

In-situ Automated Scan-assisted Repair using Hybrid Manufacturing

by

Remy Samson

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering
University of Alberta

© Remy Samson, 2021

Abstract

Mechanical parts sustain significant wear throughout their lifetime and consequently get defects such as cracks, abrasion, erosion, or breakage. For instance, turbine blades experience high pressure and temperature fields which can cause damage, corrosion, and indentation. In this case, the parts are replaced, resulting in significant time and financial costs. One possible remedy for these losses is to repair the volume using additive manufacturing (AM) through directed energy deposition processes. AM offers multiple advantages such as high precision and accuracy, a limited amount of material used for repair, and reduction of remanufacturing time. This project focuses on an in-situ, automated repair strategy by plasma transfer arc AM. A sequence of computational algorithms and a hybrid CNC machine with an integrated 3D scanner are used to repair worn parts. All the steps for repair are applied in-situ, from the scanning process to the final repair process. The integrated process results in a single-setup operation, reducing the part movements between stations and associated setup times. The approach also provides a unified digital toolchain to enable one-step metrology, subtractive manufacturing, and AM process.

Preface

Rémy Samson performed the whole program development part and experimental work presented in this thesis under the supervision of Dr. Ahmed Jawad Qureshi.

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor, Dr. Ahmed Jawad Qureshi, for his research ideas, continuous support, constant discussion, training, education, and patience, which has helped me progress throughout the research project also preparing for my future career.

I would also like to thank Dr. Thomas Lehmann, whom I had the chance to work with, for his expertise, guidance, and support to realize this project.

Thanks to all the students who have been part of the group for the last two years and who made my master's study more enjoyable. A special thank you to Lei Huang and Arthur Oliveira for their exciting discussions and suggestions on the experimental setup.

Finally, I would like to thank all my friends for their support, and I am particularly grateful to my parents for all the love and support they have given me throughout my whole education and life.

Merci à vous Quentin Champdoizeau, Alban Pouplard, Jonas Valloton and Saadman Sakib Rahman pour la bonne humeur et les fous rires.

Contents

Abstract.....	i
Preface.....	iii
Acknowledgments.....	iv
Contents	v
Chapter 1 Introduction	1
1.1. General Overview	1
1.2. Motivations.....	1
1.3. Research objectives.....	3
1.4. Organization of the thesis.....	4
1.5. Project Description.....	4
Chapter 2 State of the art	7
2.1. Introduction	7
2.2. Scanning Technologies (References).....	8
2.2.1. Contact based 3D scanning.....	8
2.2.2. Photogrammetry.....	9
2.2.3. LASER 3D scanning.....	9
2.2.4. Time-of-flight 3D Scanning.....	9
2.2.5. Structured Light 3D Scanning	10
2.3. Volume identification and 3D Model Reconstruction	11
2.4. Damaged surface Identification and extraction.....	13
2.5. Repair Process	15
2.6. AM Technologies For Repair Change picture	16
2.6.1. LASER Engineered Net Shaping and LASER Cladding.....	16
2.6.2. Electron Beam Freeform Fabrication.....	17
2.6.3. Wire Arc Additive Manufacturing.....	17
2.6.4. Plasma Transferred Arc	17
2.7. Literature Review Summary	19
Chapter 3 Initial Setup and Software Development	20
3.1. Experimental Setup Design.....	20
3.1.1. Initial setup.....	20

3.1.2.	Upgrades	22
3.2.	Scanning process	23
3.2.1.	Marker recognition.....	23
3.2.2.	Inverse kinematics	25
3.3.	3D Model Reconstruction	28
3.3.1.	Flood fill Algorithm.....	29
3.3.2.	K-D Tree	31
3.4.	Damage Volume Identification	33
3.4.1.	Sample Elimination for Generating Poisson Disk Sample Sets.....	34
3.4.2.	Iterative Closest Point Algorithm	35
3.4.3.	QuickHull Algorithm	40
3.4.4.	Planar surface detection	42
3.4.5.	Target and source.....	45
3.5.	Damaged volume extraction.....	46
3.5.1.	Logistic function	46
3.5.2.	Gap filling procedure	47
3.6.	Surface preparation	48
3.6.1.	Manufacturing constraints	49
3.6.2.	Surface preparation based on region growing algorithm.....	50
3.6.3.	Minimum Bounding Box algorithm.....	50
3.6.4.	Surface preparation based on Minkowski differences.....	53
3.7.	Pre-repaired volumes computation.....	54
3.8.	Surface reconstruction.....	59
3.8.1.	Poisson surface reconstruction algorithm	60
3.8.2.	Marching cube algorithm.....	62
3.9.	Toolpath generation.....	65
Chapter 4	Development of a Hybrid system for automated repair	67
4.1.	Software and Hardware	67
4.1.1.	Main PC	68
4.1.2.	Central Unit.....	68
4.1.3.	The Gantry system	68
4.1.4.	PTA and cutting tool.....	68
4.1.5.	3D scanner	70

4.2.	Graphical user interfaces (GUI)/ programs	70
4.2.1.	Python GUI	71
4.2.2.	MASTERCAM & APLUS	71
4.2.3.	KmotionCNC	72
4.3.	Communication inter-software.....	72
4.4.	Communication between the equipment.....	76
4.5.	Calibration of the LAMPP system	79
4.5.1.	Calibration of the Gantry system	79
Chapter 5	Experimental results.....	81
5.1.	First experimental results	81
5.2.	Experimental results using the hybrid system.....	92
5.3.	Discussions.....	94
Chapter 6	Conclusion and future work	99
6.1.	Conclusion.....	99
6.2.	Future work and applications	101
References	103

List of Tables

Table 2.1. 3D scanning technology advantages and drawbacks.....	11
Table 3.2: Flood Fill algorithm.....	30
Table 3.3: ICP Algorithm.....	38
Table 3.4: QuickHull Algorithm.....	40
Table 3.5: Gap filling procedure.....	48
Table 3.6 : Minimum bounding box	51
Table 3.7: Ip computation	55
Table 3.8: Winding points	58
Table 4.1 Actions commands shared by MASTERCAM and KmotionCNC	75
Table 5.2. Criteria of selection of Sample A, B, C.....	82
Table 5.3. Results of the ICP algorithm in terms of error and execution time computed with the presented algorithms MATLAB, and Cloud Compare algorithms.	83
Table 5.4. Results of the damage volume identification in terms of the number of identified points and execution time.....	85
Table 5.5. Comparison of the density of the damaged point cloud and V_d	85
Table 5.6. Results of the sequence of computational algorithms for the four types of damage.....	89
Table 5.7. Table of validations	99

List of Figures

Figure 1.1. Additive manufacturing process	2
Figure 1.2 Scan assisted in-situ automated repair process	3
Figure 1.3 Thesis organization	6
Figure 2.1 AM technologies and characteristics	18
Figure 3.1 Design of the central unit	21
Figure 3.2 Initial setup and equipment	21
Figure 3.3. 3D printed marker pattern	23
Figure 3.4 Calibration protocol applied on the rotary table. a) Set up of the axes aligned with the system. b) Set up of the origin of the frame F_0 . c) Results of the calibration protocol.	24
Figure 3.5 Results of the inverse kinematics process for point cloud reconstruction. The red rebuilt point cloud is overlapped with a scan of the rotary table.....	27
Figure 3.6. 2D tree organization and associated graph	31
Figure 3.7. 3D tree organization and associated graph	31
Figure 3.10. Point Sampling. a) STL version of the Stanford rabbit. b) Random sampling (10000 points). c) Poisson disk sampling (10000 points).	34
Figure 3.11. Results ICP algorithm. a) Nominal and damaged volumes. b) First iteration. c) Second iteration. d) Third iteration. e) Fourth iteration. f) Result of the ICP algorithm.	39
Figure 3.12 Quick hull algorithm. a) Initial step: select a random simplex, select a plane and find the furthest above point. b) Second step: delete all the visible facets from the selected point, create the new facets. c) Step four. d) Final Result of the quick hull algorithm.	39
Figure 3.13. Result of surface planar detection algorithm applied on a dice.....	45
Figure 3.14. Slicing process of volume H_D	49
Figure 3.15 Top view of the axis-aligned bounding box and minimum bounding box ...	52

Figure 3.16. Examples of convex hull and inverted truncated cones used as inputs in the Minkowski algorithm.	53
Figure 3.17. winding number in 2 dimensions.....	56
Figure 3.18. Winding number in 3 dimensions.....	57
Figure 3.19. Discretized intersection of the indicatrice function with a cube. 15 configurations, symmetrical cases excluded.....	61
Figure 4.1. Final setup.....	67
Figure 4.2. Components of the system for hybrid repair and inter-connections.....	70
Figure 4.3. Communication between the software applications.....	73
Figure 4.4. Control unit for configuration 2.....	77
Figure 5.1. Samples used as inputs in the sequence of computational algorithms	81
Figure 5.2. Scans of the samples A, B, and C.....	82
Figure 5.3. Results of the ICP algorithm. a), d), g) Nominal point clouds. b), c), h) Damaged point clouds. c), f), i) Overlap of the nominal and damaged point clouds.	84
Figure 5.4. Damage identification and extraction. a), b), c) Damage identification on samples A, B, C. d), e), f) Damage extraction for the samples A, B, C.	84
Figure 5.5 General overview of the repaired volumes computation. a), f), k) Nominal volumes. b), g), l) damaged volumes. c), h), m) Overlap using the ICP algorithm. d), i), n) Pre-repaired volumes Ip . e), j), o) To-be-printed volumes A	86
Figure 5.6 Comparison between the nominal volumes and the repaired volumes for sample A, B, and C	86
Figure 5.7 Results of repair process applied on Sample A. a) Printed nominal volume. b) Printed damaged volume. c) Printed pre-repaired volume Ip . d) Repaired volume	87
Figure 5.8 Results of repair process applied on Sample B. a) Printed nominal volume. b) Printed damaged volume. c) Printed pre-repaired volume Ip . d) Repaired volume	87
Figure 5.9 Results of repair process applied on Sample C. a) Printed nominal volume. b) Printed damaged volume. c) Printed pre-repaired volume Ip . d) Repaired volume	87
Figure 5.10. Samples used to conduct the repair process on different types of damage. a) Attrition. b) Abrasion. c) Cavity. d) Fracture.	88

Figure 5.11. Execution time of the flood fill algorithm as a function of the number of points on the surfaces of the damaged volume. 90

Figure 5.12. Mesh deviation after Poisson surface reconstruction between the nominal and repaired volumes. a) Attrition. b) Abrasion. c) Cavity. d) Fracture..... 91

Figure 5.13. Results of the surface preparation process. a) Damaged part. b) Results of the surface preparation algorithms. c) Results of the machining process..... 92

Figure 5.14. Deviation study conducted on a pre-repaired dice..... 92

Figure 5.15. Aluminum samples after the surface preparation process have been applied. a) Attrition, b) Abrasion, c) Cavity, d) Fracture. 94

Figure 5.16. Comparison between the computed version of *I_p* and the actual volume *I_p*. a) Attrition. b) Abrasion. c) Cavity. d) Fracture. 95

Figure 5.17. Overlap of the scans of the pre-repaired volume and damaged volume with a highlight on the damaged and pre-repaired area. 95

Figure 5.18. Normal vector distribution relative to the z-axis for the four samples. a) Attrition. b) Abrasion. c) Cavity. d) Fracture 96

List of Abbreviations

AM	Additive Manufacturing
CNC	Computer Numerical Control
CAD	Computer-Assisted Design
CAM	Computer-Assisted Manufacturing
DED	Direct Energy Deposition
FDM	Fused Deposition Modeling
NURBS	Non-Uniform Rational B-Splines
LASER	Light Amplification by Stimulated Emission of Radiation
RANSAC	RAdom SAmple Consensus
ICP	Iterative Closest Point
SVM	Support Vector Machine
FPFH	Fast Point Feature Histogram
ARUCO	Augmented Reality University of Cordoba
ARTAG	Augmented Reality TAG
CCD	Charge-Couple Device
STL	Standard Triangle Language
PCS	Prominent Cross Section
PID	Proportional Integral Derivative
GI	Gaussian Image
UB	U-shaped boundary
CHB	convex-hull boundary

LENS	LASER Engineered Net Shaping
EBF	Electron Beam Freeform
WAAM	Wire Arc Additive Manufacturing
PTA	Plasma Transferred Arc
PCA	Principal Component Analysis

List of symbols

α	Yaw axis of the rotary table
β	Pitch axis of the rotary table
$SE(3)$	Lie algebra
\mathbb{F}_0	Reference frame of the CNC machine
\mathbb{F}_r	Frame of the rotary table function of α and β
\mathbb{F}_s	Frame of the scanner
\mathbf{D}	Point cloud of the damaged volume
\mathbf{I}	Point cloud of the nominal volume
\mathbf{T}_0	Initial transformation matrix of the ICP
\mathbf{T}	Transformation matrix of the ICP
\mathbf{P}_D	Points of \mathbf{D} on planar surfaces
\mathbf{P}_I	Points of \mathbf{I} on planar surfaces
\mathbf{H}_I	Convex hull of \mathbf{I}
\mathbf{H}_D	Convex hull of \mathbf{D}
\mathbf{V}_d	Damaged volume
$\mathbf{V}_{d,D}$	Points of \mathbf{V}_d initially inside \mathbf{D}
$\mathbf{V}_{d,I}$	Points of \mathbf{V}_d initially inside \mathbf{I}
$\mathbf{V}_{n,D}$	Points of \mathbf{D} not in \mathbf{V}_d
$\mathbf{V}_{n,I}$	Points of \mathbf{I} not in \mathbf{V}_d
$\mathbf{O}_{d,D}$	Points on the edges of $\mathbf{V}_{d,D}$
$\mathbf{O}_{d,I}$	Points on the edges of $\mathbf{V}_{d,I}$
\mathbf{H}_c	Convex hull of the set of cylinders/ inverted truncated cones
\mathbf{I}_p	Pre-repaired volume
\mathbf{A}	To-be-printed volume

Chapter 1 Introduction

1.1. General Overview

Complex manufactured parts made and used in cutting-edge industries such as aerospace, medicine, or mining require high sustainability during their lifespan. In addition, most of these parts undergo mechanical constraints and unstable environmental conditions. For instance, turbine blades must be perfectly designed to keep the turbine's balance while rotating (Mishra et Al. 2015). Moreover, the blades must sustain a significant gradient of temperature during the time of flight. In mining, bucket teeth experience high corrosion and wear (Choudhry 2020). These extreme conditions lead to different wear and damage such as dents, scrapes, cracks, or bending.

A traditional way to fix a damaged part is to replace it or manually repair it. In complex components such as turbine blades, the repair process is divided into refurbishment and machining (J. Liburdi and P. Lowden 1989). The resulting manual operation turns out to be costly in terms of time. And, even if machining is now implemented with computerized numerical control (CNC) system (Tao et al. 2015), the waste of material remains significant.

1.2. Motivations

A new manufacturing process called additive manufacturing (AM) was introduced in laboratories and some industries over the last decade (Wholers and Gornet 2014). In contrast to subtractive manufacturing, AM adds materials to create the desired shape. AM uses computer-assisted design (CAD) software or 3D scan acquisition to design or reproduce a virtual version of a part. Then a computed assisted manufacturing (CAM)

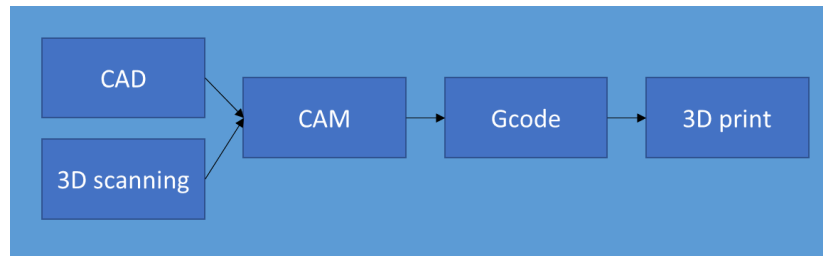


Figure 1.1. *Additive manufacturing process*

software calculates a toolpath, named g-code, that can be seen as a sequence of positions in space creating the desired geometrical shape. Finally, a controller board translates the g-code into electric signals sent to a 3D printing system that will deposit layer by layer the material (Yakout et al. 2018) (see **Figure 1.1**).

AM offers multiple advantages such as reducing manufacturing time, material consumption, and complex shape manufacturing. Moreover, computer-assisted printing ensures high accuracy and precision that satisfy the quality criteria of exacting companies such as aerospace, space, or biomedical.

Utilizing the advantages of AM, this project aims to combine AM and traditional subtractive manufacturing to repair worn components. In this project, the worn part is fixed using a specific AM process called direct energy deposition (DED).

Some projects have already been done using different AM and scanning technologies and have already turned out to be advantageous in cutting-edge fields (Rahito et al., 2019). However, repair using only additive manufacturing limits significantly the type of damage that can be fixed. Some defects, such as dents or sharp edges, cannot be restored directly using AM (Zhang et al., 2019). Moreover, even if some projects detailed the use of hybrid manufacturing, the whole process for repair is tedious, using multiple techniques and machines. Eventually, the generally developed repair strategy is based on high accuracy, and high-resolution scans obtained in laboratory conditions and do not apply in an actual

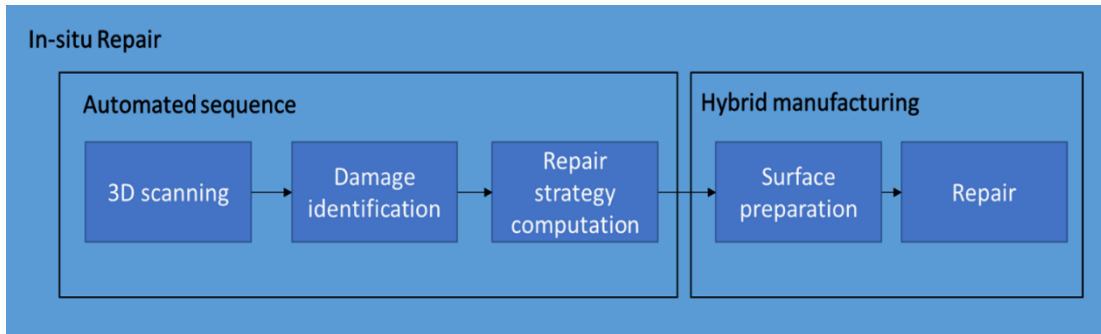


Figure 1.2 Scan assisted in-situ automated repair process

situation where the light condition is poor. This project proposes a robust in-situ automated scan-assisted repair strategy using hybrid manufacturing.

1.3. Research objectives

This research has the following objective:

“The development of a hybrid CNC machine with an integrated sequence of computational algorithms for scan-assisted, in-situ, automated repair.”

The research is built on the following hypotheses:

1. The point cloud of the damaged part obtained from the scanning process is considered noisy ($300\mu m$ of standard deviation).
2. The degrees of freedom of the hybrid system are limited to the three cartesian axes and two rotation axes.

The following research objectives are pursued (see **Figure 1.2**):

- Develop a system for reliable repair for high-value industries such as aerospace, mining, and medicine. This system must include computer vision and hybrid manufacturing equipment combining both additive and subtractive technics.
- Provide a one-stop scan-assisted in-situ automated tool chain for reliable post-repair while reducing lead time and financial costs. This tool chain must reach the

precision and repeatability required for remanufacturing process of high-value parts.

- Implement a sequence of robust computational algorithms that input the damaged part and deliver an output toolpath for repair using hybrid manufacturing.

1.4. Organization of the thesis

This thesis consists of six chapters. Chapter 1 outlines existing limitations, research motivation, and objectives. Chapter 2 details the different concepts and equipment used in this project and presents their advantages and drawbacks. Chapter 3 presents a standardized computational sequence of algorithms used for the development of a hybrid repair strategy. Chapter 4 describes its implementation into a hybrid system prototype specifically developed for hybrid repair. The architecture and development of this system are also depicted. Chapter 5 presents the results of the experiments conducted using the sequence of computational algorithms and the hybrid system. The results are divided into two categories. First, a proof of concept using the sequence of computational algorithms, and secondly, a proof of concept using the sequence and the hybrid system simultaneously. Finally, Chapter 6 provides conclusions and summarizes the research contributions, limitations, and future work. (See **Figure 1.3**)

1.5. Project Description

This project aims to develop an *automated scan-assisted in-situ repair* strategy using *hybrid manufacturing*:

- *Automated*: The automation of the process is conducted on two fronts. First, the sequence of computational algorithms in charge of the repair strategy computation

is automated. The operator only needs to enter a limited number of parameters. Then the algorithms compute the desired outputs. Secondly, the hybrid process executing the repair process is autonomous. The automation of the operation aims to reduce the time of repair and consequently decrease repair costs.

- *Scan-assisted*: A computer vision system is integrated into the system to identify and extract worn areas. A 3D scanner is used in this project.
- *In-situ*: In practice, the damaged part must be dismantled before any repair process. The sequence of computational algorithms is developed to be integrated into any robotic system to enable repair on-site to save time and money. In this project, the sequence is integrated into a 3-axis hybrid system
- *Hybrid manufacturing*: The system designed and implemented into this project can pre-repair using subtractive manufacturing (machining) and repair using additive manufacturing. Both pieces of equipment are mounted on a tool plate.

The first step for repair by hybrid manufacturing is the acquisition of the damaged area's geometry. By knowing the geometry of the worn part, an accurate AM repair process can be investigated. A 3D scanner is usually selected to obtain the data. This 3D scanner provides a point cloud of the damaged part. The second step of the project is to identify the damaged area and extract the points forming the damaged surfaces. Once the damages have been extracted, the third step of the repair process ensures the repair feasibility by computing a repair strategy. The worn part often gets damaged. Erosion, sharp angles, and tiny edges make the damaged part non-directly repairable. The repair strategy must include a surface preparation process. This step aims to create a pre-repaired volume that will be used to prepare the surfaces of the worn part using a subtractive method and create smooth

surfaces on which a new volume can be printed. Then, a to-be-printed or negative volume is computed regarding the results of the surface preparation process.

The results of the repair strategy are then exported to CAM software to generate the toolpath. A subtractive method is applied. This step is generally done using a CNC cutting machine to prepare the surfaces.

The final step is to 3D print the new printable volume on the worn part for repair using additive manufacturing technics such as direct energy deposition.

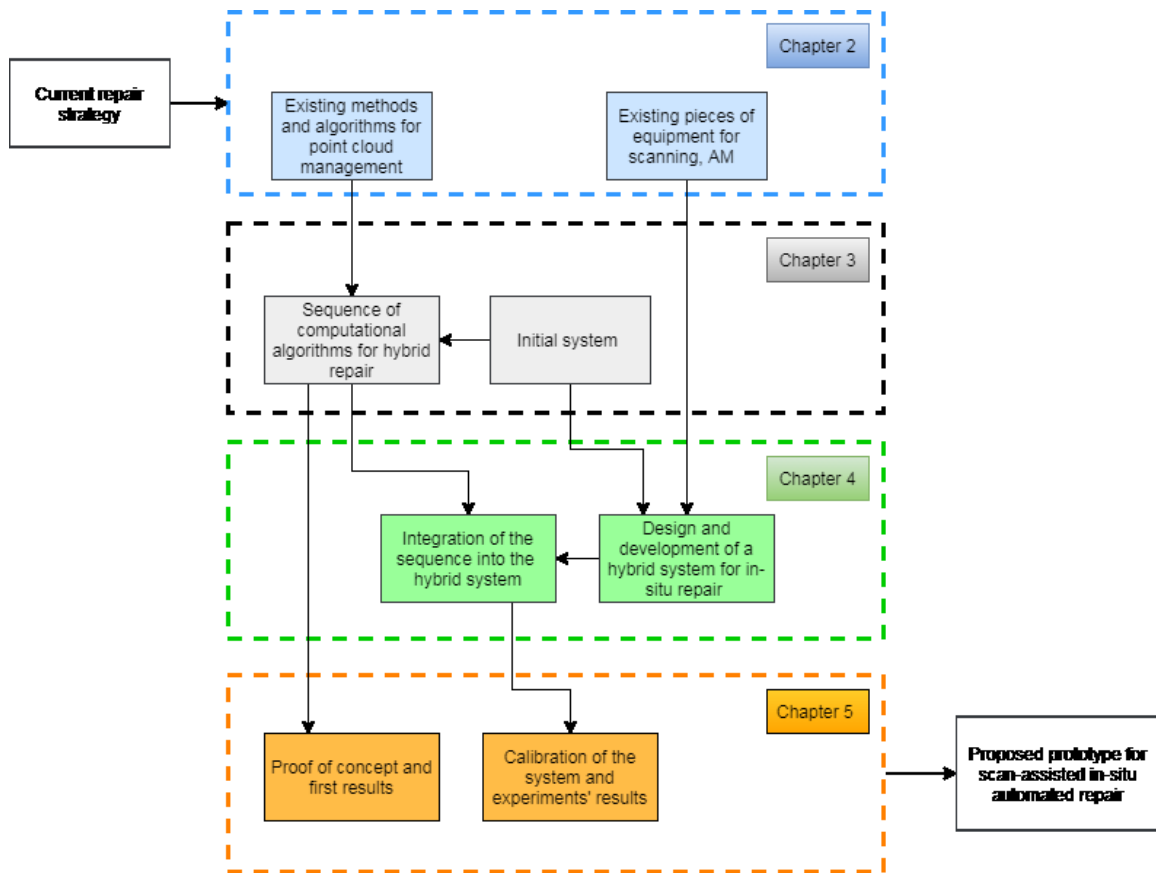


Figure 1.3 Thesis organization

Chapter 2 State of the art

This chapter presents the main challenges that must be handled to design an automated scan-assisted in-situ repair process using hybrid manufacturing. A review investigates what has been done in the past few years about every concept implemented in this process. A detailed analysis presents state-of-the-art scanning technologies, data handling, damage identification and extraction, and hybrid manufacturing.

2.1. Introduction

One of the first attempts of rapid prototyping by additive manufacturing was conducted in 1981. Direct energy deposition (DED) was used for the hard facing of an RB-211 Rolls Royce jet engine [1]. In 1989, S.Scott and L. Crump prototyped a system for extrusion of plastic or metal filament through a nozzle to build 3D objects layer-by-layer using computer-assisted design, subsequently named fused deposition modeling (FDM) [2]. For the last decade, DED has been widely used for research in AM reparation in several fields. In 2011, an analysis of a restoration process using LASER Aided AM onto turbine blade knife edges was published [3]. In 2014, a geometric reconstruction algorithm for repairing gas turbine blades was developed [4]. This study focuses on surface reconstruction by implementing non-uniform rational B-splines (NURBS) surfaces using spline curves. In 2017, Zhang et al. investigated the quality of repair using AM [5]. Micro-structure analysis and mechanical tests have been done to evaluate the characteristics of the material at the interface. Several other studies were conducted during this period and were mainly focused on geometrically complex parts produced for aeronautic, aerospace, or energy production [6]. Nevertheless, repair using AM tackles many challenges, such as developing a general

repair strategy applicable for every manufactured part and the design of a dedicated operative system and pieces of equipment.

2.2. Scanning Technologies

The scanning process and point cloud reconstruction are two of the most crucial points of scan-assisted AM for repair. Noisy scans, sensitive to reflectivity, can result in approximation during the repair process, leading to mechanical properties failures or even divergence between the geometry of the repair part and its corresponding nominal version.

The accuracy and precision of point cloud reconstruction depend on three main axes (see **Table 2.1**):

- The type of scanner chosen for scanning,
- the method used to extract the points corresponding to the object from the environment and,
- the method implemented for the alignment of the scans to rebuild the 3D point cloud.

2.2.1. Contact based 3D scanning

This scanning method used a touching probe to situate the volume in space physically. A coordinate measuring machine (CMM) is an example of contact-based 3D scanning [7]. Every time the probe touches the volume, a point is created. Contact-based 3D scanning does not use any optical instruments and consequently can scan transparent and reflective surfaces. Moreover, this process is not sensitive to light conditions. However, the time of acquisition is significant and mainly depends on the resolution required by the operator.

2.2.2. Photogrammetry

Photogrammetry involves stitching photographs of an object taken from different angles. A simple camera can be used to take the images. Then a software gathers the pictures together by identifying pixels that correspond to the same physical points. The advantage of this method is that it does not require any heavy equipment. The drawbacks are that the results of the scans depend on the resolution of the images. The time required to execute the algorithms can be significant [8].

2.2.3. LASER 3D scanning

The LASER 3D scanner is based on trigonometric triangulation to capture a 3D point cloud of the scene. A point is projected on the surface of the object, and a sensor records the produced reflection. Since the distance between the LASER and the sensor is perfectly known, the computation of the reflection angle of the LASER beam provides accurate point measurement. This technology offers accuracy and precision within a short-range distance. However, the LASER 3D scanner is sensitive to shiny or transparent surfaces [9].

2.2.4. Time-of-flight 3D Scanning

The time-of-flight 3D scanners are also widely available on the market; one was, for example, installed on the new IPAD PRO of APPLE. This technology uses the speed of light to build a 3D representation of space. It computes the shift in time between the emission and the reception of a light signal to compute the distances. This type of scanner provides medium and long scanning range useful for architecture or automated navigation. However, the resulting accuracy and precision are lower than the ones given by a LASER 3D scanner [10].

2.2.5. Structured Light 3D Scanning

Structured light 3D scanning is one of the most used 3D scanning methods in industry and research. The definition given by 3DINSIDER of Structured light 3D scanning is as follows: “a 3D scanning technology that uses a single light source that projects multiple lines on the object, all of which are tracked simultaneously by a camera or multiple cameras”. This aspect contrasts with a laser scanner, which emits multiple laser dots on the object one after the other. The structured light can be white or blue, or laser light, and the pattern consists of a series of parallel stripes [11].

Once the cameras have looked at the shape of the pattern, the point-to-scanner distance is computed to rebuild the 3D model. However, recreating the 3D model of the point cloud requires sophisticated unwrapping algorithms generally directly implemented in the 3D scanning device. The end of the '90s saw improvements in computer and camera technologies. Structured light 3D scanners are more user-friendly and financially accessible for research and commercial firms. This technology has several advantages over another scanning process. Structured light 3D scanning is more accurate than time-of-flight methods and faster than LASER scanning. Where LASER scanning must strike the object multiple times in order to get the 3D model, the cameras used for structured light 3D scanning take all the required information all at once. However, during the scanning process, some challenges must be considered. First, the 3D scanning technology relies on optical sensors, and consequently, any shadows or exterior light source can create errors during the 3D object reconstruction process. In addition, Structured light 3D scanners are sensitive to transparent or reflective materials. They can result in loss of details in the model

or wrong triangularization. One possible solution is to apply a thin opaque lacquer on the surfaces of the object.

Table 2.1. *3D scanning technology advantages and drawbacks*

Technology	Time of acquisition	Resolution	Resilience	Handling
Contact based	High	High	Very High	Low
Photogrammetry	Medium	Low	Low	Very High
Laser	Low	Very High	Low	Low
Time-of-flight	Very Low	Medium	Medium	High
Structured light	Very Low	High	Low	Low

Table 2.1 summarizes the different characteristics of each scanning technology. The repair process requires a fast and accurate scanning strategy over multiple shapes and surfaces. Consequently, a structured light 3D scanner represents the best solution for this purpose.

2.3. Volume identification and 3D Model Reconstruction

Once the scanning technology has been selected, the 3D model reconstruction process can be executed. The data sent by the scanner consists of a matrix of millions of points called point cloud and contains the cartesian coordinates of each point and the corresponding normal and RGB vectors. Effectively, handling this amount of data represents the second main challenge to be tackled. Some strategies have already been implemented to extract an object's point cloud with specific geometry from the rest of the environment's point cloud. Bokhabrine et Al. use Gaussian Image (GI) to register the normal vector of each point inside a unit sphere [12]. Then, a RAdom SAmple Consensus (RANSAC) algorithm is applied to extract the normal vectors corresponding to the points on the object's surfaces. Since only the points with a specific normal vector are selected, the environment and noisy points are omitted. Finally, a constrained Iterative Closest Point (ICP) algorithm was applied to partially overlap the scans and obtain the object's reconstructed point cloud. This

method gives a powerful and robust way to reconstruct a point cloud of an object included inside its environment. However, it can only be applied for specific geometries presenting recognizable patterns inside the Gaussian image.

Machine learning is often used for object detection into point clouds such as cars, ships, human faces, etc. Once the points corresponding to the object have been identified, they can be extracted from the point cloud. J. Huang et Al. implemented a support vector machine (SVM) algorithm for segmentation [13]. The 33-dimensional fast point feature histogram (FPFH) was considered as the descriptor. The algorithm was trained offline over 200k labeled points to classify the point cloud into planes, pipes, edges, thin pipes and applied to the point cloud for clustering. Then, the Flood-Fill algorithm was computed to obtain a list of point sets in each category. This process can identify, and extract points corresponding to a particular object and reduce the noise by choosing only the points close to their neighbors. However, machine learning algorithms such as SVM are efficient in classifying objects only on which they have been trained, limiting the possibilities for larger applications. Moreover, this method can be applied on a scan to extract points on the object's surfaces, but it does not provide a way to reconstruct the object from multiple scans.

A widely employed method for object reconstruction is using fiducial markers as ARUCO, ARTAG, or APRILTAG markers. Once the scan is triggered, several algorithms are run to extract the information from the marker. First, the image is binarized to get rid of the noise or blur. Then the blobs are detected, which enables the marker's localization in the scene. Then a marker rectification process is applied. Finally, the marker information is extracted, giving in the case of object reconstruction, the frame of the marker, and the camera pose

relative to this frame. Using a marker, the reference frame does not change among the scan, which enables the reconstruction of an object from multiple scans under different perspectives [14].

In the case where the transformation between the different scans is known, inverse kinematics can be used for point cloud reconstruction. This method will be presented more precisely in this project. (See 3.2.2)

2.4. Damaged surface Identification and extraction

Once the point cloud of the worn part has been extracted and rebuilt, the damage can be identified. Several approaches have been studied for the last ten years. H. Junjie and L. Liangyu proposed to scan a worn blade using a CCD camera and build the damage volume using a curved surface extension method [15]. The process is described as follows; a 3D measurement of the broken blade is computed using the CCD camera. The data are preprocessed to reduce noise and extract the points corresponding to the blade, and then the 3D geometry is reconstructed. Then, the surface extension using NURBS functions is computed. Finally, the surface intersections and cutting are determined to extract the damage volume. This method is robust and accurate and was even used in other studies. Zheng et Al. used it to identify subtractive and additive repair volumes. However, computing NURBS surfaces from point cloud data with many points can be computationally heavy and time-costly [16].

Zhang et Al. implemented tri-dexel data for damage reconstruction [17]. First, stereolithographic (STL) versions of the worn and nominal part were sliced according to a step ϵ . Then, the nominal and worn model cross-sections were used to align both STL files using the least squared error. The ICP process was carried out to find the rotation matrix

and the translation vector in order to compute the minimum squared error on the distance between the two cross-sections. Finally, Ray casting was implemented to generate tri-dexel data and then extract the volume of damage. This method presents a reconstructed damage accuracy of around 0.1 mm, suitable for DED processes repair purposes. However, to reconstruct STL from a point cloud, the temporal noise of the scanner must be small (around $50\mu\text{m}$) to avoid any deviation during the rebuild process.

Zheng et Al. identify the damage volume by first merging the worn part's point cloud and the nominal STL file using point-to-surface best fitting technology [18]. Then the points of the damage volume were extracted by calculating the distances of the points to the nominal surface. Once the point cloud of the damage volume has been extracted, its geometric model is reconstructed using an octree-based 3D-grid splitting process.

All the previous examples use a nominal model to compare it with the data set of the worn part and eventually extract the damage volume. However, the damaged area can also be detected by directly using point cloud information. Scans contain three types of information, the coordinates x, y, z , the normal vector n_x, n_y, n_z and the RGB vector that can be used for damage detection. Mohammadi et al. studied data sets given by a Lidar scanner [19]. The resulting point cloud was analyzed and segmented between damaged and undamaged areas using three criteria: surface variation-based damage detection, normal vector-based damage detection, and curvature-based damage detection. However, this method requires surfaces with smooth curvatures or planar surfaces to detect local changes in surface variation. Moreover, the technology developed by Mohammadi et Al. is efficient in detecting damages of building or infrastructures' walls. Still, it does not provide a manner to deal with a repair process.

Prominent cross-section (PCS) was also implemented to reconstruct the nominal surfaces from a damaged point cloud. PCS utilizes an iterative approach based on a plane to normal vectors best fitting into a sectional gauss map. The PCS algorithm returns optimal cross-sections. In the case of nominal surfaces, every normal vector associated with the points on the cross-section is tangent to the cross-section. Therefore, the remaining outlier PCS are symptomatic of topological inconsistencies underlining the presence of defects. The resulting sound cross-sections are then fitted with interpolation splines to rebuild the nominal surfaces. Piya et Al. presented in 2011 an implementation of PCS for gas turbine blades reconstruction [20]. The proposed surface reconstruction using interpolation splines represents an efficient way to rebuild continuous or polynomial curves but does not offer the possibility to rebuild complex concave shapes.

2.5. Repair Process

Going forward, once the damage volume has been identified and extracted, the repair process can take place. Some studies directly apply a repair process on the damaged part using AM technologies. It is the case for Zhang et al. in their paper published in 2017; the repair was done using laser powder technology [21]. Bi and Gasser used the same technology coupled with a PID controller to restore a worn turbine blade [3]. Wilson et Al. also proposed a damage extraction using Prominent Cross Section algorithm and repair using LASER cladding technology.

However, in most cases, the damaged part cannot be remanufactured without pre-repair machining. In fact, the geometry of the worn area might make access by the AM system impossible. In other words, the toolpath computation for repair by AM cannot proceed without avoiding collisions between the tool and the worn volume. For damages with tiny

edges or small angles, the worn area needs to be pre-processed by removing material to create smooth grooves or slots in which AM system can evolve. In addition, worn parts suffer significant erosion and corrosion around the damaged area. This contamination must be removed to ensure good mechanical properties at the interface between the worn part and the printed volume. A Computer Numerical Control (CNC) machine with mounted tools such as drills, lathes, and mills, is used to apply the pre-repair process. Zhang et Al. proposed a similar study in 2019 [22]. In the continuity of their article published in 2017, the damaged volume was identified using the same process. Then, a pre-repair process was applied on the worn part to guarantee a repair by AM using Electron Beam Melting. The boundaries of the new slot were computed from two different algorithms named U-shaped boundary (UB) and convex-hull boundary (CHB). These methods have the advantage of being efficient and optimizing the amount of removed material; moreover, it ensures good accessibility to the AM tool. However, the resulted boundaries do not consider the cylindrical shape of the mill used for pre-repair machining, which can lead to incompatibilities between the 3D volume that will be printed and the actual volume that needs to be filled for repair.

2.6. AM Technologies For Repair

Once the worn part has been pre-processed, a repair can be applied using an AM system. According to Rahito et al. [6], the leading DED technologies used for repair for the last ten years are LASER cladding, plasma transferred arc AM and electron beam.

2.6.1. LASER Engineered Net Shaping and LASER Cladding

One of the most common DED AM processes used for repair is LASER engineered net shaping (LENS). This technology involves a nozzle through which metal powder flows

and a laser beam inside an inert gas chamber. The energy provided by the LASER at the focus point induces the melting of the powder, which is deposited layer by layer on the substrate to print the metallic part. G-code controls the motion of the laser source implemented with computed assisted design (CAD) software based on the desired 3D shape [23].

2.6.2. Electron Beam Freeform Fabrication

Another DED process is the electron beam freeform fabrication (EBF). This process uses an electron beam as an energy source and a metal wire for feedstock. EBF provides high accuracy and suitable surface finish for aerospace and Aerospace industries. Parts are 3D printed inside a vacuum space which can limit large-scale applications [24].

2.6.3. Wire Arc Additive Manufacturing

Wire Arc additive manufacturing (WAAM) melts metallic drawn wire using electric arc as a power source to print a near, net-shape component layer by layer. This process can be integrated into a robotic manipulator or multi-axis table and offers the advantages of meter long print with relatively high accuracy and limited material waste [25].

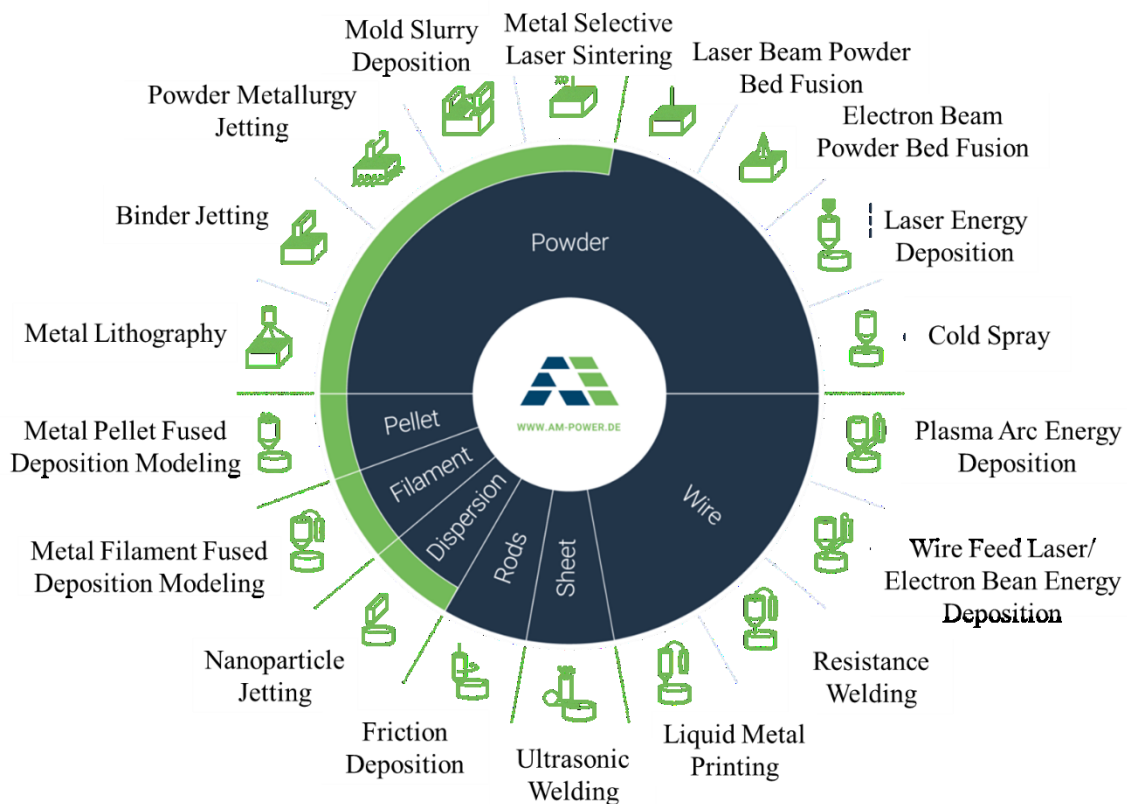
2.6.4. Plasma Transferred Arc

Plasma Transferred Arc (PTA) employs high-temperature inert gas to melt metallic powders that will be deposited layer by layer to build 3-dimensional shapes. The inert gas reaches the plasma state thanks to a high potential electric arc between a tungsten electrode on the torch and the workpiece. Argon is generally used to form the arc plasma and as a shielding gas to prevent the metal pool from atmospheric contamination.

The PTA system relies on electrical power for the control of the plasma arc. By changing the electric current, the flow of electrons is modified, which modulates the number of

ionized atoms inside the gas and the temperature. In the PTA-AM system, metal powder is deposited from a powder hopper through a torch that contains the electrode and the plasma arc. The powders are melted as they are transported within the plasma arc and deposited onto the deposit's surface. The deposited metal solidifies as the torch moves over the surface, leaving a metal bead behind the moving arc.

PTA as an AM process for repair offers multiple advantages over other AM processes. The PTA-AM process can print parts with a homogeneous gradation of different materials, which is impossible in other laser and electron beam AM processes. In addition, the PTA-AM can use an extensive range of powder distribution, around 45-150 μm . The anticipated different microstructures of different powder sizes are not detrimental to the PTA-AM process since powders are fully melted [26].



<https://www.aativemanufacturing.meia/articles/infographic-aative-family-tree>

2.7. Literature Review Summary

Repair using AM has been gaining attention in high technology industries for complex part remanufacture over the last decades. This recent interest can be explained by the technological improvements of the devices used for automated repair as 3D scanners, DED processes (see **Figure 2.1**), and CNC machines.

In the last decade, research and development for the hybrid repair of manufactured parts were mainly focused on aerospace and spatial industries. The requirements in terms of precision and time permitted developing several strategies for repairing high-value parts. In many cases, the technologies used for this purpose can only be applied to specific damage types or even part geometry. Moreover, the repair strategy generally includes multistep, multisystem processes costing time and money. Even if the recent development of AM offers multiple advantages compared to the traditional repair process, industries suffer a lack of standard strategy for automated repair.

This project aims to develop a standardized repair strategy using hybrid manufacturing. First, a sequence of computational algorithms is implemented to compute a feasible hybrid repair strategy. Then a prototype of hybrid system is developed to ensure automated scan-assisted in-situ repair. The experiments conducted with the standardized repair strategy and hybrid system are presented at the end of this thesis.

Chapter 3 Initial Setup and Software Development

This chapter starts by presenting the initial setup used as a basis for developing the hybrid repair system. Then, the equipment selection is depicted regarding the requirements of the hybrid repair process. Later, Chapter 3 details a sequence of algorithms implemented for automated in-situ repair, from the scanning process to the computation of the pre-repaired and to-be-printed volumes. Every algorithm reported in this chapter is described with mathematical concepts and pseudo-codes.

3.1. Experimental Setup Design

This section presents the setup on which the experiments will be conducted and the modification that must be carried out to adapt the initial system for scan-assisted in situ automated repair using hybrid manufacturing.

3.1.1. Initial setup

The initial state of the system is presented **Figure 3.2**. It is a three-axis CNC machine with a control box and a PC. The cartesian motion is ensured by five linear actuators, motorized with 6A stepper motors and 24V brakes. Two pairs of two parallel actuators enable the motion into x and z directions. The motion along the y-axis is permitted using one horizontal slider. The five stepper motors are controlled using two stepper driver embedded boards. An embedded controller connected between the main PC and the drivers enables the control of the CNC machine, using a joystick or G-code commands. All the components are included in a central unit. The design of the central unit is presented **Figure 3.1**.

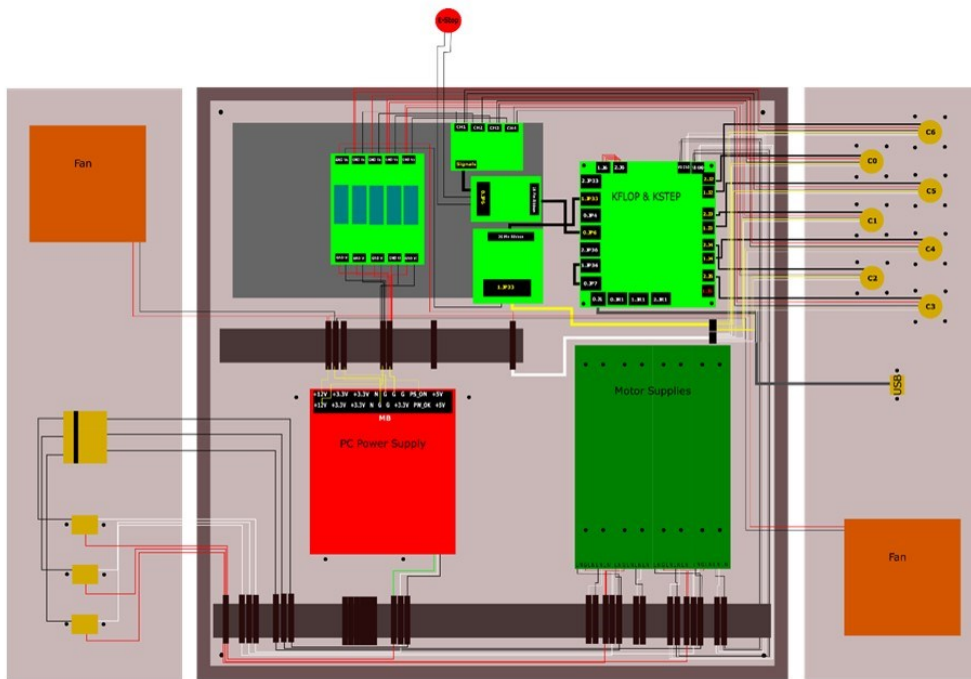


Figure 3.1 Design of the central unit

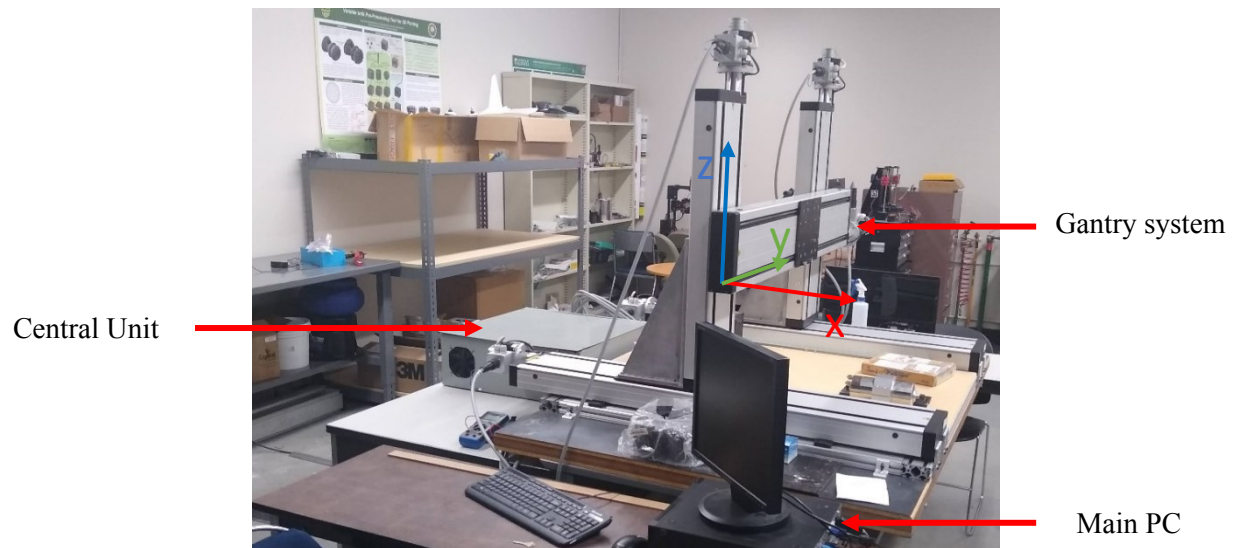


Figure 3.2 Initial setup and equipment

3.1.2. Upgrades

Several additions and upgrades were made to the software and the hardware parts to comply with the requirements of repair using hybrid manufacturing. For the metrology presented in **Figure 1.2**, the following tools were added:

- ✓ A structured light 3D scanner is chosen for acquisition time, accuracy, and repeatability as computer vision equipment. The 3D scanner can scan up to 3.2 million points over a projected surface of 590x421mm. The scanner is synchronized with a two-axis rotary table on which the damaged part is attached during the repair process. The degrees of freedom offered by the rotary table enable scanning the damaged part under different perspectives and getting details around the damaged area. The rotary table is motorized with two 3.8A stepper motors that rotate around the pitch and yaw axis.
- ✓ For surface preparation, a 1.5KW high torque AC spindle is mounted on the CNC system. Aluminum and HSS end mills can be fixed on the spindle's head for carving, drilling, and pocketing. The spindle is water-cooled using an immersed pump.
- ✓ A PTA system is chosen for repair by AM. The PTA process includes the use of a high voltage source and inert gas. Therefore, a specific power supply and torch are required. The Eutronic GAP 3000 power supply provides high electric potential between the torch and the plate, situated on the top of the rotary table.

In terms of software, a sequence of computational algorithms must be implemented to ensure automated in-situ repair. This sequence must communicate with the pieces of

equipment detailed above and provide the operator with a robust and accurate repair strategy.

3.2. Scanning process

This section details the process of point cloud reconstruction from multiple perspective scans. Two main methods have been investigated; the first one presents a process for point cloud reconstruction using marker recognition. The second one uses inverse kinematics to rebuild the surfaces of the scanned volumes.

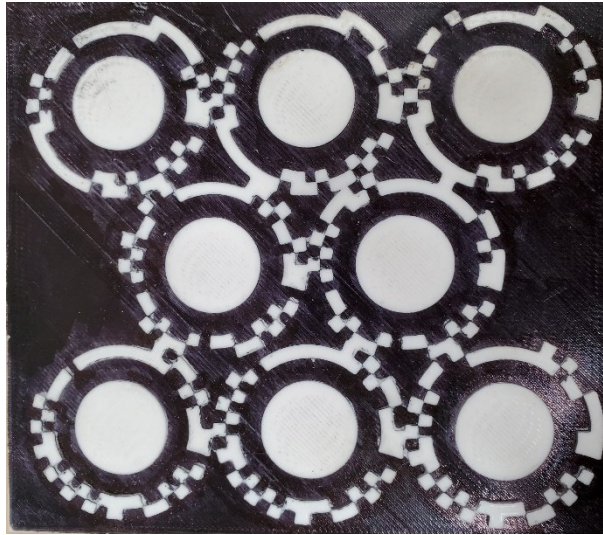


Figure 3.3. *3D printed marker pattern*

3.2.1. Marker recognition

In configuration 1, the scanning process includes three main components: the structured light 3D scanner, the marker pattern (see **Figure 3.3**), and the 2-axis rotary table. Structured light 3D scanning technology is fast and accurate while the object's surfaces are not significantly reflective or transparent. Consequently, the worn sample is painted with white non-reflective paint to avoid any issues during the scanning process. Then the sample is placed between printed carbon fiber chucks on the rotary table. The marker pattern is a

type of Fiducial marker and will be used to keep the same reference frame during the scanning process [27]. This marker is attached to the rotary table and moves with it.

The scanning sequence can be defined by three main configurations, initial, ongoing, and final configurations. At the initial configuration, the reference frame given by the marker pattern has its three axes aligned with the axis of the CNC machine. The scanner and the rotary table are both controlled by KFLOP, and G-code sequences [28]. The G-code sequence controls the scanner's trigger, the motion of the rotary table and enables an automated scanning process. During the ongoing configuration, the rotary table first rotates along a pitch direction. Then the scanner is triggered, and a scan is saved in a specific folder. Then, the table rotates in the yaw direction to offer a new perspective to the scanner. The scanner is triggered once again. These last two steps are repeated to capture the worn volume and the damaged area as accurately as possible. Eventually, the table comes back into a final configuration by realigning the axis given by the marker pattern with the ones

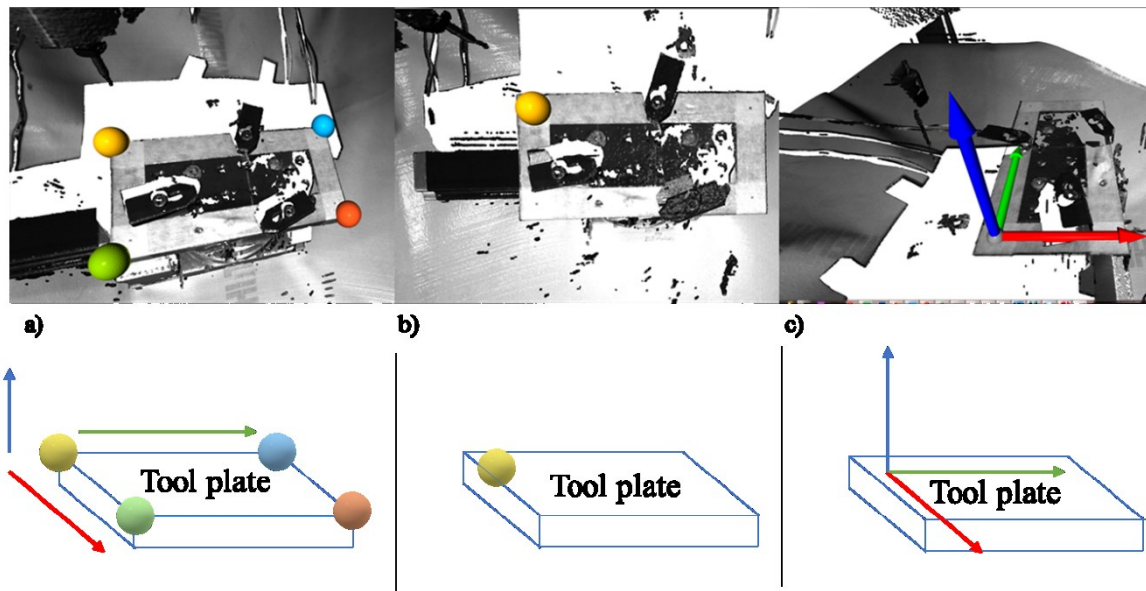


Figure 3.4 Calibration protocol applied on the rotary table. a) Set up of the axes aligned with the system. b) Set up of the origin of the frame \mathbb{F}_0 . c) Results of the calibration protocol.

of the CNC machine. The marker was 3D printed using PLA and a commercial 3D printer to avoid any possible deterioration during the repair process.

3.2.2. Inverse kinematics

In some cases, it has been observed that the sequence for point cloud reconstruction failed because the 3D scanner was not recognizing the marker pattern. Consequently, inverse kinematics can be implemented. Dr. Donald Lee Pieper defined the inverse kinematics methodology in 1968: given a desired hand position and orientation along with the various link parameters defining the structure; what are the values of the manipulator variables that place the hand at the desired position with the desired orientation? [29] From this questioning, a mathematical model can be built using matrix structures defined in special linear Lie algebra called $\mathbb{SE}(3)$ (with the Lie bracket as an operation) [30]. In this project, the transformation matrices defined in $\mathbb{SE}(3)$ provide a linear link between the position of the damaged part in the rotary table's frame and the 3D scanner's frame. This linear relation is used for point cloud reconstruction.

A first reference frame aligned with the axis of the Gantry system is defined. This frame called \mathbb{F}_0 , is used as a reference for surface preparation and repair, the origin of \mathbb{F}_0 is placed at one of the corners of the insulation plate when this one is in the initial configuration (see **Figure 3.4**). A second frame called \mathbb{F}_r is also defined, the orientation and the origin of \mathbb{F}_r are a function of α and β , the angles of rotation in Yaw and Pitch directions of the rotary table. \mathbb{F}_r and \mathbb{F}_0 are overlapped when the rotary table is in the initial configuration. Finally, a third frame is defined as \mathbb{F}_s corresponds to the frame of the 3D scanner. Solving the inverse kinematic problem is then equivalent to computing the transformation matrix

between \mathbb{F}_r and \mathbb{F}_s via \mathbb{F}_0 . In other words, the homogeneous transformation would be defined as:

$$\mathbf{T}_s = \mathbf{T}_0 \cdot \mathbf{T}_{\text{alpha}} \cdot \mathbf{T}_{\text{beta}} \quad (3.1)$$

With
$$\mathbf{T}_{\text{alpha}} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & x_\alpha \\ \sin(\alpha) & \cos(\alpha) & 0 & y_\alpha \\ 0 & 0 & 1 & z_\alpha \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{T}_{\text{beta}} =$$

$$\begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) & x_\beta \\ 0 & 1 & 0 & y_\beta \\ -\sin(\beta) & 0 & \cos(\beta) & z_\beta \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } \mathbf{T}_0 = \begin{pmatrix} r_{11} & r_{12} & r_{13} & x_0 \\ r_{21} & r_{22} & r_{23} & y_0 \\ r_{31} & r_{32} & r_{33} & z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

To find the transformation matrix between \mathbb{F}_s and \mathbb{F}_0 , the rotation matrix and the translation vector must be computed. To do so, a first scan is triggered, the point cloud is then given into the 3D scanner's coordinate system, \mathbb{F}_s . A first point is selected as the origin of \mathbb{F}_0 into \mathbb{F}_s . The coordinates of the point give the translation vector $t_0 = \begin{pmatrix} -x_0 \\ -y_0 \\ -z_0 \end{pmatrix}$ between \mathbb{F}_s and \mathbb{F}_0 .

To compute the rotation matrix \mathbf{R}_0 , four points at the four corners of the insulated plate placed at the top of the rotary table are selected. An oriented bounding box is computed from these four points. The two first edges of the bounding box, oriented in x and y directions relative to the gantry's frame, are selected as vectors to build \mathbb{F}_0 . By using a bounding box, the orthogonality of \mathbb{F}_0 is guaranteed. To compute \mathbf{R}_0 , a first rotation matrix \mathbf{R}_{yz} between the two normalized vectors aligned with x-axis in, respectively, \mathbb{F}_0 and \mathbb{F}_s is built. Given the two vectors x_0 and x_s , the rotation matrix is given by the Euler-Rodrigues formula as:

$$(3.2)$$

$$\mathbf{R}_{zy} = \mathbf{I} + [\mathbf{x}_0 \times \mathbf{x}_s] + [\mathbf{x}_0 \times \mathbf{x}_s]^2 \frac{1}{1 + \mathbf{x}_0 \cdot \mathbf{x}_s}$$

With $[\mathbf{x}_0 \times \mathbf{x}_s]$ defined as the skew-symmetric cross-product matrix of $\mathbf{x}_0 \times \mathbf{x}_s$. From this step, $\mathbf{R}_{zy}\mathbf{x}_s = \mathbf{x}_0$. Eventually, to find \mathbf{R}_0 , a simple translation around \mathbf{x}_0 axis must be applied to get $\mathbf{R}_0\mathbb{F}_s = \mathbb{F}_0$.

This rotation is given as $\mathbf{R}_{x_0} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{pmatrix}$ with $\gamma = \mathbf{R}_{yz}\mathbf{y}_s \cdot \mathbf{y}_0$.

Consequently $\mathbf{R}_0 = \mathbf{R}_{yz}\mathbf{R}_{x_0}$ and $\mathbf{T}_0 = \begin{pmatrix} \mathbf{R}_0 & t_0 \\ 0 & 1 \end{pmatrix}$.

The transformation matrix between \mathbb{F}_0 and \mathbb{F}_r is given as $\mathbf{T}_r = \begin{pmatrix} \mathbf{R}_\beta & t_\beta \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R}_\alpha & t_\alpha \\ 0 & 1 \end{pmatrix}$ with

$\mathbf{R}_\alpha = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$ and $\mathbf{R}_\beta = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix}$. To find the

translation vectors t_α and t_β expressed in \mathbb{F}_0 a triangulation process is used.

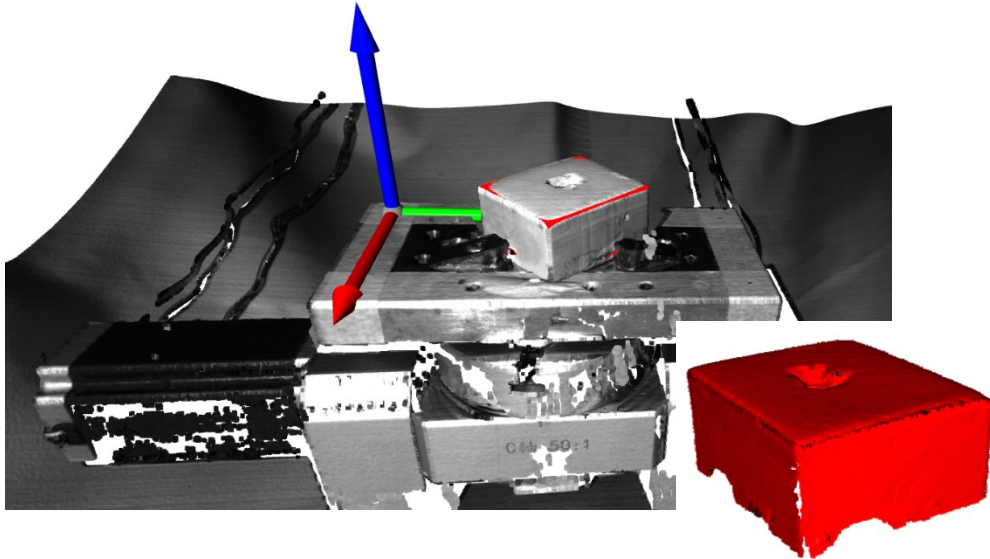


Figure 3.5 Results of the inverse kinematics process for point cloud reconstruction. The red rebuilt point cloud is overlapped with a scan of the rotary table

In the case of t_α , the insulated plate is scanned under three different values of angle α . For each configuration, the coordinates of the origin of \mathbb{F}_γ in \mathbb{F}_0 are saved. Three points O, A, and B are saved for $\alpha \in \{0^\circ, 45^\circ, 135^\circ\}$. Because the rotation is applied around the z-direction relative to \mathbb{F}_0 , all these three points are at a δ distance from the point defining the intersection between the x, y plane and the axis of rotation of α . Consequently, the point (x_α, y_α) is also at a δ distance from the three points. By taking the equation of a circle, the following equations can be written:

$$\begin{cases} x_\alpha^2 + y_\alpha^2 = \delta^2 \\ (x_A - x_\alpha)^2 + (y_A - y_\alpha)^2 = \delta^2 \\ (x_B - x_\alpha)^2 + (y_B - y_\alpha)^2 = \delta^2 \end{cases} \Leftrightarrow \begin{cases} x_\alpha = \frac{y_B(x_A^2 + y_A^2) - y_A(x_B^2 + y_B^2)}{2(x_A y_B - x_B y_A)} \\ y_\alpha = \frac{x_B(x_A^2 + y_A^2) - x_A(x_B^2 + y_B^2)}{2(x_B y_A - x_A y_B)} \end{cases} \quad (3.3)$$

Consequently $t_\alpha = \begin{pmatrix} x_\alpha \\ y_\alpha \\ 0 \end{pmatrix}$. In the same way, $t_\beta = \begin{pmatrix} x_\beta \\ 0 \\ z_\beta \end{pmatrix}$ can be defined.

For point cloud reconstruction, multiple scans are triggered with specific values of α and β , then the inverse of the homogeneous transformation $\mathbf{T}_s(\alpha, \beta)$ is applied to each point cloud for reconstruction (see **Figure 3.5**).

3.3. 3D Model Reconstruction

Once the scanning process has been executed, multiple scans have been saved in a polygon file format called PLY format. Each file contains more than three million points, also called vertices with their coordinates x, y, z in space, their corresponding normal vector

coordinates n_x, n_y, n_z and their RGB vector. However, only a small proportion of these points is interesting. Only the points captured on the surface of the worn part must be considered. The rest of the environment must be omitted. The point cloud is cropped based on each point's geometry and color to extract the interesting points efficiently. In fact, regarding the origin of the frame \mathbb{F}_0 , it is possible to crop approximately the point cloud around the worn part. The preserved space is then given as a cube with dimensions chosen by the operator. The point cloud resulting from the crop process is constituted by points on the captured surfaces of the worn part and the chucks. Moreover, some acquisitions can be considered as wrong measures due to possible reflections during the scanning process. These points have the characteristics of being without close neighbors and have an inaccurate normal vector (poor confidence map).

The RGB vectors are used as criteria for classification to remove the points corresponding to the chucks. The worn part is painted with white non-reflective paint. In addition, the chucks were printed with black carbon fiber PLA. Consequently, the difference of Euclidean norm between the RGB vectors of points on the chucks' surfaces and points on the worn part's surfaces allows the chucks to be identified and removed from the point cloud. This method enables for removing a lot of points corresponding to the chucks quickly.

A flood fill algorithm is implemented to remove wrong points coming from reflection and reduce the average noise,

3.3.1. Flood fill Algorithm

Let consider \mathbf{D} the point cloud obtained after removing the chucks' points, \mathbf{D} is then defined as $\mathbf{D} = \{\mathbf{d}_i, 0 \leq i \leq N\}$. An initial point p_s is randomly chosen with the condition that the

number of neighbors of this point approximatively corresponds to the density of the point given by the scanner. This method enables the selection of a point on the surface of the worn part, avoiding any possible outliers.

The flood fill algorithm presented in **Table 3.2** starts from one seed point and checks its closest neighborhood under a distance δ . All the neighbors inside the neighborhood are then added to the seed points set. Then, for each new point added to the neighborhood, the new neighborhood of this point is computed. This algorithm can be considered a propagation algorithm that removes the noisy and wrong points from the scan.

The point cloud \mathbf{D} is organized according to a k-d tree to increase the efficiency of the flood fill algorithm (propagation algorithm).

Table 3.2: Flood Fill algorithm

Inputs: Point Cloud \mathbf{D} , initial point d_s

Procedure

$\delta \leftarrow$ temporal noise of the scanner

$\mathbf{S} \leftarrow \{ \mathbf{d}_i \in \mathbf{D}, 1 \leq i \leq N, \|\mathbf{d}_i - \mathbf{d}_s\|_2 \leq \delta \}$

$j \leftarrow 0$

while $j < \text{length}(\mathbf{S})$

$\mathbf{S} \supset \{ \mathbf{d}_i \in \mathbf{D}, 1 \leq i \leq N, \|\mathbf{d}_i - \mathbf{d}_j\|_2 \leq \delta \}$

$j \leftarrow j + 1$

End Procedure

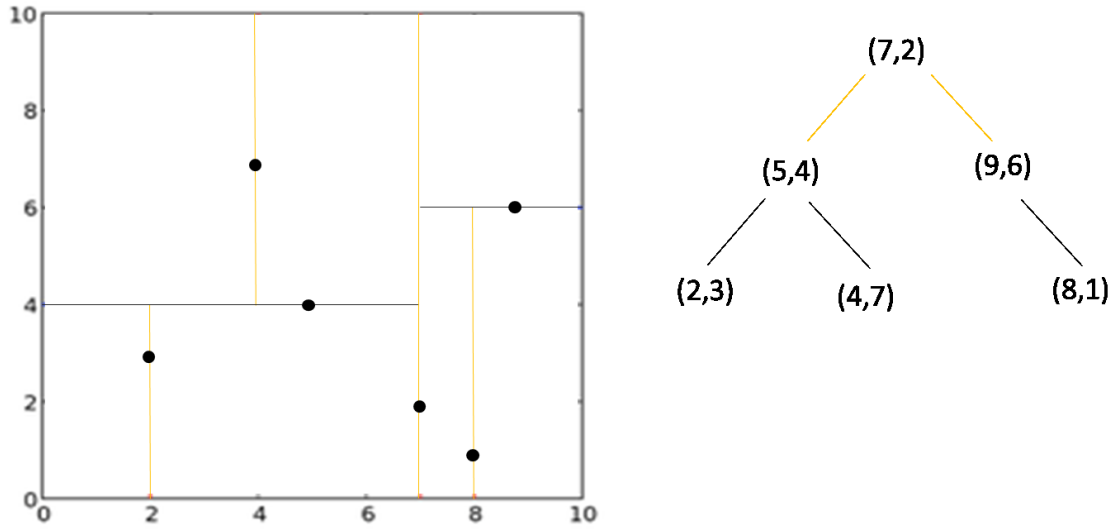


Figure 3.6. 2D tree organization and associated graph

3.3.2. K-D Tree

The k-d tree is a space-partitioning data structure for organizing points in k-dimensional space (see **Figure 3.6**) [31]. Thanks to a k-d tree organization, the complexity to find the nearest neighbor of a point is $n \log(n)$ whereas the complexity for a naïve binary search is n^2 .

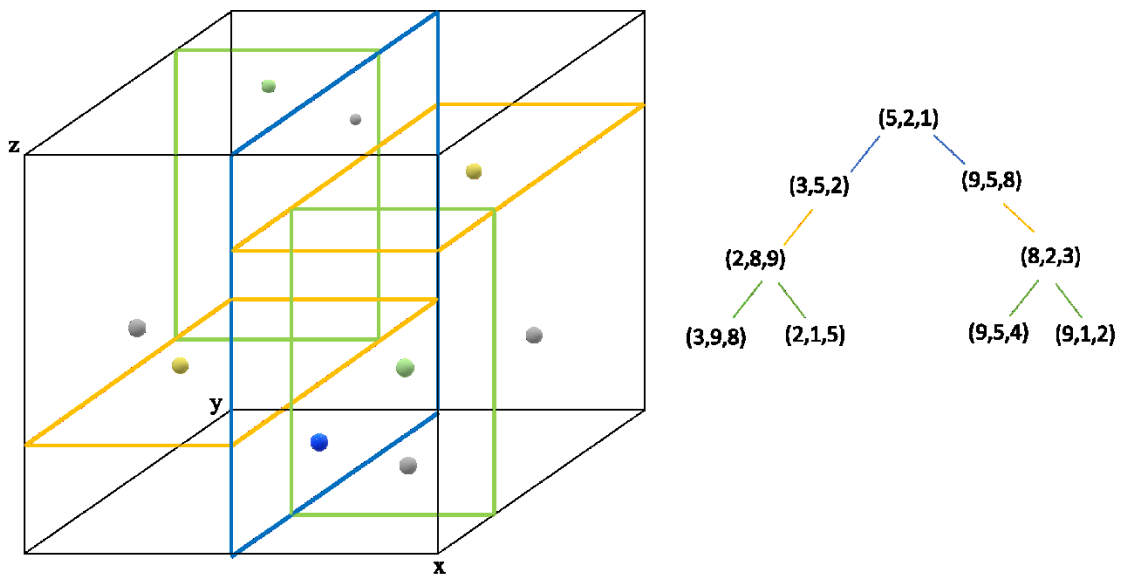


Figure 3.7. 3D tree organization and associated graph

In this project, a 3-d tree is used to partition the 3D point cloud. It consists of a binary tree in which every leaf node is a 3D coordinates vector (see **Figure 3.7**). Every non-leaf node is a hyperplane generator that divides the space into half-spaces. Consequently, points to the left of the hyperplane create a left subtree of that node, and equivalently, the points to the right of the hyperplane are represented by the right subtree. Here, each hyperplane is a 2-D plane. Different methods can be implemented to build each level of the tree and so each hyperplane. The commonly used one is the canonical method, which means that in 3 dimensions, every plane is alternatively x, y, z aligned.

The nearest neighbor search algorithm aims to find the point in the tree closest to a given input point. The following protocol details a manner to find the nearest neighbor of a point in a k-d tree:

Step 1: the point goes through the tree, layer by layer, until it reaches the bottom of the tree. For each generation, the distance between the node and the point is computed. If the newly computed distance is smaller than the previous one, the current best distance and the best candidate are updated.

Step 2: regarding step 1, it is likely, that a close point has been omitted. Consequently, the algorithm goes backward in the tree. The program computes the distance between the point and the splitting hyperplane created at the nodes for each node. If the resulting distance is smaller than the current best distance, the algorithm checks for potential better candidates on the other side of the node. If not, the algorithm continues to go backward. The loop terminates when the algorithm reaches the root node.

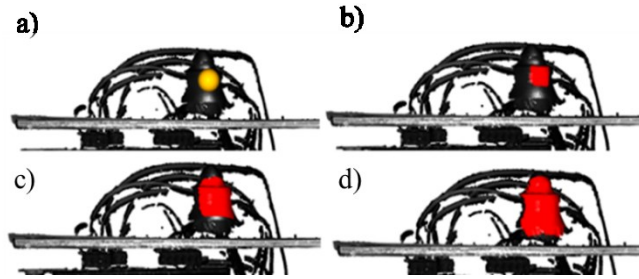


Figure 3.8. Flood fill algorithm process. a) Initial point selection. b) Volume identification 10%. c) Volume identification 40%. d) Volume identification 90%.



Figure 3.9. Result of the flood fill algorithm

The initial step combining crop and flood fill algorithms provides a filtered version of the point cloud **D** corresponding uniquely to the points captured on the surface of the scanned object by omitting the environment and the reflection points (see **Figure 3.8** and **Figure 3.9**). This process is run over all the scans. In addition, inverse kinematics enables to keep the same frame among the different scan perspectives. Eventually, by combining these two steps, a point cloud of the worn part is rebuilt.

3.4. Damage Volume Identification

Now that the 3D model of the worn part has been computed, the damaged area can be identified. To do so, a point cloud of the nominal part is used. Both nominal and damaged volumes are overlapped to identify the damaged area. The nominal point cloud is computed from the STL version of the nominal part using Poisson disk Sampling. Contrary to random

sampling, using Poisson disk sampling enables to keep the same density of points on all the object's surfaces.

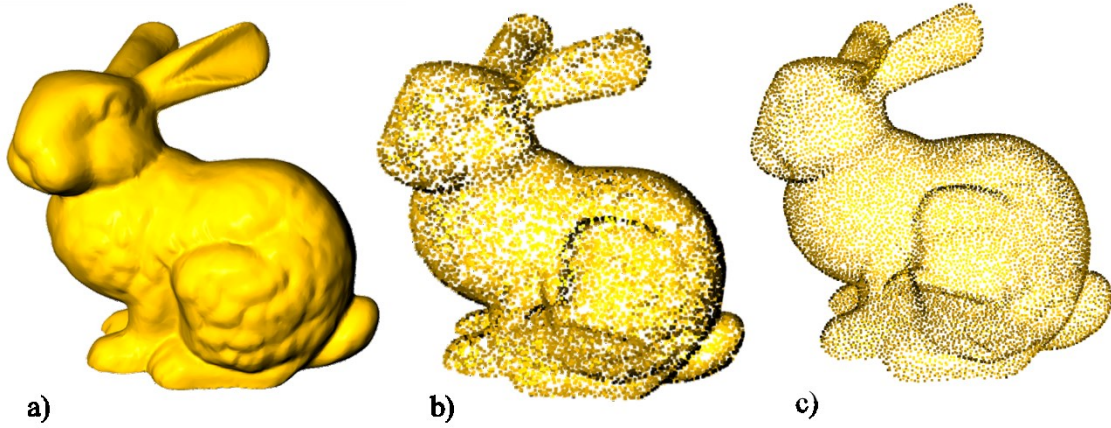


Figure 3.10. *Point Sampling. a) STL version of the Stanford rabbit. b) Random sampling (10000 points). c) Poisson disk sampling (10000 points).*

3.4.1. Sample Elimination for Generating Poisson Disk Sample Sets

The Uksel method used to compute Poisson disk Sampling is implemented into the Open3D library largely used to write the computational algorithms sequence [32]. The Uksel method starts from random samples uniformly distributed between the triangle surfaces and returns a subset of samples with uniform density (see **Figure 3.10**). For each point p_i into the initial set, a neighborhood of a certain radius is computed. This radius r_{max} depends on the volume of the sampling domain V and the number of points inside the subset N . In \mathbb{R}^3 , r_{max} is given as:

$$r_{max} = \sqrt[3]{\frac{V}{4\sqrt{2}N}} \quad (3.4)$$

The expression of r_{max} comes from the hypothesis that the volume of the sampling domain can be filled with a sphere of radius r_{max} without intersection. Then, the contributions from all the points p_j inside this neighborhood of p_i are computed as follows:

$$\omega_{ij} = \left(1 - \frac{\min(d_{ij}, r_{max})}{2r_{max}}\right)^\alpha$$

With d_{ij} the distance between p_i and p_j , α a constant. A weight defined as the sum of all the contributions is assigned to p_i . Once every point is weighted, the point with the highest weight is removed from the set. The weights are once again adjusted. This process is repeated until the subset reaches the requested number of points, here N .

The number N of points is chosen to approximatively get the same density of points as the point density of the 3D model of the worn part. Having the same density of points between the damaged and nominal point clouds significantly increases the probability of having tuples of corresponding points from both point clouds, which improves the efficiency of the iterative closest point algorithm (used to overlap the two point clouds). Let us call the two new point clouds, respectively **I** and **D**, which stand for Ideal and Damaged point clouds.

To identify the damaged surfaces from **D**, in other words, the points corresponding to the worn area, **I** and **D** are overlapped using the Iterative Closest Point algorithm (ICP).

3.4.2. Iterative Closest Point Algorithm

The ICP algorithm is an iterative approach that tries to converge to the optimal rotation matrix and translation vector, minimizing the following least square error between the two point clouds **I** and **D** [33]:

$$E(\mathbf{R}, \mathbf{t}) = \frac{1}{N} \sum_{k=1}^N \|d_k - \mathbf{R}i_k - \mathbf{t}\|^2 \quad (3.6)$$

Where \mathbf{R} is a rotation matrix in $\mathbb{S}\mathbb{O}(3)$, \mathbf{t} is a translation vector in \mathbb{R}^3 , N is the number of randomly selected points in \mathbf{I} and \mathbf{D} defined as $\max_{k \in N}(N_I, N_D)$, i_k the selected points in \mathbf{I} and d_k the selected points in \mathbf{D} . To compute \mathbf{R} and \mathbf{t} , multiple steps are required.

Center of mass

The centers of mass of \mathbf{I} and \mathbf{D} are defined as:

$$\mu_I = \frac{1}{N} \sum_{k=1}^N i_k \quad \text{and} \quad \mu_D = \frac{1}{N} \sum_{k=1}^N d_k$$

Two new point clouds \mathbf{I}' and \mathbf{D}' are computed from the given translations, $\mathbf{I}' = \{i_k - \mu_I, k \in N\} = \{i'_k\}$ and $\mathbf{D}' = \{d_k - \mu_D, k \in N\} = \{d'_k\}$

Singular Value Decomposition (SVD)

Let $\mathbf{W} = \sum_{k=1}^N i'_k \cdot d'_k{}^T$. Then the SVD of \mathbf{W} is given as:

$$\mathbf{W} = \mathbf{U} \begin{bmatrix} \sigma_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sigma_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sigma_3 \end{bmatrix} \mathbf{V}^T \quad (3.7)$$

Where \mathbf{U} and \mathbf{V} are in $\mathbb{S}\mathbb{O}(3)$ and $\sigma_1 \geq \sigma_2 \geq \sigma_3$ are the singular values of \mathbf{W} . Then, if $\text{rank}(\mathbf{W}) = 3$, the optimal solution of $E(\mathbf{R}, \mathbf{t})$ is unique and is given by:

$$\mathbf{R} = \mathbf{V}\mathbf{U}^T \quad \text{and} \quad \mathbf{t} = \mu_D - \mathbf{R}\mu_I$$

Proof

Giving $i = \{i_1, \dots, i_N\}$ and $d = \{d_1, \dots, d_N\}$ two sets of 3D points. The goal is to minimize the quantity $E(\mathbf{R}, \mathbf{t}) = \frac{1}{N} \sum_{k=1}^N \|d_k - \mathbf{R}i_k - \mathbf{t}\|^2$.

One assumes \mathbf{R} is fixed and solves the problem for \mathbf{t} :

$$\frac{\partial E}{\partial t} = 0 \Leftrightarrow \sum_{k=1}^N (\mathbf{R}i_k + t - d_k) = 0 \Leftrightarrow \mathbf{R} \sum_{k=1}^N i_k - \sum_{k=1}^N d_k + Nt = 0 \Leftrightarrow t = \mu_D - \mathbf{R}\mu_I$$

Now, t can be substituted in the expression:

$$\begin{aligned} E(\mathbf{R}, t) &= \frac{1}{N} \sum_{k=1}^N \|d_k - \mathbf{R}i_k - \mu_D + \mathbf{R}\mu_I\|^2 = \frac{1}{N} \sum_{k=1}^N \|d_k - \mu_D - \mathbf{R}(i_k - \mu_I)\|^2 \\ &= \frac{1}{N} \sum_{k=1}^N \|\mathbf{R}i'_k - \mathbf{d}'_k\|^2 \end{aligned}$$

Then,

$$\|\mathbf{R}i'_k - \mathbf{d}'_k\|^2 = i'_k{}^T i'_k + \mathbf{d}'_k{}^T \mathbf{d}'_k - 2\mathbf{d}'_k{}^T \mathbf{R}i'_k$$

Consequently, minimizing E is equivalent to find \mathbf{R} as $\mathbf{R} = \operatorname{argmax}_{\mathbf{R} \in \mathbb{S}\mathbb{O}_3} \sum_{k=1}^N \mathbf{d}'_k{}^T \mathbf{R}i'_k$

However, it can be noted that $\sum_{k=1}^N \mathbf{d}'_k{}^T \mathbf{R}i'_k = \begin{bmatrix} \mathbf{d}'_1 \\ \vdots \\ \mathbf{d}'_N \end{bmatrix} \mathbf{R} [i'_1 \quad \dots \quad i'_N] = \operatorname{tr}(\mathbf{D}'^T \mathbf{R} \mathbf{I}')$

Considering the properties of the trace, $\operatorname{tr}(\mathbf{D}'^T \mathbf{R} \mathbf{I}') = \operatorname{tr}(\mathbf{R} \mathbf{I}' \mathbf{D}'^T)$, then by taking the covariance matrix $\mathbf{S} = \mathbf{I}' \mathbf{D}'^T$ and computing the SVD of $\mathbf{S} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$, one gets:

$$\operatorname{tr}(\mathbf{R} \mathbf{I}' \mathbf{D}'^T) = \operatorname{tr}(\mathbf{R} \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T) = \operatorname{tr}(\mathbf{\Sigma} \mathbf{R} \mathbf{U} \mathbf{V}^T)$$

However, by defining \mathbf{B} as $\mathbf{\Sigma} = \mathbf{B}^2$, $\mathbf{\Sigma}$ is a diagonal square matrix, one has:

$$\operatorname{tr}(\mathbf{R} \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T) = \operatorname{tr}(\mathbf{R} \mathbf{U} \mathbf{B}^2 \mathbf{V}^T) = \operatorname{tr}((\mathbf{R} \mathbf{U} \mathbf{B})(\mathbf{V} \mathbf{B})^T) = \langle \mathbf{R} \mathbf{U} \mathbf{B}, \mathbf{V} \mathbf{B} \rangle$$

Using the inequality of Cauchy-Schwarz, $\operatorname{tr}(\mathbf{R} \mathbf{S}) \leq \|\mathbf{R} \mathbf{U} \mathbf{B}\|_2 \|\mathbf{V} \mathbf{B}\|_2 \stackrel{\perp}{=} \|\mathbf{B}\|_2^2 =$

$$\operatorname{tr}(\mathbf{B} \mathbf{B}^T) = \operatorname{tr}(\mathbf{\Sigma})$$

Consequently, $\text{tr}(\Sigma \mathbf{R} \mathbf{U} \mathbf{V}^T) \leq \text{tr}(\Sigma)$, therefore, to maximize $\text{tr}(\mathbf{D}'^T \mathbf{R} \mathbf{I}')$, $\mathbf{R} \mathbf{U} \mathbf{V}^T$ must be equal to the matrix identity. This is equivalent to say $\mathbf{R} = \mathbf{V} \mathbf{U}^T$.

This theorem can be applied if each tuple (i_k, d_k) is perfectly known. However, the tuples are unknown in the earlier situation due to non-similarities between the two point clouds. That is why an iterative approach was chosen. For each iteration, the tuples (i_k, d_k) are built from points in \mathbf{D} and their nearest neighbors in \mathbf{I} , then a new transformation is computed using SVD and is applied to \mathbf{I} . Other technics as normal shooting or point-to-plane error metrics can be used to build the tuples. The two last strategies are efficient for smooth structures and simple geometries but remain computationally heavy. In this project, the first strategy of closest point matching was chosen for two main reasons; it is suitable for complex geometry and optimizable using KD-tree search.

Table 3.3: ICP Algorithm

Inputs: Point Cloud \mathbf{I}' , \mathbf{D}' , initial

Transformation \mathbf{T}_0

Output: Transformation \mathbf{T}

Procedure

$\mathbf{I}' \leftarrow \mathbf{T}_0 \mathbf{I}'$

Compute \mathbf{R} , t via SVD of $\mathbf{I}' \mathbf{D}'^T$

$\mathbf{T} \leftarrow \Phi(\mathbf{R}, t)$

while $E(\mathbf{R}, t)$ decreased and $E(\mathbf{R}, t) > \epsilon$

do

$\mathbf{I}' \leftarrow \mathbf{T} \mathbf{I}'$

$\mathbf{D}'_{tree} \leftarrow \mathbf{3D tree of D}'$

Find corresponding points in \mathbf{D}'_{tree}

Compute \mathbf{R} , t via SVD of $\mathbf{I}' \mathbf{D}'^T$

$\mathbf{T} \leftarrow \Phi(\mathbf{R}, t)$

end while

return \mathbf{T}

end procedure

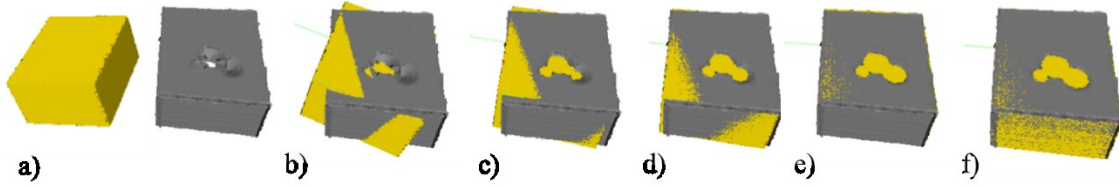


Figure 3.11. Results ICP algorithm. a) Nominal and damaged volumes. b) First iteration. c) Second iteration. d) Third iteration. e) Fourth iteration. f) Result of the ICP algorithm.

The ICP algorithm detailed in **Table 3.3** is a fast approach to overlay two point clouds with significant data sets (see **Figure 3.11**). However, two main pitfalls must be considered while implementing the ICP algorithm: the stability and convergence basin.

The result of the ICP depends on the initial homogeneous transformation \mathbf{T}_0 defined as

$$\Phi(\mathbf{R}_0, t_0) \rightarrow \mathbf{T}_0 = \begin{bmatrix} \mathbf{R}_0 & t_0 \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \in \mathbb{SE}(3).$$

If \mathbf{T}_0 is too far from the optimized solution, then the ICP diverges. Moreover, even with a fair initial transformation, the stability of the convergence is not guaranteed. To ensure convergence and increase the stability of the ICP, an automated procedure based on a genetic algorithm is developed. This procedure utilizes as inputs the convex hulls of \mathbf{I} and \mathbf{D} .

Definition: A set \mathcal{S} is convex if and only if it contains segment pq for all $p, q \in \mathcal{S}$

The convex hulls of \mathbf{I} and \mathbf{D} are computed using the Quickhull algorithm implemented in the open-source library OPEN3D.

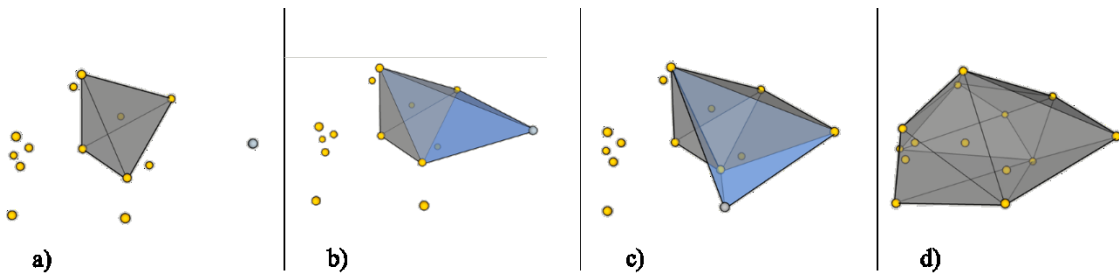


Figure 3.12 Quick hull algorithm. a) Initial step: select a random simplex, select a plane and find the furthest above point. b) Second step: delete all the visible facets from the selected point, create the new facets. c) Step four. d) Final Result of the quick hull algorithm.

3.4.3. QuickHull Algorithm

The QuickHull algorithm for the convex hull is a generalized algorithm to compute convex hull in d -dimensional space. C. Bradford Barber developed the process in 1996 [34]. The complexity is $n \log(n)$. Considering a random set of points forming a closed cone of faces with a set of ridges, called simplex. One of the facets of this simplex is selected to build a hyperplane which cuts the space in half. This hyperplane is defined by its outward-pointing unit normal and its offset from the origin. The furthest point of the point cloud above this hyperplane is found. The position of the point relative to the hyperplane is given by the inner product of the point and normal plus the offset.

Table 3.4: QuickHull Algorithm

Inputs: Simplex of $d+1$ points

Output: Convex Hull

Procedure

For each facet F

For each unassigned point p

If p is above F

Assign p to F 's outside set

For each facet F with a non-empty outside set

Select the furthest point p of F 's outside set

Initialize the visible set V to F

For all unvisited neighbors N of facets in V

If p is above N

Add N to V

The boundary of V is the set of horizon ridges H

For each ridge R in H

Create a new facet from R and p

Link the new facet to its neighbors

For each new facet F'

For each unassigned point q in an outside set of facets in V

If q is above F'

Assign q to F' 's outside set

Add each facet F' with a non-empty outside set to the set of facets F

Delete the facets in V

end procedure

Let x_0 be a point outside the hyperplane, x_i , the projection of x_0 on a plane P such as the Hessian normal form of the equation of the plane is defined as $n_1x + n_2y + n_3z + p = 0$ for every (x, y, z) in P . The signed distance d of the point x_0 relatively to the plane P is given as [35]:

$$d = (x_0 - x_i) \cdot n = \|x_0 - x_i\| \cos(\theta_0) \quad (3.8)$$

$$d = x_0 \cdot n - x_i \cdot n$$

$$d = x_0 \cdot n + p$$

If the distance is positive, the point is above the hyperplane. Then, all the visible facets from this furthest point are deleted from the simplex, and new surfaces between the newly created horizon ridges and the furthest point, are computed. The facets are visible from a point if the point is above the facet. (See **Figure 3.12** and **Table 3.4**)

Let call \mathbf{H}_I and \mathbf{H}_D respectively the convex hulls of \mathbf{I} and \mathbf{D} . The centers of mass of \mathbf{H}_I and \mathbf{H}_D , called μ_{H_I} and μ_{H_D} are subtracted to build the initial translation vector t_{h0} . In addition, the centers of mass μ_I and μ_D are subtracted to μ_{H_I} and μ_{H_D} to compute two new vectors \bar{u}_I and \bar{u}_D as follow $\bar{u}_I = \mu_{H_I} - \mu_I$ and $\bar{u}_D = \mu_{H_D} - \mu_D$. These new vectors are used to compute an initial rotation matrix \mathbf{R}_{ho} using Euler-Rodrigues formula.

Let $\sigma = \frac{\bar{u}_I}{\|\bar{u}_I\|} \times \frac{\bar{u}_D}{\|\bar{u}_D\|}$, $s = \|\sigma\|$, and $c = \frac{\bar{u}_I}{\|\bar{u}_I\|} \cdot \frac{\bar{u}_D}{\|\bar{u}_D\|}$, then the rotation matrix \mathbf{R}_{ho} is given by Euler-Rodrigues formula as:

$$\mathbf{R}_{ho} = \mathbf{I}_{3 \times 3} + [\sigma]_{\times} + [\sigma]_{\times}^2 \frac{1 - c}{s^2} \quad (3.9)$$

$$\text{With } [\sigma]_{\times} \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -\sigma_3 & \sigma_2 \\ \sigma_3 & 0 & -\sigma_1 \\ \sigma_2 & \sigma_1 & 0 \end{bmatrix}$$

To compute the initial transformation of the ICP using \mathbf{I} and \mathbf{D} , a genetic ICP algorithm is implemented with \mathbf{H}_I , \mathbf{H}_D , t_{h0} and \mathbf{R}_{h0} as inputs. The genetic algorithm executes the ICP algorithm multiple times on the convex hulls and gets a range of convergence. After each run, the distance between the two point clouds is computed as follow:

$$dist = \min_{1 < j, k < N} \|d_j - i_k\|_2 \text{ with } d_j \in \mathbf{D} \text{ and } i_k \in \mathbf{I}$$

The homogeneous transformation \mathbf{T}_0 corresponding to the minimum $dist$ is used as initial transformation for the next generation of the genetic algorithm. After a few generations, the resulting best transformation \mathbf{T}_0 is used as an initial transformation in the main ICP algorithm. The time and computational cost are decreased by using partial information and a genetic algorithm to initialize the main ICP algorithm. At the same time, convergence is guaranteed, and stability increases.

However, the function E is highly non-convex over $\mathbb{SE}(3)$. Consequently, many basins of convergence can lead the algorithm to converge to a local minimum even with relatively accurate initial transformation \mathbf{T}_0 . One possibility to decrease the number of local minima is to increase the level of correspondence between the two point clouds \mathbf{I} and \mathbf{D} or modify the source and the target in the ICP algorithm.

3.4.4. Planar surface detection

The goal of this subsection is to propose a method to increase the level of correspondence between \mathbf{I} and \mathbf{D} , in other words, decrease the minimal value of E . In fact, by finding the

planar surfaces of each point cloud, it is more likely to omit the damaged volume areas. Then, the planar surfaces identified in \mathbf{D} are at least included in the identified planar surfaces of \mathbf{I} . By applying the resulting transformation to \mathbf{D} , it is likely to decrease the optimal value of E .

The first method used to identify planer surfaces is based on the mean computed from normal vectors associated with a neighborhood of points [36].

STEP 1: A first point p_0 is randomly chosen on the point cloud.

STEP 2: Depending on the density of points, a certain number of neighbors p_i are selected, and the variance V and the mean vectors μ of the normal vectors N_i associated with these points are computed. If V_n is below a certain criterion given by the noise σ created by the scanner, then the point is on a planar surface.

STEP 3: The angles between the normal vector of its nearest neighbors N_i and the mean normal vector μ , are integrated as $\alpha_i = \cos^{-1}\left(\frac{N_i \mu}{\|N_i\| \|\mu\|}\right)$. If α_i is smaller than σ , the new point is added to the planar surface, and the mean normal vector μ is updated.

This process is repeated until each point of the point cloud has been verified. Finally, the algorithm returns surfaces and edges. The algorithm classifies the point cloud between planar surfaces and edges by modifying the tolerance on the different criteria.

The second method is based on the study of the local entropy of the point cloud. The threshold-based principal component analysis (PCA) classifies the neighborhood of points between planar, linear, or rough areas [19].

STEP 1: A first point p_0 is randomly chosen on the point cloud.

STEP 2: A neighborhood is computed around the query point p_0 using the kd-tree data structure. Then, the covariance matrix associated with the n neighbors is calculated as described below:

$$cov = \frac{1}{n+1} \sum_{i=1}^{n+1} (p_i - \bar{p})(p_i - \bar{p})^T \quad (3.10)$$

With \bar{p} defined as:

$$\bar{p} = \frac{1}{n+1} \sum_{i=1}^{n+1} p_i \quad (3.11)$$

STEP 3: An eigenvalue decomposition is applied to cov to extract three positive eigen values $\lambda_1, \lambda_2, \lambda_3$ with $\lambda_1 > \lambda_2 > \lambda_3 > 0$. The eigen values are then normalized. If the neighbors lay on a planar surface, the point distribution will spread in two directions. Consequently, two eigenvalues will be significantly bigger than the third one. In the case of rough surfaces, the point distribution lies in three dimensions, and therefore, the three eigenvalues will have a similar value.

STEP 4: A threshold is defined to determine if a point is on a planar or rough surface.

Given two expressions $c_{2D} = \frac{\sqrt{\lambda_2} - \sqrt{\lambda_3}}{\sqrt{\lambda_1}}$ and $c_{3D} = \frac{\sqrt{\lambda_3}}{\sqrt{\lambda_1}}$:

- if c_{2D} is closed to 1 and c_{3D} closed to 0, then p_0 is on a planar surface
- Else, p_0 is on a rough surface.

The second method provides good results in terms of classification. However, the computation time is heavy due to the calculation of the covariance matrix and its eigenvalue decomposition. The first method is preferred, even if the computation of the

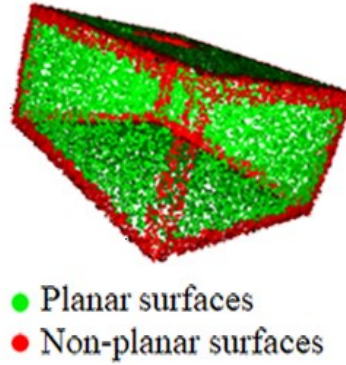


Figure 3.13. Result of surface planar detection algorithm applied on a dice.

normal vectors depends on the confidence map created by the 3D scanner and is then more sensitive to noise.

Once the planar surfaces of both **I** and **D** have been identified (see **Figure 3.13**), the following observation can be made, the set of planar points \mathbf{P}_D is included in the set \mathbf{P}_I . This observation underlines the fact that the number of correspondences between \mathbf{P}_I and \mathbf{P}_D is at least equal to the one between **I** and **D**.

3.4.5. Target and source

The repair is applied in-situ, which means that the damaged sample does not move all along the repair process. Consequently, to keep the position of **D**, the ICP is applied on **I**. In other words, **I** suffers the transformation to overlap **D**. In the previous case, the assumption $\mathbf{P}_D \subset \mathbf{P}_I$ implies that the ICP should be applied on **D**. To conserve the position of **D** in space, the ICP algorithm is computed using **D** as a moving point cloud. Then, the obtained transformation **T** is inverted and \mathbf{T}^{-1} is applied to **I**. To guarantee the feasibility of this method, the invertibility of **T** needs to be proved. As highlighted before, $\mathbf{T} \in \mathbb{SE}(\mathbf{3})$ where $\mathbb{SE}(\mathbf{3})$ forms the special Euclidean group. Consequently, for each $\mathbf{T} \in \mathbb{SE}(\mathbf{3})$, there is a

unique inverse $\mathbf{T}^{-1} \in \mathbb{SE}(3)$ as $\mathbf{T}\mathbf{T}^{-1} = \mathbf{T}^{-1}\mathbf{T} = \mathbf{I}_{4 \times 4}$. Considering $\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$, one can show that $\mathbf{T}\mathbf{G} = \mathbf{G}\mathbf{T} = \mathbf{I}_{4 \times 4}$ with $\mathbf{G} = \begin{pmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}$. Then \mathbf{G} is an inverse of \mathbf{T} so $\mathbf{G} = \mathbf{T}^{-1}$.

This proves the invertibility for all matrix $\mathbf{T} \in \mathbb{SE}(3)$.

3.5. Damaged volume extraction

Once \mathbf{I} and \mathbf{D} have been overlapped using the ICP algorithm, the damaged volume \mathbf{V}_d can be extracted. The point cloud \mathbf{V}_d includes two sets of points coming from \mathbf{I} and \mathbf{D} : one set corresponding to the rough surfaces delimitating the damages on \mathbf{D} and another set corresponding to the surfaces of the nominal volume around the area of damages.

A distance criterion between the points and their nearest neighbor is used to extract these two sets. Let call the set of points coming from \mathbf{D} , $\mathbf{V}_{d,D}$ and the points coming from \mathbf{I} , $\mathbf{V}_{d,I}$. First, the distance τ_k between each point of \mathbf{D} , and their nearest neighbor in \mathbf{I} is computed using the Kd-tree search algorithm. Then, each point of \mathbf{D} is assigned with the distance τ_k . Finally, the points distribution is classified using logistic function over τ_k .

3.5.1. Logistic function

A logistic function is a common bounded S-shaped curve defined over \mathbb{R} as:

$$L(x) = \frac{\alpha}{1 + e^{-k(x-x_0)}} \quad (3.12)$$

Where x_0 , the x value of the sigmoid's midpoint.

α , the curve's maximum value, k , the logistic growth rate or steepness of the curve

The tuple $(d_k, L(\tau_k))$ is computed with $\alpha = 1$. x_0 is defined as the inverse image of the 10th decile of the set $\{L(\tau_k), k \in N_k\}$. Then, all the points above a certain distance criterion ε are identified as points of the damaged area and are added to $\mathbf{V}_{d,D}$. The criterion ε is chosen by the operator. It is advised to choose ε greater than 0.7 to guarantee a discriminative classification. The same process is applied to the points of $\mathbf{V}_{d,I}$ with τ_k defined as the distance between a point of \mathbf{I} and its nearest neighbor on \mathbf{D} .

3.5.2. Gap filling procedure

It is admitted that \mathbf{D} is a noisy point cloud; consequently, it is likely that some noisy points of \mathbf{D} have been added to $\mathbf{V}_{d,D}$ and identically that some points of \mathbf{I} far from \mathbf{D} due misalignment during the ICP process, have been added to $\mathbf{V}_{d,I}$. To correct the possible errors, the distances between the points of $\mathbf{V}_{d,D}$ and their nearest neighbor in $\mathbf{V}_{d,I}$ are computed. The point of $\mathbf{V}_{d,I}$ with the minimum distance is selected as a starting point to run a new flood fill algorithm applied on $\mathbf{V}_{d,D} \cup \mathbf{V}_{d,I}$. The criterion δ for the flood fill algorithm (see section 3.3) is equal to the minimum distance. This process guarantees the identification of points only in the worn volume if the maximum deviation between \mathbf{I} and \mathbf{D} is inside the damaged area. In addition, the minimal distance between $\mathbf{V}_{d,D}$ and $\mathbf{V}_{d,I}$ that will be called gap Γ , is at least equals to the inverse image of ε , i.e., $\Gamma = x_0 + \frac{1}{k} \ln\left(\frac{\varepsilon}{1-\varepsilon}\right)$. Considering that Γ is greater than the average distance between two points of the same point cloud, it results in a possible loss of information at the interface. This gap Γ must be filled to avoid any omitted points at the interface between $\mathbf{V}_{d,D}$ and $\mathbf{V}_{d,I}$.

For all the points delimitating the boundaries of $\mathbf{V}_{d,D}$ and $\mathbf{V}_{d,I}$ called $\mathbf{O}_{d,D}$ and $\mathbf{O}_{d,I}$, their neighborhood, respectively inside \mathbf{D} and \mathbf{I} and under a distance Γ , are computed. The points

inside the neighborhoods are added to $\mathbf{V}_{d,D} \cup \mathbf{V}_{d,I}$. The updated set $\mathbf{V}_{d,D} \cup \mathbf{V}_{d,I}$ is \mathbf{V}_d . To efficiently compute these neighborhoods and fill the gap between $\mathbf{V}_{d,D}$ and $\mathbf{V}_{d,I}$, a gap-filling procedure is used (see **Table 3.5**). This algorithm corresponds to an iterative approach with a progressive decreasing step that aims to reduce iteration after iteration the gap between $\mathbf{V}_{d,D}$ and $\mathbf{V}_{d,I}$.

Let call $\mathbf{V}_{n,D}$ and $\mathbf{V}_{n,I}$ the two sets of points that are inside \mathbf{D} and \mathbf{I} but not in $\mathbf{V}_{d,D}$ and $\mathbf{V}_{d,I}$, i.e., $\mathbf{V}_{n,D} = \{\mathbf{d}_k \in \mathbf{D}, \mathbf{d}_k \notin \mathbf{V}_{d,D}\}$ and $\mathbf{V}_{n,I} = \{\mathbf{i}_k \in \mathbf{I}, \mathbf{i}_k \notin \mathbf{V}_{d,I}\}$.

The gap-filling procedure is repeated $\left\lceil \frac{1}{\varepsilon} \right\rceil$ times using $\mathbf{V}_{d,D}$ for the first iteration and $\mathbf{O}_{d,D}$ for the others. This procedure is also applied using $\mathbf{V}_{d,I}, \mathbf{V}_{n,I}, \mathbf{V}_{n,D}$ as input in this specific order.

Table 3.5: Gap filling procedure

Inputs: $\mathbf{V}_{d,D}, \mathbf{V}_{n,D}, \mathbf{V}_{n,I}, \varepsilon \in [0, 1]$

Output: *Outlying neighborhood* $\mathbf{O}_{d,D}$

Procedure

$\mathbf{l}_{\text{ref}} \leftarrow$ *mean distance between the points of* $\mathbf{V}_{n,I}$

For d **in** $\mathbf{V}_{d,D}$

$\mathbf{dist}_d \leftarrow$ *Compute the distance between* d *and its neighbor in* $\mathbf{V}_{n,I}$

$\mathbf{S}_d \leftarrow \{\mathbf{p} \in \mathbf{V}_{n,D}, \|\mathbf{d} - \mathbf{p}\|_2 < \varepsilon \times \mathbf{dist}_d\}$

For p **in** \mathbf{S}_d

If $\mathbf{i} \in \mathbf{V}_{n,I}, \|\mathbf{p} - \mathbf{i}\|_2 \leq \mathbf{l}_{\text{ref}} = \emptyset$

$\mathbf{O}_{d,D} \supset \{p\}$

End Procedure

3.6. Surface preparation

This section presents a method to pre-repair the worn part using subtractive methods (e.g., machining). In most cases, printing directly upon the damages is impossible because of sharp angles and tiny edges, which create areas and volumes that are not reachable by the

nozzle. Moreover, corrosion on the surface of the worn part can appear, which leads to sub-optimal mechanical properties at the interface in the case of direct repair by AM. Consequently, the worn piece must be prepared to create uniform and smooth surfaces on which the material can be deposited layer by layer. On the one hand, this step consists of correcting the previously cited manufacturing constraints by removing material with subtractive methods and creating smooth surfaces on the worn part. On the other hand, it creates a printable volume for a major type of 3D printing technology.

3.6.1. Manufacturing constraints

The degree of freedom of the AM extruder will be reduced to three axes, x, y, and z, to compute a volume printable by common DED processes or extrusion printing methods. The choice of pre-repair method will also be restrained to 3-axis computer-assisted milling process.

In this project, clearance needs to be made around the damaged area to avoid collisions between the PTA torch and the damaged part. This clearance is given as the angle formed by the high and the width of the torch, named α . However, the amount of material removed for repair must be kept as little as possible for time and cost matters. Consequently, the pre-repair process is based on a trade-off to find the minimum quantity of material removed

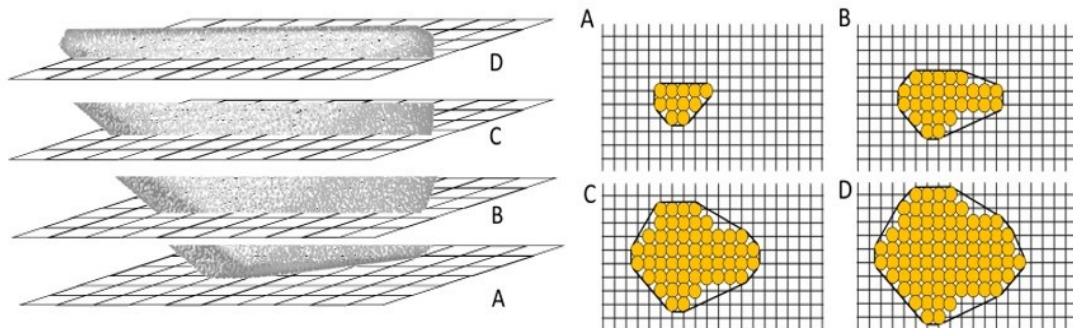


Figure 3.14. Slicing process of volume H_D

for collision-free AM repair. In addition, the diameter of the mill that will be used for pre-repair has to be considered. The volume removed after the pre-repair process must correspond to the computed volume printed for repair.

3.6.2. Surface preparation based on region growing algorithm

A slicing algorithm is used to compute a new printable volume. It consists of dividing V_d into several layers into z-direction. For each layer, new boundaries are created respecting the manufacturing constraints (see **Figure 3.14**).

First, the convex hull H_D of the damage volume V_D is computed utilizing the QuickHull algorithm and Poisson disk sample algorithm provided by Open3D (see 3.4). By converting the damage volume to a convex hull, the minimum and the maximum are global, simplifying the slicing. Moreover, accuracy issues introduced during damage identification are partially corrected.

First, H_D is sliced to the z-direction. Each layer is sliced into multiple cylinders with a diameter equal to the diameter of the drill bit. The horizontal slicing is done according to two main directions computed using the minimum bounding box algorithm.

The oriented bounding box of H_D is computed using Joseph O'Rourke's algorithm [37].

3.6.3. Minimum Bounding Box algorithm

The theorem developed by Joseph O'Rourke in 1985 proves that it is possible to find the corresponding minimum bounding box in linear time for a given convex hull (see **Table 3.6**). The minimum bounding box is given as the bounding box with the minimal intern volume. The algorithm is based on the following theorem:

“A box of minimal volume circumscribing a convex polyhedron must have at least two adjacent faces flush with edges of the polyhedron.”

The implication is verified if a convex hull edge collates with an edge of the bounding box or if two distinct hull edges flush on two adjacent box faces. In addition, the four other faces must encompass at least one vertex of the convex hull. Multiple surfaces can share this vertex. Therefore, by trying all pairs of edges, it is possible to find the minimum bounding box in a finite number of iterations.

Table 3.6 : Minimum bounding box

Inputs: Convex hull \mathbf{H}_D

Procedure

Edges $\leftarrow \{pq \text{ for } p, q \in \mathbf{H}_D\}$

for pq **in** **Edges:**

if $\exists pq'$ as $pq'.pq = 0$:

$T_{xz} \leftarrow$ transformation matrix between pq and xz plan

Edges _{t} $\leftarrow T_{xz} \cdot \mathbf{Edges}$

Box $\leftarrow \{(x_{min}, y_{min}, z_{min}), (x_{min}, y_{min}, z_{max}),$
 $(x_{max}, y_{min}, z_{min}), (x_{max}, y_{max}, z_{max})\}$

Compute volume box

if volume box $<$ volume best current:

best current \leftarrow box

else:

$T_x \leftarrow$ transformation matrix between pq and x axis

Edges _{t} $\leftarrow T_x \cdot \mathbf{Edges}$

Box $\leftarrow \{(x_{min}, y_{min}, z_{min}), (x_{min}, y_{min}, z_{max}),$
 $(x_{max}, y_{min}, z_{min}), (x_{max}, y_{max}, z_{max})\}$

Compute volume box

if volume box $<$ volume best current:

best current \leftarrow box

End Procedure

Once the oriented minimum bounding box has been computed, the two main horizontal directions u and v given as a linear combination of axes x and y are identified. A rotation matrix given as $\mathbf{R}_2 = \begin{pmatrix} \cos(\beta) & -\sin(\beta) & 0 \\ \sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$ is built. \mathbf{R}_2 corresponds to the rotation matrix between the two horizontal axes of the bounding box and the reference axis x and y .

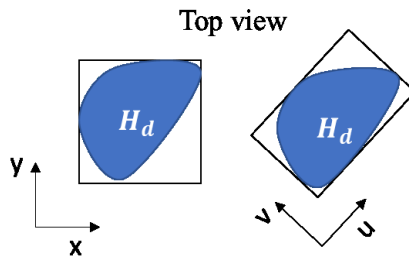


Figure 3.15 Top view of the axis-aligned bounding box and minimum bounding box

Then R_2 is applied to \mathbf{H}_D . This step enables to slice the volume according to its characteristic directions u and v and consequently minimize the quantity of material removed. (See **Figure 3.15**)

The slicing process can be divided into multiple layers with a certain thickness. For each layer, a local convex hull is computed. The convex hulls are partitioned according to a cylinder of diameter equal to the drill's diameter. Each cylinder is placed on a grid with determined positions. A cylinder can be created if one of the two following conditions is respected (see **Figure 3.14**):

- i). The grid's point on which the cylinder is created is inside the local convex hull.

ii). If there is a cylinder on the same grid's point or the previous one (in the direction of slicing) on the inferior layer.

In addition, the density of the grid controls the clearance parameter. In fact, changing the space between two adjacent points; ii) implies a modification of the steepness. The higher the density is, the smaller the clearance will be. At the end of this process, the convex point cloud of the set of cylinders \mathbf{H}_c is built and used to compute the pre-repair volume.

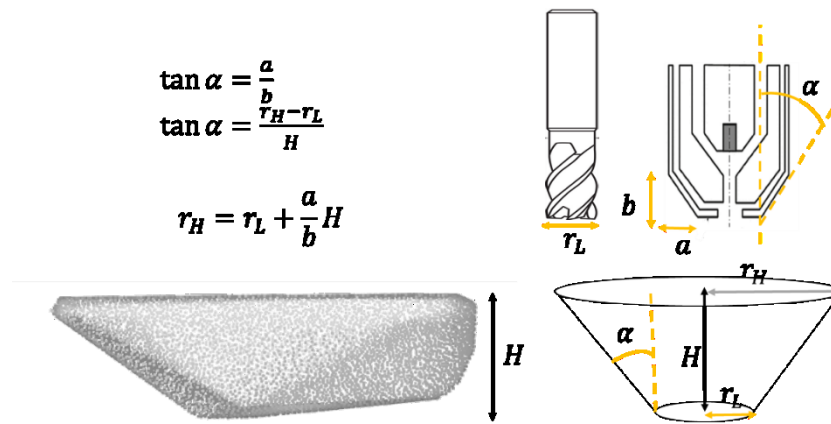


Figure 3.16. Examples of convex hull and inverted truncated cones used as inputs in the Minkowski algorithm.

3.6.4. Surface preparation based on Minkowski differences

Another possible strategy for surface preparation is to use a modified version of the Minkowski difference algorithm [38].

The Minkowski difference is generally used for collision detection. Considering two volumes A and B, the Minkowski difference executes the region swept by volume A translated to every point negated in object B. A collision occurs if the frame's origin is contained by the Minkowski difference between A and B.

This property is used to compute a surface preparation sequence. In this case, volume B is the volume \mathbf{H}_D and volume A is an inverted truncated cone. The radius of the small base

is equal to the radius of the drill used for surface preparation. The truncated cone is created from two point clouds, both forming disks with a given resolution. The space between the two disks is equal to the height of \mathbf{H}_D . In addition, the radius of the upper disk gives the clearance imposed by the geometry of the PTA torch. (See **Figure 3.16**)

For every point p in \mathbf{H}_D , an inverted truncated cone is created. The middle point of the small base is then overlapped with p . To save as much material as possible, the normal vector associated with p is also used. Eventually, the inverted truncated cone is translated as follow:

$$S_t(p) = S_0 + t_p - r_l \cdot n_p \quad (3.13)$$

With S_0 , the initial truncated cone placed on the origin. t_p is the vector coordinates of the point p used for translation of $S_t(p)$. r_l is the radius of the small base and n_p the normal vector associated to the point p .

Eventually, the volume \mathbf{H}_c is computed as the convex envelope of the set of inverted truncated cones

3.7. Pre-repaired volumes computation

This section introduces a process to adapt \mathbf{H}_c to \mathbf{I} and compute a pre-repaired volume \mathbf{I}_p will be used to mill the worn part and an additive volume \mathbf{A} that will be printed using PTA. To adapt \mathbf{I} to the results of the surface preparation algorithm, the convexity of \mathbf{H}_c can be used. In fact, adapting \mathbf{I} to the surface preparation is equivalent to removing all the points of \mathbf{I} that are inside the convex envelope \mathbf{H}_c (see **Table 3.7**). To do so, the center of mass of \mathbf{H}_c is computed. Then a point i_k of \mathbf{I} is added to \mathbf{H}_c . A new convex hull from the union

of \mathbf{H}_c and i_k is calculated. If the center of mass of this new convex hull is equal to the center of mass of \mathbf{H}_c , then i_k is added to \mathbf{I}_p else i_k is added to \mathbf{A} . This process is repeated for every point of \mathbf{I} . Then, to obtain the final pre-repaired point cloud, the points forming the interface between \mathbf{I}_p and \mathbf{H}_c must be added to \mathbf{I}_p . These points must be taken from \mathbf{H}_c and must respect the following condition: the points at the interface between \mathbf{I}_p and \mathbf{A} must be included in both \mathbf{I}_p and \mathbf{A} . This condition is equivalent to find the points of \mathbf{H}_c inside \mathbf{I} . The winding number theory is implemented to find these points.

Table 3.7: \mathbf{I}_p computation

Inputs: \mathbf{I} convex, \mathbf{H}_c convex

Procedure

$\mu_I \leftarrow$ center of mass of \mathbf{I}

$\mu_H \leftarrow$ center of mass of \mathbf{H}_c

$\mathbf{S}_I \leftarrow \{i_k \in \mathbf{H}_D\}$

For h_j *in* \mathbf{H}_c :

$\mathbf{S}_I \leftarrow \mathbf{S}_I \cup \{h_j\}$

$\mathbf{H}_{new} \leftarrow$ convex hull of \mathbf{S}_I

$\mu_{new} \leftarrow$ center of mass of \mathbf{H}_{new}

If $\mu = \mu_{new}$:

$\mathbf{I}_p \leftarrow \mathbf{I}_p \cup \{h_j\}$

For i_k *in* \mathbf{I} :

$\mathbf{S}_H \leftarrow \mathbf{S}_H \cup \{i_k\}$

$\mathbf{H}_{new} \leftarrow$ convex hull of \mathbf{S}_H

$\mu_{new} \leftarrow$ center of mass of \mathbf{H}_{new}

if $\mu \neq \mu_{new}$:

$\mathbf{I}_p \leftarrow \mathbf{I}_p \cup \{i_k\}$

End Procedure

3.8.1. Winding number

Jacobson et Al. define the winding number $\omega(p)$ as a signed integer-valued property of a point p with respect to a closed Lipschitz curve C in \mathbb{R}^2 [39]. Given to metric spaces

(X, d_x) and (Y, d_y) , a function $\varphi: X \rightarrow Y$ is said Lipschitz continuous if it exists a constant K such that for every x_1, x_2 in X :

$$d_y(\varphi(x_1), \varphi(x_2)) \leq K d_x(x_1, x_2)$$

In physical terms, let consider an observer standing at a point p , watching a point along the curve C . The winding number corresponds to the number of complete revolutions the point makes around the observer. A full trigonometric revolution will increase the winding number by one, and a clockwise one will oppositely decrease the count by one. This statement can be easily expressed in mathematical terms. For instance, in \mathbb{R}^2 , the winding number of p in cylindrical coordinates can be written as follow:

$$\omega(p) = \frac{1}{2\pi} \oint_C d\theta \tag{3.15}$$

This expression of $\omega(p)$ corresponds to the signed length of the projection of C onto the unit circle around p divided by 2π . A value of 0 means that p is outside the surface created by C , and 1 means that p is inside. In the case where C is a piece-wise function then, $\omega(p)$ can be written as $\omega(p) = \frac{1}{2\pi} \sum_{i=1}^n \theta_i$.

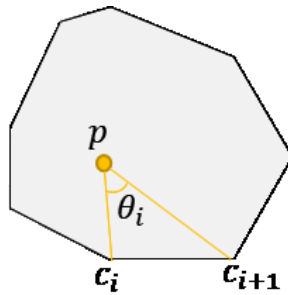


Figure 3.17. *winding number in 2 dimensions*

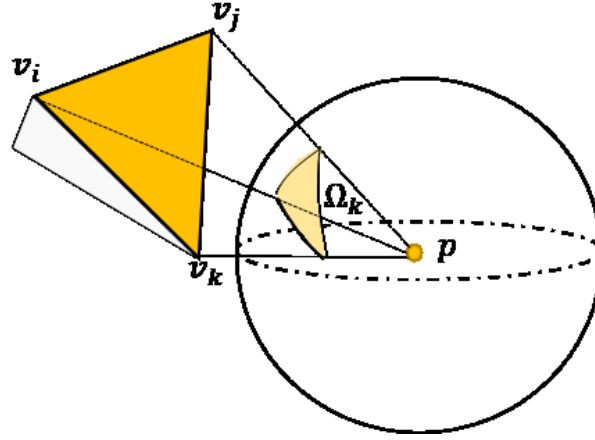


Figure 3.18. *Winding number in 3 dimensions*

Where the angle θ_i is defined as the angle between the two vectors formed by p and the two points c_i and c_{i+1} delimitating a segment of the piece-wise function. In the case of a 2D mesh, the set of points $\{c_i, 1 \leq i \leq n\}$ defines the set of vertices of the mesh (see **Figure 3.17**).

Let $a = c_i - p$ and $b = c_{i+1} - p$, then:

$$\tan(\theta_i(p)) = \frac{\sin(\theta_i(p))}{\cos(\theta_i(p))} = \frac{a \wedge b}{a \cdot b} = \frac{a_x b_y - a_y a_x}{a_x b_x - a_y b_y} \quad (3.16)$$

The properties of the winding number can be directly generalized to \mathbb{R}^3 by taking the solid angle (see **Figure 3.18**). The solid angle Ω of a closed Lipschitz surface \mathcal{S} with a respect to a point $p \in \mathbb{R}^3$ is defined using spherical coordinates to be:

$$\omega(p) = \frac{1}{4\pi} \iint_{\mathcal{S}} \sin(\phi) d\theta d\phi \quad (3.17)$$

The winding number corresponds to the number of times the closed surface \mathcal{S} wraps around the point p . In the case of triangulated piecewise-linear surfaces \mathcal{S}_k , the winding number can be defined as:

$$\omega(p) = \sum_{k=1}^m \frac{1}{4\pi} \iint_{\mathcal{S}_k} \sin(\phi) d\theta d\phi = \sum_{k=1}^m \frac{1}{4\pi} \Omega_k(p) \quad (3.18)$$

In case of solid models defined with oriented triangles $\{v_i, v_j, v_k\}$. The solid angle $\Omega_k(p)$ of a triangle surface \mathcal{S}_k with respect to p , can be defined as follow: let $a = v_i - p$, $b = v_j - p$, $c = v_k - p$, the solid angle $\Omega_k(p)$ is given as (van Oosterom and Strackee 1983)

$$\tan\left(\frac{\Omega_k(p)}{2}\right) = \frac{\det([abc])}{\|a\|\|b\|\|c\| + (a \cdot b)\|c\| + (b \cdot c)\|a\| + (c \cdot a)\|b\|} \quad (3.19)$$

Table 3.8: Winding points

Inputs: solid I_s , H_c

Procedure

Triangles $\leftarrow \left\{ \begin{bmatrix} v_{i,1} & v_{i,2} & v_{i,3} \\ v_{j,1} & v_{j,2} & v_{j,3} \\ v_{k,1} & v_{k,2} & v_{k,3} \end{bmatrix} \text{ for } \{v_i, v_k, v_j\} \in I_s \right\}$

Points $\leftarrow \{p_n, \text{ for } p_n \in H_c\}$

P_{inside} $\leftarrow \{\}$

For p_n **in** **Points:**

$\omega(p_n)$ $\leftarrow 0$

For $\{v_i, v_j, v_k\}$ **in** **Triangles:**

$a \leftarrow v_i - p_n$

$b \leftarrow v_j - p_n$

$c \leftarrow v_k - p_n$

$\omega(p_n)$ $\leftarrow \omega(p_n) + \frac{\text{atan}}{2\pi} \left(\frac{\det([abc])}{\|a\|\|b\|\|c\| + (a \cdot b)\|c\| + (b \cdot c)\|a\| + (c \cdot a)\|b\|} \right)$

if **$\omega(p_n) \geq 0$:**

P_{inside} $\leftarrow P_{inside} \cup \{p_n\}$

End Procedure

To apply the winding number and find the points of \mathbf{H}_c inside \mathbf{I} , the triangle surfaces of the solid model (STL) of \mathbf{I} and the points of \mathbf{H}_c are used (see **Table 3.8**). In addition, the point of \mathbf{H}_c added to \mathbf{I}_p are also added to the point cloud \mathbf{A} .

Consequently, at the interface between \mathbf{I}_p and \mathbf{A} , the same points are shared by the two pre-repaired point clouds. Then, to complete the structure of \mathbf{A} , the algorithm presented in **Table 3.8** is used to add the points of \mathbf{I} that are inside \mathbf{H}_c to \mathbf{A} . The final volumes \mathbf{I}_p and \mathbf{A} are perfect point clouds in the way that there is not any temporal noise at the interface.

All the steps for \mathbf{I}_p computation are allowed because the repair process is in-situ. In fact, the worn part was scanned in a reference frame of the system. Then, the nominal point cloud \mathbf{I} was overlapped with the worn part's point cloud \mathbf{D} . The damaged volume was also identified in situ. Therefore, the pre-repaired volumes \mathbf{I}_p and \mathbf{A} are also computed in-situ. Consequently, the source of error comes from the scanning process and the accuracy of the overlapping during the ICP algorithm's run. However, the accuracy of the ICP algorithm also depends on the level of correspondences between the worn and nominal points clouds. Then, the performance of the scanner also influences the accuracy of the ICP algorithm.

3.8. Surface reconstruction

The point clouds \mathbf{I}_p and \mathbf{A} represent the virtual versions of the pre-repair result using milling tools and the volume that is printed using PTA-AM. To run the pre-repair and repair processes, the LAMPP system uses CNC technology. Consequently, G-code files must be implemented, and therefore, a solid model must be built from the point clouds \mathbf{I}_p and \mathbf{A} . A solid model includes vertices, surfaces, and edges. Every surface is defined by three vertices linked together by edges to form a triangle. The set of triangles form a manifold

mesh called a solid model. A Poisson surface reconstruction algorithm is used to compute the solid model from a point cloud,

Different types of algorithms were developed during the last decades to reconstruct surfaces from point clouds. Ball pivoting strategy or Alpha shape algorithms are two of them and are considered as explicit reconstruction algorithm. In this paper, Poisson surface reconstruction (implicit reconstruction algorithm) was used for several reasons: the point density of \mathbf{I}_p and \mathbf{A} can locally be different comparing one to another, which can cause errors using explicit algorithms such as ball pivoting or alpha shape algorithms. Furthermore, this project aims to work with complex shapes and nonplanar surfaces, for which Poisson surface algorithm is perfectly suited using piece-wise polynomial functions for interpolation.

3.8.1. Poisson surface reconstruction algorithm

The Poisson surface reconstruction algorithm includes two main principles: discontinuous indicator function and isosurface extraction [40]. Considering a closed subset Ω , in three-dimensional space the discontinuous indicator function $\chi: \mathbb{R}^3 \rightarrow \mathbb{R}$ is defined as:

$$\chi(x) = \begin{cases} 1 & \text{if } x \in \Omega \\ 0 & \text{if } x \notin \Omega \end{cases} \quad (3.20)$$

Regarding the definition of χ , an integral relationship can be defined between the oriented points of the surfaces of the point clouds and χ . In fact, the gradient of χ is a vector field with a null vector almost everywhere except at points near the surface, where it is equal to the inward surface normal. Then by considering each normal vector as samples of the

discontinuous indicator function, the goal is to find χ that minimizes the following quantity:

$$E(\chi) = \int \|\nabla\chi(p) - V(p)\|^2 dp \quad (3.21)$$

Where V is the vector field of the oriented point cloud, by applying the divergence to the previous expression, this problem is equivalent to solve the Poisson equation:

$$\Delta\chi = \nabla \cdot V \quad (3.22)$$

The solution is generally approximated using a B-spline basis and then by solving a linear system. Once the discontinuous indicator function is found, the isosurfaces are extracted by seeking the level-set change.

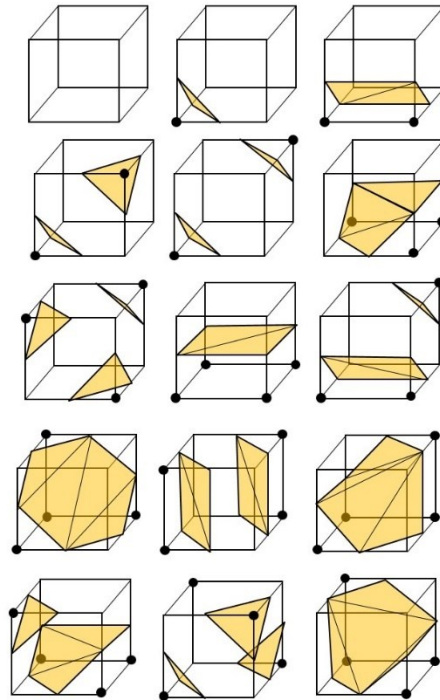


Figure 3.19. Discretized intersection of the indicatrix function with a cube. 15 configurations, symmetrical cases excluded.

3.8.2. Marching cube algorithm

One possible solution to implement the surface is to create small triangles using the marching cubes algorithm [41]. The marching cubes algorithm is based on a divide-and-conquer strategy to locate the surface in a cube made from eight vertices. The algorithm determines how the surfaces intersect the cube's edges, then moves to the next cube (see **Figure 3.19**). Each cube is encoded with a binary value. Each binary combination refers to how the surfaces intersect with the cube. Therefore, the algorithm only has to reconstruct the triangle surfaces based on the binary code associated with each cube.

In this project, the surface reconstruction was implemented using the Poisson surface reconstruction algorithm provided by the library Open3D. As underlined above, to solve the Poisson problem, the normal vectors must be computed. A normal estimation algorithm is implemented to guarantee the good orientation of each normal vector associated with the point cloud. This algorithm is based on finding the equation of the local plan associated with the neighborhood of the point where the normal vector is computed.

Considering p_1, p_2, \dots, p_N , the N points inside the neighborhood of p_1 . Find the best fitting plane is equivalent to finding c and \bar{n} as $(c_0, \bar{n}_0) = \underset{c, \bar{n}, \|\bar{n}\|=1}{\operatorname{argmin}} \sum_{i=1}^N ((p_i - c)^T \bar{n})^2$.

However, finding the best c that minimizes this expression is equivalent to looking for c_0 as $c_0 = \underset{c}{\operatorname{argmin}} \sum_{i=1}^N \|p_i - c\|^2$. c_0 is then given as the centroid of the neighborhood.

Proof:

To find the best c to minimize the previous quantity, one seeks for c so that

$$\nabla \sum_{k=1}^N \|p_i - c\|^2 = 0.$$

$$\nabla \sum_{k=1}^N \|p_i - c\|^2 = \bar{0} \quad (3.23)$$

$$\Leftrightarrow -2 \sum_{k=1}^N (p_i - c) = \bar{0}$$

$$\Leftrightarrow c = \frac{1}{N} \sum_{k=1}^N p_i = \mu$$

Consequently, the equation can be written as follow $(\mu, \bar{n}_0) = \underset{\mu, \bar{n}, \|\bar{n}\|=1}{\operatorname{argmin}} \sum_{k=1}^N (y_i^T \bar{n})^2$ with

$$y_i = p_i - \mu$$

$$\min_{\|\bar{n}\|=1} \sum_{k=1}^N (y_i^T \bar{n})^2 = \min_{\|\bar{n}\|=1} \sum_{k=1}^N \bar{n}^T y_i y_i^T \bar{n} = \min_{\|\bar{n}\|=1} \bar{n}^T \left(\sum_{k=1}^N y_i y_i^T \right) \bar{n} = \min_{\|\bar{n}\|=1} \bar{n}^T Y Y^T \bar{n}$$

In order to find the minimum using the fact that $\|\bar{n}\| = 1$, Lagrange multipliers are used.

The multivariable function \mathcal{L} is then defined as $\mathcal{L}(\bar{n}, \lambda) = \bar{n}^T Y Y^T \bar{n} - \lambda(\bar{n}^T \bar{n} - 1)$.

Consequently, seek for the best normal vector \bar{n} is equivalent to find $\nabla \mathcal{L} = 0$.

Nota Bene:

$$\frac{\partial \mathcal{L}}{\partial \bar{n}} = 2Y Y^T \bar{n} - 2\lambda \bar{n}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \bar{n}^T \bar{n} - 1 \quad (3.24)$$

Therefore, one obtains:

$$\nabla \mathcal{L} = 0$$

$$\Leftrightarrow \begin{cases} YY^T \bar{n} = \lambda \bar{n} \\ \bar{n}^T \bar{n} = 1 \end{cases}$$

It can be concluded that \bar{n} is the eigen vector of YY^T with the smallest eigenvalue. However, no preferable orientation has been given yet. The normal vectors, relative to a point already computed inside the neighborhood, are used to choose an appropriate orientation. First, a Riemannian graph is built between all the points over the point cloud. In other words, an edge between the points p_i and p_j is created only if p_i is in the neighborhood of p_j and vice versa. Then, considering this graph, a straightforward idea would be to choose an orientation for the first normal vector and propagate this orientation through the Riemannian graph. However, it can be shown in practice that the order of propagation would be important in the case of symptomatic geometries. Consequently, a safe path can be developed which consists in associating a weight $\omega_{ij} = 1 - |n_i \cdot n_j|$ to the edges of the graph. This path guarantees that the orientation of the normal vector will be chosen according to the orientation of the neighboring point with the closest normal vector or the less associated weight. With the condition that the normal n_i is flipped if $n_i \cdot n_j < 0$, the orientation can be safely propagated through the graph.

Eventually, the surfaces were rebuilt using normal surface estimation algorithm and Poisson surface algorithm. The resulting solid can be exported for toolpath generation.

3.9. Toolpath generation

At this point, two solids have been computed from I_P and A . Two tool paths must be computed for surface preparation and repair using AM. The surface preparation toolpath is computed using *MASTERCAM* and the repair toolpath using *Aplus MASTERCAM*.

MASTERCAM was created by CNC Software, Inc, founded in 1983. It contains various Computer-Aided Manufacturing software applications for machining and toolpath creation. One of the advantages of *MASTERCAM* is the possibility to define custom machines directly into the software and then specify customized parameters. In this project, the hybrid machine used for automated repair was defined into *MASTERCAM* for machining and additive processes.

The toolpath creation used in both *MASTERCAM* and *Aplus* is based on explicit mathematical solvers. First, the model is tessellated into triangular facets as they are easier to manipulate (trim, offset, etc.). Traveling salesman-type algorithms may be employed as part of the stitching strategy to link various parts of a toolpath.

Pocketing, contouring, and drilling represent the main actions run by *MASTERCAM* for surface preparation. Different parameters such as the speed or the feed rate of the machining tools can be set up regarding the material and the quantity of material that must be removed. The stock-aware tool pathing is computed using the transformed nominal volume as stock volume.

Aplus is a custom *MASTERCAM* add-on capable of computing outline, fill, boundary, or orthonormal tool paths. A machine definition file was created in regard to the requirements

of the PTA system. Finally, the volume \mathbf{A} is used to compute the toolpath for AM. The volume \mathbf{I}_p is selected as stock volume.

The tool paths are then converted into G-codes to be read by the embedded controller of the hybrid machine.

Chapter 4 Development of a Hybrid system for automated repair



Figure 4.1. *Final setup*

This chapter introduces the design and conception of the hybrid system used for in-situ automated repair. Starting from the initial setup, detailed in section 3.1.1, Chapter 4 describes the design and the integration of the different tools listed in 3.1.1 and the development of the software applications used to compute and control the hybrid repair.

4.1. Software and Hardware

This section presents the different components of the hybrid system and the software application implemented for in-situ repair.

Six different units composed the entire system: the main PC, the 3D scanner, the cutting tool, the PTA system, the central unit, and the Gantry system (see **Figure 4.1**)

4.1.1. Main PC

The operator visualizes and parametrizes the repair process through the main PC. All the computations for damage identification, surface preparation, and repair are done on this unit. Furthermore, the 3D scanner is also directly triggered from this computer.

4.1.2. Central Unit

The central unit includes a *KFLOP* embedded controller connected to two *KSTEP* stepper drivers. The two *KSTEP* drivers control the height stepper motors of the Gantry system and the related magnetic breaks. The *KFLOP* is also connected to a Huanyang inverter controlling the spindle.

4.1.3. The Gantry system

The Gantry System is a 5-axis CNC machine with three axes of translation x, y, z and two axes of rotation a, b. For each x and z-axis, two linear actuators ensure the motion. In total, five stepper motors of 2.4V@ 6A and 100Nm of holding torque are mounted on the Gantry to ensure x, y, z translation. Each stepper is secure with 24V DC magnetic breaks. Two *NEMA 23* stepper motors ensure rotation around the pitch and Yaw axis (a, b). Regarding the weight carried by the rotary table, the stepper motors have both 3Nm of holding torque. A fixture plate is mounted on the top of the rotary table and is electrically insulated from the rest of the system using high voltage, high-temperature resistant polymer spacers.

4.1.4. PTA and cutting tool

The PTA system includes a power supply, a torch, a powder hopper, and argon gas bottles.

- The power supply is a Eutronic gap 3000 power unit that delivers current and gas to the torch. The Eutronic is alimented with a transformer converting the power from 208V 60A to 30A 400V.
- The torch used for AM is an E52 torch with four ports, carrying powder and gas. A first flow in the center of the torch carries powder and argon for metal deposition. An external hollow cylinder provides an independent flow of gas for shielding.
- The powder hopper is used as a reservoir for metallic powder and delivers powder through a funnel using an auger. The auger is controlled by the embedded controller of the power supply.

The cutting tool is a high-speed AC spindle of 1.5KW. The spindle is controlled via an inverter connected to the central unit. A tool holder enables to change the diameter of the collet and the type of cutting tool used for surface preparation. A closed-loop cooling system is plugged into the spindle.

The torch, the powder hopper, and the spindle are installed on a tool's plate. The tool's plate is mounted on the y-axis linear actuator. During either surface preparation or repair, only one tool is used. Consequently, the other tool must be as far as possible from the working area. A clearance unit was designed to achieve this purpose. It consists of one linear slide on the end effector on which the PTA torch for AM is mounted. During the surface preparation, the clearance unit is in "UP" configuration, meaning that the torch is at the top extremity of the slide. Then, the torch goes down for repair. The linear slide is also controlled using the central unit.

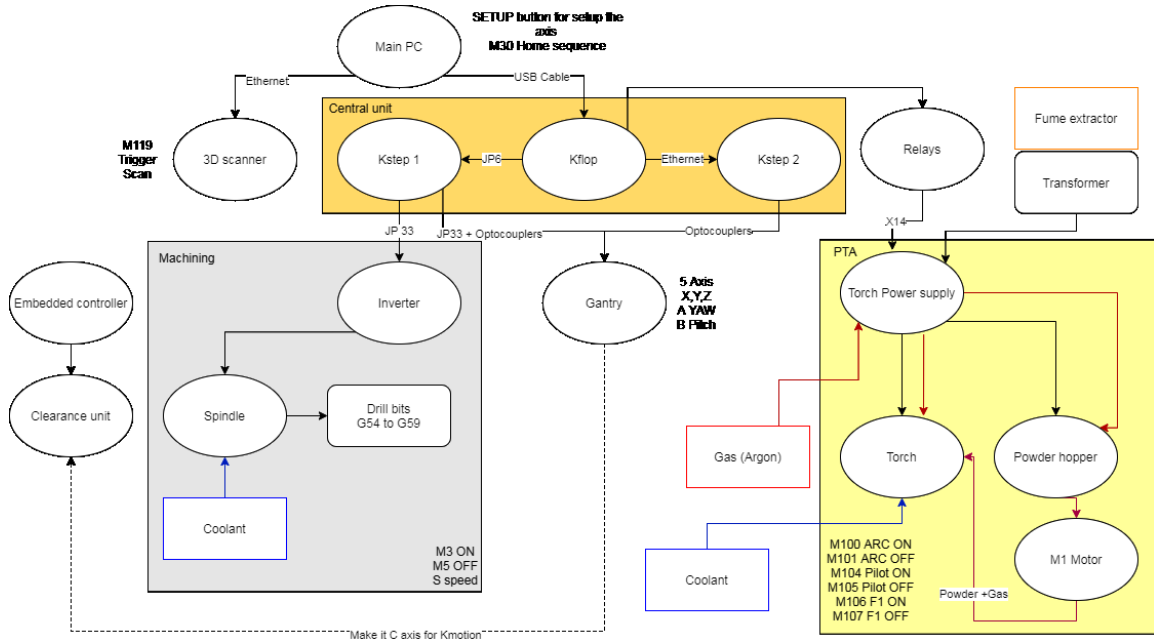


Figure 4.2. Components of the system for hybrid repair and inter-connections

4.1.5. 3D scanner

The 3D scanner is a class 3R structured light 3D scanner using red light pattern projection. This scanner is fixed on the frame at 60cm from the insulated plate. During the scanning process, the scanner remains in the same position. The rotary table offers the required degrees of freedom to scan the entire surface of the damaged part.

All the components presented in **Figure 4.1** and **Figure 4.2** are controlled using G-code commands. To create the correct sequence of commands, different software applications are used.

4.2. Graphical user interfaces (GUI)/ programs

To simplify the ergonomic of the process, only three different applications are used, one for volumes computation, the second for toolpath generation, and the last one to control the hybrid system. All the programs can communicate together. The first GUI is

implemented with Python and is used to generate the pre-repair and to-be-printed volumes. The second software is a CAM software *MASTERCAM & APLUS* that generates the toolpath for surface preparation and repair using AM. The last program is *KmotionCNC*, which is used to execute the G-codes.

4.2.1. Python GUI

The graphical interface gives the operator access to every function used for damaged detection and surface preparation. At first, the user loads the nominal version of the damaged point cloud into the sequence. Then, the operator has two choices: scan the worn part using the automated scanning sequence or directly import the worn part into the sequence. The next step consists of overlapping the two point clouds. The operator selects the source and target of the ICP algorithm and optional parameters such as planar surface detection to help the ICP converge. The damaged identification process is based on two parameters used in the logistic function; the first one is a rough approximation of the percentage of points in the undamaged part of the point cloud. The second one is used to refine the threshold. (default parameters: $\alpha = 0.9$ and $\beta = 0.8$). Once the damages have been identified, the operator must enter the geometrical parameters (a , b and r_l) of the tools used for surface preparation and repair. Then, the program returns the prepared volume \mathbf{P}_{pprep} and to-be-printed volume \mathbf{P}_{print} as STL files.

4.2.2. MASTERCAM & APLUS

The STL file of the prepared volume is imported into *MASTERCAM* software. The subtractive tool is chosen regarding the one implemented into Python GUI. The stock used for milling corresponds to the nominal volume on which have been applied the ICP algorithm for overlapping.

The STL file of the to-be-printed volume is imported into *MASTERCAM & Aplus* software. The additive tool is selected (the PTA torch). The stock used for repair by AM corresponds to the prepared volume.

Both tool paths are then generated and exported to *KmotionCNC* software.

4.2.3. *KmotionCNC*

KmotionCNC software is first triggered through Python GUI to launch the automated scanning sequence. It is also used to execute the G-codes generated by *MASTERCAM & Aplus*. Several commands are also available on *KmotionCNC*. For instance, it is possible to move the CNC machine manually with a joystick, launch the homing sequence, and set up the offset frame corresponding to the tools used for each step of the repair process.

4.3. Communication inter-software

This section describes the type of information shared by the different software applications that intervene in the repair process.

During the scanning process, the python GUI and *KmotionCNC* are called for 3D point cloud reconstruction. An automated scanning sequence written with G-code is triggered using *KmotionCNC* via a program written in Python.

This program must control the two rotation axes of the rotary table and trigger the 3D scanner. It must also be able to automatically save the scans and use them as inputs to rebuild the 3D point cloud of the object using inverse kinematics. To do so, the provided Software Development Kit (SDK) of *KFLOP* is used. This SDK gathers a bunch of C++ programs and Dynamic Link Libraries (dll). By using, the SDK, a personalized executable file can be built and opened with Python. This executable must read G-code and send action

commands to *KFLOP* regarding the specific meaning of each code. The pre-implemented *SimpleGCode.cpp* file is used and modified according to the hardware requirements to avoid starting from scratch. This C++ code contains all the information also present in *KmotionCNC* in the tool setup section.

For instance, the acceleration and the speed of each axis can be modified. All the special commands such as M100 to M119 can also be defined. The SDK provides a C++ program called *GCodeInterpreter.cpp*, in which a table with all the specific action codes is defined. For each action code such as M100, M101, ..., M119, multiple actions can be assigned. In this case, M119 code is assigned to trigger the 3D scanner and save the obtained scan. To trigger the scanner, the SDK of the Phoxi control app is used. In this SDK, a full API program is provided. This C# file is modified to trigger the scan with a given exposure and

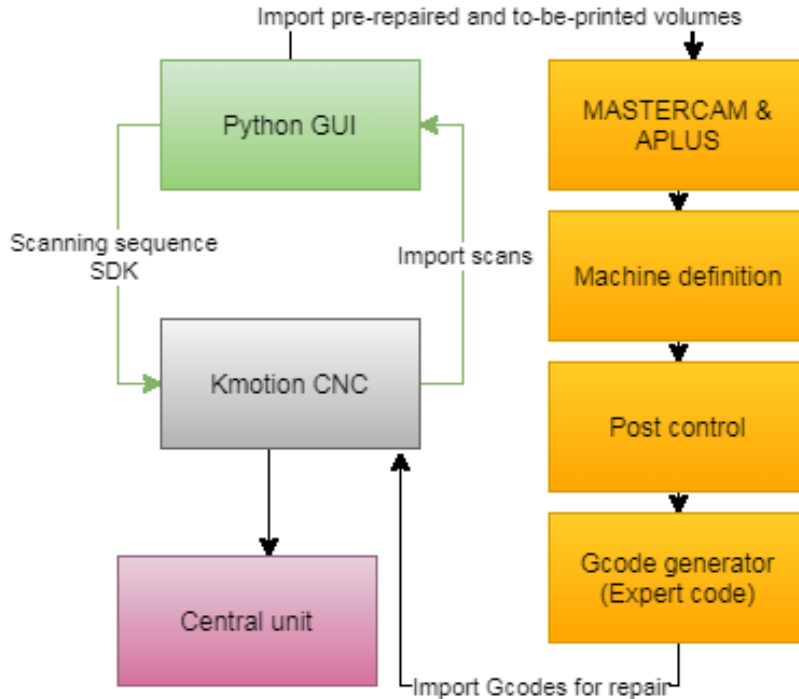


Figure 4.3. *Communication between the software applications*

laser power. A solution is then built from the C# file, and the resulting executable is linked with the M119 command using the MCodeAction table in the GCodeInterpreter file. Finally, a solution is built from the SimpleGCode.cpp file.

Once the damaged point cloud is rebuilt, the computational sequence of algorithms is executed via the Python GUI. Eventually, the prepared and to-be-printed volumes are automatically saved as STL files. Both files can be exported into any CAM software. In this project, *MASTERCAM & APLUS* is used for hybrid repair. The tool paths of the surface preparation and repair phases are computed regarding the tool's parameters chosen during the construction of the prepared and to-be-printed point clouds. The stock used for the surface preparation toolpath corresponds to the transformed nominal volume. In addition, the prepared volume is used as stock for the repair toolpath computation. The G-code corresponding to the tool paths is then computed using the integrated *Expert code* plug-in into *MASTERCAM*. However, *KmotionCNC* uses specific commands related to specific actions. Therefore, these commands must be the same for *MASTERCAM* and *KmotionCNC*.

MASTERCAM software is a computer-assisted manufacturing software. The tool paths in *MASTERCAM* (machining) and *APLUS* (additive) are all explicit mathematical 3D solvers. The model is first tessellated into triangular facets and then used to build the tool paths based on local geometry. To adapt the toolpath to the tools and machine, a virtual version of the system must be defined. To do so, control definition and machine definition files are used. They provide *MASTERCAM & APLUS* with different geometrical and operating parameters such as the unit system, the CAD model of the machine, and degree of freedom.

The machine definition also includes a post control file that helps convert the toolpath into G-code motions.

For the hybrid system, two machine definition files have been created for both machining and additive processes. A post control file has been defined for each machine definition regarding the special commands available on *KmotionCNC* (see **Table 4.1**).

Table 4.1 *Actions commands shared by MASTERCAM and KmotionCNC*

Actions	Commands
Common commands	
Translation along X axis	X
Translation along Y axis	Y
Translation along Z axis	Z
Rotate along A axis (YAW)	A
Rotate along B axis (PITCH)	B
Home	M30
General fixture OFFSET (used for calibration)	G54
Scanning	
Trigger scan	M119
Machining	
Clearance control HIGH	C
New tool	G
Select tool	55,...,58
Spindle ON	M5
Spindle OFF	M3
Spindle speed (max 500)	F
Additive	
Clearance control LOW	C
Change tool	G
Select tool	59
Current ON	M100

Current OFF	M101
Current signal	M102 P#
Powder Gas flow signal	M103 P#
Pilot ON	M104
Pilot OFF	M105
Powder Gas flow ON (F1)	M106
Powder Gas flow OFF (F1)	M107

Every special command executes internal or external C programs. The internal C programs were designed regarding the inputs and outputs available in the central unit and control the main parameters, such as the motion, the tool's offset, the spindle's speed, the powder gas flow, the arc current, and the clearance unit. External programs have also been developed to trigger the scanner using the available API. It is 20 C programs that have been written to control the Gantry system and the associated tools.

The special commands are included in the post control files at specific localizations. The tool selection, given by G55, ..., 59 is, encoded into the starting G-code section. The parameters such as gas flow or current are also defined in the post control files into the command definition section. A sketch of the communication protocol between the applications is presented in **Figure 4.3**

4.4. Communication between the equipment

The main PC is connected to the 3D scanner via an ethernet connection. The rest of the equipment is connected to a KFLOP board. The main PC sends data through a USB cable Type B. Two KSTEP boards controlling the eight J1 motor power outputs are connected to the KFLOP board through JP7 and JP5 connectors on the KFLOP and JP26 and JP36

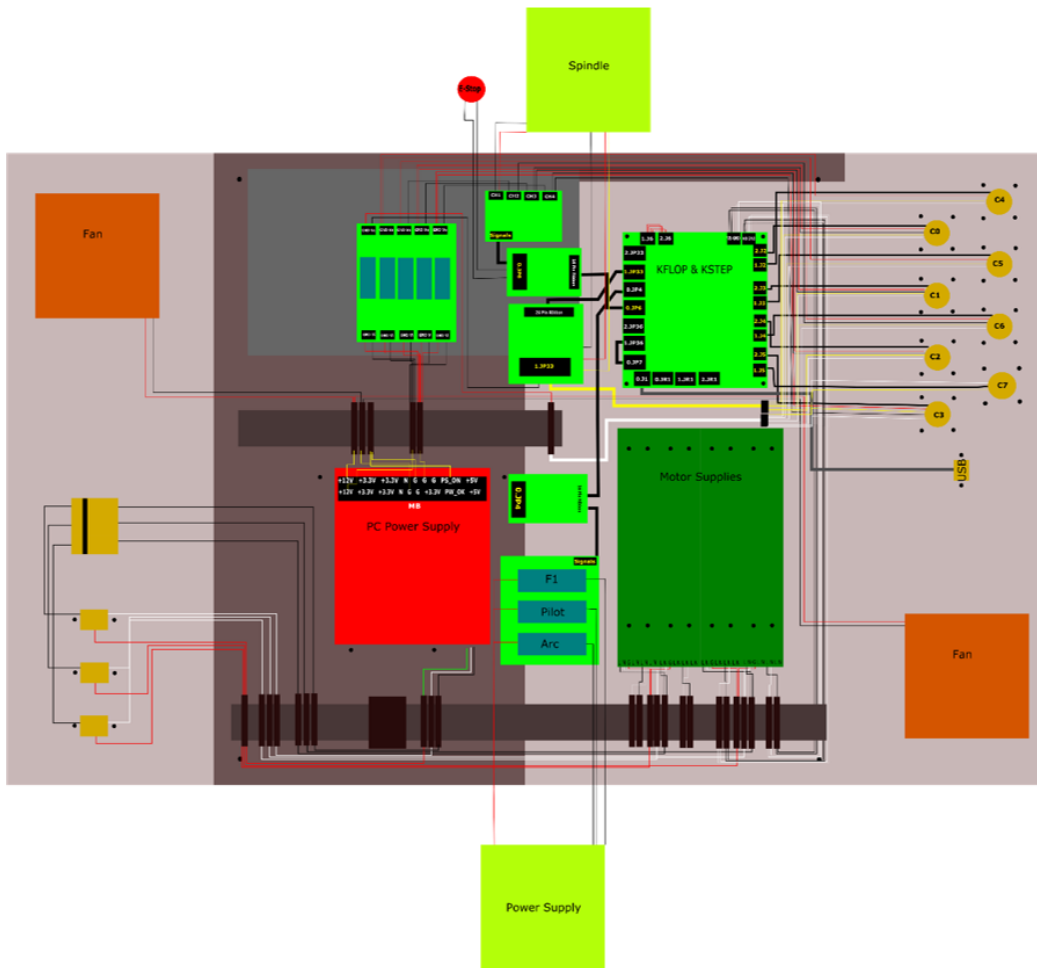


Figure 4.4. Control unit for configuration 2.

connectors on the KSTEPS. A part of the signal received on JP36 can be reoriented to the JP33 output. The JP33 connector controls the ignition of the stepper motors' breaks and the Spindle's speed via the inverter and analog signals. A second analog communication is available on the second KSTEP; however, this KSTEP is connected to the KFLOP using JP36 connector and consequently can only control the J1 motor power outputs.

❖ Configuration 1

The Eutronic power supply used for PTA welding is connected to the KFLOP board via an Arduino DUE. The communication is done using the X14 connector. Three digital I/O are available on X14 to control the arc pilot, the current, and the powder gas flow ignitions. In addition, four analog inputs are available to maintain the current and the powder gas flow according to the following protocol 10V-100%. These seven inputs received the signals from the digital and analog pins of the Arduino via relays and optocouplers. Because no analog outputs are available on KSTEP and KFLOP, the JP4 connector on KFLOP is reprogrammed to transmit 7 bits. The signal contained in the 7 bits is then used to set the amount of current and powder gas flow delivered by the power supply. To do so, a specific communication protocol between KFLOP and the Arduino board has been developed.

Communication protocol

The seven pins of the JP4 connectors are plugged with seven digital inputs on the Arduino board. This protocol aims to transmit a current or powder gas flow value that will be interpreted by the Arduino and sent to the power supply via a specific voltage value set between 0 and 10V. Consequently, the first bit corresponds to the type of parameter on which changes are applied, 0 for current and 1 for powder gas flow. The rest of the 6 bits corresponds to a decimal number coded with base 2. The range of values that can be sent is then given between 0 and 63. On *KmotionCNC*, the following commands M102 P# and M103 P# control current and powder gas flow values. (e.g.M102 P10 sets the current to 10A). This value is converted into a C program and sent to the Arduino board. A program in the Arduino board converts this base-2 signal into a base-10 float. A correction function is applied to this float sent as an analog output with a voltage between 0 and 3.3V. Optocouplers placed between X14 and the Arduino analog outputs convert the 3.3V initial

signal into a 10V signal. This communication protocol enables to control the four analog inputs of the X14 connector.

The three digital I/O on X14 are connected to the Arduino board via relays and are controlled using available digital I/O in the Arduino. C programs are used to trigger the I/O from *KmotionCNC* using specific commands detailed in **Table 4.1**.

❖ Configuration 2

The second configuration is based on the direct communication between the power supply and the central unit (see **Figure 4.4**). Remote control is also used to control the power supply manually. Three relays connected to the JP4 port of the KFLOP board control the ignition of the pilot, the gas, and the torch current. The remote control is connected in derivation with the relays so that the gas and torch ignitions can be turned on manually. The value of the current and gas flow is controlled with the remote control.

In both configurations, the spindle is connected to the central unit via an inverter. The ignition and the speed are controlled with special commands detailed in **Table 4.1**. A digital signal is used for ignition, and an analog communication is set up using the JP33 port on the first KSTEP.

4.5. Calibration of the LAMPP system

4.5.1. Calibration of the Gantry system

Each stepper motor on the gantry system is controlled with the KSTEP stepper driver controller. For inputs, A-, A+, B-, B+ connected to 4 magnets controls the stepper motor. When A- magnet is ON, it attracts the nearest teeth of the gear-shaped iron rotor. With the

teeth aligned to A- electromagnet, this would create a slight offset with the electromagnet B-. Then, A- is turned OFF and B- is turned ON, the rotor is then shifted into the direction of B-. This shift is called a step; each step is the consequence of the change of magneticity of the electromagnets. The KSTEP controller gives the sequence in which the electromagnets are turned ON and OFF. In the case of linear motion using a stepper motor, the stepper motor's shaft is connected to an endless screw with a specific pitch. The distance reached by the linear actuator consequently depends on the number of steps of the stepper motor and the pitch of the screw. To correctly calibrate the gantry system, this ratio between the actual distance and the number of steps must be defined. The unit used for this calibration is counts/inch or step/ inch; in other words, how many steps to travel one inch.

The counts/inch parameter can be modified on the tool setup screen in *KmotionCNC*. An ultrasonic sensor has been used to find a good approximation of the counts/inch given by the stepper motors. First, a cartesian axis is chosen, then a manual translation is applied using the joystick on *KmotionCNC*. The distance measured by *KmotionCNC* and the one measured by the ultrasonic sensor are compared, and the counts/inch ratio is adapted. This process is repeated for the three cartesian axis x, y, z.

Chapter 5 Experimental results

Chapter 5 is divided into two sub-sections. On the one hand, this chapter presents experimental results obtained using the sequence of computational algorithms and a commercial 3D printer. This first subsection aims to prove the robustness and efficiency of the algorithms. On the other hand, Chapter 5 describes the experiments conducted using the sequence and the hybrid system simultaneously. This second subsection brings attention to the efficiency of the hybrid repair, knowing the accuracy that the sequence can provide.

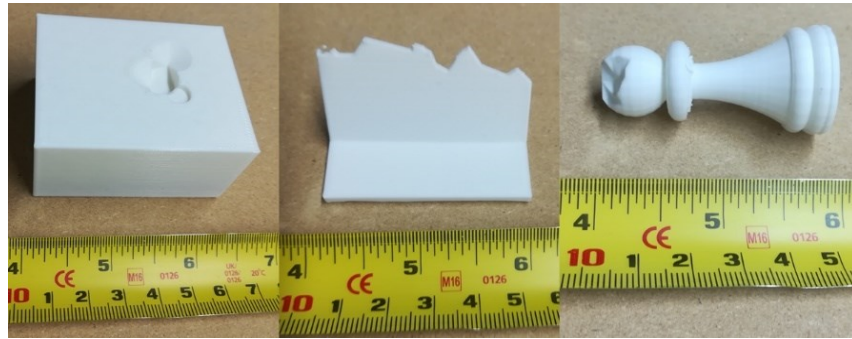


Figure 5.1. *Samples used as inputs in the sequence of computational algorithms*

5.1. First experimental results

In this section, three simple geometries are considered for testing the robustness of the sequence of algorithms. The damaged and nominal volumes were created on computer-assisted design (CAD) software. The damages were manually created. Then, the damaged volumes were printed using a commercial 3-axis 3D printer with PLA plastic. Later, the damaged volumes were scanned using a 3D scanner. The sequence of algorithms was applied to the resulting scans. Eventually, the volumes \mathbf{I}_p and \mathbf{A} were printed together using the 3-axis 3D printer. The relative position of \mathbf{A} compared to \mathbf{I}_p was maintained (as the

Table 5.1. *Criteria of selection of Sample A, B, C.*

Volumes	Convexity	Symmetry	Plane/ cylinders
Sample A	Convex	High	Planes
Sample B	Concave	Low	Planes
Sample C	Concave	High	cylinders

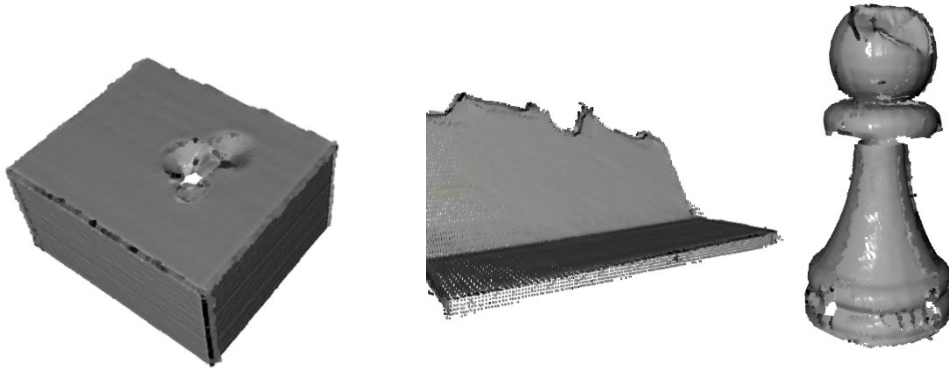


Figure 5.2. *Scans of the samples A, B, and C*

one given by the sequence of algorithms). CURA software was used to import the solid models of I_p and A and to create a single toolpath.

The three geometries A, B, C, were selected for their geometrical characteristics; respectively, a convex dice with planar surfaces, a 60-degree plane with thin walls, and a chess pawn with curved surfaces (see **Figure 5.1**). The main property that must be considered is the convexity of the volumes. The symmetry must also be considered, giving the convergence basin's size of the ICP algorithm. Finally, the global geometry of the volume, such as planar or cylindrical surfaces, is also considered as experiment parameters (see **Table 5.1**).

The three samples were scanned under six different angles (see **Figure 5.2**). Then the point cloud was rebuilt using the method detailed in 3.3.

Table 5.2. Results of the ICP algorithm in terms of error and execution time computed with the presented algorithms and, MATLAB, and Cloud Compare algorithms.

Software/ Algorithms		Sample A	Sample B	Sample C
	N	105728	43838	44546
Sequence of algorithms	$E(\mathbf{R}, t)(mm)$	0.44 ± 0.45	0.31 ± 0.22	0.41 ± 0.33
	t(s)	14.8	13.3	13.5
MATLAB	$E(\mathbf{R}, t)(mm)$	0.75	2.25	0.59
	t(s)	3.5	4.4	3.1
Cloud Compare	$E(\mathbf{R}, t)(mm)$	0.46 ± 0.43	1.34 ± 1.33	0.44 ± 0.39
	t(s)	6.1	7.6	4.5

Then the scanned point cloud and the nominal point cloud were overlapped using the ICP algorithm. **Table 5.2** lists the number of points of each worn sample, the time of execution of the ICP algorithm implemented in the sequence, and the time of execution of the ICP algorithm implemented in two other software: MATLAB and *CloudCompare*. In addition, the accuracy of the different ICP algorithms is also presented. The accuracy and time of execution computed on MATLAB and *CloudCompare* come from the best run among ten tries. To compute the error E between the two point clouds, the average distance between the points of the worn point cloud and their nearest neighbor in the nominal point cloud is computed. The standard deviation is also calculated.

The temporal noise created by the scanner is given as a normal distribution with $\sigma = 0.1mm$. Moreover, the systematic deviation created during the printing process for a desktop 3D printer is, in the literature, given as varying between 0.2 and 0.3mm [15]. Consequently, the noise included in the point cloud can be characterized by a sum of two normal distributions giving a general normal distribution of mean included between 0.2 and 0.3

mm and standard deviation equal to $\sigma_{pc} = \sqrt{\sigma_{scan}^2 + \sigma_{print}^2} \approx 0.1mm$. Finally, the

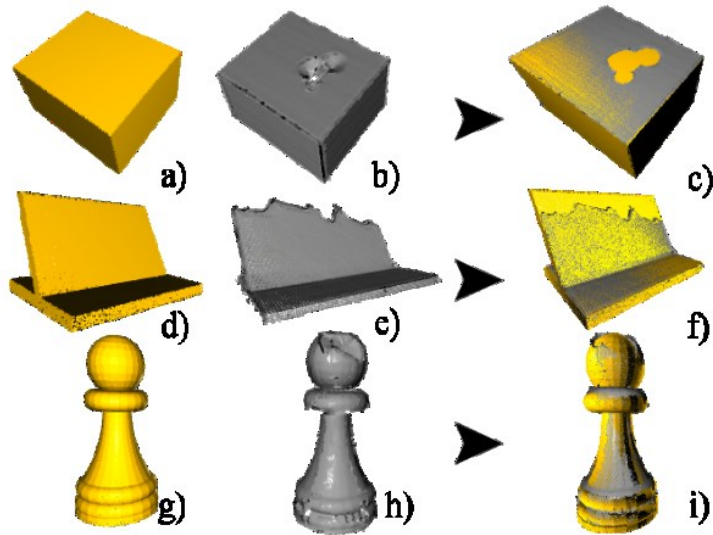


Figure 5.3. Results of the ICP algorithm. a), d), g) Nominal point clouds. b), c), h) Damaged point clouds. c), f), i) Overlap of the nominal and damaged point clouds.

range of absolute distance between the scan and the STL model for the same density of points should be between 0 and 0.6 mm with 99% confidence. Considering the importance of damage and the wrong points created during the scanning process, **Table 5.2** presents fair results (see **Figure 5.3**).

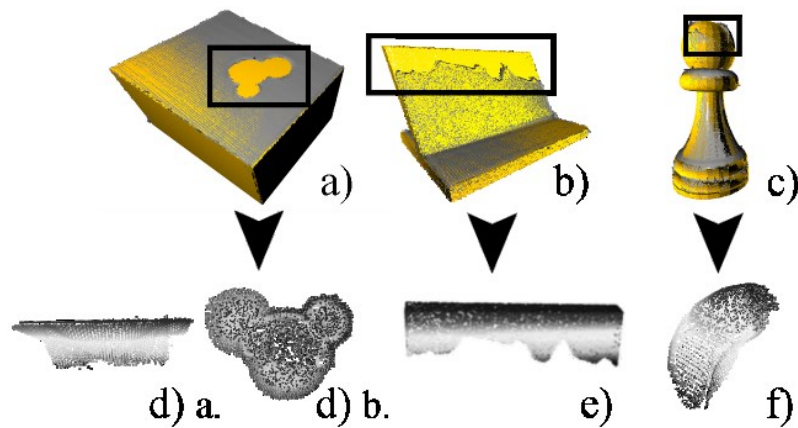


Figure 5.4. Damage identification and extraction. a), b), c) Damage identification on samples A, B, C. d), e), f) Damage extraction for the samples A, B, C.

Table 5.3. Results of the damage volume identification in terms of the number of identified points and execution time.

Samples	Number of points in the damaged area	Time of execution (s)
Sample A	5096	8.5
Sample B	7705	20.9
Sample C	3294	4.7

Then the damaged volume V_d is extracted using the method presented in section 463.5 (Figure 5.4). The time of execution includes the computation of the distances, the partitioning between damaged and undamaged areas, and the gap-filling procedure. The complexity of the gap-filling algorithm is given as $O(n^2 \ln(n))$ where n is the number of points in $O_{d,D}$ and $O_{d,I}$ after each iteration. The number of points in $O_{d,I}$ is higher in Sample B due to the number of edges which can explain the results in Table 5.3.

Table 4.4 Comparison of the density of the damaged point cloud and V_D

Density (Area/N)	Sample 1	Sample 2	Sample 3
Volume D	0.125	0.076	0.080
Volume V_D	0.130	0.071	0.068
Error %	4	7	15

Table 4.4 shows that the density of points is equivalent between the volumes D and V_D . Consequently, the average distance between the points of V_D is approximatively the same as in D . This observation proves the efficiency of the gap-filling algorithm.

Later, the convex hull H_d of each volume V_d is determined using the Quick Hull algorithm. To slice the newly computed hulls, the geometric parameters of the commercial 3D printer

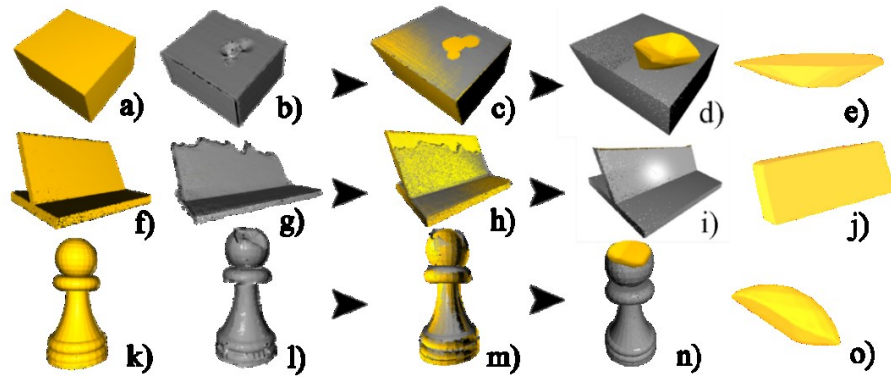


Figure 5.5 General overview of the repaired volumes computation. a), f), k) Nominal volumes. b), g), l) damaged volumes. c), h), m) Overlap using the ICP algorithm. d), i), n) Pre-repaired volumes I_p . e), j), o) To-be-printed volumes A .

were considered giving an angle of clearance equal to 71° and used as input into the Minkowski algorithm. The diameter of the drill was taken equal to 3 mm.

From the results of the slicing process, the point clouds I_p and A were computed, and the surfaces were reconstructed using Poisson surface reconstruction algorithm (see **Figure 5.5**).

The solid models were exported into CURA software. The relative positions of the pre-repaired volume and the to-be-printed volume, computed by the sequence of algorithms, were preserved. Consequently, both volumes were printed together (see **Figure 5.7**, **Figure 5.8**, **Figure 5.9**).

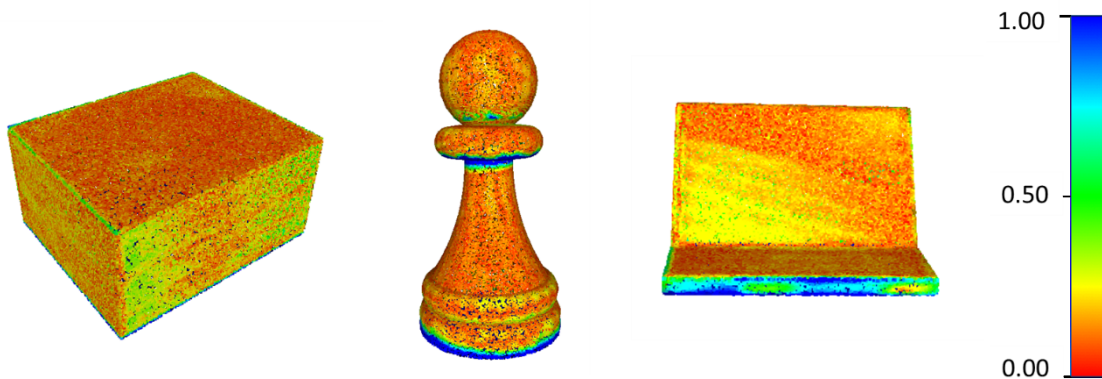


Figure 5.6 Comparison between the nominal volumes and the repaired volumes for sample A, B, and C

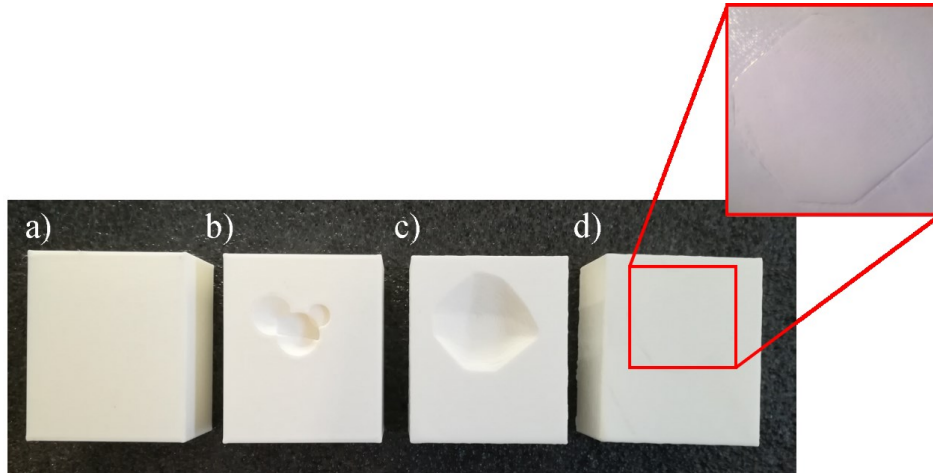


Figure 5.7 Results of repair process applied on Sample A. a) Printed nominal volume. b) Printed damaged volume. c) Printed pre-repaired volume I_p . d) Repaired volume

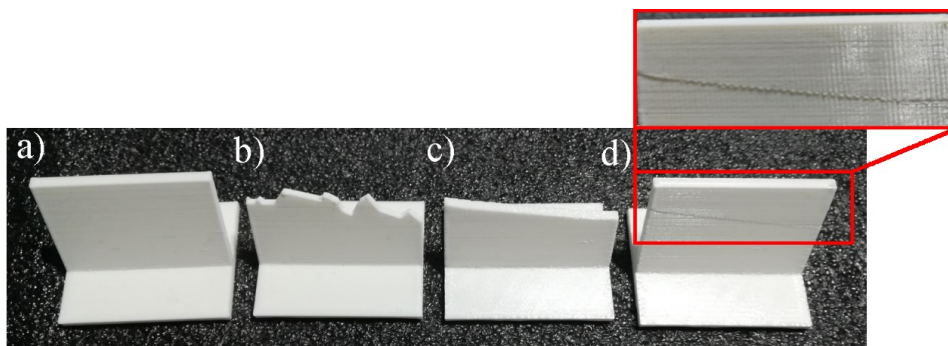


Figure 5.8 Results of repair process applied on Sample B. a) Printed nominal volume. b) Printed damaged volume. c) Printed pre-repaired volume I_p . d) Repaired volume

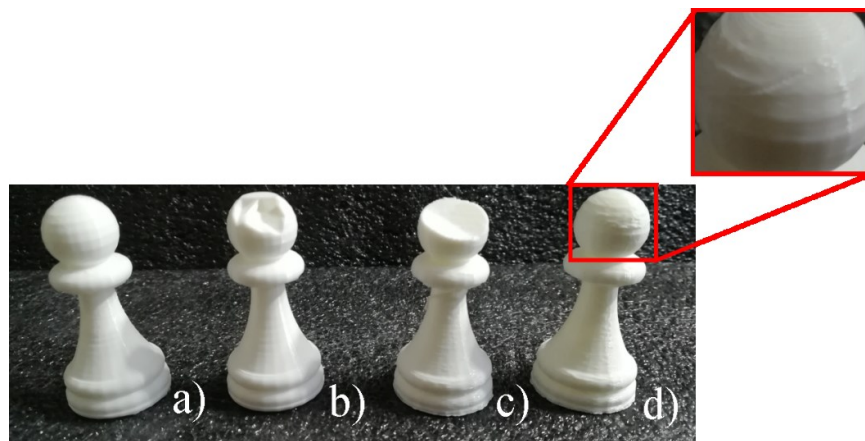


Figure 5.9 Results of repair process applied on Sample C. a) Printed nominal volume. b) Printed damaged volume. c) Printed pre-repaired volume I_p . d) Repaired volume

Figure 5.6 compares the point cloud of the nominal volume with the scan of the repaired volume. Around the damaged area, the standard deviation is less than 0.3mm, which certifies the capacity of the sequence of algorithms to restore a worn part based on its nominal version.

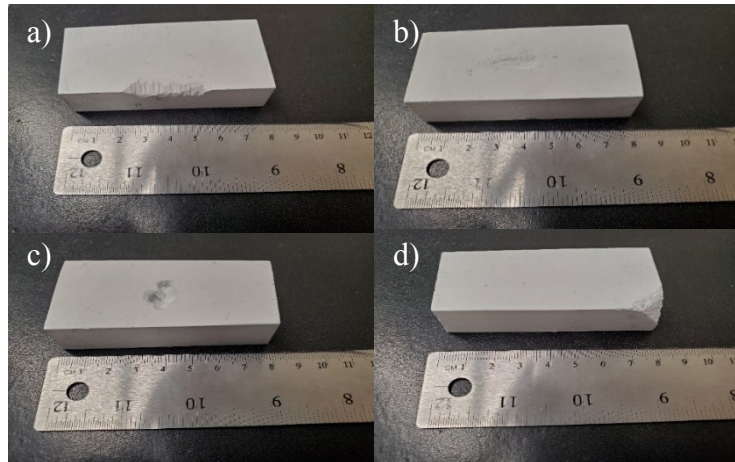
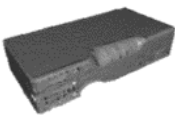
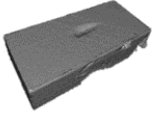

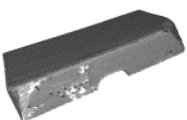
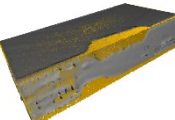
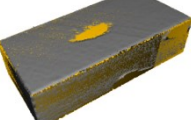
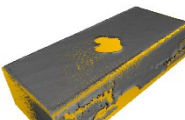
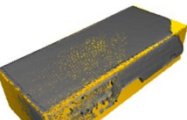
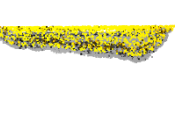
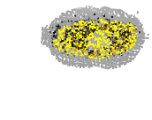
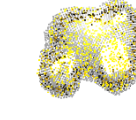

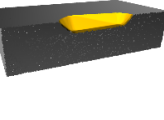
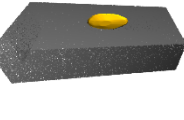
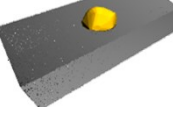



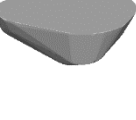




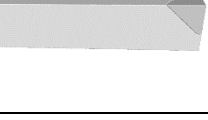


Figure 5.10. Samples used to conduct the repair process on different types of damage. a) Attrition. b) Abrasion. c) Cavity. d) Fracture.

The next set of experiments was conducted based on the shape of the damaged areas. Four aluminum dice were used. Four types of damages were manually created on the surfaces of the dices; attrition due to the contact between parts of the same assembly, abrasion caused by the interaction between the part and other materials, cavities due to accidental drilling or high-speed shocks, and fracture due to extensive collision (see **Figure 5.10**).

The samples are scanned under multiple perspectives, and the corresponding 3D point cloud is reconstructed using inverse kinematics and a flood fill algorithm. The nominal version of the damaged volumes is reproduced using CAD software, and the corresponding point cloud is computed using Poisson disk sampling. The sequence of computational algorithms returns the pre-repair volume P_{pprep} and the to-be-printed volume P_{print} regarding the geometrical characteristics of the tools used for repair.

Table 5.5. Results of the sequence of computational algorithms for the four types of damage.

Type of damage	Attrition	Abrasion	Cavity	Fracture
Scan				
Overlap				
$E(R,t)$ (mm)	0.24 ± 0.20	0.24 ± 0.15	0.25 ± 0.27	0.24 ± 0.21
Δt (s)	17	15	18	16
Damage identification				
Δt (s)	12	31	11	15
Prerepair surfaces I_p				
Removed volume (mm^3)	373	83	407	167
Δt (s)	50	37	36	36
To-be-printed volume A				
Repaired volume				
Δt (s)	4	6	5	6
Final volume (mm^3)	58445	58198	54423	49798
Nominal volume (mm^3)	58452	58210	54431	49800
Volume error	0.01%	0.02%	0.01%	0.00%
Total time (s)	93	89	70	73

The flood fill algorithm used for point cloud reconstruction includes a KD tree organization that optimized the time of computation. In fact, the complexity of the algorithm is reduced to $O(Dn^2 \ln(n))$ with n the number of points on the surfaces of the volume and D the

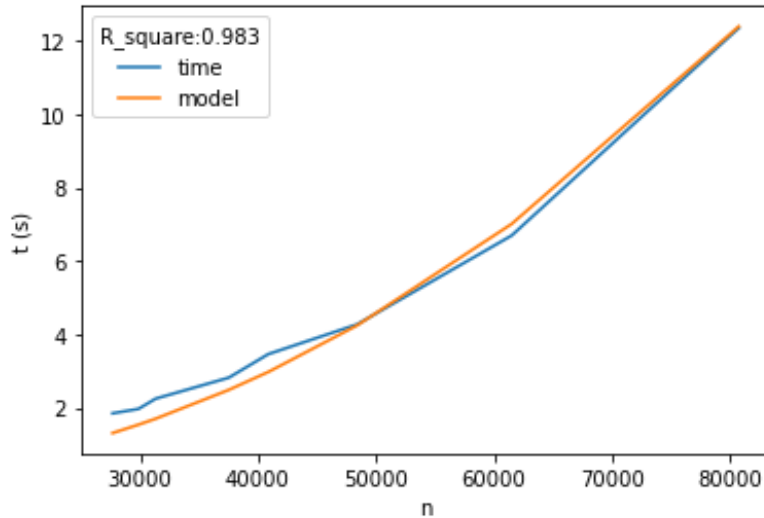


Figure 5.11. Execution time of the flood fill algorithm as a function of the number of points on the surfaces of the damaged volume.

average density of points. **Figure 5.11** shows the time of execution of the flood fill algorithm as a function of the number of points on the surface of the damaged volume for each scan. This function is fitted with the model given by the complexity of the flood fill algorithm. The value of the coefficient of determination R^2 , close to one, strengthens the fact that the complexity is $O(Dn^2 \ln(n))$.

The results of the ICP algorithm presented in **Table 5.5** show an accurate alignment of the two point clouds. Moreover, the ICP is stable among the different types of damage in terms of convergence, accuracy, and time of execution. The damage identification process enables the extraction of the volume of damages in every situation accurately. In fact, the gap-filling algorithm provides fair results no matter the geometry of the worn area (see

Table 5.5). This observation underlines the efficacy of iterative processes with progressive decreasing steps. In addition, the pre-repair and repair volumes computation delivers factual accuracy when it comes to comparing the volumes of the nominal and repaired parts (see **Figure 5.12**).

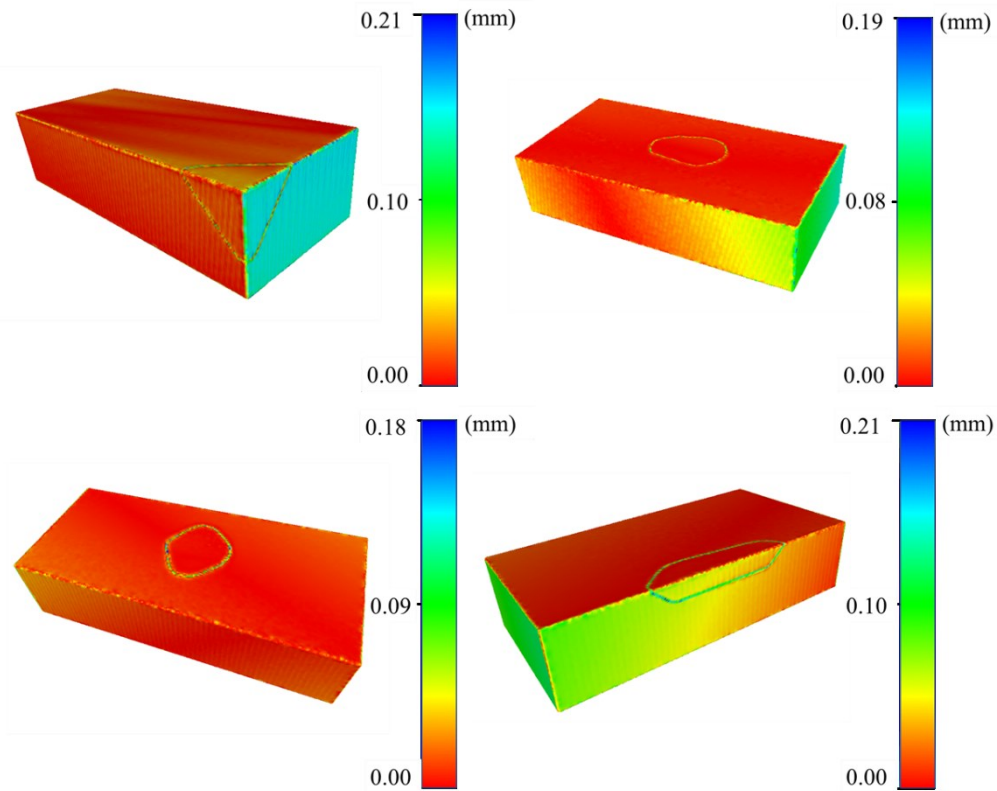


Figure 5.12. Mesh deviation after Poisson surface reconstruction between the nominal and repaired volumes. a) Attrition. b) Abrasion. c) Cavity. d) Fracture

5.2. Experimental results using the hybrid system

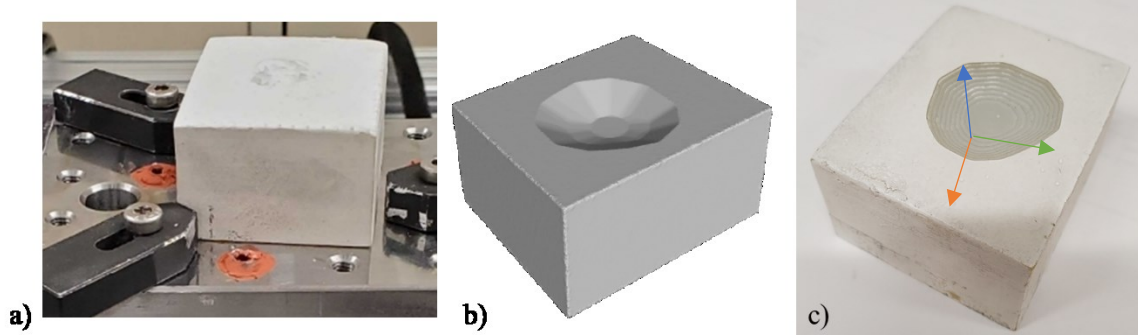


Figure 5.13. Results of the surface preparation process. a) Damaged part. b) Results of the surface preparation algorithms. c) Results of the machining process

This section presents the experiments that were conducted using the hybrid system. Only the results of surface preparation are presented. The pre-repair process is applied on acrylic dice on which damages have been created manually. The worn volume is then scanned, and the automated sequence for repair is run. Once the toolpath for surface preparation is created using *MASTERCAM*, the G-code sequence is exported into *KmotionCNC* and executed using the hybrid system. The machining process was done using a 3mm diameter flat end mill. The feed rate was kept voluntarily low at 30mm/min. (see **Figure 5.13**)

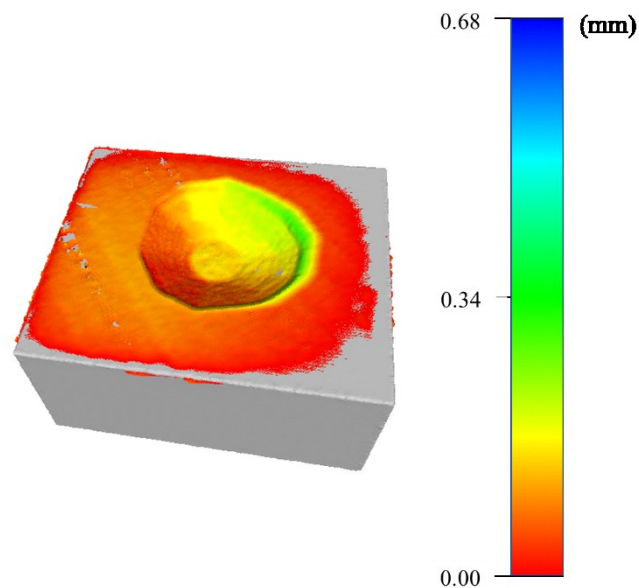


Figure 5.14. Deviation study conducted on a pre-repaired dice

A deviation study based on Hausdorff distances and quality vertex is presented **Figure 5.14**. This study compares the pre-repaired solid model obtained from the sequence of computational algorithms and the result of the machining process. The machined volume was scanned using the 3D scanner as a quality assessment tool. The maximal registered deviation is around 0.3 mm, which could correspond to the temporal noise produced by the 3D scanner. However, the maximal deviation is located into one specific area and does not match with a random and uniform repartition of noise. Consequently, other issues must have appeared during the surface preparation process. One possible source of error comes from the calibration process. The scanner origin and the Gantry origin may not be perfectly overlapped, resulting in a translated pre-repaired area. The second possibility is that the axes of the Gantry and the 3D scanner are slightly misaligned. Finally, the most likely cause of these deviations is a small error in the calibration of the linear slides. In fact, it seems that there is a wrong approximation of the number of steps corresponding to one inch of traveling distance into the y axis direction. This would explain the local deviations that occur in the y-direction when the y coordinate increases. The solution is then to recalibrate the y-axis using an ultrasound sensor.

The samples presented in **Table 5.5** are also pre-repaired using the hybrid system. The geometrical parameters set for the surface preparation are $a = 10 \text{ mm}$, $b = 30 \text{ mm}$ and $r_l = 7 \text{ mm}$. The volumes computed in 5.2 are imported into a CAM software for toolpath generation. The tool used for this purpose is a 4mm end mill with an incremental layer of 0.3mm. The samples are made of aluminum. Consequently, a two flutes end mill is used. The Spindle speed is set at 21000 rpm. The feed rate is limited to 70 mm/min and the plunge rate at 100 mm/min. Every surface preparation process took less than 10min.

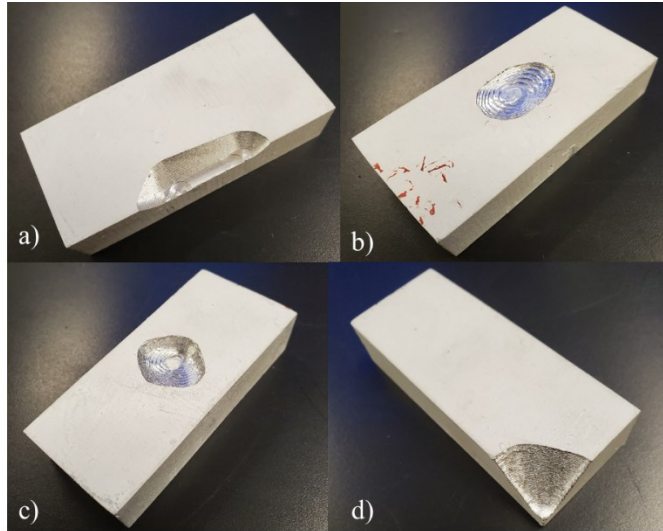


Figure 5.15. Aluminum samples after the surface preparation process have been applied. a) Attrition, b) Abrasion, c) Cavity, d) Fracture.

The results of the pre-repair process are presented in **Figure 5.15**. In order to assess the quality of surface preparation, the pre-repaired samples are scanned once again at the end of the process. A point-to-vertices distance is computed between the obtained point cloud and the STL model used for the toolpath generation. This verification permits to evaluate the differences between the volume of material that must be removed and the actual volume of material that has been removed. In fact, the milling process is a layer-by-layer process and does not perfectly follow the computed surfaces of I_p . In order to follow as much as possible, the surfaces of I_p , the maximal step down is kept low (0.3mm). The results presented in **Figure 5.16** show that the differences between the computed I_p and the actual I_p are negligible, around 0.2 mm. In addition, the verification process enables the assessment of the quality of the overlapping of the tool positioning. In fact, **Figure 5.16.c** highlights an offset between F_s and F_0 in the x,y plane. The solution of this problem is then to recalibrate the relative position of the tool in F_0 . In other words, move the tool origin until the overlap with the zero of F_s is reached.

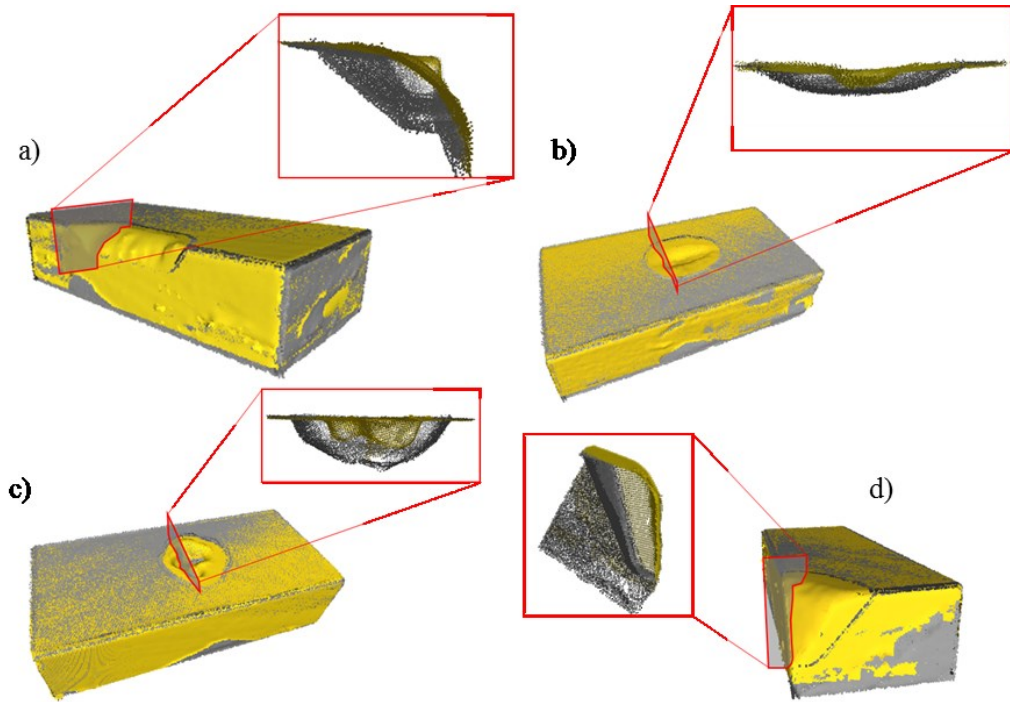


Figure 5.16. Comparison between the computed version of I_P and the actual volume I_P . a) Attrition. b) Abrasion. c) Cavity. d) Fracture.

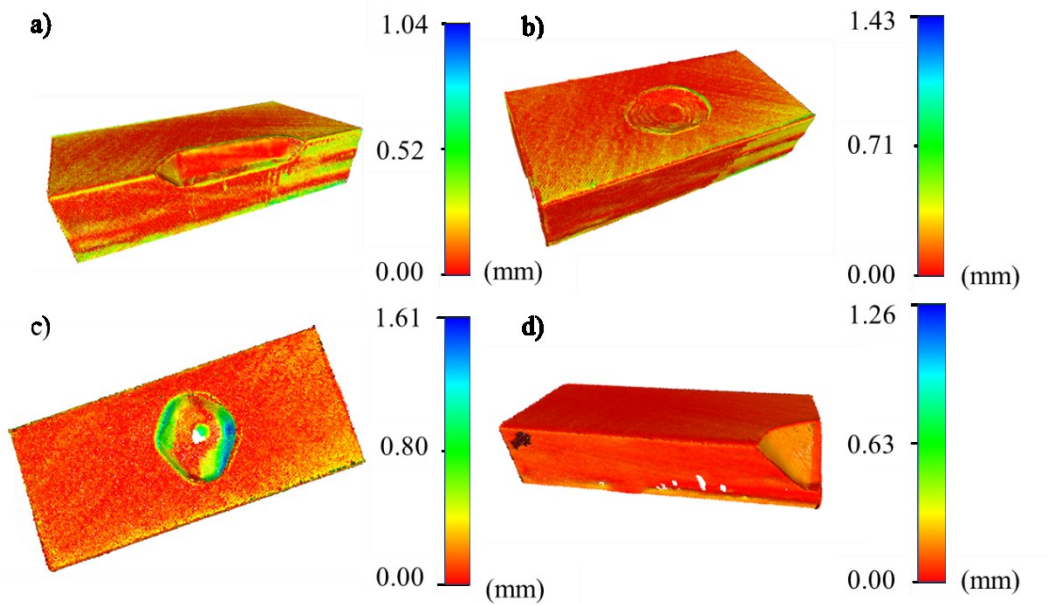


Figure 5.17. Overlap of the scans of the pre-repaired volume and damaged volume with a highlight on the damaged and pre-repaired area.

Figure 5.17. presents the overlap between the scan of the initial damaged area and the scan of the pre-repaired area after machining for the four samples. This superimposition highlights the quality of the repair. In fact, the pre-repaired area in grey, **Figure 5.17** , incorporates the initial damaged area in yellow and, at the same time, respects the geometrical constraints imposed by the tools used for repair.

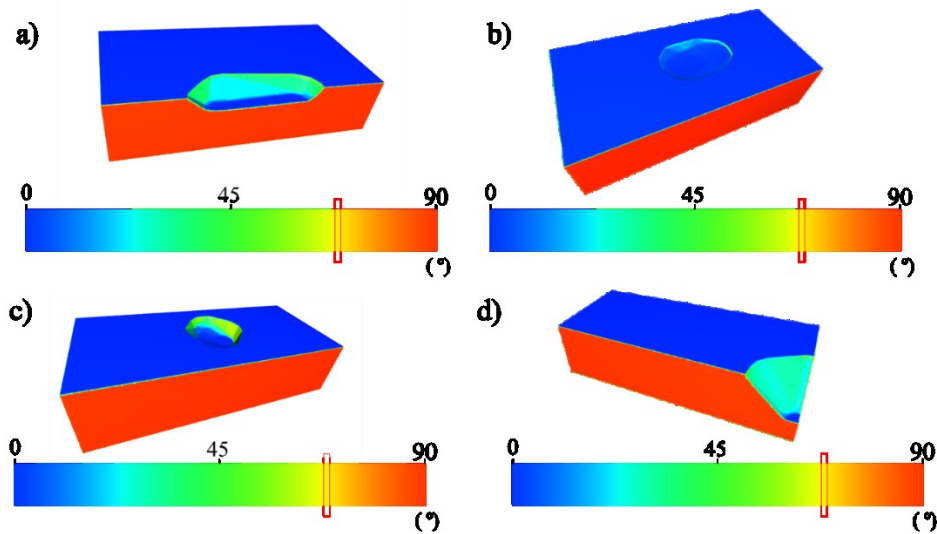


Figure 5.18. Normal vector distribution relative to the z-axis for the four samples. a) Attrition. b) Abrasion. c) Cavity. d) Fracture

Indeed, **Figure 5.18** presents the orientation of the normal vector associated with each surface. The geometrical constraints imposed earlier induce a minimum angle of 72° between the surface's normal vector and the z-axis. The red rectangles in **Figure 5.18** show the color that corresponds to this orientation. The range of colors inside the damaged area must have a wavelength larger than the one presented in the red rectangle to satisfy the geometrical constraints.

5.3. Discussions

Chapter 5 provides proof of the accuracy of the sequence of algorithms and the hybrid system. The different proves are presented through the sets of experiments and can be summarized as follow (see **Table 5.6**):

- Accuracy of the point cloud reconstruction algorithm: the rebuild point cloud is compared with the nominal version of the damaged volume. The two volumes are compared using the ICP algorithm. The resulting standard deviation is justified in section 5.1, **Table 5.2**, and **Table 5.5**.
- Accuracy of the damaged volume identification process: The density of points on the computed volume \mathbf{V}_d is compared to the point density on the damaged volume. \mathbf{V}_d is the union of $\mathbf{V}_{d,D}$ and $\mathbf{V}_{d,I}$ which are extracted based on a point-to-point distance criterion, the distance gap between $\mathbf{V}_{d,D}$ and $\mathbf{V}_{d,I}$ is at least given as the maximum deviation obtained during the ICP algorithm. A gap-filling algorithm is used to reduce this distance. The minimal distance obtained using this method is equal to the average distance given by the ICP. One way to show that the gap-filling algorithm reaches this distance is to compute the density of \mathbf{V}_d and compare it to the density of \mathbf{D} . The results are presented in **Table 4.4**. In addition, the efficiency of the damaged extraction is also justified by the results of the pre-repair process shown in **Figure 5.15**.
- Accuracy of the pre-repair and repair computation: both pre-repaired and repaired volumes are calculated using Minkowski and winding numbers algorithms. Two conditions are applied, the initial damaged volume should be included in the

repaired volume, and the geometrical constraints of the tools must be respected. **Figure 5.17** and **Figure 5.18** show that these two conditions are respected. In addition, the accuracy of the repair strategy is shown **Figure 5.6**, **Figure 5.12**, and **Table 5.5**. The volume of the repair and nominal parts are compared, the results show a difference of less than 0.02% in terms of volume and an average distance inferior to $30\mu m$.

- Accuracy of the pre-repair and repair process: the precision of the pre-repair process is presented in **Figure 5.6**, **Figure 5.16**, and **Figure 5.17**. The pre-repair process applied on the four last samples is compared to the computed pre-repaired volume I_p . The resulting deviation shows some issues in the calibration process of the hybrid process. However, despite these approximations, the system can correctly pre-repair the damaged samples. The results of repair using a polymer 3D printer turn out also to be accurate, as presented in **Figure 5.6**.

Despite the slight deviations due to approximations during the calibration process, it has been shown that the software and hardware pieces of equipment can communicate together to run the in-situ scan-assisted pre-repair process. The surface preparation and repair algorithms have autonomously computed a repair strategy using a limited number of parameters. Regarding the localization of the damaged volume given by the 3D scanner, the Gantry system has prepared the surfaces. By extension and given the fact that the interfacial surfaces are shared by the prepared and to-be-printed volumes, it can be admitted that the repair toolpath will be applied at a good location. In addition, this repair

process is ensured to be collision-free due to the geometrical parameters used in the sequence of computational algorithms.

Table 5.6. *Table of validations*

Steps of the process	Results	Third Party	Comparison	Validation
Point cloud reconstruction	Volume D	Volume I	Study of the deviation	✓
Damage identification	Volume V_d	Volume D	Study of the density	✓
Pre-repair computation	Volume I_p	Volume V_d Tools	Study of the normal vectors' orientation	✓
Repair computation	Volume A	Volume V_d Tools	Inclusion of V_d in A	✓
Pre-repair process	Scan	Volume I_p	Study of deviation	✓
Repair Process	Scan	Volume I	Study of deviation	✓

Chapter 6 Conclusion and future work

6.1. Conclusion

This thesis proposes and validates novel technology to repair manufactured parts using hybrid manufacturing. The general objectives for developing this technology are to provide an automated scan-assisted in-situ reconstruction process by using both machining and AM. This process includes a sequence of computational algorithms that creates an optimal repair strategy and the development of a hybrid system that integrates the sequence of algorithms for in-situ reconstruction. Chapter 3 introduces the algorithms used to compute the surface preparation and repair strategies for scanned damaged parts. First, point cloud reconstruction using inverse kinematics and flood fill algorithm is presented. Using reverse kinematics, the damaged part can be scanned under multiple perspectives and consequently

get a precise estimation of the damaged area. Then, the ICP algorithm, initialized with a genetic algorithm, is implemented to extract the damage's volume from the comparison of the nominal and worn parts. The stability of the ICP algorithm is increased by using planar surface detection and by modifying the source and the target in the inputs. With the experimental results, it is shown that the ability to converge is not sensitive to the part's geometry or the type of damage. In addition, the experimental results also prove that the accuracy of the ICP algorithm is equivalent to the temporal noise produced by the scanner and that the convergence is ensured even with incomplete and noisy surfaces. Then the damage's volume is extracted using a binary classification based on a Kd-tree organization and a gap-filling procedure. The gap-filling efficiency enables computing a damage's volume with an equivalent density of points that the one presents on the nominal and worn volumes. Later, the surface preparation and repair strategies are computed based on a modified Minkowski difference process and efficiently calculate the minimum amount of material that must be removed to avoid any collision during the repair process. Finally, the surfaces of both pre-repaired volume and to-be-printed volume are computed using Poisson surface reconstruction. The results of the sequence of computational algorithms for different types of damages show a difference of less than 0.02% between the nominal and repaired volumes.

The second part of the thesis presents in detail the operation of the hybrid system. Chapter 4 introduces the hardware components and the communication between the pieces of equipment. The integration of the sequence of computational algorithms into the overhaul system is also presented. The developed system is a hybrid machine combining computer vision, in-situ surface preparation, and AM repair. The damaged part is fixed on the top of

a rotary table for scanning. Then the point cloud of the damaged part is rebuilt and exported into the sequence of computational algorithms. The resulting pre-repaired and to-be-printed volumes are then exported into a CAM software for toolpath generation. Finally, the created G-code sequence is imported into the KFLOP embedded controller to repair the part. During the entire process, the damaged part remains on the top of the rotary table resulting in a gain of time compared to conventional multistep, multisystem processes. The last section of Chapter 4 presents the surface preparation results applied on a damaged acrylic cube. The pre-repair process is executed using the geometrical characteristics of an ENDER 5-PRO 3-axis 3D printer and a flat end mill of 3mm in diameter.

6.2. Future work and applications

The repair process still needs to be applied using the PTA system. In fact, a proof of concept of the sequence of computational algorithms has been successfully given using a commercial 3D printer. In addition, the surface preparation process has given fair results using the hybrid system an acrylic part. However, the PTA system enables to print with steel powder, and consequently, this process must be repeated using damaged steel parts. So far, the PTA system has been mounted and set up. The hybrid system has been calibrated and is ready to run the first experiments of in-situ hybrid repair. The next step will be to launch the sequence for repair and study the resulting repaired sample in terms of mechanical and structural properties at the interface.

Regarding the sequence of computational algorithms, the toolpath generation and execution are currently managed by external applications such as *MASTERCAM*. A possible improvement would be to include these steps into the sequence of computational algorithms using SDK as the one used for the scanning sequence.

The sequence of computational algorithms has been developed as an independent unit to be integrated into any robotic system or hybrid CNC machine. The surface preparation and repair strategies can be applied to any type of parts and damages. In addition, the choice of parameters enables the selection of any tool for repair. An independent GUI is currently being developed to make the sequence of computational algorithms user-friendly.

References

- [1] D. Brückner, F.; Lepski, *Laser cladding*, vol. 119. 2017.
- [2] M. S. Scott Crump, Minnetonka, “APPARATUS AND METHOD FOR CREATING THREE-DIMENSIONAL OBJECTS,” 5,121,329, 1989.
- [3] G. Bi and A. Gasser, “Restoration of nickel-base turbine blade knife-edges with controlled laser aided additive manufacturing,” *Phys. Procedia*, vol. 12, no. PART 1, pp. 402–409, 2011, doi: 10.1016/j.phpro.2011.03.051.
- [4] E. Yasa, “Investigation of Free-Form Surface Reconstruction Techniques for Reverse Engineering of Worn-Out Gas Turbine Blades: a Case Study,” *UMTIK 16th Int. Conf. Mach. Des. Prod.*, no. July 2014, 2014, [Online]. Available: <http://umtik.com/2014/Home.htm>.
- [5] X. Zhang, W. Li, X. Chen, W. Cui, and F. Liou, “Evaluation of component repair using direct metal deposition from scanned data,” *Int. J. Adv. Manuf. Technol.*, vol. 95, no. 9–12, pp. 3335–3348, 2018, doi: 10.1007/s00170-017-1455-y.
- [6] Rahito, D. A. Wahab, and A. H. Azman, “Additive manufacturing for repair and restoration in remanufacturing: An overview from object design and systems perspectives,” *Processes*, vol. 7, no. 11, 2019, doi: 10.3390/pr7110802.
- [7] MDM Standards, “CMM Evolution.” [Online]. Available: <https://www.mdmstandard.ro/wp-content/uploads/Istoria-CMM-in-engleza.pdf>
- [8] G. Tucci, G. Guidi, D. Ostuni, F. Costantino, M. Pieraccini, and J.-A. Beraldin, “Photogrammetry and 3D Scanning: Assessment of Metric Accuracy for the Digital Model of Danatello’s Maddalena* PHOTOGRAMMETRY AND 3D SCANNING: ASSESSMENT OF METRIC ACCURACY FOR THE DIGITAL MODEL OF DONATELLO’S MADDALENA,” p. 44879.
- [9] G. F. Marshall, “Handbook of Optical and Laser Scanning,” *Opt. Sci. Eng.*, pp. 417–476, 2004, Accessed: Sep. 17, 2021. [Online]. Available: https://www.academia.edu/5716700/Handbook_of_Optical_and_Laser_Scanning.
- [10] P. F. McManamon, “LiDAR Technologies and Systems,” *LiDAR Technol. Syst.*, Jul. 2019, doi: 10.1117/3.2518254.
- [11] P. Fechteler, P. Eisert, and J. Rurainsky, “FAST AND HIGH RESOLUTION 3D FACE SCANNING,” *Copyright*, 2007.
- [12] Y. Bokhabrine *et al.*, “3D characterization of hot metallic shells during industrial forging,” no. June 2015, 2012, doi: 10.1007/s00138-010-0297-5.
- [13] J. Huang and S. You, “Detecting objects in scene point cloud: A combinational approach,” *Proc. - 2013 Int. Conf. 3D Vision, 3DV 2013*, no. January, pp. 175–182, 2013, doi: 10.1109/3DV.2013.31.
- [14] P. Samarin, K. B. Kent, R. Herpers, and T. Saitov, “Fiducial Marker Detection

Using FPGAs,” no. 506, 2013.

- [15] H. Junjie and L. Liangyu, “Research of Key-Technique on Automatic Repair System of Plane Blade Welding.”
- [16] Y. Zheng, A. J. Qureshi, and R. Ahmad, “Algorithm for remanufacturing of damaged parts with hybrid 3D printing and machining process,” *Manuf. Lett.*, vol. 15, pp. 38–41, 2018, doi: 10.1016/j.mfglet.2018.02.010.
- [17] X. Zhang, W. Li, K. M. Adkison, and F. Liou, “Damage reconstruction from tri-dexel data for laser-aided repairing of metallic components,” *Int. J. Adv. Manuf. Technol.*, vol. 96, no. 9–12, pp. 3377–3390, 2018, doi: 10.1007/s00170-018-1830-3.
- [18] Jianming Zheng · Zhongguo Li · Xi Chen, “Worn area modeling for automating the repair of turbine blades,” pp. 1062–1067, 2006, doi: 10.1007/s00170-003-1990-6.
- [19] M. E. Mohammadi, R. L. Wood, and C. E. Wittich, “Non-Temporal Point Cloud Analysis for Surface Damage in Civil Structures,” 2019.
- [20] C. Piya, J. M. Wilson, S. Murugappan, Y. Shin, and K. Ramani, “Virtual repair: Geometric reconstruction for remanufacturing gas turbine blades,” *Proc. ASME Des. Eng. Tech. Conf.*, vol. 9, pp. 895–904, 2011, doi: 10.1115/DETC2011-48652.
- [21] H. Liu, Z. Hu, X. Qin, Y. Wang, J. Zhang, and S. Huang, “Parameter optimization and experimental study of the sprocket repairing using laser cladding,” *Int. J. Adv. Manuf. Technol.*, vol. 91, no. 9–12, pp. 3967–3975, 2017, doi: 10.1007/s00170-017-0066-y.
- [22] X. Zhang, W. Cui, W. Li, and F. Liou, “A hybrid process integrating reverse engineering, pre-repair processing, additive manufacturing, and material testing for component remanufacturing,” *Materials (Basel)*, vol. 12, no. 12, 2019, doi: 10.3390/ma12121961.
- [23] E. Toyserkani, A. Khajepour, and S. F. Corbin, “Laser Cladding,” Aug. 2004, doi: 10.1201/9781420039177.
- [24] V. R. Davé, “Electron beam (EB)-assisted materials fabrication,” 1995, Accessed: Sep. 17, 2021. [Online]. Available: <https://dspace.mit.edu/handle/1721.1/11505>.
- [25] R. Baker, “Method of making decorative articles,” p. 300, Nov. 1920.
- [26] R. M. Gage, “Arc torch and process,” Jul. 1955.
- [27] U. Manual, “PhoXi Control 1.2,” pp. 1–71.
- [28] “KFLOP User Manual 2021,” 2005.
- [29] D. L. Peiper, “The Kinematics of Manipulators Under Computer Control,” p. 174, 1968, [Online]. Available: <http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=AD0680036>.

- [30] K. Erdmann and M. J. Wilton, *Introduction to Lie Algebras*, 1st ed. London: Springer-VERLAG, 2006.
- [31] B. Louis, “Multidimensional binary search trees used for associative searching,” *Commun. ACM*, vol. 18, no. 9, pp. 509–517, Sep. 1975, doi: 10.1145/361002.361007.
- [32] C. Yuksel, “Sample Elimination for Generating Poisson Disk Sample Sets,” *Comput. Graph. Forum*, vol. 34, no. 2, pp. 25–32, May 2015, doi: 10.1111/CGF.12538.
- [33] K. S. Arun, T. S. Huang, and S. D. Blostein, “Least-Squares Fitting of Two 3-D Point Sets,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 698–700, 1987, doi: 10.1109/TPAMI.1987.4767965.
- [34] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, “The Quickhull Algorithm for Convex Hulls,” *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, 1996, doi: 10.1145/235815.235821.
- [35] W. Gellert, S. Gottwald, M. Hellwich, H. Kastner, and H. Kustner, *Concise encyclopedia of mathematics*. New York: VNR.
- [36] Y. J. Lin, R. R. Benziger, and A. Habib, “Planar-based adaptive down-sampling of point clouds,” *Photogramm. Eng. Remote Sensing*, vol. 82, no. 12, pp. 955–966, 2016, doi: 10.14358/PERS.82.12.955.
- [37] J. O’Rourke, “Finding minimal enclosing boxes,” *Int. J. Comput. Inf. Sci.*, vol. 14, no. 3, pp. 183–199, 1985, doi: 10.1007/BF00991005.
- [38] Y. T. Feng and Y. Tan, “On Minkowski difference-based contact detection in discrete/discontinuous modelling of convex polygons/polyhedra: Algorithms and implementation,” *Eng. Comput. (Swansea, Wales)*, vol. 37, no. 1, pp. 54–72, 2020, doi: 10.1108/EC-03-2019-0124.
- [39] G. Barill, N. G. Dickson, R. Schmidt, D. I. W. Levin, and A. Jacobson, “Fast winding numbers for soups and clouds,” *ACM Trans. Graph.*, vol. 37, no. 4, 2018, doi: 10.1145/3197517.3201337.
- [40] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson Surface Reconstruction,” *Eurographics Symp. Geom. Process.*, 2006.
- [41] W. E. Lorensen and H. E. Cline, “MARCHING CUBES: A HIGH RESOLUTION 3D SURFACE CONSTRUCTION ALGORITHM,” *Comput. Graph. (ACM)*, vol. 21, no. 4, 1987.