Towards automated damage detection for in-situ remanufacturing

Nguyen, Keith Wei Liang

2020

Nguyen, K. W. L. (2020). Towards automated damage detection for in-situ remanufacturing. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/146561>

<https://doi.org/10.32657/10356/146561>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 12:01:12 SGT

TOWARDS AUTOMATED DAMAGE DETECTION FOR
IN-SITU REMANUFACTURING

NGUYEN WEI LIANG, KEITH

School of Mechanical and Aerospace Engineering

2020

TOWARDS AUTOMATED DAMAGE DETECTION FOR
IN-SITU REMANUFACTURING

NGUYEN WEI LIANG, KEITH

School of Mechanical and Aerospace Engineering

A doctoral thesis submitted to The Nanyang Technological University,
Singapore

in partial fulfilment of the requirements for the degree of
Doctor of Philosophy for Engineering
Nanyang Technological University

2020

Abstract

Remanufacturing is the restoration of used products to a like-new condition, with warranty matching the original product. It is labour intensive, and faces problems of increasing labour costs, labour shortage, and varying quality of work between individuals, but can be solved by automation. However, in the marine and offshore industry, the large parts are impractical to disassemble and transport to remanufacture offsite, and hence it should be carried out in-situ. With automated in-situ remanufacturing, there is increased safety, efficiency in material use, reduced turnaround time, and in turn reduced costs, and is the motivation for it. However, there is a lack of a formal framework for such remanufacturing to ensure repeatability and accountability, and is first established in this thesis.

A laborious remanufacturing process that benefits from automation is the inspection process, specifically damage detection and classification. It is preferable to automate damage detection from 3D data over 2D images, as the 3D data of the detected damage region is required for the hybrid manufacturing process in the downstream repair procedure. Although the ideal 3D damage detection method is comparing the scanned 3D model to the as-built model, some as-built models of legacy parts may be missing, whilst other damaged parts may differ from the as-built model due to undocumented modifications, or allowable deformations. As such, this research focuses on automating damage detection from only the 3D scan data of the part, for use in downstream repair processes.

The unoptimized computational efficiency for geometric learning, and lack of training data inhibits the direct application of machine learning for damage detection from 3D data. A literature review reveals that current 3D damage detection methods are unable to detect different damage types, have assumptions on the geometry of the undamaged part, or require additional inputs, representing a significant gap within the knowledge, and is filled by this research. Selected algorithms for 3D damage detection reviewed are implemented, alongside several proposed novel algorithms for dimensionality reduction. The extracted features are then processed via machine learning, with different machine learning models compared. The models were trained and demonstrated on synthetic scan data due to the limited access to real-world data, followed by validation of the trained model on real scan of physically damaged parts. The proposed approach can detect different types of damage from 3D scan data without assumptions on the undamaged part geometry or requiring additional inputs, with a sensitivity of 95.9%, representing a substantial improvement over current 3D damage detection methodologies. Based on the visual results, damaged regions are largely identified, although the proposed approach has difficulties identifying the exact borders of the damage region. The same approach is also implemented to classify the damage type, attaining an accuracy of 73.2%. However, it should be noted that the proposed approach may wrongly identify certain intended features as damage, such as debossed text, and is also sensitive to noise in the data.

While there are other enabling technologies required to fully automate in-situ remanufacturing, this thesis only focuses on the processes leading up to the damage detection and classification.

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

21 Jan 2020

Date

Nguyen Wei Liang, Keith

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

21 JANUARY 2020
.....

Date

.....
Assoc Prof Seet Gim Lee, Gerald

Authorship Attribution Statement

This thesis contains material from 3 paper(s) published in the following peer-reviewed journal(s) / from papers accepted at conferences in which I am listed as an author.

Section 2.3.3 is published as W. L. K. Nguyen, A. Aprilia, A. Khairyanto, W. C. Pang, G. G. L. Seet & S. B. Tor (2018) Morphological Box Classification Framework for supporting 3D scanner selection, Virtual and Physical Prototyping, 13:3, 211-221, DOI: 10.1080/17452759.2018.1433950

The contributions of the co-authors are as follows:

- All the authors were involved in the discussion for the initial project direction.
- I designed the study, and performed all the laboratory work, including experimental scanning and data processing. I also analysed the data and conducted data evaluation.
- I prepared the manuscript drafts. The manuscript was revised together with Ms Aprilia, Dr Khairyanto, and Ms Pang.
- A/Prof Seet and A/Prof Tor edited the manuscript drafts.

Section 4.3.5 is published as W. L. K. Nguyen, A. Aprilia, A. Khairyanto, W. C. Pang, G. G. L. Seet & S. B. Tor (2018) Edge Detection from Point Cloud of Worn Parts. In: Proceedings of the 3rd International Conference on Progress in Additive Manufacturing (Pro-AM 2018), pp 595–600. <https://doi.org/10.25341/D45C7S>

The contributions of the co-authors are as follows:

- A/Prof Seet and A/Prof Tor provided the initial project direction and edited the manuscript drafts, and edited the manuscript drafts
- I designed the study and wrote the program code. I also collected, processed, analysed, and evaluated the data.
- I prepared the manuscript drafts. The manuscript was revised together with Ms Aprilia, Dr Khairyanto, and Ms Pang.

Chapter 3 is an extension of the publication A. Aprilia, W. L. K. Nguyen, A. Khairyanto, W. C. Pang, S. B. Tor & G. G. L. Seet (2018) Towards Automated Remanufacturing Process with Additive Manufacturing. In: Proceedings of the 3rd International Conference on Progress in Additive Manufacturing (Pro-AM 2018), pp 696–701. <https://doi.org/10.25341/D4SW2N>

The contributions of the co-authors are as follows:

- All the authors were involved in the discussion for the initial project direction.
- Ms Aprilia prepared the manuscript drafts.
- I revised the manuscript, together with Dr. Khairyanto, and Ms Pang.

Section 2.3.3 is an extension of the publication A. Khairyanto, Aprilia, W. L. K. Nguyen & G. G. L. Seet (2016) Evaluation of 3D Scanner using Elimination and Multi Criteria Decision Making (MCDM) Method. In: 2nd International Conference on Progress in Additive Manufacturing, Singapore, 16 - 19 May 2016.

The contributions of the co-authors are as follows:

- All the authors were involved in the discussion for the initial project direction.
- Dr. Khairyanto prepared the manuscript drafts.
- I revised the manuscript, together with Ms Aprilia.

21 Jan 2020

Date

Nguyen Wei Liang, Keith

Acknowledgement

The author would like to express his sincere appreciation and gratitude to Associate Professor Gerald Seet and Associate Professor Tor Shu Beng for their guidance during this research. Their patience and valuable suggestions were of great help, and is greatly appreciated.

The author would also like thank his family, especially his wife, for their patience and care, as well as those who have in one way or another contributed to the current work. With the highest regards, the author thanks God, without whom nothing would have been possible.

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Medium-Sized Centre funding scheme.

Content Page

Abstract	i
Statement of Originality	ii
Supervisor Declaration Statement	iii
Authorship Attribution Statement	iv
Acknowledgement	vi
Content Page	vii
List of Abbreviations	xi
List of Figures	xii
List of Tables	xvi
CHAPTER 1	1
Introduction	1
1.1 Background	1
1.1.1 Remanufacturing	1
1.1.2 Remanufacturing Limitations	3
1.1.3 Remanufacturing Process Flow	4
1.2 Problem Definition	10
1.3 Motivation	11
1.4 Research Goals	12
1.5 Original Contributions	13
1.6 Structure of the Report	14
CHAPTER 2	16
Literature Review	16
2.1 Remanufacturing	16
2.1.1 Process Flow	17
2.2 Cleaning	18
2.3 Inspection	21
2.3.1 Image-based Damage Detection	21
2.3.2 3D Data Acquisition	25
2.3.3 3D Scanner Selection	26

2.3.4 3D Point Cloud Pre-Processing	60
2.3.5 Surface Damage Detection	62
2.3.6 Surface Damage Classification	69
2.3.7 3D Point Cloud Machine Learning	70
2.4 Evaluation of the State-Of-The-Art	71
2.5 Summary	73
CHAPTER 3	74
Automated In-Situ Remanufacturing Process Flow	74
3.1 Existing Process Flow	74
3.2 Process Flow Requirements	74
3.3 Proposed Process Flow	76
3.3.1 Disassembly and Reassembly	77
3.3.2 Transportation and Storage	78
3.3.3 Cleaning	79
3.3.4 Inspection	79
3.3.5 Repair Process	83
3.3.6 Evaluation	87
3.4 Summary	88
CHAPTER 4	89
Feature Extraction for Damage Detection	89
4.1 Prerequisites	90
4.2 Concept	90
4.3 Common Functions	91
4.3.1 Nearest Neighbours	91
4.3.2 Normal Estimation	92
4.3.3 Curvature Estimation	93
4.3.4 Edge and Boundary Point Detection	94
4.3.5 Single Edge Tracing	95
4.3.6 Hough Transformation for Straight Edge Detection	97
4.4 Feature Extraction	99
4.4.1 Local Analysis	99
4.4.2 Neighbourhood Analysis	105

4.4.3 Global Analysis	110
4.5 Summary	117
CHAPTER 5	118
Automated Damage Detection and Classification	118
5.1 Assumptions	118
5.1.1 Size.....	119
5.1.2 Shape.....	119
5.1.3 Material	119
5.1.4 Damage Types.....	119
5.2 Experimental Data	122
5.2.1 3D Data Acquisition.....	123
5.2.2 3D Data Representation	123
5.2.3 Data Pre-Processing	124
5.2.4 Synthetic Point Cloud Data.....	125
5.2.5 Point Cloud Annotation	129
5.3 Machine Learning Models	129
5.3.1 Decision Trees.....	129
5.3.2 Discriminant Analysis.....	130
5.3.3 Support Vector Machines (SVM).....	131
5.3.4 Nearest Neighbour Classifiers	131
5.3.5 Naïve Bayes Classifiers	131
5.3.6 Ensembles (Combinations)	132
5.3.7 Neural Network.....	132
5.4 Damage Detection via Supervised Learning.....	133
5.4.1 Classifier Comparison.....	133
5.4.2 Visual Results.....	136
5.4.3 Real Application and Results	136
5.4.4 Comparison to State-of-the-art	139
5.5 Damage Classification via Supervised Learning	139
5.5.1 Classifying from Entire Dataset.....	139
5.5.2 Classifying Detected Damage.....	142
5.5.3 Real Application and Results	146

5.6 Further Experiments.....	147
5.7 Limitations	150
5.8 Summary	151
CHAPTER 6	152
Conclusion	152
6.1 Prior Literature.....	153
6.2 Summary of Work Done	155
6.3 Contributions.....	157
6.3.1 Academic Contributions.....	157
6.3.2 Engineering Contributions	159
6.4 Further Works.....	160
6.5 Concluding Remarks.....	162
References.....	163
Annex A	178

List of Abbreviations

<u>Abbreviation</u>	<u>Meaning</u>
ATC	Automatic Tool Change
CAD	Computer Aided Design
CMM	Coordinate-Measuring Machine
CNC	Computer Numerical Control
CNN	Convolutional Neural Networks
CT	Computer Tomography
DED	Directed Energy Deposition
DoG	Difference-of-Gaussians
ICP	Iterative Closest Point
ISO	International Organization for Standardization
KHK	Kohara Gear Industry Co. Ltd.
LORA	Level Of Repair Analysis
MCDM	Multi Criteria Decision Making
M&O	Marine and Offshore
NDT	Non-Destructive Testing
PCA	Principal Component Analysis
PCD	Point Cloud Data
RADAR	RAdio Detection And Ranging
RANSAC	RANdom SAmple Consensus
RE	Reverse Engineering
R-CNN	Region-based Convolutional Neural Network
SAXS	Small-Angle X-ray Scattering
SHOT	Signature of Histograms of OrienTations
SIFT	Scale-Invariant Feature Transform
SONAR	SOund Navigation And Ranging
SVM	Support Vector Machine
0D	0-Dimensional
1D	1-Dimensional
2D	2-Dimensional
3D	3-Dimensional

List of Figures

	<u>Page</u>
Fig 1.1 Marine and Offshore structures	2
Fig 1.2 Worn rack of a jack up oil rig	3
Fig 1.3 Automated in-situ remanufacturing process flow overview	5
Fig 1.4 Hybrid manufacturing machine tool configuration	9
Fig 2.1 Traditional remanufacturing process flow	18
Fig 2.2 Generic manufacturing process	18
Fig 2.3 (a) Damage; (b)Canny edge detector for detecting damage	22
Fig 2.4 Detected tyre damage from JPEG images	23
Fig 2.5 Damage detection via PCA	24
Fig 2.6 Damage detection via trained CNN	24
Fig 2.7 Detected steel damages from images using deep learning	25
Fig 2.8 Smart 3D scanner selection methodology flow-chart	27
Fig 2.9 Reflection off various surfaces	30
Fig 2.10 Reflection from different directions	31
Fig 2.11 Non-penetrating Time-of-Flight 3D scanner schematic	36
Fig 2.12 Penetrating Time-of-Flight 3D scanner schematic	37
Fig 2.13 Laser-based triangulation 3D scanner schematic	38
Fig 2.14 Photogrammetric triangulation 3D scanner schematic	38
Fig 2.15 Phase-shifting interferometry 3D scanner schematic	39
Fig 2.16 Coherence scanning interferometry 3D scanner schematic	40
Fig 2.17 Non-penetrating shading 3D scanner	41
Fig 2.18 Intensity image penetrating shading-based 3D scanner	41
Fig 2.19 Texture on an object	42
Fig 2.20 Combined outline of object	43
Fig 2.21 KHK SR8-130 rack gear	47
Fig 2.22 The available 3D scanner: (a) FARO Edge (3.7m) ScanArm HD, (b) Artec EVA, and (c) FARO Focus 3D X130	47
Fig 2.23 Mitutoyo Beyond-Crysta C544	53
Fig 2.24 Simple faces of rack	55
Fig 2.25 Reconstructed mesh from the point clouds for: (a) FARO Edge ScanArm HD, (b) Artec EVA, and (c) FARO Focus 3D X 130	57

Fig 2.26	Deviation map of mesh as compared to as-built model for (a) FARO Edge ScanArm HD, (b) Artec EVA, and (c) FARO Focus 3D X 130	57
Fig 2.27	Obtained accuracy of the compared 3D scanners	57
Fig 2.28	Time taken of each scanner	58
Fig 2.29	Outlier removal – (a) Before removal; (b) After removal	61
Fig 2.30	Point cloud registration – (a) Before registration; (b) After registration	62
Fig 2.31	Comparison of scanned data to nominal CAD model	63
Fig 2.32	Intensity map highlighting different intensity signatures at crack	64
Fig 2.33	Mean curvature distributions (c) & (d) of Undamaged (a) and Damaged (b) Sub-area respectively	65
Fig 2.34	Defective region border	65
Fig 2.35	Displacement map of points from plane	66
Fig 2.36	Representation of reference normal and normal from defect areas	67
Fig 2.37	Estimated Nominal Profile Sequence	67
Fig 2.38	Local point density for defect detection	68
Fig 2.39	Building damage types	69
Fig 2.40	Semantic segmentation via deep learning	71
Fig 3.1	Oil rig rack and pinion to be remanufactured	75
Fig 3.2	Proposed automated in-situ remanufacturing process flow	76
Fig 3.3	MX3D robotic arm welding machine (a) Concept; (b) Prototype	78
Fig 3.4	Capriol autonomous CNC mobile robot	78
Fig 3.5	Robotic arm mounted 3D scanners: (a)Artec RoboticScan; (b)Creaform MetraSCAN 3D-R; (c) FARO® Cobalt Array Imager	81
Fig 3.6	Nominal Model Reconstruction of blade tip: (a) Scanned model; (b)Surface extrapolation; and (c)Reconstructed model	83
Fig 3.7	Separation of additive repair volume (Blue) and subtractive repair volume (Red)	85
Fig 3.8	Hybrid manufacturing toolpath generation	85
Fig 3.9	LASIMM project hybrid manufacturing machine	86
Fig 3.10	Hybrid manufacturing technologies (a) Interchangeable hybrid tool heads; (b) Mounted on a robotic arm	86
Fig 4.1	Overview of features based on damage characteristics	90

Fig 4.2	(a) k-Nearest Neighbour where $k=7$, (b) Nearest neighbour based on bounding sphere with radius r	91
Fig 4.3	Normal estimation	93
Fig 4.4	Curvature estimation	93
Fig 4.5	(a) Input PCD, (b) Output PCD with high curvature points (Yellow)	94
Fig 4.6	Maximum angle for internal point (Green), and boundary point (Blue)	95
Fig 4.7	(a) Input high curvature points (Yellow), (b) Output boundary and edge distinction (Boundary points - Red, Edge points - Yellow)	95
Fig 4.8	Tracing curvature local maxima points	97
Fig 4.9	(a) Input detected edge points (Teal), (b) Output traced edge (Green)	97
Fig 4.10	Icosahedron after 0, 1, 2, and 3 tessellation steps	98
Fig 4.11	(a) Input detected edge points (Yellow), (b) Hough transform detected lines (Green & Blue)	99
Fig 4.12	Hole in point cloud from occlusion	100
Fig 4.13	Damage detection from hole in PCD – (a) Input PCD, (b) Edge and boundary detection, (c) Potential damage points (Red)	101
Fig 4.14	(a) Input PCD with traced edges (Green), (b) Edge shape irregularity results (Red)	102
Fig 4.15	(a) Input PCD with traced edges (Green), (b) Edge orientation irregularity results (Red)	104
Fig 4.16	(a) Damaged and (b) Undamaged point cloud region	104
Fig 4.17	Damaged and undamaged region curvature distribution	105
Fig 4.18	(a) Input PCD, (b) Plane fitting damage detection (Planes – Green, Blue, & Teal, Damage – Red)	107
Fig 4.19	(a) Input PCD, (b) Edge point detection (Teal), (c) Hough transform detected straight edges (Green), (d) Potential straight edge damage identified	108
Fig 4.20	(a) Input PCD, (b) Edge point detection (Teal), (c) Hough transform detected straight edges (Green), (d) Potential corner damage identified (Red)	109
Fig 4.21	(a) Input PCD, (b) Cylinder fitting damage detection (Cylinder – Green & Teal, Damage – Red)	110
Fig 4.22	Example geometry containing 1, 2, and 3 planar symmetries	111

Fig 4.23	(a) Input PCD, (b) Damage detection from single symmetry (Red)	112
Fig 4.24	(a) Input PCD, (b) Damage detection from double symmetry (Red)	112
Fig 4.25	Example geometry containing axial symmetry	113
Fig 4.26	(a) Input PCD, (b) Damage region from axial symmetry comparison (Red)	114
Fig 4.27	Positive identification of defect region in 120° rotation	114
Fig 4.28	3D SIFT (on curvature) Keypoint detection (Blue)	115
Fig 4.29	Damage detection from patterned structure (Red)	116
Fig 5.1	Surface wear: (a) Photo; (b) Cross sectional representation	120
Fig 5.2	Pitting: (a) Photo; (b) Cross sectional representation	121
Fig 5.3	Broken Damage: (a) Photo; (b) Cross sectional representation	121
Fig 5.4	Crack: (a) Photo; (b) Cross sectional representation	122
Fig 5.5	Deformation: (a) Photo; (b) Cross sectional representation	122
Fig 5.6	3D data representations	124
Fig 5.7	CAD models of: (a)I-beam; (b)Concentric shaft; (c)Gear rack; (d)Pinion gear; (e)Turbine blade	126
Fig 5.8	Modelled damage: (a)Surface wear; (b)Pit or corrosion; (c)Broken damage; (d)Crack; (e)Plastic deformation.	127
Fig 5.9	(a) Synthetic CAD model, (b) Damage detection results	136
Fig 5.10	(a) & (c) Damaged rack, (b) & (d) Damage detection result	137
Fig 5.11	(a) Titanium turbine jet compressor engine blade, (b) Damage on blade, (c) Damage detection result	137
Fig 5.12	(a) High nickel aircraft jet turbine engine blade, (b) Damage on blade, (c) Damage detection result	138
Fig 5.13	(a) Deboss model number, (b) Wrongly detected damage (Red)	138
Fig 5.14	(a) & (c) Damaged rack, (b) & (d) Damage classification results	146
Fig 5.15	(a) Titanium turbine jet compressor engine blade, (b) Damage on blade, (c) Damage classification result	146
Fig 5.16	(a) High nickel aircraft jet turbine engine blade, (b) Damage on blade, (c) Damage classification result	147

List of Tables

	<u>Page</u>
Table 2.1 Contaminants in remanufacturing	19
Table 2.2 Common cleaning strategies	20
Table 2.3 Morphological box classification framework based on scanner methodology, for non-contact 3D scanning technology	44
Table 2.4 Classification of commercially available scanners	45
Table 2.5 Classification of compared scanners	49
Table 2.6 Technical specifications of compared scanners	51
Table 2.7 Experimental results of scanners	56
Table 2.8 Summary of limitations of prior literature	69
Table 5.1 Synthetic PCD details	128
Table 5.2 Summary of synthetic PCD	128
Table 5.3 Summary of machine learning models	130
Table 5.4 Comparison of machine learning models for damage detection	134
Table 5.5 Comparison between state-of-the-art and the proposed method	139
Table 5.6 Comparison of machine learning models for damage classification from entire dataset	141
Table 5.7 Comparison of machine learning models for damage classification from detected damage	143
Table 5.8 Confusion Matrix of Neural Network for classification on entire dataset, with damage type classification accuracy wrt detected damage	144
Table 5.9 Confusion Matrix of Neural Network for classification on detected damage points, with damage type classification accuracy wrt detected damage	145
Table 5.10 Damage detection and classification sensitivity wrt variations	149

CHAPTER 1

Introduction

Remanufacturing is the restoration of used products to a like-new condition, providing them with performance characteristics and durability at least as good as those of the original product [1]. It is widely practiced due to its societal benefits from the conservation of resources and reduced waste, as well as economic benefits in terms of reduced costs [1], and is preferred over disposal. Although the terms remanufacturing and repair have been used interchangeably in the industry, the terms are used differently in this thesis, whereby repair refers to the simple process of restoring, while remanufacturing is a macro view of the entire process which not only includes the repair process, but also the inspection and evaluation process. In this chapter, the different types of remanufacturing are explored. The current limitations of remanufacturing are discussed, and automated in-situ remanufacturing is identified as a potential solution, where knowledge from different persons are in-built into an artificial intelligence expert system, used on-site. The various steps for an automated in-situ remanufacturing is surveyed, leading to the identification of the current limitations in technology to progress towards a fully automated in-situ remanufacturing. This forms the problem definition for this research, motivated by the benefits of an automated in-situ remanufacturing, following which the scope of this research work is defined in this chapter.

1.1 Background

1.1.1 Remanufacturing

In most literature, remanufacturing is commonly used to describe the restoration of complex electromechanical and mechanical integrated assemblies, or their cores. This is because these products or cores, when recovered, will have value added to them which is high relative to their market value and to their original cost [2]. This thesis, however, looks at a different form of remanufacturing, which is the remanufacturing of large individual components, typically found in Marine

and Offshore (M&O) structures as shown in Figure 1.1. An example of such component is the rack of a jack up oil rig, which will be worn from extended use (see Figure 1.2). This is because, these components, when restored to a like-new condition with performance characteristics and durability at least as good as those of the original component, will have substantial value added to them as well. Apart from the cost reduction benefits of remanufacturing such components, there is also the advantage in terms of increased efficiency in material use and reduced turnaround time, and there is industrial interest in remanufacturing such large components. As with traditional remanufacturing, there are limitations in remanufacturing, although there are additional limitations specific to remanufacturing single large components.

Figure 1.1: Marine and Offshore structures, adapted from Moan [3]

Figure 1.2: Worn rack of a jack up oil rig

1.1.2 Remanufacturing Limitations

While remanufacturing is not a new concept, there are still areas of improvement in today's context. One limitation of remanufacturing is the high dependency on skilled labour, from the damage detection, analysis, and repair decision, to the manual welding of the damaged part [4]. As the cost of such skilled labour is ever increasing, coupled with labour shortage, remanufacturing would greatly benefit from automation [5]. The transfer of workers from the operators of machineries to supervisory role would increase the workers' safety. Automation also increases the quality of remanufacturing in general, instead of relying on the expertise of skilled labour, of which results may differ between skilled individuals [6, 7]. By eliminating human errors or discrepancies, there is reduced defect rates, and greater uniformity and conformity to quality specifications, which will help to ensure accountability and compliance with statutory requirements. Additionally, automating remanufacturing can also reduce the turnaround time and the material used as compared to manual remanufacturing, as the damage can be more specifically and accurately repaired [7]. This translates to cost savings in the remanufacturing process too.

Another aspect of remanufacturing that can be improved is the widely adopted off-site remanufacturing factory model, where large amounts of damaged similar parts are transported to the remanufacturing facility to be repaired, and sent out again [8]. However, this model is not feasible for our focus of remanufacturing,

where the parts are typically large, like that found in marine and offshore structures (see Figure 1.1). To disassemble and transport large parts from such structures to an off-site remanufacturing factory is time-consuming, expensive, labour intensive, and potentially dangerous [9, 10]. Additionally, some parts are impractical to remove due to physical constraints, requiring excessive structural changes, like the crankshaft crankpin journal in a ship [11, 12]. As such, remanufacturing should be carried out in-situ instead. In-situ remanufacturing would also prevent the need to disassemble and reassemble the part, thus reducing the turnaround time [13]. This is important, as the delay in operations caused by the part being unserviceable is extremely costly [14]. Additionally, parts from the marine and offshore industry are typically exclusive, or unique, and does not benefit from the consolidation of similar cores, or damaged parts, in an off-site remanufacturing factory. It is thus advantages to have automated in-situ remanufacturing, which involves little human-computer interaction.

1.1.3 Remanufacturing Process Flow

To understand how remanufacturing can be automated and carried out in-situ, the various steps within the remanufacturing process have to be studied. Traditionally, remanufacturing comprises of the disassembly, inspection, refurbishing, reassembly, and final testing of worn durable products [1]. However, these steps could be reordered or omitted as necessary [8]. While the process flow of traditional remanufacturing has been widely documented, the traditional way of looking at remanufacturing does not encompass the full potential of automated in-situ remanufacturing, and there is no consistent consensus of what that might be. The process of such remanufacturing, especially in the context of the marine and offshore industry, is not well documented, and predominantly based on the experience of the person performing the remanufacturing. The review of literature dictates that some formalism of this aspect of work is warranted, as it would not only improve consistency, improve quality, improve efficiency and reduce cost, but also increase accountability, tracking and traceability. Hence, a process flow for automated in-situ remanufacturing is proposed and detailed in Chapter 3, as shown in Figure 1.3. An automated in-situ remanufacturing would

thus be possible if the various steps and sub-steps are automated and carried out on-site, and are introduced in the subsequent sections.

Figure 1.3: Automated in-situ remanufacturing process flow overview

1.1.3.1 Cleaning

The first procedure that the damaged part must undergo is the cleaning process, to remove any dirt or debris that is accumulated during the use of the product, as any foreign material on the part would affect subsequent processes. For this step to be automated, some vision system must be included to conduct the data acquisition of the presence of any foreign material on the surface, to ensure successful cleaning of the part.

1.1.3.2 Inspection

After cleaning the damaged part, the first major step in the automated in-situ remanufacturing process flow is the inspection step. As a 3-dimensional (3D) model of the damage region is required for the automated repair step downstream, it is resource efficient to utilise 3D scanning for the data acquisition sub-step for inspection, instead of other methods like imaging.

3D Scanning

ASTM, an international standards organization formerly known as American Society for Testing and Materials that develops and publishes voluntary consensus technical standards for a wide range of materials, products, systems, and services, defines 3D scanning as ‘a method of acquiring the shape and size of an object as a 3-Dimensional representation by recording x,y,z coordinates on the object’s surface, and through software, the collection of points is converted into digital data’ [15]. The input is a physical article, while the output is the digital spatial information of the article, usually in the form of Point Cloud Data (PCD). 3D scanning evolves from ranging technologies. The traditional applications of

3D scanning include inspection and Reverse Engineering (RE) purposes, to create new designs, modify existing designs, design large items, and repair worn parts [16]. Non-engineering applications of 3D scanning have also been emerging, from clothes fitting [17], identifying facial expressions [18], to collection of biometric data [19], and sporting applications [20]. The increased utilisation of 3D scanning would give rise to the development of new 3D scanning techniques, such as the 12-camera close-range photogrammetry [21], Magnetic Angular Rate Gravity (MARG) sensor alignment and distortion compensation methodology [22], Small-Angle X-ray Scattering (SAXS) method [23], and Dip Transformation [24].

With so many 3D scanners utilising different 3D scanning technologies available, an appropriate 3D scanner has to be selected for the digitisation of the damaged part. However, different scanning technologies would have different strengths and weaknesses, and the results would differ between different applications. Additionally, a question could be raised pertaining to the technical specifications provided by the scanners' manufacturers. This is because the technical specifications are derived based on their own experiments under conditions defined by them, and may not be the same conditions in an actual scanning process by consumers. Based on prior literature, research has been done to compare 3D scanners. However, these prior researches are carried out on limited scanners for selecting a scanner for a specific application, not including the digitisation of damaged region. Hence, this research work includes the study of 3D scanners, for the selection of an appropriate 3D scanner for digitisation of damaged components. A case study was also conducted, and results presented.

On top of providing useful information for the repair step, the PCD obtained from the 3D scanner can allow the damage information to be recorded for documentation purposes, of which the reliability of a repair job over time can be studied. However, before the PCD can be used in the remanufacturing process, the raw PCD would first have to be pre-processed for removal of noise and alignment, following which the useful information can be extracted from it.

Damage Analysis

With the collected and cleaned PCD from the 3D scanning step, the next sub-step is to extract useful information, including the damage analysis. To be able to analyse the damaged region, the damaged region has to first be automatically identified from the PCD. While there are currently algorithms proposed by prior literature for automated damage detection from PCD, these algorithms are unable to detect different damage types, have assumptions on the geometry of the undamaged part, or require additional inputs. As such, a more robust framework for damage detection from PCD would be beneficial to an automated in-situ remanufacturing and is also a focus of this research work. With the detected damage region, the damage can then be classified according to the type of damage.

As-built Model Reconstruction

On top of the information on the damaged region, the as-built model of the damaged part is also required before a decision on the reparability of the part can be made. In the best-case scenario, a Computer Aided Design (CAD) model of the as-built part is available, and can be used. However, there are cases where the as-built CAD model is unavailable [25, 26], or unusable due to the acceptable distortion of worn parts [26]. As such, the undamaged model has to be reconstructed from the PCD of the damaged part, also known as nominal model reconstruction.

Repair Decision

With information on the damaged region, together with the undamaged model, a decision can be made on the feasibility of the part being repaired and remanufactured. In cases where it is not feasible, for example where the cost of remanufacturing far exceeds the cost of replacing the part entirely, the part is to be scrapped and recycled instead.

1.1.3.3 Repair Process

The next step in the remanufacturing process is the repair step, which refers to the steps taken to restore the damaged part to the original as-built state. The key enabling technology for this process to be automated and carried out in-situ, is

the advancement of additive manufacturing, and the development of hybrid manufacturing machines.

Additive Manufacturing

Traditionally, remanufacturing of components is done manually via conventional welding by skilled workers. Although conventional welding is a form of additive manufacturing, it is plagued with an array of problems, like the low accuracy and reliability, large part deformation, and poor bonding between the filler and damaged part [7]. On the other hand, newer additive manufacturing technologies, specifically the use of Directed Energy Deposition (DED) techniques, has had more promising results [27]. Furthermore, the use of DED techniques also allows for the control of microstructure of the repaired segment, inclusive of the producing of single crystal structures [28], which may be required for some repair applications. DED techniques are thus the preferred choice for remanufacturing.

Hybrid Manufacturing

Hybrid manufacturing is a vague term used to describe the combination of different manufacturing processes. The purpose of hybrid manufacturing is to enhance the advantages of the different processes, whilst reducing the disadvantages of each process [29].

There are many definitions of hybrid manufacturing, including manufacturing by different processes [30], use of different forms of energy [31], or combination of active principles [32]. The definition adopted in this thesis is that of Rivette, Hacoët [30], and is defined as manufacturing by different processes, usually, rapid prototype process, or additive manufacturing, and conventional process, or subtractive manufacturing.

For the specific application of repair, there is a need to add material as the damages usually include material loss, and hence additive manufacturing is required. However, additive manufacturing has the limitation of low accuracy and speed [33]. The solution to this limitation is the combination of conventional milling processes with additive manufacturing in a hybrid manufacturing setup, as conventional milling processes are able to achieve higher accuracies and speed [34]. Additionally, the damage surface may not be optimal for additive

manufacturing machines to build upon. Instead, the conventional milling component of hybrid manufacturing machines can help to smoothen out the damaged region, to create a suitable surface for the additive manufacturing machine to build upon.

While there are a number of hybrid manufacturing machines available today, and many more in the research and development phase, the preferred configuration to integrate the technologies for this research is that of an industrial robotic arm with Automatic Tool Change (ATC) functionality to switch between the additive manufacturing and subtractive manufacturing heads, as well as the possibility to automatically switch in a 3D scanner, as shown in Figure 1.4. Such machines are more mobile as they are smaller than gantry type machines and are able to have a flexible reach.

Figure 1.4: Hybrid manufacturing machine tool configuration

To employ hybrid manufacturing in the remanufacturing process, there is a need for the repair process planning sub-step. This step is to determine the order of the use of the additive and subtractive manufacturing techniques to clean up the damage region, add material, and reduce it down to the functional dimensions. The output of this step would be a series of generated toolpaths for the different manufacturing techniques, which may be simulated before the actual repair process is carried out.

1.1.3.4 Evaluation

After the damaged part has been repaired, the final step of remanufacturing is to evaluate the repaired part, before it can be certified to have a like-new condition, providing them with performance characteristics and durability at least as good as those of the original product, as per the definition of remanufacturing [1]. This

is done via data acquisition and analysis. As 3D scanners are already employed in the automated in-situ remanufacturing process, it can be reused for the data acquisition step for the evaluation process. The acquired PCD can then be compared with the nominal model, to see if the repaired part is within specified dimensions. Additionally, other Non-Destructive Testing (NDT) methods commonly used in the welding industry, like ultrasonic testing, can be employed to verify that the integrity of the internal region of the repaired part is up to the required standard, before the remanufactured part can be certified for its functionality.

1.2 Problem Definition

To harness the societal and economic benefits of remanufacturing, there is an emerging necessity for automated in-situ remanufacturing, to reduce the labour related problems due to the labour intensiveness of traditional remanufacturing via automation, and to remanufacture parts typically too large to transport cost efficiently, or impractical to remove due to physical constraints [9-12], by carrying out remanufacturing in-situ. To be able to implement such an automated in-situ remanufacturing for the marine and offshore industry, there are a number of challenges that has to be addressed. The first issue faced is that the process of such remanufacturing, especially in the context of the marine and offshore industry, is not well documented, and predominantly based on the experience of the person performing the remanufacturing. The review of literature dictates that some formalism of this aspect of work is warranted, as it would not only improve consistency, improve quality, improve efficiency and reduce cost, but also increase accountability, tracking and traceability. Thus, a process flow for automated in-situ remanufacturing has to be established.

To be able to achieve automated in-situ remanufacturing, work also has to be done to improve upon individual procedures within the remanufacturing process. While there are many areas of research that has to be done to advance automated in-situ remanufacturing, this research will focus on problems within the inspection step of remanufacturing, namely the difficulties in the collection and pre-processing of necessary data, and the lack of a robust framework for automated damage detection and classification from 3D PCD.

Remanufacturing can be used to refer to the restoration of products on different levels, from the product as a whole, to sub-assemblies, to individual components. However, this research will be scoped towards the remanufacturing of metal engineering components as part of the whole assembly, of that typically found in the marine and offshore industry, including the rack and pinion of a jack up rig. Here are some of the associated issues with marine and offshore parts:

- Large parts [35]
- Lack of documentation [25, 26]
- Not designed for easy disassembly and assembly [35]
- Delays in operations due to unserviceable part is extremely costly [14]

Nonetheless, the findings of this research should be easily adaptable to other components, as well as other engineering sectors.

1.3 Motivation

The societal and economic benefits of remanufacturing are well documented, though there are still much areas of improvements in remanufacturing to further harness those benefits. By automating remanufacturing, there would be a decrease in dependency on skilled labour, increased safety, reduced turnaround time and material used, as well as increased repeatability and reliability, all of which would lead to cost savings in the remanufacturing process. In addition, there is a need to carry out the remanufacturing in-situ for industries where parts are typically too large to transport cost efficiently to off-site remanufacturing factories, or impractical to remove due to physical constraints, like that found in the marine and offshore industry [9-12]. This would omit the need to disassemble and reassemble, thus further reducing labour dependency, turnaround time, and costs. Hence, it is beneficial to automate remanufacturing, and carry it out in-situ. The need for accountability, tracking and traceability is also necessary for remanufacturing, and can be achieved with the formalism of an automated in-situ remanufacturing framework, with the additional benefits of improved efficiency and reduced cost. With these motivation in mind, the research work done for this thesis is to establish a framework and improve upon enabling technologies to support a fully automated in-situ remanufacturing.

1.4 Research Goals

The overall aim of this work is to address some technological gaps required to attain full automation of in-situ remanufacturing. The following objectives were identified to be addressed for this thesis:

- ***To establish a process flow for automated in-situ remanufacturing.*** Although there are many proposed process flows for remanufacturing available, there is little literature on the necessary processes or process flow for an automated in-situ remanufacturing. This process flow would be beneficial in identifying current limitations in attaining full automation for in-situ remanufacturing.
- ***To explore and implement PCD based damage detection algorithms.*** The state-of-the-art PCD based damage detection algorithms have not had the same success rate as image-based damage detection. As such, current algorithms will be reviewed to ascertain which algorithms could be used, while other algorithms using novel strategies for damage detection are to be proposed.
- ***To explore and acquire the data necessary for experimentation.*** Due to the lack of access to sufficient amounts of physical damaged parts, synthetic data is to be created for the training and demonstration of the model. Real world data is then used for the validation of the proposed approach, and is to be acquired by 3D scanners. The selection of an appropriate 3D scanner for reliable PCD is thus studied, with a novel classification method proposed to select an appropriate 3D scanner. Additionally, the necessary data pre-processing procedures were studied.
- ***To formulate and implement a framework for an accurate PCD based damage detection and classification.*** Based on the literature conducted, existing PCD based damage detection have assumptions and limitations, preventing them from being implemented in damage detection of differing damage types, and parts. As such, a framework should be formulated to ensure robust automated damage detection from PCD across damage types and parts.

While there are many other technological gaps for attaining full automation for in-situ remanufacturing, this project covers multiple broad research areas, and I have chosen to focus my research on the earlier portions of the remanufacturing, which is the digitisation, damage detection, and damage classification sub-steps of the inspection step, in the remanufacturing process flow.

Specific to the damage detection explored in this thesis, there may be internal damages within the structure, but is regarded as a separate problem. This is because the repair methodology differs greatly, as the internally damaged section typically has to be replaced altogether, as it is not feasible to repair via current additive manufacturing technologies. Such internal damages are also detected using well established NDT techniques, instead of the digitising technique of optical 3D scanners as presented in this thesis. Nonetheless, research work may be done in the future to combine the remanufacturing of components containing both internal and external damages.

1.5 Original Contributions

The following original contributions are claimed by this work:

- ***Classification framework for the selection of appropriate 3D scanners.*** A novel classification framework for 3D scanners was proposed to aid in the selection of an appropriate 3D scanner for any application, to overcome the limitations of current 3D scanner selection methodologies.
- ***Framework for automated in-situ remanufacturing process.*** The process of automated in-situ remanufacturing, especially in the context of the marine and offshore industry, is predominantly based on the experience of the person performing the remanufacturing. From the literature review, there is no framework available specifically for automated in-situ remanufacturing, and the traditional way of looking at remanufacturing does not encompass the full potential of automated in-situ remanufacturing, and there is no consistent consensus of what that might be. As such, an original framework for automated in-situ remanufacturing process is proposed.

- ***Multiple novel PCD based algorithms for specific damage detection.***
The proposed approach of using damage detection algorithms for feature extraction meant that novel algorithms based on specific damage characteristics could be implemented to improve the machine learning based damage detection framework.
- ***Indirect application of machine learning on geometric data for damage detection.*** Machine learning has been applied on geometric data by other researchers, but their application was either for object recognition or semantic segmentation, and not for the purpose of damage detection. The indirect application of machine learning on geometric data for damage detection represents the final claim of contribution, with no other similar application in literature based on the author's knowledge

1.6 Structure of the Report

Chapter 1 has defined the type of remanufacturing this research focuses on. The general benefits of remanufacturing are presented, together with the additional benefits available if the process is automated and carried out in-situ. The various steps for an automated in-situ remanufacturing is explored, leading to a discussion on the current limitations in technology to progress towards automated in-situ remanufacturing. This forms the problem definition of this research, and the research is motivated by the potential reduction in the dependence on skilled labour, cost, and turnaround time. The research goals are also identified, together with the original contributions claimed by the author.

Chapter 2 reviews the current state of the art of remanufacturing, from remanufacturing as a whole, to the process flow, to the individual processes within the scope of this research, namely the cleaning and inspection steps. More specifically, the current literature on damage data collection, 3D scanner selection, point cloud pre-processing, surface damage detection and classification from PCD is discussed in this chapter. A study on the 3D scanner selection was conducted to choose an appropriate 3D scanner for use in this research, and is presented in this chapter. Finally, the technological gaps in the current state of the art is identified, forming the potential research areas of this thesis.

Chapter 3 presents the proposed automated in-situ remanufacturing process flow, based on the motivation of reduced dependency on skilled labour, cost, and turnaround time. Individual steps and sub-steps of the proposed process flow are elaborated, along with modifications required for the steps to be automated and carried out in-situ. With the proposed process flow, the critical technology necessary for automated in-situ remanufacturing are identified, leading to the work in the subsequent chapters.

Chapter 4 describes the work that has been done to work around the unoptimized computational efficiency for geometric learning. The chapter begins by first defining the scope and prerequisites for the automation of damage detection from PCD. Next, the approach for dimensionality reduction is detailed. The common functions necessary for the various feature extraction steps are introduced, following which 11 feature extraction approaches are presented. The algorithms that were either adopted or novel were explained in detail, and were tested on synthetic PCD.

Chapter 5 documents the implementation of the automated damage detection and classification from 3D data. The chapter begins by presenting the creation, collection, and annotation of data necessary for this research. Next, the extracted features from Chapter 4 were used to train machine learning models for damage detection. Various machine learning models were compared, following which the most accurate model was validated on real scans of damaged components, with results presented and discussed. The proposed approach is also extended to the classification of damage types, with results presented and discussed.

Subsequently, this report concludes in Chapter 6, with the summary of work done, contributions, potential further works, as well as some concluding remarks.

CHAPTER 2

Literature Review

In Chapter 1, the benefits of having an automated in-situ remanufacturing is identified as the motivation for implementing it. In Chapter 2, the current technologies necessary for such an automated in-situ remanufacturing are studied. To understand the specific limitations of attaining automated in-situ remanufacturing, the entire process should be known and defined. Thus, Chapter 2 begins by reviewing various types of remanufacturing, and the process flow of traditional remanufacturing, to understand the steps involved in remanufacturing. This understanding would be used in formulating a process flow for automated in-situ remanufacturing in the subsequent chapter. With an overview of the remanufacturing process flow, the current state of the art of the inspection step is reviewed next in this chapter. Prior literature documenting research on the specific sub-steps within the inspection process is reviewed, including the damage data collection, 3-dimensional (3D) scanner selection, point cloud pre-processing, surface damage detection, classification, and machine learning from Point Cloud Data (PCD). Technological gaps from the critical review of the current state of the art is identified, forming the potential research areas of this thesis.

2.1 Remanufacturing

Much research has been done on remanufacturing, specifically research on design for remanufacturing, where products are made to be simpler to remanufacture [36-38]. Additionally, the remanufacturing process has been augmented by advanced information technology and management technology, to increase flexibility, predictability, and to optimise the control for remanufacturing, and is known as flexible remanufacturing [39]. However, there has been significantly lesser research on simplifying the remanufacturing process instead. Hence, this research first focuses on the improvement of remanufacturing, from the process flow standpoint, to attain automated in-situ remanufacturing. Additionally, as defined in the introduction, the scope of remanufacturing is on a single large

component, of which the remanufacturing process flow would slightly differ from that of remanufacturing a complex assembly.

2.1.1 Process Flow

Traditionally, remanufacturing consists of 5 basic steps, namely disassembly, cleaning, inspection, repair process, and reassembly [40]. Schematics of the complete process based on these 5 steps is shown in Figure 2.1. However, due to the different requirements by different industries, there may be additional or fewer steps in the process, and the steps may be reordered and repeated. For example, the cleaning step is not required for the remanufacturing of a defective part, as compared to a used part [41]. As such, a generic remanufacturing process was proposed by Sundin [8], as showed in Figure 2.2. However, these ways of looking at remanufacturing does not encompass the full potential of automated in-situ remanufacturing, and there is no consistent consensus of what that might be. From the literature search conducted, there has been no prior literature specifically documenting the process flow of automated in-situ remanufacturing, and the process is predominantly based on the experience of the person performing the remanufacturing. Nonetheless, it is important that the process flow of automated in-situ remanufacturing is established, so that the required steps and procedures can be identified and reviewed, following which limitations in the steps can be identified and tackled. Hence, this problem of a lack of process flow is addressed in Chapter 3, where a process flow for automated in-situ remanufacturing is proposed.

Figure 2.1: Traditional remanufacturing process flow

Figure 2.2: Generic manufacturing process, proposed by Sundin [8]

2.2 Cleaning

From the proposed process flow for automated in-situ remanufacturing as shown in Figure 1.3, the first step required is the cleaning step. This is especially important for marine and offshore applications, as the harsh conditions would lead to corrosion, oil and debris build up. The presence of such externalities would be digitised in the inspection step, and the captured build up would be mixed with the actual part data [42], leading to subsequent errors. A clean part would also lead to improved mechanical and corrosion performance after the additive manufacturing repair [43], and hence should be carried out. Some contaminants required to be cleaned, as presented by Liu, Zhang [44], are as shown in Table 2.1.

Table 2.1: Contaminants in remanufacturing, adapted from Liu, Zhang [44]

Contaminant		Components	Characteristics
External deposits		Dust, oil sludge, etc.	Easy to be removed, but difficult to clean thoroughly.
Residual lubrication oil		Aged viscous oil, water, salt, product of corrosion and deterioration on parts surface.	Complex components, furring, which determines corresponding cleaning methods.
Carbonised Sediment	Carbon deposition	Carboid and carbides, lubricating oil and tar, a handful of oxygen acid, ash content, etc.	Most of the components are insoluble, which are difficult to be removed.
	Film like painting	Carbon.	Low strength, easy to be removed.
	Other types of sediment	Lubricating oil, tar, a handful of carboid, carbides and ash content, etc.	Most of the components are insoluble, which are difficult to be removed.
Scale deposit		Calcium salt and magnesium salt, etc.	Soluble in acid.
Rust		Iron oxide, alumina, etc.	Insoluble in water and alkali, and soluble in acid.
Detection residues		Metal debris, perspiration, and fingerprint, etc.	Adhesive force is small, and easy to be removed.
Residues of machining		Metal debris, polishing paste, abrasive paste, lubricating liquid, coolant liquid, etc.	Adhesive force is not large, but need to clean thoroughly.

Based on prior literature, there are numerous cleaning technologies which can be used in remanufacturing, including the following:

- Organic solvent cleaning [45]
- Water jet cleaning [46]
- Abrasive blasting cleaning [47]
- Dry ice blast cleaning [48]
- Thermal cleaning [49]
- Ultrasonic cleaning [50]
- Electrolytic cleaning [51]
- Ultraviolet radiation cleaning [52]
- Plasma cleaning [53]
- Ion beam cleaning [54]
- Laser cleaning [55]
- Supercritical fluid cleaning [56]
- Biological enzyme cleaning [57]

The more common strategies and their associated application can be found in Table 2.2.

Table 2.2: Common cleaning strategies, adapted from Liu, Zhang [44]

Cleaning medium	Decontamination force	Major ingredient	Applicable contaminants
Solvent	Dissolving force and dispersion force.	Organic solvent.	Oil and grease.
Aqueous based cleaning fluid	Adsorption, wetting, permeating, emulsifying, scattering and solubilizing, etc.	Surfactant, detergent, additive agent, etc.	Various contaminants adsorbed on core surface.
Chemical solvent	Chemical reaction force.	Acid solution, alkali solution, oxidizing and reducing agents, etc.	Rust, saponified grease, etc.
Solid particle	Abrasive force, impact force.	Various solid particles.	Rust, paintcoat, etc.

The field of cleaning technologies is thus well established, and the necessary cleaning techniques should be selected based on the contaminant on the damaged part, on a case by case basis.

2.3 Inspection

Once the part has been cleaned using the appropriate techniques, the next step in the remanufacturing process is the inspection step. This includes the sub-steps data acquisition, damage analysis, as-built model reconstruction, and repair decision. As mentioned before, while there are many areas of research that has to be done to advance automated in-situ remanufacturing, the scope of this research is limited to the sub-steps data acquisition and damage analysis, and hence the state of the art of only these aspects were reviewed.

2.3.1 Image-based Damage Detection

The first sub-step within the inspection step is the detection of surface damage. Automated surface damage detection is traditionally done by image processing, and is most commonly used for structural health monitoring rather than remanufacturing. There are two approaches for image-based damage detection, the model-based methods, and the non-model-based methods [58]. The model-based approach creates a mathematical model based on the damage, and applying the model to detect damages from images. On the other hand, non-model based methods involves a computer building the model by means of training, such as the use of neural networks.

The first type of model-based method utilises edge detectors, where the damage is detected as an edge. The Sobel edge detector algorithm, based on spatial gradient algorithm, detects changes in image by convolving image data with a Sobel mask given by equations 2.1 and 2.2, of which the middle pixel of the mask results in a first-order partial derivatives [59]. The gradient magnitude independent of direction can then be calculated using equation 2.3. This approach, whilst known for its speed and simplicity, is sensitive to image noise.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad (2.1)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \quad (2.2)$$

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.3)$$

Another common edge detector used for detecting damages is the Canny edge detector, proposed by Canny [60], and is also a convolution filter which first convolves the image with a Gaussian mask to smooth the image and remove noise. The intensity gradient of the image is then computed, followed by a non-maximal suppression of the gradient to identify potential edges. A double threshold is applied to determine potential edges, followed by the tracking of edges via hysteresis. The detected edges can then be identified as cracks on the surface, as shown in Figure 2.3 [61].

Figure 2.3: (a) Damage; (b)Canny edge detector for detecting damage, adapted from Chen and Hutchinson [61]

Other edge detectors that have been used in image-based damage detection is the Fourier transform edge detector, which is a discrete transform based on frequency, where the Fourier transform is applied to determine the frequency content of an image [62]. The local maxima in the frequency space is used to identify edges in the image, or potential crack damages. An improvement of this method is the fast Fourier transform, which has taken on various forms [63, 64].

The fast Haar transform, based on the Haar wavelet [65], is also being used for edge extraction [66]. This is done by separating the image into the high and low frequency components, of which the edge features of the image are identified from the high frequency components by averaging, differencing, scaling, and thresholding. Whilst there are many other variants of image edge detectors, these detectors are unable to detect damages without distinct edges, and are thus limited in their scope.

A different approach for damage detection was proposed by Wiseman [67], where damage is detected from images based on the detection of object borders, by analysing the bit sequence length of the original JPEG image blocks within the compressed file (see Figure 2.4). The long bit sequences are presumed to be

object borders, or the damage area for his application. However, the limitation of this method is that the image must be a close picture of the damaged part.

Figure 2.4: Detected tyre damage from JPEG images, adapted from Wiseman [67]

These are just some of the model-based image-based damage detection proposed by other researchers. While they were the state-of-the-art during their time of publication, or are successful for their narrow application, model-based approaches for image-based damage detection is declining in popularity, and is being replaced by the non-model approaches, including machine learning. Machine learning, defined by Samuel [68] as “without being explicitly programmed”, is being used in image-based damage detection by giving sample data of images of damage to a machine learning algorithm, also known as training data, to make predictions on whether subsequent images are of damage or not.

Ho, Kim [69] proposes the detection of damages via the use of Principal Component Analysis (PCA) algorithm. After filtering the images, the filtered images are projected onto the PCA sub-space to calculate the distances of the input image data to all the training pattern images, where the smallest distance is found to be a match for the input image, identifying if the image contains a damage or not (see Figure 2.5).

Figure 2.5: Damage detection via PCA, adapted from Ho, Kim [69]

Cha, Choi [70] utilises a deep architecture of convolutional neural networks (CNN) for detecting damages without calculating the defect features (see Figure 2.6), with about 98% accuracy. However, the approach proposed by them is used only for the specific application of crack detection.

Figure 2.6: Damage detection via trained CNN, adapted from Cha, Choi [70]

Cha, Choi [71] also adopts similar approach, but using the faster Region-based Convolutional Neural Network (R-CNN) instead for visual inspection and damage detection, and demonstrated the approach on different materials and damage types (see Figure 2.7).

Figure 2.7: Detected steel damages from images using deep learning, adapted from Cha, Choi [71]

These are just some of the available research utilising machine learning on the image-based damage detection, which has demonstrated the efficiency and robustness of the use of machine learning. However, for our application of remanufacturing, it is beneficial to be able to detect the damage from 3D data as the downstream manufacturing processes requires it. Additionally, 3D data could potentially provide more information of the severity of damage, like depth [72]. Hence, the next few sections review the state-of-the-art for 3D data acquisition and damage detection. Nonetheless, the merits of machine learning, as demonstrated in the image-based damage detection literature, was the reason for this research attempt to apply machine learning indirectly onto the 3D data for damage detection.

2.3.2 3D Data Acquisition

For the automation of 3D damage detection and identification, some data has to be collected, which will enable a computer to do the damage identification and classification. This data has traditionally been collected using various methods, including imagery, ultrasound [73, 74], thermal imaging [75, 76], X-radiography, and 3D scanning. However, for the purpose of automation of remanufacturing, a 3D model is required for the hybrid manufacturing machine to work with. Hence, the use of 3D scanner for damage detection would prevent redundancy in the need for another data collection method. Currently, there are numerous commercially available 3D scanners, and researchers have used them to digitise damage, for the purpose of documentation and assessment, for aerospace [77, 78], automotive [79], medical [80, 81], buildings [74, 82, 83], marine and offshore [84, 85], mould and die [25], furniture [86], wood [87], and art pieces [88]. The

extensive adoption of 3D scanners for the purpose of digitising damaged parts demonstrates the capability of modern 3D scanners, thus supporting our choice of utilising 3D scanners in the remanufacturing process.

2.3.3 3D Scanner Selection

3D scanner is an important tool for the inspection step in automated remanufacturing, and for automated damage detection. Although there are both contact and non-contact 3D scanners, the contact type 3D scanners are extremely slow as compared to the thousands of points per second speed offered by the non-contact type 3D scanners, and is more suitable for inspection, hence is the preferred scanner type studied. However, there are many different non-contact 3D scanners available, and an inappropriately selected 3D scanner would result in poor results obtained, which would cause issues in the damage analysis sub-step, as well the repair process. Although 3D scanner selection is seldom regarded as a problem as most potential users select 3D scanners based on the provided technical specifications, a question could be raised pertaining to the degree of sufficiency of the technical specifications typically provided, with regards to specific applications, such as the scanning of challenging objects.

As such, multiple researchers have attempted to evaluate 3D scanners for specific applications [89-95]. However, the results obtained by these researchers are only relevant if the 3D scanner user's application is the same as that being studied. Additionally, the scanners compared are limited, and may not include other scanners available to potential users. Thus, a more generic methodology for 3D scanner selection would be beneficial.

The only relevant literature found in the area of 3D scanner selection that is not restricted by application was by Vezzetti [96], who developed a Smart 3D scanner selection methodology to tackle this issue. By integrating the morphological analysis of the object acquired with the customer needs and the 3D scanner performances, users are able to identify an optimal solution, or the most appropriate 3D scanner. A flow chart of the proposed methodology is as shown in Figure 2.8. The proposed approach analyses the sampled surface by generating the preliminary point cloud, creating the pitch map from the curvature analysis, followed by the preliminary grid improvement. The scanner is also evaluated in

parallel, by taking the upper and lower boundary curvature definition values, and evaluating the scanner capability area to obtain the scanner pitch function. This process is iterated in a control cycle with different scanners, until the optimal pitch function is obtained, which represents the most suitable scanner for the specific scanning task.

Figure 2.8: Smart 3D scanner selection methodology flow-chart, adapted from Vezzetti [96]

Although the methodology proposed by Vezzetti does help users identify an optimal 3D scanner for their specific application, the methodology requires the user to test the scanner for comparison. This means that potential users are unable to benefit from his proposed methodology, unless they have prior access to the potential scanners. In addition, a more intuitive specification with regards to the time taken and ease of use is also not provided in standard specifications, nor is it considered by Vezzetti's selection methodology, but should be incorporated as it will be beneficial to the user. Hence, the lack of supporting tools to assist potential users in choosing an appropriate 3D scanner is a gap in knowledge, and a preliminary study was conducted.

The technical specifications typically provided by 3D scanner manufacturers are first identified. Next the potential sources of error, which might lead to the

scanner's inability to attain the provided specifications, are studied, with a focus on the scanned object-related errors. The different functionalities of different scanning technology were then broken down and analysed, based on how they were able to handle the object-related challenges. This was then summarized in the form of the proposed Morphological Box Classification Framework. The proposed framework is subsequently tested in a case study, and the results presented.

2.3.3.1 Key Scanner Specifications

Manufacturers of commercially available scanners do provide technical specifications, or a data sheet, explaining of the scanners' performance. This includes the accuracy, resolution, scanner working limit, and scanner speed.

Accuracy

Accuracy is defined as the closeness of agreement between measured and actual value [97]. A high accuracy is generally preferred, as the scanned 3D PCD would be more reliable in representing the physical object scanned.

A similar specification is the precision, which is defined as the closeness of agreement between different scans of the same object [97]. When scanning, the measured points are preferred to be in a close agreement to the actual value, hence accuracy is the more relevant specification. Additionally, by definition, and scanner can never have a poor precision but good accuracy, and hence precision is not an important specification.

Resolution

Resolution is used to describe the spacing between adjacent points attained in the 3D PCD. An alternative definition of resolution is the number of points that can be measured on a surface. This would determine the smallest feature size of the scanned object that can be captured by the 3D scanner.

Scanner Working Limit

The scanner working limit is the range or area which the scanner can capture the data. The scanners are either limited by the power of the wave used, for example

the laser power, or physical limitations, like the reach of an articulate arm mounted laser scanner.

Scanner Speed

Scanner speed is the speed at which the PCD can be acquired, usually in terms of points per second. Unfortunately, this does not translate to how fast a user is able to digitise a part, as it is affected by settings like resolution, and is also affected by the scanning procedure used.

2.3.3.2 Sources of Errors

While the specifications provide a report of the performance of the scanners, the results are based on tests conducted by the manufacturers themselves. This means that the performance results are only reproducible under the same conditions, and may not hold for all situations. This could be due to errors involved in the scanning, which may be either instrumental, object-related, environmental, and methodological errors [98]

Instrument-related Errors

Instrumental errors are errors related to the 3D scanner itself. This could be fundamental errors inherent to the physics of the energy source used, or hardware related [99]. Such errors have been taken into account in the provided specifications, as it is constant, unless there is hardware changes or damages. The change in specifications due to instrumental errors can be kept in check by regular maintenance and calibration.

Object-related Errors

Object-related errors refer to errors induced due to the properties of the scanned object, and the interaction between the scanner and the object. A generalised optical 3D scanner consists of an optical source and a receiver (see Figure 2.9(a)). By processing the reflected optical wave, the 3D scanner is able to locate in a 3D space the point on the object reflecting the optical energy source, and hence return the point cloud after a series of points are captured. However, the reflected optical wave captured by the receiver may be plagued by one of the following problems, and lead to erroneous results.

Figure 2.9: Reflection off various surfaces (a-e)

Black/dark-coloured surfaces

Black or dark-coloured surfaces have the tendency to adsorb a large proportion of the optical energy, instead of reflecting it (see Figure 2.9(b)). This leads to the scanner's receiver receiving little or no reflected optical wave, and hence is unable to accurately compute the location of the point.

Transparent surfaces

Transparent surfaces allow the optical wave to pass through it (see Figure 2.9(c)). As such, there is little or no reflected energy, and causes the same problem as black or dark-coloured surfaces.

Translucent surfaces

Translucent surfaces will refract most of the optical wave. This forms a volume of scattered light beneath the surface, with the centroid of the volume being displaced from the actual point where the optical wave hits the surface (see Figure 2.9(d)). This centroid of this volume is registered by the receiver as the reflected wave due to its higher intensity, leading to a systematic bias in computed depth.

Reflective surfaces

When an optical wave reflects off any surface, there are two forms of reflection, which are specular and diffuse reflection. Specular reflection is the mirror-like reflection of waves, where the incident ray and reflected ray have the same angle to the surface normal. Diffuse reflection is the reflection of waves in many directions, or angles, due to the scattering of the waves. When scanning a typical surface, the receiver usually captures a reflected wave reflected by diffuse reflection, as the receiver is seldom at the same angle away from the surface normal as the optical source. However, in the case of a reflective surface, the specular reflection is the dominant form of reflection, with little diffuse reflection

(see Figure 2.9(e)). This leads to little or no reflected energy captured by the receiver, and cause the same problem as black or dark-coloured surfaces.

Poor accessibility to surfaces

Poor accessibility to surfaces are surfaces where the scanner is unable to scan from a location near the surface normal, and the scanners have to operate from an oblique angle instead. This could be due to self-occlusion in the direction of the surface normal. This causes a problem as at a larger angle to the surface normal, specular reflection becomes more dominant, leading to lesser diffuse reflection taking place (see Figure 2.10). This would lead to little or no reflected energy captured by the receiver, and cause the same problem as black or dark-coloured surfaces.

Figure 2.10: Reflection from Different Directions (a-b)

Combination of the challenges

The real challenge arises when the 3D scanned artefact contains a combination of the above-mentioned problems. For example, to capture the reflected ray off a reflective surface, the scanner could be positioned normal to the surface. However, this is not possible if there is poor accessibility to the surface.

The problems which are due to the surface characteristics, like absorptivity, refractivity, and reflectivity may be avoided by treating the surface an anti-glare spray. The layer formed above the existing surface will have good diffuse reflection properties, negating the issues. Thereafter, the scanned data can be offset based on the thickness of the surface treatment. However, this workaround has its limitations. It is difficult to ensure the consistency in the layer thickness on a complex surface, and would thus introduce errors. Also, surface treatment may not be feasible on some sensitive artefacts. Instead, these problems can be minimised by selecting a 3D scanner with the appropriate technology, as different

3D scanning technologies are able to respond differently to the different challenges.

Environmental-related Errors

Environmental-related errors include errors induced by ambient temperature, pressure, relative humidity, illumination, vibration, etc. [98]. Such errors can be minimised by conducting the scanning in a controlled environment, or having some form of error compensation algorithm [100].

Methodological-related Errors

Methodological errors are errors due to the chosen scanning strategy, or user being inexperienced with the scanning technology. For example, the user might select a sampling resolution much higher than the laser point accuracy, leading to oversampling, and generating of extra noise [101]. The use of the 3D scanner out of the specified range would also result in errors in the scan results. Such errors may be minimised by proper training on the scanning process, or from user experience.

While there are numerous sources of errors, the object related errors detailed in section 2.3.3.2 can potentially be solved by selecting an appropriate 3D scanner.

2.3.3.3 Classification Framework

An optical 3D scanner is a complex device carrying out various sub-functions, towards the main objective of digitising a physical object. These sub-functions may operate differently in different scanners, which leads to the difference in performance of the 3D scanners. However, the sub-functions, or methodology of the scanners, is not explicit. Hence, in order to be able to fully understand the scanner methodology of the various 3D scanners, functional decomposition is carried out. This means that the scanners functionality is broken down into simpler operations, or lower level tasks. By observing existing 3D scanning technologies from an abstract functional viewpoint, the basic functions of the different 3D scanning technologies could be identified. Each basic functions of the 3D scanner are associated to a set of strengths and limitations, due to the nature of the function, and is described in the following sections as well.

Wave Type

The 2 types of wave used in 3D scanners are electromagnetic and mechanical (or physical) wave. Due to the different spectrum of both wave types available and the specific application requirements, there is minimum conclusion that can be drawn from this classification.

Wave Class and Frequency

The type and frequency of the wave determines what class it falls under, and the varying classes have different characteristics, and is hence optimised for different applications.

Radio Wave

Radio waves are commonly used in two areas of 3D scanning, namely RADAR for long distance ranging measurement, and millimetre wave scanning for security body imaging. Due to the long wavelength, scanners utilising radio wave typically has increased range, at the expense of having a lower accuracy.

Infrared

Infrared has similar properties to visible light when used in 3D scanners. Although an infrared source and receiver is costlier than that of visible light, it is an advantage in that it is not visible to humans, and is hence not as distracting as the use of visible light in 3D scanners.

Visible Light

Visible light is the most commonly used wave in 3D scanners, as optical components used in such 3D scanners are cheap and readily available, while the results obtained can be highly accurate. It is thus the most common form of wave used in a scanner. Although the visible light may be distracting to humans in some applications, it may be advantageous as a form of visual feedback of the area being scanned in other applications.

Ultraviolet

Ultraviolet light has similar properties to visible light. However, the use of ultraviolet light is less common than visible light, as it is more expensive, and potentially harmful to our health [102].

X-ray

X-rays are penetrative electromagnetic waves, commonly used in Computer Tomography (CT) scan. Despite being costly and hazardous to health, it has the advantage of being able to 3D scan within objects, as it is able to penetrate through various materials.

Gamma Rays

Gamma rays are the most penetrative electromagnetic waveform, and is commonly used for 3D scanning of cargo for inspection due to the ability to penetrate through large amount of mass [103, 104]. Apart from applications requiring high penetrative power, it is avoided as it is more dangerous and hazardous to humans.

Infrasound

Infrasound ranging is the use of infrasonic sound waves to detect how far a point is, and the series of point can be used to reconstruct the object, or 3D scan. Such a wave is commonly used in long range SOUNd Navigation And Ranging (SONAR) for 3D scanning, and has a relatively poor resolution. Additionally, the setup has to be large to achieve reasonable directionality, which translates to 3D scanning accuracy [105].

Acoustic

Acoustic ranging works in the same manner as infrasonic ranging. Such a wave is also commonly used in SONAR applications, for 3D scanning of shipwreck for example. Compared to the use of infrasound, the resolution and accuracy is improved, but the sound may potentially be distracting to the humans as it is audible.

Ultrasound

Ultrasound is used similarly to infrasound and acoustic waves. However, it has the advantage of being inaudible to humans, and hence less distracting than audible acoustic waves. It is commonly used for 3D scanning of internal cracks in metals, or for medical imaging.

Wave Source

The wave source can be divided into two categories, namely active and passive. An active wave source 3D scanner is one which generates its own electromagnetic or mechanical wave, before analysing it, whereas a passive setup relies on detecting reflected ambient reflection. An active setup is thus preferred as a higher powered wave is less resistance to the capturing of noise, but at the expense of having to provide power for the wave source.

Radiation Property

Radiation property refers to the characteristic of the wave as it interacts with the scanned object. It may either be penetrative, where the electromagnetic or acoustic waveform is able to pass through the object and detected by a receiver on the opposite side, or non-penetrating, where reflections are used. Penetrative 3D scanners are usually more expensive, require higher powers, and are less portable, but provide additional information of the internal structure, where-as non-penetrative scanners are cheaper, but plagued with field-of-view problems like occlusion.

Working Principle

The working principle describes the mechanism the scanner employs to identify the object surface in space.

Time-of-flight

Non-penetrating

Non-penetrating time-of-flight 3D scanners work on the principle where a reflected wave pulse emitted by a source is captured by the receiver, and the time

difference between the source emission and sensor detection is used to calculate the distance of the point from the scanner, as shown in Figure 2.11.

Figure 2.11: Non-penetrating Time-of-Flight 3D scanner schematic

The distance of the point from the scanner is described by Equation 2.4 below:

$$distance = c \left(\frac{t}{2} \right) \quad (2.4)$$

where c is the speed of light, and t is the time difference between the source emission and sensor detection. The accuracy of such wave is limited by the accuracy of the sensor and timer, and hence is more commonly used for long range applications.

Penetrating

Penetrating time-of-flight 3D scanner works on the principle that the wave will travel at different speeds between air, and other material. When an object is put between the sensor and the source, the time taken for the wave to reach the sensor is reduced if an electromagnetic wave is used, but increased if a mechanical wave is used (see Figure 2.12).

Figure 2.12: Penetrating Time-of-Flight 3D scanner schematic

This difference, together with the property of the speed of wave in the material, can be used to calculate the thickness of the material in the scanned direction, as described by Equation 2.5 below:

$$thickness = \frac{v_0 v_1 \Delta t}{v_0 - v_1} \quad (2.5)$$

where v_0 is the velocity of the wave in air, v_1 is the velocity of wave through the scanned material, and Δt is the difference in time taken for the wave to travel from the source to the sensor, with and without the scanned object between. The scanned object, or scanner setup can be rotated in all axes, of which the results can be superimposed to form a reconstructed 3D model of the scanned object. Similarly, the accuracy of such wave is limited by the accuracy of the sensor and timer.

Triangulation – Laser-based

Laser-based triangulation calculates the location of a point based on the location the reflected laser falls in the sensor, as shown in Figure 2.13. The coordinate of the scanned point in space can be calculated by triangulation with Equation 2.6:

$$x = \frac{b \cdot x'}{f \cdot \cot \theta - x'} \quad (2.6)$$

where x is the coordinate of the point on the surface, b is the distance between the source and sensor, x' is the coordinate location of point on the sensor, f_l is the focal length of sensor, and θ is the angle between the sensor and scanned point.

Figure 2.13: Laser-based triangulation 3D scanner schematic

Such a working principle is only applicable to a non-penetrative 3D scanner, and gives a relatively high accuracy among all the scanning principle used, but with limited range.

Triangulation – Photogrammetric

Photogrammetric triangulation similarly calculates the location of a point by triangulation, but instead using the relative location of the same point on different cameras (see Figure 2.14).

Figure 2.14: Photogrammetric triangulation 3D scanner schematic

The distance can be computed by Equation 2.6, as shown below:

$$distance = f_l \left(\frac{b}{u_1 - u_2} \right) \quad (2.6)$$

where f_l is the focal length, b is the distance between cameras, and u is the location of point on screen.

Such a working principle is only applicable to a non-penetrative 3D scanner, and the passive setup is cheap, but leads to a high sensitivity to noise.

Interferometry – Phase shifting

Phase-shift scanners work on the principle where a wave pulse emitted by a source is captured by the receiver, and the phase difference between the source emission and sensor detection is used to calculate the distance of the point from the scanner (see Figure 2.15).

Figure 2.15: Phase-shifting interferometry 3D scanner schematic

The distance of the scanned point from the scanner is given by Equation 2.8, as shown below:

$$distance = \frac{c \cdot T}{4 \cdot \pi \cdot f} \quad (2.8)$$

where c is the speed of light, T is the captured phase shift, and f is wave modulation frequency.

Such a working principle is only applicable to a non-penetrative 3D scanner, and gives one of the highest accuracy amongst all the 3D scanner. However, this

scanning principle is restricted to single point scanning, and hence is slow in digitising an area.

Interferometry – Coherence Scanning

Coherence scanning, also known as conoscopic holography, is based on the fact that lasers consists of two rays, an ordinary and extraordinary ray. By polarising the laser wave and passing it through a uniaxial crystal, the ordinary and extraordinary rays will diffract differently, based on the distance of the reflection. The emerging rays will then be polarised again, and the different rays will interfere, forming an interference pattern called the Gabor zone lens (see Figure 2.16). This interference pattern would contain information of the distance laser travelled since being reflected, and hence the process can be repeated over the surface to digitise the surface.

Figure 2.16: Coherence scanning interferometry 3D scanner schematic

Such a working principle is only applicable to a non-penetrative 3D scanner, and has an extremely high accuracy. However, it is limited to scanning a single point at a time, and has a limited range, and is used for digitising surface roughness instead of objects.

Shading

Non-penetrating

3D scanning using an object shading is done by estimating the surface orientation, based on reflections from source to camera (see Figure 2.17). It is typically inaccurate, although it may be combined with other technologies to augment the results.

Figure 2.17: Non-penetrating shading 3D scanner

Penetrating

A penetrative shading-based 3D scanner is based on the principle that different materials would adsorb different amounts of wave. This would lead to different intensities of wave captured by the 3D scanner sensor opposite from the wave source, based on the thickness of the object (see Figure 2.18). The object or scanner can be rotated to capture intensity images from different angles, and the results can be superimposed to reconstruct the 3D model of the object.

Figure 2.18: Intensity image penetrating shading-based 3D scanner

Texture

3D scanning based on an object's texture is based on the stretching or shrinkage of a regular shape on an object surface, when viewed from different angle (see Figure 2.19). Such scanners are thus limited to scanning objects with regular shapes or textures on the surface, and is only applicable to a non-penetrative 3D scanner.

Figure 2.19: Texture on an object

Silhouette

3D scanning based on silhouette is similar to a penetrating shading-based 3D scanner, except that the wave is non-penetrating. This results in a clear interface along the outline of the object in the captured image (see Figure 2.20). Similarly, the object can be reoriented to capture silhouette from different angles, and the results can be combined to form a 3D model.

Figure 2.20: Combined outline of object

Scan Pattern

A 3D scanner can scan an object a single point at a time (0D), along a line (1D), or over an area (2D). The scanned point (0D) or line (1D) has to be swept over the surface to capture the surface, whereas the 2D scan pattern is able to capture 3D information from a single viewpoint instantaneously. Typically, the higher the order of the scan pattern, the faster the scanning is, but at the expense of accuracy, and a higher cost.

Setup

The setup describes the method of operating the 3D scanner to capture 3D information of the part. This would translate to the relative ease of use of the system.

Proposed Morphological Box Classification Framework

These basic functions were then compiled into the proposed Morphological Box Classification Framework, as shown in Table 2.3.

With this classification framework, any non-contact 3D scanner can be classified. Some commercially available scanners were classified, and the result is shown in Table 2.4.

Table 2.3: Morphological box classification framework based on scanner methodology, for non-contact 3D scanning

Wave Type	Wave Class	Frequency	Wave Source	Radiation Properties	Working Principle	Scan Pattern	Setup
Electromagnetic	Radio Wave	< 300GHz	Active	Non-penetrating (Surface)	Time-of-flight	0D	Fixed System
Electromagnetic	Infrared	300GHz - 430THz			Triangulation - Stereoscopic		
Electromagnetic	Visible Light (Optical/Imagery)	430THz - 750THz			Triangulation - Photogrammetric		
Electromagnetic	Ultraviolet	750THz - 30PHz			Interferometry - Phase Shifting	1D	System Moves
Electromagnetic	X-ray	30PHz - 30Ehz			Interferometry - Coherence Scanning		
Electromagnetic	Gamma Rays	> 10EHZ			Shading	2D	Object Moves
Mechanical	Infrasound	< 20Hz	Passive (Ambient)	Penetrating (Volume)	Texture		
Mechanical	Acoustic	20Hz - 20kHz			Silhouette		
Mechanical	Ultrasound	> 20kHz					Both system and object moves

Table 2.4: Classification of commercially available scanners

No	Brand	Model	Wave Type	Wave Class	Wave Source	Radiation Property	Working Principle	Scan Pattern	Setup
1	L3 Technologies	SafeView	Electro-magnetic	Radio-wave	Active	Penetrating	Shading	2D	Fixed System
2	Siemens Healthineers	MAGNE-TOM Verio	Electro-magnetic	Radio-wave	Active	Penetrating	Shading	1D	Both Moves
3	Microsoft	Kinect	Electro-magnetic	Infrared	Active	Non-penetrating	Triangulation	2D	Fixed System
4	FARO	Focus 3D X 130	Electro-magnetic	Infrared	Active	Non-penetrating	Interferometry - Phase Shifting	0D	Fixed System
5	FARO	Edge Scan Arm HD	Electro-magnetic	Visible Light	Active	Non-penetrating	Triangulation - Laser-based	1D	Hand-held
6	MakerBot	Digitizer	Electro-magnetic	Visible Light	Active	Non-penetrating	Triangulation - Laser-based	1D	Object Moves
7	Artec	EVA	Electro-magnetic	Visible Light	Active	Non-penetrating	Triangulation - Laser-based	2D	Hand-held
8	Zygo	NewView 8000	Electro-magnetic	Visible Light	Active	Non-penetrating	Interferometry - Coherence Scanning	2D	Fixed System
9	Bruker	Skyscan 1173	Electro-magnetic	X-Ray	Active	Penetrating	Shading	2D	Object Moves
10	SAIC	VACIS® P7500	Electro-magnetic	Gamma-ray	Active	Penetrating	Shading	1D	Object Moves
11	XYZ RGB	Live 3D Body Scanning	Electro-magnetic	Visible Light	Passive	Non-penetrating	Triangulation - Photogrammetric	2D	Fixed System
12	GE Healthcare	Voluson E8	Mechanical	Ultra-sound	Active	Penetrating	Time-of-flight	2D	Hand-held
13	EdgeTech	6205	Mechanical	Ultra-sound	Active	Penetrating	Interferometry - Phase Shifting	2D	Fixed System

Some specific classification functions, like the use of infrasound or acoustic, and texture or silhouette, are not found in the list above as they have inherent limitations, preventing the development of a decent commercial 3D scanner. They have, however, been proven to work in theory, and are used by hobbyist.

With such a classification framework, users can classify the potential 3D scanners. The classification of 3D scanners would present to the user a clear picture of the similarities and differences in the basic functions of scanners. This would in turn allow the user to understand and compare how the scanners would respond to the challenges in their scanning project, based on the differences in scanner technology. The difference in the basic functions can also indicate to the user how much time is required for their scanning project, as well as the ease of use of the scanner. For example, a 3D scanner with a 2D, or patch scan pattern will be able to capture surface data more quickly as compared to a 3D scanner with a 1D (line) or 0D (point) scan pattern. A 3D scanner with a 2D scan pattern is also easier to use as there is no need to sweep the 1D line or 0D point over the surface to capture the surface data, unless there is a difference in the setup of the different scanners too. Hence, the user is able to select a more appropriate scanner for his application.

The derived comparison of accuracy, time required, and ease of use can be used alongside other considerations, like price, physical size, and weight, to make a decision on the best scanner. A Multi Criteria Decision Making (MCDM) method, or weighted sum model, as proposed by Khairyanto et al. (2016) could be used as a scoring method to find the most appropriate 3D scanner for the user to acquire, where the value of each alternative is calculated using Equation 2.9 below:

$$V(x_j) = \sum_{i=1}^n w_i v_i(x_{ij}) \quad (2.9)$$

where $V(x_j)$ is the total value of alternative j, w_i is the weight of criterion i, representing its relative importance based on user's needs, $v_i(x_{ij})$ is the value of criterion i for alternative j, and n is the number of criterions.

2.3.3.4 Case study

To evaluate the effectiveness of the proposed classification framework, a case study was conducted.

In this case study, the most suitable 3D scanner from three available 3D scanners is to be chosen to reverse engineer an industrial rack gear. The rack gear was chosen as it is a commonly used part, but also exhibits many of the challenges faced by optical 3D scanners. It is black, with a reflective smooth surface, and there is poor accessibility to the scanner at the regions between its teeth. The specific rack used for this case study is manufactured by Kohara Gear Industry Co. Ltd. (KHK), with model number of SR8-130. A picture of this rack gear is shown in Figure 2.21. It has five gear teeth with the module of 8, and it is made of S45C mild steel. The three available 3D scanners are FARO Edge (3.7m) ScanArm HD, Artec EVA and FARO Focus 3D X130. The picture of these 3D scanners is shown in Figure 2.22.

Figure 2.21: KHK SR8-130 rack gear

Figure 2.22: The available 3D scanner: (a) FARO Edge (3.7m) ScanArm HD, (b) Artec EVA, and (c) FARO Focus 3D X130

These three 3D scanners are evaluated twice, first by using the proposed Morphological Box Classification Framework, and second, by conducting an actual experiment. They will be evaluated in terms of accuracy, time taken and ease of use. The result from the actual experiment is going to be compared to the

classification framework analysis, to evaluate the effectiveness of the proposed framework.

Using the proposed Framework

Step 1: Identify each compared scanners' basic functions from the classification table.

The basic functions of the compared scanners were identified based on the available functions from the classification table, and tabulated in Table 2.5. This can be using the information provided by the manufacturer.

Table 2.5: Classification of compared scanners

No	Brand	Model	Wave Type	Wave Class	Wave Source	Radiation Property	Working Principle	Scan Pattern	Setup
1	FARO	Focus 3D X130	Electro-magnetic	Infra-red	Active	Non-penetrating	Interferometry - Phase Shifting	0D	Fixed System
2	FARO	Edge Scan Arm HD	Electro-magnetic	Visible Light	Active	Non-penetrating	Triangulation - Lased-based	1D	Hand-held
3	Arttec	EVA	Electro-magnetic	Visible Light	Active	Non-penetrating	Triangulation - Lased-based	2D	Hand-held

Step 2: Evaluate the significant differences in basic functions of each scanner.

FARO Focus 3D X 130

FARO Focus 3D X130 has the characteristic of using infrared, interferometry-phase shifting, 0D and fixed system setup. These are the characteristics of a phase-based ranging laser scanner. It is a scanner whereby the phase difference between an emitted and a received laser wave is measured, of which the depth, or distance from the scanner, can be computed. The multiple depth points can be combined, to form a point cloud. The use of phase shifting interferometry would allow the scanner to have a longer range at the expense of accuracy, and hence the scanner is optimised for long range scanning instead of object digitisation. Due to the 0D scan pattern, the time taken for this scanning process is expected to be long.

FARO Edge (3.7m) ScanArm HD

FARO Edge ScanArm HD has the characteristic of using visible light, laser-based triangulation, 1D and hand operated system. These are the characteristics of a laser line scanner, which is known as the first non-contact 3D scanner. A laser line is projected onto the scanned artefact, which is then captured by a camera. The variation in geometry of the laser line as captured by the camera is triangulated, to give the profile of the artefact under the laser line. In order to capture the surface or area profiles, the laser line has to be swept across the artefact, in a direction perpendicular to the laser line. As the position of the scanner head is required to be known for successful arrangement and alignment of the surface profiles, such a scanner has to be used with either an automated Cartesian stage, an articulate arm, or other forms of positioning system by means of sensors for example. Such scanners typically have a higher accuracy, as the laser line is usually precise, and of high resolution. However, the drawback of such system is that they are slower as compared to 2D patch scanners, as the laser line has to be swept across the surface.

Artec EVA

Artec EVA has the characteristic of using visible light, laser-based triangulation, 2D and hand operated system. The only different from FARO Edge ScanArm HD

is the 2D pattern. This is the characteristic of a structured light scanner. The structured light scanner is similar to the laser line scanner, except that a 2D pattern is projected instead of a 1D line. The 3D profile of the artefact would be acquired by the same method of triangulation. Such scanners are typically faster than laser line scanners, at the expense of accuracy. 2D structured light scanners are also known to have a lower resolution as compared to laser line scanners. As the profiles of each capture would have overlapped the next captured profile, the 3D geometries can be stitched to each other using the overlapping region, and there is no necessity to know the spatial location and orientation of the scanner. As such, such scanners can be used without physical restrictions as a true hand-held scanner, and should be the easiest to use.

Step 3: Gather the technical specification of the compared 3D scanners.

Technical specifications of the compared scanners, as given by the manufacturers based on their own experimentations, are shown in Table 2.6.

Table 2.6: Technical specifications of compared scanners

No	Brand	Model	Volumetric Point Accuracy	Resolution / Point Spacing	Scan Limit
1	FARO	Focus 3D X 130	± 2 mm	0.009° (40,960 3D pixels on 360°)	Working distance - 0.6 - 130 m
2	FARO	Edge (3.7m) Scan Arm HD	$\pm 116 \mu\text{m}$ (91+25)	40 μm	Volume - 3.7m diameter
3	Artec	EVA	$\pm 100 \mu\text{m}$, or 0.03% over 100 cm	0.5 mm	Working Distance - 0.4 - 1 m

Step 4: Decide on which scanner is most suitable for reverse engineering application.

From the Morphological Box Classification Framework, the FARO Focus 3D X 130 FARO is likely to perform the worst in all aspects of comparison. As it is a fixed system setup, the object has to be manually rotated, or the whole scanner has to be moved between scans, to fully digitise the rack gear. This is made more difficult given that the areas between the teeth of the rack are less accessible by the scanner. Due to this, it would perform poorly in terms of time taken and ease

of use. The accuracy would also suffer if the scanner was not appropriately placed, or if insufficient scans were taken. Unfortunately, it would be impractical to scan multiple times for the required accuracy, and hand-held scanner would be significantly more efficient.

Between the FARO Edge ScanArm HD and the Artec EVA, the only difference identified by the Morphological Box Classification Framework is the scan pattern, where the former is 1D, while the latter is 2D. 1D laser line scanners are known to have a higher accuracy than 2D structured light scanners, as it is higher powered, and can reduce the challenges faced by the black, reflective rack, with difficult to access surfaces. As such, the FARO Edge ScanArm HD is likely to perform better in terms of accuracy, as compared to the Artec EVA. This is contrary to the accuracy technical specification as given by the manufacturer, and will be checked against the subsequent experiments.

On the other hand, for the purpose of reverse engineering where time taken is more critical than the accuracy, the Artec EVA will perform better in terms of speed. This is because the 1D scan pattern of the FARO Edge ScanArm HD requires the user to manually sweep the laser line over the surfaces, while the Artec EVA is able to directly capture spatial information of a 2D patch for each frame. This also means that the Artec EVA will be easier to use, as the user need not learn how to successfully sweep the 1D line over all the surfaces, but can easily and instinctively “cover” the object with 2D patches instead.

Actual experiment

Methodology

To ensure that the experiment is reliable, an as-built model is required for the rack being scanned. The original Computer Aided Design (CAD) model provided by KHK cannot be used, as it only includes 3 effective teeth, whereas there are in total of 5 teeth physically presented on the rack. In addition, the CAD model may not accurately represent the as-built part. Hence, for this experiment, a Coordinate-Measuring Machine (CMM) is used to produce the as-built model. CMM machine is commonly used as an industrial standard for inspection due to its high accuracy. The CMM used in this paper is a 3-axis CMM, the Mitutoyo

Beyond-Crysta C544 with PH10M Indexing Head and TP200 Probe, and is used together with the accompanying software MCOSMOS v2.4 with SCANPAK. A picture of this CMM machine is shown in Figure 2.23.

Figure 2.23: Mitutoyo Beyond-Crysta C544

Unlike the other 3D scanner companies who run their own test setups to validate their scanner's accuracy, the Mitutoyo CMM machine's accuracy in its specifications is given by the international standards, ISO 10360-2, and VDI/VDE2617. Equation 2.10 specifies the volumetric length measuring accuracy of the CMM used (Mitutoyo Beyond-Crysta C544 with PH10M Indexing Head and TP200 Probe) in μm , according to ISO 10360-2.

$$MPE_E = 1.9 + 3L/1000 \quad (2.10)$$

where L is a measured length in mm. For this scanned rack, the maximum inaccuracy MPE_E would be $2.4 \mu\text{m}$. This accuracy is significantly better than the other 3D scanners in this comparison, and hence is a good choice to be used to create the as-built model for comparing the remaining scanners.

All the scanners collect point cloud as raw data. To ensure that there is an impartial comparison between the scanners, the same software and workflow were used to process the point clouds. The common point cloud processing software, Geomagic Design X 2016.1.1, was selected for this purpose. The point clouds were cleaned up, registered, and meshed using the "Mesh Buildup Wizard" function in Design X.

Subsequently, the data was transferred to Geomagic Control X 2017.0.3 for comparison. The data was aligned to the as-built model, created from the Mitutoyo CMM, using “Initial Alignment” and “Best Fit Alignment”, and subsequently compared using the “3D Compare” function of shape, using the shortest projection distance as the computed error, within Control X. The standard deviation and the maximum absolute error values were directly extracted from Control X, where the standard deviation of error is given by Equation 2.11, and the maximum absolute error is the largest value among all the shortest projection distance obtained.

$$s = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N-1}} \quad (2.11)$$

where s is the standard deviation, n is the total number of points, x_i is the error of i th point, and $\bar{x}$ is the mean error of all points.

The following factors were compared:

- Accuracy
 - Standard Deviation
 - Maximum Absolute Error
- Time taken
 - Setup and Teardown Duration
 - Scan Duration
- Ease of use

When comparing the accuracy of the 3D scanners, the standard deviation of the error is preferentially used instead of the maximum error value. Additionally, the accuracy specifications provided by manufacturers also refer to the achievable standard deviation of error, when scanning in ideal conditions. It is more common to use of standard deviation of error to compare scanners because the maximum error value could be attributed to outliers caused by noise. Nonetheless, the maximum absolute error results are also presented, as a small maximum absolute error does imply a good scanner.

As the experimentation should be representative of the measurement task, the time taken for the different scanners should be the time taken by a seasoned user,

instead of a new user spending time learning how to operate the scanner. Hence, the time taken was measured after authors have scanned the object multiple times to be familiar and comfortable with the scanning procedure.

The complete scanning process also requires for the spatial information of the object to be captured from different viewpoints. Hence, the ease of use for the scanner was concluded empirically based on the level of difficulty to move or manipulate the scanner, to capture spatial information from a different point of view.

The geometry of the rack is such that the areas between the teeth are inaccessible for the optical scanners to be normal to the surface. This limitation of having to capture the surface details at an angle, coupled with the dark and reflective surface of the rack, would potentially lead to increased inaccuracies within the teeth area. As such, the comparison of the scanned data was done twice, one where the 5 simple faces are compared, as shown by the hatched areas in Figure 2.24, and another where the whole rack inclusive of the teeth is compared. The simple face comparison was done by selecting the simple faces for the alignment and comparison process within Geomagic Control X.

Figure 2.24: Simple faces of rack

Experiment result

The numerical results are as shown in Table 2.7.

Table 2.7: Experimental results of scanners

3D Scanner	Mitutoyo Beyond-Crysta C544 with PH10M Indexing Head and TP200 Probe	FARO Edge (3.7m) Scan Arm HD	Artec EVA	FARO Focus 3D X 130
Standard Deviation (Simple Faces)	0.0001*	0.0328	0.0448	0.2684
Maximum absolute deviation (Simple Faces)	0.0002*	0.1967	0.4876	0.9574
Standard Deviation	0.0002*	0.0938	0.1496	2.0309
Maximum absolute deviation	0.0004*	0.5456	1.4588	7.1382
Ease of Use (Ranking)	2	3	1	4
Setup and Teardown Duration (mins)	2	19	4	7
Scan Time Taken (mins)	374	38	9	71

* These values reflect the errors in the CAD model reconstruction from the raw PCD.

Accuracy

Meshes created from the acquired point clouds of the different scanners are shown in Figure 2.25, whereas, Figure 2.26 shows the deviation of these meshes as compared to the as-built reference model created with the Mitutoyo CMM. The red coloured area shows that the points captured by the 3D scanner is above the surface of the reference model surface, while the blue colour shows that the point is beneath the reference model surface, and the intensity of the colour shows the amount of deviation observed. The accuracy data acquired from Control X were also collated in Figure 2.27.

Figure 2.25: Reconstructed mesh from the point clouds for: (a) FARO Edge ScanArm HD, (b) Artec EVA, and (c) FARO Focus 3D X 130

Figure 2.26: Deviation map of mesh as compared to as-built model for (a) FARO Edge ScanArm HD, (b) Artec EVA, and (c) FARO Focus 3D X 130

Figure 2.27: Obtained accuracy of the compared 3D scanners

Time taken

Time taken for each scanner was collated in Figure 2.28, including the CMM scanner, which is the industrial standard for accurate inspection used as our reference model. The excessively long time taken by the CMM demonstrates the advantages of the use of optical scanners.

Figure 2.28: Time taken of each scanner

Ease of use

The ease of use of the scanners were compared based on the level of difficulty to move or manipulate the scanner, to capture spatial information from a different point of view.

The FARO Focus 3D is simple to operate by just setting the required parameters and letting it run. However, it is only able to capture the point cloud from a single point of view. Hence, for the purpose of reverse engineering, where the all-around point cloud for a part is to be acquired, it is tedious as the scanner has to be shifted, or the part has to be re-oriented multiple times for the object to be fully digitised. Also, unlike the other two scanners, there is no real-time update of the captured data, and the user has to estimate the best location to capture the most data accurately. Thus, the FARO Focus 3D is the most difficult to use.

Between the remaining two scanners, the FARO Edge ScanArm HD is the more difficult system to use due to the limitations of a physical articulate arm, which requires the user to plan ahead for the placement of the scanned object and the scanner. Although the movement of the articulate arm is designed to move freely, with 7-axis of freedom, it takes some time to get used to. In addition, the FARO

Edge ScanArm HD only captures the profile of a line, and the laser line has to be manually swept across the object's surface. Thus, it is required of the user to be experienced to know how to sweep the laser line across the object to capture all the faces.

The Artec EVA is the easiest to use, as the data is captured as a 2D surface patch each frame, instead of the need to sweep a line or a point. The option of using the scanner with a battery pack and attached tablet also means that the scanner has no physical restrictions, and is the easiest to use.

Discussion

Based on the proposed Morphological Box Classification Framework, the accuracy of the FARO Edge ScanArm HD will be the best among the three compared scanners, despite the manufacturer's specification of the Artec EVA indicating a better accuracy performance. This is similar to the results obtained via the experimentation. However, this discrepancy is not due to the manufacturers' specifications being inaccurate. Instead, as there is a lack of an international standard for comparing the various optical 3D scanners, manufacturers carry out their own tests under certain conditions to acquire the specifications of their scanners, and the tests may not be representative of the actual task of scanning by the end consumer. From the experimental results, the Artec EVA did perform within specifications, and comparably in terms of accuracy to the FARO Edge ScanArm HD for the digitisation of the simple faces. The difference in accuracy is only significant when the digitised rack teeth are being compared. As such, the proposed framework can be useful to better identify potential strengths and weaknesses in the scanners for specific applications, to aid in the choice of scanner.

The proposed Morphological Box Classification Framework is also useful in giving the user a reliable prediction of the relative time required and ease of use for the scanning process, by comparing the basic sub-functions of the scanners. This was verified by the experiments.

For subsequent experimentation in this thesis, a high accuracy scanner for digitising potentially challenging damage surfaces is required. Hence, a line laser-based scanner, working on the principle of triangulation, is recommended,

based on our classification framework, as well as the results obtained from our case study. The chosen 3D scanner to be used is the FARO Edge (3.7m) ScanArm HD.

2.3.4 3D Point Cloud Pre-Processing

With a selected 3D scanner, a damaged part can be digitised, of which the data will be in the form of point cloud. This PCD is a series of points in space representing the surface of the scanned object, with the point locations defined in a coordinate system. To allow for the extraction of useful information from the PCD, it must be pre-processed from its raw form. This includes the steps of noise removal and registration. While the order of these processes is not fixed, the more adopted approach is noise removal before the registration step, as the presence of noise might lead to errors within the registration step.

Noise Removal

The first PCD pre-processing step is noise removal. PCD, regardless of the acquisition technique, is bound to contain noise, or erroneous points. Many researchers have been working on this problem, with different approaches. Linsen [106] removes noise by the use of a low pass filter. Mederos, Velho [107], Weyrich, Pauly [108], and Dey, Goswami [109] proposes the use of Moving Least Square (MLS) fitting for noise removal. Lange and Polthier [110] proposes the use of Partial Differential Equation (PDE) for getting anisotropic geometric mean curvature flow for anisotropic fairing of a point sampled surface, and noise removal. Xie, McDonnell [111] proposes the removal of outliers by distance. Pauly, Mitra [112], Ivriissimtzis, Lee [113], Giesen, Spalinger [114], Fenn and Steidl [115], and Steinke, Schölkopf [116] proposes the use of statistical learning techniques for noise removal. Schall, Belyaev [117] uses a mean-shift based clustering procedure to associate the likelihood of a point being noise. Rusu, Marton [118] removes outlier based on mean and standard deviation of nearest neighbour distance. Sun, Schaefer [119] removes noise by L_0 minimisation, where L_0 is a metric directly related to sparsity.

As noise removal algorithms are well established, it is a functionality commonly available within reverse engineering software. Hence, for this research, the

provided functionality with Geomagic Design X is used instead, to remove noise and outliers from the data. Geomagic Design X was chosen over other available software, as the manufacturer of the scanner used for this research have tested the software with their hardware, and recommended it. An example of the results of the Geomagic Design X outlier removal is shown in Figure 2.29.

Figure 2.29: Outlier removal – (a) Before removal; (b) After removal

Registration

After the PCD is cleaned, it can be registered. 3D point clouds usually contain multiple scans of an object or scene, due to the scanner's limited field of view. The different PCD captured from different viewpoints may be differently oriented in the virtual 3D space. As such, registration is a required pre-processing step [120].

The most established registration algorithm is the Iterative Closest Point (ICP) algorithm by Besl and McKay [120], and Chen and Medioni [121]. The algorithm starts from a source and reference point cloud being roughly registered, and each point from the source point cloud is matched to the closest point in the reference point cloud. The root mean square point to point distance is minimised by transforming it with a combination of rotation and translation, after which the whole process is repeated [120]. There has been many variants of ICP proposed by different researchers to solve various problem, although the results largely depends on the application [122-126]. Apart from ICP and its variants, other registration algorithms have been proposed for both coarse and fine registration [127].

Similarly, registration algorithms are well established and commonly available within reverse engineering software, and the included functionality within Geomagic Design X is used for this research, for the same reasons as the use the outlier removal algorithm. An example of the results of the Geomagic Design X point cloud registration is shown in Figure 2.30.

Figure 2.30: Point cloud registration – (a) Before registration; (b) After registration

2.3.5 Surface Damage Detection

Automated surface damage detection is traditionally done by image processing. This has been made more reliable, with the advancement of deep learning for detection [128]. This is due to the availability of large dataset of images available. However, due to the limited amount of available PCD of damaged parts, deep learning cannot be readily extended to damage detection from PCD. Computational efficiency is also not optimised for geometrical deep learning, as traditional deep-learning architectures are based on the assumption of regularly structured 1D or 2D data, hence benefitting from modern GPU hardware [129]. Hence, other approaches apart from deep learning are common for automated damage detection from PCD, and are reviewed in this section.

To ride on the reliability of image damage detection algorithms, researchers like Rabah, Elhattab [130] and Adhikari, Moselhi [131] have used images for damage detection, after which the image is superimposed onto the PCD. While it is possible to superimpose the intensity or RGB values for the points, the added step could introduce errors to the results, as an inaccurate registration of image to the PCD would lead to wrongly detected damages. Additionally, the use of PCD in damage detection instead of images is still preferred, as the data could potentially

provide more information of the severity of damage, like depth [72]. Studies have also shown that damages difficult to locate in 2D images is found and clearly identified by the use of 3D point cloud [132]. Nonetheless, image-based damage detection could be employed alongside the PCD based damage detection proposed in this research, but will not be explored in this thesis.

Whilst some work has been done in this area of damage detection from 3D PCD, all of it have some assumptions and limitations that prevents widespread implementation of it. For example, Zheng, Li [78] makes use of the comparison of scanned data to the nominal CAD model of turbine blades (see Figure 2.31). However, the CAD model may not be readily available [25, 26], and even if it were available, the comparison may not be feasible due to the acceptable distortion of the worn part [26].

Figure 2.31: Comparison of scanned data to nominal CAD model, adapted from Zheng, Li [78]

Olsen, Kuester [83] proposes the use of intensity value of the 3D scan to identify cracks in concrete (see Figure 2.32), based on the assumption that the reflectance of a damaged region is scattered and degraded, as compared to that of a smooth undamaged surface. However, this intensity value may not be available for all 3D scanners, specifically the higher accuracy line laser scanners ideal for inspection. The intensity value is also affected by environmental lighting conditions, which will introduce a significant amount of noise to the results.

Figure 2.32: Intensity map highlighting different intensity signatures at crack, adapted from Olsen, Kuester [83]

Teza, Galgaro [133] proposes the comparison of standard distribution of Gaussian curvature in each subarea, as any damage would result in a larger standard deviation obtained (see Figure 2.33). However, the presence of excessive noise will cause this method to not work, according to the researcher. Li, Yao [77] proposes a similar method, by comparing the curvature using Interpolate Points to construct a global 4D Shepard surface, while Li, Li [25] identifies defects by assuming that the intact point cloud is not smoothly continuous with the point cloud in non-defective regions, and the Gaussian curvatures of the point cloud on the boundary of defective region would gain a sudden increase. A quadratic surface fitting is used for the curvature estimation, and a threshold is used to identify the boundary of the damage region (see Figure 2.34). Similarly, these methods are sensitive to excessive noise. In addition, these algorithms have a limitation of identifying simple damages to a surface with a rough surface. Other forms of damages, like deformation, may not be identified by these algorithms.

Figure 2.33: Mean curvature distributions (c) & (d) of Undamaged (a) and Damaged (b) Sub-area respectively, adapted from Teza, Galgaro [133]

Figure 2.34: Defective region border, adapted from Li, Li [25]

Liu, Chen [134] identifies damage by the displacement of points from a manually identified plane (see Figure 2.35). Khoshelham, Elberink [135] utilises Hough transformation for plane identification, followed by feature extraction,

classification, and feature selection to identify damage areas. Chmielewski, Laszewicz-Śmiałowska [86] proposes a similar method as Khoshelham, and applies it to defect detection in furniture elements. Peng, Yue [87] assumes that damages in wood are only depressions, and classifies points whose height falls below a threshold as damage. However, as these methods work based on the assumption the damage falls on a planar surface, they will not work for damages on other surfaces. Edris, Jawad [79] converts the point cloud height information to intensity value of a 2D image, and uses image processing tools to identify damaged regions. Whilst this method utilises the advancement in research for image damage identification, it is limited to damages viewable on a 2D plane.

Figure 2.35: Displacement map of points from plane, adapted from Liu, Chen [134]

Torok, Golparvar-Fard [72] compares the reconstructed mesh normal to either other surface normal, axial direction, or from a reference point, to detect damage mesh with a normal that differs by a significant amount. Guldur and Hajjar [136] proposes the detection of damage in structures by assuming that damaged regions in the PCD would have surface normal inconsistent from the surrounding surface patch, object skeleton, or reference point (see Figure 2.36). As these methods are based on the assumption that the part has some form of structure, they will not work on parts without structures, or localised scan data.

Figure 2.36: Representation of reference normal and normal from defect areas, adapted from Guldur and Hajjar [136]

Tjahjowidodo and Han [137] proposes estimating the nominal profile data on selected splines by fitting a smooth cubic spline function (see Figure 2.37), and subsequently identify the damage by a Boolean operation. This is based on the assumption of a smooth nominal profile, where the defect portion is smaller than the nominal profile, and the profile should not have any undercut.

Figure 2.37: Estimated Nominal Profile Sequence, adapted from Tjahjowidodo and Han [137]

Ghahremani, Khaloo [138] proposes analysing the point density of the PCD, of which damage areas are assumed to have a higher point density than the rest of the model (see Figure 2.38). However, this method limits the detection of

damaged regions to depressions, where the surface area of the part increases due to a damage, and would also wrongly detect damages from part features.

Figure 2.38: Local point density for defect detection, adapted from Ghahremani, Khaloo [138]

While there is a considerable amount of automated damage detection algorithms developed, they all have specific scopes and assumptions, either on the type of damage, or on the geometry of the undamaged part, preventing widespread application of each algorithm. The limitations are summarised in Table 2.8. This is a major problem for our application of detecting damage in the marine and offshore industry, as the undamaged geometry of the part is not known, and a variety of types of damage can be present. This issue prevents the automation of the damage detection sub-step of the inspection step, and hence the eventual automation of in-situ remanufacturing. This gap in knowledge is thus addressed and studied in Chapters 4 and 5, by proposing the use of machine learning on extracted features, to accurately and robustly detect damage automatically from PCD.

Table 2.8: Summary of limitations of prior literature

Limitations	Methods
Scope of Damage Types	Teza, Galgaro [133]; Li, Yao [77]; Li, Li [25]; Ghahremani, Khaloo [138]
Assumptions on Part Geometry	Liu, Chen [134]; Khoshelham, Elberink [135]; Torok, Golparvar-Fard [72]; Guldur and Hajjar [136]; Tjahjowidodo and Han [137]; Chmielewski, Laszewicz-Śmietańska [86]; Peng, Yue [87]
Additional Input	Zheng, Li [78]; Olsen, Kuester [83]; Rabah, Elhattab [130]; Adhikari, Moselhi [131]; Edris, Jawad [79]

2.3.6 Surface Damage Classification

The use of 3D data for damage classification has been explored by others, although it is for varying fields.

Rehor [139] classifies the damage into the damage type from 3D range data (see Figure 2.39), but the scope of her work is on buildings and the types of damages as defined by Schweier and Markus [140].

Figure 2.39: Building damage types, adapted from Schweier and Markus [140]

Djamaluddin, Mitani [141] also studied the damage on buildings from coal mining, but his classification from 3D range data is based on the extent of the damage, and is classified as either very slight, slight, sever, and very severe.

Lee and Park [142] studied damages on pipelines, and subsequently attempt to classify the type of damages, but the scope of damage is only limited to having a notch in the pipe, or with loose bolts in the pipe.

Huang, Koh [143] proposed a PCA-based damage classification using improved layerwise theory, while Katunin [144] used computed tomography 3D data of a potentially damaged component, and wavelet analysis to detect and classify damage to composite structures after water-jet cutting. The application and scope of damage types of their research is however limited to delaminated composite structures, or air pockets.

There is considerably less amounts of automated damage classification algorithms developed, with majority of them tailored for the building and construction industry. The other literature on damage classification has a very specific scope of damage types as defined by the researchers, and is insufficient for our application on understanding the types of damage which will help in the downstream decision of remanufacturing. As such, this gap in knowledge is also addressed and studied in Chapter 5, by proposing the use of machine learning on extracted features, to accurately and robustly classify damage automatically from PCD.

2.3.7 3D Point Cloud Machine Learning

With the success of image-based machine learning, researchers have been trying to apply machine learning on 3D data as well.

The most popular application of machine learning on point cloud is Pointnet and its variants, as it is able carry out object classification and semantic segmentation (see Figure 2.40), and the neural network is robust with respect to input perturbation and corruption [145]. However, the success of their approach is due to the regularity of the geometries for classification and segmentation. On the other hand, the direct application of machine learning onto PCD for the purpose of damage detection, or the training of a machine learning model with raw 3D data without any dimensional reduction or feature extraction, is limited by the irregularity of damage regions, having a large variance in shape, texture, and size. Additionally, the state-of-the-art research focuses on the application of machine

learning on PCD of common items, where there is PCD is readily available for training, unlike damaged parts with limited PCD available.

Figure 2.40: Semantic segmentation via deep learning, adapted from Qi, Su [145]

To overcome such limitations of direct geometric machine learning, but still exploit the virtues of machine learning, researchers like Bendich, Gasparovic [146], Elbaz, Avraham [147], Tootooni, Dsouza [148] and Czerniawski, Sankaran [149] have made use of dimensionality reduction via feature extraction for their required purposes, and this research seeks to adopt a similar approach for dimensionality reduction, with a set of extracted features defined for our purpose.

2.4 Evaluation of the State-Of-The-Art

The literature review was first conducted on the trends of remanufacturing, of which a key enabling technology for the future of automated remanufacturing is additive manufacturing technologies, specifically the Directed Energy Deposition technology. This has unlocked the potential of repairing single large metal components, like that found in the marine and offshore industry. However, to disassemble and transport such large parts to an off-site remanufacturing factory is time-consuming, expensive, labour intensive, and potentially dangerous, or impractical to remove in some cases due to physical constraints, requiring excessive structural changes. This has led to an interest in in-situ remanufacturing. However, the review of literature revealed that there is no

automated in-situ remanufacturing framework available. The formalism of such a framework is warranted, as it would not only improve consistency, improve quality, improve efficiency and reduce cost, but also increase accountability, tracking and traceability. As such, this research addresses this gap in the state-of-the-art by proposing an automated in-situ remanufacturing process flow.

Next, a literature review was conducted on the individual processes necessary for automated in-situ remanufacturing. The first step for automated in-situ remanufacturing is the cleaning step. The current state of the art of cleaning for remanufacturing is reviewed, and is found to be comprehensive in cleaning a wide variety of contaminants. As such, the cleaning technique used should be selected on a case by case basis, based on the contaminants present on the damaged part. Additionally, there technology for automated cleaning is found to be available in current literature, and can be implemented in-situ.

The next step of automated in-situ remanufacturing is the inspection process, which begins with the data acquisition. It is found that there are numerous 3D scanning technologies available, including automated solutions that are portable, and can be used for in-situ digitisation. However, for the purpose of remanufacturing, an appropriate 3D scanner has to be selected for the digitisation of the damaged part, and a study was conducted to select an appropriate scanner for this research, which is the FARO Edge (3.7m) ScanArm HD. The pre-processing of data, in the form of noise removal and registration, was also found to be well-established and commercially available, and hence, commercial solutions for the data pre-processing will be used in this research.

However, in the subsequent step in the inspection process, which is the data analysis, a limitation in the state-of-the-art technology was found, more specifically the damage detection process. While image-based damage detection is established and highly accurate, the detected results cannot be used readily by downstream repair processes, and the damage detection from 3D data is preferred. It is observed that although the algorithms proposed by other researchers were able to successfully detect the damaged region in their experiments, they all have some form of limitations in terms of scope of damage types detected, assumptions on part geometry, or require additional inputs. These pre-assumptions were valid

as the researchers have scoped their work for a specific application. However, the algorithms proposed may not be reliable in detecting a different form of damage, or on a different part. These limitations of the current state of the art of PCD damage detection algorithms is also addressed in this thesis.

As with image-based damage detection and classification, the use of machine learning can be promising to increase the accuracy and robustness. The last area reviewed was thus on the use of machine learning on geometric data, more specifically point cloud data. The issues with machine learning on 3D data are found to be due to unoptimized computational efficiency for geometric learning, as well as insufficient training data. However, due to the success of the use of machine learning in image-based damage detection, this research seeks to use machine learning indirectly on the 3D data, for damage detection and classification.

In summary, the technological gaps from the critical review of the current state-of-the-art are identified, which also forms the focus of this thesis, and are as follows:

- Absence of a process flow for automated in-situ remanufacturing
- Limitations in PCD based damage detection algorithms
- Limitations in the direct application of machine learning on geometric data

2.5 Summary

This chapter has reviewed the current literature on the types of remanufacturing, remanufacturing process flow, and the cleaning and inspection remanufacturing processes. It is found that there are limitations in the state-of-the-art in terms of the absence of a process flow for automated in-situ remanufacturing, limitations in PCD based damage detection algorithms, and limitations in the direct application of machine learning on geometric data. As such, these identified gaps within the knowledge are addressed in the subsequent chapters.

CHAPTER 3

Automated In-Situ Remanufacturing Process Flow

The traditional way of looking at remanufacturing does not encompass the full potential of automated in-situ remanufacturing, and there is no consistent consensus of what that might be. As such, a new process flow is required to formalise this aspect of work, as it would not only improve consistency, improve quality, improve efficiency and reduce cost, but also increase accountability, tracking and traceability. Such a process flow for automated in-situ remanufacturing is not available in current literature, and hence is proposed in this chapter. This is done by studying the steps and sub-steps of the current practices in remanufacturing, and filtering out the redundant steps in automated in-situ remanufacturing. Details on the modifications required for each step and sub-step to attain automation, and be carried out in-situ, is also presented in this chapter. With the proposed process flow, there can be more accountability, tracking, and traceability on the remanufacturing process, on top of improved efficiency and reduced cost.

3.1 Existing Process Flow

The traditional remanufacturing process flow as proposed by Steinhilper [40] was based on the 5 basic steps of disassembly, cleaning, inspection, repair process, and reassembly, (see Figure 2.1). However, due to the different requirements by different industries, there may be additional or fewer steps in the process, and the steps may be reordered and repeated. As such, a generic remanufacturing process was proposed by Sundin [8] (see Figure 2.2). From the review of current literature, there are no specific process flow for automated in-situ remanufacturing, and will be proposed in this chapter, based on that proposed by Sundin [8].

3.2 Process Flow Requirements

As discussed in Chapter 1, remanufacturing can greatly benefit from automation, in terms of the economical and societal benefits. Additionally, although it is impractical to disassemble and transport the large parts from the marine and

offshore industry to an offsite remanufacturing facility, the remanufacturing can be carried out on site, and such in-situ remanufacturing would be able to benefit from the same economical and societal benefits. As such, the proposed process flow should be customized for an automated in-situ remanufacturing.

Automation

To attain a high level of automation in remanufacturing, each remanufacturing process and sub process should be automated, or have as minimal human-computer interaction as possible. The proposed process flow should thus encompass all the necessary process and sub-process details, and how automation can be carried out in each individual process and sub-process. Additionally, the process flow should include the links between each process and sub-process, of how the parts and data flow between them.

In-Situ

As discussed in Chapter 1, a class of potential remanufactured parts are the large parts in the marine and offshore industry, for example the rack and pinion as shown in Figure 3.1. The disassembly and transportation of such parts are impractical and expensive, and as such, in-situ remanufacturing is ideal. The proposed process flow should thus be adapted to in-situ processes and sub-processes within the in-situ remanufacturing framework.

Figure 3.1: Oil rig rack and pinion to be remanufactured

3.3 Proposed Process Flow

The remanufacturing framework is adapted from the generic remanufacturing process proposed by Sundin [8] (see Figure 2.2), where selected generic processes are ordered and adapted for the purpose of automated in-situ remanufacturing. The proposed framework schematic is shown in Figure 3.2. The main modification to the generic remanufacturing process is the removal of the disassembly, transportation, storage, and assembly steps.

Figure 3.2: Proposed automated in-situ remanufacturing process flow

In an automated in-situ remanufacturing process as presented in Figure 3.3, the damaged part is first cleaned of any foreign material. The damaged part is then inspected, to detect any damages and analyse it, as well as to reconstruct the nominal model. A decision will be made on the reparability of the part, following which if the part is to be repaired, the repair process is planned before the actual

repair process. The repair job has to be evaluated, and the repaired part certified before it is returned to its service. The details of each steps and sub-steps will also be elaborated in the following sections.

3.3.1 Disassembly and Reassembly

The first step in traditional remanufacturing is the disassembly step. This is because, in traditional remanufacturing, the damaged parts are small and economical enough to be transported to an off-site remanufacturing factory, and there is a sufficient number of damaged parts to benefit from consolidating the remanufacturing process in a single location. Hence, it is necessary to disassemble the damaged part for transportation. However, as the damaged part within the scope of this project is typical large and exclusive, like that from the marine and offshore industry, an in-situ remanufacturing model is preferred if the damage is accessible. Hence, there is no need to disassemble for transportation, and hence reassemble. Some examples where remanufacturing has been conducted in-situ are the repair of hulls[150] and propellers[151].

In traditional remanufacturing, the repair process is also carried out manually using conventional welding and traditional milling. The machines used for this repair process are bulky and immobile, and hence there is a need for disassembly for the damaged region to be made accessible. However, with the advancement of mobile manufacturing machines, specifically robotic arm manufacturing machines like the MX3D Robotic Arm Welding Machine (see Figure 3.3) for additive manufacturing, and the Capriol Autonomous Computer Numerical Control (CNC) Mobile Robot (see Figure 3.4) for subtractive manufacturing, the damaged region can be worked on directly without the need for disassembly. This is especially so for the case of the marine and offshore industry, where most parts are larger and more spaced out than parts from other industries. Hence, the step of disassembly and reassembly can be omitted if the damaged region is accessible by the mobile hybrid machines. Nonetheless, should the damage be inaccessible, a partial disassembly could be carried out instead. In addition, only visible surface damage can be remanufactured in-situ, as any internal damage would require full accessibility of the damaged component for inspection and testing, for example using NDT (Non-Destructive Testing).

Figure 3.3: MX3D robotic arm welding machine (a) Concept; (b) Prototype, adapted from MX3D [152]

Figure 3.4: Capriol autonomous CNC mobile robot, adapted from Capriol [153]

3.3.2 Transportation and Storage

For the same reasons as above for remanufacturing on-site, in-situ remanufacturing would also remove the necessity for transportation, storage, and related processes. The advancements of mobile hybrid manufacturing machines allow for the required machines to be transported to the damaged part, instead of vice versa. This would also eliminate the need for storage, as the repaired part is operational immediately after repair and certification. There is, however, the need

to transport the necessary equipment for remanufacturing to the damaged parts, if the equipment is not stored on-site.

3.3.3 Cleaning

A step within the remanufacturing process flow that cannot be omitted is the cleaning step. This is especially important for marine and offshore applications, as the harsh conditions would lead to corrosion, oil and debris build up. The presence of such externalities would be digitised in the inspection step, and the captured build-up would be mixed with the actual data [42], leading to subsequent errors. A clean part would also lead to improved mechanical and corrosion performance after the additive manufacturing repair [43], and hence has to be carried out. Based on the literature review conducted, the current state of the art of cleaning for remanufacturing is found to be comprehensive in cleaning a wide variety of contaminants.

The cleaning process is traditionally carried out manually. However, companies have been working towards the automation of the cleaning process [154], for example, the use of industrial high-pressure waterjet on a robotic arm for cleaning [155], or an oscillatory moving tool mounted on a robot arm [156]. Based on these literatures, before the automated cleaning process itself, it is necessary for the system to acquire data of the damaged part geometry and unclean coverage, and can be in the form of a 3D scan, or photographs. This will allow for the automated cleaning system to plan the appropriate cleaning procedure and path [157]. Due to the compact nature of the robot arm mounted cleaning solutions, they can be implemented in-situ as well, enabling the cleaning step to be automated and in-situ.

3.3.4 Inspection

After the cleaning process, the next step necessary for remanufacturing is the inspection step. In traditional remanufacturing, inspection refers to the basic damage detection, usually done manually and visually. This would aid in the decision of whether the damage is repairable, and how the damage is to be repaired. Whilst this is a seemingly straightforward step, it is traditionally highly dependent on skilled labour. Hence, the automation of this step can be beneficial

to the overall remanufacturing process, including the subsequent steps like the repair process.

Data Acquisition

Similar to automating the cleaning process, it is necessary to have digital information of the damaged part for the computer system to automate the repair process. As such, this inspection step should include the sub-step of the data acquisition of the damaged part. The earlier adopters of automation in the remanufacturing process have made use of images for the inspection process, for the detection and classification of the damage, as well as for the repair decision. However, as a 3D model of the damage region is required for the automated repair step downstream, it is resource efficient to utilise 3D scanning for the data acquisition sub-step for inspection, instead of other methods like imaging. As such, the use of a 3D scanner for scanning the 3D geometry is preferred in this process.

One important consideration within this step, is the selection of the 3D scanner to be used to digitise the damaged part, as the choice of 3D scanner will greatly affect the subsequent results. A study of the choice of 3D scanner was explored in detail in Section 2.3.3, of which the line laser-based scanner, working on the principle of triangulation, is recommended for the purpose of remanufacturing.

To automate the scanning process, such a scanner can be mounted onto a robot arm to automatically navigate around line of sight occlusion issues. Such setups are commercially available, like the Artec RoboticScan (see Figure 3.5(a)), Creaform MetraSCAN 3D-R (see Figure 3.5(b)), and the FARO® Cobalt Array Imager (see Figure 3.5(c)). Such commercial solutions come with the associated software that helps in the operation and automation of the scanning process to 3D scan the damaged part, of which the data output is typically Point Cloud Data (PCD).

Figure 3.5: Robotic arm mounted 3D scanners: (a)Artec RoboticScan [158]; (b)Creaform MetraSCAN 3D-R[159]; (c) FARO® Cobalt Array Imager[160]

Like all data, the PCD is likely to contain noise. Additionally, as the raw data from 3D scanners is typically a series of point cloud from different point of views, the sets of point clouds may not be accurately aligned. As such, the PCD has to be cleaned of noise and aligned. These two processes of noise removal and registration are readily available in commercial software, like Geomagic Design X, as discussed previously in Section 2.3.4. Examples of noise removal and point cloud alignment are as shown in Figures 2.29 and 2.30. Additionally, the default parameters can be tuned in advanced to the scanner system, allowing the process to be automated with minimal human intervention.

With the cleaned and aligned 3D scan, the useful information can then be extracted from the PCD.

Damage Analysis

The next step of extracting useful information from the PCD is to detect the damage, and is the first part of damage analysis. Ideally, the damage can be detected by a simple comparison, or Boolean operation, of the scanned model and the as-built model. However, the as-built model may not be readily available [25, 26], and even if it were available, the comparison may not be feasible due to the acceptable distortion of the worn part [26]. Alternatively, the damage can be detected directly from the scanned PCD. Whilst there has been some research done into automated damage detection from PCD, all of the proposed approaches have assumptions and limitations in terms of scope of damage types detected, assumptions on part geometry, or require additional inputs, which prevent them from being implemented as a comprehensive solution, as discussed in Section 2.3.5. Additionally, there is no relevant works done on the classification of damage type from PCD. As such, a comprehensive automated damage detection

and classification framework is proposed to address these limitations, and described in detail in Chapters 4 and 5.

Nominal Model Comparison/ Reconstruction

Within the inspection step of automated remanufacturing is also the sub-step of acquiring the nominal model, which is the model of the part without the damage, as it was designed. In traditional remanufacturing conducted manually, a virtual model may not be necessary, as an experienced remanufacturing engineer could decide based on his professional experience on how the part should be repaired to. However, for remanufacturing to be automated, it is necessary to acquire the nominal model, so that the scan data of the damaged model can be compared to the nominal model to identify how the repair is to be done, in terms of how much material the additive manufacturing machines should add, and how much material the subtractive manufacturing machines should remove. This nominal model may be obtained from available documentation, although it may not be readily available [25, 26], or is unusable due to the acceptable distortion of the worn part [26]. Alternatively, this can be produced by reconstructing it from the scanned PCD of the damaged part, also known as nominal model reconstruction. There are some literature proposing automated approaches for nominal model reconstruction, like the extension of the cross section of a turbine blade model [26] (see Figure 3.6), or the extension of the neighbouring undamaged region based on surface curvatures [161]. However, in these researches, the damages were identified manually. As such, it is important to accurately identify the damages automatically, which is the work done and presented in Chapters 4 and 5.

Figure 3.6: Nominal Model Reconstruction of blade tip: (a) Scanned model; (b) Surface extrapolation; and (c) Reconstructed model, adapted from Gao, Chen [26]

Repair Decision

The last step as part of the inspection process is the decision on whether the part is repairable or not. This is done based on the identification of the type and size of damage, of which the feasibility of a repair job is studied. The size of the damage can first be extracted via a Boolean operation between the scanned damage model and the nominal model. Together with the damage classification, a repair decision can be made, based on different approaches available in existing literature, like the use of life-cycle cost parameters [162-164], Level Of Repair Analysis (LORA) [165-167], or a combinatorial approach [168].

3.3.5 Repair Process

After a damaged part is deemed to be feasible to repair, a repair process has to be developed to repair the damaged region.

Repair Process Planning

The repair process begins with the planning sub-step, where the details of additive and subtractive manufacturing techniques is planned. Typically, the order of repair is subtractive manufacturing, then additive manufacturing, followed by subtractive manufacturing. The initial subtractive manufacturing process is necessary to prepare the damage surface to ensure a proper bond to the added material in the additive manufacturing process, by cleaning out the damaged region from any impurities on the surface [169], or reducing the surface roughness of the surface where material is to be added [170]. Additionally, the

subtractive manufacturing process is necessary to modify the geometry of the damage region to be accessible to the additive manufacturing machine, for example the widening of a crack by milling to allow for the additive manufacturing tool head to reach the bottom of the crack. Thereafter, the additive manufacturing process can be conducted, to replace material lost due to the damage, or from the initial subtractive manufacturing process. However, reviews of the state-of-the-art additive manufacturing processes have identified limitations in terms of dimensional accuracy and surface roughness [171-173]. As such, to obtain a higher dimensional accuracy and low surface roughness, it may be necessary to add more material in the additive manufacturing step than necessary, followed by machining it down in the final subtractive manufacturing step, as the subtractive manufacturing technologies are able to produce surfaces with better accuracy and surface roughness than that of additive manufacturing [174, 175].

Apart from the order of the manufacturing steps, the toolpaths have to be generated and simulated before the actual repair process. This area has been greatly researched, with researchers formulating mathematical models and algorithms that can automate this step, together with the choice of manufacturing method [176-180], as shown in Figures 3.7 and 3.8.

Figure 3.7: Separation of additive repair volume (Blue) and subtractive repair volume (Red), adapted from Zheng, Qureshi [180]

Figure 3.8: Hybrid manufacturing toolpath generation, adapted from Ruan, Eiamsa-ard [176]

Hybrid Manufacturing

The repair process can then be carried out in-situ with a mobile manufacturing machines using industrial robot, such as the MX3D Robotic Arm Welding Machine (see Figure 3.3) for additive manufacturing, and the Capriol Autonomous CNC Mobile Robot (see Figure 3.4) for subtractive manufacturing. Alternatively, a true hybrid manufacturing machine can be used, like that of the LASIMM Project (see Figure 3.9) which utilises different robotic arms for the different manufacturing tools, or the Hybrid Manufacturing Technologies interchangeable hybrid tool head mounted on an industrial robot (see Figure 3.10). Given the portability of these commercially available hybrid manufacturing machines, in-situ remanufacturing is possible.

Figure 3.9: LASIMM project hybrid manufacturing machine, adapted from [181]

Figure 3.10: Hybrid manufacturing technologies (a) Interchangeable hybrid tool heads; (b) Mounted on a robotic arm, adapted from [182]

3.3.6 Evaluation

To assess the results of the repair, the repaired part is to be evaluated.

Data Acquisition

For the evaluation process to be automated, digital data has to be acquired for the system to evaluate the repaired part. As the primary focus of this repair process is to bring the damaged part back to the geometry of the intended design, the data required is of the geometry of the repaired part. As such, the same 3D scanners used in the previous inspection step can be used to digitise the repair part, to acquire a 3D point cloud for this data acquisition sub-step.

Data Analysis

The scanned point cloud data can then be analysed to evaluate the repair process. This can be done by comparing it with the nominal model from the previous sub-process, using a simple Boolean operation after the scanned model has been aligned to the nominal model. If the deviation between the models is within the pre-specified tolerances, the geometry of the repaired part can be accepted, otherwise the repair process can be repeated as necessary.

Testing

Once the part has been repaired, it is also necessary to check the condition of the repaired part apart from the geometry, based on industrial requirements. This could be in the form of mechanical load testing, or NDT methods commonly used in the welding industry, like ultrasonic testing. Such tests are based on the requirements of the individual parts, and may differ between parts. As a guideline, the same test carried out during the manufacturing of the part should be conducted on the remanufactured part as well.

When the repaired part has passed these certification tests, the part can be certified to have a like-new condition, providing them with performance characteristics and durability at least as good as those of the original product, as per the definition of remanufacturing [1].

3.4 Summary

In summary, an automated in-situ remanufacturing process flow is proposed to increase the efficiency of remanufacturing, from a cost and time perspective. This is done by studying the process flow of remanufacturing currently, and eliminating the redundant steps. The necessary processes are then identified, with the available technology of each process detailed. With the formalism of the in-situ remanufacturing process flow proposed in this chapter, there can also be increased accountability, tracking and traceability on top of improved consistency, improved quality, improved efficiency and reduced cost in such remanufacturing, and is the motivation behind this work. The formalism of the in-situ remanufacturing process flow also enabled the identification of limitations in the state-of-the-art for individual processes, of which the PCD damage detection and classification steps were identified as a critical missing technology, and is addressed in the following chapters, of which solutions are proposed.

CHAPTER 4

Feature Extraction for Damage Detection

One specific process within remanufacturing identified in the literature review is the automated surface damage detection and classification. This is necessary in our automated in-situ remanufacturing, as the parts are typically large, and there is a unique need to merely find local defect, and add and remove material based on the neighbourhood region. It is also identified in the literature review that it is preferable to automate damage detection from 3D data over 2D images, as the 3D data of the detected damage region is required for the hybrid manufacturing process in the downstream repair procedure. The available state-of-the-art for 3D damage detection is found to be limited in terms of the scope of damage type detected, assumptions on part geometry, or require additional inputs. This hinders the automation of the damage analysis sub-step within the automated remanufacturing. In an attempt to solve the limitations of the state-of-the-art, machine learning is considered as a viable approach. However, past research done has identified that the computational efficiency is not optimised for geometrical deep learning. As such, this chapter documents how dimensional reduction can be performed on the geometric dataset, which is the 3D scanned data of the damaged part.

This chapter begins by the defining of prerequisites, to identify the necessary conditions and scope for the proposed automated damage detection framework to be implemented. Next, the concept for an automated damage detection and classification methodology is presented, following which the dimensional reduction steps for feature extraction is detailed. All the computational works presented in Chapter 4 and 5 are done within Matlab R2019b.

4.1 Prerequisites

The following prerequisites are required or assumed for the proposed framework:

- The original CAD model of the undamaged part is not available
- A sufficiently accurate PCD is obtained (dependant on selected 3D scanner, mounting solution of scanner, user-technique for scanner operation, and lighting conditions)
- The scan PCD is be unordered
- The PCD may be incomplete (i.e. partial scan, and/or occlusions)

4.2 Concept

As mentioned, from the literature reviewed, the computational efficiency is not optimised for geometrical deep learning. However, machine learning is still a promising approach, due to its success in damage detection from images. Hence, as a workaround, the 3D data will undergo dimensionality reduction via feature extraction. The extracted features are based on the characteristics of a damage in which a professional search for in a potentially damaged region. The proposed approach is similar to how a human positively identifies a damage, by seeking characteristics of a damage, whereby the damage can be confirmed if several characteristics of a damage is identified. An overview of the 11 features are as shown in Figure 4.1. The extracted features can subsequently be used to train a machine learning model, of which the model can be used to predict other points if they are part of a damage region or not, and is presented in Chapter 5.

Figure 4.1: Overview of features based on damage characteristics

4.3 Common Functions

To increase the efficiency of the proposed framework, some common functions used across different feature extraction algorithm were identified, including nearest neighbour, normal estimation, curvature estimation, edge point detection, edge tracing, and Hough transformation. These common functions will be used to process the PCD, and results fed into the different algorithms, or used when necessary within each algorithm, and is presented in this section.

4.3.1 Nearest Neighbours

Geometric properties of a point within point clouds is commonly defined by the local neighbourhood of points instead of the point itself. Hence, it is important to define the local neighbourhood of points, or nearest neighbours. The nearest neighbours can be defined as the k -nearest neighbour (see Figure 4.2(a)), or all points within a bounding sphere with radius r (see Figure 4.2(b)), depending on application.

Figure 4.2: (a) k -Nearest Neighbour where $k=7$, (b) Nearest neighbour based on bounding sphere with radius r

A spatial decomposition approach, the k -dimensional tree, or k -D tree, proposed by Friedman, Bentley [183], is adopted to organise the points in a k -dimensional space, as it guarantees a $O(\log n)$ complexity. The 3D Euclidean distance of a queried point to its neighbouring points are calculated, given by equation 4.1 [184].

$$d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2} \quad (4.1)$$

where p_x, p_y, p_z are the x, y , and z coordinates of point p , and q_x, q_y, q_z are the x, y , and z coordinates of point q . Thereafter, for finding the k -nearest neighbour,

the 3D Euclidean distances are ranked, of which k number of points with the lowest 3D Euclidean distance from the queried point are recorded as the k -nearest neighbours. If the points within a bounding sphere is to be identified, any point with the 3D Euclidean distance to the queried point less than the bounding sphere radius r is recorded as nearest neighbours.

4.3.2 Normal Estimation

Some scanner PCD includes the surface normal information at the location of the points. However, other scanners do not provide this information, but the normal can be estimated based on the tangent plane to the surface. The Principal Component Analysis (PCA) of a covariance matrix from the k -nearest neighbour proposed by Rusu [185] is adopted, with the covariance matrix given by Equation 4.2 below:

$$C = \frac{1}{k} \sum_{i=1}^k \xi_i \cdot (p_i - \bar{p}) \cdot (p_i - \bar{p})^T \quad (4.2)$$

where k is the number of nearest neighbour, ξ_i is the possible weight, usually 1, p_i is the query point, and $\bar{p}$ is the 3D centroid of nearest neighbours.

Thereafter, the eigenvalues λ_j and corresponding eigenvectors v_j of C are obtained, with $j \in \{1, 2, 3\}$, given by Equation 4.3 below:

$$C \cdot \vec{v}_j = \lambda_j \cdot \vec{v}_j \quad (4.3)$$

The resultant normal estimated is ambiguous in terms of direction, and may not be consistent over the entire PCD. The method proposed by Hoppe, DeRose [186] can be used, whereby a geometrically close surface will have the point normal consistently oriented, given by Equation 4.4 below:

$$\vec{n}_i \cdot \vec{n}_j \approx 1 \quad (4.4)$$

where $\vec{n}_i$ and $\vec{n}_j$ are neighbouring points.

The orientation of neighbouring point is then propagated from a random point, based on Equation 4.5, giving the results as shown by the blue arrows in Figure 4.3.

$$\vec{n}_i \cdot \vec{n}_j < 0 \Rightarrow \vec{n}_j = -\vec{n}_i \quad (4.5)$$

Figure 4.3: Normal estimation

4.3.3 Curvature Estimation

The curvature of the surface can also be estimated, based on the relationship between the eigenvalues of the covariance matrix from the normal estimation, using the Equation 4.6[185]:

$$\sigma_p = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \quad (4.6)$$

where λ_0 is the minimum eigenvalue. Results of the curvature estimation is as shown in Figure 4.4, where red is the regions of low curvature, and blue is the regions of high curvature.

Figure 4.4: Curvature estimation

4.3.4 Edge and Boundary Point Detection

The edge of the object refers to the interface between discontinuous surfaces. This abrupt change in geometry can be identified as a high curvature value. The detection of edges can thus be done by first thresholding of points having high curvatures, as obtained from section 4.3.3. Results are as shown in Figure 4.5(b).

Figure 4.5: (a) Input PCD, (b) Output PCD with high curvature points (Yellow)

From the previous step, points on the boundary of a surface would also appear as an edge. These boundary points occur when there is occlusion, forming a hole in the PCD, or where an incomplete scan is taken. To identify such boundary points, the algorithm proposed by Gumhold, Wang [187] is adopted, and the algorithm is as follows:

1. For every point p , project its k -nearest neighbours onto an xy -coordinate system orthogonal to the surface normal direction associated with point p ;
2. Compute the angle formed between the lines connecting point p and each pair of adjacent neighbouring points (e.g. the angle represented by the red arrow in Figure 4.6);
3. Find the points p with angles above a defined threshold, and is identified as a boundary point (see Figure 4.6);

Figure 4.6: Maximum angle for internal point (Green), and boundary point (Blue)

Mathematically, if $\Theta = \{\theta_1 \dots \theta_k\}$ is the array of sorted angles between lines formed by two points in the neighbourhood with a point p , p will be a point found on the boundary of a hole, given by equation 4.7:

$$\max (\alpha = \theta_{i+1} - \theta_i) \geq \alpha_{th} \quad (4.7)$$

where α_{th} is the maximum threshold angle.

The detected boundary points can be separated from the list of detected edge points, to give the true edge points on geometrical edges. Results of this algorithm are as shown in Figure 4.7(b), where the boundary edges are coloured red, while the geometrical edges left yellow coloured.

Figure 4.7: (a) Input high curvature points (Yellow), (b) Output boundary and edge distinction (Boundary points - Red, Edge points - Yellow)

4.3.5 Single Edge Tracing

Using the edge detection algorithm from the previous section, points with high curvature values are assumed to be an edge point. However, on a geometrical

corner of the scanned object, a band of high curvature points are identified as edge points, instead of a single line of points. This cannot be avoided by simply raising the curvature threshold and is due to one of the following reasons.

Firstly, a 3D scanner works by sampling points off the object's surface. If the physical object does contain a perfect sharp edge, the 3D scanner may not sample points at the sharp point of the corner. Instead, the scanner may sample points on both sides of the corner, resulting in a band of multiple points having high curvature values, instead of a single line.

Secondly, the physical part's edge would have small fillet on all the edges, due to the limitations in manufacturing, or wear. As such, the usual practice is that if the fillet has a significantly small radius, it is considered a corner, or a geometrical edge. This small corner may, however, be captured by the 3D scanner, if the scanner resolution is smaller than the fillet radius. This results in a band of high curvature points as well.

For some applications, including damage detection, it is beneficial to be able to identify the intended edge of the PCD of an object. While there are available line fitting algorithms for straight lines or curves, the edges of a damage typically do not conform to these basic representations. Hence, an edge tracing algorithm is proposed, based on the non-maximal suppression used in the Canny edge detector for 2D edges in images [60]. The proposed algorithm is as follows:

1. For each point detected as being located on an edge, compare the curvature with the nearest neighbourhood points within a bounding sphere with radius r . The radius used should be larger than the width of the band of edge points. A large r used would lead to a less segmented line, at the expense of details around corners and bent edges;
2. If the curvature of the surface where the queried point is located is identified to be the largest, i.e. is at the centre of the edge, store the point as an edge trace point;
3. For each edge trace point, search for the k -nearest neighbours with $k=2$, and record the line formed by linking the point with its 2 nearest neighbours as an edge trace line. (See Figure 4.8). A point may be used to

form more than two edge lines, if it is the 2-nearest neighbour of all the linked edge points, for example, a corner.

Figure 4.8: Tracing curvature local maxima points

The result of the proposed algorithm is as shown in Figure 4.9(b)

Figure 4.9: (a) Input detected edge points (Teal), (b) Output traced edge (Green)

4.3.6 Hough Transformation for Straight Edge Detection

Hough transformation is a feature extraction technique using a voting scheme, originally used for machine analysis of bubble chamber photographs [188]. In a 3D context, it is commonly used in plane detection[189, 190], and also in line detection[191, 192]. For this thesis, the Hough transformation used is for the line detection as proposed by Jeltsch, Dalitz [191], and utilises the Hough parameter space regularisation.

The line is parameterised using for parameters x' , y' , ϕ , and θ , as proposed by Roberts [193], where x' and y' can be obtained using Equations 4.8 and 4.9 below:

$$x' = \left(1 - \frac{b_x^2}{1+b_z}\right)p_x - \left(\frac{b_x b_y}{1+b_z}\right)p_y - b_x p_z \quad (4.8)$$

$$y' = \left(\frac{b_x b_y}{1+b_z}\right)p_x + \left(1 - \frac{b_y^2}{1+b_z}\right)p_y - b_y p_z \quad (4.9)$$

where p is an arbitrary point with $\vec{p} = (p_x, p_y, p_z)$, and b is given by the Equation 4.10:

$$b = \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} \cos \phi \cos \theta \\ \sin \phi \cos \theta \\ \sin \theta \end{pmatrix} \quad (4.10)$$

where ϕ is the horizontal orientation, and θ is the altitude, or elevation.

A point on the line can thus be obtained with Equation 4.11 below:

$$p = x' \cdot \begin{pmatrix} 1 - \frac{b_x^2}{1+b_z} \\ -\frac{b_x b_y}{1+b_z} \\ -b_x \end{pmatrix} + y' \cdot \begin{pmatrix} -\frac{b_x b_y}{1+b_z} \\ 1 - \frac{b_y^2}{1+b_z} \\ -b_y \end{pmatrix} \quad (4.11)$$

The Hough transformation is then conducted using tessellated icosahedron (see Figure 4.10), where the closest x' and y' values are found, and the respective bin gains a vote. Subsequently, local maxima can be found from the accumulator arrays, to be identified as lines.

Figure 4.10: Icosahedron after 0, 1, 2, and 3 tessellation steps, adapted from [190]

An example of the Hough transformation was done on the detected edged, as shown in Figure 4.11(b), with the highest bin count shown by the green line, and the second highest shown by the blue line.

Figure 4.11: (a) Input detected edge points (Yellow), (b) Hough transform detected lines (Green & Blue)

4.4 Feature Extraction

With the common functions detailed, the actual dimensionality reduction processes can be introduced. The feature extraction process is broken down into three categories, namely the local analysis, the neighbourhood analysis, and the global analysis.

4.4.1 Local Analysis

The first set of features of a potential damage region is based on the characteristics within the damaged region itself.

4.4.1.1 Hole

Damage in the form of deep recessions would introduce self-occlusion to optical 3D scanners (see Figure 4.12). This means that the bottom area of the recession cannot be captured by the optical 3D scanner, and the missing data forms a hole in the 3D point cloud. This hole is easily identifiable as 3D scanners have a fairly consistent resolution over an object surface. Such boundary points can be identified using the algorithm as discussed in Section 4.3.4.

Figure 4.12: Hole in point cloud from occlusion

In some instances, the PCD is from a partial scan, and the edges of the point cloud is also identified as a border. In such instances, the surrounding edge should not be considered as part of the damage. All boundaries form a closed loop, regardless of it being an internal (hole) or external boundary, and can be traced from a random seed. A count of the number of points that make up the boundary can be compared to a defined threshold, and boundary with large point count is likely to be the external boundary from an incomplete scan, and hence ignored. Results of this approach conducted on a part as annotated in Figure 4.13(a) are as shown in Figure 4.13(b), where the yellow points are the detected edge points, the blue points are the external boundary points, and the green points are the internal hole boundary points.

The potential damaged region can then be grown from the internal hole boundary points, by including points directly or indirectly connected to the boundary points, stopping when all connection points before an edge is included. The result is as shown by the red points in Figure 4.13(c).

The first limitation of this approach is that designed deep recessions, like drilled holes, are not damage, but would still be identified by this algorithm (see Figure 4.13(c)). This is because the inaccessibility of such designed features would also be captured as a hole in the PCD.

Figure 4.13: Damage detection from hole in PCD – (a) Input PCD, (b) Edge and boundary detection, (c) Potential damage points (Red)

The second limitation of this approach is that the hole in the point cloud could be due to errors in the scanning. During the 3D scanning, poor practices could lead to missing 3D data and a hole in the PCD, which will be captured by this algorithm. Limitations of the scanning object, including surface reflectance properties, transparency, occlusions and accessibility limitations would also result with holes in the point cloud, and should be minimised by selecting an appropriate 3D scanner, as detailed in Section 2.3.3, together with good 3D scanning practices.

The last limitation of this approach is that the occluded region of the damage has to be wider than resolution of the 3D scanner. If the size of the hole in the point cloud is similar to the PCD resolution, the hole will not be detectable, and this algorithm would not be able to extract this feature.

4.4.1.2 Edge Shape Irregularity

The edge of damage refers to the interface between the damage region and the original surface. It is typically detected by the edge detection algorithm described in section 4.3.5, unless it is the rare cases whereby the damaged surface forms a smooth continuous interface with the original surface.

The edge of the defect is usually irregular, meaning that it is neither straight nor has a simplified mathematical representation, typical of most engineering edges. Hence, to detect a damaged edge, the edges can be traced, and checked for its consistency.

The proposed algorithm uses the following steps:

1. For each edge trace point detected using the algorithm proposed in Section 4.3.5, compute the angle between the 2 edge trace lines it is part of. If an edge point belongs to more than 2 edge, ignore the point;
2. For each detected edge trace point, find the k-nearest edge trace points;
3. For the k-nearest neighbours and their associated edge angles as calculated from step 1, compute the standard deviation of the angles. If the standard deviation exceeds a pre-defined threshold, the point is identified as potentially being part of a damage;

Results of this algorithm conducted on a part shown in Figure 4.14(a) is shown in Figure 4.14(b), where the green edge trace are undamaged edges with low deviation of edge angles, while the red edge trace are potential edges of damage with high standard deviation of edge angles exceeding a predefined threshold.

Figure 4.14: (a) Input PCD with traced edges (Green), (b) Edge shape irregularity results (Red)

The limitation of this algorithm is that it is highly sensitive to noise, as the noise would result in inconsistently oriented edges, and be detected by the algorithm (see Figure 4.14 (b)). Additionally, if the edge of the damage region is regular, the damage region will not be identified by this algorithm.

4.4.1.3 Edge Orientation Irregularity

Most traditional engineering part have multiple edges which are parallel to each other. On the contrary, damaged edges are usually in an unspecified direction. Hence, by comparing the orientation of edges in a damaged part, erroneously oriented edges may signify that it is the edge of the damage.

The proposed algorithm uses the following steps:

1. From the edge trace lines detected using the algorithm proposed in Section 4.3.5, parameterise the lines into the parameters x' , y' , ϕ , and θ , using Equations 4.8 to 4.11;
2. The edge trace lines are sorted into ϕ and θ bins based on a icosahedron (see Figure 4.10), as similarly oriented edges would have lines with similar ϕ (horizontal orientation), and θ (elevation);
3. If the count in a bin exceeds a predefined threshold, there is sufficient edges with that orientation to be assumed to be a designed edge. Hence, all the other edges that do not fall within these bins are assumed to be potentially from a damage.

Results of this algorithm is as shown in Figure 4.15(b), where the green, blue, and yellow line are the traced edges with a large quantity of edges in the same direction, while the red line are edges oriented in an infrequent direction.

As this approach is tuned to detecting erroneously oriented edges in engineering objects assumed to have parallel edges, round edges will be identified by this algorithm (see Figure 4.15 (b)).

Figure 4.15: (a) Input PCD with traced edges (Green), (b) Edge orientation irregularity results (Red)

4.4.1.4 Rough Surface

Typical engineering surface are manufactured to a specific surface roughness. Damaged surface on the other hand tends to be irregular, leading to a higher surface roughness. A high surface roughness would imply a larger curvature value, with a large fluctuation of curvature values over an area. As mentioned by Teza, Galgaro [133], a change in shape would also result in a large curvature value, and hence it is not a significant parameter to use for damage detection. Instead, the standard deviation of curvature of neighbouring points can be computed using Equation 2.11, and if it is larger than a defined threshold, the point will be identified as part of a potential damage region.

For a sample patch of PCD for a damaged (see Figure 4.16(a)) and undamaged (see Figure 4.16(b)) region, the histogram result is obtained as shown in Figure 4.17.

Figure 4.16: (a) Damaged and (b) Undamaged point cloud region

Figure 4.17: Damaged and undamaged region curvature distribution

The limitation of this approach is that the damage has to be irregular. Hence, a damaged region with a smooth surface would not be identified by this algorithm, for example a perfectly chipped corner. In addition, the damaged part should not be textured, or have a high surface roughness, as it would be identified by this algorithm as well.

4.4.2 Neighbourhood Analysis

Another characteristic of a damage is that it disrupts the regularity of the engineering part. As such, to extract the features of a potential damage, some common geometries are first detected from the PCD, and the PCD is compared to the fitted model. Deviations from the fitted model would be identified as potential damage. The chosen geometries to be checked for are planes, edges, corners, and cylinders. While such an analysis can be conducted on more complex geometries such as cone, sphere, and torus, they are uncommon geometries in engineering components, and hence excluded. Freeform surfaces formed with higher order bivariate polynomials, or splines can also be detected and compared, but the methodology and results would be similar to the detection of rough surface as described previously in Section 4.4.1.4, hence is omitted.

4.4.2.1 Deviation from Plane

Engineering components typically have multiple planar surfaces. A damage in the planar surface can easily be identified if there are any holes or deviations from a detected plane.

The plane detection algorithm used is the RANdom SAmple Consensus (RANSAC) iterative method, proposed by Fischler and Bolles [194], as it is reliable and robust [195, 196]. The RANSAC algorithm makes use of the following steps:

1. Randomly select 3 unique points which are not collinear $\{p_i, p_j, p_k\}$ from P (since 3 non-collinear unique points define a plane);
2. Extract the plane formed by the 3 points;
3. Compute the distance of all the points to the identified plane;
4. Count the number of points where the distance falls below a specified threshold;
5. Iterate steps 1-4 k times;
6. The sets with points count above a specified threshold is identified as planes;

The number of iterations k can be estimated using equation 4.12, given by Fischler and Bolles [194].

$$k = \frac{\log(1-z)}{\log(1-w^n)} \quad (4.12)$$

where z is the user defined desired probability of success, typically set between 0.9 to 0.99, w is the probability of selecting one good inlier, and n is the number of data points per selection, which is 3 for the case of plane detection.

After each plane is detected, points can be identified as potential defects if it falls within a specified range from the planar points, where the lower limit is to compensate for deviation due to surface roughness or scanner inaccuracies, and the upper limit is to remove points that do not belong to a damage on this plane. However, the edge of the plane would contain neighbouring points which are part of another face, but would be wrongly detected as potential damage. These points can be ignored by creating a convex hull about each plane, and ignoring points within a defined distance from the convex hull edges. The result of this approach

conducted on a part as annotated in Figure 4.18(a) can be seen in Figure 4.18(b), where the blue, green and teal points are on detected planes, and the red points are identified potential damage.

Figure 4.18: (a) Input PCD, (b) Plane fitting damage detection (Planes – Green, Blue, & Teal, Damage – Red)

Apart from only being able to detect damages on planar surfaces, the limitation of this method is that original protrusions or cavities within the planar faces of the engineering part, which are not damage, are also identified by this algorithm (see Figure 4.18(b)).

4.4.2.2 Edge Discontinuity

If a damage forms on a single continuous edge of a part, the edge will be disrupted, and the region between colinear edges could be a potentially damage area.

Using the Hough parameter space regularisation, collinear edges can be identified as they would have the same parameters x' , y' , ϕ , and θ . The absent portion of the collinear edges could thus be considered as a potentially damaged edge. Results of this approach conducted on a part as annotated in Figure 4.19(a) can be seen in Figure 4.19(d), with the intermediate results shows in Figures 4.19(b-c).

Apart from being able to detect potential damage which falls on the edge, this approach may wrongly capture designed discontinuities in edges. To minimise this limitation, a threshold can be set for the distance between colinear edges, to be as big as the largest expected damage, with all other larger discontinuity assumed to be an intended feature rather than a potential damage.

Figure 4.19: (a) Input PCD, (b) Edge point detection (Teal), (c) Hough transform detected straight edges (Green), (d) Potential straight edge damage identified

4.4.2.3 Corner Discontinuity

A damage could also occur at the location where three or more edges meet, or a corner. This can be identified by potentially intersecting lines at a point, when extended towards the same direction.

In the Hough parameter space, each distinct point has a finite set of tessellated icosahedron lines passing through it. For each set of tessellated icosahedron lines of a point, if there are three or more detected lines represented by the local maxima, the point is potentially a corner. The scanned PCD could then be checked against the detected corner, and the deviation from the corner is identified as a potential damage. Results of this approach conducted on a part as annotated in Figure 4.20(a) can be seen by the red lines in Figure 4.20(d), with the intermediate results shows in Figures 4.20(b-c).. Similarly, the distance between the missing corner and the existing edge from the PCD could be threshold, to ensure that unintended corners are not identified as damage.

Figure 4.20: (a) Input PCD, (b) Edge point detection (Teal), (c) Hough transform detected straight edges (Green), (d) Potential corner damage identified (Red)

Apart from being limited to detecting potential corner damages, this approach may wrongly identify an intentionally missing corner as a potential damage, for example a corner fillet (see Figure 4.20(d)).

4.4.2.4 Deviation from Cylinder

Engineering parts also tend to have cylindrical components, for example axes or shafts. As such, the PCD could be searched for potential cylinders, and checked for damage.

The same method as the plane fitting, RANSAC, is used to detect cylinders. The difference, however, is that 5 points are selected instead of 3 points used in the

plane fitting RANSAC algorithm. Thereafter, points can be identified as defects if it falls out of the specified threshold, where the lower limit is to compensate for deviation due to surface roughness or scanner inaccuracies, and the upper limit is to remove points that do not belong to this cylinder (see Figure 4.21(b)).

Figure 4.21: (a) Input PCD, (b) Cylinder fitting damage detection (Cylinder – Green & Teal, Damage – Red)

Apart from only being able to detect damages on cylinders, the limitation of this approach is that original protrusions or cavities within the cylindrical faces of the engineering part which are not damage, like keyways, could be captured by this algorithm as potential damage.

4.4.3 Global Analysis

Finally, potential damage can be identified via the global analysis, based on the principal where the engineering part has repeated features. By using a Boolean operation to compare the feature containing a damage with one that contains no damage, the damaged area can be identified.

Traditional engineering parts tend to be symmetrical [197-200]. By comparing the symmetrical areas of a point cloud, an undamaged portion of the point cloud can be used to compare the damaged region of the point cloud, and the damaged region can be detected from the deviation in the compared points.

An object may possess one, or a combination of numerous symmetries, and is described in detail by Rosen [201]. However, for the purpose of finding symmetries in engineering parts, the focus will be on the two common

symmetries, namely planar symmetry and axial symmetry. Additionally, patterned features will be identified and compared to, to detect potential damages.

4.4.3.1 Planar Symmetry Deviation

An object possesses a planar symmetry when each point on the object surface corresponds to another surface point on the opposite side of the symmetry plane of the object (see Figure 4.22).

Figure 4.22: Example geometry containing 1, 2, and 3 planar symmetries

While an object may possess an infinite number of planes of symmetry, engineering parts typically has three or less symmetrical planes. For the purpose of damage identification, a single axis of symmetry is sufficient to detect potential damage, as it is highly unlikely that exactly the same damage is present on both sides of symmetry of the damaged part. However, additional symmetrical planes would be useful to provide more points to compare the potentially damaged surface, of which the number of points P' available for comparison is given by Equation 4.13 below:

$$P' = 2^n - 1 \quad (4.13)$$

where n is the number of planes of symmetry.

The algorithm used for planar symmetry detection is adapted from Wang and Oliveira [202], which utilises Hough transformation due to its insensitivity to noise. The algorithm is as follow:

1. Compute the bisector plane for each pair of points in the PCD;
2. Cast a vote for the computed plane;
3. If 1, 2, or 3 planes alone receives the most vote, it is a symmetry plane;

When a symmetry plane is found, the point cloud can be copied and reflected in the symmetry plane. The Euclidean distance between each point from the original dataset and the closest point of the reflected dataset will be computed, and the points with an Euclidean distance larger than a predefined threshold is identified as a potential damage point. The results of the points with Euclidean distance exceeding a threshold is as shown by the red points in Figure 4.23(b). When a part contains more than one symmetry plane, the point cloud can be copied and reflected in the other symmetry plane for comparison, to return a more robust result (see Figure 4.24(b)).

Figure 4.23: (a) Input PCD, (b) Damage detection from single symmetry (Red)

Figure 4.24: (a) Input PCD, (b) Damage detection from double symmetry (Red)

The limitation of this method is that the points which are on the opposite side of the symmetry plane is also identified as a potential damage region, and hence the

actual damaged region has to be confirmed by comparison in second symmetry plane if available, or by other damage detection algorithm in the machine learning model.

Additionally, components with small asymmetrical features could also be wrongly identified as a damage.

4.4.3.2 Axial Symmetry Deviation

An object possesses axial symmetry when each point corresponds to an entire circle of points, about the axis of symmetry, with the plane of the circle orthogonal to the axis. Cylinders is the only basic geometry which possesses such symmetry, and shafts made up of a series of coaxial cylinders will exhibit the same axial symmetry (see Figure 4.25).

Figure 4.25: Example geometry containing axial symmetry

Using the results of the plane detection in Section 4.4.3.1, if a series of planes received as many votes, the intersection of the planes would return the axial symmetry. Thereafter, the component can be rotated about the axis of symmetry twice, at 120° interval, and similarly, the Euclidean distance between each point from the original dataset and the closest point of the rotated dataset will be computed, with deviations identified as potential damages. Sample results are shown in Figure 4.26(b), where points with Euclidean distance exceeding a

defined threshold is coloured red. The reason for the choice of rotating twice at 120° instead of once at 180° is to ensure a positive identification of the defect region, as the undamaged region will not have a recorded deviation in both comparisons (see Figure 4.27).

Figure 4.26: (a) Input PCD, (b) Damage region from axial symmetry comparison (Red)

Figure 4.27: Positive identification of defect region in 120° rotation

Similar to the cylinder fitting damage detection algorithm, the limitation of this method is that axial symmetrical parts containing additional features like slots and keyways will be captured by the algorithm as well.

4.4.3.3 Patterned Feature Deviation

Engineering parts may not only be symmetrical, but also have repeated structures with the part itself. As such, the duplicated structure can be compared with each other, and the deviations between the identical structures can be a potential damage. The algorithm is based on common PCD object recognition methodology, except that the object used for detection is a portion of the PCD.

To be able to detect repeated structures, the point cloud is first searched for keypoints, or interest points. Such keypoints are points within the PCD that is unique to the local neighbourhood point distribution. This means that the keypoints should ideally be detected across all the repeated structures in the part. The detected keypoints are then augmented with a descriptor, or an information, of its neighbourhood.

The keypoint detection algorithm used is based on the 2D Scale-Invariant Feature Transform (SIFT) algorithm by Lowe [203], where the local maxima and minima in a Difference-of-Gaussians (DoG) scale-space is detected. The use of SIFT for 3D keypoint detection was first implemented by Flint, Dick [204]. Instead of using the surrounding pixels in the 2D implementation for comparison, the k -nearest neighbour is used. Additionally, instead of computing the intensity gradient for the DoG, the normal gradient, or curvature, is used to detect interesting points with large geometrical changes, or a corner. The user-specified inputs are the number of down sampling operations (i.e. the number of octaves to reach the smallest scale), the number of scales we generate between down sampling (i.e. the number of layers per octave), and the minimum threshold for the normal gradient value for a point to be qualified as a keypoint. Sample result for the detected keypoints is shown by the blue squares in Figure 4.28.

Figure 4.28: 3D SIFT (on curvature) Keypoint detection (Blue)

A 3D descriptor is then computed for each keypoint, of which the chosen descriptor is the Signature of Histograms of Orientations (SHOT) by Tombari, Salti [205]. This is done by computing the histogram of normal for a 3D volume, following which grouping all the local histograms to form a robust and high descriptiveness 3D descriptor.

Thereafter, a structure is sampled from the PCD. This is done by randomly selecting a seed keypoint, and its bounding sphere, of radius r , neighbouring keypoints. This sample patch will be compared with the rest of the keypoints for correspondences, where similar model descriptors based on Euclidean distance is computed and threshold.

Subsequently, the correspondences are checked for geometry consistency, and is an extension of the method proposed by Chen and Bhanu [206], which utilises the Iterative Closest Point (ICP) algorithm. The ICP algorithm would yield a good result if there is a large number of correspondences between the sampled keypoints and keypoints from the rest of the PCD.

Finally, if a positive correspondence is detected, where the majority of keypoints in a pair of keypoint sets compared are found to be closely aligned and represent a geometric match, the alignment can be further refined as per the ICP algorithm, and deviation between the sample patch and the rest of the PCD can be computed. The area with detected deviation past the pre-defined threshold will be identified by the algorithm as a potential damage area, and the sample results can be seen as red points in Figure 4.29.

Figure 4.29: Damage detection from patterned structure (Red)

Alternatively, if there is no correspondence, a new point is sampled, and the process is repeated. Similar to the damage detection from symmetric features, multiple patterned structure can be compared to give a positive identification of a damaged region.

In addition to being limited to detecting damage from only parts with repeated structures, this approach is sensitive to the pre-defined parameters, which may result in the erroneous identification of potential damages.

4.5 Summary

Machine learning is identified as a promising approach for 3D damage detection, although prior literature has identified that the computational efficiency is not optimised for geometrical deep learning. As such, this chapter details how the geometric 3D scanned data can undergo dimensionality reduction, to allow the extracted features to be used to train a machine learning model and subsequently detect and classify damage. The chapter begins by first defining of prerequisites, to identify the necessary conditions and scope for the proposed automated damage detection framework to be implemented. Next, the concept for an automated damage detection and classification methodology is presented, including the identification of 11 damage characteristics that humans look out for when finding damage regions. Some common functions that are reused in multiple of the feature extraction algorithms are detailed, following which the 11 feature extraction approaches are presented, together with the sample results of each feature extraction algorithm. These extracted features can subsequently be used to train a machine learning model in Chapter 5.

CHAPTER 5

Automated Damage Detection and Classification

To work towards automated in-situ remanufacturing, the inspection process of the remanufacturing has to be automated as well, specifically damage detection and classification. It is preferable to automate damage detection from 3D data over 2D images, as the 3D data of the detected damage region is required for the hybrid manufacturing process in the downstream repair procedure. However, the literature review conducted reveals that the available state-of-the-art for 3D damage detection is found to be limited in terms of the scope of damage type detected, assumptions on part geometry, or require additional inputs. As such, machine learning is identified as a promising approach, due to its success in damage detection from images. However, the unoptimized computational efficiency for geometric learning, and lack of training data inhibits the direct application of machine learning for damage detection from 3D data. A feature extraction approach was presented in Chapter 4, whereby the geometric data undergoes dimensionality reduction, allowing machine learning to be applied on the data. This chapter addresses the second limitation, by presenting the steps taken to overcome the limitations of having a lack of training data, as well as the implementation of machine learning for 3D damage detection and classification.

This chapter begins by listing the assumptions made for the proposed approach. Next, the creation, collection, and annotation of data necessary for this research in detailed. Subsequently, the steps taken to implement machine learning to automate damage detection is presented, including the training and comparison of machine learning models, using the extracted features. Additionally, the proposed solution is extended to the next sub-process of inspection, which is the classification of damage type, which would be useful in the repair decision step in remanufacturing.

5.1 Assumptions

For the proposed approach, the following assumptions were made, based on the scope defined by this thesis.

5.1.1 Size

While the scope of this thesis is for the marine and offshore industry, where parts are typically large, the proposed approach is independent of the size of the damaged part, but dependant on the scan resolution relative to the part instead. More specifically, the scan resolution should be sufficient to capture the features of the part, as well as the details of the damaged region. For this thesis, the 3D scanner chosen for this study, the FARO Edge Scan Arm HD (3.7m), has a resolution of 40 μ m, and hence the minimum detectable damage size should be larger than twice that of the resolution.

5.1.2 Shape

The damaged part is assumed to be that of a functional engineering part, which is defined to have regular geometries like planar surfaces, cylindrical surfaces, spherical surfaces, edges, corners, radiused fillets, and chamfers, or smooth freeform surfaces like that found on a turbine blade. This assumption is made to allow for ease of predictability in the undamaged region as compared to the damaged region.

The other assumption on the shape of the damaged part is from the perspective of the 3D scanner and the scanning process, where the damaged region has to be accessible without occlusion, so that the 3D data can be obtained to be processed with the proposed approach, as detailed in section 4.1.

5.1.3 Material

The damaged part material is assumed to be that of a metal, as the scope of this thesis is for the marine and offshore industry. The damage region is thus assumed to have similar geometric characteristics between different metals, which is different from other materials, for example the damage region on that of plastics.

The material is also assumed to have optical properties that are suitable for it to be scanned by the optical 3D scanner.

5.1.4 Damage Types

The extent of a physical damage can vary from micro damage to a complete failure, where the severed piece contains significant functional features. However,

for this thesis, the scope is limited to surface damages large enough to be detectable by optical laser scanners, and has not been severed.

There are many types of surface damages and classifications of them, with the most established reference being the damage types in ISO 8785:1999 [207]. However, for this thesis, a detailed classification with 31 different types of surface damage is not required. Instead, a broader damage classification based on the geometry of the damage region is used, with the following five categories: surface wear, pit or corrosion, broken damage, crack, and plastic deformation. The first four categories are adapted from Zhang, Yang [208], and the fifth category of deformation was added to make the list complete for our application. These five damage categories were chosen as they include the typical damage found in the marine and offshore industry, and is the scope of the proposed approach. Additionally, each category has a distinct difference in the damage geometry, and hence the differing algorithms would have different success rates in detecting the different damages, which will be further elaborated later in this chapter.

Surface Wear

Surface wear is the erosion of the surface, seen in parts with sliding contact (see Figure 5.1(a)). The damaged region is typically shallow and wide (see Figure 5.1(b)).

Figure 5.1: Surface wear: (a) Photo adapted from D'Antonio [209]; (b) Cross Sectional Representation

Pitting or Corrosion

Pitting or corrosion is a series of small sized holes (see Figure 5.2), commonly caused by extreme environments.

Figure 5.2: Pitting: (a) Photo adapted from Belzona [210]; (b) Cross Sectional Representation

Broken damage

Broken damage is a damage caused by impact or cyclic loading, where there is a significant amount of material loss (see Figure 5.3). The damage region is typically similarly wide and deep. A damaged part severed into two pieces may be categorised into a broken damage. However, if the broken off part is too large, it would be too costly to repair via additive manufacturing. Instead, alternative approaches like welding the broken off part may be considered, but is out of the scope of this research.

Figure 5.3: Broken Damage: (a) Photo adapted from kansascitytdi.com [211]; (b) Cross Sectional Representation

Crack

Cracks are damage caused by impact or cyclic loading (see Figure 5.4(a)), and the damage region is typically thin but deep (see Figure 5.4(b)).

Figure 5.4: Crack: (a) Photo adapted from Engineering [212]; (b) Cross Sectional Representation

Plastic Deformation

Plastic deformation refers to the irreversible change in shape or size of an object (see Figure 5.5). It is also included as a form of damage, as the functionality and integrity of the part may be compromised, and should be repaired. Although there is a possibility of plastic deformation without any loss or gain in volume of the part, it is usually accompanied with other types of damages, resulting in a difference in volume of deformed part.

Figure 5.5: Deformation: (a) Photo adapted from Sardar and Bandopadhyay [213]; (b) Cross Sectional Representation

5.2 Experimental Data

Like all experiments, data is necessary to be experimented upon. However, unlike image-based research, there is limited 3D data of damages available in open source repositories in the internet. Additionally, these 3D data available are typically of concrete structures, including buildings and bridges, and are inadequate for the scope of this research. As such, this section looks at the

collection of data necessary for this research, as well as the processing and annotation required for benchmarking.

5.2.1 3D Data Acquisition

The first essential step is the collection of data. It is important to investigate the data acquisition process as poor practices would lead to erroneous data. These would lead to downstream errors throughout the experimental processes, making the experimental results unreliable. To obtain 3D data of a damaged component, a 3D scanner is to be used to capture the spatial information of both the part, as well as the damage itself. As there are numerous 3D scanners commercially available, a 3D scanner that is most suitable for our application is to be chosen. For our purpose of scanning a damaged object, where a high accuracy scanner for digitising potentially challenging damage surfaces is required, the FARO Edge (3.7m) ScanArm HD was chosen to scan damaged parts, based on the results from Section 2.3.3.

5.2.2 3D Data Representation

3D data can be represented in various ways as shown in Figure 5.6, including descriptors, graphs, mesh, multi-view, octree, point cloud, projections, RGB-D, and voxels [214]. They can be classified into Euclidean-structured data and non-Euclidean data, where the Euclidean-structured data refers to 3D data that has an underlying grid structure that allows for a global parametrization and a common system of coordinates, while the non-Euclidean data do not have the gridded array structure where there is no global parametrization. The different representations can be acquired as raw data from different 3D scanners and can also be converted from one form to the other. However, for this research, the type of 3D representation chosen to be worked on is Point Cloud Data (PCD), as it is the most common raw data form of 3D scanners, including the selected scanner for this research. Additionally, the use of the PCD as scanned by the scanner would eliminate any potential conversion errors, as well as remove the need for additional time or computational resources for the conversion.

Figure 5.6: 3D data representations, adapted from Ahmed, Saint [214]

5.2.3 Data Pre-Processing

The raw PCD acquired directly from the 3D scanner cannot be used immediately for inspection, due to imperfections in the data. The source of imperfection could be due to noise, or misalignment, and would result in error in the inspection process if not corrected. The raw PCD should thus be pre-processed to remove noise or alignment errors. The raw PCD was processed by the noise removal algorithm first, as the presence of noise would affect the registration algorithm, leading to inaccurate results.

5.2.3.1 Noise Removal

There is a variety of source of noise in the PCD, as elaborated in Section 2.3.4. Whilst a properly selected 3D scanner would minimize these noises for the specific application, the most appropriate 3D scanner might still be subjected to noise, and it should be removed by means of post-processing.

As reviewed in Section 2.3.4, noise removal algorithms are well established, and it is a function commonly available within reverse engineering software. The provided functionality with Geomagic Design X, termed as “Filter Noise”, is used for this research, to remove noise and outliers from the data. Geomagic Design X was chosen over other available software, as the manufacturer of the scanner used for this research have tested the software with their hardware, and

recommended it. The recommended Max. Poly-Vertex Count Per Noisy Cluster setting is used, with acceptable results, as shown in Figure 2.29.

5.2.3.2 Registration

The raw PCD also contains multiple scans of the object from different viewpoints, due to a 3D scanner's limited field of view. If the different scans are improperly aligned to each other, the combined PCD would not be an accurate representation of the scanned part.

Similarly, as reviewed in Section 2.3.4, registration algorithms are well established and commonly available within reverse engineering software, and the included functionality within Geomagic Design X, termed as "Align Between Scan Data", is used for this research, for the same reasons as the use the outlier removal algorithm. The "Global and Fine" option was used, with the "max average deviation" set at 0.012mm.

5.2.4 Synthetic Point Cloud Data

Although it is ideal to conduct all the experiments and training of model on real scan data, large amounts of scanned data are required for the purpose of training a machine learning model. Unfortunately, there is a limitation on the quantity of physically damaged parts available for this research, and it is also excessively time consuming and resource intensive to create large amounts of damaged parts to be scanned. As such, PCD was synthesized for the purpose of demonstrating the algorithms and training the machine learning model. The steps for synthesizing the PCD in explained in the following sections.

5.2.4.1 Computer Aided Design Modelling

For the purpose of training our model, the synthetic data models were selected to balance between the requirement of including all the common geometries, as well as is of industrial interest for the actual part to be remanufactured. The selected model was modelled in the Computer Aided Design (CAD) Modelling software, SolidWorks, and are as shown in Figure 5.7(a-e).

Figure 5.7: CAD models of: (a)I-beam; (b)Concentric shaft; (c)Gear rack; (d)Pinion gear; (e)Turbine blade

I-beam section

The I-beam section (see Figure 5.8(a)) was selected as it is a common engineering component. It contains simple geometries, consisting of mainly planar surfaces, with some internal radius.

Concentric shaft

The concentric shaft (see Figure 5.8(b)) was selected as it is another common engineering component and is frequently repaired and remanufactured manually. It contains only concentric cylinders, capped with planar surfaces.

Gear rack

The gear rack (see Figure 5.8(c)) was selected as it is a component of interest to be remanufactured by the oil and gas industry and is used in the jack-up rigs. It contains mainly planar surfaces, with surfaces which are not parallel to the cartesian plane. For this model, a smaller scale rack was chosen, and is based on the KHK US-SR8-130.

Pinion gear

The pinion gear (see Figure 5.8(d)) was also selected as a complimentary part to the gear rack for the same reasons. It contains an array of curved surfaces with an involute profile. The selected gear to compliment the gear rack is based on the KHK US-SS8-12.

Turbine blade

The turbine blade (see Figure 5.8(e)) represents a high value part that the industry would like to remanufacture. From a geometric perspective, the freeform surface of the blade would be different from that of the previous models, which contains only planar and cylindrical surfaces. The selected blade model is based on the General Electric LM2500 gas turbine blade.

5.2.4.2 Damaging the CAD Model

To represent a damaged part, the damage is modelled directly onto the CAD model, based on the characteristics of the 5 different types of damage as described in section 5.1.4. Example of each of the 5 types of damage is as shown in Figure 5.8.

Figure 5.8: Modelled damage: (a)Surface wear; (b)Pit or corrosion; (c)Broken damage; (d)Crack; (e)Plastic deformation.

5.2.4.3 Sampling CAD Model

The CAD model was converted to point cloud by means of sampling of the CAD model surface in Geomagic Design X. The point cloud sampling resolution was set to $120\mu\text{m}$, the same as the standard resolution setting of the FARO Edge (3.7m) ScanArm HD. Additionally, the point cloud was resampled in Geomagic Design X, to remove the grid pattern of the first sampled points from the CAD surface, which is not present in actual scanned data,. The same resolution of $120\mu\text{m}$ is maintained.

5.2.4.4 Noise Generation

To ensure that the synthetic data is representative of the real-world data, white noise is added to the synthetic data. The FARO Edge (3.7m) ScanArm HD, which is the scanner chosen to subsequently scan actual damaged parts for testing of the proposed model, has an accuracy with a standard deviation of $116\mu\text{m}$. As such, the white noise was generated in the PCD using Matlab, by moving the points in the direction of the normal or opposite to the normal, with the Euclidean distance between the originally sampled points to the error-induced points having a standard deviation of $116\mu\text{m}$.

With the above 4 steps, some PCD was synthesized, with the details as shown in Table 5.1 and a summary of the breakdown of the number of points in Table 5.2.

Table 5.1: Synthetic PCD details

No.	Model	Damage Type	Total Number of Points	Number of Damage point
1	I-beam section	Wear	4,293,203	88,769
2	I-beam section	Pitting	4,282,423	116,400
3	I-beam section	Broken	4,282,291	96,583
4	I-beam section	Crack	4,289,739	62,482
5	I-beam section	Deformation	4,281,191	113,077
6	Concentric shaft	Wear	694,022	13,652
7	Concentric shaft	Pitting	688,316	9,366
8	Concentric shaft	Broken	687,754	14,452
9	Concentric shaft	Crack	695,114	3,996
10	Concentric shaft	Deformation	683,476	21,913
11	Gear rack	Wear	3,330,655	69,226
12	Gear rack	Pitting	3,325,342	42,272
13	Gear rack	Broken	3,329,712	63,146
14	Gear rack	Crack	3,325,110	33,866
15	Gear rack	Deformation	3,326,539	113,026
16	Pinion gear	Wear	5,261,548	12,662
17	Pinion gear	Pitting	5,270,246	97,214
18	Pinion gear	Broken	5,271,966	81,719
19	Pinion gear	Crack	5,263,821	3,633
20	Pinion gear	Deformation	5,254,014	153,259
21	Turbine blades	Wear	2,570,584	62,323
22	Turbine blades	Pitting	2,568,623	27,361
23	Turbine blades	Broken	2,573,396	21,466
24	Turbine blades	Crack	2,574,319	35,706
25	Turbine blades	Deformation	2,569,737	56,171

Table 5.2: Summary of synthetic PCD

Total Number of Points	80,693,141
Points in Undamaged Region	79,279,400
Percentage of Points in Unamaged Region	98.25%
Points in Damaged Region	1,413,741
Percentage of Points in Damaged Region	1.75%
Points in Wear Region	246,632
Points in Pitting Region	292,613
Points in Broken Region	277,366
Points in Cracked Region	139,684
Points in Deformed Region	457,446

5.2.5 Point Cloud Annotation

To validate the results of this research, the results on whether a point is a damage or not should be compared to ground truths. As such, both the synthetic and real cloud data were manually labelled to serve as ground truth. This was done by manually selecting the points which are identified by a professional to be part of the damage, and the results of the detected damage can be compared to that of the annotated ground truth. Additionally, the damage type was also annotated based on one of the 5 damages as shown in Section 5.1.4.2. With the PCD annotated to serve as our ground truth, the results of this research works can be evaluated.

5.3 Machine Learning Models

This dataset will subsequently be used to train and compare various machine learning models, to be used for damage detection and classification. The Classification Learner App in the Statistics and Machine Learning Toolbox within Matlab is used to train the Machine Learning model, using a 5-fold cross validation. A summary of the available machine learning model and their characteristics is as shown in Table 5.3

5.3.1 Decision Trees

Decision tree learning is a predictive learning model commonly used in data mining[216]. A decision tree, or a simple representation for classification, is built by splitting the input data into subsets based on the classification features. Recursive partitioning continues by iterating this process on each derivative subset, until the target variable is reached by a subset node, or when the division no longer augments the predictions.

The decision tree model within Matlab assessed in this thesis are the coarse tree, medium tree, and fine tree, where the maximum number of subsets for each division is 4, 20, and 100 respectively.

Table 5.3: Summary of machine learning models, adapted from [215]

Classifier Type	Machine Learning Model	Prediction Speed	Memory Usage	Interpretability
Decision Tree	Fine Tree	Fast	Small	Easy
Decision Tree	Medium Tree	Fast	Small	Easy
Decision Tree	Coarse Tree	Fast	Small	Easy
Discriminant Analysis	Linear Discriminant	Fast	Small	Easy
Support Vector Machines	Linear SVM	Medium	Medium	Easy
Support Vector Machines	Quadratic SVM	Slow	Large	Hard
Support Vector Machines	Cubic SVM	Slow	Large	Hard
Support Vector Machines	Fine Gaussian SVM	Slow	Large	Hard
Support Vector Machines	Medium Gaussian SVM	Slow	Large	Hard
Support Vector Machines	Coarse Gaussian SVM	Slow	Large	Hard
Nearest Neighbour Classifiers	k-Nearest Neighbour	Slow	Medium	Hard
Naïve Bayes Classifiers	Kernel Naïve Bayes	Slow	Medium	Easy
Ensembles (Combinations)	Boosted Trees	Fast	Low	Hard
Ensembles (Combinations)	Bagged Trees	Medium	High	Hard
Ensembles (Combinations)	Subspace Discriminant	Medium	Low	Hard
Ensembles (Combinations)	RUSBoost Trees	Fast	Low	Hard
Neural Network	Shallow Network	Slow	High	Hard

5.3.2 Discriminant Analysis

The discriminant analysis approach builds a discriminant function based on a combination of features that characterizes the input data. The linear discriminant analysis used for this research defines the distance of a data point from the centre of a category, and creates a new set of axes to group members of the same category as close together as possible, and move the groups as far apart from one

another as possible. These new axes are the discriminant axes and are linear combinations of the original input variables.

5.3.3 Support Vector Machines (SVM)

SVM works by building a hyperplane or set of hyperplanes in a high-dimensional space. The hyperplanes furthest away from the nearest training-data point is chosen for an optimal separation, as there would be a lower generalization error of the approach for a larger separation. As the input data is typically not linearly separable as is, the original finite-dimensional space is mapped into a much higher-dimensional space for more better separation.

The SVM models within Matlab assessed in this thesis are the linear SVM, quadratic SVM, cubic SVM, fine gaussian SVM, medium gaussian SVM, and the coarse gaussian SVM.

5.3.4 Nearest Neighbour Classifiers

The k-nearest neighbours algorithm (k-NN) is a non-parametric method where an input data point is classified by the majority of its neighbours, with the input data point being assigned to the classification most common among its k nearest neighbours. Due to the large size of our data of 80,693,141 points with 11 features each, the nearest neighbour classifier model would take excessively long to train and predict, and hence is omitted from this study despite being a highly accurate model theoretically.

5.3.5 Naïve Bayes Classifiers

The Naïve Bayes classifiers is based on the application of the Bayes' theorem, which describes the probability of an occurrence based on prior knowledge of circumstances that might be associated to the outcome. It is a conditional probability model that assumes that predictors are independent. The classifier may work well even if the assumption is invalid.

The Naïve Bayes classifiers used within Matlab is the kernel Naïve Bayes, where higher accuracy levels are achieved by utilising a kernel density estimation, where the probability density function of a random variable is approximated based on inferences about the population.

5.3.6 Ensembles (Combinations)

Ensembles are learners that combines the results from many weak learners to form a high-quality ensemble model.

Boosted Trees

Boosted trees refer to the combination of adaptive boosting techniques with decision tree learners. The adaptive boosting techniques uses multiple iterations where a new weak learner is added to the ensemble, and a weighting vector is adjusted to focus on examples that were misclassified in previous rounds.

Bagged Trees

Bagged trees refer to the combination of random forest bag techniques with decision tree learners. The bagged, or bootstrap-aggregated decision trees combine the results of many decision trees, which reduces the effects of overfitting and improves generalization over individual decision trees. This is done by growing the decision trees using bootstrap samples of the data.

Subspace Discriminant

Subspace discriminant refers to the combination of subspace with discriminant analysis, where the random subspace method attempts to decrease the association between estimators in an ensemble by conducting the discriminant analysis on random samples of features instead of the entire feature set.

RUSBoost Trees

RUSBoost Trees refers to the combination of Random undersampling boosting (RUSBoost) with decision tree learners. The algorithm takes N , the number of members in the class with the fewest number in the training data, as the basic unit for sampling. Classes with more members are under sampled by taking only a subset of the data with N observations from each of the K classes, following which the decision tree classification approach is applied.

5.3.7 Neural Network

This data was also used to train a Neural Network in the Neural Network Pattern Recognition/Classification app within Matlab. Unlike the other machine learning classification models, neural networks are composed of simple elements

operating in parallel, where the elements are inspired by biological nervous systems. The neural network automatically generates identifying characteristics or hidden layers based on the provided examples. It then computes the difference between the processed output of the network and a target output, and alters its weighted associations according to a learning rule and this error value, thus improving the model.

The neural network in the Neural Network Pattern Recognition/Classification app within Matlab is a shallow neural network, which is a two-layer feed-forward network, with a sigmoid transfer function in the hidden layer, and a softmax output neurons, or a linear transfer function in the output layer. The network is trained with the Levenberg-Marquardt backpropagation algorithm [217, 218].

5.4 Damage Detection via Supervised Learning

The feature extraction is applied to all the synthetic PCD, with each point being assigned with 11 bits of information for each feature, where 1 represents the presence of the feature, and 0 represents the absence of the feature. These features are used as predictors in the machine learning model. Additionally, there is an extra 2 bit of data based on the label given to each point during the annotation stage as presented in Section 5.2.5, on whether the point is part of the damage as identified by professional or not, as well as the damage type. These will be used as the response for the models. For this section, only the annotation on whether the point is a damage or not is used for damage detection.

5.4.1 Classifier Comparison

The detailed results of the trained machine learning models are as shown in Table 5.4.

Speed

The decision tree and logistic regression classifiers are the fastest based on the results, with low training times and high prediction speed. This can be attributed to the simplicity of the model. However, this is done at the expense of the accuracy.

Table 5.4: Comparison of machine learning models for damage detection

No	Machine Learning Model	Prediction Speed (obs/sec)	Training Time (s)	True Positive (TP)	True Negative (TN)	False Positive (FP)	False Negative (FN)	Accuracy	Sensitivity
1	Fine Tree	1,200,000	3628.24	1,148,665	67,682,584	11,596,816	265,076	85.3%	81.3%
2	Medium Tree	860,000	5133.51	1,148,806	67,682,443	11,596,957	264,935	85.3%	81.3%
3	Coarse Tree	1,200,000	4226.64	1,150,361	67,035,343	12,244,057	263,380	84.5%	81.4%
4	Linear Discriminant	510,000	6362.10	1,067,374	62,599,514	16,679,886	346,367	78.9%	75.5%
5	Linear SVM	810,000	8061.31	1,355,919	78,610,984	668,416	57,822	99.1%	95.9%
6	Quadratic SVM	410,000	8542.54	1,345,316	78,218,121	1,061,279	68,425	98.6%	95.2%
7	Cubic SVM	780,000	9077.17	1,339,237	77,740,041	1,539,359	74,504	98.0%	94.7%
8	Fine Gaussian SVM	230,000	17835	1,309,690	76,720,577	2,558,823	104,051	96.7%	92.6%
9	Medium Gaussian SVM	290,000	16359	1,332,592	78,069,459	1,209,941	81,149	98.4%	94.3%
10	Coarse Gaussian SVM	490,000	13323	1,315,769	76,875,885	2,403,515	97,972	96.9%	93.1%
11	Boosted Trees	200,000	11026	1,229,531	72,281,920	6,997,480	184,210	91.1%	87.0%
12	Bagged Trees	170,000	35848	1,252,150	73,389,005	5,890,395	161,591	92.5%	88.6%
13	Subspace Discriminant	120,000	9425.45	1,139,899	67,368,578	11,910,822	273,842	84.9%	80.6%
14	RUSBoosted Trees	240,000	5033.06	1,179,767	68,942,573	10,336,827	233,974	86.9%	83.5%
15	Kernel Naive Bayes	150,000	1056.82	1,229,672	71,878,314	7,401,086	184,069	90.6%	87.0%
16	Neural Network	-	3051	1,348,992	78,537,218	742,182	64,749	99.0%	95.4%

Accuracy

The accuracy represents the ratio of the number of points that has been accurately classified as damaged or undamaged, relative to the total number of points in the model. Mathematically, it is given by equation 5.1 below:

$$Accuracy = \frac{TP+TN}{Total} \quad (5.1)$$

where TP is the True Positive, or number of points correctly identified to be part of the damage region, and TN is the True Negative, or the number of points correctly identified points that do not belong to a damage region.

The linear SVM classifier achieved the highest accuracy of 99.1%, closely followed by the neural network model with an accuracy of 99%. However, as the size of the damage region is significantly smaller than that of the part, a large proportion of the data, amounting to 98.25%, are of the points that are from the undamaged region. Thus, the results of the detection of points in the undamaged region heavily influences the overall accuracy result. This makes the accuracy parameter not the most significant value for our application towards automating damage detection for remanufacturing, as it is more important to detect more of the damage, instead of accurately detecting a point in an undamaged region.

Sensitivity

The sensitivity represents the ratio of correctly detected damage points to the total number of points that are part of the actual damage, and is represented mathematically by the equation 5.2 below:

$$Sensitivity = \frac{TP}{TP+FN} = \frac{Correctly\ Detected\ Damage\ Points}{Actual\ Damage\ Points} \quad (5.2)$$

where TP is the True Positive, or number of points correctly identified to be part of the damage region, and FN is the False Negative, or the number of points that are part of a damage region but were not identified.

For our specific application towards automating damage detection for remanufacturing, it is important that any damage within the model is detected. Hence, sensitivity is a more significant result than accuracy. The linear SVM classifier outperforms the other classifiers, with a sensitivity of 95.9%. As such,

the linear SVM classifier model was used as our selected model for damage detection.

5.4.2 Visual Results

The results presented above on the accuracy and sensitivity of the model for damage detection assumes each point in the point cloud as an individual data point, with a set of features tagged to the point as calculated from section 4.4. However, a damage typically refers to a damaged region, which is made up of a group of points. As such, the results from the linear SVM model were reviewed visually, where the points were plot out in Matlab, with the TP, TN, FP and FN points plot with different colours. An example of the result is as shown in Figure 5.9.

Figure 5.9: (a) Synthetic CAD model, (b) Damage detection results

Upon visual inspection of the 25 models, majority of the points within each of the 25 damage regions were identified to belong to a damage. Additionally, some undamaged regions surrounding a damage were incorrectly identified as points of a damage, as seen in Figure 5.8. As such, for our training data, the results presented in the previous section is also representative of how accurate the machine learning model is in identifying the boundaries of each damage, on top of identifying the damage region itself.

5.4.3 Real Application and Results

The scanned data of some physically damaged parts were used to validate the proposed method, and was input into the trained SVM model. The results are as shown in Figures 5.10 to 5.12.

Figure 5.10: (a) & (c) Damaged rack, (b) & (d) Damage detection result

Figure 5.11: (a) Titanium turbine jet compressor engine blade, (b) Damage on blade, (c) Damage detection result

Figure 5.12: (a) High nickel aircraft jet turbine engine blade, (b) Damage on blade, (c) Damage detection result

From the above figures, it can be seen that all the damages were able to be detected, although there were discrepancies at the boundaries of the detected damage with respect to the boundaries drawn by a professional. However, this is not a major issue, as the actual boundary of each damage or defect is also ambiguous between professionals, and the main objective of detecting majority of the area as a damage has been achieved.

In addition to detection actual damages, the proposed approach has also identified intended features that have a similar characteristic to that of a damage. For example, the text deboss on the rack used for labelling the brand and model number was detected as a damage, as shown in Figure 5.13. The results should thus be used as an aid to detect damage, with a human making the repair decision of the detected damage.

Figure 5.13: (a) Deboss model number, (b) Wrongly detected damage (Red)

5.4.4 Comparison to State-of-the-art

The main objective of this thesis is for a robust damage detection using only the 3D PCD. The proposed method has overcome two of the three limitations of damage detection methods proposed by prior literature, in that there is no assumption on the undamaged part geometry made, and there is no need for additional inputs. To validate that the last limitation of the has been addressed in the proposed method, in terms of the scope of detected damage types of the methods proposed in prior literature, the accuracy and the sensitivity of the proposed method is compared to other methods proposed by prior literature. The results in terms of the accuracy and sensitivity are presented in Table 5.5.

Table 5.5: Comparison between state-of-the-art and the proposed method

Methods	Accuracy	Sensitivity
Teza, Galgaro [133]	88.5%	87.5%
Li, Yao [77]	91.0%	89.6%
Li, Li [25]	90.4%	88.2%
Ghahremani, Khaloo [138]	86.5%	87.3%
The Proposed Method	99.0%	95.5%

From the results, both the accuracy and sensitivity has seen an improvement. Upon further analysis of each damage model, it is observed that some damages could not be identified by some of the state-of-the-art methods, leading to the poorer performance.

5.5 Damage Classification via Supervised Learning

The next step of this research looks at the classification of damage types, as an input to the subsequent step in the inspection process, which is the repair decision.

5.5.1 Classifying from Entire Dataset

The first approach to classify damage types is to train a classifier model with the entire dataset, whereby the response classes were either of the 5 damage types, or if it was not a damage. The same synthetic PCD was used as a training data for damage classification, except that the response used was the damage type instead of simply whether it was a damage. Similarly, the dataset is input into the Classification Learner App in the Statistics and Machine Learning Toolbox within Matlab to train a Machine Learning model, using a 5-fold cross validation,

as well as a Neural Network in the Neural Network Pattern Recognition/Classification app within Matlab.

To validate if this approach is still as good as the previous approach in detecting points within the damaged region, the damage detection accuracy and damage detection sensitivity were calculated from the Matlab results, given by the equations 5.3 and 5.4 below:

$$\text{Damage Detection Accuracy} = \frac{\text{Points correctly identified as nondamage} + \text{Points classified as any damage types}}{\text{Total number of points}} \quad (5.3)$$

$$\text{Damage Detection Sensitivity} = \frac{\text{Damage points classified as any damage types}}{\text{Total number of damage points}} \quad (5.4)$$

These values were calculated to disregard the damage type classification accuracy, and solely focus on the accuracy and sensitivity of the model itself. Additionally, to access the accuracy of classification, the following equations 5.5 and 5.6 were used:

$$\text{Damage Classification Accuracy (wrt all damage)} = \frac{\text{Sum of all points correctly classified damage}}{\text{Total number of damage points}} \quad (5.5)$$

$$\text{Damage Classification Accuracy (wrt detected damage)} = \frac{\text{Sum of all points correctly classified damage}}{\text{Total number of points classified as any damage}} \quad (5.6)$$

The detailed results are as shown in Table 5.6.

From the results, the best classifier is the Neural Network, both for damage detection, as well as damage classification. However, when compared to the damage detection results from the previous section, there is a drop in the percentage accuracy as well as sensitivity. This reduction in damage detection accuracy and sensitivity can be attributed to the finer classifiers classifying the data into 6 responses, instead of the two responses from the previous section.

Table 5.6 : Comparison of machine learning models for damage classification from entire dataset

No	Model Type	Speed obs/sec	Training Time (s)	Damage Detection Accuracy	Damage Detection Sensitivity	Damage Classification Accuracy (wrt all damage)	Damage Classification Accuracy (wrt detected damage)
1	Fine Tree	500,000	5842.94	79.8%	73.1%	53.4%	58.6%
2	Medium Tree	770,000	6443.53	81.3%	76.3%	54.9%	60.4%
3	Coarse Tree	1,600,000	6021.47	79.9%	75.3%	54.3%	60.9%
4	Linear Discriminant	350,000	4289.02	75.9%	71.0%	55.4%	59.2%
5	Linear SVM	250,000	8055.08	92.8%	90.2%	65.8%	70.8%
6	Quadratic SVM	88,000	48474	93.6%	89.1%	66.7%	72.2%
7	Cubic SVM	110,000	1658507	95.8%	89.5%	67.0%	74.0%
8	Fine Gaussian SVM	3,900	115694	94.2%	86.1%	64.8%	72.3%
9	Medium Gaussian SVM	3,800	118131	94.9%	90.3%	63.9%	68.6%
10	Coarse Gaussian SVM	3,900	147615	91.0%	87.5%	65.1%	69.3%
11	Boosted Trees	77,000	35107	86.5%	83.3%	62.7%	66.8%
12	Bagged Trees	71,000	41256	89.6%	82.3%	61.1%	66.0%
13	Subspace Discriminant	24,000	35184	83.2%	78.1%	54.0%	60.2%
14	RUSBoosted Trees	77,000	49029	80.4%	76.7%	58.5%	63.7%
15	Kernel Naive Bayes	3,500	1861085	88.7%	78.7%	57.8%	64.5%
16	Neural Network	-	2328	97.0%	91.4%	69.7%	76.2%

5.5.2 Classifying Detected Damage

To prevent the loss in accuracy and sensitivity of detecting damage, a two-step approach to classification was experimented, where the detected damage points from Section 5.2 are extracted and used to train a classifier model. The comparison between the different classifier models are presented in Table 5.7, where the Neural Network model has the highest damage classification accuracy of 76.3%. This is comparable to the Neural Network model trained on the entire dataset, which has a damage classification accuracy with respect to detected damage, of 76.2%. The accuracy of classification for each damage type with respect to the detected damages are similar as well. However, when comparing between the results of the direct classification approach against the two-step approach, a larger number of points are correctly classified to their damage type using the two-step approach, as seen from the confusion matrix as shown in Table 5.8 and 5.9. Thus, it is preferable to adopt the two-step approach separating damage detection and classification for an increased accuracy in both damage detection as well as damage classification.

Table 5.7: Comparison of machine learning models for damage classification from detected damage

No.	Model Type	Speed obs/sec	Training Time(s)	Damage Classification Accuracy (wrt all damage)	Damage Classification Accuracy (wrt detected damage)
1	Fine Tree	1,300,000	66.86	56.2%	58.7%
2	Medium Tree	1,300,000	62.99	58.0%	60.5%
3	Coarse Tree	1,400,000	59.46	59.7%	61.0%
4	Linear Discriminant	610,000	48.33	57.6%	59.3%
5	Linear SVM	310,000	6131.78	65.9%	70.8%
6	Quadratic SVM	110,000	326034	69.3%	72.2%
7	Cubic SVM	140,000	14169377	72.8%	74.2%
8	Fine Gaussian SVM	4,900	862199	70.5%	72.1%
9	Medium Gaussian SVM	4,800	975152	66.6%	68.4%
10	Coarse Gaussian SVM	5,000	963829	65.7%	69.5%
11	Boosted Trees	100,000	5581.13	65.5%	66.8%
12	Bagged Trees	91,000	5594.09	62.1%	65.9%
13	Subspace Discriminant	29,000	5554.88	59.6%	60.3%
14	RUSBoosted Trees	100,000	5576.62	61.3%	63.6%
15	Kernel Naive Bayes	4,500	12683529	60.9%	64.4%
16	Neural Network	-	235	73.2%	76.3%

Table 5.8: Confusion Matrix of Neural Network for classification on entire dataset, with damage type classification accuracy wrt detected damage

		Non-Damage	78,610,984*	120113	129152	135211	66307	217633
		Surface Wear	57,822*	163021	33681	4762	12264	23718
		Pitting/ Corrosion		68.66%	14.18%	2.01%	5.16%	9.99%
		Broken Damage		41191	201089	10584	11685	20648
				14.44%	70.51%	3.71%	4.10%	7.24%
		Crack		1598	3132	195851	17866	44364
				0.61%	1.19%	74.52%	6.80%	16.88%
		Plastic Deformation		5060	6850	3846	96434	22089
				3.77%	5.10%	2.86%	71.82%	16.45%
				15384	20935	18407	2923	378534
				3.53%	4.80%	4.22%	0.67%	86.78%
		Non-Damage	Surface Wear	Pitting/ Corrosion	Broken Damage	Crack	Plastic Deformation	
		Predicted Class						

Table 5.9: Confusion Matrix of Neural Network for classification on detected damage points, with damage type classification accuracy wrt detected damage

True Class		No Damage	78,610,984*	121753	140905	127459	64917	213382
		Surface Wear	57,822*	160394	33585	4748	12334	23875
Pitting/ Corrosion	68.27%	14.30%		2.02%	5.25%	10.16%		
	Broken Damage	40996		197295	10305	11685	20608	
Crack	14.60%	70.24%		3.67%	4.16%	7.34%		
	1590	3123		194535	18038	45481		
	0.61%	1.19%		74.03%	6.86%	17.31%		
Plastic Deformation	5055	6791		3917	95066	22261		
	3.80%	5.10%		2.94%	71.43%	16.73%		
	15296	20816		18478	2994	386650		
	3.44%	4.69%		4.16%	0.67%	87.04%		
	No Damage	Surface Wear	Pitting/ Corrosion	Broken Damage	Crack	Plastic Deformation		
	Predicted Class							

*These results are obtained from the damage detection step using the Linear SVM model

5.5.3 Real Application and Results

The trained neural network was used to classify the damage from the real PCD, and a sample result is as shown in Figures 5.14 to 5.16.

Figure 5.14: (a) & (c) Damaged rack, (b) & (d) Damage classification results

Figure 5.15: (a) Titanium turbine jet compressor engine blade, (b) Damage on blade, (c) Damage classification result

As seen from the above figures, all the damages from our sample damage were largely identified accurately by the proposed method.

It is worth mentioning that there are instances where a damage region cannot be classified into a single damage type. This is because some damages are a result of varying causes, which results in the damage being a combination of varying damage types. For example, in Figure 5.16, the damage in the high nickel aircraft jet turbine blade is a combination of a deformation and a crack. The proposed damage classification approach will not be able to classify the damage as a combination of 2 or more damage types. However, the results will still be useful in providing feedback on the repair decision for a damaged part.

Figure 5.16: (a) High nickel aircraft jet turbine engine blade, (b) Damage on blade, (c) Damage classification result

5.6 Further Experiments

The proposed approach has been shown to have better performance than existing state of the art methods available in terms of damage detection, as well as the capability to classify damage. For the experimentation conducted in the previous sections, the synthetic data created to train the machine learning models and the scanned data of physically damaged parts used for the validation of the proposed

approach were of similar damage dimensions and similar scan resolutions. This is done to simulate the ideal real-world application scenario, where the same scanner, which will output data of the same resolution, is to be used to scan damaged parts from the same industry, which will be of similar object and damage dimensions. Such a setup is preferred, as the machine learning model will have training data representative of the damages to be automatically detected and classified downstream. However, there may be instances whereby it is difficult to attain such a match in terms of object and damage dimensions, as well as scan resolution between the training data and actual application. As such, this section documents the further experiments done to test the robustness of the approach to variations in the application, in terms of damaged size and scan resolution.

To ensure that a common baseline is available for comparison, as well as due to the lack of access to large physically damaged parts or higher resolution scanners, the experimentation was conducted on data that was adapted to simulate the variations in the damage and part size, and scan resolution. The scan data from the damaged rack, as shown in Figure 5.10(a), was used.

To simulate the variation in the scan resolution, the scanned PCD was resampled within Geomagic Design X, using the uniform ratio option. The PCD was either down sampled with a sampling ratio of 0.25 and 0.5 to simulate a lower resolution scan, or up sampled with a sampling ratio of 2 and 4 to simulate a higher resolution scan.

To simulate the variation in the dimension of the damaged part, the scanned PCD was scaled by multiplying the coordinated by a factor of 0.25, 0.5, 2, and 4, simulating a part size which is quartered, halved, doubled, and quadrupled respectively. However, as this procedure will lead to the spatial resolution, or point cloud density, of the PCD to change. As such, the scaled PCD is resampled with a sampling ratio equivalent to the dimensional scale factor, to maintain the resolution regardless of dimensional scale.

The variations in dimension and scan resolution were also combined and experimented with. The linear SVM model trained with the unmodified training data from the section 5.1.4 was selected for the damage detection step, while the

neural network was used to classify the detected damages. The results are as shown in Table 5.10.

Table 5.10: Damage detection and classification sensitivity wrt variations

			PCD coordinate scale factor (simulating scanned part size scale)					Sensitivity
			←Smaller size←		Original	→Larger size→		
			0.25	0.5	1	2	4	
Sampling ratio (simulating scan resolution ratio)	→Lower resolution→	0.25	6.1%	12.2%	16.7%	14.5%	14.4%	Detection
			4.9%	9.7%	13.2%	11.6%	11.5%	Classification
		0.5	26.1%	67.7%	83.7%	76.6%	68.3%	Detection
			20.7%	54.2%	67.1%	61.2%	54.6%	Classification
	Original	1	62.5%	90.7%	95.7%	94.5%	93.3%	Detection
			49.8%	71.5%	76.1%	75.5%	75.0%	Classification
	←Higher resolution←	2	52.1%	71.8%	92.4%	87.5%	84.6%	Detection
			41.4%	56.9%	73.6%	69.0%	67.0%	Classification
		4	40.3%	61.0%	88.5%	84.5%	82.7%	Detection
			32.3%	48.2%	69.7%	67.8%	65.5%	Classification

Damage Size Variation

From Table 5.10, any variation in damage size will lead to a decrease in sensitivity of the proposed approach using the same trained model. This is expected, as some of the dimensionality extraction feature extraction processes utilizes algorithms that are dependent on the scale, like the bounding sphere nearest neighbour radius. As such, some features might not be accurately extracted, leading to a lower sensitivity in damage detected. Additionally, the machine learning model was trained using damaged parts with similar sized damage. As such, variations in damage size would reduce the sensitivity of the proposed approach, although the proposed approach still performs decently for parts and damage size no smaller than half, or larger than the trained parts. This is due to the smaller part size with smaller damages leading to some damage

region too small to be captured by the proposed method, hence the sensitivity of a reduced part is poorer as compared to an enlarged part.

Scan Resolution Variation

Similarly, the variations in the scan resolution leads to the sensitivity of the proposed approach, using the same trained model. This is also expected, as some of the dimensionality extraction feature extraction processes utilizes algorithms that are dependent on the spatial resolution, like the k nearest neighbour. As such, some features might not be accurately extracted, leading to a lower sensitivity in damage detected. Additionally, the machine learning model was trained using parts with similar spatial resolution, and the machine learning model would not have taken account for the variations in resolution, hence a drop in sensitivity. It is also noted that the proposed approach fails catastrophically when the resolution is quartered, and it is likely due to the lack of available data points over the damaged region. Additionally, small damages like surface scratches or corrosion will not be captured due to the low resolution.

Hence, it is most ideal to use the proposed approach with training data similarly sized and with a similar scan resolution as the part where the damage detection and classification is to be applied. Nonetheless, decent results are still obtainable for parts that are enlarged or shrunk no less than half of the training data, and for scan resolution no less than halved than the training data.

5.7 Limitations

Despite the higher accuracies and sensitivity rate than the state of the art, the proposed methodology is sensitive to noise, similar to that of prior literature. This is unavoidable as the irregularities of noise is similar to that of the irregularities of damage. As such, it is essential to have good data acquisition and pre-processing practices to minimize the noise present in the data, inclusive of selecting an appropriate 3D scanner, following the scanning best practices recommended by the manufacturer, as well as cleaning up the data by means of noise removal techniques.

Additionally, the proposed approach has not been computationally optimized, and the computation time increases with the amount of PCD data, as defined by

the scan resolution and the damaged part size. The proposed approach is thus unable to be used as-is in an instantaneous real-time application.

5.8 Summary

In this chapter, the creation, collection, and annotation of data necessary for this research is documented. Next, the actual framework for a robust damage detection and classification is implemented, using the extracted features from the dimensionality reduction of the 3D data in Chapter 4. With the extracted features, various machine learning models are to be trained, of which the different models are described. The model with the best accuracy and sensitivity result was the SVM model for the damage detection process. A direct approach for damage classification via machine learning on the extracted features was also tested, of which the results revealed a drop in terms of the accuracy of the damage detection. As such, a two-step approach was proposed, whereby the damage will be detected by the SVM model, following which the detected damages are classified using a Neural Network, and the results show an overall increase in accuracy of damage detection as well as classification results. The results were validated with some real scanned physically damaged parts, and the results as compared to the state-of-the-art ascertain the effectiveness of the proposed approach.

CHAPTER 6

Conclusion

This thesis opens with an introduction to traditional remanufacturing, which is practiced for the societal benefits from the conservation of resources and reduced waste, as well as economic benefits in terms of reduced costs. However, traditional remanufacturing is labour intensive, and faces problems of increasing labour costs, labour shortage, and varying quality of work between individuals. As such, automated remanufacturing is suggested as a potential solution to the labour problems, where knowledge from different persons are in-built into an artificial intelligence expert system.

The marine and offshore industry is an industry that can potentially practice more remanufacturing of their large components, and gain from the economical and societal benefits of remanufacturing. If the large damaged parts from the marine and offshore industry are remanufactured, like the rack or pinion of a jack up rig, it would be less costly and more resource efficient than replacing the damaged part altogether. However, to disassemble and transport large parts from such structures to an off-site remanufacturing factory is time-consuming and expensive. This is unlike traditional remanufacturing, where the damaged parts are small and economical enough to be transported to an off-site remanufacturing factory, and there is also a sufficient number of damaged parts to benefit from consolidating the remanufacturing process in a single location. Additionally, it is labour intensive and potentially dangerous to disassemble and transport the large parts, while some parts are impractical to remove due to physical constraints, requiring excessive structural changes, like the crankshaft crankpin journal in a ship. As such, remanufacturing should be carried out in-situ in the marine and offshore industry, to harness the potential societal and economic benefits.

The recent advancements in additive manufacturing technologies, like the Directed Energy Deposition technology, is also viewed as an enabling technology for automated in-situ remanufacturing. As such, this thesis investigates the current state-of-the-art for automated in-situ remanufacturing of large parts for

the marine and offshore industry, and addresses some of the limitations found for such remanufacturing.

6.1 Prior Literature

This research begins with a literature review of current remanufacturing practices and limitations. Due to the distinctiveness of automated in-situ remanufacturing, the traditional way of looking at remanufacturing does not encompass the full potential of automated in-situ remanufacturing, and there is no consistent consensus of what that might be in prior literature. As such, a new process flow is required to formalise this aspect of work, as it would not only improve consistency, improve quality, improve efficiency and reduce cost, but also increase accountability, tracking and traceability. The proposed automated in-situ remanufacturing framework will be based on the generic remanufacturing process flow proposed by Sundin [8], where the individual processes and subprocesses will be selected, ordered, and modified according to the requirements of automated in-situ remanufacturing.

With the automated in-situ remanufacturing framework, the state-of-the-art for the individual processes within the framework is reviewed. Amongst the reviewed processes, it is revealed that a critical limitation in technology for automated in-situ remanufacturing is the inspection process, more specifically the damage detection and classification sub-process.

Traditionally, the inspection process is conducted manually. However, to automate this process, the system should be able to acquire the data of the damaged part, followed by analysing it in terms of damage detection and classification. The literature review revealed that image-based damage detection methods are reliable, especially those which employ machine learning. However, in our application of automated in-situ remanufacturing, a 3D model of the damage region is required for the automated repair step downstream. It is thus resource efficient to utilise 3D scanning for the data acquisition sub-step for inspection, instead of other methods like imaging. Additionally, while some researchers have superimposed the image-based damage detection results on the 3D model, it is revealed that it is subjected to alignment errors. The use of 3D

data for inspection is also preferred by researchers in prior literature, as the data could potentially provide more information of the severity of damage, like depth.

A review of the use of 3D model for damage detection was conducted, of which it is revealed that the current state-of-the-art have limitations in terms of the scope of damage types detected, assumptions on part geometry, or require additional inputs. As such, the next part of this thesis focuses on proposing a robust framework for automated damage detection and classification from Point Cloud Data (PCD).

As with image-based damage detection and classification, the use of machine learning can be promising to increase the accuracy and robustness, based on prior literature. The last area reviewed was thus on the use of machine learning on geometric data, more specifically point cloud data. The issues with machine learning on 3D data are found to be due to unoptimized computational efficiency for geometric learning, as well as insufficient training data. However, due to the success of the use of machine learning in image-based damage detection, this research seeks to use machine learning indirectly on the 3D data, for damage detection.

In summary, the technological gaps from the critical review of the current state-of-the-art are identified, which also forms the focus of this thesis, and are as follows:

- Absence of a process flow for automated in-situ remanufacturing
- Limitations in PCD based damage detection algorithms
- Limitations in the direct application of machine learning on geometric data

6.2 Summary of Work Done

In Section 1.4, the following were listed down as objectives for this research work, to fill the gaps in the current state-of-the-art:

- To establish a process flow for automated in-situ remanufacturing.
- To explore and implement PCD based damage detection algorithms.
- To explore and acquire the data necessary for experimentation.
- To formulate and implement a framework for an accurate PCD based damage detection and classification.

The following sub-sections will assess whether the research work has achieved its intended objectives:

To establish a process flow for automated in-situ remanufacturing.

In Section 2.1.1, the related process flow is studied as a reference. The traditional way of looking at remanufacturing does not encompass the full potential of automated in-situ remanufacturing, and there is no consistent consensus of what that might be in existing literature. As such, a new process flow based on the generic remanufacturing process proposed by Sundin [8] is proposed in Chapter 3 to formalise this aspect of work. Each individual process was detailed in the chapter, on the importance of each process and location within the proposed process flow, as well as how it may be automated and carried out in-situ. The current state of the art for each process was also presented, together with any limitations in technology. The proposed automated in-situ remanufacturing framework would not only improve consistency, improve quality, improve efficiency and reduce cost, but also increase accountability, tracking and traceability.

To explore and implement PCD based damage detection algorithms.

From the proposed automated in-situ remanufacturing framework, it is revealed that there is a unique need to merely find local defect, and add or remove material based on the neighbourhood region. However, the literature review revealed that the state-of-the-art damage detection methodologies from 3D data have limitations in that they are unable to detect different damage types, have assumptions on the geometry of the undamaged part, or require additional inputs.

Although machine learning is a promising approach for damage detection from 3D data, it cannot be applied directly due to unoptimized computational efficiency for geometric learning. A method of dimensionality reduction is proposed, whereby the extracted features are based on damage characteristics. The prior literature on PCD based damage detection algorithms were reviewed in Section 2.3.5., of which the algorithms relevant in identifying our intended damage characteristics were implemented in Sections 4.4.1.4, 4.4.2.1, and 4.4.2.4. Additionally, eight other novel algorithms were proposed for the detection of other damage characteristics as identified by a professional, in Section 4.4, excluding three sub-sections mentioned previously. The implemented algorithms were reviewed to accurately extract the intended features.

To explore and acquire the data necessary for experimentation.

With dimensionality reduction as the solution to the issue of the unoptimized computational efficiency for geometric learning, the other problem with machine learning is the need for large amounts of data. The lack of available data for training and experimentation was addressed in Section 5.1 in two parts. The first approach of creating data is to scan physically damaged parts, and the issues pertaining to a poorly selected 3D scanner was addressed by the study documented in Section 2.3.3. Subsequently, the necessary pre-processing procedures necessary to clean up the data were presented in Section 5.1.3. However, due to the lack of large quantities of training data of damaged parts available for this research, synthetic data was created to be used as a training dataset, and is documented in Section 5.1.4. Subsequently, to validate our experiments, the acquired and synthetically created data were annotated manually to be used as our ground truth, as explained in Section 5.1.5.

To formulate and implement a framework for an accurate PCD based damage detection and classification.

The annotated training data was then used to train various machine learning model for both damage detection and classification, based on the extracted features for dimensionality reduction, and the results for the different models and approaches were compared. In Section 5.2, the Support Vector Machine (SVM) model returned the highest accuracy and sensitivity for damage detection. The

proposed approach can detect different types of damage from 3D scan data without assumptions on the undamaged part geometry or requiring additional inputs, with a sensitivity of 95.9%, representing a substantial improvement when compared to current 3D damage detection methodologies. The proposed approach was validated with real scanned data of physically damaged parts, and based on the visual results, damaged regions are largely identified, although the proposed approach has difficulties identifying the exact borders of the damage region. Section 5.3 extends the machine learning approach to damage type classification, of which the proposed two-step approach, separating the detection step and classifications step, was able to achieve better results in terms of accuracy in both damage detection as well as classification, attaining a damage type classification accuracy of 73.2%. However, it should be noted that the proposed approach may wrongly identify certain intended features as damage, such as debossed text, and is also sensitive to noise in the data. It is thus demonstrated that machine learning could be applied to geometric problems, with the proper dimensionality reduction.

6.3 Contributions

6.3.1 Academic Contributions

The following original academic contributions are claimed by this work:

- ***Classification framework for selection of appropriate 3D scanners.***

Unlike available methodology in prior literature for the selection of an appropriate 3D scanner, the proposed approach is not limited to a specific application, nor does it require the user to have prior access to the potential scanners to test the scanners. Additionally, the proposed approach is able to predict the practicality of potential scanners, in terms of the time taken and ease of use, which is unavailable in past researches. The proposed morphological box classification framework for classifying 3D scanners is also novel, to include different combinations of the numerous 3D scanning technologies, instead of the state-of-the-art branched tree classification approach. The proposed classification framework can thus be used as a template to search for the ideal scanner, based on the

requirement of the scan job, and matching it with the functionality of the scanner. Additionally, novel scanners can be made based on combinations that do not exist commercially previously.

- ***Framework for automated in-situ remanufacturing process.*** Although there are various frameworks for remanufacturing available in prior literature, none of them are specifically for automated in-situ remanufacturing, with the process for in-situ remanufacturing conducted in the marine and offshore industry predominantly based on the experience of the person performing the remanufacturing. As such, a framework for automated in-situ remanufacturing process is proposed, based on a generic remanufacturing approach by Sundin [8].
- ***Multiple novel PCD based algorithms for specific damage detection.*** The proposed damage detection and classification approach is based on the machine learning of extracted features, which are the damage characteristics. This approach allows for the use of algorithms to detect very specific damage characteristics, which may not be present in all damages. Such algorithms are not available in prior literature, and several of it have been developed as part of this research.
- ***Indirect application of machine learning on geometric data for damage detection.*** There has been no work done using machine learning on geometric data for damage detection in past literature. Additionally, it is documented by researchers that the unoptimized computational efficiency for geometric learning, and lack of training data hinders the direct application of machine learning on geometric data. As such, an indirect application of machine learning on geometric data for damage detection and classification via dimensionality reduction is proposed and presented, with no other similar 3D damage detection and classification approach in prior literature based on the author's knowledge. Although the scope of this thesis is for the indirect application of machine learning on geometric data for damage detection, the success of the proposed approach suggests the possibility of indirect application of machine learning on geometric data for other purposes, for example for reverse

engineering or object recognition for autonomous vehicles. All these can be made possible using traditional 2D machine learning architecture as the dimensionality reduction proposed in this thesis reduces the data to a 2D data, hence harnessing the years of development in machine learning.

6.3.2 Engineering Contributions

The main engineering contribution of this research is that this work will be beneficial in the adoption of automated in-situ remanufacturing by the industry. The formalism of the framework proposed for automated in-situ remanufacturing can be used as a reference to implement automated in-situ remanufacturing, instead of basing the remanufacturing on the experience of the person performing it. The proposed framework will also increase the accountability, tracking and traceability in the remanufacturing process.

This research also proposes a robust 3D damage detection and classification solution, which is an enabling technology in the inspection process of automated in-situ remanufacturing, and will be useful in the implementation of such remanufacturing. While the as-is proposed approach and trained machine learning model is well suited to parts of similar part size and scanned with the same resolution, the engineering value of this research lies in the extendibility of the proposed approach. To augment the features that have been extracted from the dimensionality reduction of the 3D data, other forms of data can be included in the model. For example, data on the material, or material properties, can be added as another feature to be trained and learnt by the machine learning model. As different materials would also result in slight differences in damage shape and size, having the material or material property is likely to improve the accuracy of the trained machine learning model. Examples of other data inputs can include the manufacturer, in-service duration, and part operational conditions, which can be used by the machine model to more accurately predict if the part has damage, or the types of the damage involved, based on the history of the part. With the increased implementation of automated in-situ remanufacturing, we can experience the societal benefits from the conservation of resources and reduced waste, as well as economic benefits in terms of reduced costs. The labour

problems of increasing labour costs, labour shortage, and varying quality of work between individuals would also be minimised, and there will be improved consistency, improved quality, improve efficiency, and increase accountability, tracking and traceability.

6.4 Further Works

The direction in which the future work takes could be divided into the following headings:

1. *Field implementation of the proposed in-situ remanufacturing framework*
2. *Field testing of the proposed automated damage detection approach*
3. *Training the model with more real scanned data of physically damage parts, together with non-geometric data*
4. *Using the results in the subsequent process of the automated remanufacturing process flow*

Field implementation of the proposed in-situ remanufacturing framework

To fully validate the effectiveness and accuracy of the proposed automated in-situ remanufacturing framework, the framework should be implemented in the field, where an actual large part in the marine and offshore industry is being remanufactured in-situ by an automated system. Unfortunately, such a project is of a large scale, requiring the coordination of multiple technologies, and is out of the scope of this thesis.

Field testing of the proposed automated damage detection approach

While the proposed automated damage detection approach was validated on real scan of physically damaged parts, the approach can be further validated on the detection of damage of large parts in the field, scanned on site. This was unfortunately not possible due to the lack of access to large physically damaged parts on site for such field validation.

Training the model with more real scanned data of physically damage parts, together with non-geometric data

Additionally, due to the lack of access to physically damaged parts, this research made use of synthetic data to train the machine learning model. While it was validated to have sufficient accuracy in both damage detection and classification on a real scan of a physically damaged part, it can be more accurate and reliable if real scanned data was used to train the model instead. This is because, should the machine learning model be trained with scanned data from damaged parts from a specific industry, the specific characteristics of the damage commonly found in the specific industry can be more influential in the trained model, resulting in an increase in accuracy. Additionally, with sufficient real scanned data, the features to be extracted as discussed in Section 4.4 can be reviewed, of which features with minimal influence on the model can be modified or removed totally, reducing the computational cost and time for the processing. Additionally, with the availability of real damaged parts, non-geometric data like the material or material property, manufacturer, in-service duration, or part operational conditions can be used to augment the machine learning model, and trends based on a certain property or history can be discovered to identify different types and extent of damages.

Using the results in the subsequent process of the automated remanufacturing process flow

This research addresses the issues within automating inspection, to work towards automating in-situ remanufacturing. The results from the damage detection and classification is thus to be utilised by subsequent processes in remanufacturing, and integrating the results could lead to the surfacing of other problems. Additionally, if the proposed method for damage detection and classification is to be integrated into subsequent steps like the repair decision and the nominal model reconstruction, the extracted features could provide valuable information in these processes, and thus is an area of further work.

6.5 Concluding Remarks

Automation and in-situ remanufacturing are two areas which remanufacturing can advance towards, to further harness the benefits of remanufacturing and overcome the current limitations of it. However, there is no prior literature specifically documenting the process flow of automated in-situ remanufacturing, and the process is predominantly based on the experience of the person performing the remanufacturing. In this research, an automated in-situ remanufacturing framework is proposed. The problems pertaining to the inspection step of remanufacturing was also investigated, leading to a robust automated 3D damage detection and classification framework proposed, with the results exceeding any state-of-the-art methodologies to the best of the author's knowledge. Together with advancements in other enabling technologies implemented alongside this research, full automation and in-situ remanufacturing can be achieved. This research also opens up the possibilities for the indirect application of machine learning on geometric data for other purposes, for example for reverse engineering or object recognition for autonomous vehicles, of which non-geometric data can be included to augment any system.

References

1. Lund, R.T., *Remanufacturing, United States experience and implications for developing nations*. 1983: Center for Policy Alternatives, Massachusetts Institute of Technology.
2. MatsumotoDr, M. and W. IjomahDr, *Remanufacturing*. Handbook of sustainable engineering, 2013: p. 389-408.
3. Moan, T., *Reliability-based management of inspection, maintenance and repair of offshore structures*. Structure and Infrastructure Engineering, 2005. **1**(1): p. 33-62.
4. Seitz, M.A. and K. Peattie, *Meeting the Closed-Loop Challenge: The Case of Remanufacturing*. California Management Review, 2004. **46**(2): p. 74-89.
5. Chen, C., et al., *A review on remanufacture of dies and moulds*. Journal of Cleaner Production, 2014. **64**: p. 13-23.
6. Guldur, B., *Laser-based structural sensing and surface damage detection*. 2014: Northeastern University.
7. Eiamsa-ard, K., et al. *Part repair using a hybrid manufacturing system*. in *Proceedings of the Sixteenth Annual Solid Freeform Fabrication Symposium, Austin, TX*. 2005.
8. Sundin, E., *Product and process design for successful remanufacturing*. 2004, Linköping University Electronic Press.
9. Allen, J., et al., *A review of recent developments in the design of special-purpose machine tools with a view to identification of solutions for portable in situ machining systems*. The International Journal of Advanced Manufacturing Technology, 2010. **50**(9-12): p. 843-857.
10. Law, M., H. Rentzsch, and S. Ihlenfeldt, *Development of a dynamic substructuring framework to facilitate in situ machining solutions using mobile machine tools*. Procedia Manufacturing, 2015. **1**: p. 756-767.
11. Torims, T., *The application of laser cladding to mechanical component repair, renovation and regeneration*. Daaam international scientific book, 2013. **12**: p. 587-608.
12. Uriarte, L., et al., *Machine tools for large parts*. CIRP Annals, 2013. **62**(2): p. 731-750.
13. Jin, X., et al., *Assembly strategies for remanufacturing systems with variable quality returns*. IEEE Transactions on Automation Science and Engineering, 2013. **10**(1): p. 76-85.
14. Cotton, H., *Welded steel for offshore construction*. Proceedings of the Institution of Mechanical Engineers, 1979. **193**(1): p. 193-206.
15. ASTM, *Standard Terminology for Additive Manufacturing Technologies*. 2012, ASTM International.

16. Motavalli, S. and B. Bidanda, *A part image reconstruction system for reverse engineering of design modifications*. Journal of Manufacturing Systems, 1991. **10**(5): p. 383-395.
17. Paquette, S., *3D scanning in apparel design and human engineering*. IEEE Computer Graphics and Applications, 1996. **16**(5): p. 11-15.
18. Stuyck, T., et al., *HR-Kinect-High-Resolution Dynamic 3D Scanning for Facial Expression Analysis*, in K.U. Leuven, Faculty of Engineering, ESAT/PSI, IBBT-K.U. 2014.
19. Viana, J.H.M., S. Hinduja, and P.J.S. Bártolo, *Estimation of biometric parameters from cattle rump using free-hand scanning and a 3D data processing algorithm*. Virtual and Physical Prototyping, 2016. **11**(3): p. 167-172.
20. Friel, K., et al. *The use of 3D scanning for sporting applications*. in *Three-Dimensional Imaging, Visualization, and Display 2015*. 2015. International Society for Optics and Photonics.
21. Pesce, M., L.M. Galantucci, and F. Lavecchia, *A 12-camera body scanning system based on close-range photogrammetry for precise applications*. Virtual and Physical Prototyping, 2016. **11**(1): p. 49-56.
22. Grivon, D., E. Vezzetti, and M.G. Violante, *Development of an innovative low-cost MARG sensors alignment and distortion compensation methodology for 3D scanning applications*. Robotics and Autonomous Systems, 2013. **61**(12): p. 1710-1716.
23. Georgiadis, M., et al., *3D scanning SAXS: a novel method for the assessment of bone ultrastructure orientation*. Bone, 2015. **71**: p. 42-52.
24. Aberman, K., et al., *Dip transform for 3D shape reconstruction*. ACM Trans. Graph., 2017. **36**(4): p. 1-11.
25. Li, L., et al., *An integrated approach of reverse engineering aided remanufacturing process for worn components*. Robotics and Computer-Integrated Manufacturing, 2017. **48**: p. 39-50.
26. Gao, J., et al., *An integrated adaptive repair solution for complex aerospace components through geometry reconstruction*. The International Journal of Advanced Manufacturing Technology, 2008. **36**(11): p. 1170-1179.
27. Gibson, I., D. Rosen, and B. Stucker, *Directed Energy Deposition Processes*, in *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*. 2015, Springer New York: New York, NY. p. 245-268.
28. Vilar, R. and A. Almeida, *Repair and manufacturing of single crystal Ni-based superalloys components by laser powder deposition—A review*. Journal of Laser Applications, 2015. **27**(S1): p. S17004.
29. Karunakaran, K., et al., *Low cost integration of additive and subtractive processes for hybrid layered manufacturing*. Robotics and Computer-Integrated Manufacturing, 2010. **26**(5): p. 490-499.

30. Rivette, M., J. Hacoët, and P. Mognol, *A graph-based methodology for hybrid rapid design*. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2007. **221**(4): p. 685-697.
31. Nau, B., A. Roderburg, and F. Klocke, *Ramp-up of hybrid manufacturing technologies*. CIRP Journal of Manufacturing Science and Technology, 2011. **4**(3): p. 313-316.
32. Lauwers, B., F. Klocke, and A. Klink. *Advanced manufacturing through the implementation of hybrid and media assisted processes*. in *International Chemnitz Manufacturing Colloquium*. 2010.
33. Suryakumar, S., et al., *Weld bead modeling and process optimization in hybrid layered manufacturing*. Computer-Aided Design, 2011. **43**(4): p. 331-344.
34. Liang, H., H. Hong, and J. Svoboda, *A combined 3D linear and circular interpolation technique for multi-axis CNC machining*. Transactions-American Society of Mechanical Engineers Journal of Manufacturing Science and Engineering, 2002. **124**(2): p. 305-312.
35. Teng, X., et al., *Maintainability of ship and marine equipment [J]*. Journal of Shanghai Maritime University, 2007. **1**: p. 017.
36. Hatcher, G.D., W.L. Ijomah, and J.F.C. Windmill, *Design for remanufacture: a literature review and future research needs*. Journal of Cleaner Production, 2011. **19**(17): p. 2004-2014.
37. Ijomah, W.L., et al., *Development of design for remanufacturing guidelines to support sustainable manufacturing*. Robotics and Computer-Integrated Manufacturing, 2007. **23**(6): p. 712-719.
38. Ijomah, W.L., et al., *Development of robust design-for-remanufacturing guidelines to further the aims of sustainable development*. International Journal of Production Research, 2007. **45**(18-19): p. 4513-4536.
39. Jukun, Y., Z. Sheng, and C. Peizhi. *Advanced Information-Based Remanufacturing Technology and Its Application*. in *Digital Manufacturing and Automation (ICDMA), 2010 International Conference on*. 2010. IEEE.
40. Steinhilper, R., *Remanufacturing: the ultimate form of recycling*. Fraunhofer IRB Verlag, 1998.
41. Steinhilper, R. *Recent trends and benefits of remanufacturing: from closed loop businesses to synergetic networks*. in *Proceedings Second International Symposium on Environmentally Conscious Design and Inverse Manufacturing*. 2001.
42. Schoefs, F. and A. Clement. *Multiple inspection modeling for decision making and management of jacket off-shore platforms: effect of false alarms*. in *International Forum on Engineering Decision Making, Switzerland*. 2004.
43. AlShaer, A.W., L. Li, and A. Mistry, *The effects of short pulse laser surface cleaning on porosity formation and reduction in laser welding of*

- aluminium alloy for automotive component manufacture*. Optics & Laser Technology, 2014. **64**: p. 162-171.
44. Liu, W., et al., *Study on remanufacturing cleaning technology in mechanical equipment remanufacturing process*, in *Re-engineering Manufacturing for Sustainability*. 2013, Springer. p. 643-648.
 45. Kikuchi, E., Y. Kikuchi, and M. Hirao, *Analysis of risk trade-off relationships between organic solvents and aqueous agents: case study of metal cleaning processes*. Journal of Cleaner Production, 2011. **19**(5): p. 414-423.
 46. Guha, A., R.M. Barron, and R. Balachandar, *An experimental and numerical study of water jet cleaning process*. Journal of Materials Processing Technology, 2011. **211**(4): p. 610-618.
 47. Kambham, K., et al., *Copper slag: optimization of productivity and consumption for cleaner production in dry abrasive blasting*. Journal of Cleaner Production, 2007. **15**(5): p. 465-473.
 48. Yang, S.-C., K.-S. Huang, and Y.-C. Lin, *Optimization of a pulsed carbon dioxide snow jet for cleaning CMOS image sensors by using the Taguchi method*. Sensors and Actuators A: Physical, 2007. **139**(1-2): p. 265-271.
 49. Li, M., et al., *Pre-treatment of remanufacturing cleaning by use of supercritical CO₂ in comparison with thermal cleaning*. Clean Technologies and Environmental Policy, 2015. **17**(6): p. 1563-1572.
 50. Niemczewski, B., *Observations of water cavitation intensity under practical ultrasonic cleaning conditions*. Ultrasonics sonochemistry, 2007. **14**(1): p. 13-18.
 51. Cuthberston, J., *Electrolytic cleaning and polishing*. Journal of the Institution of Electrical Engineers, 1955. **1**(9): p. 601-602.
 52. Takahagi, T., et al., *The formation of hydrogen passivated silicon single - crystal surfaces using ultraviolet cleaning and HF etching*. Journal of applied physics, 1988. **64**(7): p. 3516-3521.
 53. Kegel, B. and H. Schmid, *Low-pressure plasma cleaning of metallic surfaces on industrial scale*. Surface and Coatings Technology, 1999. **112**(1-3): p. 63-66.
 54. Razek, N., A. Schindler, and B. Rauschenbach, *Ultra-high vacuum direct bonding of ap-n junction GaAs wafer using low-energy hydrogen ion beam surface cleaning*. Vacuum, 2007. **81**(8): p. 974-978.
 55. Ye, Y., et al., *Laser cleaning of particle and grease contaminations on the surface of optics*. Optik-International Journal for Light and Electron Optics, 2012. **123**(12): p. 1056-1060.
 56. Della Porta, G., M. Volpe, and E. Reverchon, *Supercritical cleaning of rollers for printing and packaging industry*. The Journal of supercritical fluids, 2006. **37**(3): p. 409-416.
 57. Basketter, D., et al., *Enzymes in cleaning products: an overview of toxicological properties and risk assessment/management*. Regulatory Toxicology and pharmacology, 2012. **64**(1): p. 117-123.

58. Fang, X., H. Luo, and J. Tang, *Structural damage detection using neural network with learning rate improvement*. Computers & structures, 2005. **83**(25-26): p. 2150-2161.
59. Sobel, I. and G. Feldman, *A 3x3 isotropic gradient operator for image processing*. a talk at the Stanford Artificial Project in, 1968: p. 271-272.
60. Canny, J., *A computational approach to edge detection*. IEEE Transactions on pattern analysis and machine intelligence, 1986(6): p. 679-698.
61. Chen, Z. and T.C. Hutchinson, *Image-based framework for concrete surface crack monitoring and quantification*. Advances in Civil Engineering, 2010.
62. Cooley, J.W. and J.W. Tukey, *An algorithm for the machine calculation of complex Fourier series*. Mathematics of computation, 1965. **19**(90): p. 297-301.
63. Van Loan, C., *Computational frameworks for the fast Fourier transform*. Vol. 10. 1992: Siam.
64. Cochran, W.T., et al., *What is the fast Fourier transform?* Proceedings of the IEEE, 1967. **55**(10): p. 1664-1674.
65. Haar, A., *Zur theorie der orthogonalen funktionensysteme*. Mathematische Annalen, 1910. **69**(3): p. 331-371.
66. Stanković, R.S. and B.J. Falkowski, *The Haar wavelet transform: its status and achievements*. Computers & Electrical Engineering, 2003. **29**(1): p. 25-44.
67. Wiseman, Y., *The effectiveness of JPEG images produced by a standard digital camera to detect damaged tyres*. World Review of Intermodal Transportation Research, 2013. **4**(1): p. 23-36.
68. Samuel, A., *Some Studies in Machine Learning Using the Game of Checkers*. IBM Journal of Research and Development, 1959. **3**(44): p. 206-226.
69. Ho, H.-N., et al., *An efficient image-based damage detection for cable surface in cable-stayed bridges*. Ndt & E International, 2013. **58**: p. 18-23.
70. Cha, Y.J., W. Choi, and O. Büyüköztürk, *Deep learning - based crack damage detection using convolutional neural networks*. Computer - Aided Civil and Infrastructure Engineering, 2017. **32**(5): p. 361-378.
71. Cha, Y.J., et al., *Autonomous structural visual inspection using region - based deep learning for detecting multiple damage types*. Computer - Aided Civil and Infrastructure Engineering, 2018. **33**(9): p. 731-747.
72. Torok, M.M., M. Golparvar-Fard, and K.B. Kochersberger, *Image-based automated 3D crack detection for post-disaster building assessment*. Journal of Computing in Civil Engineering, 2013. **28**(5): p. A4014004.
73. Lading, L., et al., *Fundamentals for remote structural health monitoring of wind turbine blades-a preproject*. Annex B. Sensors and non-

destructive testing methods for damage detection in wind turbine blades. 2002.

74. Michaels, T., et al. *Damage detection in plate structures using sparse ultrasonic transducer arrays and acoustic wavefield imaging.* in *AIP Conference Proceedings*. 2005. AIP.
75. Avdelidis, N., et al. *Structural integrity assessment of materials by thermography.* in *Conf. Damage in Composite Materials CDCM, Stuttgart, Germany*. 2006.
76. Rantala, J., D. Wu, and G. Busse, *Amplitude-modulated lock-in vibrothermography for NDE of polymers and composites.* *Research in Nondestructive Evaluation*, 1996. 7(4): p. 215-228.
77. Li, J., et al. *Reconstruction of broken blade geometry model based on reverse engineering.* in *Intelligent Networks and Intelligent Systems (ICINIS), 2010 3rd International Conference on*. 2010. IEEE.
78. Zheng, J., Z. Li, and X. Chen, *Worn area modeling for automating the repair of turbine blades.* *The International Journal of Advanced Manufacturing Technology*, 2006. 29(9): p. 1062-1067.
79. Edris, M., M. Jawad, and Z. Zakaria. *Surface defect detection and Neural Network recognition of automotive body panels.* in *Control System, Computing and Engineering (ICCSC), 2015 IEEE International Conference on*. 2015. IEEE.
80. Fantini, M., et al., *3D restitution, restoration and prototyping of a medieval damaged skull.* *Rapid Prototyping Journal*, 2008. 14(5): p. 318-324.
81. Bonse, U., et al., *3D computed X-ray tomography of human cancellous bone at 8 μm spatial and 10– 4 energy resolution.* *Bone and mineral*, 1994. 25(1): p. 25-38.
82. Park, H., et al., *A new approach for health monitoring of structures: terrestrial laser scanning.* *Computer - Aided Civil and Infrastructure Engineering*, 2007. 22(1): p. 19-30.
83. Olsen, M.J., et al., *Terrestrial laser scanning-based structural damage assessment.* *Journal of Computing in Civil Engineering*, 2009. 24(3): p. 264-272.
84. Papadopoulos, G., et al. *3D-surface reconstruction for partially submerged marine structures using an Autonomous Surface Vehicle.* in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. 2011. IEEE.
85. Kurniawati, H., et al. *Infrastructure for 3D model reconstruction of marine structures.* in *The Twenty-first International Offshore and Polar Engineering Conference*. 2011. International Society of Offshore and Polar Engineers.
86. Chmielewski, L.J., et al. *Defect detection in furniture elements with the Hough transform applied to 3D data.* in *Proceedings of the 9th*

International Conference on Computer Recognition Systems CORES 2015. 2016. Springer.

87. Peng, Z., L. Yue, and N. Xiao, *Simultaneous Wood Defect and Species Detection with 3D Laser Scanning Scheme*. International Journal of Optics, 2016.
88. Levoy, M., et al. *The digital Michelangelo project: 3D scanning of large statues*. in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 2000. ACM Press/Addison-Wesley Publishing Co.
89. Vezzetti, E., *Computer aided inspection: design of customer-oriented benchmark for noncontact 3D scanner evaluation*. The International Journal of Advanced Manufacturing Technology, 2009. **41**(11): p. 1140-1151.
90. Fourie, Z., et al., *Evaluation of anthropometric accuracy and reliability using different three-dimensional scanning systems*. Forensic Science International, 2011. **207**(1): p. 127-134.
91. Persson, A., et al., *A three-dimensional evaluation of a laser scanner and a touch-probe scanner*. The Journal of prosthetic dentistry, 2006. **95**(3): p. 194-200.
92. Nuikka, M., et al., *Comparison of three accurate 3d measurement methods for evaluating as-built floor flatness*. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2008. **37**(B5): p. 129-134.
93. Martínez, S., et al., *Analysis of laser scanning and strategies for dimensional and geometrical control*. The International Journal of Advanced Manufacturing Technology, 2010. **46**(5): p. 621-629.
94. Kersten, T., et al., *Comparative geometrical investigations of hand-held scanning systems*. ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2016: p. 507-514.
95. Mantič, M., et al., „*Influence of selected digitization methods on final accuracy of 3D model*”. Production Management and Engineering Sciences, 2015: p. 475-480.
96. Vezzetti, E., *Pitch function comparison methodology for supporting a smart 3D scanner selection*. Precision Engineering, 2010. **34**(2): p. 327-337.
97. Standardization, I.O.f., *ISO 5725-2: 1994: Accuracy (Trueness and Precision) of Measurement Methods and Results-Part 2:... Methods for the Determination of Repeatability and Reproducibility*. 1994: International Organization for Standardization.
98. Reshetyuk, Y., *Investigation and calibration of pulsed time-of-flight terrestrial laser scanners*. 2006, KTH.
99. Hebert, M. and E. Krotkov, *3D measurements from imaging laser radars: how good are they?* Image and vision computing, 1992. **10**(3): p. 170-178.

100. Harris, J. and A. Spence, *Geometric and quasi-static thermal error compensation for a laser digitizer equipped coordinate measuring machine*. International Journal of Machine Tools and Manufacture, 2004. **44**(1): p. 65-77.
101. Cosarca, C., A. Jocea, and A. Savu, *Analysis of error sources in terrestrial laser scanning*. RevCAD—Journal of Geodesy and Cadastre, 2009. **9**(1): p. 115-124.
102. World Health Organization, W., *Environmental Health Criteria 14. Ultraviolet radiation*. 1979: United Nations Environment Programme/WHO, Geneva, Switzerland.
103. Runkle, R.C., et al., *Photon and neutron interrogation techniques for chemical explosives detection in air cargo: A critical review*. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 2009. **603**(3): p. 510-528.
104. Zhigang, Z., Z. Li, and L. Jiayan. *3D Measurements in Cargo Inspection with a Gamma-Ray Linear Pushbroom Stereo System*. in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*. 2005.
105. Friedman, N., *The naval institute guide to world naval weapon systems*. 1997: Naval Institute Press.
106. Linsen, L., *Point cloud representation*. 2001: Univ., Fak. für Informatik, Bibliothek.
107. Mederos, B., L. Velho, and L.H.d. Figueiredo, *Smooth surface reconstruction from noisy clouds*. Journal of the Brazilian Computer Society, 2004. **9**(3): p. 52-66.
108. Weyrich, T., et al., *Post-processing of Scanned 3D Surface Data*. SPBG, 2004. **4**: p. 85-94.
109. Dey, T., S. Goswami, and J. Sun, *Smoothing noisy point clouds with Delaunay preprocessing and MLS*. The Ohio State University Technical Report No. OSU-CISRC-3/04-TR17, 2004.
110. Lange, C. and K. Polthier, *Anisotropic smoothing of point sets*. Computer Aided Geometric Design, 2005. **22**(7): p. 680-692.
111. Xie, H., K.T. McDonnell, and H. Qin. *Surface reconstruction of noisy and defective data sets*. in *Proceedings of the conference on Visualization'04*. 2004. IEEE Computer Society.
112. Pauly, M., N.J. Mitra, and L.J. Guibas, *Uncertainty and variability in point cloud surface data*. SPBG, 2004. **4**: p. 77-84.
113. Ivriissimtzis, I., et al. *Neural mesh ensembles*. in *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*. 2004. IEEE.
114. Giesen, J., S. Spalinger, and B. Schölkopf. *Kernel methods for implicit surface modeling*. in *Advances in neural information processing systems*. 2005.

115. Fenn, M. and G. Steidl, *Robust local approximation of scattered data*, in *Geometric Properties for Incomplete Data*. 2006, Springer. p. 317-334.
116. Steinke, F., B. Schölkopf, and V. Blanz. *Support vector machines for 3D shape processing*. in *Computer Graphics Forum*. 2005. Wiley Online Library.
117. Schall, O., A. Belyaev, and H.-P. Seidel. *Robust filtering of noisy scattered point data*. in *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*. 2005. IEEE.
118. Rusu, R.B., et al., *Towards 3D point cloud based object maps for household environments*. *Robotics and Autonomous Systems*, 2008. **56**(11): p. 927-941.
119. Sun, Y., S. Schaefer, and W. Wang, *Denoising point sets via L0 minimization*. *Computer Aided Geometric Design*, 2015. **35**: p. 2-15.
120. Besl, P.J. and N.D. McKay, *A method for registration of 3-D shapes*. *IEEE Transactions on pattern analysis and machine intelligence*, 1992. **14**(2): p. 239-256.
121. Chen, Y. and G. Medioni, *Object modelling by registration of multiple range images*. *Image and vision computing*, 1992. **10**(3): p. 145-155.
122. Greenspan, M. and G. Godin. *A nearest neighbor method for efficient ICP*. in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. 2001. IEEE.
123. Trucco, E., A. Fusiello, and V. Roberto, *Robust motion and correspondence of noisy 3-D point sets with missing data*. *Pattern recognition letters*, 1999. **20**(9): p. 889-898.
124. Zinßer, T., J. Schmidt, and H. Niemann. *A refined ICP algorithm for robust 3-D correspondence estimation*. in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*. 2003. IEEE.
125. Jost, T. and H. Hugli. *A multi-resolution scheme ICP algorithm for fast shape registration*. in *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*. 2002. IEEE.
126. Sharp, G.C., S.W. Lee, and D.K. Wehe, *ICP registration using invariant features*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002. **24**(1): p. 90-102.
127. Salvi, J., et al., *A review of recent range image registration methods with accuracy evaluation*. *Image and Vision computing*, 2007. **25**(5): p. 578-596.
128. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning*. *Nature*, 2015. **521**(7553): p. 436-444.
129. Masci, J., et al. *Geometric deep learning*. in *SIGGRAPH ASIA 2016 Courses*. 2016. ACM.

130. Rabah, M., A. Elhatab, and A. Fayad, *Automatic concrete cracks detection and mapping of terrestrial laser scan data*. NRIAG Journal of Astronomy and Geophysics, 2013. **2**(2): p. 250-255.
131. Adhikari, R., O. Moselhi, and A. Bagchi, *Image-based retrieval of concrete crack properties for bridge inspection*. Automation in construction, 2014. **39**: p. 180-194.
132. Torok, M., M. Fard, and K. Kochersberger, *Post-Disaster robotic building assessment: Automated 3D crack detection from image-based reconstructions*, in *Computing in Civil Engineering (2012)*. 2012. p. 397-404.
133. Teza, G., A. Galgaro, and F. Moro, *Contactless recognition of concrete surface damage from laser scanning and curvature computation*. NDT & E International, 2009. **42**(4): p. 240-249.
134. Liu, W., S. Chen, and E. Hauser, *LiDAR-based Bridge Structure Defect Detection*. Experimental Techniques, 2011. **35**(6): p. 27-34.
135. Khoshelham, K., S.O. Elberink, and S. Xu, *Segment-based classification of damaged building roofs in aerial laser scanning data*. IEEE Geoscience and Remote Sensing Letters, 2013. **10**(5): p. 1258-1262.
136. Guldur, B. and J. Hajjar. *Automated Damage Assessment from 3-D Laser Scans*. in *Proceedings of the Istanbul Bridge Conference*. 2014.
137. Tjahjowidodo, T. and M. Han. *Automatic defect detection and the estimation of nominal profiles based on spline for free-form surface parts*. in *Advanced Intelligent Mechatronics (AIM), 2015 IEEE International Conference on*. 2015. IEEE.
138. Ghahremani, K., et al., *Damage Detection and Finite-Element Model Updating of Structural Components through Point Cloud Analysis*. Journal of Aerospace Engineering, 2018. **31**(5): p. 04018068.
139. Rehor, M., *Classification of building damage based on laser scanning data*. The Photogrammetric Journal of Finland, 2007. **20**(2): p. 54-63.
140. Schweier, C. and M. Markus. *Assessment of the search and rescue demand for individual buildings*. in *Proceedings of the 13th World Conference on Earthquake Engineering, Vancouver, BC, Canada*. 2004.
141. Djamaluddin, I., Y. Mitani, and T. Esaki, *Evaluation of ground movement and damage to structures from Chinese coal mining using a new GIS coupling model*. International Journal of Rock Mechanics and Mining Sciences, 2011. **48**(3): p. 380-393.
142. Lee, C. and S. Park, *Damage classification of pipelines under water flow operation using multi-mode actuated sensing technology*. Smart Materials and Structures, 2011. **20**(11): p. 115002.
143. Huang, B., B.-H. Koh, and H.S. Kim, *PCA-based damage classification of delaminated smart composite structures using improved layerwise theory*. Computers & Structures, 2014. **141**: p. 26-35.

144. Katunin, A., *Damage detection and classification in composite structure after water-jet cutting using computed tomography and wavelet analysis*. Emerg Technol Non-Destruct Test VI, 2016. **78**: p. 557-562.
145. Qi, C.R., et al. *Pointnet: Deep learning on point sets for 3d classification and segmentation*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.
146. Bendich, P., et al. *Multi-scale local shape analysis and feature selection in machine learning applications*. in *2015 International Joint Conference on Neural Networks (IJCNN)*. 2015. IEEE.
147. Elbaz, G., T. Avraham, and A. Fischer. *3D point cloud registration for localization using a deep neural network auto-encoder*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.
148. Tootooni, M.S., et al., *Classifying the dimensional variation in additive manufactured parts from laser-scanned three-dimensional point cloud data using machine learning approaches*. Journal of Manufacturing Science and Engineering, 2017. **139**(9): p. 091005.
149. Czerniawski, T., et al., *6D DBSCAN-based segmentation of building point clouds for planar object classification*. Automation in Construction, 2018. **88**: p. 44-58.
150. Mfame. *Water-Backed Welding Procedure Helps Avoid Drydocking*. 2016 [cited 2020 20 October]; Available from: <https://mfame.guru/water-backed-welding-procedure-helps-avoid-drydocking/>.
151. Welding, M. *Maritime Propeller Repairs Certified for In-Situ Welding*. 2015 [cited 2020 22 October]; Available from: <https://www.marinelink.com/news/propeller-certified385506>.
152. MX3D. *MX3D Robots*. 2019; Available from: <https://mx3d.com/>.
153. Capriol. *Robotics Solutions*. 2019; Available from: <https://capriol.us/robotics-solutions/>.
154. Golisano Institute for Sustainability, R.I.o.T., Rochester, NY, *Technology Roadmap for Remanufacturing in the Circular Economy*. 2017.
155. Zhang, S., et al., *Theoretical Analysis and Simulation of High Pressure Water Jet Cleaning Process*. The Open Mechanical Engineering Journal, 2015. **9**(1).
156. Kabir, A.M., et al. *Planning algorithms for multi-setup multi-pass robotic cleaning with oscillatory moving tools*. in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*. 2016. IEEE.
157. James, B., *Modeling of Complex Parts for Industrial WaterJet Cleaning*, in *Industrial Engineering*. 2019, University of Arkansas, Fayetteville.
158. Artec. *RobotiScan*. 2019; Available from: <https://www.artec3d.com/portable-3d-scanners/robotic-scan>.
159. Creaform. *MetraSCAN 3D-R*. 2019; Available from: <https://www.creaform3d.com/en/metrology-solutions/robot-mounted-optical-cmm-scanners-metrascan-r#gref>.

160. Faro. *FARO® Cobalt Structured Light Scanner*. 2019; Available from: <https://www.faro.com/en-sg/products/3d-manufacturing/faro-cobalt-array-imager/>.
161. Shen, Z., et al., *Smoothing and Reconstruction Strategy for Part Repair Using a Laser Displacement Sensor*. Journal of Engineering Science & Technology Review, 2018. **11**(5).
162. Barros, L.L., *The optimization of repair decision using life-cycle cost parameters*. IMA Journal of Management Mathematics, 1998. **9**(4): p. 403-413.
163. Dhillon, B.S., *Life cycle costing for engineers*. 2009: Crc Press.
164. Frangopol, D., M.P. Enright, and A.C. Estes. *Integration of maintenance, repair, and replacement decisions in bridge management based on reliability, optimization, and life-cycle cost*. in *Presentations from the 8th International Bridge Management Conference, Denver, Colo.* 1999. Citeseer.
165. Saranga, H. and U.D. Kumar, *Optimization of aircraft maintenance/support infrastructure using genetic algorithms—level of repair analysis*. Annals of Operations Research, 2006. **143**(1): p. 91.
166. Gutin, G., et al., *Level of repair analysis and minimum cost homomorphisms of graphs*. Discrete Applied Mathematics, 2006. **154**(6): p. 881-889.
167. Basten, R.J., J.M.J. Schutten, and M.C. Van Der Heijden, *An efficient model formulation for level of repair analysis*. Annals of Operations Research, 2009. **172**(1): p. 119.
168. Barros, L. and M. Riley, *A combinatorial approach to level of repair analysis*. European Journal of Operational Research, 2001. **129**(2): p. 242-251.
169. Krolczyk, G., et al., *Influence of argon pollution on the weld Surface Morphology*. Measurement, 2015. **70**: p. 203-213.
170. Dawood, H.I., et al., *The influence of the surface roughness on the microstructures and mechanical properties of 6061 aluminium alloy using friction stir welding*. Surface and Coatings Technology, 2015. **270**: p. 272-283.
171. Turner, B.N. and S.A. Gold, *A review of melt extrusion additive manufacturing processes: II. Materials, dimensional accuracy, and surface roughness*. Rapid Prototyping Journal, 2015. **21**(3): p. 250-261.
172. Bikas, H., P. Stavropoulos, and G. Chrysosolouris, *Additive manufacturing methods and modelling approaches: a critical review*. The International Journal of Advanced Manufacturing Technology, 2016. **83**(1-4): p. 389-405.
173. Townsend, A., et al., *Surface texture metrology for metal additive manufacturing: a review*. Precision Engineering, 2016. **46**: p. 34-47.

174. Chu, W.-S., et al., *Hybrid manufacturing in micro/nano scale: A Review*. International journal of precision engineering and manufacturing-green technology, 2014. **1**(1): p. 75-92.
175. Zhu, Z., et al., *A review of hybrid manufacturing processes—state of the art and future perspectives*. International Journal of Computer Integrated Manufacturing, 2013. **26**(7): p. 596-615.
176. Ruan, J., K. Eiamsa-ard, and F.W. Liou, *Automatic process planning and toolpath generation of a multiaxis hybrid manufacturing system*. Journal of manufacturing processes, 2005. **7**(1): p. 57-68.
177. Ren, L., et al., *Integrated process planning for a multiaxis hybrid manufacturing system*. Journal of Manufacturing Science and Engineering, 2010. **132**(2): p. 021006.
178. Zhu, Z., V. Dhokia, and S.T. Newman, *The development of a novel process planning algorithm for an unconstrained hybrid manufacturing process*. Journal of manufacturing processes, 2013. **15**(4): p. 404-413.
179. Chen, M. and P. Abrishami, *A mathematical model for production planning in hybrid manufacturing-remanufacturing systems*. The International Journal of Advanced Manufacturing Technology, 2014. **71**(5-8): p. 1187-1196.
180. Zheng, Y., A. Qureshi, and R. Ahmad, *Algorithm for remanufacturing of damaged parts with hybrid 3D printing and machining process*. Manufacturing Letters, 2018. **15**: p. 38-41.
181. LASIMM. *LASIMM project goes live!* 2019; Available from: http://www.lasimm.eu/news-detail_08.html.
182. McKinney. *The AMBIT™ PE-1 extruder for Polymer Hybrid Manufacturing Released in Europe*. 2019; Available from: <http://www.hybridmanutech.com/news.html>.
183. Friedman, J.H., J.L. Bentley, and R.A. Finkel, *An algorithm for finding best matches in logarithmic expected time*. ACM Transactions on Mathematical Software (TOMS), 1977. **3**(3): p. 209-226.
184. Tabak, J., *Geometry: The Language of Space and Form*. 2014: Facts On File, Incorporated.
185. Rusu, R.B., *Semantic 3d object maps for everyday manipulation in human living environments*. KI-Künstliche Intelligenz, 2010. **24**(4): p. 345-348.
186. Hoppe, H., et al., *Surface reconstruction from unorganized points*. Vol. 26. 1992: ACM.
187. Gumhold, S., X. Wang, and R.S. MacLeod. *Feature Extraction From Point Clouds*. in *IMR*. 2001.
188. Hough, P.V. *Machine analysis of bubble chamber pictures*. in *Conf. Proc.* 1959.
189. Tarsha-Kurdi, F., T. Landes, and P. Grussenmeyer. *Hough-transform and extended ransac algorithms for automatic detection of 3d building roof*

- planes from lidar data.* in *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*. 2007.
190. Borrmann, D., et al., *The 3d hough transform for plane detection in point clouds: A review and a new accumulator design*. 3D Research, 2011. **2**(2): p. 3.
 191. Jeltsch, M., C. Dalitz, and R. Pohle-Fröhlich. *Hough Parameter Space Regularisation for Line Detection in 3D*. in *VISIGRAPP (4: VISAPP)*. 2016.
 192. Dalitz, C., T. Schramke, and M. Jeltsch, *Iterative Hough Transform for Line Detection in 3D Point Clouds*. Image Processing On Line, 2017. **7**: p. 184-196.
 193. Roberts, K.S. *A new representation for a line*. in *Computer Vision and Pattern Recognition, 1988. Proceedings CVPR'88., Computer Society Conference on*. 1988. IEEE.
 194. Fischler, M.A. and R.C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*. Communications of the ACM, 1981. **24**(6): p. 381-395.
 195. Nguyen, A. and B. Le. *3D point cloud segmentation: A survey*. in *Robotics, Automation and Mechatronics (RAM), 2013 6th IEEE Conference on*. 2013. IEEE.
 196. Yang, M.Y. and W. Förstner. *Plane detection in point cloud data*. in *Proceedings of the 2nd int conf on machine control guidance, Bonn*. 2010.
 197. Hargittai, I. and M. Hargittai, *Symmetry: A unifying concept*. 1994: Shelter Publications, Inc.
 198. Weyl, H., *Symmetry* Princeton Univ. Press, Princeton, NJ, 1952.
 199. Feynman, R.P., R.B. Leighton, and M. Sands, *Six not-so-easy pieces: Einstein's relativity, symmetry, and space-time*. 2011: Basic Books.
 200. Liu, Y., *Computational symmetry*. Wenner Gren International Series, 2000(1): p. 231-246.
 201. Rosen, J., *Symmetry discovered: Concepts and applications in nature and science*. 2012: Courier Corporation.
 202. Wang, J. and M.M. Oliveira. *Improved scene reconstruction from range images*. in *Computer Graphics Forum*. 2002. Wiley Online Library.
 203. Lowe, D.G. *Object recognition from local scale-invariant features*. in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. 1999. Ieee.
 204. Flint, A., A. Dick, and A. Van Den Hengel. *Thrift: Local 3d structure recognition*. in *Digital Image Computing Techniques and Applications, 9th Biennial Conference of the Australian Pattern Recognition Society on*. 2007. IEEE.
 205. Tombari, F., S. Salti, and L. Di Stefano. *Unique signatures of histograms for local surface description*. in *European conference on computer vision*. 2010. Springer.

206. Chen, H. and B. Bhanu, *3D free-form object recognition in range images using local surface patches*. Pattern Recognition Letters, 2007. **28**(10): p. 1252-1262.
207. European Standard, I.O.f.S., *Geometrical Product Specification (GPS) — Surface imperfections — Terms, definitions and parameters*. 1999, British Standard. p. 20.
208. Zhang, Y., et al., *Remanufacturing-Oriented Geometric Modelling for the Damaged Region of Components*. Procedia CIRP, 2015. **29**: p. 798-803.
209. D'Antonio, C., *Surface Damage on a Transfer Gear Shaft*. ASM International, Handbook of Case Histories in Failure Analysis., 1992. **1**: p. 299-300.
210. Belzona. *Belzona 1511 (Super HT-Metal) High Temperature Repair*. 2017; Available from: <http://www.belzona.com/en/products/1000/1511.aspx>.
211. *02J Trans repair | Kansas City TDI*. 2017; Available from: <http://kansascitytdi.com/02j-trans-repair/>.
212. Engineering, F.P. *Metal Stitching Crack Repair & Applications What's Metal Stitching Crack Repair & It's Applications? Metal Stitching Crack Repair Johor Bahru (JB), Malaysia Service, Repair, Maintenance | Federal Parts Engineering*. 2017; Available from: http://www.fpe.com.my/index.php?ws=showproducts&products_id=1146144&cat=Metal%20Stitching%20Crack%20Repair%20&subcat=What%27s%20Metal%20Stitching%20Crack%20Repair%20&%20It%27s%20Applications?%20.
213. Sardar, J. and D. Bandopadhyaya, *Development and fabrication of cement reinforced polypropylene composite material spur gear*. Journal of Polymer engineering, 2014. **34**(8): p. 775-786.
214. Ahmed, E., et al., *A survey on Deep Learning Advances on Different 3D Data Representations*. arXiv preprint arXiv:1808.01462, 2018.
215. Mathworks. *Choose Classifier Options- MATLAB & Simulink*. 2020 [cited 2020 10 Nov]; Available from: <https://www.mathworks.com/help/stats/choose-a-classifier.html#bunt0ky>.
216. Rokach, L. and O.Z. Maimon, *Data mining with decision trees: theory and applications*. Vol. 69. 2008: World scientific.
217. Levenberg, K., *A method for the solution of certain non-linear problems in least squares*. Quarterly of applied mathematics, 1944. **2**(2): p. 164-168.
218. Marquardt, D.W., *An algorithm for least-squares estimation of nonlinear parameters*. Journal of the society for Industrial and Applied Mathematics, 1963. **11**(2): p. 431-441.

Annex A

Algorithm 4.3.1a: k Nearest Neighbour

Input: PCD, k

Output: $kNearestNeighbours$

- 1: split all the points in PCD using a k-D tree into partitions
 - 2: **for** each point in PCD , **do**
 - 3: **if** point is near splitting plane, **then**
 - 4: merge adjacent partitions temporarily
 - 5: **end if**
 - 6: compute the 3D Euclidean distances between the queried point and other points in partition
 - 7: sort all the 3D Euclidean distances
 - 8: store indices of k number of points with lowest 3D Euclidean distances in $kNearestNeighbours$
 - 9: **end for**
-

Algorithm 4.3.1b: Bounding Sphere Nearest Neighbours

Input: $PCD, radius$

Output: $BoundingSphereNearestNeighbours$

- 1: split all the points in PCD using a k-D tree into partitions
 - 2: **for** each point in PCD , **do**
 - 3: **if** point is near splitting plane, **then**
 - 4: merge adjacent partitions temporarily
 - 5: **end if**
 - 6: compute the 3D Euclidean distances between the queried point and other points in partition
 - 7: **if** 3D Euclidean distance $< radius$, **then**
 - 8: store indices of points with lowest 3D Euclidean distances in $BoundingSphereNearestNeighbours$
 - 9: **end if**
 - 10: **end for**
-

Algorithm 4.3.2: Normal Estimation

Input: *PCD*

Output: *EstimatedNormal*

```
1:  for each point in PCD, do
2:      search for k Nearest Neighbour using Algorithm 4.3.1a
3:      compute the covariance matrix of the point and kNearestNeighbours
4:      compute the eigenvalues and corresponding eigenvector
5:      store eigenvector as EstimatedNormal
6:  end for
7:  for all points in PCD, do
8:      if the product of the query point EstimatedNormal and neighbouring
        point EstimatedNormal < 0, then
9:          reverse the direction of the EstimatedNormal
10:     end if
11: end for
```

Algorithm 4.3.3: Curvature Estimation

Input: *PCD*

Output: *EstimatedCurvature*

```
1:  for each point in PCD, do
2:      search for k Nearest Neighbour using Algorithm 4.3.1a
3:      compute the covariance matrix of the point and kNearestNeighbours
4:      compute the eigenvalues and corresponding eigenvector
5:      compute the ratio of the minimum eigenvalue to the sum of all
        eigenvalues
6:      store the ratio as EstimatedCurvature
7:  end for
```

Algorithm 4.3.4: Edge and Boundary Point Detection

Input: *PCD*, *cThreshold*, *angThreshold*

Output: *EdgePoint*, *BoudaryEdgePoint*

```
1:  for each point in PCD, do
2:      compute Curvature Estimation using algorithm 4.3.3
3:      if EstimatedCurvature > cThreshold, then
4:          store the point as EdgePoint
5:      end if
6:  end for
7:  for each EdgePoint, do
8:      compute Normal Estimation using algorithm 4.3.2
9:      create xy-coordinate system orthogonal to the EstimatedNormal
10:     search for k Nearest Neighbour using Algorithm 4.3.1a
11:     project kNearestNeighbours onto xy-coordinate system
12:     compute the angle formed between the lines connecting the queried
        EdgePoint and each pair of adjacent kNearestNeighbours
13:     if the angle > angThreshold, then
14:         store the queried point as a BoudaryEdgePoint
15:         remove queried point from EdgePoint
16:     end if
17: end for
```

Algorithm 4.3.5: Single Edge Tracing

Input: *PCD*

Output: *EdgeTracePoint*, *EdgeTraceLine*

```
1:  for each point in PCD, do
2:      search for Edge Point using algorithm 4.3.4
3:  end for
4:  for each EdgePoint, do
5:      compute Curvature Estimation using algorithm 4.3.3
6:      search for Bounding Sphere Nearest Neighbours using Algorithm
      4.3.1b
7:      compare EstimatedCurvature of queried point with the
      kNearestNeighbours
8:      if queried point EstimatedCurvature is the largest, then
9:          store point as EdgeTracePoint
10:     end if
11: end for
12: for each EdgeTracePoint, do
13:     search for k Nearest Neighbours using Algorithm 4.3.1a, with k=2
14:     store the 2 lines formed by linking the queried point with the
      kNearestNeighbours as an EdgeTraceLine, unless it is a duplicate
15: end for
```

Algorithm 4.3.6: Hough Transformation for Straight Edge Detection

Input: *EdgeTraceLine*, *lThreshold*

Output: *StraightEdge*

```
1:  for each EdgeTraceLine, do
2:      compute the Hough transformation parameters  $x'$ ,  $y'$ ,  $\phi$  and  $\theta$ 
3:      compute the difference of each parameter to the equivalent parameter
      of all the lines in a tessellated icosahedron
4:      compute the sum of the difference of all the parameters
5:      store the line in the bin of the tessellated icosahedron line where the
      sum of difference is the least
6:      if the count of stored lines with a tessellated icosahedron line >
      lThreshold, then
7:          store lines as StraighEdge
8:      end if
9:  end for
```

Algorithm 4.4.1.1: Hole

Input: *PCD, hThreshold*

Output: *HolePoints*

```
1:  for each point in PCD, do
2:      find Boundary Point with algorithm 4.3.4
3:  end for
4:  while there are remaining BoudaryEdgePoint
5:      select random BoudaryEdgePoint as seed point
6:      search for k Nearest Neighbour BoudaryEdgePoint using Algorithm
        4.3.1a
7:      repeat k Nearest Neighbour search from kNearestNeighbour, until no
        new BoudaryEdgePoint is found
8:      count number of BoudaryEdgePoint found
9:      if count < hThreshold, then
10:         store all found points as HolePoints
11:      end if
12: end while
```

Algorithm 4.4.1.2: Edge Shape Irregularity

Input: *PCD, sThreshold*

Output: *IrregularEdgeShape*

```
1:  conduct the single edge tracing on the PCD using algorithm 4.3.5
2:  for each EdgeTracePoint, do
3:      compute the angle formed between the two EdgeTraceLine from the
        EdgeTracePoint
4:  end for
5:  for each EdgeTracePoint, do
6:      find the k Nearest Neighbour EdgeTracePoint using Algorithm 4.3.1a
7:      compute the standard deviation of the stored angle value as calculated
        in step 3, of the kNearestNeighbour
8:      if the standard deviation > sthreshold, then
9:         store queried EdgeTracePoint as IrregularEdgeShape
10:     end if
11: end for
```

Algorithm 4.4.1.3: Edge Orientation Irregularity

Input: *PCD*, *bThreshold*

Output: *IrregularEdgeOrientation*

- 1: conduct the single edge tracing on the *PCD* using algorithm 4.3.5
 - 2: **for** each *EdgeTraceLine*, **do**
 - 3: do the Hough Transformation for Straight Edge Detection from Algorithm 4.3.6
 - 4: **end for**
 - 6: **if** bin count < *bThreshold*, **then**
 - 7: store the points that form the *EdgeTraceLine* in the low count bin as *IrregularEdgeOrientation*
 - 8: **end if**
-

Algorithm 4.4.1.4: Rough Surface

Input: *PCD*, *rThreshold*

Output: *RoughSurfacePoint*

- 1: **for** each point in *PCD*, **do**
 - 2: estimate the curvature using algorithm 4.3.3
 - 3: **end for**
 - 4: **for** each point in *PCD*, **do**
 - 5: search for *k* Nearest Neighbour using Algorithm 4.3.1a
 - 6: compute the standard deviation of curvature at the *kNearestNeighbour* points
 - 7: **if** standard deviation > *rThreshold*, **then**
 - 8: store point as *RoughSurfacePoint*
 - 9: **end if**
 - 10: **end for**
-

Algorithm 4.4.2.1: Deviation from Plane

Input: *PCD*, *kRepeat*, *planeLL*, *pThreshold*, *planeUL*

Output: *ObjectPlane*, *PlaneDeviationPoint*

```
1:  for kRepeat number of times, do
2:      select randomly 3 unique and non-colinear points from PCD
3:      extract the plane formed by the 3 points
4:      compute the distance of all points to the plane
5:      count number of points with distance less than planeLL
6:      if number of points < pThreshold, then
7:          store plane as ObjectPlane
8:      end if
9:  end for
10: for each ObjectPlane, do
11:     search for points with distance less than the defined upper limit
        distance, planeUL
12:     form a convex hull of detected points
13:     remove points near boundary of the convex hull
14:     remove points that are within the lower limit distance, planeLL, from
        the ObjectPlane, to account for surface roughness of the plane
13:     store remaining points as PlaneDeviationPoint
14: end for
```

Algorithm 4.4.2.2: Edge Discontinuity

Input: *PCD*, *eThreshold*

Output: *EdgeDiscontinuityPoints*

```
1:  conduct the single edge tracing on the PCD using algorithm 4.3.5
2:  for each EdgeTraceLine, do
3:      do the Hough Transformation for Straight Edge Detection from
        Algorithm 4.3.6
4:  end for
5:  search for colinear StraightEdge
6:  search for missing segments of colinear StraightEdge
7:  if length of missing segments of colinear StraightEdge < eThreshold,
    then
8:      store points near missing segments of colinear StraightEdge as
        EdgeDiscontinuityPoints
9:  end if
```

Algorithm 4.4.2.3: Corner Discontinuity

Input: *PCD*

Output: *CornerDiscontinuityPoints*

```
1:  conduct the single edge tracing on the PCD using algorithm 4.3.5
2:  for each EdgeTraceLine, do
3:      do the Hough Transformation for Straight Edge Detection from
        Algorithm 4.3.6
4:  end for
5:  for each set of tessellated icosahedron lines of a point, do
6:      if there are three or more detected StraightEdge represented by the
        local maxima, i.e. there is a corner, then
7:          find deviated PCD points from corner
8:          store deviated PCD points as CornerDiscontinuityPoints
9:      end if
10: end for
```

Algorithm 4.4.2.4: Deviation from Cylinder

Input: *PCD*, *kRepeat*, *cylinderLL*, *cyThreshold*, *cylinderUL*

Output: *ObjectCylinder*, *CylinderDeviationPoint*

```
1:  for kRepeat number of times, do
2:      select randomly 5 unique and non-collinear points from PCD
3:      extract the cylinder formed by the 5 points
4:      compute the distance of all points to the cylinder
5:      count number of points with distance less than cylinderLL
6:      if number of points < cyThreshold, then
7:          store plane as ObjectCylinder
8:      end if
9:  end for
10: for each ObjectCylinder, do
11:     search for points with distance less than the defined upper limit
        distance, cylinderUL
12:     form a convex hull of detected points
13:     remove points near boundary of the convex hull
14:     remove points that are within the lower limit distance, cylinderLL,
        from the ObjectCylinder, to account for surface roughness of the
        plane
13:     store remaining points as CylinderDeviationPoint
14: end for
```

Algorithm 4.4.3.1: Planar Symmetry Deviation

Input: *PCD*, *psThreshold*, *sdThreshold*

Output: *SymmetryPlane*, *PlanarSymmetryDeviationPoint*

```
1:  for each pair of point in PCD, do
2:      compute the bisector plane
3:      case a vote for the bisector place
4:  end for
5:  for each bisector plane, do
6:      if vote of bisector plane > psThreshold, then
7:          store bisector plane as SymmetryPlane
8:          reflect the points in the PCD in the SymmetryPlane
9:          for each point in PCD, do
10:             compute the 3D Euclidean distance to the nearest mirrored
                PCD
11:             if distance > sdThreshold, then
12:                 store point as PlanarSymmetryDeviationPoint
13:             end if
14:          end for
15:      end if
16:  end for
```

Algorithm 4.4.3.2: Axial Symmetry Deviation

Input: *PCD*, *saThreshold*

Output: *SymmetryAxis*

```
1:  for each pair of point in PCD, do
2:      compute the bisector plane
3:      case a vote for the bisector place
4:  end for
5:  if a series of planes received as many votes, then
6:      store the intersection of the planes as SymmetryAxis
7:      rotate the points in the PCD by 120° and 240° about the
          SymmetryAxis
8:      for each point in PCD, do
9:          compute the 3D Euclidean distance to the nearest 120° and 240°
              rotated PCD
10:         if both distances > saThreshold, then
11:             store point as PlanarSymmetryDeviationPoint
12:         end if
13:     end for
14:  end if
```

Algorithm 4.4.3.3: Patterned Feature Deviation

Input: *PCD*

Output: *PatternedFeatureDeviation*

- 1: search in the *PCD* for keypoints the 2D Scale-Invariant Feature Transform (SIFT) algorithm by Lowe [203]
 - 2: compute the 3D descriptor for each keypoint using the Signature of Histograms of Orientations (SHOT) by Tombari, Salti [205]
 - 3: **while** there are keypoints still available to check, **do**
 - 4: search for Bounding Sphere Nearest Neighbours keypoints using Algorithm 4.3.1b
 - 5: compare keypoints in *PCD* with *BoundingSphereNearestNeighbours*
 - 6: **if** a positive correspondence, or match, is found, **then**
 - 7: align *BoundingSphereNearestNeighbours* with matching *PCD* points, using the Iterative Closest Point (ICP) algorithm by Chen and Bhanu [206]
 - 8: compare the *BoundingSphereNearestNeighbours* against the matching *PCD* points
 - 9: store any deviation as *PatternedFeatureDeviation*
 - 10: **end if**
 - 11: **end while**
-