

**Towards an Autonomous Robot-based Laser Cladding Repair  
Process: A Framework for Damage Detection, Localization and Path  
Planning**

by

Habiba Zahir Imam

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

Master of Science

in

Engineering Management

Department of Mechanical Engineering

University of Alberta

© Habiba Zahir Imam, 2021

## **Abstract**

With piling scientific evidence and growing public concerns about climate change and depletion of natural resources, policymakers are being forced to implement stringent environmental regulations. One such sector under scrutiny for the concerning pace at which it is consuming natural resources is the manufacturing sector. Repair and remanufacturing are deemed sustainable approaches due to their capability of restoring value in a damaged component and bringing it to like-new condition. However, in contrast to a manufacturing process benefiting from an automated environment, the automation level for repair and remanufacturing processes remains low. Moreover, the traditional repair process is tedious and time-consuming. This is mainly due to the stochastic return of used parts, making this process difficult to automate.

With the aim of moving the repair industry towards autonomy, this study proposes a novel repair framework. The developed methodology presents a vision-based Robotic Laser Cladding Repair Cell (RLCRC) that has three features: (a) an intelligent inspection system that uses a deep learning model to automatically detect the damaged region in an image; (b) employing computer vision-based calibration techniques for converting damaged region in pixels to spatial coordinates and extracting the damaged volume; (c) generating a tool-path for depositing material to repair the worn component. In this research, the repair of fixed bends is selected as the case study. Fixed bends are cylindrical components used in directional drilling and are present in copious amounts in the oil and gas sector.

The proposed RLCRC employs visual sensors (camera and time-of-flight sensor) to provide automatic and time-efficient damage detection and localization. At first, the performance of different deep learning models utilizing varying datasets is compared to obtain a model best suited for being implemented in the RLCRC. Captured images are analyzed by the selected model for the presence of damage. If damage is found, the model classifies and encloses the region of interest in a bounding box. Then, an algorithm is

developed that leverages the pinhole camera calibration technique to localize the damage location spatially. By sending this location to a Time-of-Flight (ToF) sensor, the three-dimensional point cloud data containing the damage volume is acquired. Finally, a simplified tool-path generation method is explored that leverages off the polar coordinate system for depositing material in the damaged cavity to repair the component. Supported by case studies, the results obtained herein validate the efficacy of the proposed framework. Thereby enabling automatic damage detection and damaged volume extraction for worn fixed bends. Following the suggested framework, a time reduction of more than 63% is reported.

# Preface

This thesis is an original work by Habiba Zahir Imam. Two journal papers and one conference paper related to this thesis have been submitted or published and are listed below. As such, the thesis is organized by following the paper-based thesis guideline.

1. H. Imam, Y. Zheng and R. Ahmad\*, "An efficient tool-path planning approach for repair of cylindrical components via laser cladding," *Journal of Remanufacturing*, 2020. DOI: <https://doi.org/10.1007/s13243-020-00096-6>.
2. H. Imam, Y. Zheng, P. Martinez and R. Ahmad\*, "Vision-based damage localization method for an autonomous robotic laser cladding process," *Procedia CIRP* (under review)
3. H. Imam, H. Al-Musaibeli and R. Ahmad\*, "A computer vision-based spatial damage localization method for an autonomous robotic laser cladding repair process," *Robotics and Computer-Integrated Manufacturing* (submitted)

*“No two things have been combined better than knowledge and patience.”*

-Prophet Muhammad (PBUH)

## **Acknowledgments**

At the outset, I would like to thank my supervisor, Dr. Rafiq Ahmad, for his support, patience, and opportunity to work on this exciting project. I am also grateful to the guidance provided by my friend and colleague, Dr. Yufan Zheng, his co-supervision helped bring this research to fruition. In addition, I sincerely appreciate the discussions with my friend and colleague Dr. Pablo Martinez, nudging me to look at problems from different angles.

I would also like to express my sincere gratitude to our industry partner, Group Six Technologies Inc., for providing me with data, particularly Matt Dewar, for his unwavering support, friendship, cooperation, and accommodation for my experiments during COVID-19. I sincerely hope the findings from this research can be of value to them. As well, Hamdan Al-Musaibeli, who was instrumental in collecting data and conducting experiments on the vision system.

Social support is especially important during graduate studies and I want to acknowledge my friends and colleagues in the office who have made my MSc journey a memorable experience: Alireza Gharahi, Mohammad Alghamdy, Feiyu Ja, Mario Soriano, David Baca and Abraham Reyes. To my main support systems, Matti, Hina, and Salma, for always being there for me and quite literally being my biggest cheerleaders.

Finally, to my parents, for all their selfless love, prayers, and words of encouragement in every step of my life. I owe everything that I am and will ever accomplish to them.

# Table of Contents

Chapter 1: Introduction .....	1
1.1 Background .....	1
1.2 Motivation .....	3
1.3 Thesis Objectives .....	6
1.4 Methodology .....	7
1.5 Thesis Outline.....	9
Chapter 2: Vision-Based Damage Detection and Identification Method for an Autonomous Robotic Laser Cladding Process .....	11
2.1 Introduction .....	11
2.2 Methodology for Damage Detection and Identification.....	13
2.2.1 Vision-based RLCRC.....	14
2.2.2 Configuration of the Object Detection Model .....	15
2.3 Results and Analysis .....	18
2.3.1 Case Study 1: Damage Detection Method Implementation.....	19
2.3.2 Case Study 2: Improvement of Damage Detection Method .....	23
2.4 Discussion and Limitation.....	27
2.5 Conclusion.....	28
Chapter 3: A Computer Vision-based Spatial Damage Localization Method for an Autonomous Robotic Laser Cladding Repair Process.....	30
3.1 Introduction .....	30

3.2 Methodology for Damage Localization.....	34
3.2.1 Hybrid Localization Framework.....	34
3.2.2 Autonomous Robotic Laser Cladding Repair Cell .....	36
3.2.3 Spatial Localization.....	39
3.3 Validation of the Setup.....	46
3.4 Case Study Results .....	49
3.4.1 Case Study 1: Implementation of Damage Localization Method .....	49
3.4.2 Case Study 2: Implementation of Damage Localization Method .....	51
3.5 Discussion and Conclusion .....	53
Chapter 4: An Efficient Tool-Path Planning Approach for Repair of Cylindrical Components via Laser Cladding.....	55
4.1 Introduction .....	55
4.2 Methodology for Tool-Path Generation.....	57
4.2.1 Point Cloud Acquisition and Analysis .....	57
4.2.2 Tool-Path Generation.....	60
4.3 Results and Discussion.....	62
4.4 Conclusion.....	66
Chapter 5: Conclusion, Discussion & Future Work.....	68
5.1 Conclusions .....	68
5.2 Research Contributions .....	69
5.3 Limitations and Future Work .....	71



Bibliography.....77

## List of Tables

Table 1.1: Environmental and economic impact of remanufacturing and repair relative to manufacturing a new version of a product [4].	2
Table 2.1: Different architectures used in literature for damage localization.	17
Table 2.2: Benchmark mAP score values of a Faster R-CNN model trained on COCO dataset with ResNet50 and Inceptionv2.	19
Table 2.3: Hyperparameters for training the model.	21
Table 2.4: Comparative analysis of the architectures trained with two labels.	21
Table 2.5: Comparative analysis of the architectures trained with one label.	22
Table 2.6: Hyperparameters for improving the training of the model.	25
Table 2.7: Comparative analysis of the architectures trained on the new dataset with one label.	26
Table 3.1: Calibration Parameters.	42
Table 3.2: Parameters to consider for getting the angle of wear.	43
Table 3.3: Validation of the calibration setup using tape markers.	48
Table 3.4: Analysis of results from case study 1.	50
Table 3.5: Analysis of results from case study 2.	52
Table 4.1: Algorithm for tool-path generation.	62

# List of Figures

Figure 1.1: Concept of a linear economy model vs a circular economy model.....	1
Figure 1.2: Flowchart outlining a traditional repair process.....	4
Figure 1.3: Side view and isometric view of a sample fixed bend with length and diameter dimensions in mm.....	6
Figure 1.4: Methodology and objectives within the proposed repair framework for damaged fixed bends.....	8
Figure 2.1: Setup of the RLCRC.....	15
Figure 2.2: The schematic architecture of the Faster R-CNN used .....	16
Figure 2.3: Sample augmented images from training dataset, with pad and damage region annotated.....	20
Figure 2.4: Training and validation losses vs number of steps.....	22
Figure 2.5: Testing dataset with bounding box output. Trained with (a,b) two labels ‘pad’ and ‘damage’; (c,d) one label ‘pad’.....	23
Figure 2.6: Sample augmented images from the new training dataset containing only images with the same camera setting.....	25
Figure 2.7: Resulting metric plots showing (a) $mAP @ 0.5:0.95 IOU$ & (b) $mAP @ 0.5 IOU$ ; (c) validation loss and (d) training loss.....	26
Figure 3.1: Hybrid damage localization pipeline.....	35
Figure 3.2: Side view and axial view demonstrating the length ( $l_1, l_2$ ) and angle ( $\theta_l, \theta_t$ ) of wear respectively .....	36
Figure 3.3: Setup of the RLCRC.....	37
Figure 3.4: Camera offset (in mm) from the global coordinate origin .....	38
Figure 3.5: Perspective of the damaged part from the camera .....	39
Figure 3.6: Reprojection error output from the calibration code for each of the calibration images .....	41
Figure 3.7: Camera parameters output from the calibration code for each of the calibration images .....	42
Figure 3.8: Extracting angle of wear from video based on frame number .....	45
Figure 3.9: Three tape markers with known edge coordinates for calibration validation .....	47
Figure 3.10: Error tolerance in z to account for the calibration error .....	49
Figure 3.11: Fixed bend for case study 1, with the pad and axes annotated.....	50

Figure 3.12: Point cloud representing the damaged component for case study 1.....	51
Figure 3.13: Fixed bend for case study 2, with the pad and axes annotated.....	52
Figure 3.14: Point cloud representing the damaged component for case study 2.....	53
Figure 4.1: Representation in a) Polar coordinate system b) cartesian coordinate system .....	58
Figure 4.2: Flowchart comparing the existing approach to remanufacturing with the proposed method.....	59
Figure 4.3: Setup of the RLCS in RoboDk software .....	61
Figure 4.4: Setup of the robot with the scanner .....	63
Figure 4.5: Input: a) extracted damaged point cloud b) preprocessed damaged point cloud .....	64
Figure 4.6: Red lines indicate the generated tool-path a) without and b) with surface function interpolation.....	65
Figure 4.7: a) Before repair and b) after repair carried out by the proposed algorithm....	66
Figure 5.1: Proposed repair framework with future work plan .....	76

## List of Abbreviations

VRPs	Value-Retention Processes
ToF	Time-of-Flight
CAD	Computer-Aided Design
LC	Laser Cladding
LMD	Laser Metal Deposition
RLCRC	Robotic Laser Cladding Repair Cell
CNN	Convolutional Neural Network
R-CNN	Region-based Convolutional Neural Network
ResNet	Residual Network
YOLO	You Only Look Once
SSD	Single Shot multi-box Detector
RPN	Region Proposal Network
ROI	Region Of Interest
FC	Fully Connected
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
COCO	Common Objects in Context

<i>mAP</i>	Mean Average Precision
<i>AP</i>	Average Precision
<i>IoU</i>	Intersection-over-Union
RGBD	Red, Green, Blue-Depth
LIDAR	Light Detection and Ranging
SLAM	Simultaneous Localization and Mapping
DED	Direct Energy Deposition
RE	Reverse Engineering
RANSAC	Random Sample Consensus
NURBS	Non-Uniform Rational B-Splines
STL	Stereolithography
MFT	Mesh Following Technique
pcd	Point Cloud File
RLCS	Robot Laser Cladding System

# Chapter 1: Introduction

## 1.1 Background

Repair or Remanufacturing can simply be defined as the process of restoring value in a damaged component with the goal of reusing it. Sustainable manufacturing can be looked at as a circular economy model where a new component is designed, manufactured and used up until it reaches its end-of-life state. Next, instead of discarding the part, it is repaired and brought back to like-new condition. Circular economy is a paradigm shift from traditional linear economy models (see Figure 1.1), which is a ‘take, make and waste’ approach of manufacturing [1]. With rising climate change concerns, governments and industries are being forced to ‘re-think’ society's current operability. In particular, there has been a shift in public awareness from being uninformed or neglectful of natural resource exploitation by industries to outright protesting for their protection. Amidst the growing public pressure, policymakers and businesses are left scrambling to adopt sustainable manufacturing practices.

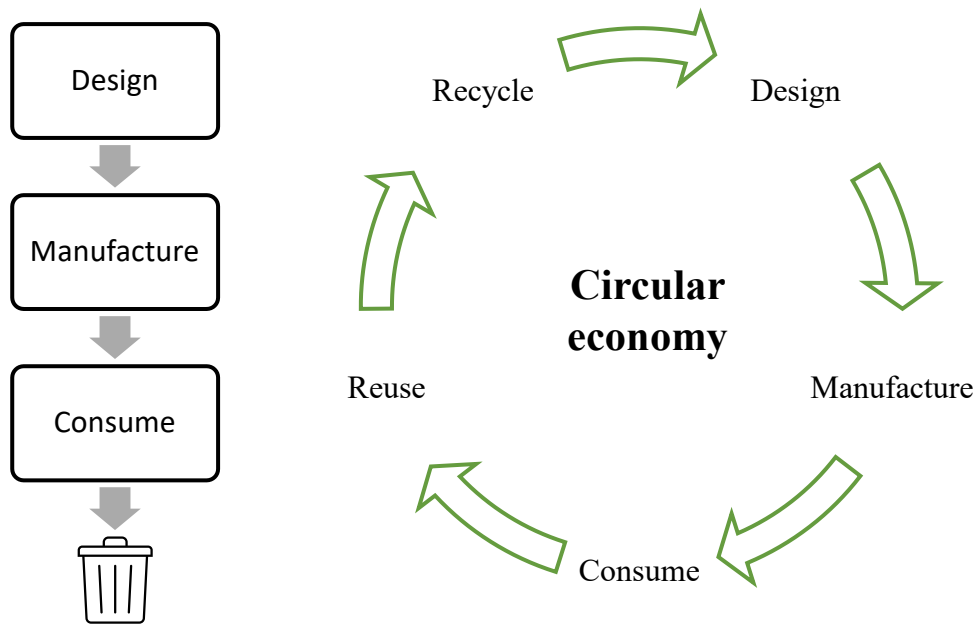


Figure 1.1: Concept of a linear economy model vs a circular economy model

The concept of remanufacturing at an industrial level first emerged during World War I for tank remanufacturing [2]. It was brought to the fore during World War II, when manufacturing industries' focus drastically shifted from everyday production to military production, taking a toll on the amounts of labor and material available to citizens [3]. Scarcity in resources led to a rampant remanufacturing and reuse of common parts with the aim to keep society functioning. Since then, technological advances have led to great strides being made in the remanufacturing industry.

It is important to note that repair and remanufacturing are commonly used interchangeably, even though there are some differences between the two. Remanufacturing is a standardized industrial process where damaged parts are brought back to a similar or higher quality as compared to their manufactured counterparts. Remanufacturing can also add new functionalities to existing parts. In comparison, repair restores the worn component to a state where it can fully function for its initially intended purpose. The United Nations environment program published a report characterizing repair and remanufacturing collectively as Value-Retention Processes (VRPs); remanufacturing being a full-service life VRP and repair being a partial service life VRP [4]. Some noteworthy statistics extracted from the report are shown in Table 1.1.

Table 1.1: Environmental and economic impact of remanufacturing and repair relative to manufacturing a new version of a product [4].

	<b>Remanufacturing vs Manufacturing</b>	<b>Repair vs Manufacturing</b>
Environmental	Reduced production waste by 90%	Reduced production waste by 97%
	Reduced energy consumption by 79%	Reduced energy consumption by 93%
	Reduced emissions by 89%	Reduced emissions by 96%
Economic	44% cost savings	Up to 95% cost savings
	Increased employment opportunity up to 120%	Lowered employment opportunity by 70%



The product recovery process that is the focus of this thesis aims to restore the original functionality of damaged parts, and therefore falls into the repair category. Nonetheless, the terms repair and remanufacturing will be used interchangeably unless otherwise specified.

## **1.2 Motivation**

Globally, the United States constitutes the largest portion of remanufacturing output value, leading with an annual output worth of \$75 billion, followed by \$7.28 billion in the United Kingdom and \$4.8 billion in Japan [5]. However, the remanufacturing industry only represents 2% of all manufacturing in the United States and 1.9% in the European Union [6]. This low number is due to the unstable nature of a remanufacturing process along with unpredictable quality, quantity and timing of returned worn parts as compared to a well-established manufacturing framework [7].

During remanufacturing, a used product is first disassembled, the separate parts are inspected for wear and subsequently repaired. Finally, the repaired components are assembled back into a remanufactured product. The specific technology used for repairing is highly variable and is dependent on the type and purpose of remanufacturing. Based on the mechanical components, industries and researchers specialize in the repair of specific components, most commonly aircraft components [8] and turbine blades [9].

Basic primitives such as planes, cylinders, spheres and cones make up most of the manufactured objects [10]. Amongst these objects, cylinders are the most widely produced primitive for their usage in pipelines, drill bits, gas cylinders, sleeves, connectors, etc. [11]. A study found that repairing cylinder heads by laser cladding cuts environmental impact by 63.8% as compared to manufacturing [12].

The fourth industrial revolution or Industry 4.0 is fueled by data and machine learning and is causing a stir in the manufacturing industry. Promising to optimize production and increase efficiency, businesses are widely adopting this new wave of technology in fear of being left behind by the competition. However, the current state of repair is struggling to

keep up with industry 4.0 due to its excessive reliance on a human operator for damage detection and localization. The existing repair methods are “inefficient, experience-based and error-prone”[9].

A traditional repair process generally follows six steps as outlined in Figure 1.2. The first step is acquiring three-dimensional data of the entire surface of a damaged component. Quite often the geometric information of the original part, the so-called nominal model, is missing, leading to the reconstruction of the nominal model geometry from the damaged model. Registration of the nominal and damaged models is carried out, in which both models are aligned to extract the repair patch. The information from this patch is used to generate a tool-path for material deposition.

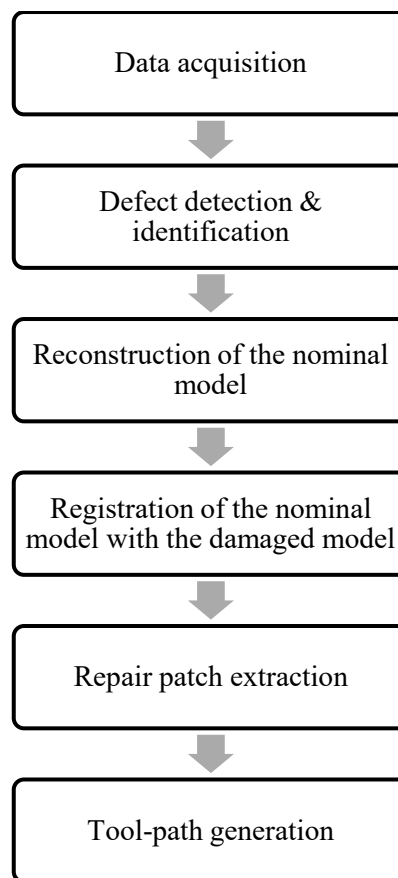


Figure 1.2: Flowchart outlining a traditional repair process

For primitive shapes like cylinders with a known nominal radius, the traditional repair framework can be modified by eliminating time-consuming and labor-intensive steps like three-dimensional geometry acquisition of the entire damaged surface, reconstruction of the nominal model geometry and registration of the nominal model with the damaged model. Damaged volume extraction is the most important step in repair as it is the basis on which the tool-path is generated. Moreover, to resolve the aforementioned issues of traditional repair processes and to bring the benefits of automation to the manufacturing industry, it is important to integrate computer vision-based deep learning techniques into the field of repair.

In order to address the challenges mentioned above, this thesis aims to introduce a novel framework for repairing cylindrical components. As such, an intelligent, automatic and fast repair solution is explored. The specific type of cylindrical component that is analyzed in this thesis is a fixed bend (see Figure 1.3) which is a kind of bent housing used in the oil and gas industry. As the name suggests, a bent housing has a ‘bend’ in it and is employed in directional and horizontal drilling for its advanced deflection capability [13]. A sample fixed bend is demonstrated in Figure 1.3, with the pad on it annotated. The pad has a length of 144mm, a diameter of 134mm and an angle of 120° around the fixed bend surface. It can be seen clearly that the pad extends out as compared to the rest of the bend. During a drilling job, as the pad is in direct contact with the external environment, it is the area that incurs the most wear over time. For context, manufacturing a fixed bend with a 165mm diameter typically costs 3,800 CAD but repairing costs 600 CAD. Repair costs can range from anywhere between \$500-\$700 depending on the depth of wear on the pad because the more the wear, the more the material that needs to be deposited. With a cost reduction of more than 84%, it is worthwhile to explore efficient repair strategies for fixed bends.

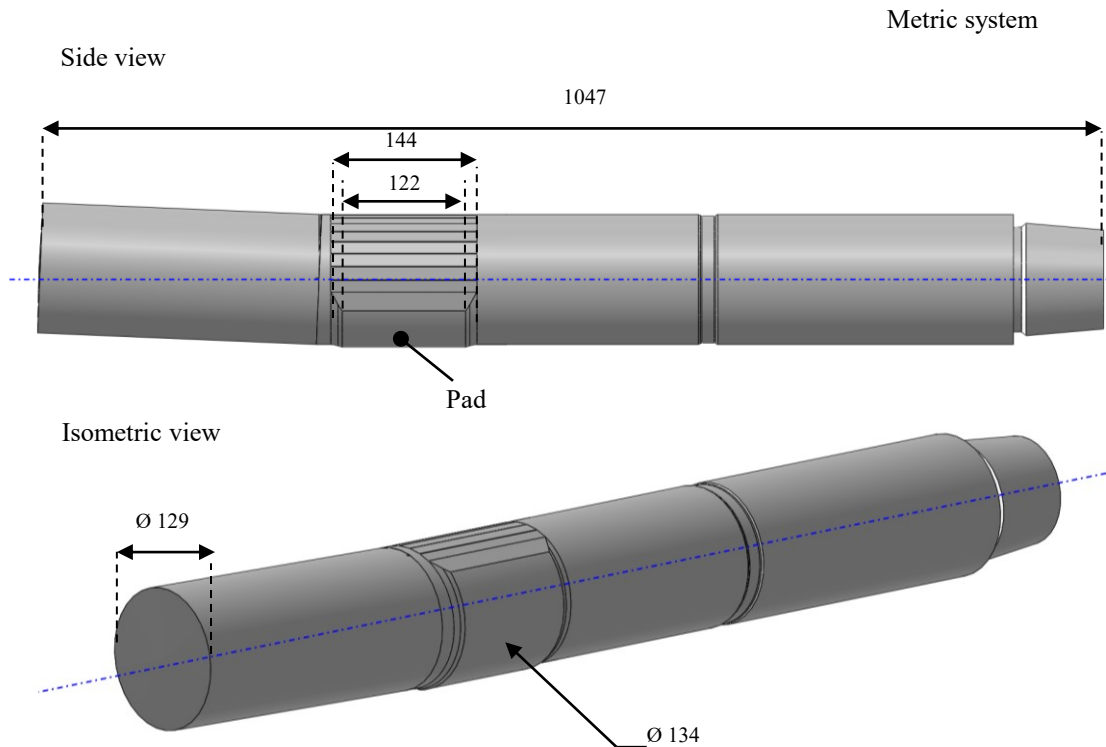


Figure 1.3: Side view and isometric view of a sample fixed bend with length and diameter dimensions in mm

### 1.3 Thesis Objectives

The main objective of this research is to develop an intelligent, automatic and speedy repair framework for damaged fixed bends by the use of a robotic laser cladding system.

As such, the objectives of this research include the following tasks (**O<sub>x</sub>**):

- **O<sub>1</sub>**: Analyze the effect of a training dataset on a damage detection model by training various models with varying datasets.
- **O<sub>2</sub>**: Develop a method for localization of damaged areas on part surfaces in 3D space based on two-dimensional image data.
- **O<sub>3</sub>**: Provide an efficient tool-path generation approach for the repair of cylindrical components.


## 1.4 Methodology


By integrating industry 4.0 technologies, the proposed methodology aims to move the repair industry towards full autonomy. The first objective stated in Section 1.3 is realized by proposing a novel repair framework (see Figure 1.4) for damaged fixed bends. This framework takes the damaged cylindrical component as an input and outputs a repaired component without the need of acquiring the entire damaged surface and excludes time-consuming and labor-intensive processing steps like damage reconstruction and registration.

As a prerequisite for damage detection and damage localization, an intelligent experimental setup is designed that integrates deep learning and a computer vision system to automatically detect and localize the worn region on a damaged component. Images of the damaged component are captured by the vision system and inspected by the deep learning model. Once the model detects damage in an image, it outputs the 2D location of the damaged region in pixel coordinates. By incorporating calibration techniques, the vision system then spatially localizes the damage and outputs the 3D world coordinates of the damaged region. With three-dimensional information of the damaged area, a time-of-flight sensor (ToF) is used to capture the repair patch's geometry. Based on the acquired repair volume, a tool-path is generated to deposit material in the worn region. The proposed repair solution for fixed bends carries out automatic damage detection, localization, scanning and tool-path generation, making the process more reliable, cost and time-effective.

Legend

 Technologies

 Processes

 Input/Output

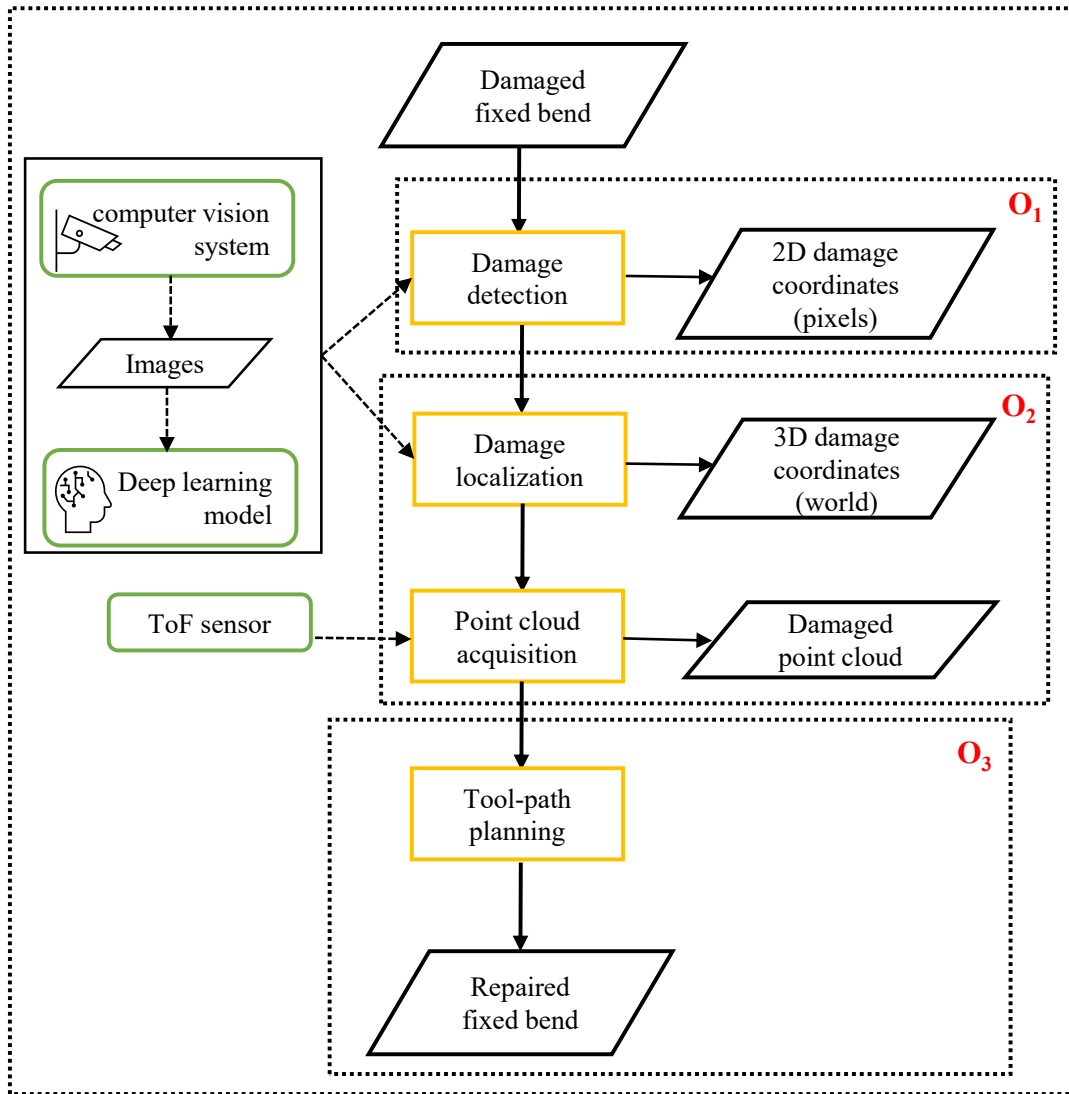


Figure 1.4: Methodology and objectives within the proposed repair framework for damaged fixed bends

## 1.5 Thesis Outline

This chapter provides a concise background on repair and remanufacturing, followed by the limitations in the automation of a repair process, and the motivations for undertaking this research. A brief statement on the objectives of this thesis and an overview of the proposed framework are also presented.

Chapter 2: presents a deep learning and computer vision-based damage inspection system capable of identifying and detecting damaged regions in images taken in a repair cell. Several different configurations of deep learning models are trained with varying datasets to obtain a model that is particularly suitable for damage detection on fixed bends. The output obtained from this chapter is the location of the damaged area in pixel coordinates ( $O_1$ ).

Chapter 3: proposes a calibrated computer vision-based damage localization robotic repair cell for detecting damage on the surface of a component by integrating the deep learning-based damage inspection system established in Chapter 2. The proposed cell takes pixel information provided from the deep learning model and converts it into three-dimensional data with respect to the world coordinate frame. The design of experiment and the validation of the system followed by two case studies are presented in this chapter. Point cloud data containing three-dimensional information of the damaged region is outputted from this chapter ( $O_2$ ).

Chapter 4: presents a straightforward method for tool-path generation directly from damaged point cloud data for worn cylindrical fixed bends. Regardless of the availability of the exact nominal computer-aided design (CAD) model, this approach creates a tool-path specifically for cylindrical components with a known build-up radius. Additionally, this method enhances the point cloud data's information density and discounts the tedious surface reconstruction and registration steps ( $O_3$ ).

Finally, Chapter 5: summarizes the work done in this thesis and the resulting conclusions. Furthermore, the limitations and future research directions which can extend the presented work are discussed.



# **Chapter 2: Vision-Based Damage Detection and Identification Method for an Autonomous Robotic Laser Cladding Process**

## **2.1 Introduction**

Laser Cladding (LC) or Laser-based Direct Metal Deposition (LMD) is an attractive additive manufacturing technique that has garnered considerable interest for mechanical engineering applications in the aerospace, oil and gas industry, etc. [14]. This well-established industrial process works by focusing a high-power laser beam to generate a molten pool on the substrate along with continuously directing material through a coaxial nozzle into that weld pool where it solidifies [15]. This layer-by-layer manufacturing technique can increase time and cost efficiency compared to conventional technologies like casting, forging, and machining [16].

For several decades, LMD is actively researched as an effective technology for repair and remanufacturing [17]. Repair or remanufacturing is credited for increasing the manufacturing sector's sustainability by bringing damaged or worn metal parts back to like-new conditions. The process generally involves identifying and locating damages on a part's surface and then depositing material to restore the original geometry. Today, innovators in the field of repair seek to develop strategies to boost the level of automation for repair and maintenance, thereby boosting the flexibility of the process [18]. It has been shown that when integrated with a robotic manipulation system LMD shows an increased geometric flexibility, accessibility and saves production time [19].

In robotic laser cladding applications, inspection of the worn area is currently a manual process. The damage is visually identified by an operator who then uses a laser scanner to capture the surface geometry of the defect [20], [21]. This defect geometry is used to generate a repair strategy for the part. The method of manual inspection brings its own set of challenges, e.g., being error-prone, time-consuming, costly and inefficient [22]. This is

especially problematic when the scale of the part increases. Due to a human operator's limitations, the more monotonous the task is, the more the chance of oversight. By taking a modern approach of integrating robotics with computer vision, an automatic inspection can be performed that can improve efficiency, accuracy and cost savings [23].

Computer vision is an interdisciplinary field that seeks to understand, automate, and replace human visual tasks in any working environment. Moving towards an autonomous Robotic Laser Cladding Repair Cell (RLCRC), a significant amount of research is being done using Artificial Intelligence (AI), more specifically supervised learning methods such as deep learning to inspect the laser welding process [24], [25]. Convolutional Neural Networks (CNN) are being trained to monitor and identify weld defects and melt pools during the laser cladding procedure [26], [27]. Region-based Convolutional Neural Network (R-CNN) is a deep learning object detection approach where the R stands for regions of interest in an image. R-CNN first generates region proposals then uses CNN to extract features, locate and classify objects. Computer vision techniques paired with R-CNN's are being extensively adopted across many disciplines such as construction, transportation, materials science, geoscience and food production for automatic object detection and classification [28], [29]. They have also made their way into manufacturing for damage detection and classification [30], [31]. Additionally, intelligent vision-based practices are being implemented on shop floors for classifying and automating the repair inspection process [32].

R-CNN's are deep and complex networks that require a significant amount of time and data to reach desirable results [33]. Transfer learning is a promising learning framework that essentially transfers knowledge learned in a previous task to a novel task; proven to save time and give effective results when data is scarce [34].

Faster R-CNN has a notably speedier object detection time compared with previous image classification and object detection models [35]. It was developed to function closest to real-time, reaching ten times the speed of Fast R-CNN [36]. You Only Look Once

(YOLO) and Single Shot multi-box Detector (SSD) Mobilenet have a higher detection speed than Faster R-CNN but a notably lower accuracy [37].

Based on the above literature review, it is evident that computer vision-based deep learning techniques have a lot of potential and have shown promising results across many disciplines. Despite that, remanufacturing remains to be heavily reliant on human intervention for damage detection and localization. There is immense scope for these intelligent strategies to be used to automate the damage detection process in a real RLCRC. This would be incumbent in achieving a fully autonomous repair system. With recent advances in computer vision and the availability of abundant data, it is economically worthwhile to explore the use of this technology in an RLCRC. This study first proposes an integration of a vision sensor in a repair cell to record image data of damaged components. Then, two case studies are carried out utilizing two different datasets. These case studies perform analyses to compare the viability, accuracy and time efficiency of popular feature extractors for damage detection purposes. Finally, based on the results, the most fitting model configuration is selected and the results and evaluation are presented.

The chapter is organized as follows: Section 2.2 explains the methodology utilized in this study, including the setup of the vision-based RLCRC and the different object detection model configurations that are considered. Section 2.3 presents two case studies with their respective datasets and comparative analysis results. This is followed by the discussion and limitations. Finally, the conclusions drawn from the study are discussed.

## **2.2 Methodology for Damage Detection and Identification**

Due to the reasons mentioned in Section 1.2, this study focuses on the detection and identification of damaged areas on cylindrical fixed bends (see Figure 1.3). More specifically, the *'pad'* on fixed bends and the *'damage'* present on the pad (see Figure 2.3). The pad is wrapped around the cylindrical surface of the fixed bend. For worn fixed bends, it is essential to distinguish the location of the pad, as it is the area that mostly incurs

damage and must be repaired. The damage present on the pad is usually of irregular geometry.

It is important to note here that this method must work hand in hand with a depth sensor to obtain an accurate volumetric representation of the damage. The results from this chapter serve as an input for a spatial damage localization method described in Chapter 3:.

### **2.2.1 Vision-based RLCRC**

The robot arm used is of the specification Fanuc-R-1000iA/80F, which is a high-speed handling robot for medium payloads. A camera of the model UVC-G3-Bullet/UVC-G3-AF is mounted on a laser head. A laser head provides a focused laser beam and the filler material for deposition during a laser cladding process. The laser head is attached to the robot arm for stability in maneuvering. The camera is mounted such that it is parallel to the robot arm and the laser head and facing down at a right angle to the part's surface. A schematic of the setup of the cell is demonstrated in Figure 2.1.

During the inspection, the camera view of the damaged part remains unchanged. This is done by mounting the cylindrical fixed bend on a turntable and moving the robot arm to a fixed position. Noting the pose of the robot at this position, ensures consistency in every damaged fixed bend the vision-based RLCRC inspects.

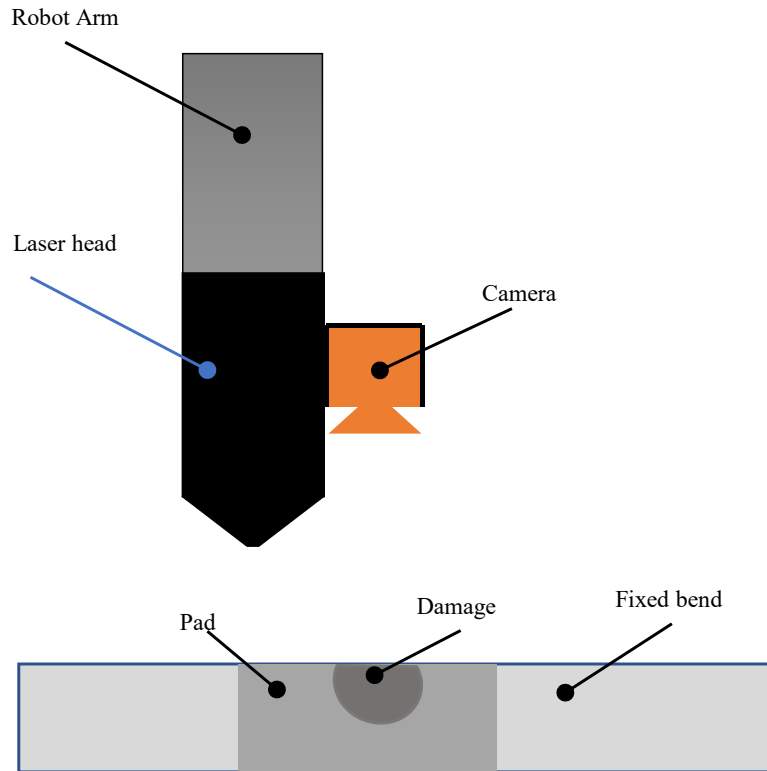


Figure 2.1: Setup of the RLCRC

## 2.2.2 Configuration of the Object Detection Model

As established in Section 2.1, the aim of this chapter is to select the best configuration of a model for damage detection that functions closest to real-time. Therefore, this study utilizes Faster R-CNN for damage detection and localization.

Faster R-CNN is an object detection architecture that comprises a feature extraction network, a Region Proposal Network (RPN) and a Region of Interest (ROI) network (see Figure 2.2). The function of a Faster R-CNN model is fourfold: 1) the pre-processed images go through a pre-trained CNN (e.g., ResNet, Inception) to extract features and acquire a feature map; 2) the RPN generates possible regions of interest in the feature map; 3) the ROI pooling extracts a feature vector of a fixed size from the feature map; and 4)

the fixed size feature map then goes through fully connected (FC) layers that predict the class label (classification) and the bounding box (regression) for each ROI. Depending on the particular configuration of a Faster R-CNN model, i.e., the type of CNN being used, there is a trade-off between the precision with which it detects an object and the total computing time it requires [38]. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC), initiated in 2010, is an annual challenge that allows more than fifty institutions to categorically develop and compare object detection algorithms [39]. Partaking in that competition, Inception (initially known as GoogleNet [40]) and ResNet [41] won first place in the years 2014 and 2015, respectively [39].

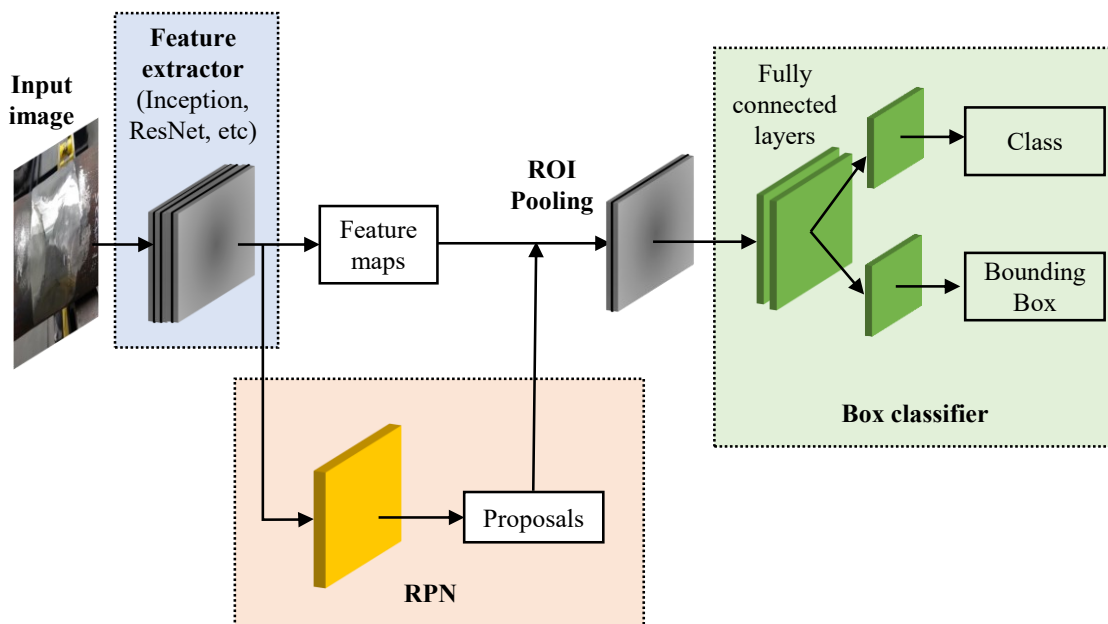


Figure 2.2: The schematic architecture of the Faster R-CNN used

The open-source TensorFlow object detection API is used with TensorFlow version 1.5. Applying the concept of transfer learning, the network is initialized with weights pre-trained on the Common Objects in Context (COCO) [42] dataset and present in the TensorFlow model zoo. This study analyzes Faster R-CNN's performance with Inception and ResNet, due to the superior results achieved by these feature extractors in the software

contest (ILSVRC). Table 2.1 highlights the specific version of these architectures together with the reference for where they are used in literature and their objective.

Table 2.1: Different architectures used in literature for damage localization

Meta-architecture	Feature extractor	Objective	Reference
Faster R-CNN	ResNet50	Crack detection	[43]
Faster R-CNN	Inception-V2	Defect detection	[44]

The trained model is evaluated on the validation dataset by performing classification and regression. Classification is performed to check whether a pad or damage exist on fixed bend images. While regression evaluates the difference between the predicted location of the damage and the manually annotated locations. This study primarily evaluates Mean Average Precision ( $mAP$ ) or Average Precision ( $AP$ ) as opposed to other object proposal proxy metrics because it is a widely used metric for object detection [36].  $AP$  or  $mAP$  is commonly used interchangeably and is calculated by taking the area under the precision-recall curve.  $Precision$ ,  $Recall$  and  $mAP$  are calculated by Equations (2-1)-(2-3) as shown:

$$Precision = \frac{TP}{TP + FP} \quad (2-1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2-2)$$

$$mAP = \frac{1}{11} \sum_{Recall_i} Precision(Recall_i) \quad (2-3)$$

where  $TP$  indicates the number of true positives, i.e., the number of correctly detected damages over all images in the validation dataset;  $FP$  indicates the number of false positives, i.e., images incorrectly predicted as having damage;  $FN$  indicates the number of

false negatives, i.e., failing to predict damage in an image that has damage; *Recall* is segmented evenly into 11 parts where  $i = (0,0.1,0.2,\dots,0.9,1)$ .

The *mAP* score is computed over a predetermined Intersection-Over-Union (*IoU*) threshold. *IoU* is a measure of overlap between the predicted bounding box and the ground truth bounding box and is calculated by Equation (2-4).

$$IoU = \frac{B \cap B_{gt}}{B \cup B_{gt}} \quad (2-4)$$

where  $B$  represent the predicted bounding box and  $B_{gt}$  represent the ground truth bounding box.

A higher *IoU* score would mean the predicted bounding box is heavily overlapping with the ground truth bounding box. A lower score would mean there is minimal overlap between the predicted and ground truth boxes. Typically, a prediction is deemed *TP* if it has an  $IoU > 0.5$  or *FP* if it has an  $IoU < 0.5$ . However, the MS COCO dataset comprising 80 object categories uses a standard *mAP* metric which is evaluated on *mAP @ IoU=0.5:0.05:0.95*; that begins from  $IoU=0.5$  and goes to  $IoU=0.95$  with increments of 0.05. This means that an average of ten different *IoU* thresholds is computed to provide a single *mAP* value. This metric is publicly available and will be used to perform an analysis comparing the values obtained from our trained model with the benchmark listed from Microsoft (MS) COCO dataset [45].

In the remainder of this study, *mAP* values stand for *mAP* calculated at  $IoU=0.5:0.05:0.95$ , unless otherwise specified.

## 2.3 Results and Analysis

Experiments were carried out on Google Colaboratory (RAM~ 12.6 GB, GPU: Tesla K80,12 GB, Disk~ 33GB). To attain an optimal solution for the specific object detection problem at hand, two different case studies were performed. The dataset utilized for each case study and their results are presented in the following sub-sections. The feature



extractors were compared on their *mAP* and detection speed values. Publicly available *mAP* scores from the COCO dataset were used as a reference, which are outlined in Table 2.2 [45].

Table 2.2: Benchmark mAP score values of a Faster R-CNN model trained on COCO dataset with ResNet50 and Inceptionv2

Feature extractor	COCO mAP
ResNet50	30%
Inception v2	28%

### 2.3.1 Case Study 1: Damage Detection Method Implementation

#### 2.3.1.1 Dataset

To develop a database containing images of damaged fixed bends, 72 images (resolution: 1920 x 1080 pixels) of 8 different types of fixed bends were collected. R-CNN's require a massive amount of training data to generate a high-performing model. This can be a burdensome task as obtaining a large amount of data is expensive and often not readily accessible. To overcome this problem, data augmentation is a widely embraced practice. For this study, different types of geometric (horizontal flip and vertical flip) and photometric (grayscale, hue, and exposure) augmentation techniques were applied to render the training model more robust and resilient to lighting and camera setting changes. Moreover, the images were resized to 416 x 416 pixels to reduce the training time. Figure 2.3 shows sample images from the expanded training dataset, which was enlarged by augmentation from 72 to 221 images. With the location of the pad and damaged region annotated in red and black, respectively.

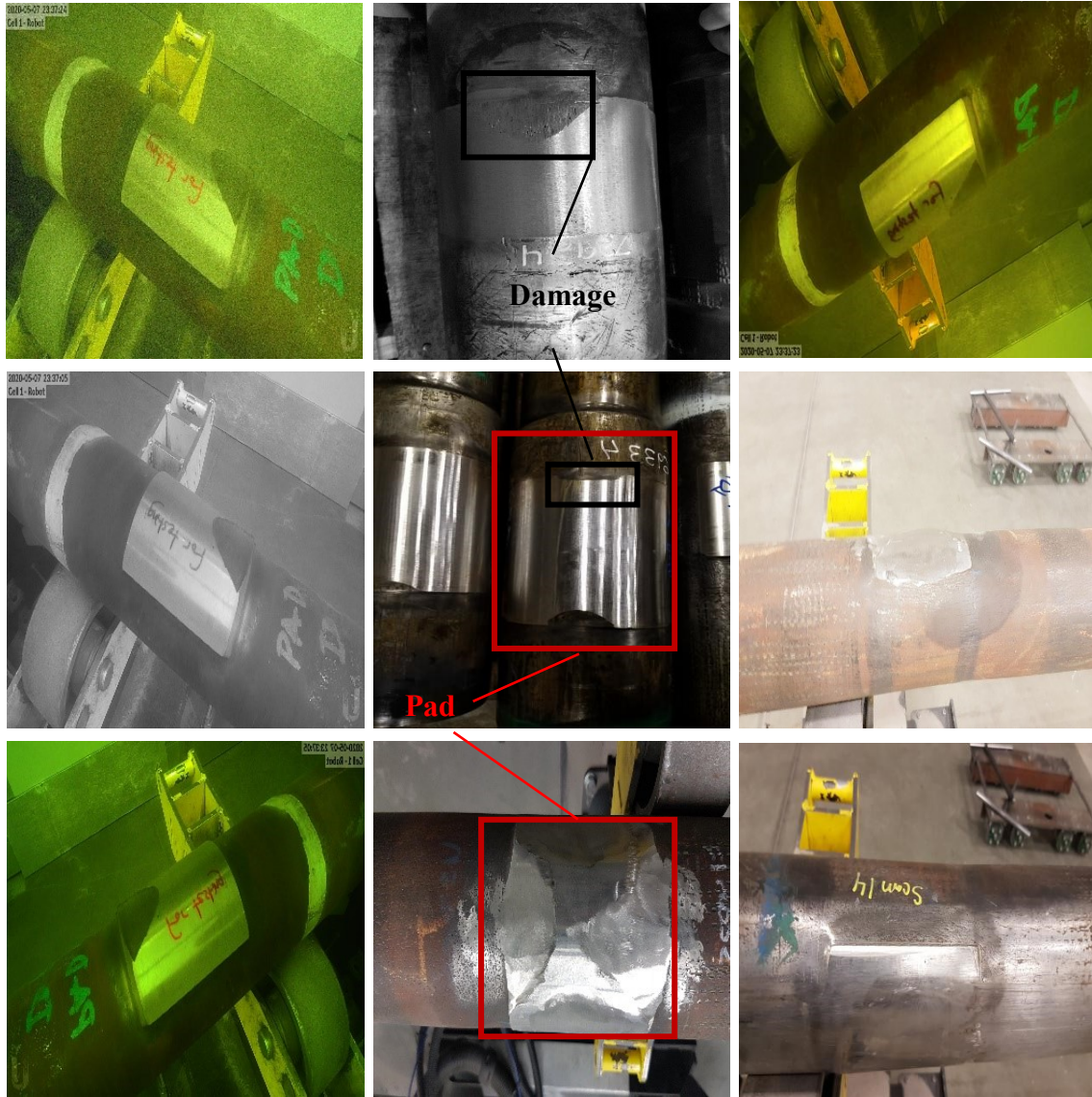


Figure 2.3: Sample augmented images from training dataset, with pad and damage region annotated

The extended dataset was annotated using labelling [46], a graphical image annotation tool. The two labels for classification were ‘*damage*’ and ‘*pad*’ on fixed bends. The dataset was then randomly split into 70%, 20% and 10% for training, validation and testing data, respectively.

Values for the hyperparameters in the feature extractor configuration pipeline were chosen by monitoring the progression of the  $mAP$  values and training and validation losses. The parameters were adjusted iteratively to minimize losses and maximize  $mAP$  values while keeping the duration of training manageable with the available computational resources. The set of values found to be optimal are given in Table 2.3.

Table 2.3: Hyperparameters for training the model

Num of steps	Batch size	Learning rate	Score threshold	Momentum optimizer
20,000	12	0.001	0.2	0.9

### 2.3.1.2 Comparative Analysis and Results

Additionally, time taken for inference per image also plays an important role when implementing it in a real-world scenario. Thereby, two feature extractors were compared on their  $mAP$  and detection speed values and their results are tabulated in Table 2.4, with the superior scores formatted in bold.

Table 2.4: Comparative analysis of the architectures trained with two labels

Feature extractor	‘Fixed Bends’ $mAP$	Detection speed (ms/image)
ResNet50	<b>52.8%</b>	1.48
Inception v2	49.1%	<b>1</b>

Compared to COCO (see Table 2.2), the results with the ‘Fixed Bends’ dataset are favorable, which is expected since COCO is a diverse dataset with 80 or more object categories. With only two categories (‘*pad*’ and ‘*damage*’), higher  $mAP$  values should be achievable.

From Figure 2.3, it is apparent that the two labels have similar features and a constant overlap in the images. These factors were hypothesized to create bias and variance in the model, resulting in the relatively low  $mAP$  scores. To investigate this hypothesis, the models were trained again but this time with one label (*'pad'*). For the reason established earlier in Section 2.2, that the most vital information when repairing fixed bends is the location of the pad. Table 2.5 outlines the resulting  $mAP$  and detection speed values output from the model when trained with one label and it is evident that the performance of the model has drastically improved.

Table 2.5: Comparative analysis of the architectures trained with one label

Feature extractor	'Fixed Bends' $mAP$	Detection speed (ms/image)
ResNet50	<b>70.4%</b>	1.48
Inception v2	60.8%	<b>1</b>

It is observed from the results provided above that ResNet50 outperforms Inception v2, for both trainings completed. Training and validation losses for ResNet50, trained for detecting one label, are presented in Figure 2.4. Inference was performed on models of ResNet50 with one label and two label configurations and the bounding box predictions are displayed in Figure 2.5.

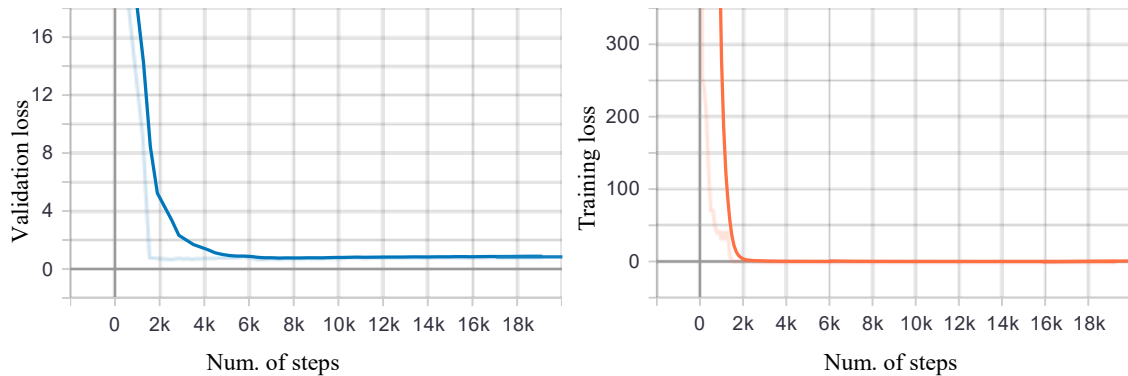


Figure 2.4: Training and validation losses vs number of steps

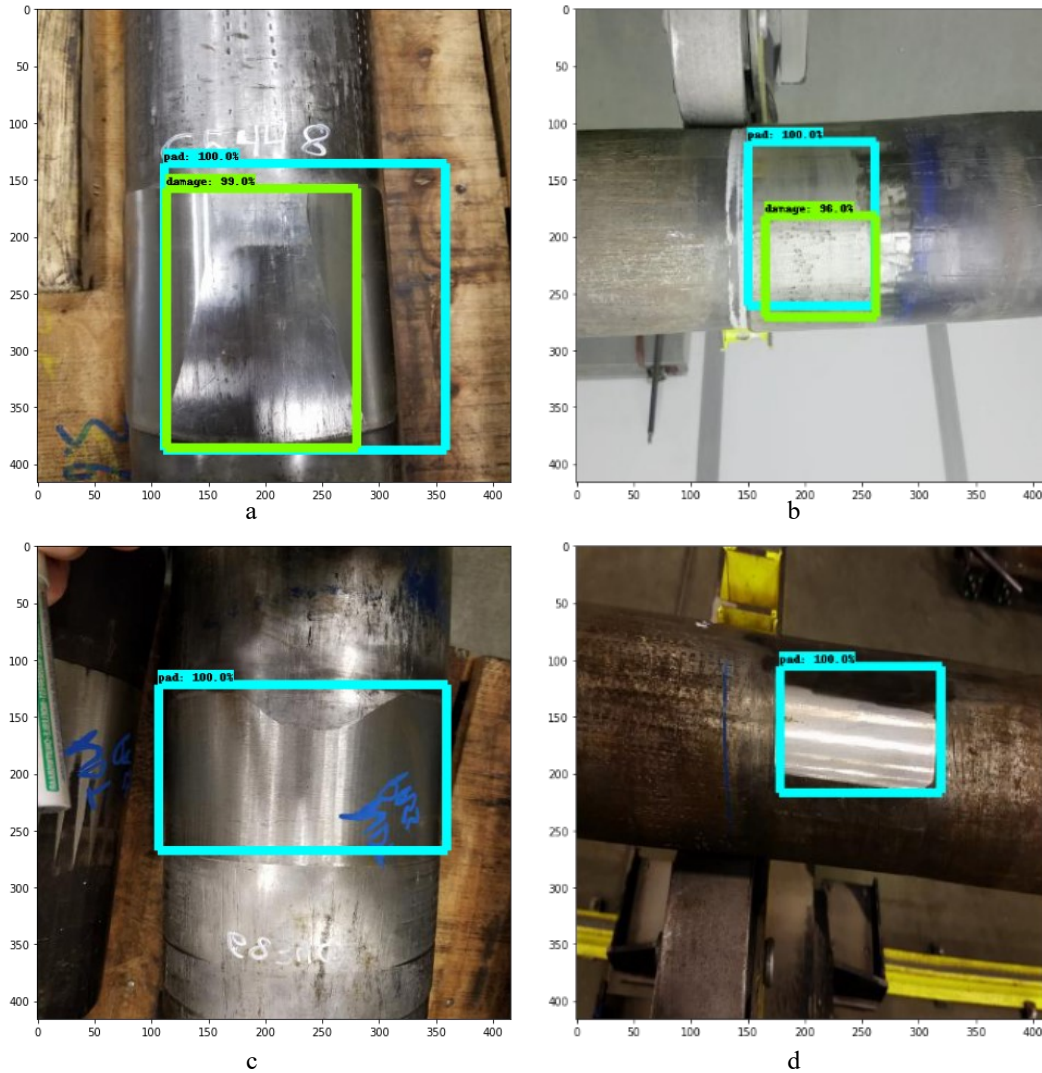


Figure 2.5: Testing dataset with bounding box output. Trained with (a,b) two labels '*pad*' and '*damage*'; (c,d) one label '*pad*'

## 2.3.2 Case Study 2: Improvement of Damage Detection Method

### 2.3.2.1 Dataset

For autonomous damage detection, the position of the camera in the RLCRC will remain unchanged, meaning the images from the camera of the workstation will always be taken

from the same setting. It was hypothesized that training the model with images of different fixed bends taken from the same position, will further improve the accuracy of the deep learning model. To this aim, a new dataset was formed containing images of similar orientation as those that the model will be expecting to see while it carries out damage detection.

A new dataset was formed that comprised 437 original images (resolution: 1920 x 1080 pixels) of four different fixed bends. Similar to the first dataset, the images were resized to 416 x 416 pixels. The images were annotated using `labellmg` [46], this time for one label '*pad*' because of the higher performance achieved using one label as observed in section 2.3.1.2. The dataset was then randomly split into 70%, 20% and 10% for training, validation and testing data, respectively. The following augmentation techniques were randomly applied to the images in the training dataset:

1. Crop, 0% min zoom; 20% max zoom
2. Hue, between -25% and +25%
3. Saturation, between -25% and +25%
4. Brightness, between -25% and +25%
5. Exposure, between -16% and +16%

Each image in the training set has 3 augmented versions, totaling the dataset to be consisting of 1049 images. Such that, the training, validation and testing dataset contained 918, 87 and 44 images, respectively. Figure 2.6 shows sample images from the training dataset.

Training and validation loss function values were closely monitored as training progressed in order to tune the parameters of the model. The learning rate was increased incrementally to obtain a trade-off value between the rate of convergence and over-shooting. Optimal results were found using similar hyperparameter values as in the first case study (section 2.3.1.2), except for the learning rate going up from 0.001 to 0.002, as shown in Table 2.6.

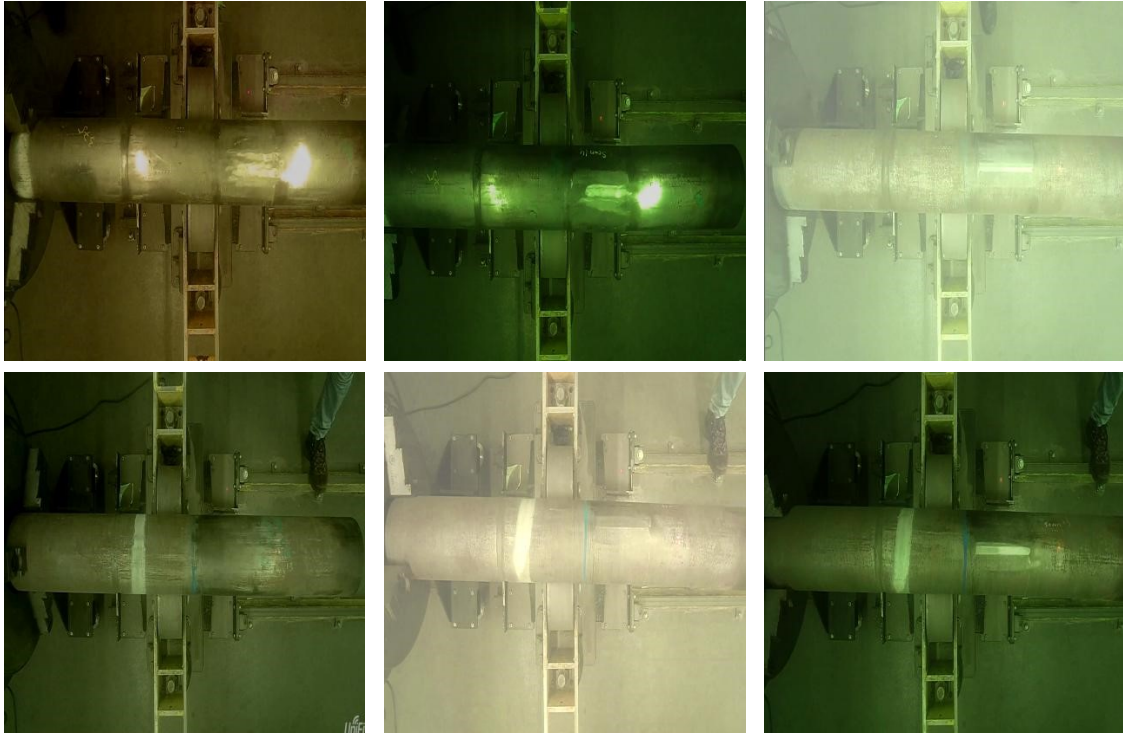


Figure 2.6: Sample augmented images from the new training dataset containing only images with the same camera setting

Table 2.6: Hyperparameters for improving the training of the model

Num of steps	Batch size	Learning rate	Score threshold	Momentum optimizer
20,000	12	0.002	0.2	0.9

### 2.3.2.2 Comparative Analysis and Results

Table 2.7 outlines the new results obtained from the model trained in case study 2, with the superior results formatted in bold. The new model significantly outperforms the model trained in case study 1. The highest achieved *mAP* score with the ResNet50 architecture

increases from 70.4 in case study 1 to 88.7 in case study 2. This practically represents an improvement in accuracy of 26%.

Table 2.7: Comparative analysis of the architectures trained on the new dataset with one label

Feature extractor	‘Fixed Bends’ $mAP$	Detection speed (ms/image)
ResNet50	<b>88.7%</b>	1.48
Inception v2	79.4%	<b>1</b>

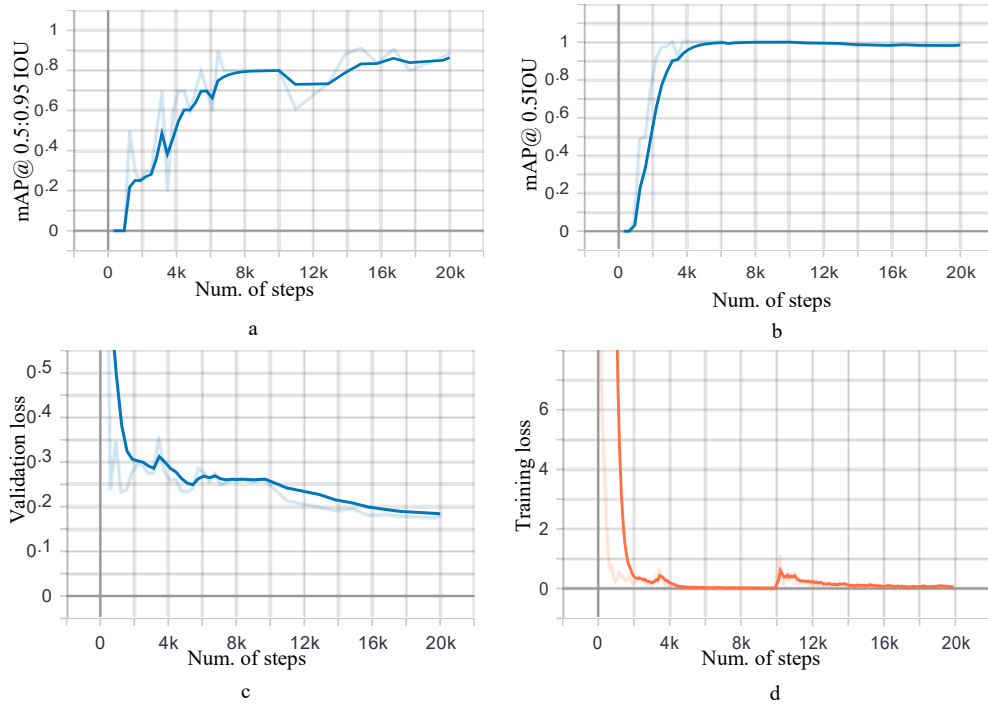


Figure 2.7: Resulting metric plots showing (a)  $mAP @ 0.5:0.95 IOU$  & (b)  $mAP @ 0.5 IOU$ ; (c) validation loss and (d) training loss



Figure 2.7 illustrates the resulting metric plots obtained from the ResNet50 model. The training and validation losses, as shown in Figure 2.7 (c, d), both decrease to a point of stability, which implies there is no overfitting. This model is assessed over several *IoU* metrics ( $IoU=0.50:0.05:0.95$ ), which means the model has to be performing well at every *IoU* threshold for it to achieve a high *mAP* score. Figure 2.7(b) shows the *mAP* value at 0.50 *IoU*, reaching 100%.

## 2.4 Discussion and Limitation

For case study 1, the model is being trained to detect pads and damages on fixed bends. Both classes have similar features and always overlap, which adds another level of complexity for the model. To mitigate this, the model is trained with only one label, '*pad*.' Doing so increases the *mAP* by 33%, from 52.8 to 70.4. but this comes at the cost of not detecting the damage and pad separately. Since pads' detection is more vital for fixed bends, this approach is considered the optimal choice.

Figure 2.5 shows that the prediction of pad locations is more accurate for one label as opposed to two. For example, Figure 2.5a shows how, when working with two labels, the pad's bounding box does not cover the full width of the pad as opposed to Figure 2.5c, where the pad width is entirely covered. It appears that with two labels, the '*pad*' and '*damage*' bounding boxes tend to share some of their boundary lines which leads to the localization of the pad being inaccurate. This could be due to the RPN and ROI regression being faulty. Analyzing alternate ways of object detection that support better decision overlap could improve the detection results.

The aim of this study is to develop an intelligent vision system that can identify and localize a damaged area. This localization process is performed using a fixed camera orientation, which means that the view of the camera remains unchanged throughout the process and between different parts. Therefore, it is more important to have a specialized model to localize the '*pad*' surface with a higher accuracy for the setup proposed rather than a robust model with a much lower accuracy. For this purpose, a second case study is

performed with a new dataset consisting of images taken from the same camera orientation for one label '*pad*'. The results from this second model indicate a *mAP* score of 88.7, which is an increase of 26% compared to results from case study 1 for one label. Overall, an improvement of 35.9 in *mAP* is achieved by moving from a more diverse dataset trained with two labels to a less diverse dataset with one label. This represents a relative increase in accuracy of 68%.

The first case study is carried out on a relatively small dataset of 72 original images of eight fixed bends, whereas the second case study has 437 original images of four fixed bends. Results from the second case study are more favorable as the objective is to obtain a well-trained, more specialized model for detecting damages in a specific environment. A much larger and diverse dataset would enable the R-CNN to more accurately recognize features of the damage and the pad and yield a more robust and higher performing model. Furthermore, GPU limitation on the Google Colab platform restricted testing for a greater number of steps and experimenting with different hyperparameter settings to fine-tune the model and enhance performance. Access to more computational power would also make it possible to train and compare deeper architectures like ResNet101, for which training is more computationally expensive.

## **2.5 Conclusion**

Damage identification and localization in remanufacturing is a manual visual task. It can be time-consuming, tedious and prone to error. With recent advances in computer vision, increased availability of computational power and access to a large amount of data, it is now worthwhile to explore the use of this technology in remanufacturing. This chapter proposes a machine learning-based method for automatic visual detection and localization of damages in a robotic laser cladding repair cell process. To accomplish this, two configurations of Faster R-CNN utilizing transfer learning are employed. Two case studies are performed on different datasets, case study 1 with a more diverse set of images and case study 2 with more similar images. The comparative analyses of their performance are

also carried out. For case study 1, the model is trained with one label and two labels. The highest *mAP* score obtained while training for two labels is 52.8 using ResNet50 as the feature extractor. With the same feature extractor, the *mAP* score increases to 70.4 when training for only one label. For case study 2, the model is trained with one label only. The resulting model outperforms those from case study 1, reaching a maximum *mAP* of 88.7. The best model configuration in all cases is found to be Faster R-CNN with ResNet50 as the feature extractor. This model achieves a detection speed of 1.48 ms, rendering it potentially viable for real-time application. Promising results from this study demonstrate the potential of vision-based R-CNN technology in the field of repair and remanufacturing.

# **Chapter 3: A Computer Vision-based Spatial Damage Localization Method for an Autonomous Robotic Laser Cladding Repair Process**

In Chapter 2:, a comparative analysis between different feature extractors was carried out using the meta-architecture of Faster R-CNN. The results indicated a superior performance by Faster R-CNN together with ResNet50. This chapter is an extension of the automatic damage detection framework proposed in the previous chapter. A novel damage localization framework is proposed that will detect damage in 2D images and translate the damage location to 3D world coordinates to be used for automating a robotic laser cladding repair process. The model trained in Section 2.3 will be used to identify and localize the damage in 2D. The main objective is to localize damage on worn cylindrical components automatically.

## **3.1 Introduction**

A crucial strategy to achieve an environmentally conscious manufacturing sector is through repair or remanufacturing. Repair or remanufacturing is defined as the process of bringing a damaged or worn component to like-new condition. It is a well-established field that has garnered a lot of attention in research and industry due to its direct contribution to sustainable development by promoting product reuse [47].

Repairing large volumes of worn components in industries is a monotonous and tedious task that, due to human dependence, can result in the process being time-consuming with a low throughput of parts and yielding inconsistent results. This issue is particularly relevant nowadays, as industries operate in a rapidly changing environment due to influencing factors like new technologies and global competition, which have created a necessity for industries to permanently transform their structures and technologies to survive in this dynamic setting [48].

The fourth industrial revolution, or “Industry 4.0,” is aiming to revolutionize the face of the industry by advancing from conventional manufacturing to futuristic smart factories [49]. As such, to achieve autonomy in a repair process, it is important to have a system integrated with a robot. Aside from monetary and ecological advantages, robotic integration provides significant benefits to workplace health and safety [50]. However, the current state of repair is struggling to keep up with Industry 4.0 as it requires excessive human intervention for damage inspection and localization [51].

A traditional repair process usually comprises six main steps (see Figure 1.2): 1) data acquisition; 2) defect detection and identification; 3) reconstruction of the nominal model; 4) registration of the nominal model with the damaged model; 5) repair patch extraction; and 6) tool-path generation. This process is heavily reliant on a human operator to acquire spatial information of the entire damaged component in the shape of three-dimensional point clouds by either using a structured light-based or a laser triangulation-based scanner [52], [53]. Some efforts have been made to automate this process by mounting a laser scanner on a robot arm to acquire the 3D point clouds of the component [54]. 3D laser scanning is a widely accepted approach that produces precise and reliable point cloud data. After obtaining a digital model of the worn part, the next step is to inspect the model and localize the damage. To localize the damage, either a nominal CAD model needs to be present, or the nominal model needs to be reconstructed from the damaged model. Registration is performed to align the damaged model with the nominal model by finding optimal point-to-point correspondences [55]. By comparing the registered models, the three-dimensional geometry of the repair patch is extracted. This step is critical to the repair process because the repair volume's geometrical information is the basis on which a tool-path for deposition is generated [56].

3D laser scanning methods require the usage of expensive equipment and result in dense point clouds that contain large amounts of data. Moreover, geometric alignment of the point cloud data is a time-consuming and challenging task [57]. Aprilia et al. [58] concluded that to achieve an automated remanufacturing process; there exists an urgent

need to develop algorithms for damage detection and localization that do not require human involvement. This is partly due to geometrical information of the nominal model being inaccessible or unsuitable to use.

With the advancements in computer vision, computational resources and robot control, there is a growing research interest for robotic inspection of components in manufacturing and remanufacturing industries [59]. This is giving rise to a plethora of image-based localization methods that are simpler and more cost-efficient in comparison to the point cloud based methods [60], [61]. In addition to deep learning models mentioned previously in section 2.1, equipping robots with suitable sensors can provide them with the ability to perceive their environment. Utilization of RGB images as well as depth images enhances the performance of robotic inspection systems by replacing human operators. Obtaining depth information is key in these automated robotic systems to give the robot a 3-dimensional vision. Depth data can be acquired by RGB-Depth sensors, stereo vision-based systems or range sensors (e.g., LIDARs, time-of-flight sensors). In the realm of RGB-Depth sensors, Microsoft Kinect is the most popular RGBD camera being used for 3D reconstruction as it has the ability to achieve simultaneous correspondence between 2D and 3D environment, i.e., matching a pixel in a 2D image to a point in a 3D point cloud [62]. Kinect is employed either to align color and depth information [63] or coupled with deep learning models to carry out object detection in 3D [64]. However, it has limited depth measurement capabilities due to poor resolution and range restrictions [65].

Stereo vision (two cameras) and multi-vision (multiple cameras) are other common ways of extracting a 3D point corresponding to a 2D image pixel by finding the intersection point of the cameras' projection rays. For any camera-based world point reconstruction method, it is important to register the camera coordinate system to the world coordinate system [66]. This is done by calibrating each camera and obtaining its internal and external parameters in the shape of intrinsic and extrinsic matrixes.

Measurements from a camera compared to laser scanners are fast but lack accuracy, especially if it is data from a stereo vision system. To tackle this, a stereo vision system

and a laser scanner can be employed to work together to autonomously locate and reconstruct the part of interest [67]. Liu et al. [68] used stereo vision and a laser scanner configuration to localize damage on components for repair. However, they used markers to identify and isolate the damaged areas, making the method unintelligent, time-consuming and only implementable on a small scale.

Using the same working principle as binocular stereo vision, a multi-vision system incorporates several visual sensors to synchronously acquire images from varying viewpoints to reconstruct a model. Multiple viewpoints help get rid of any occlusions that might occur and are more accurate compared to their binocular counterpart, especially when reconstructing a complex large-sized object [69], [70]. However, a multi-view system leads to issues of calibrating multiple cameras and the position of each camera; this results in a high cost for setup and maintenance and the process being reasonably time-consuming [71], [72].

Over the last few years, Simultaneous Localization And Mapping (SLAM) using visual sensors has actively been researched due to its simple configuration and its capability of real-time reconstruction [73], [74]. However, ‘tracking failure’ is a common problem caused due to camera disturbances that can corrupt the estimated map and hinder the workability of the system [75], [76].

Based on the state-of-art reviewed above, it is evident that intelligent and autonomous damage localization methods have not been researched in the area of remanufacturing and repair. Contrary to current damage localization techniques in repair that lack intelligence and the ability to independently make decisions, there is plenty of research in the field of material science targeting intelligence-based repair processes [27], [77]. Common works include autonomous real-time monitoring of a laser welding and laser cladding repair process to track weld characteristics and adjust important weld parameters such as power of the heat source, material deposition rate, etc. [24], [25]. However, to be able to achieve full autonomy of a repair procedure, it is imperative that the robotic system is able to perceive the worn area on the component before making a smart decision.

Similar to areas such as autonomous driving and construction [78], [79], artificial intelligence technologies can be leveraged to make the damage identification and localization process in a repair cell autonomous and intelligent. The aim of this study is to propose a methodology focusing on automatically localizing and quantifying the damage on a worn cylindrical component. We introduce a novel damage quantification method that makes use of a camera to obtain a video feed of the worn component coupled with a ToF sensor. The intention behind this study is to provide an overview of the intelligent robotic laser cladding repair cell while highlighting the important design choices made and the mathematical concepts used in the approach. As such, the calibration of the system is covered, including the calculation of intrinsic and extrinsic matrixes and the validation of the camera model. Based on the validated calibration model, two case studies are carried out and their respective error analyses are presented. To the best of our knowledge, ours is the first autonomous repair pipeline based on intelligent machine learning algorithms incorporating a 2D camera and a time-of-flight sensor.

The rest of the chapter is structured as follows: Section 3.2 discusses the developed methodology in detail, including the proposed framework and the spatial localization technique. Section 3.3 validates the proposed localization system. In Section 3.4, two case studies are performed on worn cylindrical fixed bends, followed by a discussion of the results and limitations. In the final section, the conclusions drawn from the study are elaborated.

## **3.2 Methodology for Damage Localization**

### **3.2.1 Hybrid Localization Framework**

This study proposes an autonomous detection and localization strategy based on deep learning and computer vision techniques. The worn area is first localized in 2-dimensional space and then quantified in 3-dimensional coordinates. The rationale behind the term ‘hybrid localization framework’ is the concurrent usage of a camera and a ToF sensor to acquire data. Compared to the traditional repair method, the proposed method scans only



the worn surface autonomously making it more accurate and time and cost efficient. Figure 3.1 shows a flowchart outlining the proposed damage localization pipeline in detail.

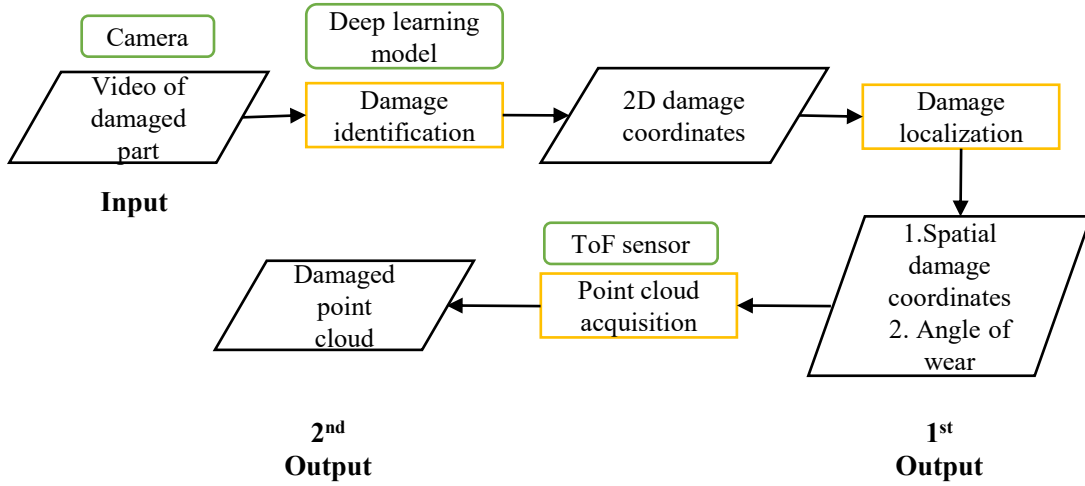


Figure 3.1: Hybrid damage localization pipeline

The input to the framework is a video of the damaged component recorded by a camera. This video is essentially a series of images and is fed into the trained Faster R-CNN model with a ResNet50 feature extractor as described in 2.3. The output from this deep learning model is a bounding box drawn around the region of interest containing the damage and its subsequent 2-dimensional coordinates. Using computer vision techniques, a damage localization process is carried out that uses these 2-D coordinates as an input to obtain the first output that contains critical data for repairing cylindrical components. This output entails the knowledge of the location of the worn area on the component in spatial coordinates and how much the wear encircles the cylindrical surface. This means a range of axial coordinates  $(l_1, l_2)$  and angle coordinates  $(\theta_l, \theta_t)$  as illustrated in Figure 3.2. Here the length of wear is defined by  $l_1$  and  $l_2$  and the angle of wear starts at the leading edge  $(\theta_l)$  and goes till the trailing edge  $(\theta_t)$

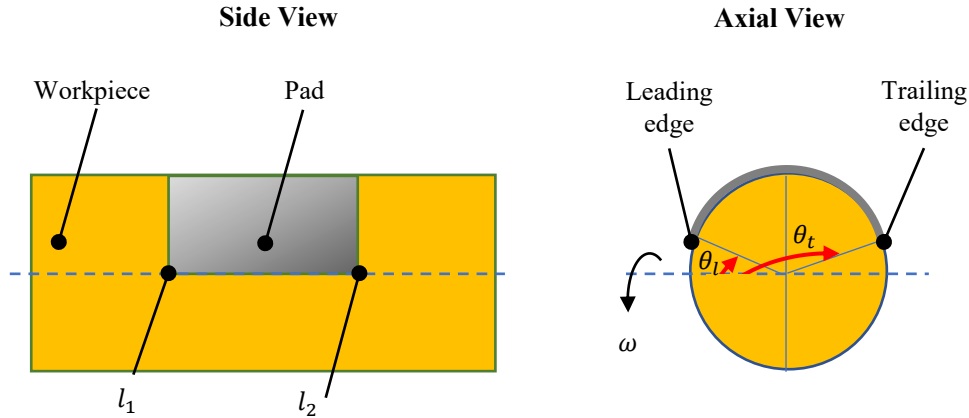


Figure 3.2: Side view and axial view demonstrating the length ( $l_1$ ,  $l_2$ ) and angle ( $\theta_l$ ,  $\theta_t$ ) of wear respectively

The data from the first output is sent as an input to a ToF sensor. Instead of scanning the entire component, the scanner will scan only the worn surface area bounded by the known axial and angular coordinates. The resulting output from the ToF sensor is the point cloud containing the 3-dimensional information of the damaged surface. Finally, this acquired point cloud can be used to generate an adequate tool-path for repair. A method for tool-path generation for cylindrical components using damaged point clouds will be introduced in Chapter 4:

### 3.2.2 Autonomous Robotic Laser Cladding Repair Cell

A UVC-G3-Bullet/UVC-G3-AF camera and a Keyence IL-300 ToF sensor are fitted on the laser head in a way that both the sensors are parallel to the laser head and are facing down at a right angle to the part axis (see Figure 3.3). This arrangement eliminates the tedious angle calculations, making the camera's mathematical model much easier to solve. The origin of the global coordinate system is in the center of the turntable, as annotated in Figure 3.3. The damaged cylindrical component will be mounted on this turntable for damage localization and repair.

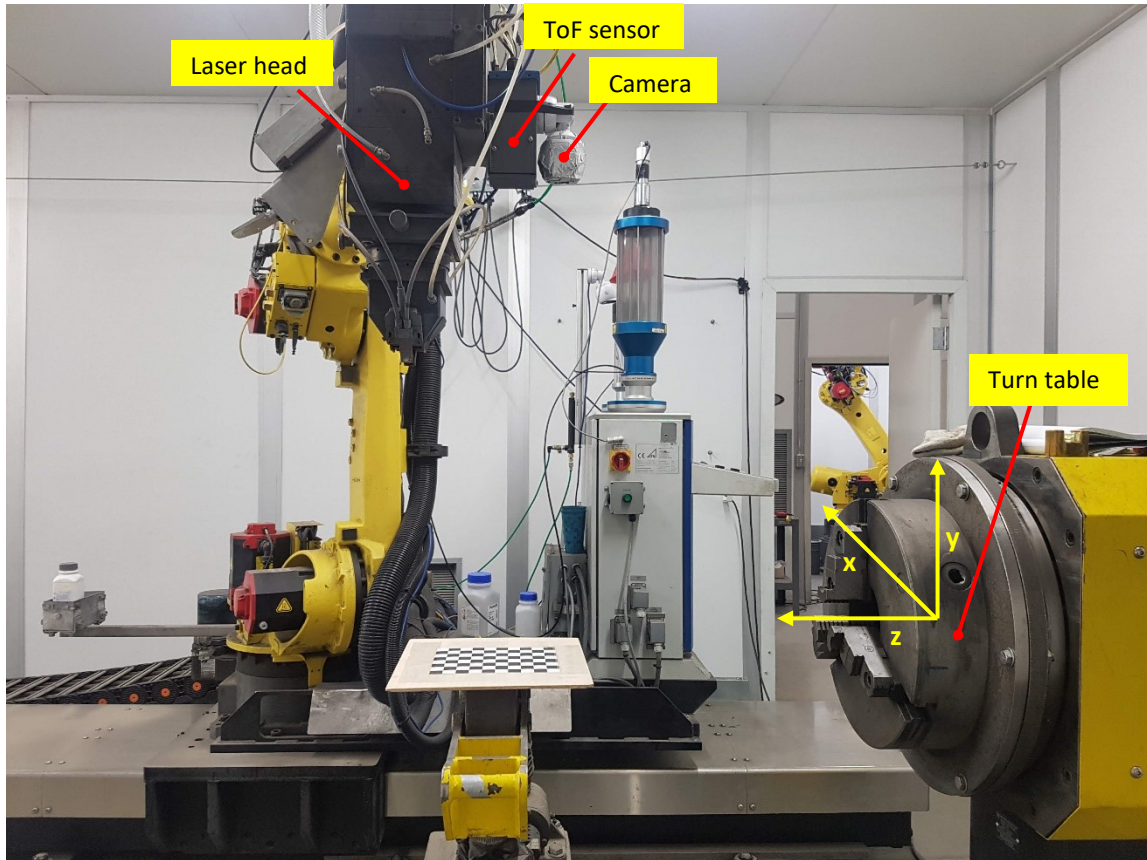


Figure 3.3: Setup of the RLCRC

To ensure consistency, the Fanuc-R-1000iA/80F robot will move to a fixed position called ‘localization home’ before spatially localizing each worn component. At the ‘localization home’ position, the laser head's pose with respect to the turntable is set and is saved in the system’s memory. It is important to note here that the coordinates of the laser head with respect to the global world frame can be read off from the machine at this position. However, for camera calibration, the position of the camera lens with respect to the global origin must be known. A schematic showing the entire setup with the arrangement of the camera from a side view and an axial view is shown in Figure 3.4. The offset between the camera lens and the turntable is measured manually, and the dimensions are illustrated in Figure 3.4. The side view shows the offset of the camera lens to the

turntable in the z-direction i.e., 499 mm, whereas the axial view indicates the offset in the x and y direction which is -10 mm and 700 mm, respectively.

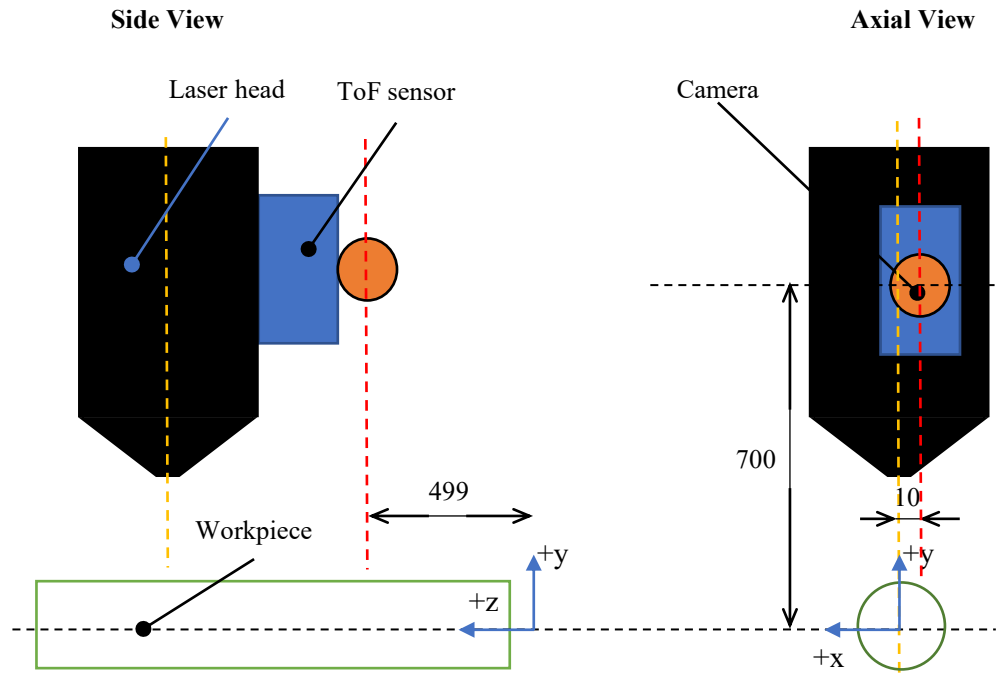


Figure 3.4: Camera offset (in mm) from the global coordinate origin

The damaged component is mounted on the turntable so that the part has pure rotational motion, and the axis of rotation is the longitudinal axis of the cylinder which is aligned with the center of the turntable. This rotational motion is the angular velocity of the turntable and is denoted by ( $\omega$ ) To ensure there is no bending of the part due to only one of its ends being fixed, the straightness of its axis is scrutinized with a spirit level. The camera has a motion sensor that automatically records a video once it detects movement. Exploiting this feature, the turntable is rotated at a known  $\omega$  and the video is recorded for one complete rotation. Figure 3.5 Shows the perspective from the camera of the damaged part. Here the camera has a top view of the part such that the length of wear is in the z-direction and the width of wear is in the x-direction.

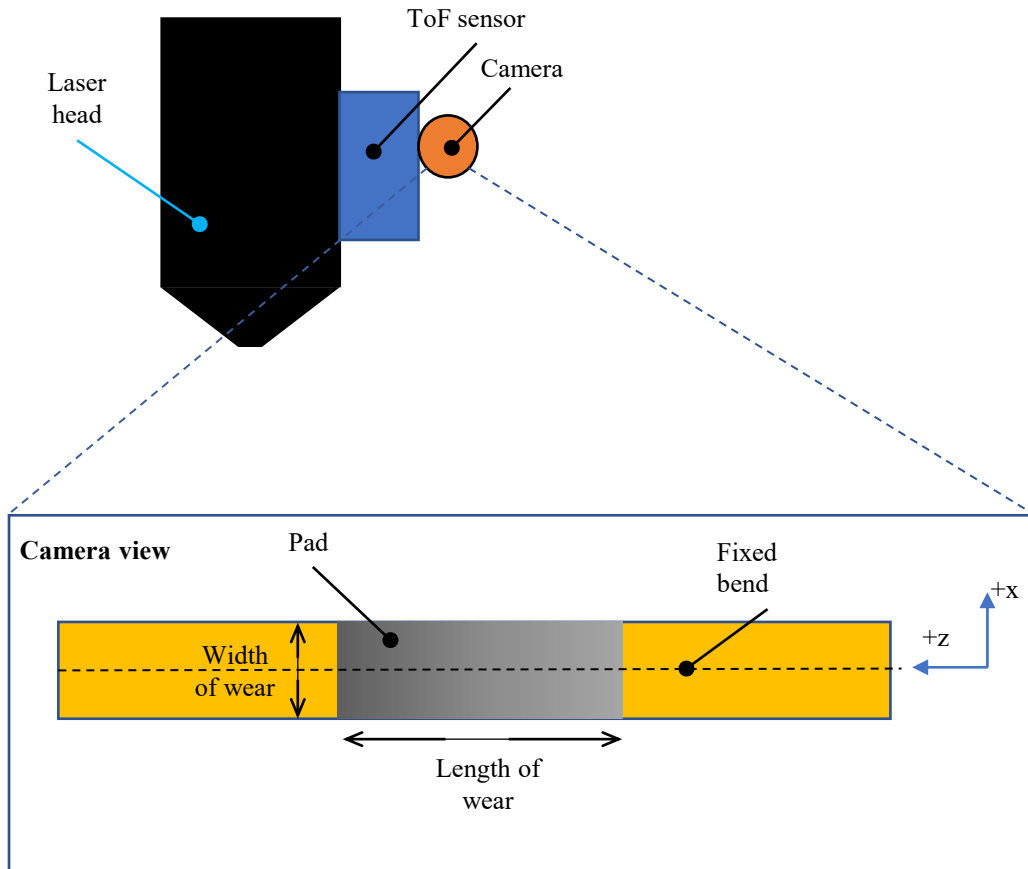


Figure 3.5: Perspective of the damaged part from the camera

### 3.2.3 Spatial Localization

#### 3.2.3.1 Camera Model

Calibration is an essential prerequisite for the utilization of a camera in an autonomous repair system. An imprecise calibration influences all parts of an autonomous repair framework, leading to inaccurate localization, scanning, and laser cladding results. Knowledge of both intrinsic and extrinsic parameters of the camera is required for a true projection of a 3-dimensional point in space to a 2-dimensional image point. Here the intrinsic parameters represent the internal characteristics of the camera, e.g., optical center and the focal length, whereas the extrinsic parameters determine the orientation of the

camera in the real-world space. Using the pinhole camera model, the relationship between the global (world) coordinates and the camera coordinates is shown below:

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A [R|t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3-1)$$

where:

$$A = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \quad (3-2)$$

$$R_z = \begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T = [t_x \quad t_y \quad t_z]^T$$

$(u, v)$  are the 2-dimensional image coordinates (horizontal, vertical) and  $(X, Y, Z)$  are the 3-dimensional global coordinates;  $(A)$  is the camera intrinsic matrix with  $(\alpha)$  and  $(\beta)$  being the focal lengths expressed in pixel units,  $(u_0, v_0)$  are the image coordinates of the principal point which is usually at the image center; and  $(R|t)$  represents the rotational and translational matrixes from the camera coordinate system to the global coordinate system, called the extrinsic matrix of the model.

Finally, the 3-dimensional global coordinates of the damaged area can be calculated as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{pmatrix} z_c & \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} A^{-1} - t \end{pmatrix} R^{-1} \quad (3-3)$$

### 3.2.3.2 Calibration Method

As established earlier, to reconstruct the 3-dimensional world coordinates of any image from its image coordinates, it is important to compute the camera's intrinsic and extrinsic parameters. The intrinsic matrix is calculated using MATLAB's computer vision toolbox; 18 images of a 7x9 asymmetric checkerboard of 20mm square size are collected from different angles and fed into the code. The code finds the checkerboard pattern and returns the detected corner points with the reprojected error and the camera parameters as shown in Figure 3.6 and Figure 3.7, respectively.

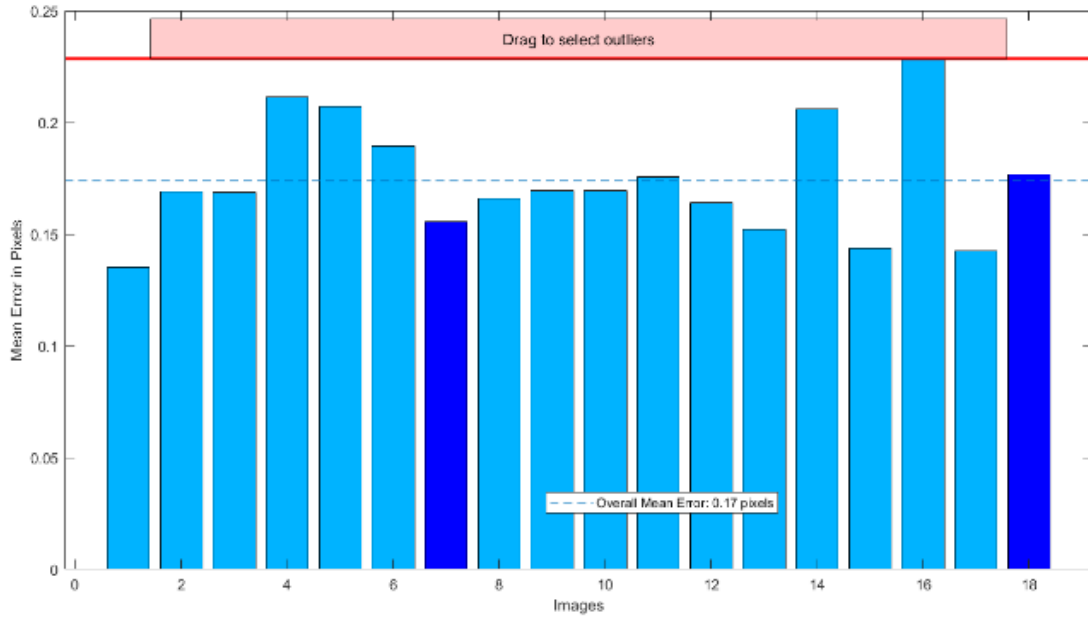


Figure 3.6: Reprojection error output from the calibration code for each of the calibration images

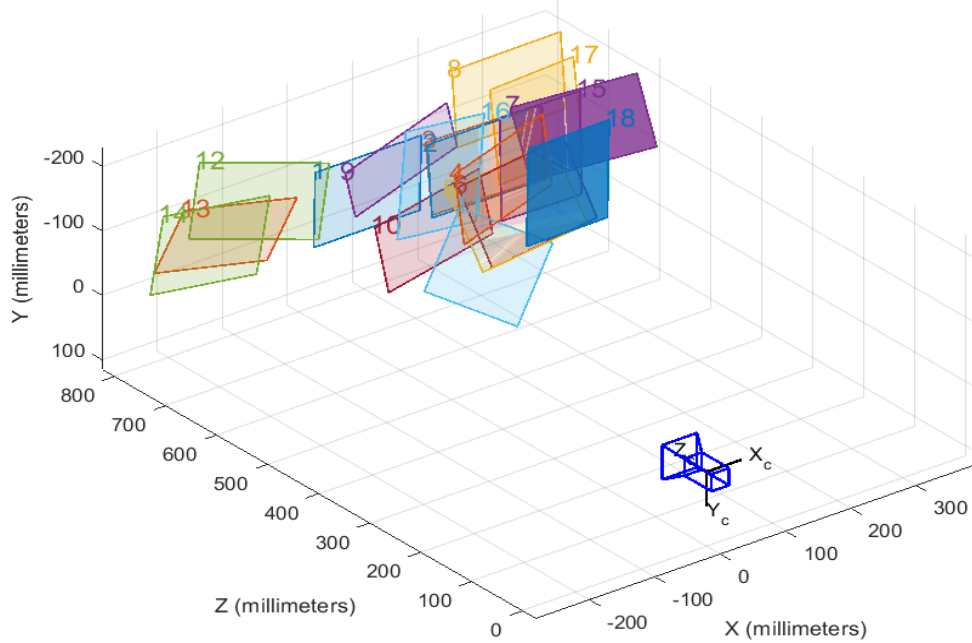


Figure 3.7: Camera parameters output from the calibration code for each of the calibration images

Since the camera is fixed at the ‘localization home’ position during the inspection and localization process, the intrinsic and extrinsic matrixes obtained here remain unchanged and are valid for every image. The calibration results are outlined in Table 3.1.

Table 3.1: Calibration Parameters

Mean projection error	0.17
Intrinsic matrix	$A = \begin{bmatrix} 1284 & 0 & 950 \\ 0 & 1280 & 562 \\ 0 & 0 & 1 \end{bmatrix}$
Extrinsic matrix	$(R t) = \begin{bmatrix} 0 & 0 & 1 & -10 \\ -1 & 0 & 0 & 700 \\ 0 & -1 & 0 & 499 \end{bmatrix}$



### 3.2.3.3 Obtaining the Angle of Wear

In addition to the length of wear in the axial direction, it is also possible to calculate the width of wear, as seen in Figure 3.5, from the 3-dimensional cartesian coordinates obtained. However, the entirety of the damage area can not be represented in a single frame of the video but over a series of frames. This is due to the damage extending in the tangential direction over multiple frames. Since different frames can not be used to quantify the height of the damage, an innovative approach is taken that uses a series of images from a video. The video is recorded at a known frame rate (fps, frames per second) while rotating the turntable counterclockwise at a set  $\omega$ . These parameters are outlined in Table 3.2. As mentioned earlier, the camera that is used has a motion-activated sensor, so it can autonomously record a full rotation of the cylindrical part. Once the worn component is mounted on the turntable, the robot arm moves to the ‘localization home’ position and then the part is rotated a full 360°. It is important to note that at the ‘localization home’ position, the turntable's angular coordinate is 0. This rotation is recorded by the camera so that frame 1 corresponds to 0° and the last frame corresponds to 360°.

Table 3.2: Parameters to consider for getting the angle of wear

Frame rate	30 fps
$\omega$	41.52°/s

The recorded video is fed frame by frame into the Faster R-CNN model proposed in Section 2.3 for damage detection. The frames in the video are labeled corresponding to their frame number. When damage is detected, the model draws a bounding box around it and outputs the pixel coordinates of the box. Note that each bounding box has two u-coordinates ( $u_l$  and  $u_r$  for left and right) and two v-coordinates ( $v_l$  and  $v_t$  for top and bottom).

The two edges of the damage, that are the top and bottom boundaries, will be called leading edge and trailing edge with respect to the rotational direction of the part in the

video as shown earlier in Figure 3.2. The angle of wear goes from  $\theta_l$  to  $\theta_t$  which is the desired output from the program. The program will go through the video frame by frame. As the part rotates, the leading edge of the wear will start appearing in the image, followed by the trailing edge, as shown in Figure 3.8. The vertical image coordinate of the part center axis is denoted by  $v_c$  and it remains constant at 525 pixels for every image. As the sequence of frames progresses, the leading and the trailing edge pass by the known location  $v_c$ . The angle coordinates of the edges can be obtained by determining when the edges pass by the part center axis. In some frame, the leading edge will be closest to the center axis (see position 2); this frame is denoted as ‘start frame’ or frame with leading edge ( $f_l$ ). In some frame, the trailing edge will be the closest to the center axis (see position 4); this frame is denoted as ‘end frame’ or frame with trailing edge ( $f_t$ ).

Moving on, the axial length coordinates ( $l_1$  and  $l_2$ ) will be obtained by using the left ( $u_l$ ) and right ( $u_r$ ) coordinates of the bounding box. For each bounding box, the absolute distance between  $u_l$  and  $u_r$ , called  $\Delta u$ , will change as the sequence of the frames progresses, depending on how much of the pad is visible in a specific frame. Across all bounding boxes, the minimum value of  $u_l$  and the maximum value of  $u_r$  will be checked and saved as  $u_{lmin}$  and  $u_{rmax}$ , representing the left and the right pixel coordinate, respectively. Here, the difference between  $u_{lmin}$  and  $u_{rmax}$  is equivalent to the length of the pad in pixels. Taking only the lowest and the highest value will ensure the selected coordinates contain the entirety of the pad. Finally, using Equation (3-3), the pixel coordinates  $u_{lmin}$  and  $u_{rmax}$  will be converted to world coordinates  $l_1$  and  $l_2$ , respectively.

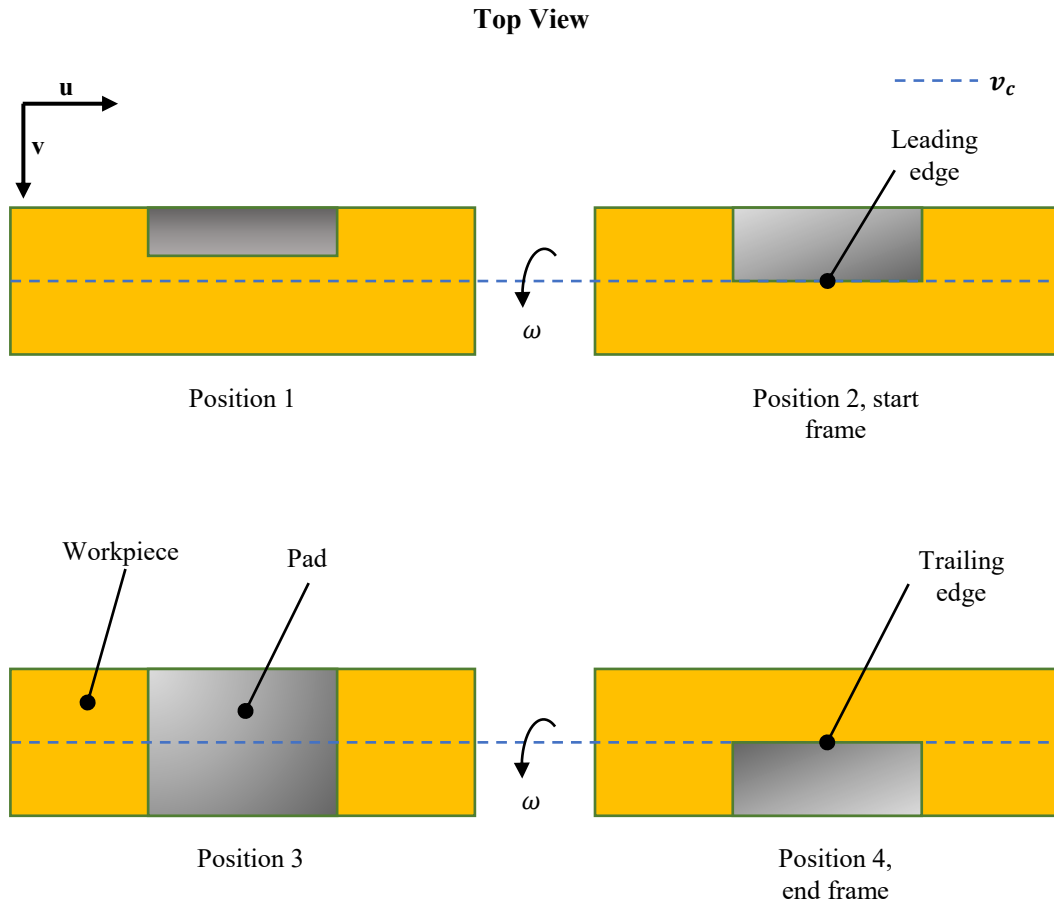


Figure 3.8: Extracting angle of wear from video based on frame number

A simplified pseudo-code for the generation of  $u_{lmin}$ ,  $u_{rmax}$ ,  $f_l$  and  $f_t$  can be found below.

1. Extract all frames from the video and save them in chronological order;
2. Carry out inference on each frame;
3. Save the bounding box coordinates for all frames in a list;
4. Initialize  $\Delta v_l$  and  $\Delta v_t$  to store v-distance information for leading/trailing edge during the loop;
5. Initialize  $u_{lmin}$  and  $u_{rmax}$  as temporary variables to determine the minimum and maximum  $u$  coordinates across the bounding boxes;

6. For each bounding box, compare the coordinates  $v_l$  and  $v_t$  of the bounding box with  $v_c$ :
  - a. If  $|v_l - v_c| < \Delta v_l$  then save the index of the current bounding box as  $f_l$ ;
  - b. If  $|v_t - v_c| < \Delta v_t$  then save the index of the current bounding box as  $f_t$ ;
  - c. Repeat for all bounding boxes;
7. For each bounding box, compare the coordinates  $u_l$  and  $u_r$  of the bounding box:
  - a. If  $u_l < u_{lmin}$ , then update  $u_{lmin} = u_l$ ;
  - b. If  $u_r > u_{rmax}$ , then update  $u_{rmax} = u_r$ ;
8. The final values of  $f_l$  and  $f_t$  are the indices of the ‘start’ and ‘end’ frames,  $u_{lmin}$  and  $u_{rmax}$  are the start and end point of the axial wear.

After computing  $f_l$  and  $f_t$ , the angular coordinates can be calculated based on the frame rate and  $\omega$  using Equation (3-4). The angular position of the turntable at frame 1 is known ( $\theta_0 = 0$ ).

$$\theta_l = \frac{f_l}{fps} \times \omega \quad (3-4)$$

$$\theta_t = \frac{f_t}{fps} \times \omega$$

$$\Delta\theta = \theta_t - \theta_l \quad (3-5)$$

where  $\theta_l$  and  $\theta_t$  stand for the angular coordinates of the leading edge and trailing edge, respectively;  $f_l, f_t$  represent the frame number with the leading and trailing edge, respectively;  $\Delta\theta$  stands for the total angular displacement.

Finally,  $\theta_l, \theta_t, l_1$  and  $l_2$  will be the input to the ToF sensor to scan the worn surface of the cylinder.

### 3.3 Validation of the Setup

To ensure the robotic repair cell is adequately calibrated, a validation process is performed. A fixed bend of radius 60 mm is mounted on the turntable and three equal lengths of black

tape are applied to its surface (see Figure 3.9). The edges of the tapes are marked in yellow. The robot is moved to its ‘localization home’ position and an image is captured. The real-world coordinates representing the edges of the three tapes are then measured and recorded. Based on the pixel coordinates obtained from the image, the world coordinates of each point are calculated.

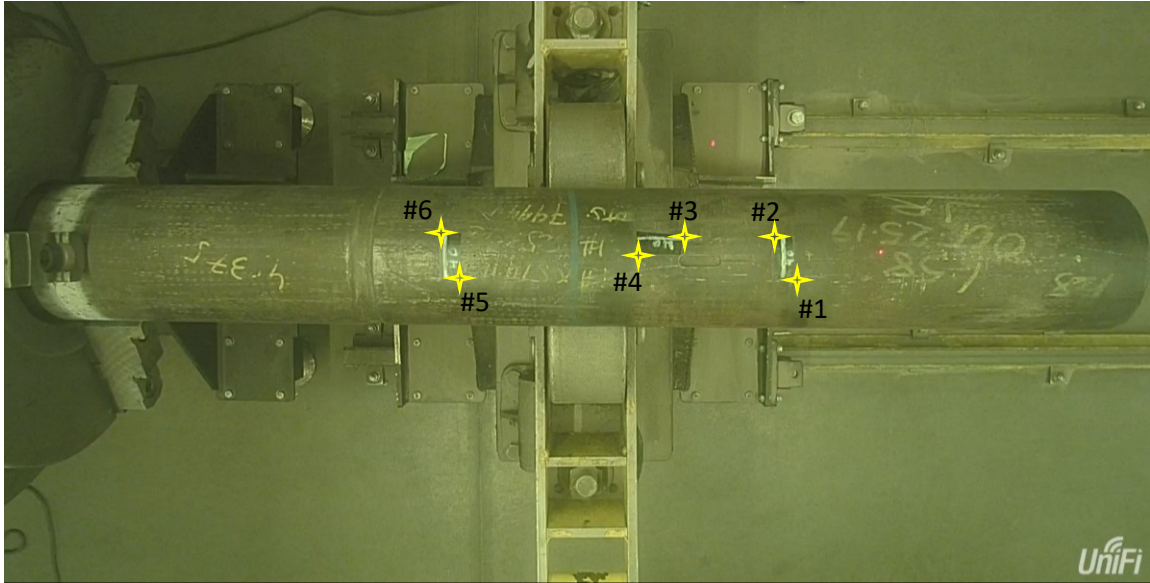


Figure 3.9: Three tape markers with known edge coordinates for calibration validation

The performance of the proposed calibration model is assessed by calculating the root mean square error, which is:

$$\Delta = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} \quad (3-6)$$

where  $x_i, y_i, z_i$  are the calculated world coordinates of the pad location; and  $x, y$  and  $z$  are the real-world coordinates taken manually by guiding the robot arm to the pad location.

The relative error percentage is then defined as:

$$\sigma = \frac{\Delta}{\sqrt{x^2 + y^2 + z^2}} \times 100 \quad (3-7)$$

Table 3.3 outlines the real-world measured coordinates, the computed world coordinates and the resulting error values and their average and standard deviation. Since the localization is only relevant to the axial coordinates that correspond to the z axis (the angular coordinates are calculated independent of the spatial coordinates), it will be more fitting to calculate the error in z.

Table 3.3: Validation of the calibration setup using tape markers

Point	Measured coordinates (mm) $[x, y, z]^T$	Calculated coordinates (mm) $[x_i, y_i, z_i]^T$	$\Delta$ (mm)	$\sigma$ (%)	$e_z$ (mm)
#1	$\begin{bmatrix} -16 \\ 60 \\ 674 \end{bmatrix}$	$\begin{bmatrix} -13.969 \\ 60 \\ 682.972 \end{bmatrix}$	9.199	1.358	8.972
#2	$\begin{bmatrix} 18 \\ 60 \\ 657 \end{bmatrix}$	$\begin{bmatrix} 21.5 \\ 60 \\ 664.981 \end{bmatrix}$	8.715	1.320	7.981
#3	$\begin{bmatrix} 19 \\ 60 \\ 583 \end{bmatrix}$	$\begin{bmatrix} 21 \\ 60 \\ 591.211 \end{bmatrix}$	8.451	1.441	8.211
#4	$\begin{bmatrix} 4.3 \\ 60 \\ 544 \end{bmatrix}$	$\begin{bmatrix} 7.888 \\ 60 \\ 552.495 \end{bmatrix}$	9.222	1.684	8.495
#5	$\begin{bmatrix} -18 \\ 60 \\ 395 \end{bmatrix}$	$\begin{bmatrix} -15 \\ 60 \\ 406.290 \end{bmatrix}$	11.682	2.921	11.290
#6	$\begin{bmatrix} 20 \\ 60 \\ 377 \end{bmatrix}$	$\begin{bmatrix} 23.5 \\ 60 \\ 389.343 \end{bmatrix}$	12.830	3.356	12.343
Average			10.016	2.014	9.549
Standard deviation			1.796	0.891	1.818

From these error values, an error margin can be calculated that needs to be added to the calculated axial ( $z$ ) coordinate to ensure the pad is entirely represented by the coordinates. This is illustrated in Figure 3.10, where  $\bar{e}_z$  and  $s_z$  are the average and the standard deviation of the error in  $z$ , respectively. Assuming the error values in  $z$  follow a normal distribution, about 95% of the values will be within  $\pm 2s_z$  of  $\bar{e}_z$ .



Figure 3.10: Error tolerance in  $z$  to account for the calibration error

### 3.4 Case Study Results

Utilizing the localization technique presented in Section 3.2.3, two case studies are presented in this section. These case studies are carried out on two different parts with varying radii. Following are the results from the case studies and how they compare to the measured or actual value. It is important to note that the length of wear was manually measured with a measuring tape and the angle was noted by reading the angular coordinate of the turntable at the location of the leading edge and the trailing edge.

#### 3.4.1 Case Study 1: Implementation of Damage Localization Method

The first case study was done on a fixed bend of radius 60 mm with a known pad length and wear angle, as demonstrated in Figure 3.11. From the inference, 262 frames were extracted. It was found that the values of  $f_l$  and  $f_t$  are 1 and 87, respectively. The values of  $u_{lmin}$  and  $u_{rmax}$  are 1049 and 1323, respectively. Results from this case study are tabulated in Table 3.4. The results show a total angular displacement error of  $0.408^\circ$  and a length error of 9.91 mm. The values of the length and angle coordinates containing the repair patch with the error tolerance (see Figure 3.10) added to the  $l_1$  and  $l_2$  coordinates are

sent to the ToF sensor and the equivalent point cloud is obtained as shown in Figure 3.12. The time it takes to acquire the point cloud of the damage surface is 16 minutes, whereas obtaining the entire surface geometry would take 45 minutes.

Table 3.4: Analysis of results from case study 1

	$\theta_i$	$\theta_t$	$\Delta\theta$	$l_1$ (mm)	$l_2$ (mm)	$\Delta l$ (mm)
Measured	0°	120°	120°	540	660	120
Calculated	0°	120.408°	120.408°	545.94	675.85	129.91
Error	0°	0.408°	0.408°	5.94	15.85	9.91

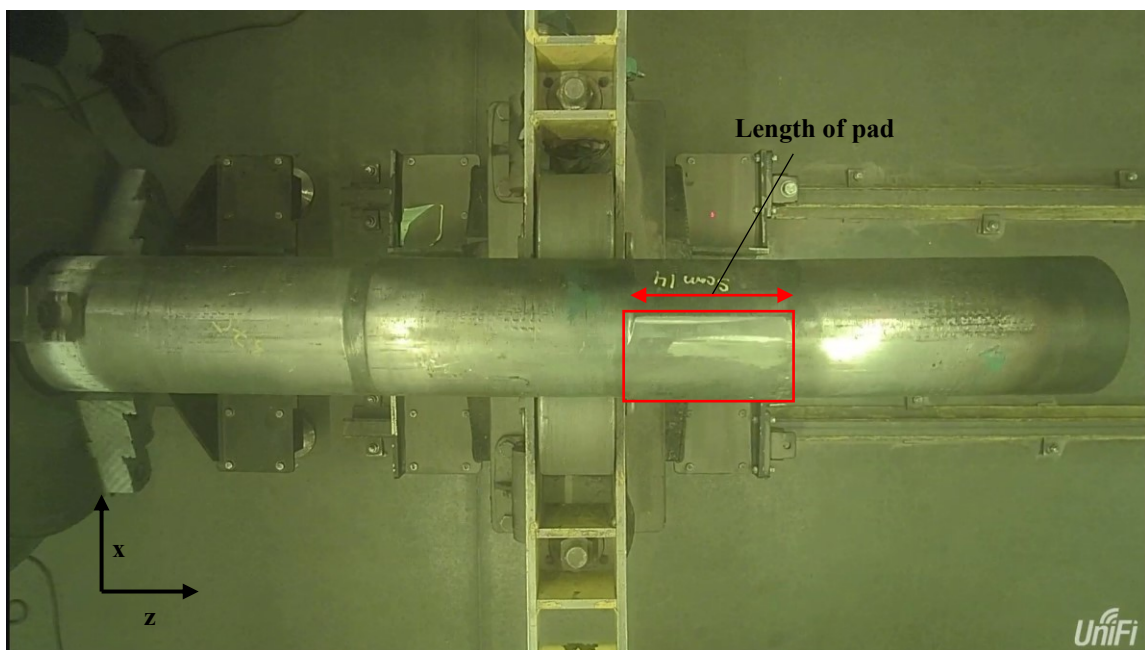


Figure 3.11: Fixed bend for case study 1, with the pad and axes annotated



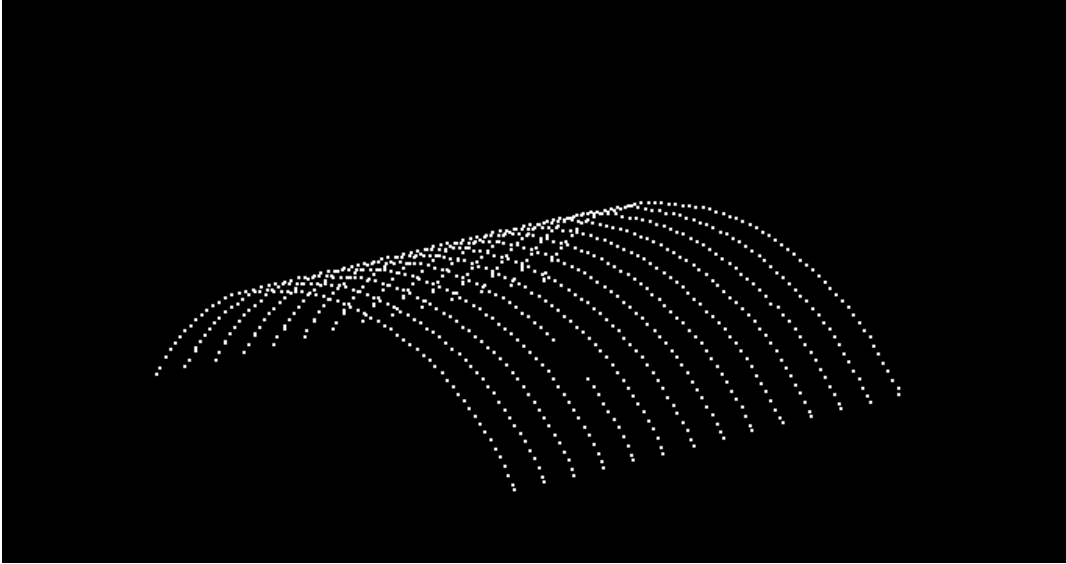


Figure 3.12: Point cloud representing the damaged component for case study 1

### 3.4.2 Case Study 2: Implementation of Damage Localization Method

The second case study is done on a part shown in Figure 3.13 with a radius 91 mm. From the inference, 265 frames were extracted. It was found that the values of  $f_l$  and  $f_t$  are 52 and 132, respectively. The values of  $u_{lmin}$  and  $u_{rmax}$  are 1441 and 1757, respectively. Table 3.5 presents the results obtained and the corresponding error analysis. The results show a total angular displacement error of  $0.72^\circ$  and a length error of 0.17 mm. The values of the length and angle coordinates containing the repair patch with the error tolerance (see Figure 3.10) added to the  $l_1$  and  $l_2$  coordinates are sent to the ToF sensor and the equivalent point cloud is obtained, as shown in Figure 3.14. The time it takes to acquire the point cloud of the damage surface is 20 minutes, whereas capturing the entire surface geometry would take 54 minutes.

Table 3.5: Analysis of results from case study 2

	$\theta_l$	$\theta_t$	$\Delta\theta$	$l_1$ (mm)	$l_2$ (mm)	$\Delta l$ (mm)
Measured	73	183	110°	731	881	150
Calculated	71.97°	182.69°	110.72°	731.8	881.63	149.83
Error	1.03°	0.31°	0.72°	0.8	0.63	0.17

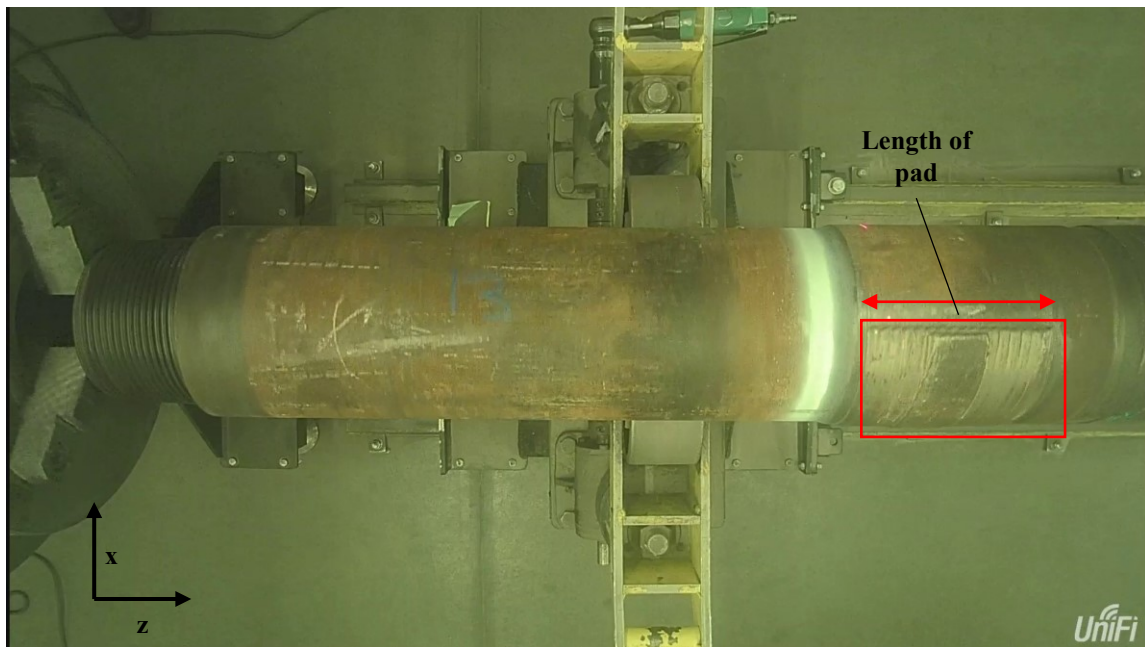


Figure 3.13: Fixed bend for case study 2, with the pad and axes annotated

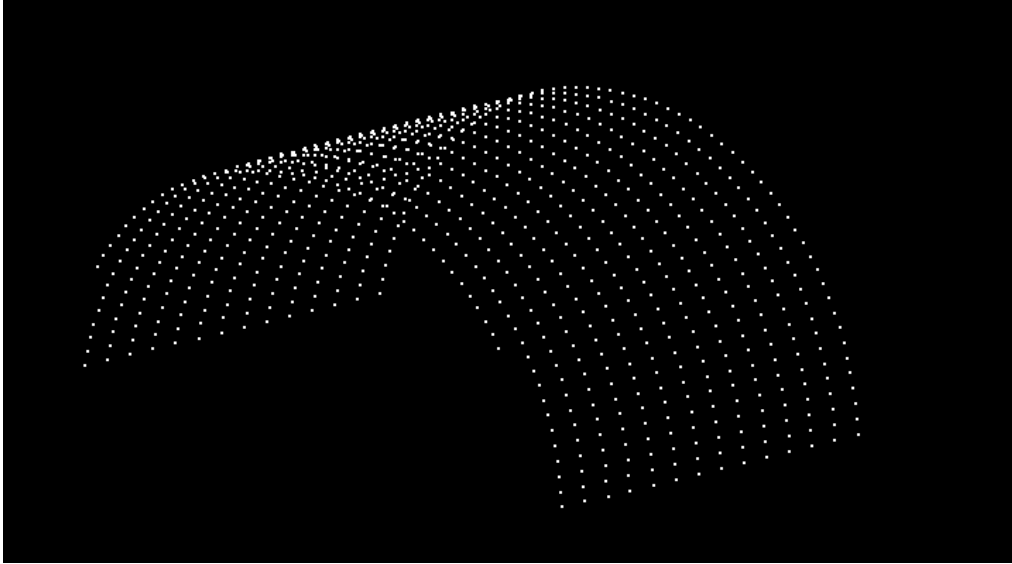


Figure 3.14: Point cloud representing the damaged component for case study 2

### 3.5 Discussion and Conclusion

Repair or remanufacturing is a green manufacturing strategy that helps to reduce industrial waste by reducing cost, energy, and, most importantly, material consumption. Damage localization is a crucial step in repair that typically relies heavily on a human operator. The proposed robotic laser cladding repair system is able to autonomously locate and quantify the damaged areas on worn components. In order for the ToF sensor to autonomously scan only the damaged region, the robotic system needs to be calibrated so that two-dimensional information from a camera can be translated into three-dimensional global coordinates. To achieve this, the camera in the RLCRC is calibrated and the calibration parameters with respect to the global coordinate origin are found. Since the damaged region wraps around a cylindrical part's surface, the main goal is to obtain the longitudinal and angular coordinates of the damage patch. The damaged part is mounted on the turntable and the robot arm is moved to the 'localization home' position. The turntable is rotated for a full 360° and whilst it rotates, a video is recorded for the deep learning model to identify and localize the damaged region on 2D image data. Based on the spatial localization

parameters presented in section 3.2.3, this information is converted to 3D world coordinates providing the axial and angular ranges of the damaged region. To corroborate the 2D to 3D conversion, a validation between the measured and calculated 3D coordinates is conducted that shows an average relative error of 2.018%. Moreover, two case studies are presented where the calculated length and angle of wear are compared to the measured values. For case study 1, the results show a length error of 9.91mm and an angular displacement error of 0.408°, respectively. For case study 2, the length error and angular displacement error come out to be 0.17 mm and 0.72°, respectively.

The aim of this study is to significantly reduce scanning time and costs that arise from traditional repair methods, where the entire surface area of a damaged component is scanned. By developing a novel system that autonomously scans only the damaged surface, this proposed method is time and cost-efficient. For instance, in case study 1, scanning the entire worn model was found to take 45 minutes, whereas scanning only the damaged region takes a mere 16 minutes. In case study 2, since the part had a larger diameter, it took 20 minutes to scan the damage region as compared to 54 minutes for scanning the entire surface. On average, the scanning time was reduced by more than 63%. This study's promising results indicate the possibility of implementing an intelligent, autonomous and fast damage localization system in repair cells.

# **Chapter 4: An Efficient Tool-Path Planning Approach for Repair of Cylindrical Components via Laser Cladding**

## **4.1 Introduction**

The manufacturing industry plays a major role in technological advancement and accounts for a large part of the global energy and materials consumption [80]. The majority of components in engineering facilities consist of cylindrical and planar geometries [81]. Cylinders are widely used to produce parts for industrial plants, the construction and automobile sectors in the form of pipelines, drill bits, gas cylinders, etc. For manufacturing industries to keep up to par with the rising sustainability challenge, it is critical to analyze industrial waste and propose new ways to recycle end-of-life components [82]. Worn cylindrical parts contribute a significant amount to this scrap, which is why comprehensible and coherent methods need to be employed to minimize the number of lost resources. These methods mostly consist of additive manufacturing technologies to bring the part to a new state and often optimize the components topology for greater longevity [83], [84].

Direct energy deposition (DED) is the most widely used additive repair technology for its advantages which are: 1) a smaller heat-affected zone as compared to welding; 2) an extensive range of deposition rate; and 3) can be easily implemented with a robot to automate repair. This boosts the quality, reliability and repeatability of the remanufactured part [85], [86]. Laser metal deposition (LMD) or laser cladding (LC) is a laser-based DED process that directs a laser beam onto a metallic surface to form a melt pool into which powder is deposited [85].

Remanufacturing involves six steps, amongst which reverse engineering (RE) is the first and vital stage for intelligent repair. The acquisition of geometries and specific model measurements is necessary to examine and treat the damaged area [87]–[90]. Geometrical

data can be obtained in the shape of three-dimensional point clouds by scanning surfaces using a laser scanner [91], [92]. After obtaining a digitalized model of the worn part, the damage needs to be localized. To localize the damage, either the part's nominal CAD model needs to be present, or the nominal model needs to be reconstructed from the damaged model [93], [94]. Different methods are used to model the surface depending on the complexity; for primitive surfaces, random sample consensus (RANSAC) or Hough Transform is used whereas, for more complex curved surfaces, non-uniform rational B-splines (NURBS) is applied [82], [95]. The repair volume can be extracted with both models present via a Boolean operation or a distance-based filtering operation [96]–[98]. Finally, a tool path must be generated, which is suitable for the chosen additive repair strategy.

The available literature on the state-of-the-art shows that damaged models in the point cloud format are commonly converted to the stereolithography (STL) format or a computer-aided design (CAD) model to be reconstructed [84], [90]. Converting the acquired data to different formats increases the probability of errors being introduced. To combat this, Mineo et al. [99] develop a Mesh Following Technique (MFT) that takes in an STL file as an input and generates a tool-path directly. This method generates curve trajectories or tool-paths lying on the tessellated surface by locating the coordinates of the points on the mesh. It is tested on a free-form curved surface and gives valid results. However, this concept is based on acquiring a tessellated CAD model that is a true depiction of the workpiece surface by polygons [93]. Masood et al. [100] and Zhang et al. [95] also discuss the tool-path generation directly from point clouds. Their methods consider using a Bezier surface interpolation algorithm to reconstruct the surface. This works well for complex curved surface structures and allows them to rebuild damages when the nominal geometry is unknown. However, reconstruction is computationally expensive and results in some deviation between the reconstructed and the nominal model [101]. Therefore, for the repair of primitive surfaces like cylinders and spheres with a known nominal geometry, reconstruction is not favorable. Using a simpler and more direct approach, tool-path generation for these surfaces can be accomplished more efficiently.

However, tool-path generation methods for primitive surfaces are rarely mentioned in the literature.

Another difficulty with high precision repairs is that the damaged surface must be scanned with a high resolution because the quality of the repair is analogous to the accuracy of the information of the scanned surface. Obtaining high-resolution scans is a time-consuming process [102]. Therefore, from an extensive review, the following challenges were identified in this area: 1) generating a tool-path directly from point cloud data; and 2) adapting the resolution and/or the information of the point cloud data based on the laser cladding system. This paper proposes an effective approach to tool-path generation from point clouds, specifically for repairing damaged cylindrical components using laser cladding technology.

## **4.2 Methodology for Tool-Path Generation**

### **4.2.1 Point Cloud Acquisition and Analysis**

For this work, a Keyence IL-300 time-of-flight sensor scans the physical component with a laser light source and a measurement range of 160 to 450 mm to acquire the coordinates. The sensor is mounted on the same robotic arm that holds the laser cladding system to execute the repair. Thereby it is ensured that the coordinate system remains consistent throughout all steps of the process. A point cloud file (.pcd) stores the digital model obtained from the scan. A point cloud is a representation of a set of points in a 3-dimensional coordinate system, well-defined by x, y and z. In this data format, the model surface is represented by a set of points in 3D space [103]. The point cloud data obtained in the scanning procedure contains the repair patch geometry in Cartesian coordinates. They are then converted into the equivalent points in cylindrical coordinates by Equation. (4-1). Polar coordinates work well with objects that have a rotating symmetry along their longitudinal axis. Here  $r$  is the radius of the cylinder.

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \arctan y/x \tag{4-1}$$

$$z = z$$

Now, by plotting these polar data points in a Cartesian coordinate system, the cylindrical geometry is turned into a plane surface. Figure 4.1 illustrates the radial coordinate  $r$  of a perfect cylinder is constant for every point on its surface. Turning  $r$  into a Cartesian coordinate (i.e.,  $z$ ), the perfect cylinder becomes a perfect plane with constant  $z$ . As a result, the tool-path for a cylindrical part can now be generated on a flat digital representation of the geometry, which greatly simplifies the material buildup process. It should be noted, leveraging off the polar points for tool-path generation for a cylinder is not found in the literature.

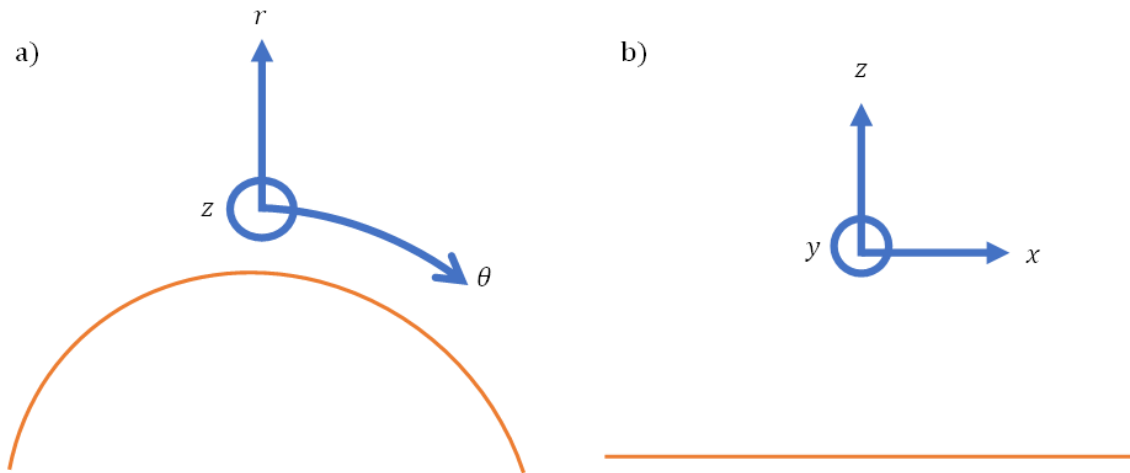


Figure 4.1: Representation in a) Polar coordinate system b) cartesian coordinate system

In Figure 4.2, the traditional remanufacturing process and the novel process are compared by means of a flowchart, where the process flows are represented by a black dashed line and a red line. Here it can be seen that after point cloud acquisition, the conventional process goes through several intermediate steps in order to repair the part. These steps are important when working with surfaces that have free form geometry and



intricate details. However, with primitive surfaces, e.g., cylinders, a much faster and direct approach can be followed. This approach takes into consideration the polar representation of the damaged point cloud data and directly generates a deposition tool-path in order to repair the part. Registration and volume extraction are omitted because a nominal model is not required as an input.

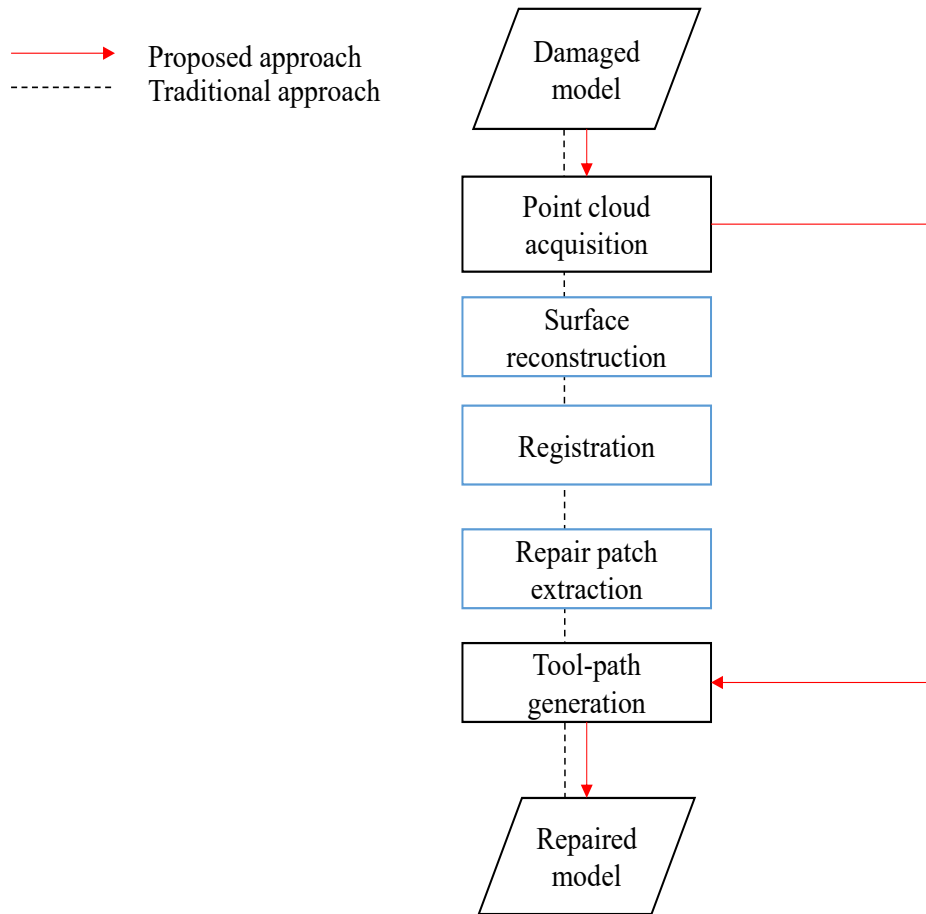


Figure 4.2: Flowchart comparing the existing approach to remanufacturing with the proposed method

## 4.2.2 Tool-Path Generation

The process of scanning the damaged surface geometry can be time-consuming, especially if a fine spatial resolution is required to achieve high precision repairs. The information obtained from the scan is only as accurate as the data point resolution. This can be challenging when the step size of material deposition is smaller than the data resolution. Figure 4.3 shows the setup of the robot laser cladding system (RLCS) in a simulation environment in RoboDK software. The setup is done in a way that the cylindrical workpiece is clamped on the chuck. The damaged component has pure rotational motion, and the axis of rotation is the center of the chuck. The frame is calibrated with the chuck and the longitudinal axis of the cylinder is aligned with that of the chuck. To examine the level and unbending of the part, it is further scrutinized with a spirit level. A laser scanner is mounted on the robot arm and this extracts the coordinates of the damaged component. Note, the scanner is mounted on the same robotic arm that holds the laser cladding system to execute the repair. Thereby it is ensured that the coordinate system remains consistent throughout all steps of the process. Since this cell is used to carry out the entire remanufacturing process, it is crucial to reduce the scanning time while maintaining a high-quality repair. This is because, every second the cell is engaged in scanning, could be used to repair an additional part and thus increase profit.

In the following, the logic of the tool-path planning algorithm is explained. Table 4.1 shows a pseudocode representing the algorithm. The coordinates are processed as mentioned earlier in section 4.2.1. The robotic arm moves in the z-axis, which is the direction normal to the surface of the cylinder. The inputs to the program are the height of each layer and the pitch of the machine, which represent the distance from the base to the top of each layer and the width of each bead clad, respectively. The program takes the nominal z value, which is the radius of the nominal cylinder, and subtracts this from all z values of the point cloud. By doing so, the highest z value becomes 0 and the difference between this and the lowest ( $z_{min}$ ) is the total depth of repair. To obtain the total number of layers, this value is divided by the layer height ( $l_h$ ), which is constant for each machine.

The layers are then defined from the bottom of the damage upwards, starting at  $z_{min}$  with the spacing  $l_h$ . To determine a toolpath, the points in each layer are compared to the points of the damaged surface, starting with the bottom layer. If the  $z$  coordinate of a surface point is lower than the  $z$  coordinate of a layer, material needs to be deposited at the corresponding point in that layer and consequently, the point is added to the toolpath. The points in each layer are processed line by line in the  $y$ -direction, which is an analogue to the cladding procedure of the robot. As a result, the finished toolpath contains the points in the correct order to be processed.

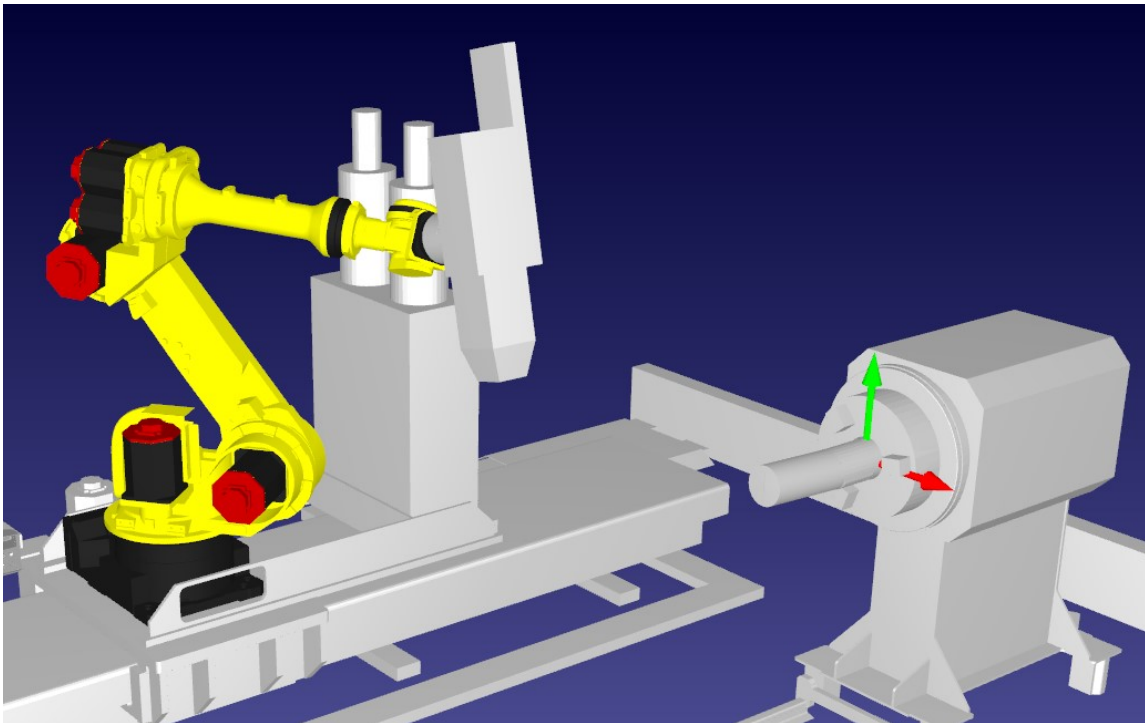


Figure 4.3: Setup of the RLCS in RoboDk software

However, when a part with a low scan resolution needs to be repaired with a fine pitch, to be able to deposit material at the sub-scan resolution appears to be a problem. Sub-scan resolution is any resolution lower than the resolution of the scan. Pitch is the material thickness a machine is capable of depositing. To combat this, a surface function

interpolation method is introduced, which generates data points along the repaired part's longitudinal axis, and interpolation is carried out to get new values of  $z$ . In order to save processing time, the algorithm works by generating values for  $y$  in the range of the previous values in the point cloud, but the step size depends on the machine-specific pitch.

Table 4.1: Algorithm for tool-path generation

---

*Input:* damaged point cloud  $pc_d$ , layer height  $l_h$ , pitch of the machine  $p_m$  and nominal height  $z_{nom}$

1. Interpolate  $pc_d$  in  $y$  to match  $p_m$
2. Find the depth of damaged volume as  $d = z_{nom} - z_{min}$
3. Find the number of layers =  $d/l_h$ , save  $z$  coordinates for all layers
4. **For** each layer, starting at  $z_{min}$ :
  - Divide the layer into  $y$ -lines, i.e., all points of equal  $y$  values
  - For** each line in that layer:
    - For** each point in line:
      - If**  $z < z$  (layer):
        - Add point to list of points for tool-path
      - If**  $z > z$  (layer):
        - Go to the next point

*Output:* List of all points to be included in the tool-path, in the correct order for tool-path planning (i.e., layered from the bottom up and split into  $y$ -lines).

---

### 4.3 Results and Discussion

This paper explores a coherent approach to generating a tool-path for cylindrical components. For this purpose, the experimental setup is shown in Figure 4.4. The robot used is a Fanuc-R-1000iA/80F, which is a high-speed handling robot for medium payloads. As can be seen, the time-of-flight sensor is mounted on the robot arm to ensure consistency in the coordinate system. The substrate is a 4330 steel with an Inconel alloy as the deposition material.

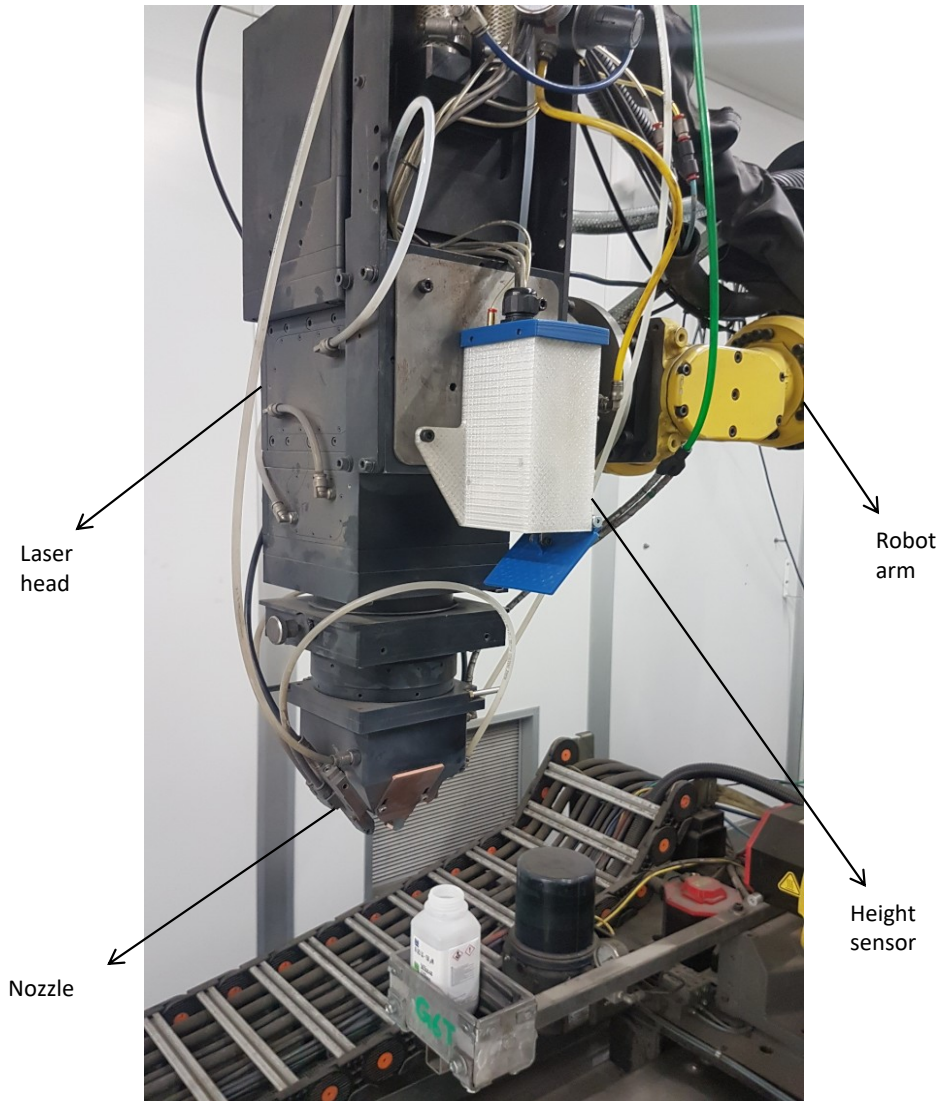


Figure 4.4: Setup of the robot with the scanner

In the proposed system, as explained in Section 4.2.1 point clouds are extracted from a damaged fixed bend. Figure 4.5a illustrates the obtained point clouds of the damaged data. This data is then preprocessed and converted to polar coordinates, as shown in Figure 4.5b.

The converted polar coordinate data is then input into the algorithm discussed earlier. The program works by checking every point and depositing material accordingly. It is

tested with a low-resolution scan of the point cloud that has a spacing of 4mm. This means the distance between each point and its neighboring point is 4mm.

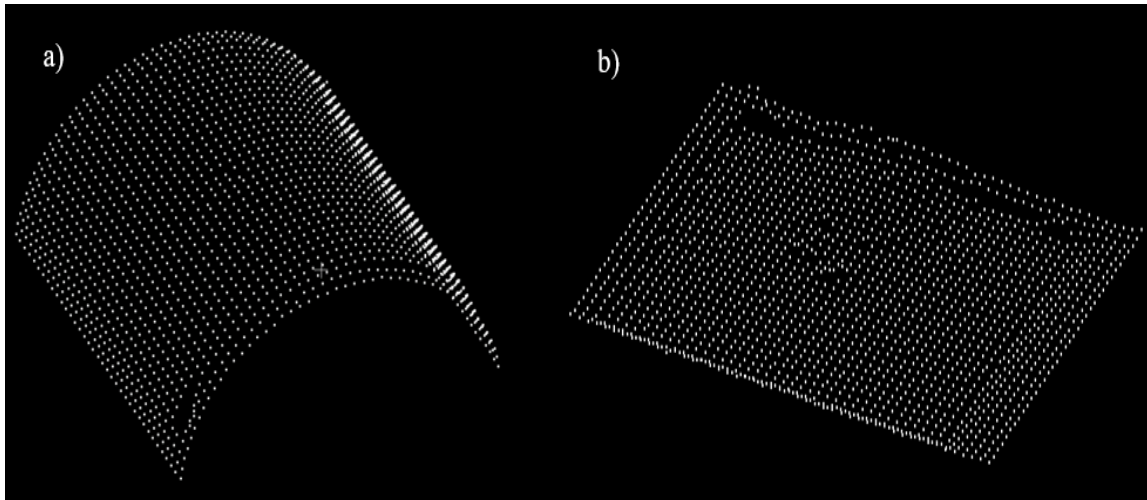


Figure 4.5: Input: a) extracted damaged point cloud b) preprocessed damaged point cloud

The algorithm is tested on a machine with a pitch of 2.8mm. It takes 2.8 seconds on a regular desktop computer to generate a tool-path from this data. The result is depicted by a wireframe model, as shown in Figure 4.6. This wireframe model is a visual portrayal of a 3D object where each vertex or edge is connected by a light blue line. These blue lines give a good graphic of the part's geometry; from the deep valleys and bumps along the line, it is clear the surface geometry is uneven. The red lines illustrate the path the tool will follow whilst depositing material. For a precise clad, the valleys in the model should have red lines crossing them as this is where the material is missing. However, it can be seen clearly from Figure 4.6a how the algorithm results in an imprecise clad as there are deep valleys that have no deposition. This imprecision comes as no surprise because the program works by only checking each point which is a distance of 4mm, whereas the machine is capable of depositing every 2.8mm. The machine resolution is not fully utilized. Nevertheless, following the surface interpolation function mentioned in Section 2.2, the preprocessed polar coordinate points generated are adapted to the machine resolution. This adaptation causes the distance between point clouds to now be 2.8mm. As seen in Figure

4.6b, this results in the material being deposited with a much higher precision, i.e., the material can now be deposited at sub-scan resolution. The subsequent wireframe model has additional red lines covering the valleys, indicating the drastic improvement of the tool-path. It should be noted here that the clad can only be as accurate as the pitch of the machine to avoid overlapping of the clad beads. Therefore, promising results are obtained by this tool-path program, integrated with the surface interpolation function.

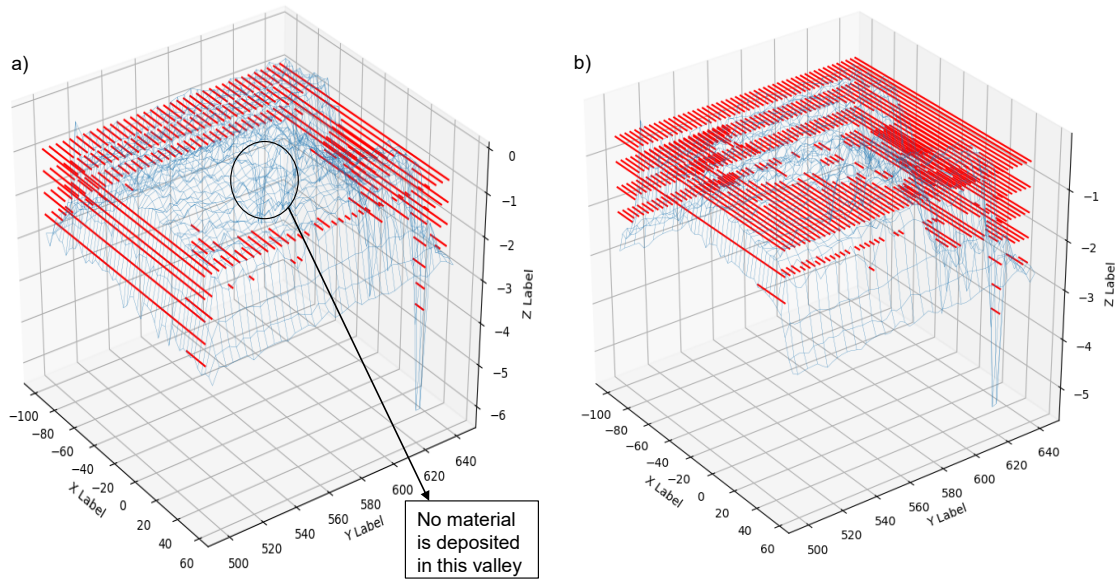


Figure 4.6: Red lines indicate the generated tool-path a) without and b) with surface function interpolation

Moving on, the tool-path generated with the proposed algorithm is used to carry out the repair of fixed bends on the machine demonstrated in Figure 4.4. The process parameters are a power of 3 KW, powder feed rate of 30 gm/min and cladding speed of 480 mm/min. The tolerance, i.e., the allowable height deviation between each layer is  $\pm 0.1$  mm and the maximum accuracy is bounded by the layer thickness, which depends on the system used, is to be no more than 1 mm over the expected buildup height. This is ensured by recording the coordinate values on the surface with a time-of-flight sensor. Figure 4.7 shows the condition of a fixed bend before and after repair and the results meet the expected quality and tolerance. While significantly reducing the effort of registration-based approach and

not requiring the nominal model as an input. These results validate the general functionality of the method proposed in this paper.

a)



b)

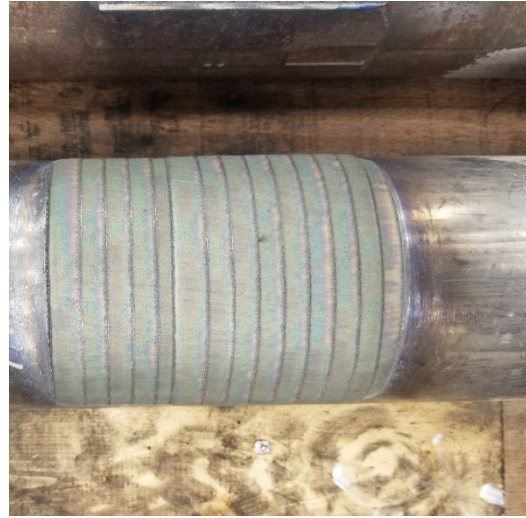


Figure 4.7: a) Before repair and b) after repair carried out by the proposed algorithm

## 4.4 Conclusion

LMD process is extensively used for repairing mechanical components, the vast majority of which have cylindrical geometry. Automation of this repair process is important as manual repair produces varying quality and is labor-intensive. From a thorough analysis of current literature in remanufacturing, the generation of a tool-path directly from point cloud data was identified as the first persisting problem. The problem was resolved for cylindrical components by setting up an RLCS that uses a novel approach to scan and preprocess only the damaged point cloud data. In the presented method, the input point cloud data was preprocessed by changing the coordinate system from cartesian to polar coordinates. In doing so, the cylindrical geometry was converted to a flat plan. This data was then fed into the tool-path algorithm and the component was rebuilt while making sure the consistency of the coordinate system is maintained, as opposed to traditional



approaches that involve several intermediate steps. The second challenge was adapting the resolution of the point cloud data based on the LC system. Getting a high-quality scan is a time-consuming process, but it is crucial for a high-quality repair. This was solved by integrating a surface interpolation function in the tool-path algorithm, which considers low-resolution scan and is able to adapt the input data according to that of the RLCS. The remanufacturing framework proposed in this paper significantly reduces the total time of repair. For context, the tool-path algorithm ran in 2.8 seconds without requiring notable computational power, whereas an intermediate step of registration took Li et al. [96] 1,713 seconds to run. Therefore, these favorable results demonstrate the practicality of the proposed approach.

# Chapter 5: Conclusion, Discussion & Future Work

## 5.1 Conclusions

In pursuit of a more sustainable manufacturing sector, the concept of a circular economy model is very relevant. Remanufacturing and repair technologies make the circular model possible by restoring value in damaged components so that they can be reused again. Not only are they environmentally friendly approaches, but they also offer significant economic advantages as compared to manufacturing a new part. Currently, a general repair framework is reliant on a human operator to acquire three-dimensional point cloud information of a damaged component. Several post-processing steps are required in order to extract the damaged volume. Based on the repair patch information, an adequate tool-path is generated that is deposited by a robotic repair system to restore the damaged part's geometry. As the volume of parts and the part size increases, the process gets more time-consuming and tedious. Since the method is reliant on a human operator, the monotonousness of the process can lead to inconsistent results. Additionally, to keep up with the competition arising from industries transitioning to Industry 4.0, exploring alternative autonomous repair strategies is important.

The research presented aims to investigate the integration of deep learning and computer vision technologies into a robotic repair process. The focus lies on cylindrical components since their geometry is among the most common in the manufacturing industry. A novel framework is proposed that enables an automatic repair of damaged fixed bends in an RLCRC. This method achieves automatic damage detection and volume extraction since the two most vital pieces of data during a repair process are the damaged region's location and the volume of damage.

To achieve automatic damage detection, a computer vision-based damage inspection system is designed that collects image data of a damaged part in the repair cell. Sets of images are run through a trained deep learning model (Faster R-CNN) that localizes the damaged region on an image. This model's output is a damage location in the form of 2D

pixel coordinates of a bounding box on the image. Several Faster R-CNN models are trained with varying feature extractors and datasets to obtain a model that is best suited for inspection of worn components in the RLCRC.

Moving on, the vision system in the RLCRC is calibrated with respect to a known global coordinate origin. In conjunction with the deep learning model, this calibrated system is able to automatically detect and quantify the damaged region on the surface of the worn component that is mounted in the repair cell. By means of the calibration parameters, 2D pixel coordinates are converted to 3D spatial coordinates containing the damaged region's location in three-dimensional space. Since the system is calibrated, the robot arm fitted with a ToF sensor can move to the specific location of the damage and acquire volumetric information of the damaged region. A deposition tool-path is generated by converting the acquired point cloud data from cartesian coordinates to polar coordinates.

The proposed system is validated through case studies and is found to obtain sufficient accuracy. Intelligent technologies are used to automatically extract the three-dimensional information of only the damaged volume instead of the entire part surface. By doing so, time-consuming and labor-intensive intermediate steps are eliminated and a tool-path is generated directly from the damaged point cloud data. The presented framework makes it faster and more efficient to repair cylindrical fixed bends.

## **5.2 Research Contributions**

The research presented in this thesis explores the use of cutting-edge technologies for repairing damaged fixed bends. Overall, the work displayed in this thesis endorses the automation of fixed bend components. The main contributions of this research are as follows:

- Proposed a novel framework for repairing damaged fixed bend components automatically and with improved time efficiency.

- Implemented a computer vision-based setup into an existing robotic laser cladding system and calibrated the system with respect to a global origin. Validation on the calibrated system was performed by comparing measured (real world) values to calculated (calibration system output) values and a relative error of 2.018% was reported between them.
- Trained a deep learning-based damage detection and classification model by using Faster R-CNN with varying feature extractors and training datasets. Analyzed the impact of a training dataset on the accuracy of a trained model by performing two comparative case studies. ResNet50 outperformed Inception v2 in both case studies. Case study 1 was trained on a diverse dataset, first for two labels 'pad' and 'damage' and then for one label 'pad', achieving a mAP of 52.8% and 70.4%, respectively. For case study 2, a new dataset was created containing images with less variation between them, aiming to achieve a higher-performing model for the RLCRC setup. This new model was trained with one label 'pad' and achieved a mAP of 88.7%.
- Developed a method for converting 2D pixel coordinates from images obtained with the designed computer vision system to 3D world coordinates on the surface of cylindrical components. Since the damaged region wraps around the fixed bend, an algorithm was created to obtain the damaged area's axial and angular coordinates. Two case studies were carried out to verify the proposed method and an error analysis between the measured and calculated values of longitudinal and angular coordinates was performed. An average axial and angular error of 5.04 mm and 0.564° was found, respectively. Additionally, a reduction in scan time of more than 63% was achieved by only scanning the damaged region instead of the entire surface of the component.
- Proposed an efficient way of generating a tool-path directly from damaged point cloud data by converting cartesian coordinates to polar coordinates. In a case study performed for a sample geometry, the tool-path is computed within 2.8 seconds by a regular office computer. The generated tool-path meets the tolerance

requirements of +/- 0.1 mm height deviation between each layer and an accuracy requirement of being less than 1mm over the expected build-up height. Furthermore, an interpolation method was implemented that enables the point cloud data to be fitted to any deposition step size of the laser cladding system, making the algorithm adaptable to the resolution of different machines. This results in a more precise material deposition that is truer to the damaged area's real geometry. Regardless of their resolution, all point clouds were preprocessed to adapt to the machine resolution resulting in material being deposited with a higher precision.

### **5.3 Limitations and Future Work**

Despite the efficacious achievement of its goals, the research presented in this thesis is subject to the following limitations that should be tackled by the work of future researchers:

- Limited size of the dataset is often a big constraint when using deep learning models for object detection purposes. The confidence with which a model can accurately recognize features is based on the amounts of data it ‘sees’ and learns from. For some fixed bends, the ‘pad’ feature is not easily distinguishable from the rest of the surface, even to a human operator. This is an even bigger challenge for a vision system due to the added illumination and light reflection from the component’s metallic surface. Results obtained in Chapter 2: indicate that a large amount of annotated data is required to achieve a robust model. Acquiring that data and labeling it are both tedious and costly processes. However, to adapt the current framework to encompass all prospective fixed bend repair processes, further training with a large dataset is required. Furthermore, different machine learning approaches should be explored to find a more computationally efficient way of detecting damages.
- In this research, two case studies are carried out in Section 2.3 with varying datasets to obtain a suitable deep learning model for defect detection in fixed bends. Case

study 1 looks at two scenarios by keeping all parameters constant except the number of labels or the variability of objects that the model needs to detect. An improvement in *mAP* of 30.33% is achieved in the second scenario by reducing the number of labels. For the second case study, the dataset contains images with less variation in magnification, angle and orientation of the fixed bend images as compared to the first case study. This improves the *mAP* performance by almost 70% as compared to the first case study with two labels. A more comprehensive robustness analysis should be performed by varying these parameters (i.e., angle, orientation, magnification and variability of defects) independently to gain a deeper understanding on what factors the model is most sensitive to.

- Manual measurements of the distance between the camera lens and the ToF sensor are taken that might be flawed due to the spherical shape of the lens. This approach relies on calibration parameters that are manually measured and are used in the algorithm to obtain spatial coordinates of the damage. A faulty calibration can lead to the model having an inherent bias. In the future, having a setup designated for repairing fixed bends requires gaining insight on specific areas of improvement by performing additional tests.
- A fixed bend with a known radius is mounted on the turntable during the calibration process. The calibration model of the RLCRC is sensitive to distance in the radial direction. This means that for all parts with similar radii, results within the precision tolerance can be obtained. However, when high variability of the radius occurs, the calibration results might lack precision and the RLCRC will need to be calibrated again. For this thesis, varying radii are not a problem as there are only two different fixed bend diameters that are considered and they both achieved good calibration results. This can be due to the radius of both parts being relatively close to each other, within 30 mm. Nevertheless, more tests to quantify the tolerance for varying radii should be performed. For instance, for what range of radii can one setup be used while remaining under a specified error tolerance.

- During an inspection process, the position of the camera with respect to the origin remains unchanged. This means that the vision system only has a limited field of view. For a fixed bend with damage on only the ‘pad’ region, this system is able to detect damages as the position of the pad is roughly anticipated. However, for a more generalized cylindrical component that can have damage anywhere on its surface, it will be important to consider a system where the camera can move during the inspection. It will also be interesting to explore the possibility of inspecting and repairing cylindrical components that possess multiple damages along the surface. A much deeper understanding of calibration will be crucial for implementing such a flexible vision system.
- Currently, angular coordinates of the damaged region are obtained using frame numbers from a recorded video. Relying on the motion sensor capabilities to obtain a complete rotation video might not be an efficient method as any small movement can trigger the video recording. Different post-processing methods for trimming the video to only include a full 360° rotation should be explored to consider a factory environment. Additionally, videos filmed with the part rotating at different angular velocities can be tested to analyze the effect of it on the accuracy of obtaining the angular coordinates.
- A method for filtering false detections during inference needs to be implemented in the spatial localization algorithm to ensure that the algorithm is robust against errors in the deep learning model. The current localization algorithm in Section 3.2.3 is sensitive to outliers in detection.
- The repair framework proposed in this work is tailored to repairing fixed bends with a known nominal radius. In actuality, that information might not always be available. Furthermore, different methods should be analyzed to extend this repair framework to account for a more generalized rather than a specialized repair pipeline. For instance, other cylindrical components should be tested rather than repairing only fixed bends, and their performance noted. Based on the performance, modifications to this existing framework can be made.

- Currently, the repair framework works automatically. However, the framework can be extended to achieve a fully autonomous repair. For instance, after the operator mounts the damaged component in the repair cell, the vision system will have the capability to detect the damaged region, localize it and send the 3D damage coordinates to the robot. The robot arm will then move to the damage location and acquire the repair patch's three-dimensional point cloud data. All of these steps can be performed without human intervention but require a deeper integration of the visual sensors into the robot environment and significant modifications to the repair framework's logic.
- It is important to note here that the tool-path method leverages off a cylindrical coordinate system and testing the sensitivity of the algorithm for non-cylindrical components is currently out of the scope of this research. However, experimenting with geometries similar to cylinders should be a possibility and an interesting idea to explore. In the future, the methodology can be extended further by varying the scanning speed for the deposition of one track. Scanning speed refers to the speed at which the deposition medium moves while depositing material onto the surface. Thereby, the rate of material deposition can be controlled based on the geometry and depth of the wear. In the same pass, more material can be deposited in deeper valleys while depositing less material on the edges.
- Utilizing the computer vision system setup in this study, the cladding process can be monitored in real-time and a feedback loop can be created that is capable of adjusting the cladding parameters i.e., laser power, cladding speed, powder feed rate during the process. A deep learning model can be trained to output suitable parameters based on the video feed from the vision system that account for minimal defect formation and high clad quality. Relevant material science knowledge will be required to prepare the criteria for adjusting the clad parameters.
- To truly embrace the concept of continuous learning, it will also be important to continually improve the repair framework. Based on the resulting repaired component, the individual repair steps can be adjusted to improve the overall



quality and efficiency of the repair process. This would entail the following: 1) training the deep learning model periodically with new data; 2) optimizing the damage localization algorithm to extract an exact repair volume; and 3) optimizing the tool-path generation algorithm to reduce material wastage and improve deposition efficiency. By continuously learning from its missteps and failures, the framework can remain up to date with the changes in the parts that are repaired.

The illustration in Figure 5.1 clearly highlights the aforementioned future work with a dotted red line. This figure is an extension of the framework flowchart presented in Section 1.4. Here six main areas that can be improved in the future are: 1) training the deep learning model with more data; 2) improving the spatial localization algorithm by filtering frames from the deep learning model that contain outliers; 3) realizing the vision system to achieve real-time monitoring and control the cladding process to adjust the process parameters instantaneously; 4) generalizing the repair framework to include all cylindrical components; 5) implementing repair in an autonomous environment; and 6) continuously improving the repair framework by learning from the resulting repaired component and adjusting the process accordingly.

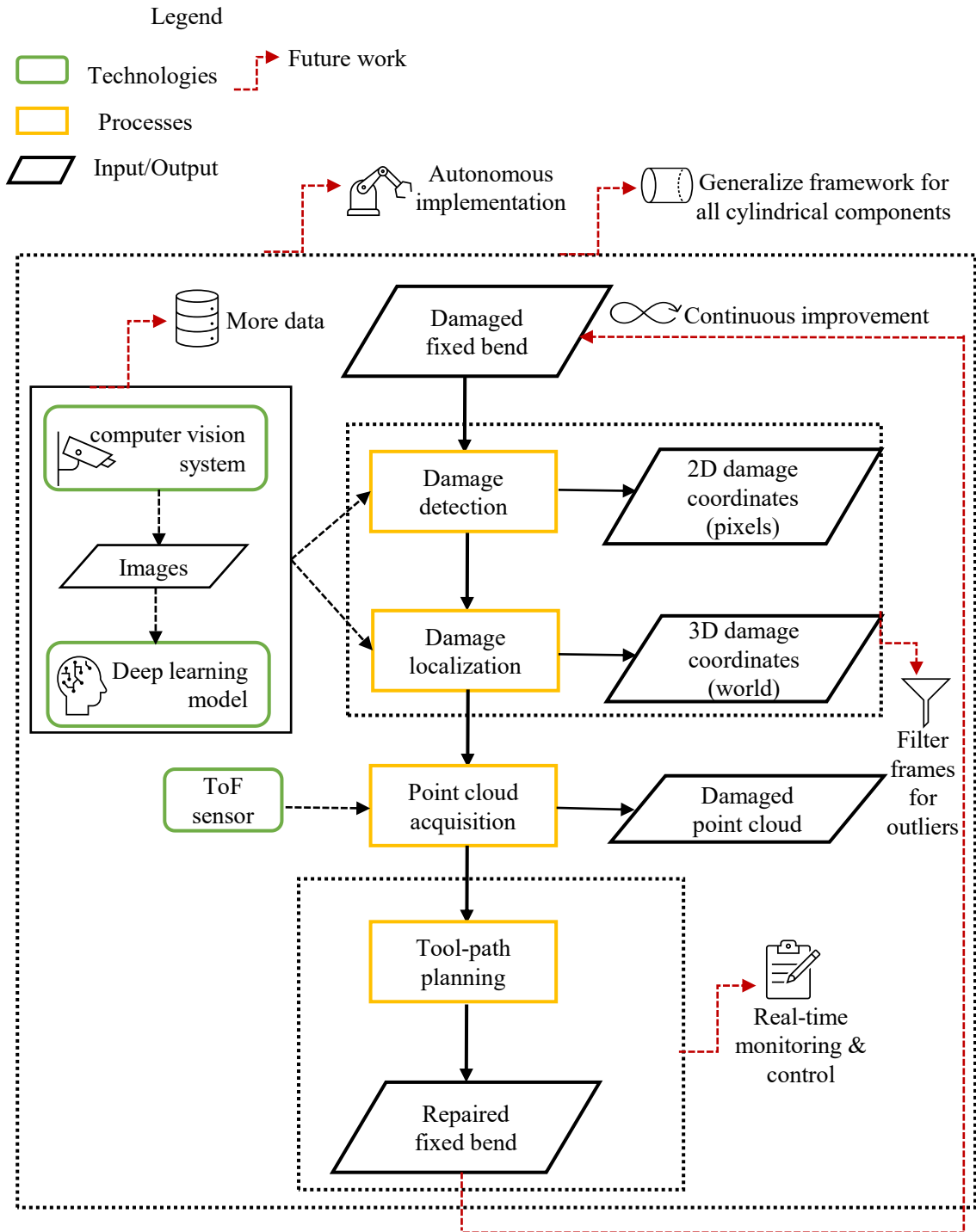


Figure 5.1: Proposed repair framework with future work plan

## Bibliography

- [1] C. Garcés-Ayerbe, P. Rivera-Torres, I. Suárez-Perales, and D. I. L. D. La Hiz, “Is it possible to change from a linear to a circular economy? An overview of opportunities and barriers for european small and medium-sized enterprise companies,” *Int. J. Environ. Res. Public Health*, vol. 16, no. 5, Mar. 2019, doi: 10.3390/ijerph16050851.
- [2] M. Charter and C. Gray, “Remanufacturing and product design,” *Int. J. Prod. Dev.*, vol. 6, no. 3–4, pp. 375–392, 2008, doi: 10.1504/IJPD.2008.020406.
- [3] J. Östlin, “On Remanufacturing Systems: Analysing and Managing Material Flows and Remanufacturing Processes,” Institutionen för ekonomisk och industriell utveckling, Linköping, 2008.
- [4] N. Nasr *et al.*, “Re-defining Value – The Manufacturing Revolution. Remanufacturing, Refurbishment, Repair and Direct Reuse in the Circular Economy,” Nairobi, Kenya, 2018.
- [5] J. Cao, X. Chen, X. Zhang, Y. Gao, X. Zhang, and S. Kumar, “Overview of remanufacturing industry in China: Government policies, enterprise, and public awareness,” *J. Clean. Prod.*, vol. 242, p. 118450, Jan. 2020, doi: 10.1016/j.jclepro.2019.118450.
- [6] C. M. Lee, W. S. Woo, and Y. H. Roh, “Remanufacturing: Trends and issues,” *Int. J. Precis. Eng. Manuf. - Green Technol.*, vol. 4, no. 1, pp. 113–125, Jan. 2017, doi: 10.1007/s40684-017-0015-0.
- [7] A. Kumar, R. B. Chinnam, and A. Murat, “Hazard rate models for core return modeling in auto parts remanufacturing,” *Int. J. Prod. Econ.*, vol. 183, pp. 354–361, Jan. 2017, doi: 10.1016/j.ijpe.2016.07.002.
- [8] Q. Liu, M. Janardhana, B. Hinton, M. Brandt, and K. Sharp, “Laser cladding as a

- potential repair technology for damaged aircraft components,” *Int. J. Struct. Integr.*, vol. 2, no. 3, pp. 1757–9864, 2011, doi: 10.1108/17579861111162914.
- [9] J. Zheng, Z. Li, and X. Chen, “Worn area modeling for automating the repair of turbine blades,” *Int. J. Adv. Manuf. Technol.*, vol. 29, no. 9–10, pp. 1062–1067, 2006, doi: 10.1007/s00170-003-1990-6.
- [10] M. Attene and G. Patanè, “Hierarchical Structure Recovery of Point-Sampled Surfaces,” *Comput. Graph. Forum*, vol. 29, no. 6, pp. 1905–1920, Sep. 2010, doi: 10.1111/j.1467-8659.2010.01658.x.
- [11] T. T. Tran, V. T. Cao, and D. Laurendeau, “Extraction of cylinders and estimation of their parameters from point clouds,” *Comput. Graph.*, vol. 46, pp. 345–357, 2015, doi: 10.1016/j.cag.2014.09.027.
- [12] Z. Liu *et al.*, “Environmental benefits of remanufacturing: A case study of cylinder heads remanufactured through laser cladding,” *J. Clean. Prod.*, vol. 133, pp. 1027–1033, 2016, doi: 10.1016/j.jclepro.2016.06.049.
- [13] T. Ma, P. Chen, and J. Zhao, “Overview on vertical and directional drilling technologies for the exploration and exploitation of deep petroleum resources,” *Geomechanics and Geophysics for Geo-Energy and Geo-Resources*, vol. 2, no. 4. Springer, pp. 365–395, 01-Dec-2016, doi: 10.1007/s40948-016-0038-y.
- [14] C. Leyens and E. Beyer, “Innovations in laser cladding and direct laser metal deposition,” in *Laser Surface Engineering: Processes and Applications*, Elsevier Inc., 2015, pp. 181–192.
- [15] M. Gharbi *et al.*, “Influence of a pulsed laser regime on surface finish induced by the direct metal deposition process on a Ti64 alloy,” *J. Mater. Process. Technol.*, vol. 214, no. 2, pp. 485–495, 2014, doi: 10.1016/j.jmatprotec.2013.10.004.
- [16] E. Brandl, A. Schoberth, and C. Leyens, “Morphology, microstructure, and hardness of titanium (Ti-6Al-4V) blocks deposited by wire-feed additive layer

- manufacturing (ALM),” *Mater. Sci. Eng. A*, vol. 532, pp. 295–307, 2012, doi: 10.1016/j.msea.2011.10.095.
- [17] W. W. Liu, Z. J. Tang, X. Y. Liu, H. J. Wang, and H. C. Zhang, “A Review on In-situ Monitoring and Adaptive Control Technology for Laser Cladding Remanufacturing,” *Procedia CIRP*, vol. 61, pp. 235–240, 2017, doi: 10.1016/j.procir.2016.11.217.
- [18] E. Uhlmann, F. Heitmüller, M. Manthei, and S. Reinkober, “Applicability of industrial robots for machining and repair processes,” *Procedia CIRP*, vol. 11, pp. 234–238, 2013, doi: 10.1016/j.procir.2013.07.042.
- [19] Y. Ding, R. Dwivedi, and R. Kovacevic, “Process planning for 8-axis robotized laser-based direct metal deposition system: A case on building revolved part,” *Robot. Comput. Integr. Manuf.*, vol. 44, pp. 67–76, 2017, doi: 10.1016/j.rcim.2016.08.008.
- [20] H. Zheng, M. Cong, H. Dong, Y. Liu, and D. Liu, “CAD-based automatic path generation and optimization for laser cladding robot in additive manufacturing,” *Int. J. Adv. Manuf. Technol.*, vol. 92, no. 9–12, pp. 3605–3614, 2017, doi: 10.1007/s00170-017-0384-0.
- [21] H. Z. Imam, Y. Zheng, and R. Ahmad, “An efficient tool-path planning approach for repair of cylindrical components via laser cladding,” *J. Remanufacturing*, vol. 10, no. 3, pp. 1–10, Nov. 2020, doi: 10.1007/s13243-020-00096-6.
- [22] X. Huang, Z. Liu, X. Zhang, J. Kang, M. Zhang, and Y. Guo, “Surface damage detection for steel wire ropes using deep learning and computer vision techniques,” *Meas. J. Int. Meas. Confed.*, vol. 161, 2020, doi: 10.1016/j.measurement.2020.107843.
- [23] E. McLaughlin, N. Charron, and S. Narasimhan, “Combining deep learning and robotics for automated concrete delamination assessment,” in *Proceedings of the*

*36th International Symposium on Automation and Robotics in Construction, ISARC 2019*, 2019, pp. 485–492, doi: 10.22260/isarc2019/0065.

- [24] W. Cai, J. Z. Wang, P. Jiang, L. C. Cao, G. Y. Mi, and Q. Zhou, “Application of sensing techniques and artificial intelligence-based methods to laser welding real-time monitoring: A critical review of recent literature,” *J. Manuf. Syst.*, vol. 57, no. May, pp. 1–18, 2020, doi: 10.1016/j.jmsy.2020.07.021.
- [25] P. Pant and D. Chatterjee, “Prediction of clad characteristics using ANN and combined PSO-ANN algorithms in laser metal deposition process,” *Surfaces and Interfaces*, vol. 21, no. July, p. 100699, 2020, doi: 10.1016/j.surfin.2020.100699.
- [26] C. Xia *et al.*, “A review on wire arc additive manufacturing: Monitoring, control and a framework of automated system,” *J. Manuf. Syst.*, vol. 57, no. August, pp. 31–45, 2020, doi: 10.1016/j.jmsy.2020.08.008.
- [27] Y. Zhang, G. S. Hong, D. Ye, K. Zhu, and J. Y. H. Fuh, “Extraction and evaluation of melt pool, plume and spatter information for powder-bed fusion AM process monitoring,” *Mater. Des.*, vol. 156, pp. 458–469, 2018, doi: 10.1016/j.matdes.2018.07.002.
- [28] P. Martinez, M. Al-Hussein, and R. Ahmad, “Intelligent vision-based online inspection system of screw-fastening operations in light-gauge steel frame manufacturing,” *Int. J. Adv. Manuf. Technol.*, vol. 109, no. 3–4, pp. 645–657, 2020, doi: 10.1007/s00170-020-05695-y.
- [29] A. Reyes-Yanes, P. Martinez, and R. Ahmad, “Real-time growth rate and fresh weight estimation for little gem romaine lettuce in aquaponic grow beds,” *Comput. Electron. Agric.*, vol. 179, 2020, doi: <https://doi.org/10.1016/j.compag.2020.105827>.
- [30] Y. Zheng, H. Mamledesai, H. Imam, and R. Ahmad, “Deep Learning-based Automatic Damage Recognition and Spatial Localization for

- Remanufacturing/Repair,” vol. 18, no. 6, pp. 381–385, 2021, doi: 10.14733/cadconfP.2020.381-385 Introduction:
- [31] Y. J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, “Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types,” *Comput. Civ. Infrastruct. Eng.*, vol. 33, no. 9, pp. 731–747, 2018, doi: 10.1111/mice.12334.
- [32] C. Nwankpa, S. Eze, W. Ijomah, A. Gachagan, and S. Marshall, “Achieving remanufacturing inspection using deep learning,” *J. Remanufacturing*, vol. 10, no. 3, 2020, doi: 10.1007/s13243-020-00093-9.
- [33] J. Zhong, T. Lei, and G. Yao, “Robust Vehicle Detection in Aerial Images Based on Cascaded Convolutional Neural Networks,” *Sensors*, vol. 17, no. 12, p. 2720, Nov. 2017, doi: 10.3390/s17122720.
- [34] S. Panigrahi, A. Nanda, and T. Swarnkar, “A Survey on Transfer Learning,” *Smart Innov. Syst. Technol.*, vol. 194, no. 10, pp. 781–789, 2021, doi: 10.1007/978-981-15-5971-6\_83.
- [35] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 580–587, 2014, doi: 10.1109/CVPR.2014.81.
- [36] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.
- [37] Á. Arcos-García, J. A. Álvarez-García, and L. M. Soria-Morillo, “Evaluation of deep neural networks for traffic sign detection systems,” *Neurocomputing*, vol. 316, pp. 332–344, 2018, doi: 10.1016/j.neucom.2018.08.009.
- [38] C. Lee, H. J. Kim, and K. W. Oh, “Comparison of faster R-CNN models for object

- detection,” *Int. Conf. Control. Autom. Syst.*, vol. 0, no. Iccas, pp. 107–110, 2016, doi: 10.1109/ICCAS.2016.7832305.
- [39] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.
- [40] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, vol. 07-12-June-2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.
- [41] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
- [42] T. Y. Lin *et al.*, “Microsoft COCO: Common objects in context,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8693 LNCS, no. PART 5, pp. 740–755, doi: 10.1007/978-3-319-10602-1\_48.
- [43] D. Kang, S. S. Benipal, D. L. Gopal, and Y. J. Cha, “Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning,” *Autom. Constr.*, vol. 118, no. May, 2020, doi: 10.1016/j.autcon.2020.103291.
- [44] Z. Guo *et al.*, “Defect detection of nuclear fuel assembly based on deep neural network,” *Ann. Nucl. Energy*, vol. 137, 2020, doi: 10.1016/j.anucene.2019.107078.
- [45] T. Y. Lin *et al.*, “Microsoft COCO: Common objects in context,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8693 LNCS, no. PART 5, pp. 740–755, doi: 10.1007/978-3-319-10602-1\_48.
- [46] Tzutalin, “LabelImg,” *Git code*, 2015. [Online]. Available:



<https://github.com/tzutalin/labelImg>.

- [47] C. M. Lee, W. S. Woo, and Y. H. Roh, “Remanufacturing: Trends and issues,” *Int. J. Precis. Eng. Manuf. - Green Technol.*, vol. 4, no. 1, pp. 113–125, Jan. 2017, doi: 10.1007/s40684-017-0015-0.
- [48] E. Westkämper, “Strategic Development of Factories under the Influence of Emergent Technologies,” *CIRP Ann. - Manuf. Technol.*, vol. 56, no. 1, pp. 419–422, Jan. 2007, doi: 10.1016/j.cirp.2007.05.100.
- [49] R. French, M. Benakis, and H. Marin-Reyes, “Intelligent sensing for robotic re-manufacturing in aerospace - An industry 4.0 design based prototype,” in *Proceedings - 2017 IEEE 5th International Symposium on Robotics and Intelligent Sensors, IRIS 2017*, 2018, pp. 272–277, doi: 10.1109/IRIS.2017.8250134.
- [50] M. Kerin and D. T. Pham, “A review of emerging industry 4.0 technologies in remanufacturing,” *J. Clean. Prod.*, vol. 237, p. 117805, 2019, doi: 10.1016/j.jclepro.2019.117805.
- [51] Y. Zheng, H. Mamledesai, H. Imam, and R. Ahmad, “A Novel Deep Learning-based Automatic Damage Detection and Localization Method for Remanufacturing/Repair,” *Comput. Des. Appl.*, vol. 18, no. 6, pp. 1359–1372, 2021, doi: 10.14733/cadaps.2021.1359-1372.
- [52] X. Zhang, W. Li, X. Chen, W. Cui, and F. Liou, “Evaluation of component repair using direct metal deposition from scanned data,” *Int. J. Adv. Manuf. Technol.*, vol. 95, no. 9–12, pp. 3335–3348, 2018, doi: 10.1007/s00170-017-1455-y.
- [53] X. Zhang, W. Li, and F. Liou, “Damage detection and reconstruction algorithm in repairing compressor blade by direct metal deposition,” *Int. J. Adv. Manuf. Technol.*, vol. 95, no. 5–8, pp. 2393–2404, 2018, doi: 10.1007/s00170-017-1413-8.
- [54] I. Jovančević *et al.*, “3D Point Cloud Analysis for Detection and Characterization of Defects on Airplane Exterior Surface,” *J. Nondestruct. Eval.*, vol. 36, no. 4, 2017,

doi: 10.1007/s10921-017-0453-1.

- [55] J. Ma, J. Zhao, and A. L. Yuille, “Non-rigid point set registration by preserving global and local structures,” *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 53–64, 2016, doi: 10.1109/TIP.2015.2467217.
- [56] X. Zhang, W. Cui, and F. Liou, “Voxel-Based Geometry Reconstruction for Repairing and Remanufacturing of Metallic Components Via Additive Manufacturing,” *Int. J. Precis. Eng. Manuf. - Green Technol.*, no. 0123456789, pp. 291–297, 2021, doi: 10.1007/s40684-020-00291-7.
- [57] B. Bellekens, V. Spruyt, and M. Weyn, “A Survey of Rigid 3D Pointcloud Registration Algorithms,” *Fourth Int. Conf. Ambient Comput. Appl. Serv. Technol. Ambient*, no. c, pp. 8–13, 2014, doi: 10.1021/ol901113t.
- [58] A. Aprilia, W. L. K. Nguyen, A. Khairyanto, W. C. Pang, S. B. Tor, and G. Seet, “Towards automated remanufacturing process with additive manufacturing,” *Proc. Int. Conf. Prog. Addit. Manuf.*, vol. 2018-May, pp. 696–701, 2018, doi: 10.25341/D4SW2N.
- [59] A. Caggiano and R. Teti, “Digital factory technologies for robotic automation and enhanced manufacturing cell design,” *Cogent Eng.*, vol. 5, no. 1, p. 1426676, Jan. 2018, doi: 10.1080/23311916.2018.1426676.
- [60] F. Blais, “Review of 20 years of range sensor development,” *J. Electron. Imaging*, vol. 13, no. 1, p. 231, Jan. 2004, doi: 10.1117/1.1631921.
- [61] S. Byun, K. Jung, S. Im, and M. Chang, “Registration of 3D Scan Data Using Image Reprojection,” *Int. J. Precis. Eng. Manuf.*, vol. 18, no. 9, pp. 1221–1229, 2017, doi: 10.1007/s12541-017-0143-z.
- [62] F. Fooladgar and S. Kasaei, “A survey on indoor RGB-D semantic segmentation: from hand-crafted features to deep convolutional neural networks,” *Multimed. Tools Appl.*, vol. 79, no. 7–8, pp. 4499–4524, Feb. 2020, doi: 10.1007/s11042-019-

7684-3.

- [63] X. Song, J. Zheng, F. Zhong, and X. Qin, “Modeling deviations of rgb-d cameras for accurate depth map and color image registration,” *Multimed. Tools Appl.*, vol. 77, no. 12, pp. 14951–14977, Jun. 2018, doi: 10.1007/s11042-017-5081-3.
- [64] S. Song and J. Xiao, “Deep sliding shapes for amodal 3D object detection in RGB-D images,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 808–816, doi: 10.1109/CVPR.2016.94.
- [65] K. Khoshelham and S. O. Elberink, “Accuracy and resolution of kinect depth data for indoor mapping applications,” *Sensors*, vol. 12, no. 2, pp. 1437–1454, Feb. 2012, doi: 10.3390/s120201437.
- [66] F. Castaldo and F. A. N. Palmieri, “Camera system: pinhole model, calibration and reconstruction,” 2013.
- [67] Z. Wang, R. Liu, T. Sparks, H. Liu, and F. Liou, “Stereo vision based hybrid manufacturing process for precision metal parts,” *Precis. Eng.*, vol. 42, pp. 1–5, Oct. 2015, doi: 10.1016/j.precisioneng.2014.11.012.
- [68] R. Liu, Z. Wang, T. Sparks, F. Liou, and C. Nedic, “Stereo vision-based repair of metallic components,” *Rapid Prototyp. J.*, vol. 23, no. 1, pp. 65–73, 2017, doi: 10.1108/RPJ-09-2015-0118.
- [69] Z. Tian, L. Jianhua, L. Shaoli, T. Chengtong, and J. Peng, “A 3D reconstruction method for pipeline inspection based on multi-vision,” *Meas. J. Int. Meas. Confed.*, vol. 98, pp. 35–48, Feb. 2017, doi: 10.1016/j.measurement.2016.11.004.
- [70] M. Bitzidou, D. Chrysostomou, and A. Gasteratos, “Multi-camera 3D object reconstruction for industrial automation,” in *IFIP Advances in Information and Communication Technology*, 2013, vol. 397, no. PART 1, pp. 526–533, doi: 10.1007/978-3-642-40352-1\_66.
- [71] Q. Peng, L. Tu, K. Zhang, and S. Zhong, “Automated 3D scenes reconstruction

using multiple stereo pairs from portable four-camera photographic measurement system,” *Int. J. Opt.*, vol. 2015, 2015, doi: 10.1155/2015/471681.

- [72] D. Chrysostomou, N. Kyriakoulis, and A. Gasteratos, “Multi-camera 3D scene reconstruction from vanishing points,” in *2010 IEEE International Conference on Imaging Systems and Techniques, IST 2010 - Proceedings*, 2010, pp. 343–348, doi: 10.1109/IST.2010.5548495.
- [73] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “MonoSLAM: Real-time single camera SLAM,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007, doi: 10.1109/TPAMI.2007.1049.
- [74] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, “DTAM: Dense tracking and mapping in real-time,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2320–2327, doi: 10.1109/ICCV.2011.6126513.
- [75] A. Khan, C. Mineo, G. Dobie, C. Macleod, and G. Pierce, “Vision guided robotic inspection for parts in manufacturing and remanufacturing industry,” *J. Remanufacturing*, 2020, doi: 10.1007/s13243-020-00091-x.
- [76] T. Taketomi, H. Uchiyama, and S. Ikeda, “Visual SLAM algorithms: A survey from 2010 to 2016,” *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1. Springer, p. 16, 02-Dec-2017, doi: 10.1186/s41074-017-0027-2.
- [77] C. S. Lin, Y. C. Huang, S. H. Chen, Y. L. Hsu, and Y. C. Lin, “The application of deep learning and image processing technology in laser positioning,” *Appl. Sci.*, vol. 8, no. 9, 2018, doi: 10.3390/app8091542.
- [78] W. Yang, Z. Li, C. Wang, and J. Li, “A multi-task Faster R-CNN method for 3D vehicle detection based on a single image,” *Appl. Soft Comput. J.*, vol. 95, 2020, doi: 10.1016/j.asoc.2020.106533.
- [79] G. H. Beckman, D. Polyzois, and Y. J. Cha, “Deep learning-based automatic volumetric damage quantification using depth camera,” *Autom. Constr.*, vol. 99, no.

November 2018, pp. 114–124, 2019, doi: 10.1016/j.autcon.2018.12.006.

- [80] L. Smith and P. Ball, “Steps towards sustainable manufacturing through modelling material, energy and waste flows,” *Int. J. Prod. Econ.*, vol. 140, no. 1, pp. 227–238, 2012, doi: 10.1016/j.ijpe.2012.01.036.
- [81] J. Lee, H. Son, C. Kim, and C. Kim, “Skeleton-based 3D reconstruction of as-built pipelines from laser-scan data,” *Autom. Constr.*, vol. 35, pp. 199–207, 2013, doi: 10.1016/j.autcon.2013.05.009.
- [82] S. Rahimifard and A. J. Clegg, “Aspects of sustainable design and manufacture,” *Int. J. Prod. Res.*, vol. 45, no. 18–19, pp. 4013–4019, 2007, doi: 10.1080/00207540701608511.
- [83] Y. Liu, T. Bobek, and F. Klocke, “Laser path calculation method on triangulated mesh for repair process on turbine parts,” *CAD Comput. Aided Des.*, vol. 66, pp. 73–81, 2015, doi: 10.1016/j.cad.2015.04.009.
- [84] J. Liu, Q. Chen, Y. Zheng, R. Ahmad, J. Tang, and Y. Ma, “Level set-based heterogeneous object modeling and optimization,” *CAD Comput. Aided Des.*, vol. 110, pp. 50–68, 2019, doi: 10.1016/j.cad.2019.01.002.
- [85] J. Mazumder, A. Schifferer, and J. Choi, “Direct materials deposition: designed macro and microstructure,” *Mater. Res. Soc. Symp. - Proc.*, vol. 542, pp. 51–63, 1999, doi: 10.1557/proc-542-51.
- [86] S. Nowotny, S. Scharek, E. Beyer, and K. H. Richter, “Laser beam build-up welding: Precision in repair, surface cladding, and direct 3D metal deposition,” *J. Therm. Spray Technol.*, vol. 16, no. 3, pp. 344–348, 2007, doi: 10.1007/s11666-007-9028-5.
- [87] R. Ahmad, S. Tichadou, and J. Y. Hascoet, “Generation of safe and intelligent tool-paths for multi-axis machine-tools in a dynamic 2D virtual environment,” *Int. J. Comput. Integr. Manuf.*, vol. 29, no. 9, pp. 982–995, 2016, doi:

10.1080/0951192X.2015.1130258.

- [88] R. Ahmad, S. Tichadou, and J. Y. Hascoet, “A knowledge-based intelligent decision system for production planning,” *Int. J. Adv. Manuf. Technol.*, vol. 89, no. 5–8, pp. 1717–1729, 2017, doi: 10.1007/s00170-016-9214-z.
- [89] J. Liu, Y. Zheng, R. Ahmad, J. Tang, and Y. Ma, “Minimum length scale constraints in multi-scale topology optimisation for additive manufacturing,” *Virtual Phys. Prototyp.*, vol. 14, no. 3, pp. 229–241, 2019, doi: 10.1080/17452759.2019.1584944.
- [90] O. Yilmaz, N. Gindy, and J. Gao, “A repair and overhaul methodology for aeroengine components,” *Robot. Comput. Integr. Manuf.*, vol. 26, no. 2, pp. 190–201, 2010, doi: 10.1016/j.rcim.2009.07.001.
- [91] R. Ahmad and P. Plapper, “Human-Robot Collaboration: Twofold Strategy Algorithm to Avoid Collisions Using ToF Sensor,” *Int. J. Mater. Mech. Manuf.*, vol. 4, no. 2, pp. 144–147, 2015, doi: 10.7763/ijmmm.2016.v4.243.
- [92] P. Tang, D. Huber, B. Akinici, R. Lipman, and A. Lytle, “Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques,” *Autom. Constr.*, vol. 19, no. 7, pp. 829–843, 2010, doi: 10.1016/j.autcon.2010.06.007.
- [93] J. Gao, X. Chen, D. Zheng, O. Yilmaz, and N. Gindy, “Adaptive restoration of complex geometry parts through reverse engineering application,” *Adv. Eng. Softw.*, vol. 37, no. 9, pp. 592–600, 2006, doi: 10.1016/j.advengsoft.2006.01.007.
- [94] Y. Zheng, J. Liu, Z. Liu, T. Wang, and R. Ahmad, “A primitive-based 3D reconstruction method for remanufacturing,” *Int. J. Adv. Manuf. Technol.*, vol. 103, no. 9–12, pp. 3667–3681, 2019, doi: 10.1007/s00170-019-03824-w.
- [95] X. Zhang, W. Li, W. Cui, and F. Liou, “Modeling of worn surface geometry for engine blade repair using Laser-aided Direct Metal Deposition process,” *Manuf. Lett.*, vol. 15, pp. 1–4, 2018, doi: 10.1016/j.mfglet.2017.11.001.

- [96] L. Li, C. Li, Y. Tang, and Y. Du, “An integrated approach of reverse engineering aided remanufacturing process for worn components,” *Robot. Comput. Integr. Manuf.*, vol. 48, no. November 2015, pp. 39–50, 2017, doi: 10.1016/j.rcim.2017.02.004.
- [97] J. Um, M. Rauch, J. Y. Hascoët, and I. Stroud, “STEP-NC compliant process planning of additive manufacturing: remanufacturing,” *Int. J. Adv. Manuf. Technol.*, vol. 88, no. 5–8, pp. 1215–1230, 2017, doi: 10.1007/s00170-016-8791-1.
- [98] J. M. Wilson, C. Piya, Y. C. Shin, F. Zhao, and K. Ramani, “Remanufacturing of turbine blades by laser direct deposition with its energy and environmental impact analysis,” *J. Clean. Prod.*, vol. 80, pp. 170–178, 2014, doi: 10.1016/j.jclepro.2014.05.084.
- [99] C. Mineo, S. G. Pierce, P. I. Nicholson, and I. Cooper, “Introducing a novel mesh following technique for approximation-free robotic tool path trajectories,” *Propuls. Power Res.*, vol. 6, no. 2, pp. 192–202, 2017, doi: 10.1016/j.jcde.2017.01.002.
- [100] A. Masood, R. Siddiqui, M. Pinto, H. Rehman, and M. A. Khan, “Tool path generation, for complex surface machining, using point cloud data,” *Procedia CIRP*, vol. 26, pp. 397–402, 2015, doi: 10.1016/j.procir.2014.07.076.
- [101] F. Pomerleau, F. Colas, and R. Siegwart, “A Review of Point Cloud Registration Algorithms for Mobile Robotics,” *Found. Trends Robot.*, vol. 4, no. 1, pp. 1–104, 2015, doi: 10.1561/23000000035.
- [102] P. Wilkes *et al.*, “Data acquisition considerations for Terrestrial Laser Scanning of forest plots,” *Remote Sens. Environ.*, vol. 196, pp. 140–153, 2017, doi: 10.1016/j.rse.2017.04.030.
- [103] C. K. Chua, C. H. Wong, and W. Y. Yeong, “Software and Data Format,” *Stand. Qual. Control. Meas. Sci. 3D Print. Addit. Manuf.*, pp. 75–94, 2017, doi: 10.1016/b978-0-12-813489-4.00004-0.

