Development and Application of an Automated "Scan to Plan" System for Precision Paint Ablation Using LiDAR and Laser Cameras on Robotic Arms

James Guo

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Electrical Engineering

University of Washington

2023

Committee:

Linda Shapiro

Santosh Devasia

Program Authorized to Offer Degree:

Electrical and Computer Eningeering

University of Washington

Abstract

Development and Application of an Automated "Scan to Plan" System for Precision Paint Ablation Using LiDAR and Laser Cameras on Robotic Arms

James Guo

Chair of the Supervisory Committee:
Linda Shapiro

The thesis presents an innovative approach to automating aircraft maintenance processes, particularly in paint and coating removal. It introduces a "Scan to Plan" system that combines LiDAR scanning and laser cameras with robotic arms for precision paint ablation. The work emphasizes the development of a modular system design, allowing for flexible adaptation to various industrial applications. The prototype demonstrates significant accuracy in scanning and paint ablation on curved aluminum surfaces, highlighting its applicability in laser surface operations beyond paint removal, such as micromachining and cladding. The thesis also explores potential enhancements to the system and its broader applications in industrial automation, paving the way for more efficient and precise maintenance procedures.

TABLE OF CONTENTS

GLOSSARY

LIDAR: Stands for Light Detection and Ranging. A remote sensing method that uses light in the form of a pulsed laser to measure distances to a target.

ROBOTIC ARMS: Mechanized arms used in industrial applications, capable of precise movements and manipulations.

END EFFECTOR: A peripheral device attached to the robot arm's wrist, typically a tool for the specefic application.

PAINT ABLATION: The process of removing paint from surfaces.

MODULAR SYSTEM DESIGN: A design approach where a system is built with interchangeable modules, allowing for flexibility and adaptability in various applications.

SURFACE SEGMENTATION: The process of dividing an image into multiple segments (sets of pixels) to simplify and change the representation of an image into something more meaningful and easier to analyze.

POINT CLOUD: A collection of data in 3D Cartesian coordinates (x,y,z), typically produced by 3D imaging devices.

TESSELLATION: Also known as tiling, the process of covering a surface with patterns of geometry shapes.

WAYPOINT: Points on the surface for one indvidual step of the ablation process for the "Scan to Plan" system.

OPERATION POINT: : The coordinates and orientation the laser end effector moves to for each step of the ablation process for the "Scan to Plan" system.

ACKNOWLEDGMENTS

I am deeply grateful to Macha Vidyaaranya for his pivotal role in the "Scan to Plan" project. His collaboration was essential in shaping this thesis.

I extend my heartfelt thanks to Mike Kaiwi, Megan Taylor, Kaitlyn Lewis, and Peter Correa. Our collective efforts on the robot laser ablation project were not only enriching but also instrumental in my growth. Their unwavering support and encouragement have been invaluable.

Special appreciation goes to Professor Santosh Devasia and Professor Joseph Garbini for their expert guidance and constructive feedback throughout the robot laser ablation project.

I am greatful to Sam Pedigo, whose mentorship and technical insights substantially contributed to this project.

I would like to express my gratitude to Professor Linda Shapiro for her insightful feedback and thorough review of this thesis.

A sincere thank you to Ben Wong, whose ideas and knowledge were crucial in the completion of this thesis.

I also acknowledge Matt Johnson and Kady Gregersen for their guidance in the robot laser ablation project. Their experience and valuable feedback were fundamental in driving the project forward.

DEDICATION

To my sister and father, Jenny Guo and Charles Guo, who always supported me no matter what.

Chapter 1
INTRODUCTION

This thesis addresses the routine practice in aircraft maintenance where the paint and coating are removed approximately every five years for reasons like fading paint, branding changes, and structural inspections. Typically, a commercial aircraft undergoes complete paint and coating removal four to five times during its operational life. [7]

The focus of this work is the "Scan to Plan" framework, a systematic approach designed for automating the interaction between robots and parts. This is achieved by scanning the part, creating a model, and generating a sequence of operations for the robotic arm and end effector to effectively navigate the surface of the part. While its primary application is in laser paint ablation, the framework is adaptable for a variety of laser surface operations, including micromachining and cladding.

A prototype of the "Scan to Plan" system has been constructed and assessed for accuracy in scanning and paint ablation on curved aluminum coupons. The prototype demonstrated its capabilities by effectively ablating paint from dynamically positioned curved aluminum plates within a robotic ablation cell.

Furthermore, the thesis investigates potential enhancements and broader applications of the "Scan to Plan" framework.

Chapter 2
BACKGROUND AND MOTIVATION

2.1 Laser Paint Ablation

Current practices in aircraft maintenance, such as chemical stripping, media blasting, or hand sanding, have significant drawbacks compared to laser ablation. The primary advantages of laser ablation include:

- **Reduced hazardous waste**: Chemical stripping and medial blasting generate considerable hazardous waste, whereas laser ablation significantly reduces this, by over 90%, amounting to a reduction of thousands of pounds of waste per aircraft. [7]

- **Lower ergonomic impact**: Hand sanding methods expose workers to vibration and ergonomic stress, contributing to chronic nerve and musculoskeletal disorders. Laser ablation minimizes these risks. [3]

- **Increased efficiency**: While traditional methods can take up to a week or more to remove paint from an aircraft, laser-ablated processes can be completed in three to four days. [7]

2.2 Scan to Plan

High-powered (Class IV) lasers, used for paint ablation, pose hazards, necessitating stringent control measures for industry compliance. These include light-tight enclosures, as Class IV lasers are harmful to both eyes and skin and have fire risks. [5] Typically, these lasers are operated remotely using computer numerical control (CNC) methods.

Operating with a similar robot work cell [1] , The "Scan to Plan" system thus is designed with these objectives:

- **Modular Process**: Allowing each step to be tailored for different scenarios and creating a smart process for surface analysis and generation of robotic and laser operation plans.

- **Surface Adaptability**: Unlike CNC machines that require individual programming for each task, "Scan to Plan" can autonomously scan and analyze surfaces in real-time, automating the robot work sequence design. This increases efficiency and reliability, especially in maintenance scenarios with variable surface conditions and different parts.

- **Application Flexibility**: The system's modular nature allows for easy adaptation to various industrial applications beyond laser ablation, such as micromachining and metal cutting, by simply altering the planning and machine code output modules. This transforms a single robot cell into a multi-functional workspace, saving costs and factory floor space. [11]

- **User-Friendly Operation**: "Scan to Plan" aims to minimize operator intervention, enhancing ease of use in industrial settings. This translates to reduced training requirements and increased operational efficiency.

[1] See Chapter 3: Hardware and Software Configuration for specifics.

Chapter 3

HARDWARE AND SOFTWARE CONFIGURATION

3.1 Hardware Configuration

The effectiveness of the "Scan to Plan" system is assessed through the process of material removal from bent aluminum plates within a robotic ablation work cell, adhering to established safety standards. The design of this work cell, in line with the principles of the "Scan to Plan" framework, encompasses the following integral components:

- **Part Holding Mechanism**: A specialized table forms this mechanism, crafted to firmly secure and immobilize the targeted part during the procedure. The positioning of the part is such that it appears prominently against the background, creating an effect of suspension.
- **Robotic Arm**: The system utilizes a state-of-the-art industrial robotic arm. The configuration of the arm's joints, its kinematic capabilities, and the spatial arrangement of both the robot and its end effector are meticulously controlled and accurately mapped through the robot arm's specialized software.
- **End Effector Configuration**: The end effector is outfitted with a laser scanning head, trio of distance measuring sensors, and a LiDAR camera. The exact measurements of these components are known, which guarantees a consistent understanding of their spatial positioning relative to the robot's base frame.

All these elements, including the laser on the end effector, the LiDAR camera, and the robotic arm, are harmoniously integrated and managed via their respective software systems on a consolidated computer platform. This setup enables the translation of operational sequences into precise instructions for the machinery.

3.2 Software Configuration

For the "Scan to Plan" system, the following software tools were integral to its implementation:

- **LiDAR Data Capture**: Utilizing the Intel RealSense L515 LiDAR camera, the system leverages its accompanying open-source RealSense SDK and Open3D integration for capturing and processing LiDAR data. [8]

- **3D Data Processing**: Open3D, an open-source library, is employed for the handling and manipulation of 3D data, facilitating the transformation of raw LiDAR data into usable 3D point clouds. [14]

- **Image Processing and Segmentation**: The Segment Anything Model (SAM) API is utilized for advanced image processing and segmentation tasks, essential for accurate surface analysis and preparation for ablation. [9]

- **Surface Analysis and Path Planning**: Employing the scikit-learn machine learning library, the system utilizes its capabilities for surface fitting and polynomial regression, critical for generating efficient and effective robotic trajectories and laser ablation paths. [10]

Each of these software components plays a pivotal role in the "Scan to Plan" system, enabling the capture, analysis, and processing of data required for automated laser ablation processes. The integration of these software tools with the hardware setup forms the backbone of the system, ensuring its operational efficacy and reliability.

Chapter 4
SYSTEM OVERVIEW

The "Scan to Plan" system is engineered to automate the acquisition and analysis of part geometries within a robotic work cell, followed by the formulation and execution of a laser ablation strategy on the part's surface. This chapter provides a synopsis of the system's operation, dividing into phases of scanning and planning, and discusses the design considerations and modular implementation of the system.

Figure 4.1 illustrates the comprehensive framework, encompassing the scanning of the part surface and subsequent planning for the ablation task. The system assumes a precise setup of hardware as detailed in Chapter 3, with a LiDAR camera affixed to the end effector and a robotic arm programmed to navigate the complex contours of aircraft components.

4.1 System Workflow

4.1.1 Scanning

The scanning process involves capturing the topography of the part's surface. The LiDAR camera, mounted on the robotic arm, is maneuvered to scan the entire surface, ensuring that a comprehensive 3D model is generated. This model includes all necessary details to inform the planning process.

4.1.2 Planning

Post-scan, the system transitions to planning. This phase encompasses the development of a robotic trajectory and the configuration of laser parameters tailored to the specific ablation task. The planning software analyzes the 3D model to determine the most efficient path for the robotic arm and the optimal laser settings for effective paint removal.

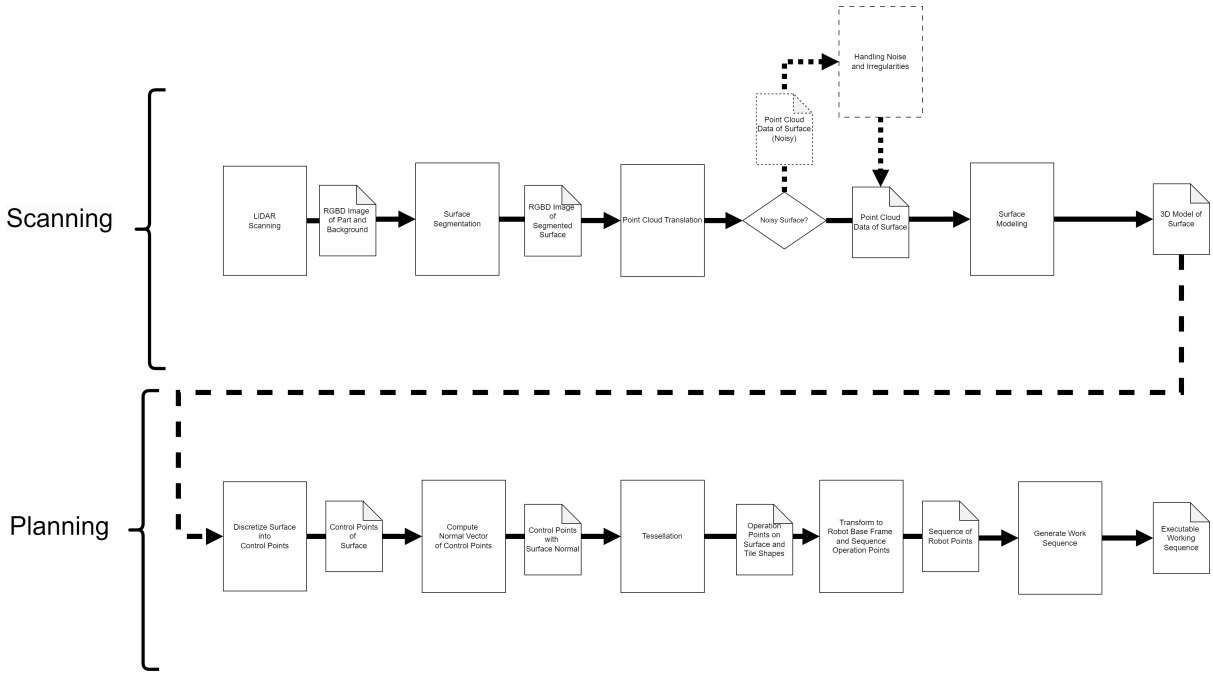


Figure 4.1: "Scan to Plan" System Framework.

This flow chart provides a visual representation of the "Scan to Plan" system's operational flow, delineating the stages of scanning and planning with their corresponding inputs and outputs.

4.2 Discussion of System Design and Implementation

The "Scan to Plan" system, initiated as a prototype for ablating curved painted metal coupons, showcases the potential for a more generalized modular system. This adaptability is a cornerstone of the system's design, allowing for future applications across a wide spectrum of industrial tasks. Each subsystem within the "Scan to Plan" framework is discussed in subsequent chapters, following a three-part structure: "Objectives", "Implementation", and "Limitations".

4.2.1 Objectives

Each subsystem's objectives are outlined, providing an understanding of the high-level functional requirements and expected outcomes. This includes a definition of the inputs the subsystem requires and the outputs it generates, thereby establishing the criteria for modularity and interchangeability within the broader system architecture.

4.2.2 Implementation

In these sections, we describe how the prototype was developed to accomplish the subsystem objectives. We provide insights into the practical aspects of building the system, the methodologies employed, and the software and hardware integrations that were necessary for the proof-of-concept.

4.2.3 Limitations

While the prototype serves as a tangible representation of the "Scan to Plan" vision, it is subject to certain constraints. This portion of each chapter candidly addresses the current limitations of the system's implementation, discussing the hurdles faced during development and areas where the system may not yet meet industrial scalability or versatility.

By dissecting the system into these components, we aim to provide a thorough exposition of the "Scan to Plan" framework's current state and its potential trajectory for future

applications and enhancements, as will be further explored in Chapter 8.

Chapter 5

SCANNING AND DATA PROCESSING

This chapter outlines the scanning phase, where a LiDAR camera is used to capture 3D images of an object and its surrounding work cell. The acquired 3D image data is processed to create a detailed three-dimensional model, representing the spatial characteristics of the object's surface. The methodology includes several distinct steps, which are elaborated on below.

5.1 LiDAR Scanning

5.1.1 Objective

INPUT: - Object Positioned in Work Cell
OUTPUT: - RGBD Image of Object and Work Cell Background
The scanning procedure initiates with the placement of the object in the work cell, followed by the capture of its color, texture, and 3D data, along with the background of the work cell.

5.1.2 Implementation

In this thesis, the Intel L515 LiDAR camera was used to obtain RGBD images as its primary output. Figure 5.1 presents an example of an RGBD image acquired using the LiDAR camera, shown as a two-panel figure: on the left, the RGB image, and on the right, the corresponding depth image. The RGB image captures color and texture details, while the depth image depicts the distance (z value) of each pixel from the camera.

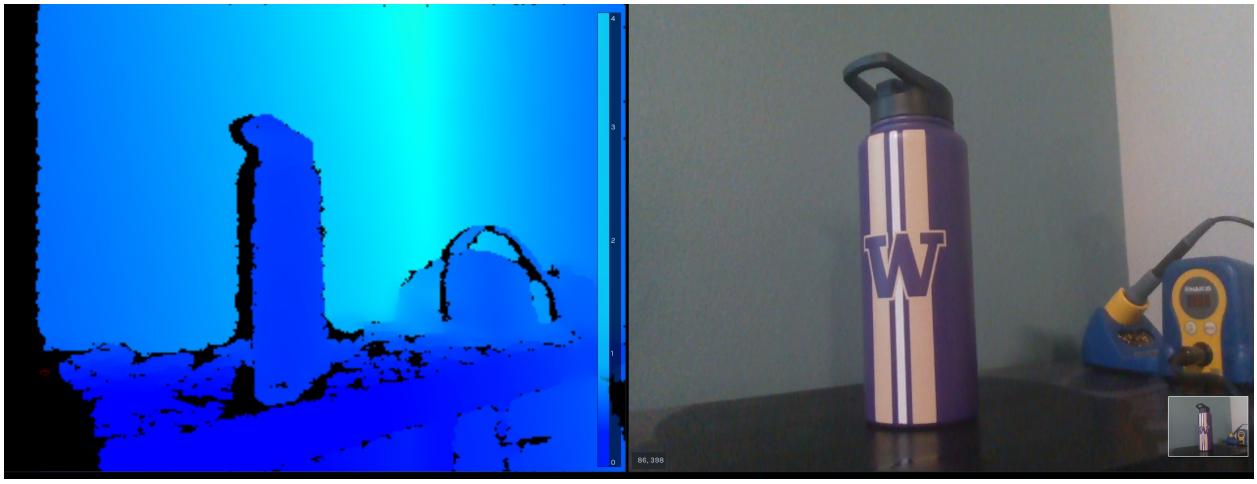


Figure 5.1: Example of an RGBD image.

Left panel: Depth image. Right panel: RGB image. The depth image includes the distance of surface pixels from the camera's perspective.

For general applications, the RGB and depth image can have different resolutions for data optimization, where the color and texture of each depth image pixel can be determined using interpolation and resize techniques. However, image interpolation and resizing inevitably lead to a loss of data, particularly concerning texture and color details. [6] In the "Scan to Plan" system use case, it is crucial to maximize the precision of the scan, thus we matched the resolutions of both images. This resolution alignment guarantees that each pixel within the RGBD image corresponds directly to four stored values: the RGB color values and the depth or distance measurement from the camera. This direct correspondence ensures that no texture or color data is sacrificed for the part surface for the subsequent Surface Segmentation step in Section 5.2 below, resulting in better and sharper segmentation outcomes.

5.1.3 Limitation

Limited singular perspective: The "Scan to Plan" system, in its current form, encounters limitations in scanning and planning for surfaces with complex three-dimensional shapes, such as deeply curved aluminum coupons. The system's field of view, restricted to a sin-

gle angle, is inadequate for capturing the full extent of such surfaces. For comprehensive scanning, techniques like Iterative Point Cloud (ICP) algorithm [2] and advanced panoramic image stitching [4] are necessary to amalgamate data from various perspectives.

5.2 Surface Segmentation

5.2.1 Objective

INPUT: RGBD Image of Object and Background

OUTPUT: Segmented Image-based Pixel Mask of the Object

The subsequent step following the acquisition of the RGBD image involves segmenting the object from the background in the image. This is critical for isolating the surface data necessary for modeling the object's surface and planning the ablation procedure. An ideal segmentation process would produce a pixel mask with precise edges, enhancing the accuracy and efficiency of surface modeling and procedure planning.

5.2.2 Implementation

To segregate the surface data from the background in the LiDAR captured image, we employed two methods, utilizing the RGB and depth images independently. The initial approach applied point cloud clustering techniques to the depth image and its corresponding point cloud. In contrast, the second method utilized machine learning techniques in computer vision to segment the surface using the RGB image. Though differing in data types, both methods share a similar conceptual framework with two core steps: Segmenting points or pixels based on a specified point prompt, which should be located on the surface cluster intended for segmentation. The specific techniques employed in each method are described in the following subsections.

K-D Tree Nearest Neighbor

This technique relies solely on the depth image, generating a point cloud first and segmenting it into clusters. Different from our final approach using the RGBD image, this method directly processes the point cloud from the depth image as outlined in Section 5.3 Point Cloud Translation. Figure 5.2 illustrates the segmentation process. For the algorithm of this method, we utilized the K-D Tree data structure [1] for Nearest Neighbor queries to cluster

surface points based on proximity and the defined radius parameter. Algorithm 1 presents the algorithm's pseudo-code.

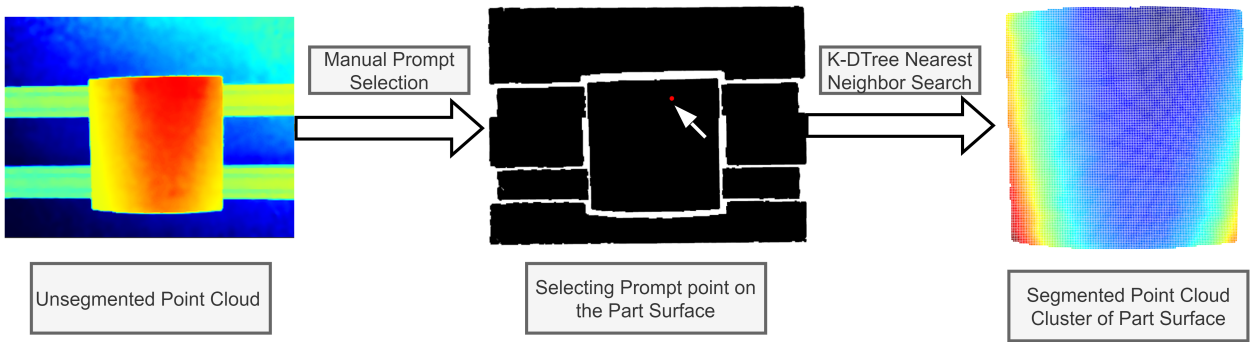


Figure 5.2: Segmentation of Surface via K-D Tree Nearest Neighbor with an Origin Point Prompt.

The initial point chosen by the user serves as the starting point for the nearest-neighbor algorithm.

Algorithm 1 Region Growing Segmentation Algorithm

Require: User-selected point on the point cloud's surface.

Require: Radius r for nearest neighbor consideration.

Ensure: Segmented surface point cloud from the seed point.

```
Initialize an empty KD tree surfacePoints.
Set seedPoint based on user selection.
Add seedPoint to surfacePoints.
Initialize queue pointQueue and enqueue seedPoint.
while pointQueue is not empty do
    currentPoint ← pointQueue.dequeue()
    Find neighbors of currentPoint within radius r.
    for each neighborPoint in neighbors do
        if neighborPoint is not in surfacePoints then
            Add neighborPoint to surfacePoints.
            Enqueue neighborPoint into pointQueue.
        end if
    end for
end while
return surfacePoints as segmented surface point cloud.
```

Segment Anything Model

Our second method takes advantage of the RGB image data, using texture and color information for segmentation. As previously mentioned in Section 5.1 LiDAR Scanning, we ensure one-to-one mapping of RGB and depth image pixels to minimize data loss.

We implemented Meta's Segment Anything Model (SAM) API [9] in our system for segmenting the surface from the background. SAM is a versatile image segmentation model, supporting various input prompts, including foreground/background points, bounding boxes,

pixel masks, and text prompts. Its segmentation capabilities are robust, allowing for the generation of precise masks based on point prompts, automatic segmentation, and creation of multi-layered masks for complex scenarios.

Similar to the K-D Tree method, we utilized a surface point as an input prompt in SAM to identify the segmentation mask. The SAM API was instrumental in determining the most accurate mask representing the part's surface. Figure 5.3 demonstrates this segmentation process. Additionally, it was feasible to implement the prompt as the center of the image/camera as this method uses images as input directly.

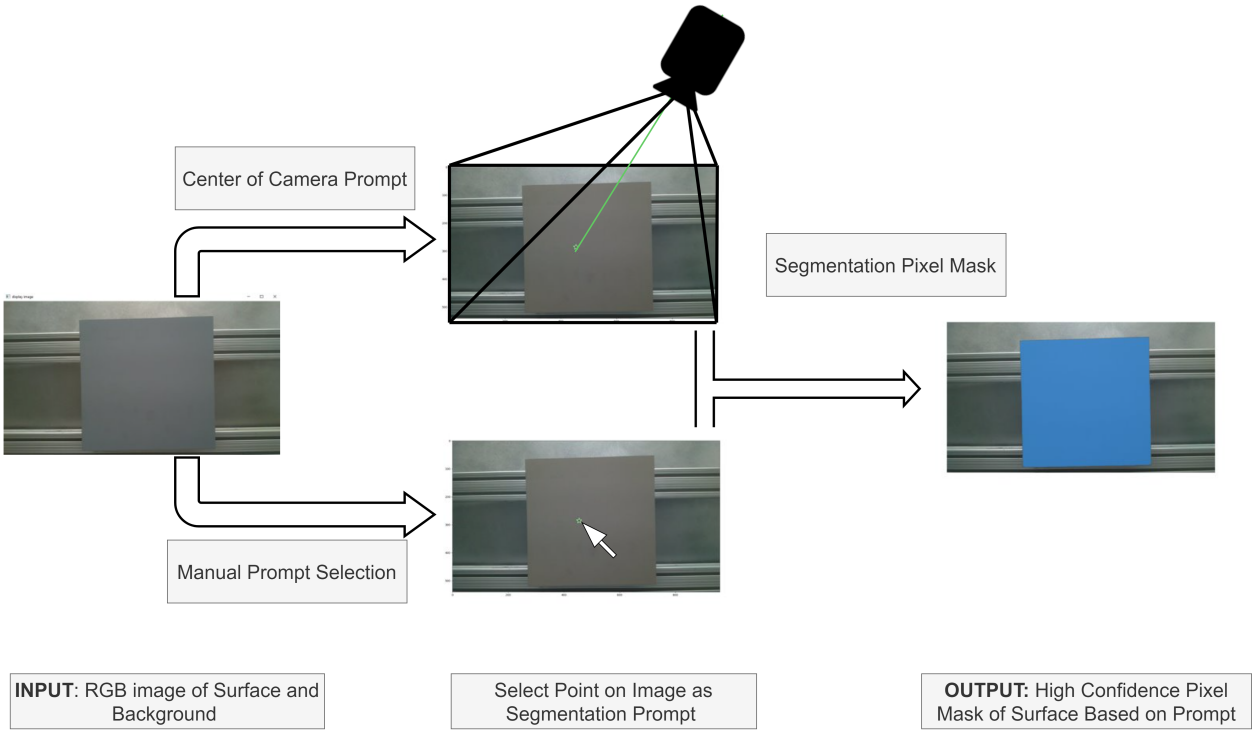


Figure 5.3: Process of Segmenting Part Surface via Point Prompt.

Utilizing Meta's SAM API, segmentation is performed through a point prompt on the part surface, either manually or automatically at the image's center.

Although the current use of SAM parallels the K-D Tree method in its basic functionality, leveraging SAM's advanced features offers significant potential for system design improve-

ment, as discussed in Section 4.2: Discussion of System Design and Implementation. Show in Figure 5.1 here is the extra data SAM gathers for each mask generated, enabling more interesting data processing and applications.Additionally, more usage of the extra data and functionalities can be found in Chapter8: Discussion and Future Work below.

Data	**Description**
segmentation	the mask
area	the area of the mask in pixels
bbox	the boundary box of the mask in XYWH format
predicted_iou	the model's own prediction for the quality of the mask
point_coords	the sampled input point that generated this mask
stability_score	an additional measure of mask quality
crop_box	the crop of the image used to generate this mask in XYWH format

Table 5.1: List of Fields for Generated Masks in the SAM API.

5.2.3 Limitations

For the K-D tree nearest neighbor method, the edge definition remains imprecise, as it produces a direct point cloud from the capture, lacking clear and comprehensive boundary definitions for the surface. The SAM model provides a 2D image bounding box for each segmented mask, which is the smallest box encompassing the mask, offering data for potential future work on surface edges.

The current methods for prompting are limited to fixed points. Exploring advanced prompting methods, such as text-based prompts, could enhance the functionality. Developing prompts that can recognize painted surfaces on industrial vehicles, like airplanes, is an area for future advancement.

5.3 Point Cloud Translation

5.3.1 Objective

INPUT: RGBD Image of Surface

OUTPUT: Point Cloud of Surface

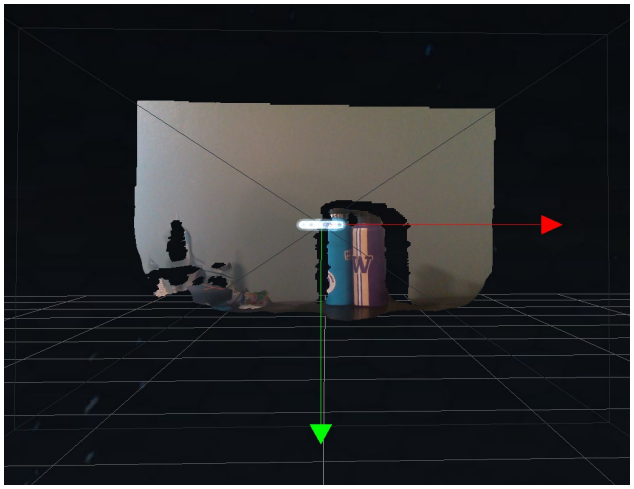
Section 5.1 LiDAR Scanning discusses the depth image in the RGBD image, which represents the distance of each pixel from the LiDAR camera's perspective. To facilitate the creation of a 3D model for planning ablation paths on the part surface, it is crucial to transform the pixels from the depth image into three-dimensional Cartesian coordinates. This transformation must align with the work cell's frame of reference to ensure that the surface model and the work cell operate within the same spatial frame of reference. Such alignment is vital for accurate spatial coordination. The result of this process is a point cloud, a set of discrete (x, y, z) Cartesian coordinates in space.

5.3.2 Implementation

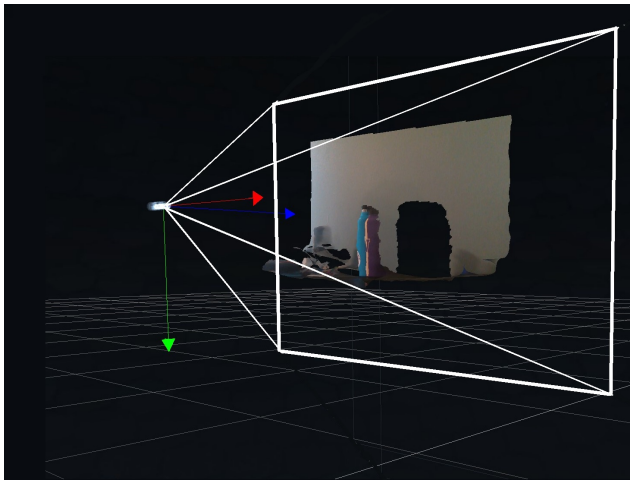
The conversion of depth image data into a point cloud is based on the pinhole camera model principles. This model dictates the projection of 3D points onto a 2D plane. Figure 5.4 illustrates an example of pixel projection using the pinhole method, along with the camera's field of view frustum.

The pinhole camera model conceptualizes a camera with a small, infinitesimal aperture (pinhole) through which light enters and projects onto the camera's internal flat surface. LiDAR cameras function by emitting laser pulses that reflect off surfaces and calculate distance based on the time taken for the light to travel, using the time-of-flight principle. Commonly available LiDAR and depth cameras, similar in aperture size and sensor placement, align with the pinhole camera model's assumptions.

The intrinsic properties of the camera, known as intrinsic camera parameters (e.g., focal length, camera position, perspective), shape the projection geometry. These parameters dictate the transformation and representation of these points in the camera's captured image.



(a) Front View of a Point Cloud Projection.



(b) Side View of a Point Cloud Projection.

Figure 5.4: This figure displays a LiDAR-captured point cloud through the pinhole camera model, depicting the view frustum that defines the camera's field of view in 3D Cartesian space. The view frustum is highlighted on the bottom side view of the point cloud along with the camera frame of view. The presented point cloud exemplifies the pinhole method for spatial mapping.

Accurate intrinsic camera parameters are critical for precise point cloud translation. In this implementation, it is assumed that the camera's intrinsic parameters are known, utilizing standard Intel L515 LiDAR cameras with readily available specifications. [8]

The resultant point cloud is an accurate 3D representation of the surface, assigning spatial coordinates to each point, originating from the camera's central point. Understanding the camera's origin, coupled with the robot arm's kinematics and end effector dimensions, allows for precise surface location data relative to the robot's base frame, after necessary reference frame transformations.

5.3.3 Limitations

The coupon samples in the "Scan to Plan" prototype were tested under ideal conditions, using a diffused, matte painted surface to reduce scan anomalies and noise. In real-world scenarios, factors like lighting, material reflection, and external influences often introduce undesirable anomalies and noise in a single capture, not accurately representing the 3D surface of the part. Section 5.4 Addressing Noise and Anomalies addresses some of these challenges.

5.4 Addressing Noise and Anomalies

5.4.1 LiDAR Scanning Anomalies

5.4.2 Purpose

INPUT: Noisy Initial Surface Scan
OUTPUT: Processed Data Suitable for Analysis

In some instances, a single LiDAR scan may not be sufficient. Challenges typically emerge when using a LiDAR sensor on highly reflective surfaces, which often generate a "specular highlight" phenomenon at the most illuminated parts of the object, resulting in false impressions of depressions and protrusions. As depicted in Figure 5.5, the contrast in scan quality between a metal sheet's reflective side and its diffused painted counterpart

is stark. Additionally, surfaces with a semi-gloss finish may exhibit a ripple-like distortion during scanning, as illustrated in Figure 5.6. To accurately gauge the object's surface, these inconsistencies must be addressed.

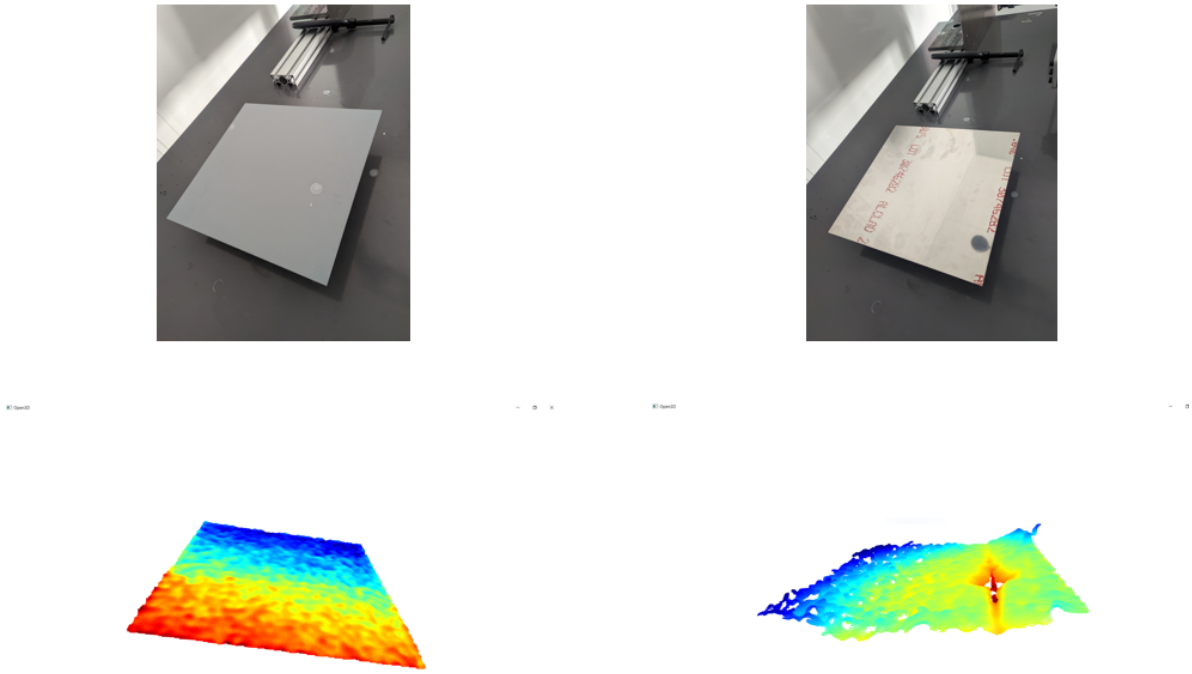


Figure 5.5: Comparison of a shiny and a diffusely painted metal sheet surface with corresponding LiDAR scan point clouds below.

Left: diffuse painted surface. Right: shiny metallic surface.

In the development of our "Scan to Plan" system for the robotic work cell, we initially planned to use only non-reflective "matte" surfaces, making noise elimination an optional feature. Nonetheless, during the system's design, we investigated and integrated techniques to mitigate such noises and anomalies for potential use in specific cases, as detailed in the following sections.

5.4.3 Implementation

Noise Elimination

Reflective Surfaces Figure 5.5 demonstrates how scanning a reflective metallic surface yields a significantly distorted point cloud, rendering most of the data unusable. In such cases, data processing methods are not effective. The simplest solution is to apply an anti-reflective LiDAR coating, readily available in the market, to render the surface more suitable to LiDAR scanning.

Semi-Gloss Surfaces Our experiments on semi-gloss surfaces revealed that the observed ripples typically appear as abrupt, high-frequency distortions in surface height. As shown in Figure 5.6, employing the Open3D *statistical_outlier_removal* function [14] proved effective in eliminating these ripples. This function filters out anomalies by assessing the local spatial context within the point cloud and discarding points that significantly deviate from their immediate surroundings.

While this process of distortion elimination is useful, it often leads to a point cloud with gaps, which compromises the completeness of the data representation, as shown in the aforementioned figure. Addressing this issue involves integrating different surface perspectives, given that noise exhibits varying patterns at different camera angles, as depicted in Figure 5.7. This concept is elaborated further in 5.4.3 below.

Synthesizing Comprehensive Views via Iterative Closest Point Method

The Iterative Closest Point (ICP) method is favored for consolidating point clouds or images from different viewpoints into a single, unified point cloud. The method merges two point clouds of the same object from different perspectives with an estimation of relative positions as input. [2] In situations where a single snapshot is insufficient, as outlined in 5.4.3, ICP can merge multiple scans from varied angles to create a detailed point cloud accurately representing the surface. Cleaning the scans of distortions and merging the cleansed point

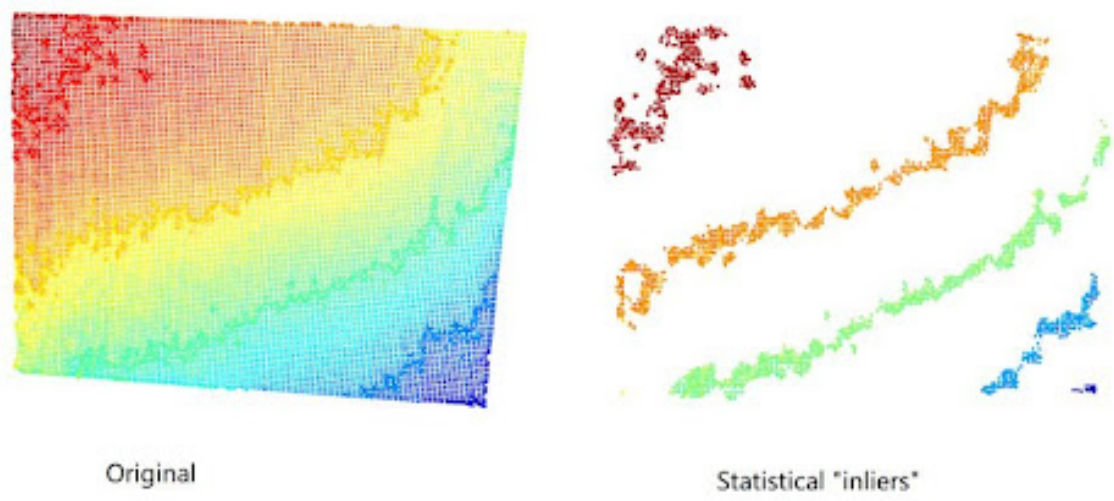


Figure 5.6: Use of statistical outlier removal for a semi-glossy surface noise reduction. Left: original surface. Right: isolated noise points.

clouds from diverse perspectives using ICP can effectively fill gaps caused by noise, as these gaps change with the scanning angle, as shown in Figure 5.7. Our prototype utilized the ICP feature from the OPEN3D library. [14] The robot arm software's precise control also aids in accurately estimating the positional differences for subsequent captures required by the ICP algorithm. Figure 5.8 showcases an example of noise elimination using multiple angles of a subject.

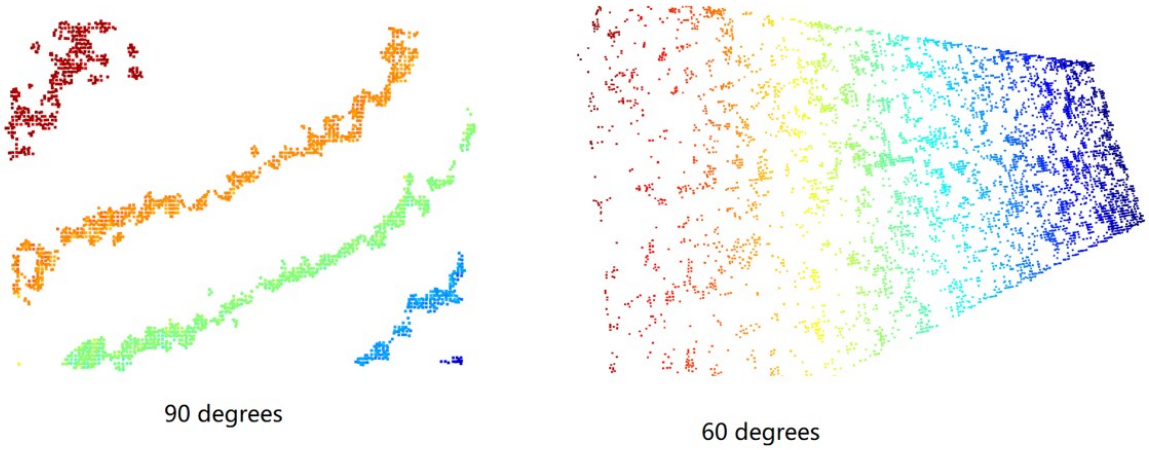


Figure 5.7: Different LiDAR camera angle of the same surface has different noise patterns. Left: original surface. Right: isolated noise points.

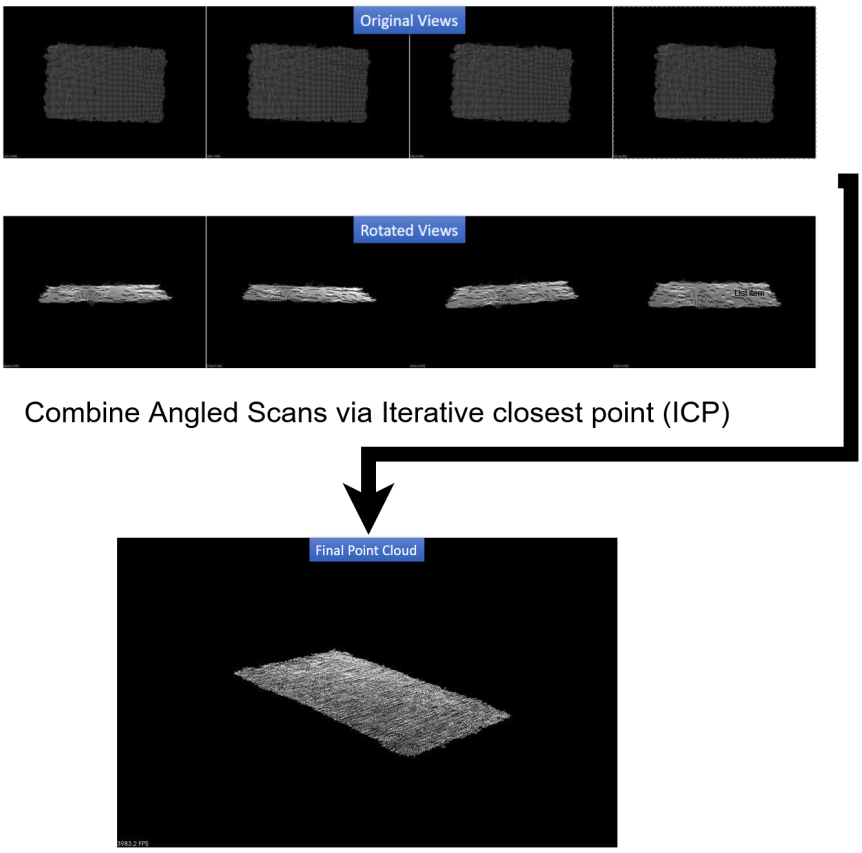


Figure 5.8: Utilizing Iterative Closest Point for merging different angle scans of a surface into an accurate composite.

5.4.4 Limitations

The noise handling techniques implemented were mainly experimental and preliminary, as they were developed early in the project without the assumption of using only diffused matte surfaces in the prototype. Therefore, further testing and development are necessary to establish a robust and efficient noise management system, which includes systematic noise evaluation.

5.5 Surface Representation and Modeling

5.5.1 Objective

INPUT: Point Cloud of Part Surface

OUTPUT: 3D representation of Surface

The goal in surface modeling is to convert a point cloud, which depicts an object's surface, into a more comprehensive model for planning applications. This transformation involves creating a refined 3D model from the point cloud data, facilitating effective sampling for planning.

5.5.2 Implementation

The "Scan to Plan" initiative utilized polynomial regression, specifically fitting a 3rd degree polynomial to the point cloud data. This polynomial comprises a single linear layer with nine inputs and one output, representing the features and the resultant z-value, respectively. The features consist of the x and y coordinates, expressed in powers and combinations up to the third degree, as detailed in Table 5.2. The fitting process seeks to identify the best coefficients, w_1 to w_9, and a bias term b, to match the points in the point cloud. The corresponding equation can be expressed as:

$$z = w_1 \cdot x^3 + w_2 \cdot y^3 + w_3 \cdot x^2y + w_4 \cdot y^2x + w_5 \cdot x^2 + w_6 \cdot y^2 + w_7 \cdot xy + w_8 \cdot x + w_9 \cdot,$$

Features	Coefficients
x^3	w_1
y^3	w_2
x^2y	w_3
y^2x	w_4
x^2	w_5
y^2	w_6
xy	w_7
x	w_8
y	w_9
1	b

Table 5.2: Table of features for the 3rd degree polynomial used, with the corresponding coefficients.

We implemented polynomial regression using scikit-learn [10] to formulate a basic regression model trained as a 9 feature machine learning model.

This model results in a part surface representation that is independent of the point cloud and can be sampled using any x and y coordinates. These coordinates are instrumental in locating control points on the surface for planning, further discussed in Chapter 6: Planning the Ablation Procedure.

5.6 Limitations

The model struggles with defining the edges of surfaces, as it relies solely on the initial point cloud derived from the K-D Tree output. Determining whether a point or vector projection is on the surface requires direct reference to the point cloud and identifying the closest point.

The model is limited to handling a single continuous curve. Surfaces with discontinuities

would benefit from piece-wise definitions. During experiments with SAM, segmenting surfaces into smaller curves showed promise. Incorporating computer vision techniques might be beneficial, particularly for identifying discontinuities marked by changes in lighting or texture.

The polynomial model's susceptibility to overfitting and underfitting, known as the "Runge Phenomenon" [12], limits its ability to represent complex surfaces. Adding higher degree coefficients does not necessarily enhance its modeling capacity.

The model's limits have not been fully explored. Its simplicity, akin to linear regression fitting, has only been tested on sample materials bent in a single direction, not accounting for more complex shapes or multi-directional bends.

Chapter 6

PLANNING THE ABLATION PROCEDURE

6.1 Comprehending the Laser Ablation End Effector and Operational Plane

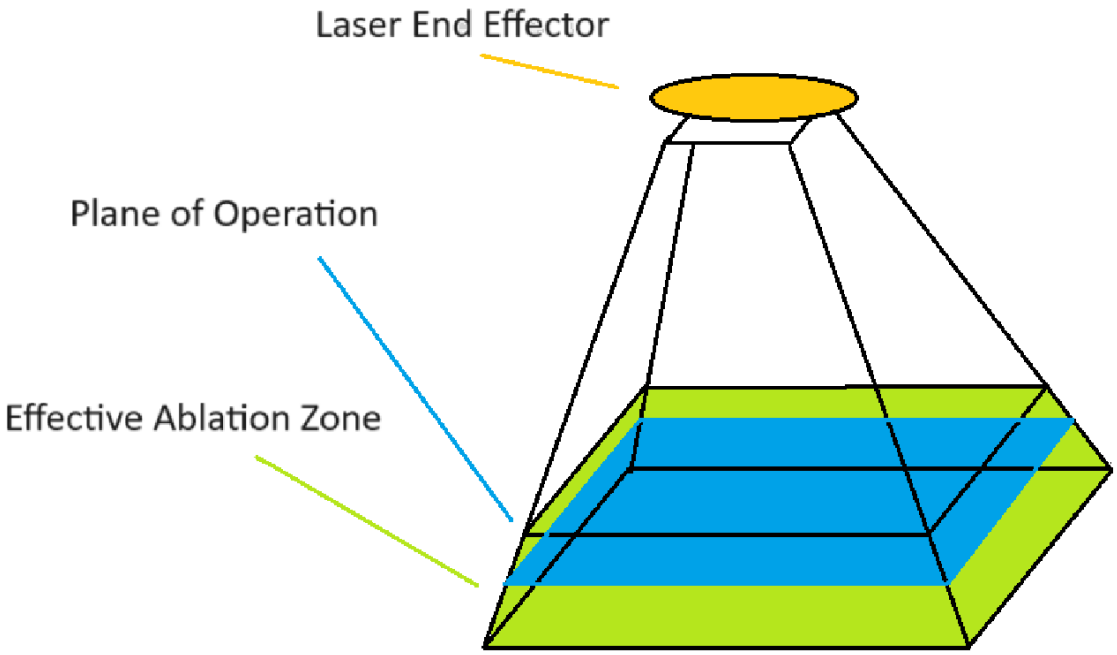


Figure 6.1: Characteristics of the laser end-effector.

The frustum shape emanating from the laser end effector represents the laser's field of view. The effective ablation area is indicated in green, with the blue plane denoting the operational plane.

To effectively design a laser ablation process, understanding the laser ablation end effector's functionality is essential. This component typically utilizes a programmable pulse

laser and has an "effective ablation zone" in a frustum shape, as shown by the green area in Figure 6.1. The laser operates within its focal length in this area, maximizing effectiveness. However, the laser is programmed as raster image patterns, approximating the "effective ablation zone" as an "operational plane", illustrated as the blue area in Figure 6.1.

6.2 Implementing a Tessellation Approach

Utilizing the laser end effector's capability to adhere to predefined paths and the robot arm's precise control over its position, a typical technique for surface ablation is a "tile"approach. This strategy involves segmenting the entire surface into smaller, controllable shapes or "tiles", which are then methodically ablated by the laser. The tessellation process, which entails dividing the surface into these tiles, facilitates sequential treatment of each tile.

6.2.1 Present Implementation of Tessellation Pattern: Square Tiles

In our preliminary demonstration of the "Scan to Plan" system, square tiles were utilized, as a grid pattern is straightforward to define and compute, particularly in a Euclidean 3D space with Cartesian coordinates, the environment in which our robot, laser, and LiDAR camera operate. Figure 6.2 illustrates a sequence of square tessellations performed on a single-curved metal plate, akin to our application.

6.3 Waypoint Generation on Surfaces

6.3.1 Objective

INPUT: 3D Surface Model
OUTPUT: Surface Operation Points

The primary step in the ablation process planning is the identification of waypoints. These waypoints represent the central points of each surface tile. In a square tile configuration, these points create a grid pattern, uniformly spaced across the surface. The grid dimensions are slightly smaller than the tile size, ensuring a slight tile overlap for complete surface

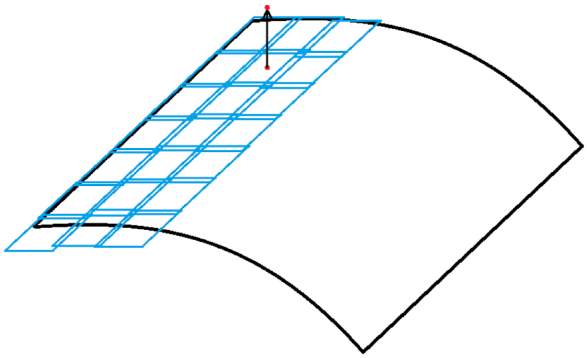


Figure 6.2: Sequence of Square Tessellation.

The laser pauses at the normal vector center of each tile, maintaining the focal length distance, ablates a square of the surface, and waits for the robot to transition to the next square.

coverage.

6.3.2 Implementation

The approach to determine waypoints employs the data from the K-D Tree segmentation method[1] and polynomial regression surface modeling[2], using the point cloud, a surface prompt point, and the polynomial description of the surface. This method omits additional data from the SAM segmentation, such as the surface's x-y bounding box, discussed further in detail in the limitations section below.

The waypoint identification is executed through a recursive exploratory algorithm. This algorithm initiates from a surface prompt point, expanding to adjacent points until it exceeds the defined surface boundary. Figure 6.3 depicts the generation of waypoints along one x and y axis, with the process similarly extending in the opposite directions. Algorithm 2 provides the recursive pseudocode for this methodology.

---

[1]Refer to 5.2.2.

[2]Refer to 5.5.

Algorithm 2 Surface Waypoint Generation Algorithm

Require: 3D Surface Model as a point cloud.

Require: Surface prompt point.

Require: Polynomial surface formula.

Ensure: Set of surface operation points (waypoints).

```
Establish a grid size g, smaller than tile size for overlap.
Initialize an empty list waypoints.
Set seedPoint as initial operation point.
Insert seedPoint into waypoints.
function ExpandWaypoints(currentPoint, direction)
    while currentPoint is within the point cloud limits do
        Compute the next point in direction using g and polynomial formula.
        if next point is valid and on the surface then
            Append next point to waypoints.
            ExpandWaypoints(next point, direction)
        end if
    end while
end function
ExpandWaypoints(seedPoint, (1, 0))          ▷ Expanding in X direction
ExpandWaypoints(seedPoint, (−1, 0))         ▷ Expanding in -X direction
ExpandWaypoints(seedPoint, (0, 1))          ▷ Expanding in Y direction
ExpandWaypoints(seedPoint, (0, −1))         ▷ Expanding in -Y direction
return waypoints as operation points.
```

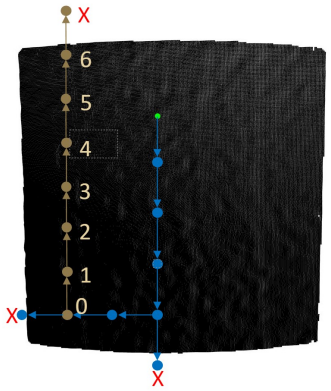


Figure 6.3: Waypoint generation along one direction, beginning at the green seed point and expanding to the boundary.

6.3.3 Limitations

Inefficient Tile Utilization: Not all generated points are optimally positioned. Some may be on surface edges, causing tiles to extend beyond the surface. This results in uneven overlap, as shown in Figure 6.2. Incorporating the surface's bounding box/plane from SAM analysis of the LiDAR RGB image can improve point determination and direction of exploration.

Challenges with Rotated Surfaces: For non-axis-aligned surfaces, tile placement may be suboptimal, especially along diagonal edges. A bounding box from the SAM segmentation implementation [3] can provide clarity on surface limits to allow smarter tile placement.

6.4 Calculation of Normal Vectors and Operational Point Identification

6.4.1 Objective

INPUT: Operational waypoints, 3D Surface Model

OUTPUT: Laser End Effector Operational Points

Planning the ablation task involves determining the laser end effector's position and

[3]Refer to 5.2.2.

orientation at each waypoint. This precision is crucial for effective tile ablation. These precise locations and orientations are referred to as "operational points".

6.4.2 Implementation

The first step is calculating the surface normal vector at each waypoint, as detailed in Section 5.5. By using the surface's definition and the waypoint's position, the normal vector at a point is obtained via the cross product of surface gradients in x and y directions at that point. After establishing the normal vector, the operational point is positioned a focal length away from the waypoint along this vector. The laser orientation will be the inverse of the normal vector. The robotic work cell's control over the laser end effector's orientation and position enables precise navigation to these operational points for optimal ablation.

6.5 Reference Frame Transformation and Instruction Sequencing

6.5.1 Objective

INPUT: Set of Operational Points
OUTPUT: Ordered Sequence of Operational Points for Procedure Execution

This phase aims to sequence the operational points to formulate final instructions for the robot work cell.

6.5.2 Implementation

The ablation process begins from the bottom of the part, moving upwards. Starting from the top could result in ablated dust settling on lower areas, reducing ablation effectiveness. The process initiates from the bottom-left region.

6.6 Base Coordinate Frame Transformation and Work Cell Instruction Transmission

6.6.1 Objective

INPUT: Ordered Sequence of Operational Points for Procedure **OUTPUT**: Robot Arm and Laser Ablation Commands

The final step involves converting operation points into instructions for the robot arm and laser. The tile-based ablation process does not require movement of the end effector during tile ablation, simplifying the sequence to an open-loop process: the robot moves to each operation point, the laser ablates, then moves to the next point.

6.6.2 Implementation

First, operation points are converted to the robot base frame reference, as discussed in Chapter 3. Knowledge of the robot arm and LiDAR mounted end effector's positional data facilitates this transformation.

Afterwards, a script is used to translate the sequence of operation points to the robot arm PLC language. Since our tile pattern is a square grid with identical patterns, the laser only needs to read one tile pattern beforehand. A running Python script on the robot work cell operation computer is used to time and send the open-loop procedure commands sequentially during execution of the ablation process to the robot arm and laser.

Chapter 7

TESTING AND RESULTS

The "Scan to Plan" system underwent evaluation through a proof-of-concept prototype, integrated with robot work cell hardware. Testing primarily focused on basic curved aluminum coupons.

The procedure involved scanning the coupons, generating a 3D model from the data, and formulating an ablation trajectory. Utilizing a laser ablation end effector, the robotic arm accurately traced these paths, efficiently stripping paint from the coupons without harm to the base material. This demonstrates the system's proficiency with elementary curved surfaces. Figure 7.1 displays the ablation operation in the robot work cell, while Figure 7.2 presents a completely ablated coupon surface.

In industrial contexts, the standard objective for paint removal thickness per sweep is approximately 1mm. Our tests showed that each laser pass removed the approximate desired thickness of paint. As illustrated in Figure 7.3, consecutive passes resulted in the elimination of additional paint layers. Both Figure 7.2 and Figure 7.3 exhibit the outcomes of ablating singular and multiple layers.

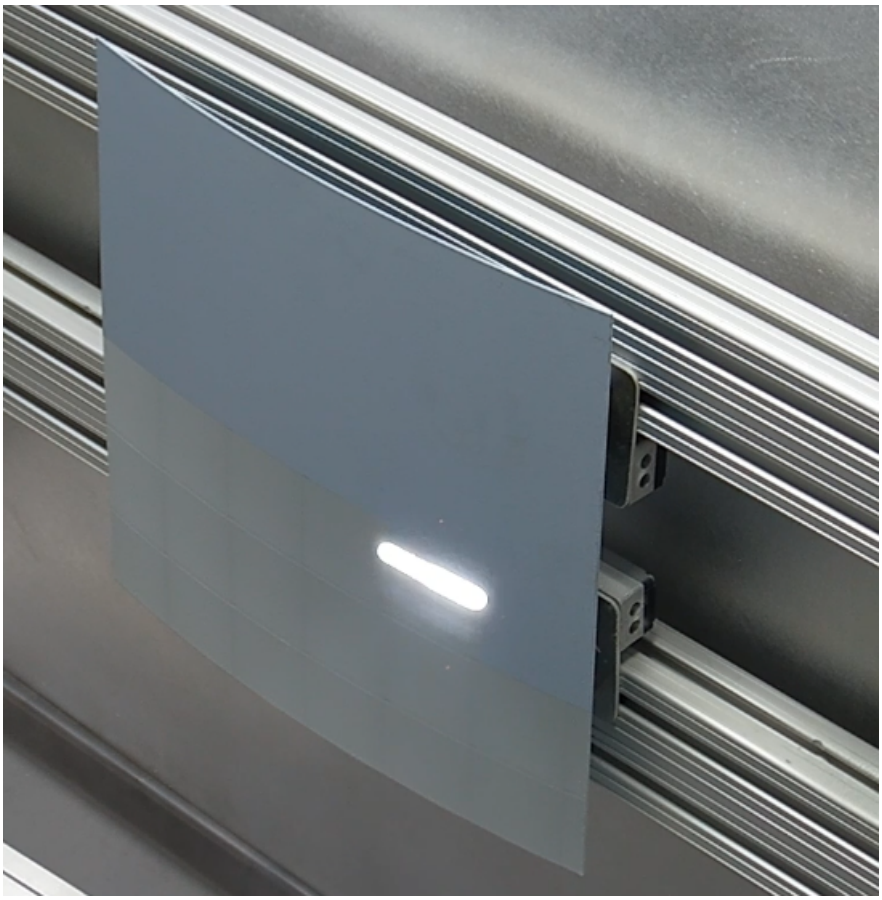


Figure 7.1: Image of the robotic paint ablation process.

The ongoing ablation sequence in our work cell, featuring the laser's interaction with the paint, visible as a "white flash". The contrast between the square tiles and varying layers of paint in ablated and non-ablated sections is noticeable.

Figure 7.2: Depiction of a Square Tessellation pattern.

The laser halts at the normal vector center of each tile at a set focal distance, removes a square of the surface, and waits for the robot to transition to the next square.

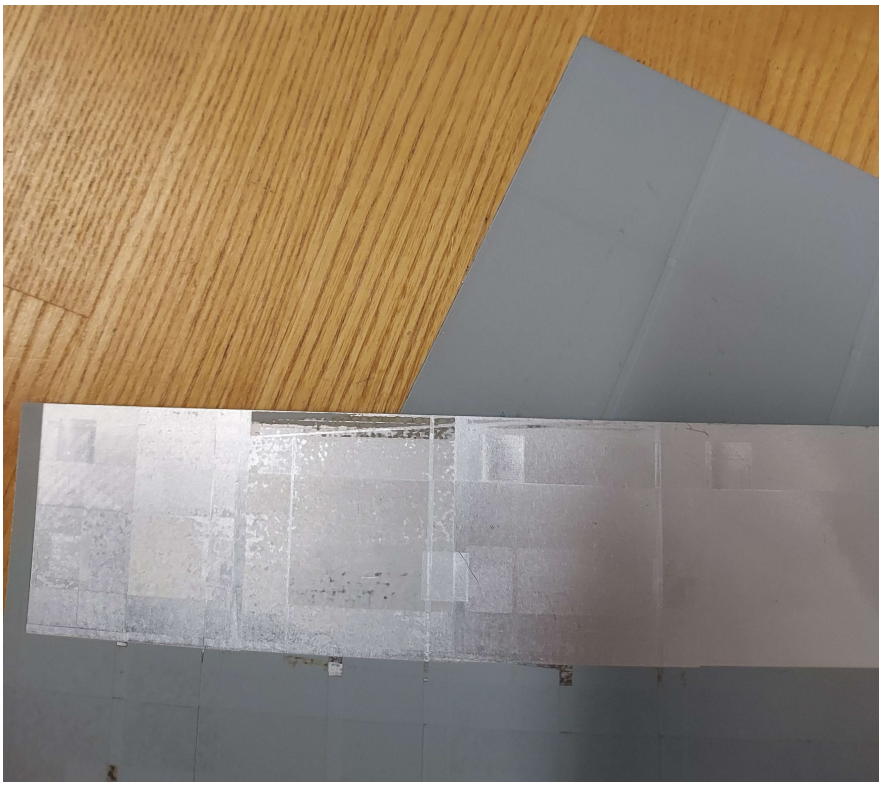


Figure 7.3: Contrast of paint layers on two coupons: single versus multiple passes.

The bottom coupon demonstrates multiple paint layers removed after numerous passes, in contrast to the top coupon which shows a single layer removed, exemplifying the control over ablation depth and layers.

Chapter 8
DISCUSSION AND FUTURE WORK

As elaborated in Chapter 2: Background and Motivation, the "Scan to Plan" system, initially concentrating on basic rectangular curved surfaces and designed for varied applications, presents a platform for further exploration and enhancement. This chapter discusses potential improvements and intriguing research questions arising from our development and study of the system.

8.1 Scanning and Modeling the Curves

8.1.1 Current Limitations

The effectiveness of our system in handling simple curved surfaces, such as rectangular coupons, has been established. However, the methodology has not been extended to more intricate surfaces. The polynomial fitting used in modeling these surfaces is prone to overfitting or underfitting issues, as illustrated by Runge's phenomenon.

8.1.2 Future Work

To enhance curve handling and the modeling of complex surfaces, we propose two key improvements to the "Scan to Plan" system:

Piecewise Segmentation of the Part

Segmenting complex industrial plane parts into smaller, manageable piecewise curves could prove beneficial. As we discussed in Section 5.1: LiDAR Scanning, using RGBD images allows for segmentation of point cloud data based on RGB imagery. Figures below demonstrate potential applications in an industrial context, using our SAM implementation:

Segmenting Discontinuity Figure 8.1 illustrates the decomposition of a surface into three distinct, well-defined segments.

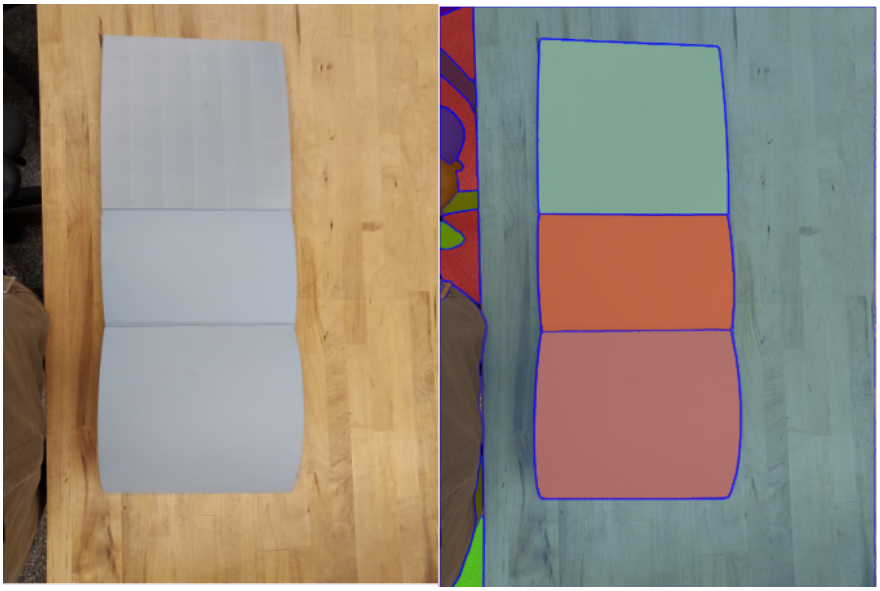


Figure 8.1: Example of segmenting three overlapping curved surfaces into piecewise sections. This approach simplifies the modeling by dividing the surface into three manageable parts.

Excluding Holes from Surfaces Industrial vehicles and equipment often feature surfaces with rivets or screws, presenting small holes that are challenging to differentiate from LiDAR scan noise. By applying computer vision techniques alongside RGB imaging, we can more accurately identify and measure these holes. Figure 8.2 demonstrates using SAM to detect holes on a tape measure as a preliminary example.

Identifying Paint Defects and Corrosion Detecting areas with chipped paint or corrosion is crucial for maintenance. Computer vision and RGB images can effectively segment these defective areas. Figure 8.3 shows a prototype of segmenting chipped paint and corrosion on one of our painted coupons.

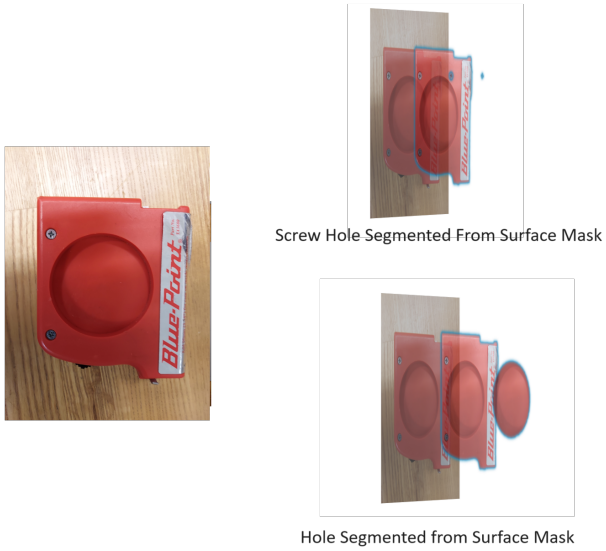


Figure 8.2: Example of SAM identifying and excluding holes from the surface of a tape measure.

Employing computer vision, we can discern and omit features like screw holes and center holes in the segmentation process.

Modeling the Surface

Despite the piecewise segmentation, challenges like Runge's phenomenon in simple polynomial fitting persist. Hence, exploring more advanced surface modeling techniques is essential. The field of computer graphics and computer-aided design offers various methods for modeling and rendering complex 3D objects, including bezier surfaces, presenting a research opportunity for adaptable scanning methods.

Wider Field of View

Our current model utilizes data from a singular perspective, focusing on "relatively flat" parts. The ICP method, as discussed in Section 5.4: Addressing Noise and Anomalies, could

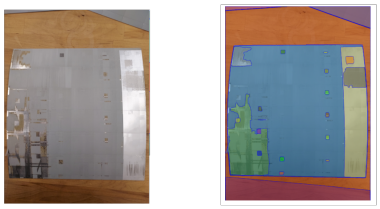


Figure 8.3: Example of SAM segmenting areas with damaged paint and corrosion.

The technique allows for the segmentation of irregular surface areas, such as those with corrosion or damaged paint, for differentiated handling.

be adapted to integrate multiple perspectives for 3D object modeling from various angle-derived point clouds. Implementing a multi-angled capturing method using the robot work cell and the LiDAR's positioning capabilities could provide a more comprehensive view of complex objects, enhancing the model's 3D capabilities.

8.2 Planning the Procedure

8.2.1 Limitations of Current Work

The simple square tile pattern employed in our system, as detailed in Section 6.2: Implementing a Tessellation Approach and Chapter 6: Planning the Ablation Procedure, presents certain limitations in addressing complex surfaces.

8.2.2 Future Work

Enhanced Surface Model and Tessellation System

Adopting a more sophisticated tessellation system could significantly expand the system's adaptability. Various methods exist in computer graphics, mathematics, and computer-aided design for tessellating 3D objects accurately.

Bézier Surface Tessellation An application worth exploring is the tessellation of Bézier surfaces in computer graphics. These surfaces are defined through a control mesh of points, creating a smooth, continuous approximation. Similarly, the "Scan to Plan" system could generate control points from downsampled point clouds, potentially defining Bézier surfaces. Computer graphics often render 3D objects using triangles, necessitating the approximation of smooth curves with linear elements. The system could thus develop advanced tessellation techniques using mathematical principles similar to those employed in computer graphics.

Refined Surface Boundaries

Our current model does not explicitly define surface boundaries, leading to inefficiencies in tile creation and potential challenges. As indicated in 5.2.2, computer vision segmentation provides bounding boxes for each segment, aiding in the precise handling of surface edges. Using the pinhole projection method (referenced in 5.3), we can establish a 3D boundary for the surface, enhancing procedural planning.

Chapter 9

CONCLUSION

In this project, we endeavored to create a "Scan to Plan" framework, aimed at addressing practical needs in robotic surface operations. This framework represents an initial step towards integrating scanning and planning in robotic systems, specifically tailored for tasks involving surface interactions.

We successfully developed a preliminary working prototype of this system and conducted its initial testing within a robot laser ablation work cell environment. While this prototype serves as a foundational model, it is important to note that it is an early-stage implementation, primarily demonstrating the basic feasibility and application of the "Scan to Plan" concept in a controlled setting.

The completion of this project has highlighted several potential areas for further research and development. These areas include exploring ways to enhance the system's capabilities and adaptability to various use cases. The insights gained from this project suggest that there is significant room for improvement and innovation in the "Scan to Plan" system, which could be explored in future work to make robotic surface operations more efficient and versatile

BIBLIOGRAPHY

[1] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, sep 1975.

[2] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[3] Massimo Bovenzi, Anna Della Vedova, Pietro Nataletti, Barbara Alessandrini, and Tullio Poian. Work-related disorders of the upper limb in female workers using orbital sanders. *International Archives of Occupational and Environmental Health*, 78(4):303–310, 2005.

[4] Matthew Brown and David G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2006.

[5] Occupational Safety DC: U.S. Dept. of Labor and Health Administration. *OSHA Technical Manual. Washington (1999)*. For sale by the Superintendent of Documents, U.S. Government Printing Office, 1999.

[6] Rafael C. González and Richard E. Woods. *Digital Image Processing, Fourth edition*. Pearson Education Limited, 2018.

[7] Kady Gregersen. A light solution to a gritty issue - laser paint removal for airplanes is safe, fast and environmentally friendly. *Boeing Innovation Quarterly*, 2021.

[8] IntelRealSense. Intelrealsense/librealsense: Intel® realsense sdk, 2016.

[9] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] IPG Photonics. *Industrial Fiber Lasers for Materials Processing*. IPG Photonics, 2019.

[12] Carl Runge. Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten. *Zeitschrift für Mathematik und Physik*, 46:224–243, 1856.

[13] David Wolff. *OpenGL 4.0 Shading Language Cookbook*. Packt Publishing, 2011.

[14] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.