrusion (red) is notably low, whereas under-extrusion (blue) is more pronounced, particularly within the corners and the top edge. This instance perfectly exemplifies a well-executed print, as the intended shape has been successfully realized in the built object. However, it's crucial to analyze the height deviation map for further insights, as illustrated in Figure 6.10. This step is essential because it enables us to uncover additional details that are visible from Figure 6.1, such as the air bubble mentioned and other layer imperfections.

Figure 6.9 fails to depict the tear in the extruded filament caused by the presence of an air bubble. This omission is attributable to the fact that a preceding layer was identified as the filter's aggressiveness couldn't be increased further, given that the lower right corner is part of the current layer but contains lower values than the air bubble region.

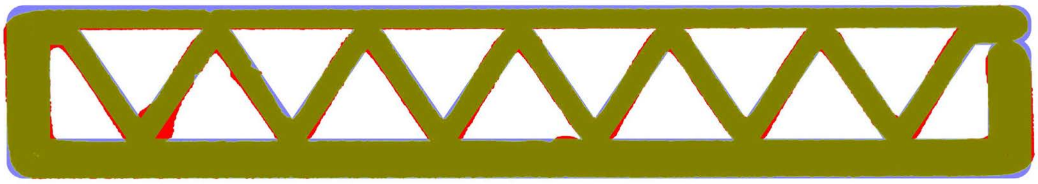


Figure 6.9: Layer 14 shape deviation map fresh

In Figure 6.10, the height deviation map for Layer 14 is depicted. This visualization employs a color scheme where blue indicates values surpassing the anticipated height, red signifies values falling short of the height, and a lighter shade indicates values aligning with the expected height. This illustration effectively conveys two significant pieces of information.

Firstly, it becomes evident that the measurements are not perfectly aligned with the printing process, resulting in an artificial tilt in the results. Specifically, on the right side, there is a predominance of lower heights, on the left side, heights generally exceed the expected values, while the central region closely adheres to the anticipated height. This behavior effectively justifies the suspected tilt angle.

Secondly, it highlights the presence of the air bubble, the filament retraction, and the premature layer stop. Additionally, it reveals the points where adjacent filaments come into contact, adding additional depth to the analysis.

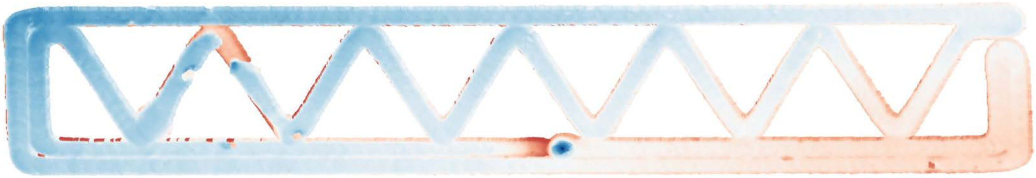


Figure 6.10: Layer 14 height deviation map fresh

Finally, Figure 6.9 and Figure 6.10 are stored as images to keep in a database that stores the as-built process information. The information is most valuable when observed together. For instance, while Figure 6.10 excels at illustrating various defects, it does not effectively represent the extent of over-extrusion, a quality better captured by Figure 6.9. Collectively, these images offer a more holistic understanding of the print's intricacies when observed together.

6.2.2 Layer 21 - Final layer

The examination of the top scan progresses to its second phase, focusing on the last layer – layer 21. This layer bears significant importance as both fresh and dry state comparisons are made. Moreover, being the final iteration, it lacks the subsequent layers that could potentially mask under extrusions or rectify defects.

For a thorough assessment of the last layer, it's essential to begin by analyzing the height deviation exhibited by the preceding layer, layer 20. Figure 6.11 highlights specific regions, namely the left side and the upper right corner, where values surpass the anticipated heights. The upper right corner presents a particularly intriguing scenario. Situated at the juncture where the transition from the outer to inner contour occurs, this corner experiences a significant drag force owing to the 180° turn. By Layer 20, a cleavage has already formed, creating an opening in the transition from the outer to the inner contour. The combination of drag force and the absence of side support results in the corner being pulled inward, causing it to elevate slightly. This preliminary insight sets the stage for the comprehensive analysis of the final layer's characteristics.

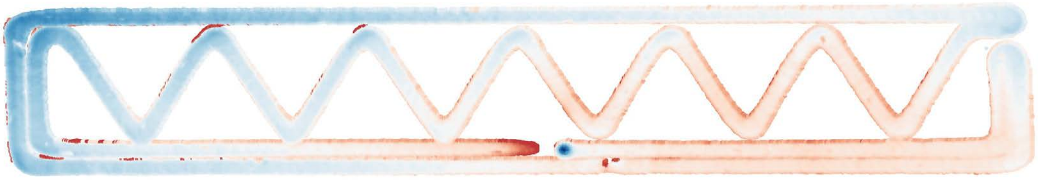


Figure 6.11: Layer 20 height deviation map fresh

Scan data

Both Figure 6.12 and Figure 6.13 provide visual representations of the uppermost layer in both its initial fresh state and its final dry state. While this layer wasn't fully extruded, it serves as an excellent example to demonstrate shape deviation and highlight the limitations posed by filtering techniques. Figure 6.12 presents an accurate depiction of the layer, aside from a few exceptions. Notably, the uppermost right point and the top and left filament sections highlighted in green deviate from the actual layer extrusion as seen in Figure 6.5. These deviations can be attributed to the filter employed. These points, although appearing as part of the current layer, are actually remnants of previous layers, as explained before with Figure 6.11.

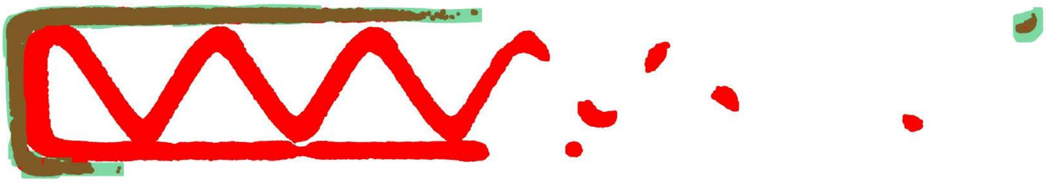


Figure 6.12: Final layer as-built scan fresh

As the clay dried, it shrank. Figure 6.2 shows that the shrinkage was not uniform and the sides tilted upwards. This explains why Figure 6.13 appears to have more extrusion on the right than before and less on the top.

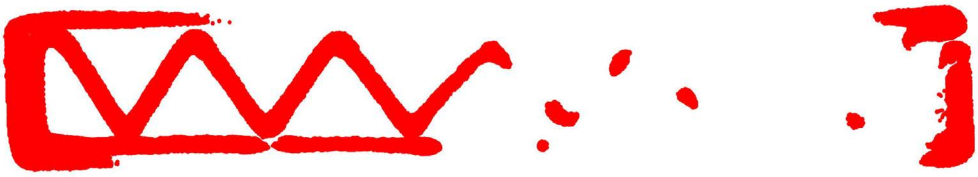


Figure 6.13: Final layer as-built scan dry

Shape deviation map

Figure 6.14 illustrates the shape deviation of the last layer when extruded. It clearly shows that more than half of the layer is missing. Additionally, it shows some extrusion in undesired locations. This is always towards the right while under extrusion towards the left. It would seem to be more a positioning problem than an over and under-extrusion. This makes sense as the implementation is not very robust and is unstable when the layer is not fully extruded, as it does not contain enough data to correctly estimate an appropriate alignment.

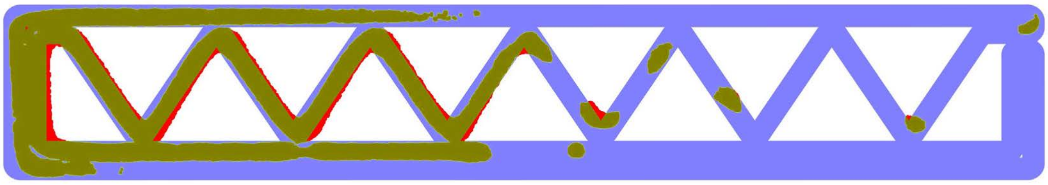


Figure 6.14: Final layer shape deviation map fresh

Once the print has dried, it is compared to the designed layer. It is evident in Figure 6.15 how the shrinkage happened on both sides and on the width. Here the alignment is also done to the center, resulting in a symmetrical shrinkage that is not necessarily true but was assumed. This is the reason why only the center part of the built structure is correctly aligned with the designed layer. This comparison effectively highlights the significant alterations that the as-built structure undergoes once it dries, emphasizing the noticeable deviations from the initially designed version.

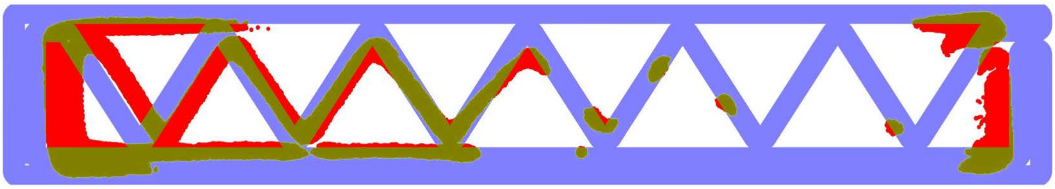


Figure 6.15: Final layer shape deviation map dry

Height deviation map

Once more, the height deviation is essential to complement the shape deviation. Figure 6.16 show on the left side in dark red the portion of the scan that is originally from the previous layer. It also shows in dark red the furthest point to the right. These particular sections posed a challenge for filtering, as applying the filter would have excluded the second-to-last point on the right side, which is indeed part of the genuine extrusion. This complex interplay between adjacent layers and the filtering process adds depth to the analysis and showcases the complexities of the data interpretation.

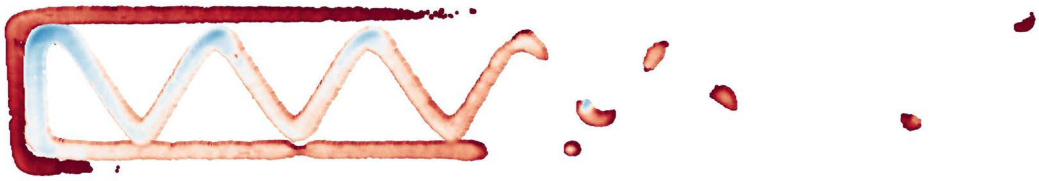


Figure 6.16: Final layer height deviation map fresh

Contrastingly, in the dried state, Figure 6.17, the left region now aligns more closely with the expected height. At the same time, the central portion of the layer appears to have dipped below the anticipated level. A new section has emerged on the right side, as described with Figure 6.13, and notably, it remains within the lower range of the tolerance spectrum.

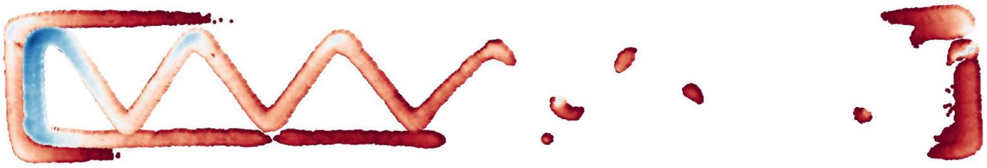


Figure 6.17: Final layer height deviation map dry

Moreover, the top filament that was present from the previous layer is gone. This absence is attributed to the non-uniform nature of the shrinkage process. These observations emphasize the complexity of the filtering process, particularly in this scenario. The areas marked in dark red, e.g., the lower filament, which is part of the current layer, or the right side, which is not part of the current layer. However, the presence of the left filament marked in light red, which is also not part of the current layer, further complicates the interpretation. These intricate variations emphasize the difficulties associated with filtering the dried state but all results in general.

6.3 Lateral Scan

Figure 6.18 and Figure 6.19 show the side view of the lateral scan. Figure 6.19 in particular shows a blop of over-extrusion in the first layer as shown in Figure 6.3, Figure 6.18 on the other hand shows a slight overhang in the last layer

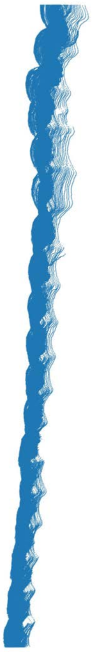


Figure 6.18: Left side lateral view

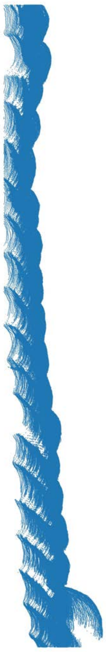


Figure 6.19: Right side lateral view

The over-extrusion anomaly, depicted as a distinct blop in Figure 6.19 and more detailed in Figure 6.4 may also be observed in Figure 6.20. It is clear how the color is significantly darker and concentrated in one location.

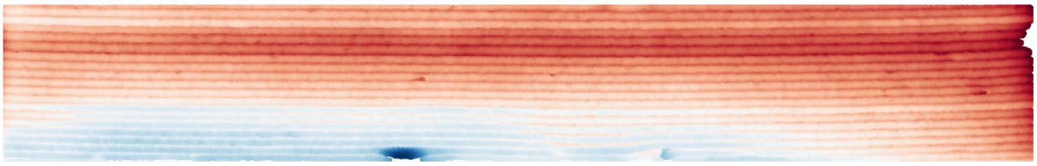


Figure 6.20: Right side wet front view

6.3.1 Lateral heat map

Figure 6.21 illustrates the front view of the left side. The heatmap effectively highlights deviations and inter-filament connections, helping in a more insightful analysis. For instance, the progressive inward inclination of the print along its height is noticeable from the illustration. This intriguing phenomenon can be attributed to a convergence of two potential contributing factors. Firstly, a minor misalignment in the transformation angle, potentially arising from the pre-print measurement. The influence of the print weight could have induced subtle adjustments to this angle during execution. Secondly, the issue could originate from the internal structure's inadequate contact with the outer wall. As the printing progressed vertically, the lack of robust connections seemingly prompted a gradual shift in the upper layers, resulting in the observed inward tilt.

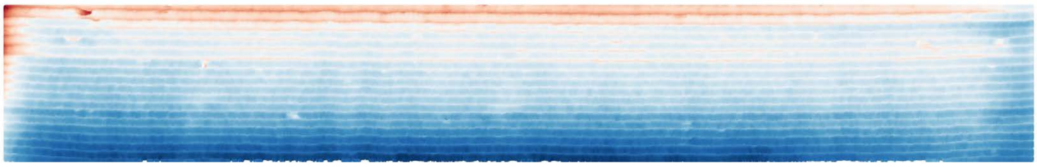


Figure 6.21: Left side wet front view

Figure 6.22 shows the same scan after the structure dried. Here, it becomes more evident the non-uniform shrinkage along the height. The top layer had a significantly higher shrinkage rate than the lower layer. Additionally, it is observed that the wall is less homogeneous along the length. The structure is observed to have a crack nearly in the center. This crack is outside the scope of the study but is attributed to the uneven drying process.

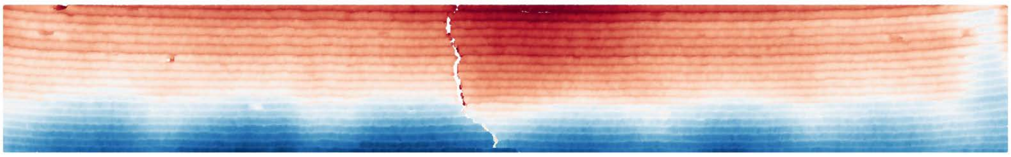


Figure 6.22: Left side dry front view

6.3.2 Post processing volume

Once the wall structure is thoroughly examined from both the side and front perspectives, it becomes easier to grasp the necessary infill volume. Table 6.1 provides various statistical measurements for the post-processing of the wall. Of these, the average volume per millimeter of the wall stands out as the most critical value, representing the actual material needed to achieve the desired finishing state.

Furthermore, Table 6.1 includes the maximum and minimum values, which offer insights into the potential material savings attainable with an improved print. The standard deviation, on the other hand, reflects the wall's roughness and aids in determining the uniformity with which the material must be distributed for optimal results.

Table 6.1: Statistics of required material

Scan	Max [mm^2]	Min [mm^2]	Average [mm^2]	Std
Left fresh	374.25	241.10	274.97	11.29
Left dry	422.58	287.88	352.69	24.82
Right fresh	578.38	397.99	439.59	32.46
Right dry	538.25	306.66	428.22	31.09

Furthermore, Table 6.1 serves to emphasize the contrast between the right and left sides of the wall. This table effectively quantifies the impact of the aforementioned blop, which was previously addressed. Evidently, this irregularity contributes to an approximate 50 % surge in material consumption compared to the other side in the fresh state. Moreover, the table accentuates the pronounced disparities observed across various profiles, particularly in the context of the right wall. These discrepancies can be directly attributed to the influence of the blop anomaly. The resultant material distribution becomes noticeably variable due to this imperfection in the right wall's surface.

Conversely, on the left wall, a distinct variability emerges when examining the dry state. This discrepancy is mainly attributed to two distinct factors. Primarily, the uneven shrinkage of the upper layers of the print results in this substantial divergence. Additionally, the inward inclination further amplifies this deviation, creating a compounding effect on the material distribution and contributing to the elevated deviation observed.

6.4 Discussions

The limitations outlined in section 3.8 introduce a level of uncertainty to the obtained results. An illustrative example is in the as-designed layer creation process, where the dpi is utilized to determine line thickness within the plot. However, a more robust implementation is critical to ensure that the base measurement plot truly represents the intended layer design. This limitation becomes particularly evident in critical sections like corners and turning points, mirroring the locations previously identified with under-extrusion issues in section 6.2.

Furthermore, the lack of reliable contact between the inner structure and outer wall emerges as a key factor leading to failure. It is essential to accurately identify this specific problem. Presently, the shape deviation map falls short of accomplishing this task due to its broad tolerance range. To address this, a tighter tolerance could be strategically employed at these critical junctures. This enhancement would ensure that the identified layer for shape deviation corresponds to the current layer rather than also capturing the previous one.

In sum, the outcomes are closely aligned with expectations, effectively highlighting both the strengths and limitations of the implementation. The results appropriately illustrate areas of good results, while also properly identifying instances where the as-built object deviated from the intended structure. However, one aspect that remains to be further developed is a more practical alternative for visually embedding these results within the IFC file. This would ideally provide a comprehensive and intuitive representation of the results, enhancing their interpretability within the designed file.

Chapter 7

Application examples and Conclusions

After going through the theoretical background in chapter 2 and understanding the current needs of the industry and the push for advanced technology. Followed by the methodology and an understanding of what this research would do. All the way through to the setup and its implementation, to finally get to the test study and proof of application. This chapter will close up the topic by suggesting examples of current applications, recommendations on future developments, a study conclusion, and some final words.

7.1 Application examples

Some examples of the current applications will be given here, but more importantly, a short path into further developing the tools to get a more extensive range of applications.

7.1.1 Current applications

Currently, the applications are limited to specimens similar to those analyzed in this study. They must be planar prints, oriented in the x-axis, and be uniaxial prints. By uniaxial, it means that it is not curved or L-shaped. That the print is exclusively around one axis. This is due to the scanning limitation exclusively in one direction.

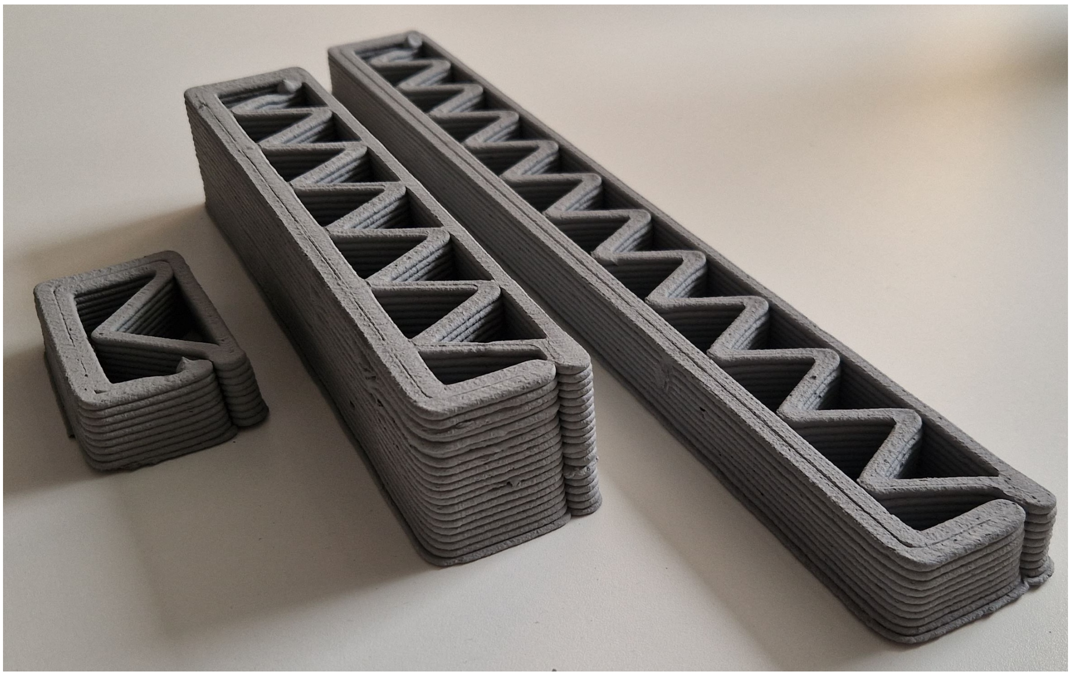


Figure 7.1: Multiple possible samples

A second option, exclusively for top scans, would be possible for any shape as long as the distance from the printing axes never exceeds 40mm on either side from the center. That way, it is inside the Field of View (FOV) (as described in Figure 4.2), and a center may be found to position with the IFC shape.

7.1.2 Future work

0. Independent post-processing

This is an independent step that has to occur at any given time before the real-time feedback. It will allow for the post-processing implemented in this research to be used without supervision. For this, it will require adjustments to understand what is an acceptable defect or what is not. Create outputs depending on what has been scanned to then be used for decision-making. This step does not depend on previous steps but is essential for proper and reliable communication between collaborative robots.

1. The scanning process should be done for any given path

This is the most significant development that can happen, it will allow, with nothing else needed to analyze any other geometry and not just the ones described above. It will enable checking on L-shaped structures, nonplanar prints, curved structures, and so on, as seen in Figure 7.2 and Figure 7.3. It will allow the study of an infinite possibility of structures without any further adjustments.

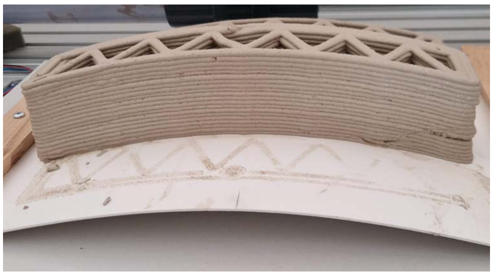


Figure 7.2: Non planar print

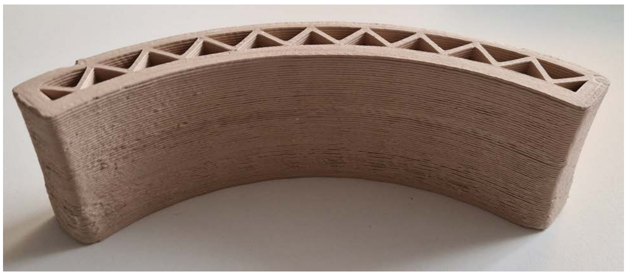


Figure 7.3: Curved print

2. The scanning process should work with variable speed profiles

As already mentioned, the scanning is done at a constant speed, nevertheless, if the scanning path is not a straight line, it is essential to adjust the speed profile for curves and tilts. This ties it to the previous recommendation but is critical for proper results. This will also imply a modification in how the data is treated, as it will need the position and speed to determine the spacing between profiles if similar technology is used.

3. A proper calibration should be implemented between printing and scanning robots

This would allow both robots to work on the same reference coordinate system reducing errors and also manual work. The problem described in subsection 5.3.1 would be overcome, also, it would be the first step towards collaboration. Additionally to that, it would make it easier to scan a specific region automatically and not with manual help due to the current limitations with positioning.

4. Collaboration between robots

Once the robot can scan on any path, have a variable speed, and share local coordinates with the printing robot, they must communicate with each other. They must have a real-time update of the position of the other robot and check for overlaps. Check not only for the endpoint but also the attachments and the body of the robot.

5. Real-time feedback information between robots

Once all the steps before have been accomplished, the end goal will be to use the scanned data to provide feedback information to the printing robot in real-time, and either trigger an imminent fix or an adjustment in the printing process depending on the needs. The RTDE library is of utmost importance as it will allow unsupervised changes on the path in real-time. It would additionally be necessary to brainstorm and come up with possible scenarios and decisions on what each scenario should trigger as a response to the printing robot.

7.2 Conclusions

The implementation proves that images and statistical values are a great way to store the results of the as-built object compared to the design. Additionally, it showed flexibility for other structures that do not have a uniform geometry for all layers, and more importantly, the implementation was designed to be easily expanded and maintained. Additionally, this study has proven effective at showcasing the importance of doing progressive scanning and real-time adjustments to improve the as-built object using deviations from the as-designed model. For this, it is necessary to improve the implementation and get the robots to effectively collaborate to increase efficiency and limit unsafe scenarios.

It is concluded that the infill volume is highly sensitive to defects. This was expected as a design decision is not to remove material but limit the post-processing to adding material to the level of the critical point. Additionally, it is observed that the top layers have a higher relative shrinkage rate that should be measured to overcome this during future designs.

Lastly, the shrunk version should be considered. A comparison between the as-designed version with the as-built when dry proves impractical when the designed version is used to extrude the fresh object and does not consider the shrinking process the object overgoes. This generates extremely high deviations that do not provide useful information for real-time adjustments.

7.3 Final words

After going through this study, it is clear how much there is to study in this field. This study was limited to the digital twin and information fabrication modeling world. It is crucial to keep the research in this field and continue milestone after milestone to achieve a real-time feedback information loop that will allow the extrusion process and the sensors to communicate and use the information to improve the process autonomously. As explained in subsection 7.1.2, this is the ultimate goal of a DT to have the latest information and use it in real-time to independently modify the extrusion with improvements and log the reasoning behind it.

Also, studies should be done, and tools should be implemented flexibly because they must be independent of the hardware used. Alternatively, slight modifications can be done when using a different robot, extruder, and, most importantly, other sensors. As explained during the current research, this is important as often research is done on a small scale, as was the case here, but it was to be done in a way that can then be extrapolated to a larger scale. If this is done, it can not be that the research has to start again because of changes in the hardware and an implementation fixed to it.

Appendix A

More Results

A.1 Top Scan

A.1.1 Raw

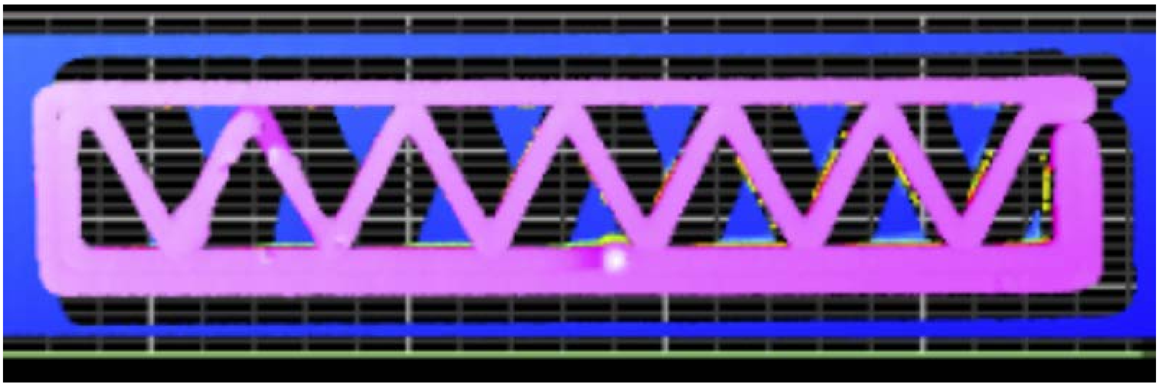


Figure A.1: Scan raw wet data layer 14

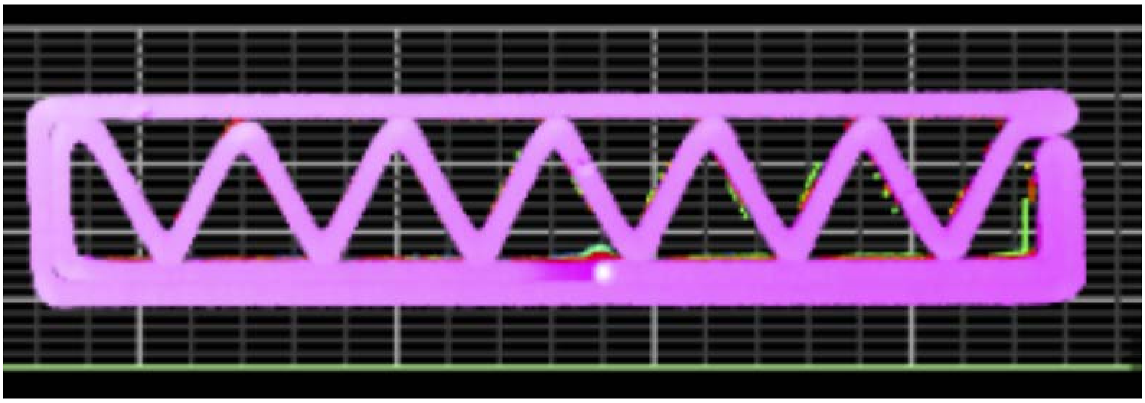


Figure A.2: Scan raw wet data layer 17

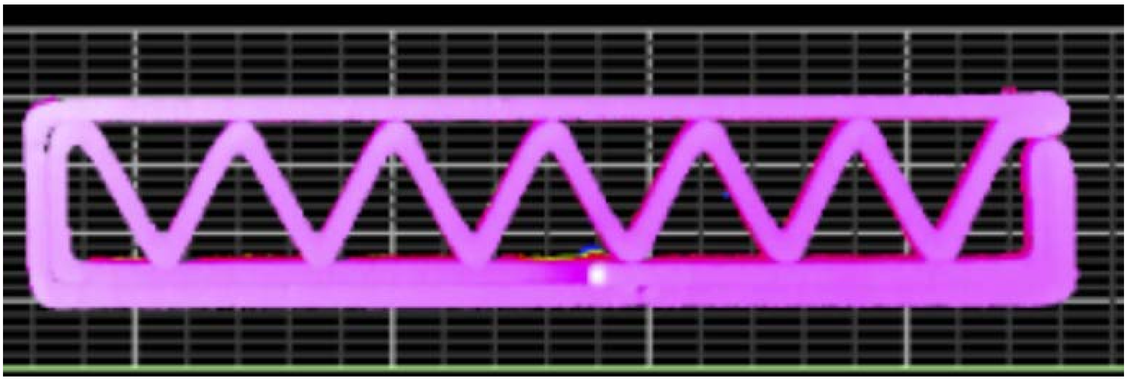


Figure A.3: Scan raw wet data layer 20

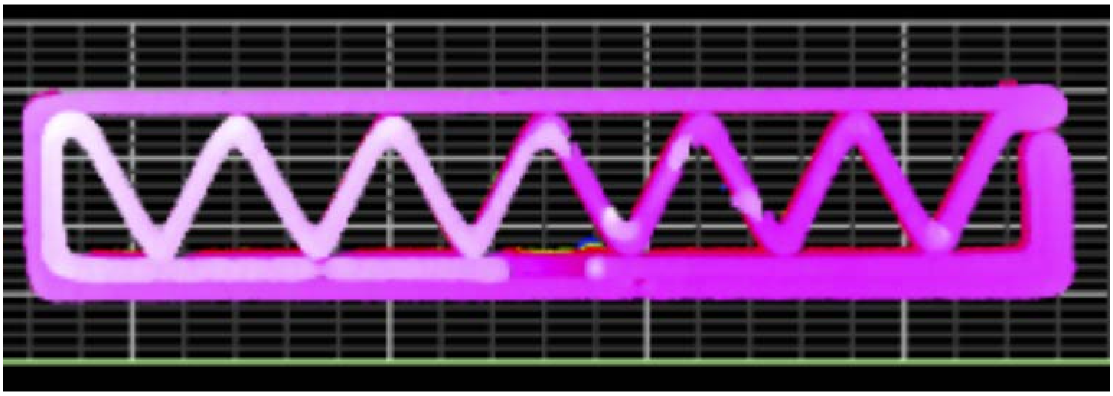


Figure A.4: Scan raw wet data layer 21

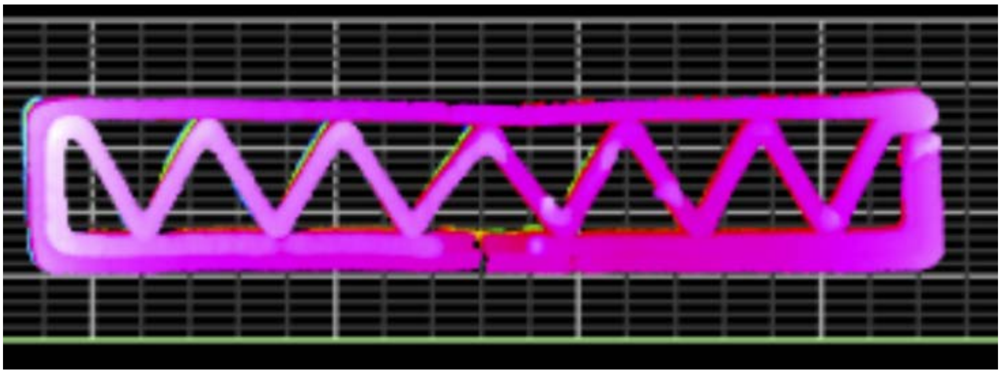


Figure A.5: Scan raw dry data layer 21

A.1.2 Fresh state

Layer 14

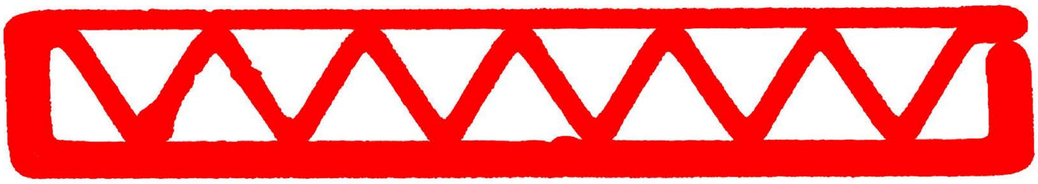


Figure A.6: Scan wet data layer 14

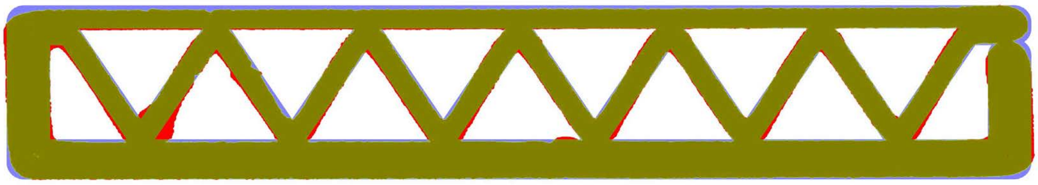


Figure A.7: Shape map wet data layer 14

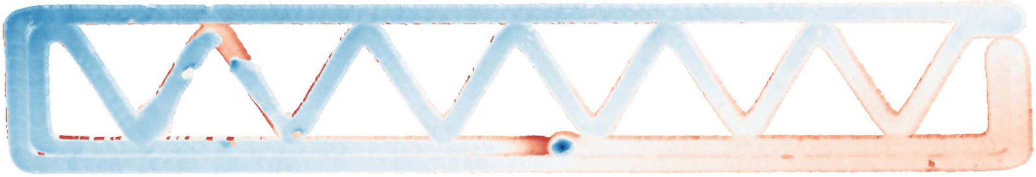


Figure A.8: Height map wet data layer 14

Layer 17

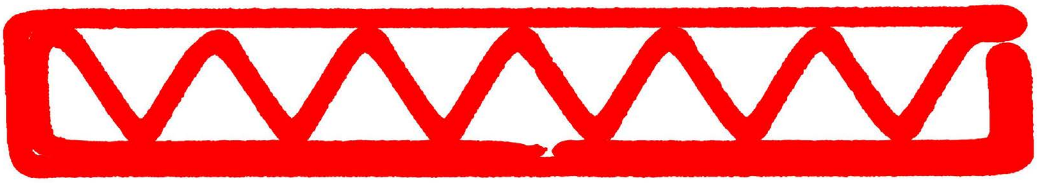


Figure A.9: Scan wet data layer 17

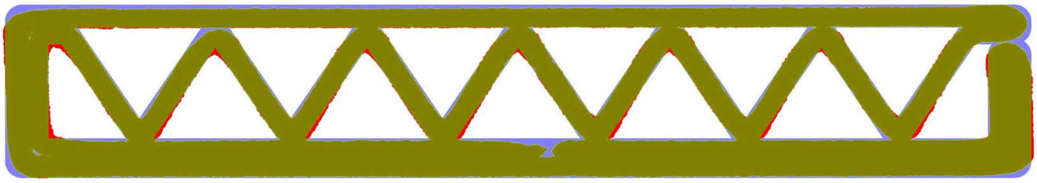


Figure A.10: Shape map wet data layer 17

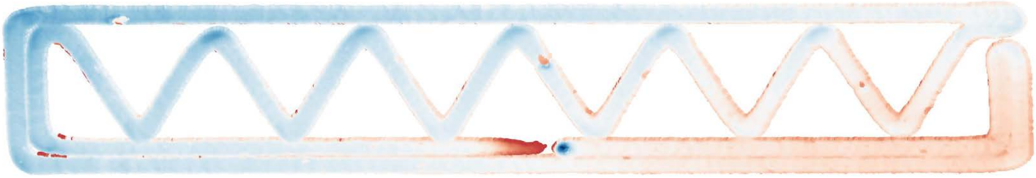


Figure A.11: Height map wet data layer 17

Layer 20

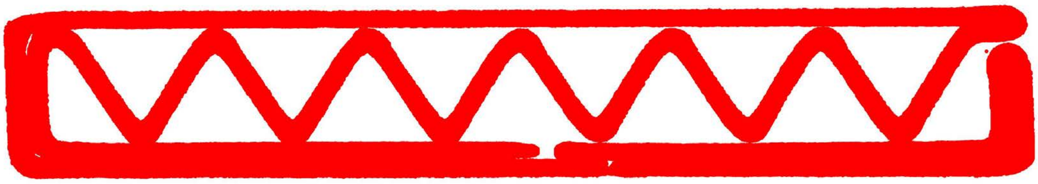


Figure A.12: Scan wet data layer 20

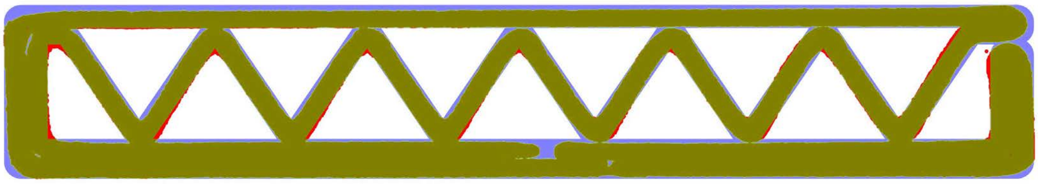


Figure A.13: Shape map wet data layer 20

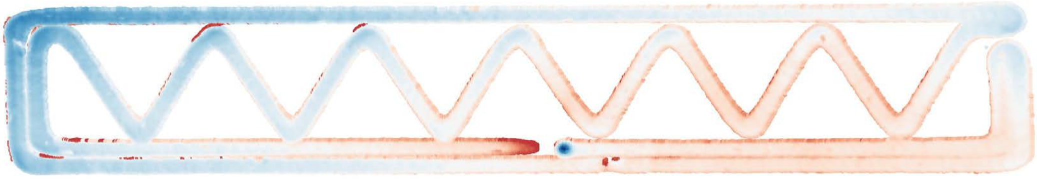


Figure A.14: Height map wet data layer 20

Layer 21

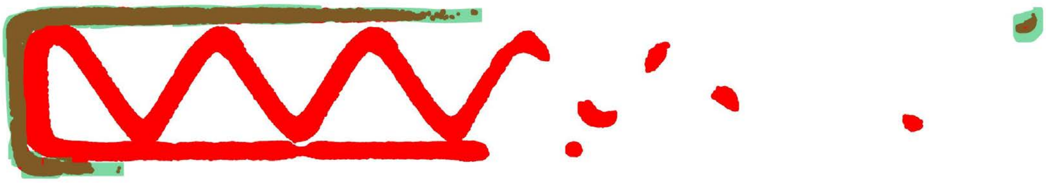


Figure A.15: Scan wet data layer 21

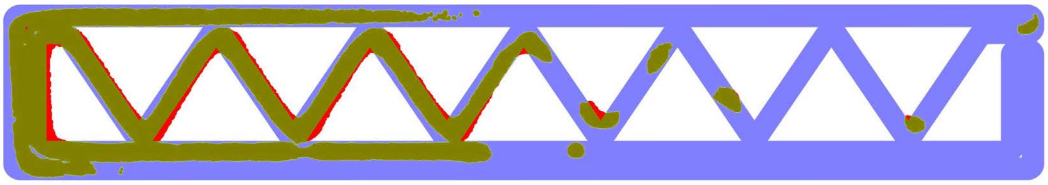


Figure A.16: Shape map wet data layer 21

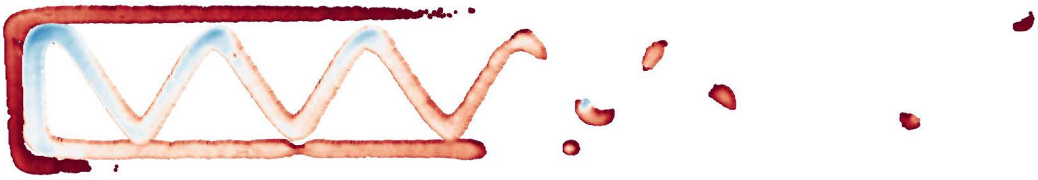


Figure A.17: Height map wet data layer 21

A.1.3 Dry state

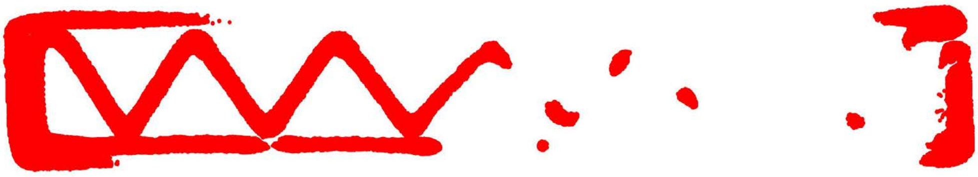


Figure A.18: Scan dry data layer 21

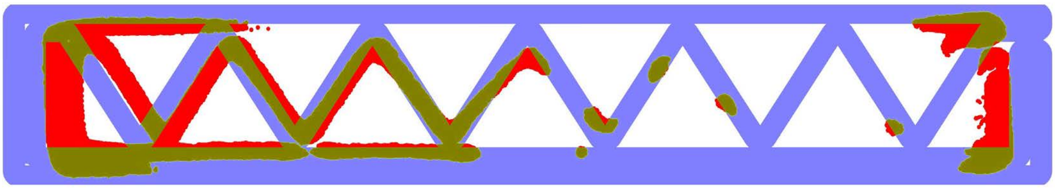


Figure A.19: Shape map dry data layer 21

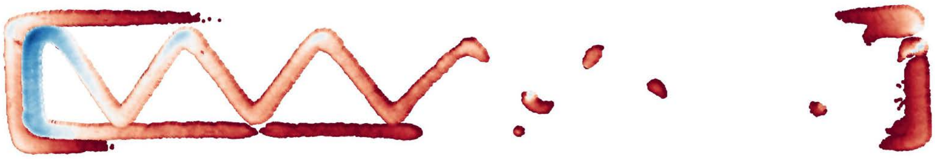


Figure A.20: Height map dry data layer 21

A.2 Lateral Scan

A.2.1 Left side

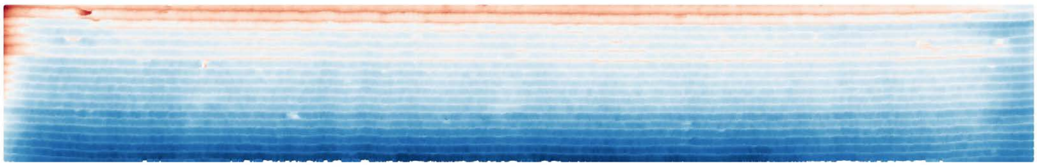


Figure A.21: Scan data left front side wet

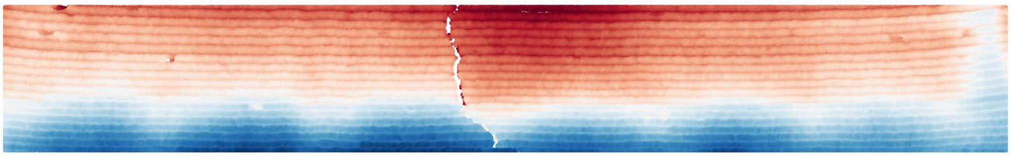


Figure A.22: Scan data left front side dry

A.2.2 Right side

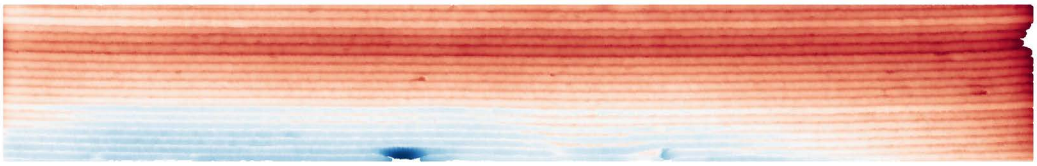


Figure A.23: Scan data right front side wet

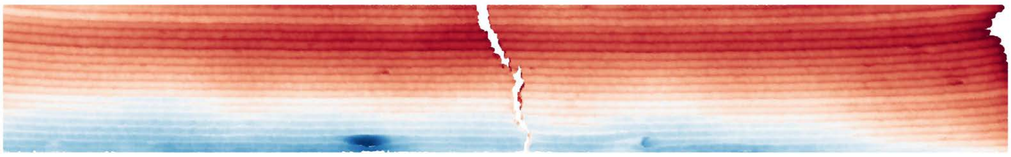


Figure A.24: Scan data right front side dry

Appendix B

Code

https://gitlab.lrz.de/00000000014A96C7/master-thesis.git

Bibliography

AHUETT-GARZA, H., & KURFESS, T. (2018). A brief discussion on the trends of habilitating technologies for industry 4.0 and smart manufacturing [Industry 4.0 and Smart Manufacturing]. *Manufacturing Letters*, *15*, 60–63. https://doi.org/https://doi.org/10.1016/j.mfglet.2018.02.011

ANINGER, A. (2022). *From fabrication information models to simulation models* (Master's thesis). Technische Universität München.

BALLA, M., HAFFNER, O., KUČERA, E., & CIGÁNEK, J. (2023). Educational case studies: Creating a digital twin of the production line in tia portal, unity, and game4automation framework. *Sensors*, *23*(10). https://doi.org/10.3390/s23104977

BORRMANN, A., BEETZ, J., KOCH, C., & LIEBICH, T. (2018). Chapter 6 industry foundation classes – a standardized data model for the vendor-neutral exchange of digital building models. In *Building information modeling: Technology foundations and industry practice*. Springer.

BUSWELL, R., de SILVA, W. L., JONES, S., & DIRRENBERGER, J. (2018). 3d printing using concrete extrusion: A roadmap for research. *Cement and Concrete Research*, *112*, 37–49. https://doi.org/10.1016/j.cemconres.2018.05.006

COMMINAL, R., da SILVA, W. R. L., ANDERSEN, T. J., STANG, H., & SPANGENBERG, J. (2020a). Influence of processing parameters on the layer geometry in 3d concrete printing: Experiments and modelling. In *RILEM bookseries* (pp. 852–862). Springer International Publishing. https://doi.org/10.1007/978-3-030-49916-7_83

COMMINAL, R., da SILVA, W. R. L., ANDERSEN, T. J., STANG, H., & SPANGENBERG, J. (2020b). Modelling of 3d concrete printing based on computational fluid dynamics. *Cement and Concrete Research*, *138*, 106256. https://doi.org/10.1016/j.cemconres.2020.106256

COMMINAL, R., SERDECZNY, M. P., PEDERSEN, D. B., & SPANGENBERG, J. (2019). Motion planning and numerical simulation of material deposition at corners in extrusion additive manufacturing. *Additive Manufacturing*, *29*, 100753. https://doi.org/10.1016/j.addma.2019.06.005

CORRADINI, F., & SILVESTRI, M. (2022). Design and testing of a digital twin for monitoring and quality assessment of material extrusion process. *Additive Manufacturing*, *51*, 102633. https://doi.org/10.1016/j.addma.2022.102633

DASKALOVA, M. (2018). The "digital twin" – a bridge between the physical and the digital world. https://cobuilder.com/en/the-digital-twin-a-bridge-between-the-physical-and-the-digital-world/

DIMITROV, A., & GOLPARVAR-FARD, M. (2014). Robust nurbs surface fitting from unorganized 3d point clouds for infrastructure as-built modeling. https://api.semanticscholar.org/CorpusID:3072748

DÖRFLER, K., DIELEMANS, G., LACHMAYER, L., RECKER, T., RAATZ, A., LOWKE, D., & GERKE, M. (2022). Additive manufacturing using mobile robots: Opportunities and challenges for building construction. *Cement and Concrete Research*, *158*, 106772. https://doi.org/10.1016/j.cemconres.2022.106772

FLOW-3D. (2021, September 22). *Dtu digital twins for additive manufacturing*. Flow 3D. https://youtu.be/Q3nNg0FUMPU

Glaessgen, E., & Stargel, D. (2012). The digital twin paradigm for future nasa and u.s. air force vehicles. https://doi.org/10.2514/6.2012-1818

Hack, N., & Kloft, H. (2020). Shotcrete 3d printing technology for the fabrication of slender fully reinforced freeform concrete elements with high surface quality: A real-scale demonstrator. In *RILEM bookseries* (pp. 1128–1137). Springer International Publishing. https://doi.org/10.1007/978-3-030-49916-7_107

Helou, J. (2023). Rtde ur control. https://gitlab.lrz.de/00000000014AB830/rtde-ur-control

Kagermann, H., Wahlster, W., & Helbig, J. (2013). *Recommendations for implementing the strategic initiative industrie 4.0 – securing the future of german manufacturing industry* (Final Report of the Industrie 4.0 Working Group). acatech – National Academy of Science and Engineering. München. http://forschungsunion.de/pdf/industrie_4_0_final_report.pdf

Kruger, J., Cho, S., Zeranka, S., Viljoen, C., & van Zijl, G. (2020). 3d concrete printer parameter optimisation for high rate digital construction avoiding plastic collapse. *Composites Part B: Engineering*, *183*, 107660. https://doi.org/10.1016/j.compositesb.2019.107660

LMI Technologies, I. (2019). *Gocator line profile sensor user manual*. Version Version 5.2.x. LMI.

Mollah, M. T., Comminal, R., Serdeczny, M. P., Pedersen, D. B., & Spangenberg, J. (2021). Stability and deformations of deposited layers in material extrusion additive manufacturing. *Additive Manufacturing*, *46*, 102193. https://doi.org/10.1016/j.addma.2021.102193

Mollah, M. T., Comminal, R., Serdeczny, M. P., Pedersen, D. B., & Spangenberg, J. (2022). Numerical predictions of bottom layer stability in material extrusion additive manufacturing. *JOM*, *74*(3), 1096–1101. https://doi.org/10.1007/s11837-021-05035-9

Mollah, M. T., Serdeczny, M., Comminal, R., Šeta, B., Brander, M., Pedersen, D., & Spangenberg, J. (2022). A numerical investigation of the inter-layer bond and surface roughness during the yield stress buildup in wet-on-wet material extrusion additive manufacturing, 4.

Morwald, T., Richtsfeld, A., Prankl, J., Zillich, M., & Vincze, M. (2013). Geometric data abstraction using b-splines for range image segmentation. *2013 IEEE International Conference on Robotics and Automation*. https://doi.org/10.1109/icra.2013.6630569

Mörwald, T., Balzer, J., & Vincze, M. (2016). Modeling connected regions in arbitrary planar point clouds by robust b-spline approximation. *Robotics and Autonomous Systems*, *76*, 141–151. https://doi.org/10.1016/j.robot.2015.11.006

MUTESI, E. T., & KYAKULA, M. (2011). Application of ict in the construction industry in kampala. *Second International Conference on Advances in Engineering and Technology*, *20*, 263–269.

OZTEMEL & GURSEV. (2020). Literature review of industry 4.0 and related technologies. *Journal of Intelligent Manufacturing*, *31*(1), 127–182. https://doi.org/10.1007/s10845-018-1433-8

PAOLINI, A., KOLLMANNSBERGER, S., & RANK, E. (2019). Additive manufacturing in construction: A review on processes, applications, and digital planning methods. *Additive Manufacturing*, *30*, 100894. https://doi.org/https://doi.org/10.1016/j.addma.2019.100894

REINOLD, J., NERELLA, V. N., MECHTCHERINE, V., & MESCHKE, G. (2022). Extrusion process simulation and layer shape prediction during 3d-concrete-printing using the particle finite element method. *Automation in Construction*, *136*, 104173. https://doi.org/10.1016/j.autcon.2022.104173

SERDECZNY, M. P., COMMINAL, R., PEDERSEN, D. B., & SPANGENBERG, J. (2019). Numerical simulations of the mesostructure formation in material extrusion additive manufacturing. *Additive Manufacturing*, *28*, 419–429. https://doi.org/10.1016/j.addma.2019.05.024

SLEPICKA, M., HELOU, J., & BORRMANN, A. (2023). Real-time data exchange (rtde) robot control integration for fabrication information modeling.

SLEPICKA, M., MAWAS, K., BORRMANN, A., & GERKE, M. (2022). Digital twinning in additive manufacturing – closing the digital-physical-digital loop by automated integration of captured geometric data into fabrication information models.

SLEPICKA, M., VILGERTSHOFER, S., & BORRMANN, A. (2021). Fabrication information modeling: Closing the gap between building information modeling and digital fabrication. *Proceedings of the 38th International Symposium on Automation and Robotics in Construction (ISARC)*. https://doi.org/10.22260/isarc2021/0004

SLEPICKA, M., VILGERTSHOFER, S., & BORRMANN, A. (2022). Fabrication information modeling: Interfacing building information modeling with digital fabrication. *Construction Robotics*, *6*(2), 87–99. https://doi.org/10.1007/s41693-022-00075-2

SPANGENBERG, J., da SILVA, W. R. L., COMMINAL, R., MOLLAH, M. T., ANDERSEN, T. J., & STANG, H. (2021). Numerical simulation of multi-layer 3d concrete printing. *RILEM Technical Letters*, *6*, 119–123. https://doi.org/10.21809/rilemtechlett.2021.142

TAO, F., ZHANG, H., LIU, A., & NEE, A. Y. C. (2019). Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*, *15*(4), 2405–2415. https://doi.org/10.1109/TII.2018.2873186

TAUFIK, M., & JAIN, P. K. (2016). A study of build edge profile for prediction of surface roughness in fused deposition modeling. *Journal of Manufacturing Science and Engineering*, *138*(6). https://doi.org/10.1115/1.4032193

TUHAISE, V. V., TAH, J. H. M., & ABANDA, F. H. (2023). Technologies for digital twin applications in construction. *Automation in Construction*, *152*, 104931. https://doi.org/https://doi.org/10.1016/j.autcon.2023.104931

What is industry 4.0? (2021). https://dat4zero.eu/what-is-industry-4-0/

ZHANG, L., ZHOU, W., & CHEN, X. (2022). Digital twins and additive manufacturing. In *Digital twins: Basics and applications* (pp. 27–35). Springer International Publishing. https://doi.org/10.1007/978-3-031-11401-4_4