Disturbance Reduction in Automated Inspection Systems for Enhanced Robust Image Data Collection

SangYoon Back

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

Master of Science

University of Washington

2023

Committee:
Xu Chen
Santosh Devasia
Kritika Manohar

Program Authorized to Offer Degree:
Department of Mechanical Engineering

University of Washington

Abstract

Disturbance Reduction in Automated Inspection Systems for Enhanced Robust Image Data Collection

SangYoon Back

Chair of the Supervisory Committee:
Xu Chen
Department of Mechanical Engineering

This thesis presents a comprehensive study aimed at reducing disturbances in automated inspection systems to enhance the robustness and accuracy of image data collection. The research focuses on identifying common sources of interference and errors in automated inspection systems, particularly in industrial and manufacturing settings. Through a series of experiments and simulations, the study evaluates various techniques for mitigating these disturbances, including advanced algorithms for image processing, noise reduction, and calibration methods.

The core of the thesis lies in developing a novel framework that integrates enhanced sensor technologies, improved image processing algorithms, and machine learning techniques to increase the reliability and efficiency of automated inspection systems. The proposed solution is designed to be adaptable to different types of environments and machinery, ensuring broad applicability.

Extensive testing demonstrates that the implemented strategies significantly minimize errors caused by external factors such as lighting variations, vibrations, and other environmental influences. The results indicate a notable improvement in the precision and consistency of image data captured by automated inspection systems.

This research not only contributes to the field of automated inspection but also has im-

plications for broader applications where accurate image data collection is critical. The findings offer valuable insights for the development of more robust and efficient automated systems in various industrial applications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to College of Mechanical Engineering, Mechatronics, Automation, and Control Systems Lab and General Electric Global Research for their support which made this project possible. I would also like to express my heartfelt gratitude to Dr. Xu Chen and Dr. Krithika Manohar for their mentorship and guidance, my committee members Dr. Santosh Devasia for their guidance, Colin Acton and Arun Nandagopal for their help and support throughout this project, all the members of the MACS lab for their valuable inputs.

DEDICATION

to my parents.

Sung-Hwan Bernhard & Barney Bernhard.

Chapter 1

INTRODUCTION

1.1 Background and Motivation

The primary objective of this study is to develop an end-of-arm tool and an image acquisition and classification system, specifically designed for high-precision inspection applications in the aerospace industry. This research aims to mitigate variance in data caused by unpredictable environmental factors, which is often a common shortfall of traditional manual inspection methods.

Traditionally, the aerospace industry has relied on human visual judgment for inspections. This method, grounded in experience, encounters significant limitations, particularly in its ability to detect fine defects and maintain consistent accuracy. Despite the reliance on skilled inspectors, studies indicate that human accuracy in such tasks is approximately 75 % [4]. This is particularly concerning when contrasted with the aerospace Guidelines, which set allowable defect sizes in the range of 0.001” to 0.0005”, sizes that are challenging for human inspectors to consistently detect [1].

Manual inspections, by their very nature, are not only physically and mentally demanding but also prone to human error, a critical factor that can result in sub-optimal throughput of high-demand aerospace parts. The physical and cognitive strain placed on inspectors during long and meticulous examination processes can exacerbate these errors, further diminishing the efficacy of the inspection process.

These challenges, coupled with the inherent subjectivity of human judgment, underscore the pressing need for more advanced, objective, and consistent methods of inspection. The reliance on human inspectors, while rooted in traditional practices, highlights a significant gap in the industry’s ability to meet growing demands for precision, efficiency, reliability, and traceability. As the aerospace industry continues to evolve and the complexity of components increases, the limitations of the current inspection methodology become even more pronounced, paving the way for the integration of more advanced and technologically advanced inspection approaches.

Figure 1.1: Traditional Aerospace Industry's Inspection System [13]

GE's development of an AI-based defect detection model, achieving an accuracy rate of 94%, showcases the potential of advanced technological solutions in this field. However, these systems still struggle with robustness under varying environmental conditions, a critical factor for reliable aerospace component inspection.

In response to this pressing need, this study embarks on the development of innovative path planning routines for feature rich image acquisition, underpinned by a vision-based control scheme. These routines are designed to accommodate and adapt to variations in surface finish, geometry, and curvature, ensuring that every image accurately and precisely captures the intricate surface features of the component under inspection. This approach is complemented by the formulation of novel image and video acquisition protocols, strategically optimized to maintain flawless image quality across diverse operational conditions. The core objective is to capture images that are not just high quality but are also devoid of disturbances typically encountered in dynamic inspection environments. The study's fo-

cus on developing robust algorithms for image quality analysis addresses the limitations of current machine vision solutions by enhancing data reliability [9]. These algorithms improve upon traditional methods by rejecting data that fails specific quality checks, ensuring only high-standard data is used. This precision is vital for meeting the stringent requirements of aerospace Guidelines and boosting accuracy in defect detection, a crucial aspect in the context of safety-critical inspection processes in the aerospace industry.
The shift towards human-supervised automation in safety-critical inspection processes is catalyzed by an unwavering commitment to uphold high-quality standards amidst the ever growing demand for aerospace components. Integration of machine vision systems in this domain is a transformative step, elevating the traditional role of human inspectors. These systems amalgamate the precision of state-of-the-art cameras with the robust analytical prowess of advanced AI methods and computer vision algorithms, heralding a new era of inspection processes. They not only enhance accuracy and reliability, effectively minimizing human error and fatigue, but also significantly improve traceability, accountability, and precision in inspections. This evolution in inspection technology is crucial for meeting the high demands and rigorous standards of the aerospace industry, ensuring more thorough and reliable evaluations of aerospace components.

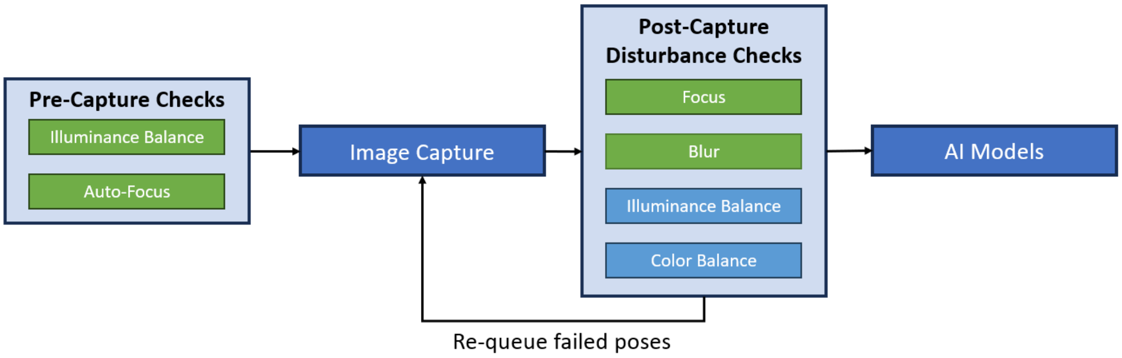


Figure 1.2: System Disturbance Rejection System

The integration of machine vision systems in aerospace inspections, which combines state-of-the-art cameras with advanced computer vision algorithms, represents a transformative step towards more reliable, objective, and efficient inspection processes. Key to the effectiveness of these systems is the acknowledgment that automated defect detection models are only as good as the input images they receive. Recognizing this, our research incorporates a critical framework for both pre-capture optimization and post-capture quality checks, ensuring that the image quality is optimal before it passes into the defect detection pipeline.

Pre-capture optimization involves calibrating the imaging system to suit the specific inspection environment and the aerospace component being examined. This step includes adjusting factors such as lighting, angle, and focus to capture the maximum number of surface features and minimize divergence from training set. Post-capture quality checks then scrutinize these images to verify their clarity and completeness, ensuring that any defect detection made by the system is based on the highest quality data. This dual approach of optimization and quality assurance significantly diminishes the likelihood of false positives and overlooked defects, thereby enhancing the overall accuracy of the defect detection process.

This research marks a pivotal advancement in aerospace industry inspections. By integrating these comprehensive image quality control measures, we aim to validate the augmented capabilities of these advanced systems in actual manufacturing environments. This approach directly addresses the pressing need for agility, precision, and adaptability in the inspection of aerospace parts, considering their diverse geometries, materials, and defect detection criteria. The study's emphasis on the integrity and clarity of each image for analysis reflects a fundamental belief in the importance of 'good data.' This approach underlines a key hypothesis: no bad data should be allowed into our model. It represents a commitment to rigorous data quality standards, which is crucial for ensuring accuracy and efficiency in the aerospace sector. By prioritizing high-quality data, the study aims to significantly enhance the accuracy and reliability of inspection processes.

1.2 Literature Review and Related Work

The work of Abhishek Arun Kulkarni in "Motion Planning and Image Capturing for Robotic Inspection of a Curved Surface Subject to Imaging Constraints"[10] offers valuable insights into advanced robotic inspection techniques. Kulkarni's research primarily focuses on optimizing the generation of robotic viewpoints using clustering techniques, particularly the K-means binary search algorithm. This method strategically forms clusters based on the geometric properties of the surface, ensuring that each cluster's curvature aligns within the camera's depth of field. Such an approach is pivotal in achieving optimal image quality, as it carefully considers the camera's field of view relative to the surface curvature.

Furthermore, Kulkarni emphasizes the importance of maintaining optimal imaging conditions. His research underlines the necessity of positioning the camera's optical axis perpendicular to the surface at a distance precisely equal to the focal length. This arrangement is crucial for capturing the highest quality images, as it ensures the subject remains within the camera's sharp focus area.

A significant contribution of Kulkarni's thesis is the introduction of the bit-masking technique in automated waypoint generation. This technique is particularly relevant for robotic arms like the UR5e, which possess six degrees of freedom. This modification eliminates the ambiguity in the final image orientation and guarantees consistent, comprehensive coverage of the inspected surface, irrespective of the robot's orientation. By adopting a circular bit-mask, slightly smaller than the minor field of view dimension, Kulkarni's approach effectively addresses potential inconsistencies in imaging, ensuring uniformity and completeness in surface coverage.

Incorporating these elements from Kulkarni's research provides a strong foundation for understanding the current state and challenges in robotic inspection. It also highlights the importance of precise planning and control in robotic systems for effective surface inspection. Integrating these concepts into the study showcases the evolution of research in this area, building upon established methodologies to enhance the accuracy and efficiency of robotic inspection systems.

Chapter 2

ROBOT SETUP AND SYSTEM ARCHITECTURE

2.1 Robot Hardware Setup

The primary objective of this study is the development of an advanced End-of-Arm Tool (EOAT) and an accompanying image acquisition and classification system, tailored for high-precision inspection applications within the aerospace industry. This system is a culmination of advanced robotic and imaging technologies, designed to mitigate variances in data quality due to environmental factors, which has been a challenge in manual and automated inspection methods.

At the heart of the EOAT is a high-resolution 61MP Sony mirrorless camera with a macro lens, chosen for its ability to capture extremely fine details necessary for the accurate detection of minute defects in aerospace components. This camera's 61 megapixel count is pivotal for ensuring that even the smallest imperfections are visible, making it an invaluable asset for precision inspections.

Surrounding the camera are 384 individually addressable LEDs. This extensive array of LEDs allows for granular control of the lighting conditions, ensuring that each part is illuminated optimally for inspection. Fine-tuned lighting is crucial for highlighting defects and ensuring that images captured are of the highest quality, devoid of shadows or glare that could obscure crucial details and conforming to the lighting conditions in the training set.

The choice of the Raspberry Pi 4 Model B, with its Power over Ethernet (PoE) Hat, for controlling the hardware of the end-of-arm tool (EOAT) is strategic. It not only simplifies the system's architecture by merging power and network connectivity but also makes the system easy to adopt. This standardization, along with pre-built drivers and software for camera and lighting control, allows users to simply connect the Raspberry Pi to the network and use provided API calls for device control, showcasing an innovative and readily adoptable approach to inspection technology.

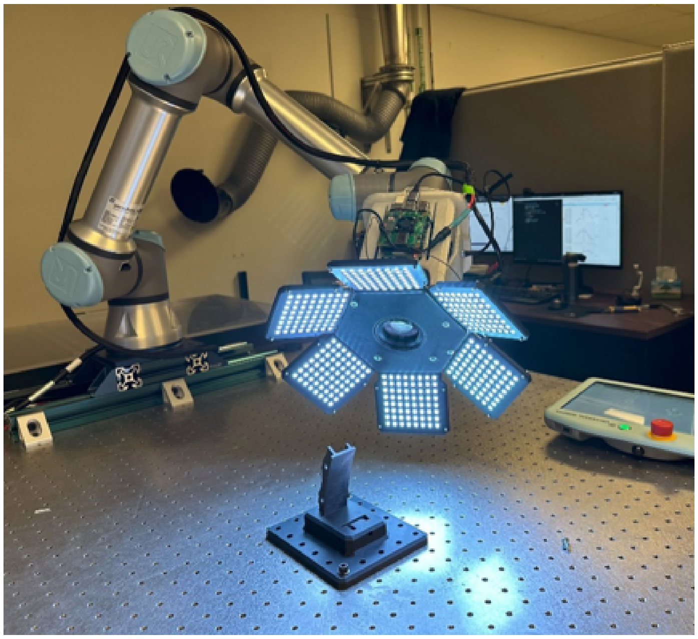


Figure 2.1: UR5e Robot Setup with Mounting LED Panel and Camera at the End-effector

2.2 System Architecture

In the arena of robotic inspection systems, the generation and implementation of viewpoints for precise inspections stand as a testament to the harmonious fusion of spatial data processing, computational algorithms, and robotic engineering. The system, particularly designed for aerospace component inspections, showcases an intricate balance between technical precision and operational efficiency.

The core of the system's operation lies in its advanced software suite, equipped with algorithms for automated viewpoint generation from solely a surface model or point cloud of the part. This aspect of the software ensures that the camera and lighting are always in the optimal position for each inspection task, catering to the unique requirements of various aerospace components.

Central to the robot telemetry generation problem is the formulation of position vectors and quaternions. Each potential viewpoint is represented by a position vector $P = (x, y, z)$, indicating the precise 3D location for the camera. Additionally, the camera's orientation at these points is defined using quaternions $Q = (q_z, q_y, q_z, q_w)$, known for their efficiency in handling complex 3D rotations. These states fully define the position and orientation of the camera at each viewpoint. The integration of these components into a Pose Matrix ensures a unified directive for positioning and orienting the camera.

To enhance this process, the system employs K-means clustering algorithms, which segments the spatial data into clusters based on geometric proximity and similarity of normals. This segmentation, driven by a feature matrix derived from the spatial data, groups points into clusters, each denoting a distinct region of the inspection object. The number of clusters is optimized through a binary search algorithm, ensuring comprehensive coverage within the camera's field of view.

The system's architecture, powered by the Robot Operating System (ROS), plays a pivotal role in implementing these calculated viewpoints. ROS nodes, each designed for specific tasks such as image processing and robot control, are activated to begin the inspection

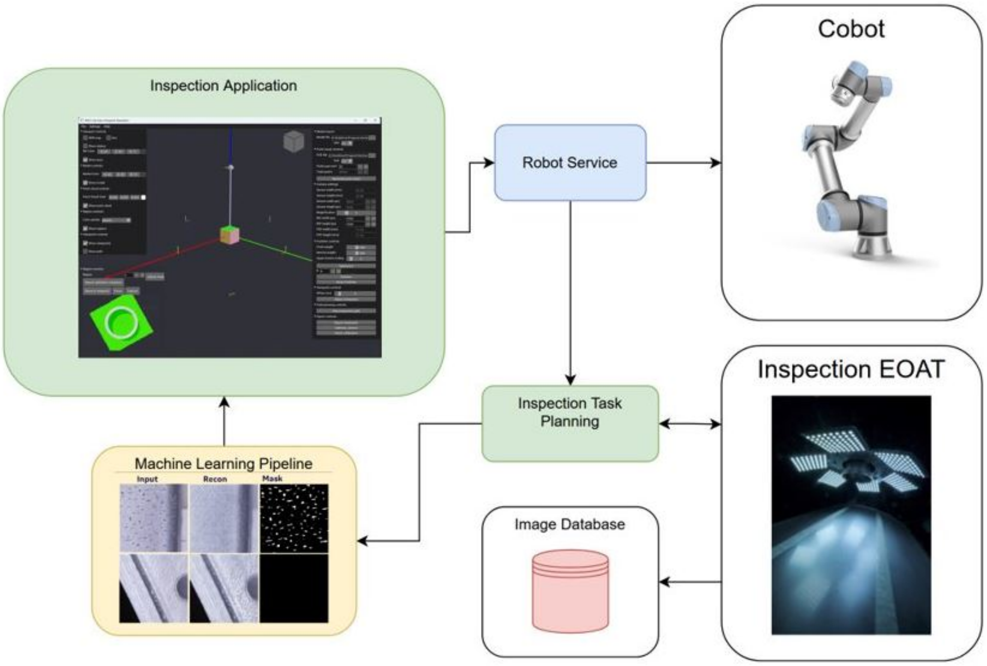
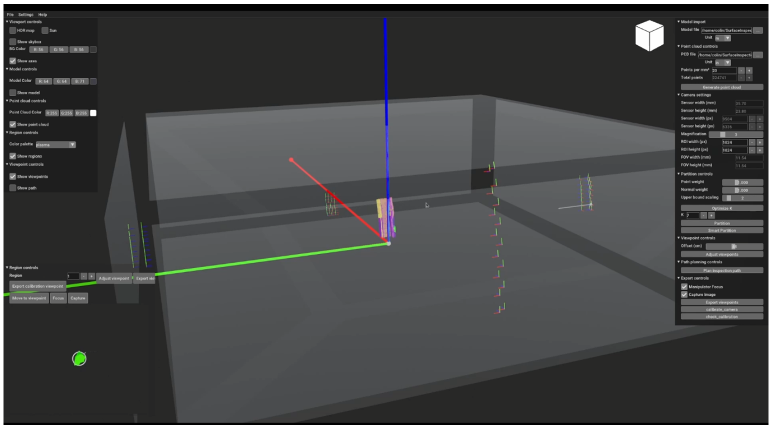


Figure 2.2: System Architecture Overview

process. Among these, the auto-partitioning node is key, as it autonomously generates optimal viewpoints for inspection by calculating the most advantageous angles and positions for image capture. The viewpoints are then organized into logical sequences based on their proximity and orientation relative to the component, allowing the robotic arm to follow an efficient path during the inspection.

The viewpoints generated are then organized into sequences based on their proximity and orientation relative to the component. This organization streamlines the inspection process, creating an orderly and logical path for the UR5e robotic arm to follow. Known for its accuracy, and it has a large domain of possible joint states and end-effector positions. The arm is guided by advanced path planning and control algorithms, ensuring precise and repeatable movements for consistently positioning parts under the camera.

The system's capability for real-time adjustments to the poses, accommodating changes



☐ HDR map ☐ Sun
☐ Show skybox
BG Color R: 56 G: 56 B: 56
☑ Show axes
Model Color R: 64 G: 64 B: 71
☐ Show model
Point Cloud Color R:255 G:255 B:255
☑ Show point cloud
Color palette plasma
☑ Show regions
☑ Show viewpoints
☐ Show path
Region 1

Model file /home/colin/SurfaceInspec
Unit m
PCD file /home/colin/SurfaceInspecti
Unit in
Points per mm² 20
Total points 224741
Sensor width (mm) 35.70
Sensor height (mm) 23.80
Sensor width (px) 9504
Sensor height (px) 6336
Magnification 3
ROI width (px) 1024
ROI height (px) 1024
FOV width (mm) 11.54
FOV height (mm) 11.54
Point weight .000
Normal weight .000
Upper bound scaling 2
K 7
Offset (cm)
☑ Manipulator Focus
☑ Capture Image

Figure 2.3: Auto View-point Generation GUI

in the object's position or responding to sensor feedback, further underscores its accuracy and effectiveness. The result is a highly precise and reliable inspection system capable of detailed inspections, pivotal in industries where high-precision is of utmost importance.

Chapter 3

MOTION BLUR AND OUT OF FOCUS DISTURBANCE REJECTION

3.1 Problem Statement and Need

Addressing the challenges of using macro lenses in robotic inspection systems, particularly for aerospace components, requires a nuanced and control-based approach. Macro lenses are indispensable for their high-resolution imaging capabilities, necessary for detecting minute details [17]. However, their very shallow depth of field means that even minor deviations from the optimal focus distance can result in a significant loss of focus. Additionally, motion blur in dynamic inspection environments, caused by slight movements of either the camera or the object during image capture, can further compromise image quality.

One significant challenge is the reliance on dead-reckoning methods and in-lens autofocus systems in viewpoint generation. Dead-reckoning to the viewpoint alone often lead to inconsistent focus on the region of interest, especially due to the narrow depth of field characteristic of macro lenses. Furthermore, errors in viewpoint generation can lead to variances in defect scale across different cells and parts if paired with in-lens autofocus. This inconsistency can result in inaccuracies in defect identification and measurement, thereby undermining the reliability of the inspection process.

To counter these challenges, a refined approach involving the implementation of high-level control over the manipulator is proposed, specifically aimed at optimizing image sharpness before capturing a high-resolution image for defect detection. The first step towards optimizing focus is to determine a metric for the sharpness of the image.

Subsequently, the robotic system executes a well-planned trajectory along the manipulator's z-axis, the principal axis of the camera. Throughout this trajectory, the system continuously measures and records the sharpness of the image at various z-offsets. This measurement is crucial for identifying the point of optimal sharpness, where the image details are most clearly defined. Once the point of optimal sharpness and its corresponding z-offset are determined, the system returns the manipulator to this precise position for imaging. This ensures that the imaging process captures the highest quality images with the most precise details, significantly enhancing the consistency and reliability of the image data acquisition process.

3.2 Focus-Value Method and Calculation

In the field of image processing, particularly in applications such as inspection, determining the focus of an image is a critical aspect. Human observers have the innate ability to discern whether an image is well-focused; however, computer systems lack this intuitive judgment. Unlike human perception, which can often reliably identify focused images, computer systems necessitate clearly defined criteria for making such distinctions. The creation of precise, quantifiable benchmarks is essential to enable these systems to accurately determine focus. The establishment of these benchmarks is a crucial step in automated image processing and analysis, facilitating the effective identification of focused images by the computer.

Autofocus algorithms in image processing are generally categorized into two types: those based on the statistical variance of pixel values, and those that analyze the spatial-frequency content of the image. This research primarily utilizes the former category, employing methods based on the computation of the variance of the image gradient or image Laplacian. The selection of this approach is based on two key considerations[8].

Firstly, the aim is to provide a system where the focus quality of an image can be visually ascertained. Utilizing algorithms that assess the variance in image gradients or Laplacians generates intuitive and visually interpretable results. These results facilitate more informed decisions regarding the focus quality of an image, both by the system and the user.

Secondly, the chosen approach is designed for real-time application in dynamic environments. As the robotic system maneuvers and adjusts, the algorithm continuously calculates focus values to determine the most suitable position for image data acquisition. This real-time calculation is vital for scenarios where the subject or the imaging apparatus is in motion, ensuring consistent capture of well-focused images even under varying conditions. Therefore, this research emphasizes methods that provide clear visual feedback for focus assessment and are capable of supporting live, continuous focus calculation. This aligns with the practical needs of high-precision robotic imaging systems, where real-time focus adjustment is essential.

Several methods have been proposed in the literature to solve the problem of autofocus [14]. These methods define a focus function which measures the relative focus-value of images at different object distances, identifying the distance at which the function returns the largest value as the best focus.

The Grey Level Local Variance Method provides a quantitative approach to evaluate this focus. This method is predicated on the understanding that a well-focused image will display significant variations in grey levels across its pixels. The method utilizes a series of equations to calculate these variations, both locally within small regions of the image and globally across the entire image.

In the Grey Level Local Variance Method, it is anticipated that an image with good focus will exhibit significant variations in grey levels. The measurement of focus in this method is derived from the global variance, which is determined by calculating the variance at every point (m,n) across the image, with m varying from 1 to M and n from 1 to N.

The local variance at a point (m,n) in the image is calculated using the formula:

$$lv(m,n) = \frac{1}{w_x w_y} \sum_{i}^{w_x} \sum_{j}^{w_y} [I(m+i, n+j) - \bar{I}]^2 \tag{3.1}$$

In this equation, $I(m+i, n+j)$ represents the intensity of a pixel and $\bar{I}$ is the mean intensity value within a window of size $w_x \times w_y$ around the point (m,n). This calculation finds the variance of grey levels in the specified window, reflecting the local focus quality.

The average grey level within the window is calculated as:

$$\bar{I} = \frac{1}{w_x w_y} \sum_{i}^{w_x} \sum_{j}^{w_y} I(m+i, n+j) \tag{3.2}$$

This average is used to assess the deviation of pixel intensities in the local window, necessary for calculating local variance.

The overall focus measure of the image is given by the global variance, calulated as:

$$VAR(I) = \frac{1}{NM} \sum_{m}^{M} \sum_{n}^{N} [lv(m,n) - (\bar{lv})]^2 \tag{3.3}$$

Here, $lv(m,n)$ is the local variance, and $\bar{lv}$ is the mean of all local variances. This equation assesses how much the local focus quality varies across the entire image.

The average of all local variances is computed by:

$$\bar{lv} = \frac{1}{NM} \sum_{m}^{M} \sum_{n}^{N} lv(m,n) \tag{3.4}$$

This mean provides a baseline to calculate the global variance. By utilizing these calculations, the Grey Level Local Variance Method comprehensively assesses an image's focus.

A higher global variance suggests a sharper and better-focused image. This method is particularly significant in automated image processing and analysis, providing a systematic and objective way to evaluate image focus, overcoming the subjective limitations of human visual assessment.

In image processing, the Sobel filter is a well-known technique used to detect edges within an image. It functions through the application of two 3 × 3 convolutional kernels, designed to highlight horizontal and vertical edges by calculating the gradient of the image intensity at each pixel. These kernels, often referred to as the Sobel operators, are central to the Gradient Magnitude Based Methods, which are used to assess image focus.

The Sobel operator plays a crucial role in this method. It is applied using specific convolution masks, designed to detect horizontal and vertical edges in the image. The masks, represented as S_x and S_y, are defined as:

$$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \qquad S_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \tag{3.5}$$

The S_x kernel is used to detect horizontal edges by calculating the gradient along the x-axis, while S_y detects vertical edges by focusing on the gradient along the y-axis. By applying these kernels to an image, the Sobel filter effectively enhances areas of high spatial frequency, which correspond to edges.

Sharper edges in an image are indicative of better focus, thus making the analysis of image gradients a critical factor in focus determination. The Gradient Magnitude Based Methods leverage this principle by estimating the gradient magnitude at each point in the image, which is particularly useful in applications like a high resolution image.

The gradient magnitude at each point (m,n) is calculated using the formula:

$$S(m,n) = \sqrt{[G_x(m,n)]^2 + [G_y(m,n)]^2} \tag{3.6}$$

In this equation, $G_x(m,n)$ and $G_y(m,n)$ are the results of convolving the input image $I(m,n)$ with the Sobel kernels S_x and S_y. This process yields a measure of edge sharpness at each point by combining horizontal and vertical edge information.

To aggregate the gradient information across the image, the variance of gradient magnitudes is computed. This focus measure is defined as:

$$SOB\_VAR(I) = \sum_{m}^{M} \sum_{n}^{N} [S(m,n) - \bar{S}]^2 \quad for\ S(m,n) > T \tag{3.7}$$

Where T is a threshold value, $S(m,n)$ is the gradient magnitude, and $\bar{S}$ is the mean of these magnitudes. This variance calculation highlights areas with significant edge sharpness, indicative of a well-focused image.

The average of all gradient magnitudes across the image is given by:

$$\bar{S} = \frac{1}{NM} \sum_{m}^{M} \sum_{n}^{N} S(m,n) \tag{3.8}$$

This mean value is utilized to understand the deviation of each point's gradient magnitude from the average, aiding in the variance calculation.

Through these methods, the Gradient Magnitude Based Methods, utilizing the Sobel filter, provide an objective and quantitative approach to assess the focus of an image, a key aspect in ensuring image quality in high-precision fields.

In image processing, the Second Derivative Based Methods, particularly using the Laplacian filter, are employed to accentuate high spatial frequencies that are associated with sharp edges in an image. These methods focus on enhancing these frequencies to bring forth finer image details that are crucial in determining focus quality.

The Laplacian operator is a well-known second derivative operator used in this method. It is represented by a convolution mask, which is applied to the input image to highlight areas of rapid intensity change, corresponding to edges. The Laplacian mask is defined as:

$$L = \frac{1}{6}\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \tag{3.9}$$

This mask focuses on both horizontal and vertical edge details by evaluating the second derivative of the image intensity. The factor $\frac{1}{6}$ is a normalization constant.

The convolution of the input image $I(m, n)$ with the Laplacian mask L is performed at each point in the image. This process emphasizes the edges by highlighting areas where the intensity changes significantly. The result of this convolution is represented as $L(m, n)$.

To aggregate the edge information across the image, the variance of the absolute values of the Laplacian convolution is computed. This focus measure is defined by the equation:

$$LAP\_VAR(I) = \sum_{m}^{M}\sum_{n}^{N}[|L(m, n)| - \bar{L}]^2 \tag{3.10}$$

Here, $|L(m, n)|$ represents the absolute value of the Laplacian convolution at each point, emphasizing the strength of edges irrespective of their direction. $\bar{L}$ is the mean of these absolute values across the image. This variance calculation assesses how much the edge sharpness varies throughout the image, with a higher variance indicating more pronounced edges and thus a better-focused image.

The mean of the absolute values of the Laplacian convolution across the image is calculated as:

$$\bar{L} = \frac{1}{NM} \sum_{m}^{M} \sum_{n}^{N} |L(m, n)| \tag{3.11}$$

This mean serves as a reference point to determine the deviation of each point's edge strength from the average, aiding in the variance calculation.

Through these methods, the Second Derivative Based Methods, particularly with the use of the Laplacian filter, offer a precise and quantitative approach to evaluate the focus of an image. This method is especially significant in fields requiring detailed edge analysis to ascertain image quality.

In the intricate process of robotic inspection, particularly for systems equipped with high-resolution cameras, the focus region's quality is of utmost importance. A specific area of interest, typically the central 1024×1024 pixel region of the captured image, is scrutinized due to its critical role in defect detection. This focus region is selected based on the recognition that, in cases of complex surface shapes, it's often only the central portion of the image that remains in sharp focus. Consequently, this region becomes the primary target for analysis to ensure the highest quality of the captured data.

To assess the focus quality within this region, two distinct filtering techniques are employed: the Sobel and Laplacian filters. These filters are instrumental in edge detection, a key factor in determining image focus. The Sobel filter operates on the principle of the first derivative, identifying gradients in intensity that signify edges within the image. Conversely, the Laplacian filter, based on the second derivative, highlights areas of rapid intensity change, thereby indicating edges more prominently.

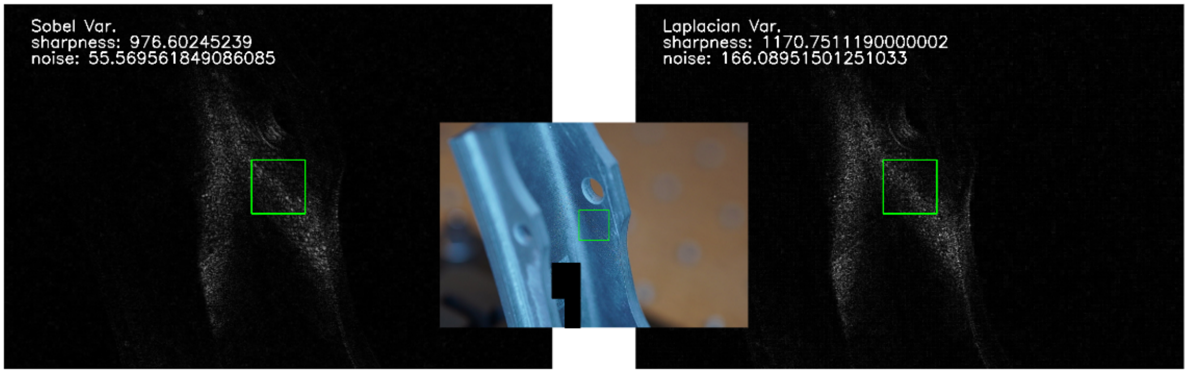


Figure 3.1: Comaparison Between Sobel and Laplacian Filters' Focus-Value and Noise.

A crucial aspect of this analysis is the quantification of focus values and noise within the central focus region. The presence of 'white particles' in the edge detection results is indicative of higher focus values - essentially, more defined edges correlate to a sharper image. However, a pivotal distinction arises in the level of noise generated by each filter. Noise, in this context, refers to the false identification of edges, which can lead to misleadingly high focus values - a phenomenon known as false positives.

Comparative analysis reveals that while the Laplacian filter may produce higher focus values due to its sensitivity to rapid intensity changes, it also introduces significantly more noise compared to the Sobel filter. This noise is detrimental to the accuracy of focus assessment, as it can erroneously suggest a well-focused image when, in reality, the sharpness may be inadequate.

The disparity in noise levels between the two filters is attributed to their underlying principles. The Sobel filter, being a first derivative filter, is less susceptible to minor variations in pixel intensity, thereby reducing the likelihood of false edge detection. On the other hand, the Laplacian filter, based on the second derivative, is more prone to amplifying noise, which can skew focus assessments, especially in dynamic environments where the camera or the inspected object is in motion.

In light of these findings, the decision to favor the Sobel filter over the Laplacian filter is a strategic one. The Sobel filter's reduced noise level makes it a more reliable tool for assessing focus in the central region of the image. This reliability is crucial for ensuring that the system accurately determines the optimal position of the robot's end-effector along the z-axis. Accurate focus assessment directly impacts the quality of the captured images, which in turn influences the accuracy of the defect detection process.

The choice of the Sobel filter underscores the commitment to precision and accuracy in robotic inspection systems. By minimizing noise and false positives in focus assessment, the system enhances its ability to provide consistent and reliable data for defect detection, an essential requirement in industries where image quality is paramount.

3.3 Developing Manipulator Focus Algorithm

In the domain of robotic inspection systems, particularly those utilizing macro lenses with a 1:3 magnification ratio, the challenge of maintaining optimal focus is amplified due to the lens's inherently limited depth of field, which is typically around 2cm. This narrow depth of field means that any deviation beyond this 2cm range from the focal point results in a loss of image sharpness, rendering the image blurry and unsuitable for precise defect detection. Consequently, accurately setting the camera position for each inspection task is paramount, yet it is complicated by various uncertainties inherent in real-world applications.

These uncertainties include disparities between the actual setup and its representation in simulation environments like RViz, the complexity and size variations of the parts being inspected, and the positioning of the camera on the robotic arm. Additionally, variances in the automated viewpoint generation pipeline contribute to the challenge, as they can lead to inconsistencies in the camera's positioning relative to the optimal focal distance.

Users can set an initial offset, a starting point for the focal adjustments, based on calculations of the optimal focal distance. However, due to the aforementioned uncertainties, achieving focus consistently at this initial offset is not always feasible. Real-world setup variations, such as slight misalignments or differences in the positioning of the camera and the object, can cause deviations from the expected focal point. The complexity and size of the part being inspected also introduce variables that can affect the camera's focus, as different shapes and sizes may require adjustments in the camera's position to maintain the optimal focal range.

Furthermore, the camera's placement on the robotic arm and the inherent inaccuracies in the auto viewpoint generation pipeline can lead to variations in the camera's actual versus expected position, further complicating the task of consistently achieving sharp focus. These factors collectively necessitate the development of an algorithm capable of dynamically finding the optimal focal distance, adjusting for the various uncertainties and ensuring that the images captured are consistently focused and of high quality, suitable for detailed inspection tasks.

Therefore, to mitigate these challenges and enhance the reliability of the inspection process, the proposed approach involves developing and implementing a robust algorithm specifically designed to adjust the camera's position and focus in real-time. This algorithm will account for the various uncertainties and variables in the system, enabling it to dynamically determine and adjust to the optimal focal distance for each inspection task. The goal is to ensure that despite the inherent challenges and complexities, the inspection system consistently captures sharp, focused images, which are crucial for accurate defect detection and quality control.

The Manipulator Focus Algorithm is intricately designed to optimize the focus of a camera mounted on a robotic arm, especially crucial when using macro lenses for high-resolution imaging. This algorithm is particularly relevant for inspections where the detection of minute, safety-critical defects is essential. The algorithm can be described as follows:

Algorithm 1 Manipulator Focus Algorithm for Robotic Inspection System

```
Initialize: Set initial offset and define movement range.
Define: Initial Range $R = \pm 1$cm along the z-axis of the end-effector.
Define: Threshold focus value, Threshold.
Initialize: Optimal focus value, $F_{\text{opt}} = 0$.
Initialize: Optimal position, $P_{\text{opt}}$.
while $F_{\text{opt}} <$ Threshold do
    for each z-offset $z$ in range $R$ do
        Move camera to position $z$.
        Capture image at position $z$.
        Calculate focus metric, $F_{\text{current}}$, using Sobel filter.
        if $F_{\text{current}} > F_{\text{opt}}$ then
            Update $F_{\text{opt}}$ to $F_{\text{current}}$.
            Update $P_{\text{opt}}$ to current position.
        end if
    end for
    Expand range $R$ by ±1cm and repeat if needed.
end while
if $F_{\text{opt}} \geq$ Threshold then
    Move camera to $P_{\text{opt}}$ at reduced speed (60% reduction).
    Fine-tune position within ±1cm of $P_{\text{opt}}$ to confirm optimal focus.
    Update $P_{\text{opt}}$ to this final position.
end if
Move camera to $P_{\text{opt}}$ for optimal focus.
Capture the final image at $P_{\text{opt}}$.
Move to the next pose.
```

An iterative process is employed to find the optimal focus:

1. **Initialization**: The algorithm begins by establishing an initial reference point, known as the initial offset. This serves as the starting position from which the system will make adjustments to find the optimal focus. An initial movement range of ±1cm is set along the z-axis of the end-effector. This range is deliberately chosen to provide sufficient scope for the camera to adjust its focus and locate the point where the image sharpness is maximized.

2. **Focus Metric Calculation**: At each z-offset within the range, the camera captures an image. For each image, the algorithm calculates a focus metric, F_{current}, which quantitatively assesses the sharpness of the image. The Sobel filter, known for its edge detection capabilities, is employed here. By analyzing the variance of the Sobel-filtered image, the algorithm gauges the level of focus, with higher variance indicating sharper images.

3. **Optimization Loop**: This loop is the heart of the algorithm. It iteratively compares the calculated focus metric at each position with the current optimal focus value, F_{opt}. When F_{current} exceeds F_{opt}, the algorithm updates F_{opt} to this new value and records the corresponding camera position as P_{opt}, the new optimal position. The process continues until F_{opt} reaches or surpasses a predefined threshold, indicating that the image at P_{opt} has achieved the desired level of sharpness.

4. **Range Expansion and Repetition**: If the optimal focus is not achieved within the initial range, the algorithm expands the movement ± 1 cm more in range to explore further. This expansion is crucial to cover a broader scope, increasing the likelihood of finding the point of optimal focus.

5. **Fine-Tuning Focus**: Once the optimal focus value surpasses the threshold, the robot moves to the current optimal focus value position. At this stage, the robot's

movement speed is reduced by about 60% for precise positioning. The robot then makes additional adjustments within a ±1cm range from this position to fine-tune and confirm the optimal focus value. This step is critical for ensuring that the final focus position is as accurate as possible, ensuring the highest image quality for inspection.

6. **Final Positioning and Imaging**: Once the optimal focus value is achieved, the camera is moved to P_{opt}. This position is where the camera can capture the sharpest and most detailed image, crucial for accurate defect detection. The final image captured at this position is then used for inspection purposes, offering the highest quality and clarity needed for precise analysis.

The collection of rosbag data has been pivotal in understanding the performance and efficiency of the initial manipulator focus algorithm. By analyzing focus values, offset positions, and time taken for each inspection pose, the system's developers gained valuable insights into its operational dynamics. This analysis revealed that the initial approach, while accurate, was time-consuming, particularly for each pose's focusing phase.

Armed with this data, the team realized the need to reduce the time taken per inspection pose without compromising the quality of focus. The solution came in the form of an algorithmic update - the integration of a feature that remembers the previous optimal offset position. By starting each new inspection pose from the last known optimal position, the system significantly reduces the time required to achieve the optimal focus. This update has been a direct result of analyzing the collected data, demonstrating a responsive and adaptive system design.

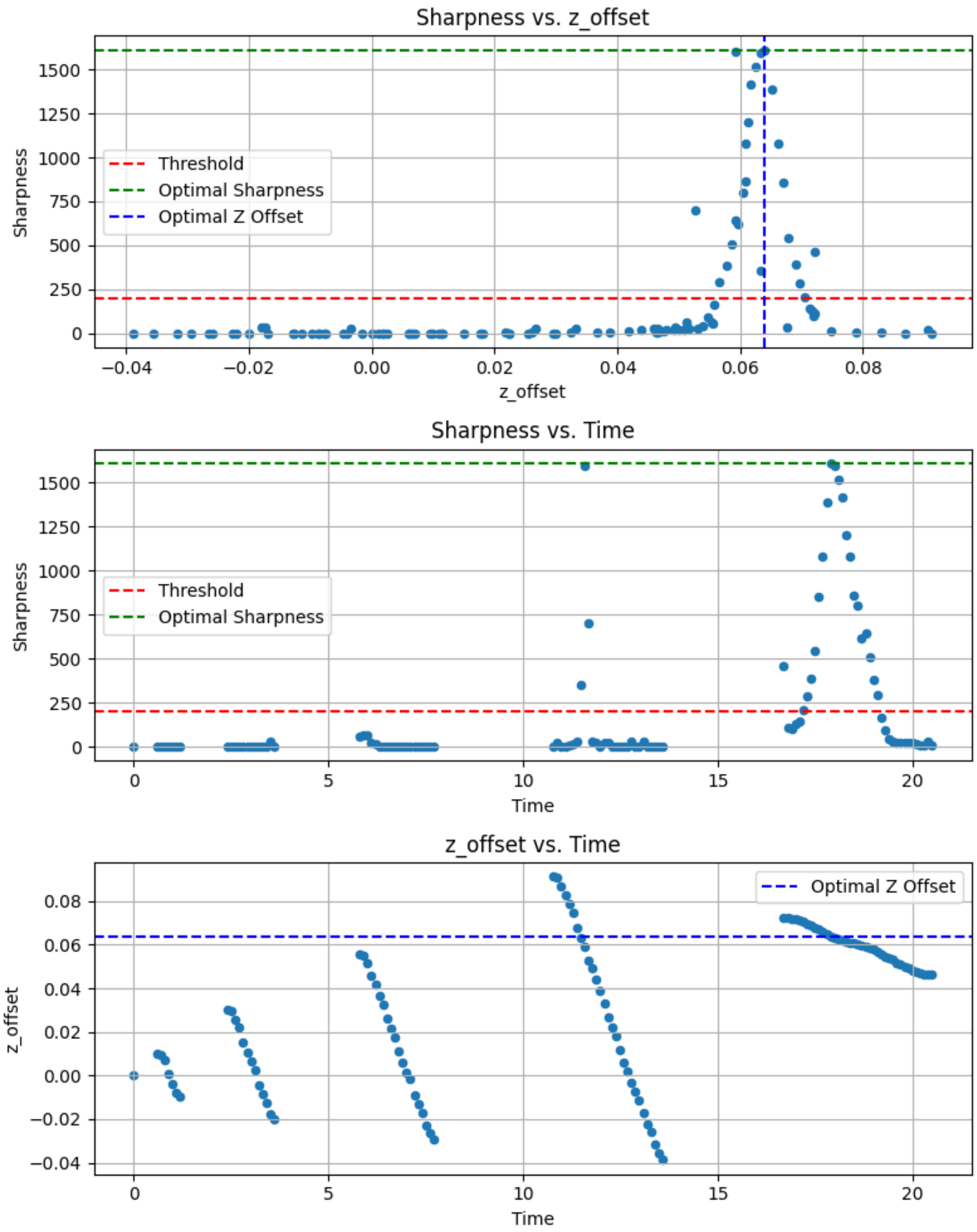


Figure 3.2: Rosbag Data with Initial Manipulator Focus Algorithm. 1.) Plot shows Focus Value vs. Offset (m), 2.) Focus Value vs. Time (sec), 3.) Offset (m) vs. Time (sec)

In the initial version of the algorithm, the robotic system started each new inspection viewpoint without prior knowledge of the optimal focus position, leading to longer durations to achieve focus. This process, while effective, was time-consuming, especially if the user-set initial offset was significantly far from the actual focal point. Now, the system retains the memory of the previously identified optimal offset position. When moving to a new viewpoint, it initially positions itself based on this remembered offset. This approach significantly reduces the time taken to find the optimal focus for each subsequent viewpoint, as the starting point is already close to the expected focal distance. This enhancement has proven to reduce the focusing time from approximately 27 seconds to about 6.8 seconds per viewpoint, thereby increasing the throughput of the inspection process.

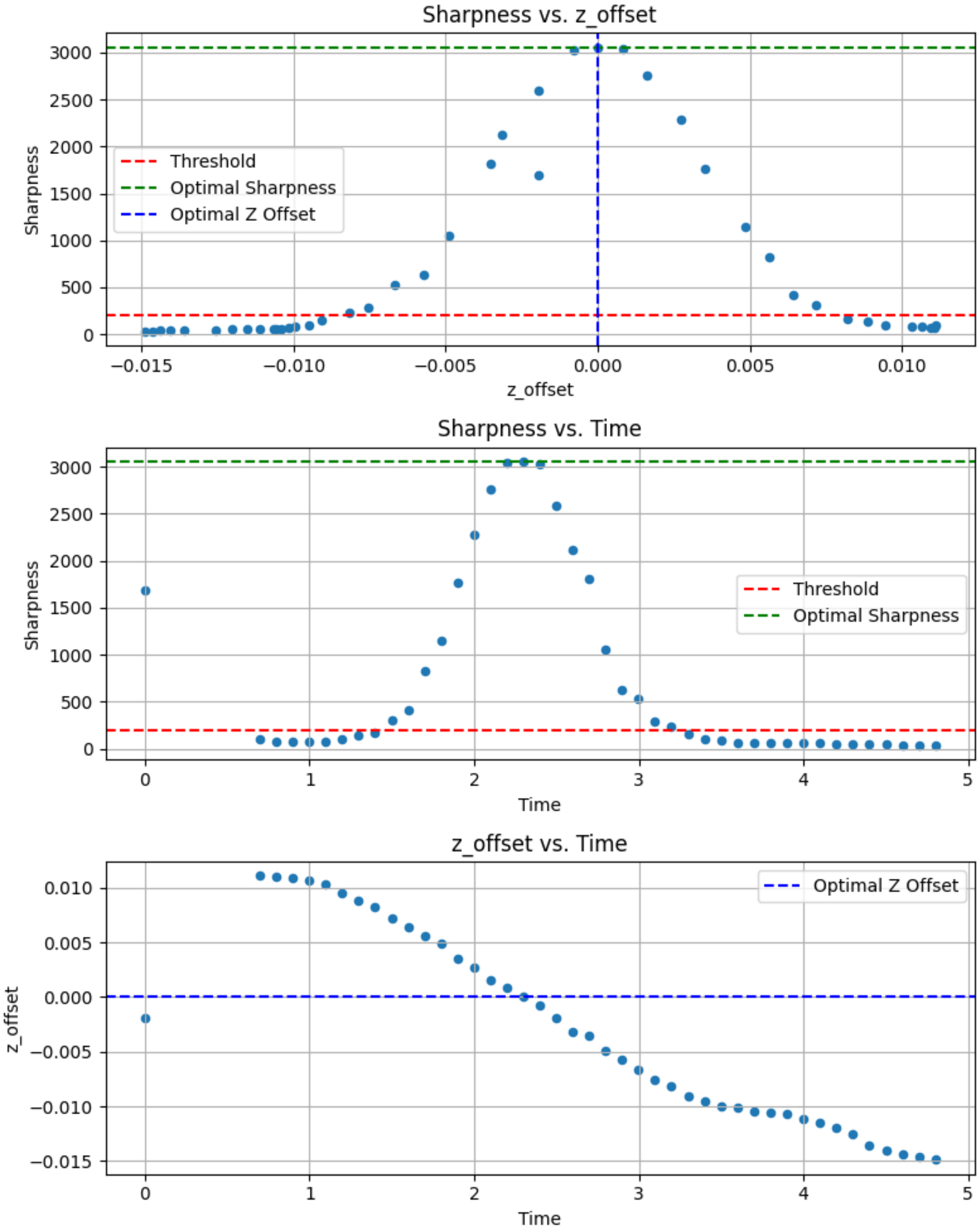


Figure 3.3: Rosbag Data with Update Manipulator Focus Algorithm. 1.) Plot shows Focus Value vs. Offset (m), 2.) Focus Value vs. Time (sec), 3.) Offset (m) vs. Time (sec)

Another innovative feature is the initial offset calibration integrated into the system's GUI. Previously, users had to estimate the initial offset value, which could lead to inefficiencies if the guess was not close to the optimal focal distance. The new calibration feature automates this process by moving the robot towards the object and then backward, systematically determining the optimal starting offset position. This automated calibration process not only eliminates the need for guesswork but also ensures that the initial offset is as close to the optimal focus position as possible, further enhancing the system's efficiency and accuracy.

By employing this algorithm, the robotic inspection system can consistently capture high-quality images, essential for detecting minute defects, especially when using macro lenses with shallow depth of fields. The algorithm's design ensures the adaptability and precision required in dynamic inspection environments, making it a critical component of advanced robotic inspection systems.

3.4 Camera Lens Autofocus vs. Manipulator Focus Algorithm

In the domain of high-precision robotic inspection systems, especially those utilizing macro lenses, the decision to opt for a manipulator focus algorithm over the camera lens' autofocus is driven by the critical need for precision and consistency in imaging. This strategic choice is backed by several compelling reasons:

1. **Inherent Limitations of In-Lens Autofocus:**

   Uncertainty and Positional Error: While in-lens autofocus systems are generally efficient, they may not meet the exacting precision required in robotic inspection systems. The viewpoint generation mechanism inherently introduces positional errors, resulting in discrepancies between the camera's actual position and its optimal focal distance. This challenge is particularly pronounced with macro lenses, which have a shallow depth of field, thereby necessitating extremely precise focusing.

   Beyond the limitations of the autofocus mechanism itself, several factors contribute to the variance in scale of surface features, crucial in high-precision tasks. These include:

Real-world Setup Variability: In practical applications, inconsistencies in the setup, such as variations in the environment or alignment, can introduce additional uncertainty, affecting the camera's focusing accuracy.

Simulation vs. Reality: Discrepancies between simulated models and actual conditions can lead to deviations in expected outcomes, especially regarding the camera's position and focusing parameters.

Camera Positioning and Focal Length: The starting point of the camera's focal length and its precise positioning play a critical role in achieving the desired focus. Any misalignment can significantly impact the accuracy of the inspection.

Complexity of Part Shape and Size: The diverse shapes and sizes of the parts being inspected add another layer of complexity. The variance in these physical characteristics can affect the camera's ability to maintain a consistent focus across different viewpoints.

Generated Viewpoints: The viewpoints generated for inspection, whether through manual setting or algorithmic determination, can vary widely. This variance might lead to differing focal distances and angles, further complicating the focusing process and potentially introducing errors in scale representation.

These factors collectively contribute to the variances that affect the relationship between the camera's focal distance and its positioning relative to the part being inspected. Such variances are critical as they directly influence the scale of surface features captured in the images, which is a key determinant in the accuracy of defect detection algorithms.

Variance in Image Scale: A critical but often overlooked issue with autofocus systems is their tendency to alter the scale of surface features within images. This variance in scale is detrimental in precision inspection systems where consistent scale and dimensional accuracy are paramount. Such scaling inconsistencies could compromise the effectiveness of machine learning-based defect detection algorithms, potentially resulting in false positives or negatives.

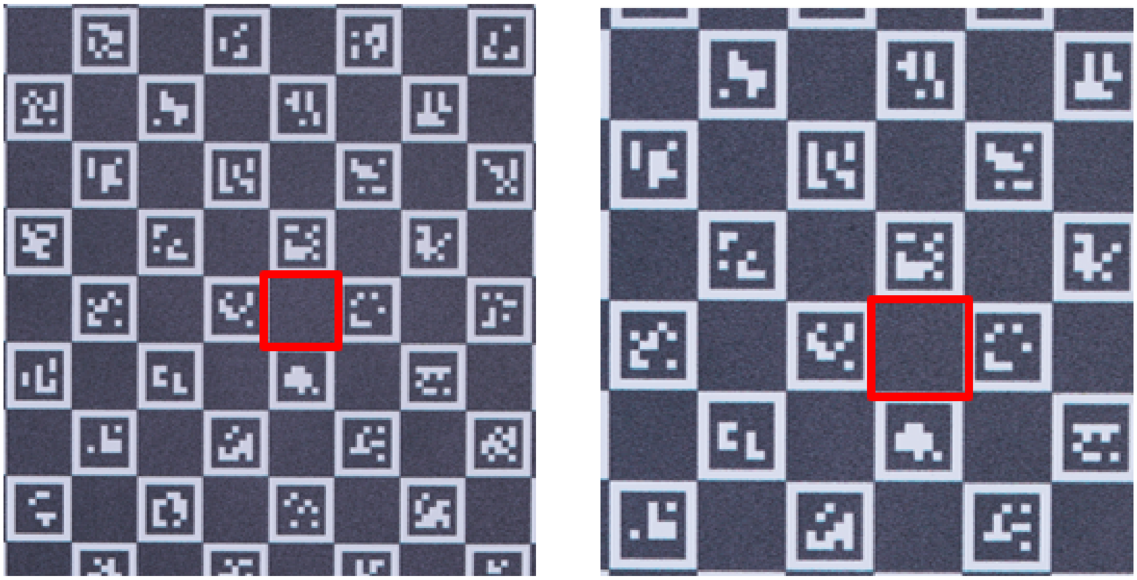


Figure 3.4: In-Lens Autofocus Image Comparison Between Two Different Variance Images

2. **Comparative Testing - In-Lens Autofocus vs. Manipulator Focus Algorithm:**

   To substantiate the effectiveness of the manipulator focus algorithm, a controlled comparative test can be conducted. This involves capturing images of a markerboard at varying errors in initial offset (from 0mm to 40mm), first using the camera's autofocus at this location, and then starting the manipulator focus algorithm from the same place and then applying the manipulator focus algorithm.

   The test includes selecting a defined area on the markerboard and measuring the average pixel size of its sides for each image. This quantitative approach facilitates a direct comparison of how the variance in image scale changes relative to positional error under different focusing methods.

   Analyzing the consistency of pixel size in both sets of images will provide insights into the impact of each focusing method on image scale uniformity. Any disparity in image scale directly influences the accuracy of defect detection when using machine learning algorithms.

   Results Analysis: The results revealed that while the manipulator focus algorithm consistently maintains a constant pixel-to-length ratio within a 10-pixel length of the line segment, irrespective of variance changes, the camera's in-lens autofocus system shows an increase in pixel length of the line segment with increasing variance. This result indicates that in-lens autofocus introduces variance in scale of surface features. Such variance can lead to erroneous defect detection – smaller defects might be overlooked (false negatives) or non-defective features might be incorrectly flagged as defects (false positives) due to altered scale.

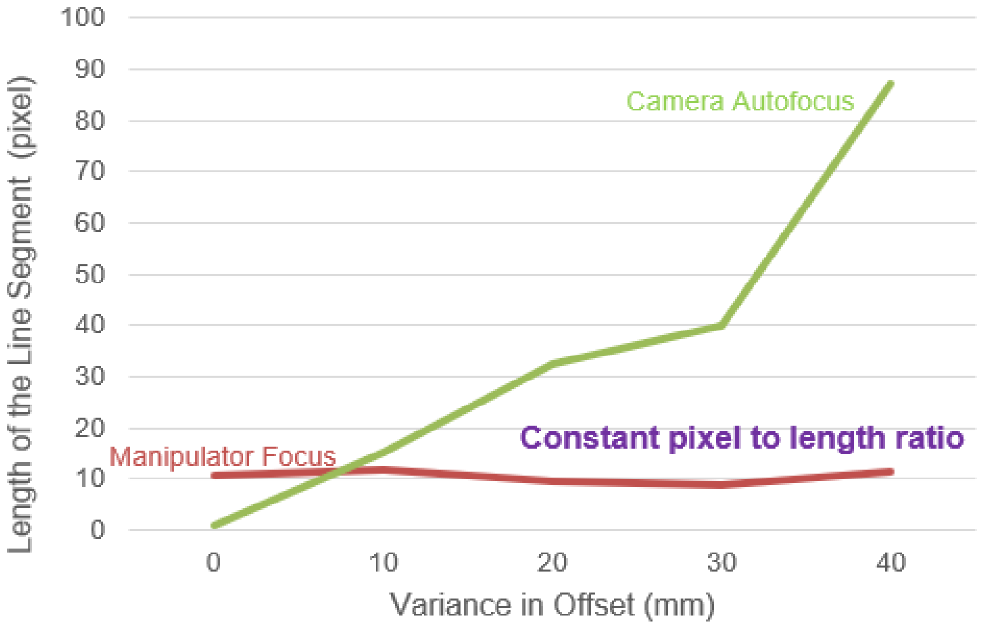


Figure 3.5: Comparison In-Lens Autofocus and Manipulator Focus with Different Variances in Offset

3. **Implications and Operational Efficiency:**

   Despite the quicker focusing capability of in-lens autofocus, its marginal inaccuracies render it less suitable for high-precision tasks. The manipulator focus algorithm, though potentially slower, ensures a superior level of focus accuracy and consistent magnification.

   The manipulator focus algorithm is integral to the overall efficiency of the inspection system. By maintaining a robust and consistent image scale, it lays a solid foundation for reliable and accurate machine learning analyses, a crucial factor in sectors like aerospace component inspection where precision is critical.

Chapter 4

EVEN ILLUMINANCE DISTRIBUTION ON COMPLEX SURFACE DISTURBANCE CHECK

4.1 Problem Statement and Need

The proposed addition emphasizes the need for the inspection system to operate effectively in uncertain environments, where ambient lighting conditions are variable and can significantly impact image quality and consistaency. Recognizing this challenge, the system is designed to observe and counteract these ambient sources of variance, ensuring consistent illuminance across different operational settings [6]. This is crucial as the system might be deployed in locations with lighting conditions different from where it was trained. To enhance the defect detection AI model's robustness against such variances, digital data augmentation techniques are employed, complemented by the active even illuminance control. This approach aims to maintain the accuracy and reliability of the inspection system, regardless of the changing environmental lighting conditions.

Prior attempts at illuminance optimization, such as Mark Gerges's 2021 Master's thesis [5], have shown promise. Gerges demonstrated the use of a bounded value least squares optimization for even light distribution on flat surfaces. However, this approach had limitations: it lacked a feedback loop, and did not consider the texture properties of the part or relied on a controlled lighting envirionment. This gap highlights the need for a more versatile and dynamic illuminance control system capable of operating without knowledge of the environmental lighting conditions.

Digital data augmentation strategies, commonly employed to simulate diverse lighting conditions, often fall short in capturing real-world lighting complexities. This shortfall impacts the performance of models trained on such datasets when applied in actual lighting conditions. The inadequacy of these techniques in replicating true lighting scenarios underlines the need for an advanced approach to illuminance management in imaging environments.

The proposed solution involves an advanced control system integrating an LED array with a camera setup. This system leverages depth and normal maps of the subject to estimate and manage changes in illuminance across the part's surface. The goal is to achieve uniform light distribution using model-predictive control techniques, ensuring consistency in brightness across all images.

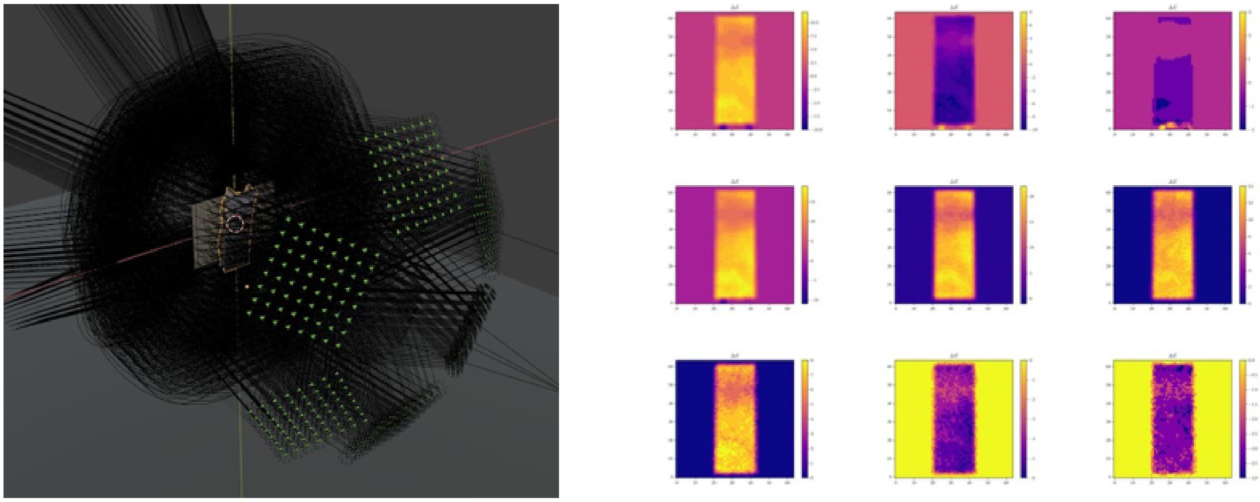


Figure 4.1: Simulation environment in Blender. Positions of LEDs are shown in green, angle of irradiation shown in black. The part is outlined in orange, and the camera is in the middle of the LED array pointing towards the part (left), Example of change in illuminance across part after change in LED intensity from data gathered in Blender. (right)

At the core of this system is a mathematical model that dictates the illuminance at each point on the part's surface. This model takes into account the intensity of the LEDs (denoted as I), and combines it with factors like the direction of light, the surface normals, and the distance between each LED and the surface point. The equation as:

$$E = A * I \tag{4.1}$$

Succinctly represents this relationship, illustrating how changes in LED intensity directly affect the overall illuminance (E) based on the matrix representing geometric relationship between each LED and each point on the surface (A).

To accommodate dynamic changes in the environment, the system also models the change in illuminance (ΔE)as a function of the depth map (D), and variations in LED intensities (ΔI). The dynamic relation is captured in the eqautions:

$$E_{k+1} = E_k + f(D_k, \Delta I_k) \quad \mathit{and} \quad \Delta E_{k+1} = f(D_k, \Delta I_k) \tag{4.2}$$

These equations highlight the system's capability to adapt to real-time changes in surface orientation and ambient lighting conditions.

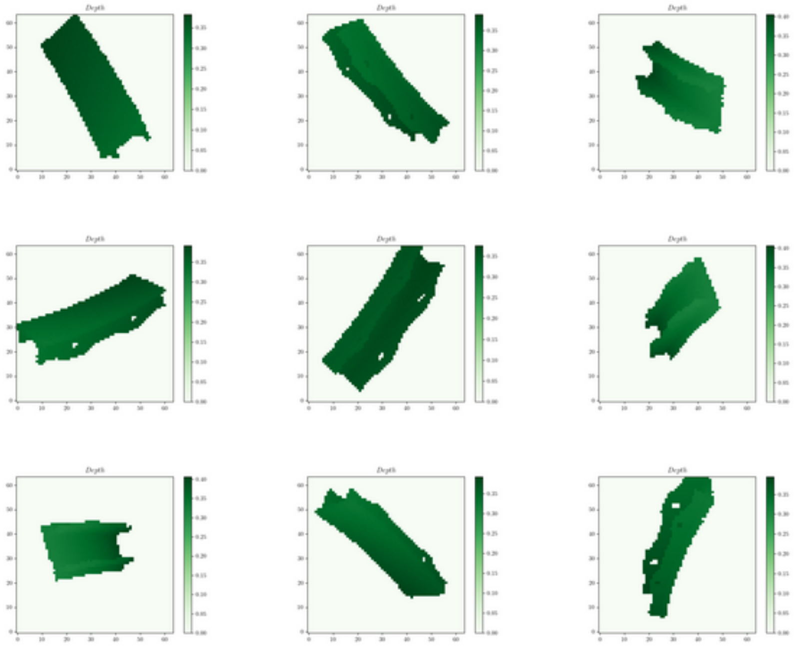


Figure 4.2: Subset of data used to train the VAE. Depth and normal maps allow for spatial input to illuminance distribution model. VAE provides reduction in dimensionality.

Recognizing these gaps, our research aims to extend this modeling approach to more complex scenarios. We propose the development of a system capable of adapting to complex textures and surface geometries, ensuring even illuminance distribution in diverse conditions. The goal of this advancement lies in integrating Model Predictive Control (MPC) with a dynamic LED panel system. MPC will enable the system to make informed, predictive adjustments to the LED panels, adapting to real-time changes in surface characteristics and ambient lighting.

To achieve this, a foundational step involves training a model capable of understanding and predicting the nuances of complex surface illuminance. We employ a Variational Autoencoder-Deep Convolutional Generative Adversarial Network (VAE-DCGAN) model for this purpose. Initially, the VAE-DCGAN model is trained using data generated in Blender. This training process involves capturing and analyzing images under various lighting conditions, enabling the model to learn and predict the effects of different LED intensity adjustments on complex surfaces.

4.2 Modeling of Illuminance Distribution Changes on Complex Surfaces via VAE-DCGAN

The process of training the model involved the collection of diverse data sets using Blender, a 3D rendering tool. This included capturing images before and after altering light intensities, alongside obtaining depth and normal maps from the camera. These maps provided crucial information about the scene's geometry and surface properties. Depth maps helped in understanding the spatial layout, while normal maps offered insights into surface orientation. This comprehensive data collection was essential for the model to learn the relationship between changes in LED intensity, depth, normals, and the resulting illuminance delta maps, thus ensuring the training of an effective and robust model.

4.2.1 Variational Autoencoder (VAE) Architecture and Result

In this implementation, a Variational Autoencoder (VAE) is employed for image processing tasks. The architecture of the VAE is composed of two primary components: an encoder and a decoder. The encoder's role is to compress the input data which is 64x64x1

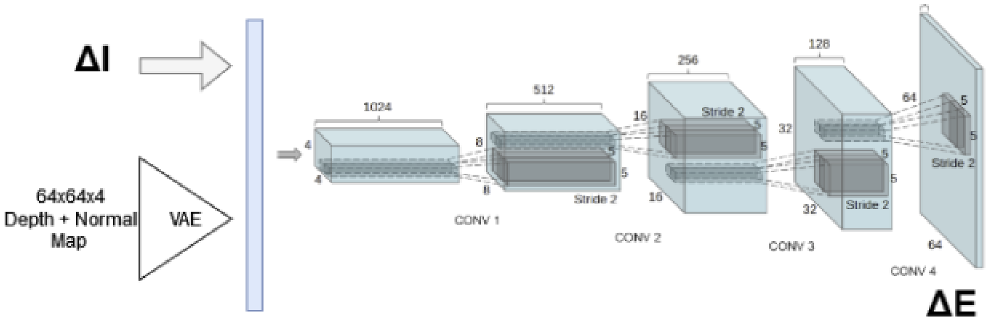


Figure 4.3: ANN Consisting of VAE-DCGAN Model Architecture. [7]

depth map into a lower dimensional latent space representation, while the decoder reconstructs the data from this latent space.

The encoder is constructed using a series of convolutional layers, each followed by batch normalization and LeakyReLU activation functions. Dropout is also included to prevent overfitting. This structure effectively compresses the input image into a lower-dimensional latent space. The latent representation is captured in two forms: a mean (μ) and a variance (log_var), which together define a probability distribution over the latent space.

The decoder part of the VAE works in reverse. Starting from the latent space representation, it utilizes a series of transposed convolutional layers, again followed by batch normalization and LeakyReLU activations, to upscale the compressed representation back to the original image dimensions. The final layer of the decoder ensures the output has the same number of channels as the input images, using a Tanh activation function.

The reparameterization trick is a crucial aspect of the VAE. It allows the model to backpropagate through random sampling by introducing a random noise component (ϵ) combined with the latent space distribution parameters (μ and log_var). This step is essential for training the VAE. For the loss function, the VAE employs a combination of reconstruction loss (using mean squared error) and the Kullback-Leibler divergence (KLD). The reconstruction loss ensures the decoded images are as close as possible to the original inputs, while the KLD loss regularizes the latent space by encouraging the distribution to approximate a standard normal distribution. Hyperparameter tuning is a critical part of training the VAE. This process involves experimenting with different learning rates, latent dims, and batch sizes to find the optimal combination that minimizes the validation loss. The best hyperparameters are then used for the final training of the model.

In practice, this VAE model is utilized for processing image data, particularly for tasks like image reconstruction and generating new images from learned latent representations. The implementation includes functions to train the model, validate its performance, and visualize reconstructed images alongside their original counterparts. The use of a custom dataset class allows for flexible data handling, ensuring the model can be adapted to various types of image data.

Training experiments were run for the VAE with multiple sets of hyperparameters and amounts of data.

First, a set of 10,000 depth map datapoints were collected, with the part under inspection being rotated randomly around the x, y, and z axes for each image. This resulted in a high diversity of data. Hyperparameter tuning was performed with this batch, but the results were suboptimal. The train loss was far under the val loss for all combinations of hyperparameters, and the reconstructed images, while visibly approaching the ground truth, were not yet well reconstructed.

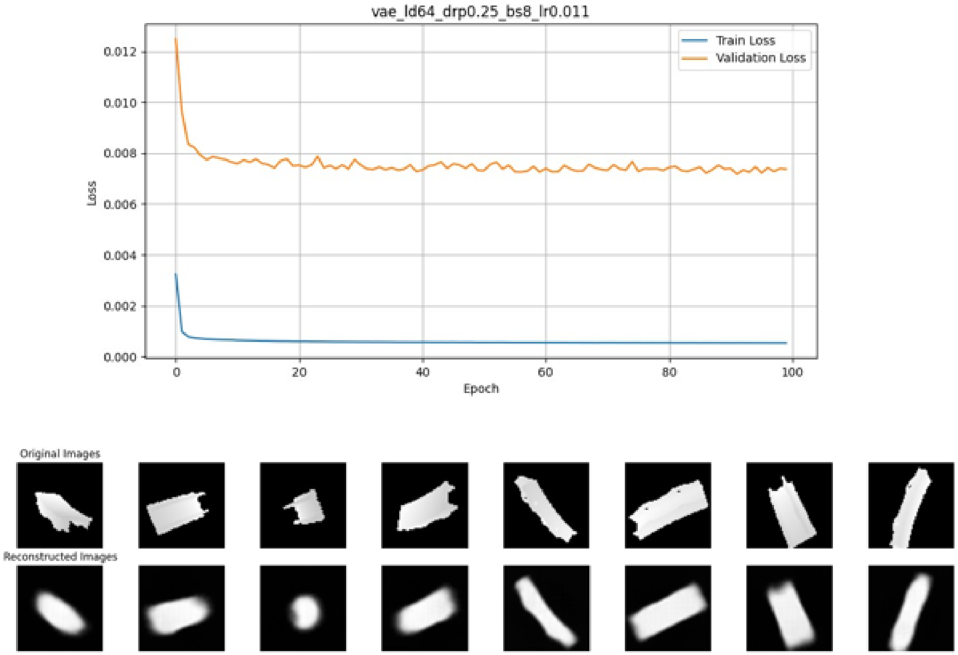


Figure 4.4: VAE 10k datapoints training loss curve and original (top) vs. reconstruction (bottom). Latent dimension = 64, Dropout = 0.25, Batchsize = 8, Learning rate = 0.011.

Next a batch of 100,000 depth map data points were collected with the part rotated solely around the z-axis. Furthermore, a learning rate scheduler was added to assist convergence to a minimal value. Cosine annealling was selected due to recommendations and its common use in training neural networks.

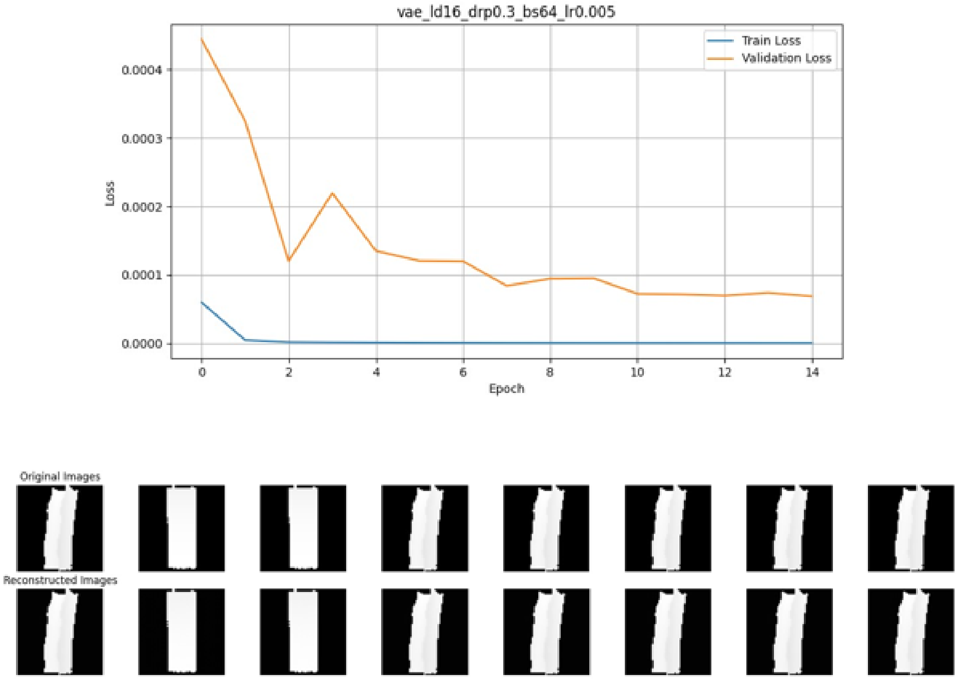


Figure 4.5: VAE 100k datapoints training loss curve original (top) vs. reconstruction (bottom). Latent dimension = 16, Dropout = 0.3, Batchsize = 64, Cosine Annealing Scheduler.

The successful hyperparameter tuning of the VAE led to a significant improvement in model performance, as evidenced by a marked reduction in validation loss. The validation loss reaching a value under 0.0001 indicates a high level of accuracy in the VAE's ability to reconstruct images. This level of accuracy suggests that the VAE is capable of producing reconstructions that are almost identical to the original images, demonstrating its effectiveness in processing and reconstructing depth map data. This achievement underscores the model's precision and reliability in handling complex image processing tasks.

4.2.2 Deep Convolutional Generative Adversarial Network(DCGAN) Architecture and Result

The model incorporates a Deep Convolutional Generative Adversarial Network (DCGAN) that works in tandem with a Variational Autoencoder (VAE) [15]. The DCGAN utilizes encoded image data from the VAE, represented in a 64-dimensional latent space, combined with changes in 384 LED intensity levels. This setup is crucial for generating high-quality, realistic illuminance delta maps, essential for the analysis.

The architecture of the DCGAN comprises two main components: a generator and a discriminator. The generator network begins by merging the latent vectors from the VAE with the LED intensity data. It then upscales this combined input through several transposed convolutional layers, each equipped with batch normalization and Rectified Linear Unit (ReLU) activation functions. This design transforms the compact latent vector into a larger-dimensional space, achieving the desired output image size. The generator's final layer uses a Hyperbolic Tangent (Tanh) activation function, aligning the output with the pixel value range of the input images.

Conversely, the discriminator is a Convolutional Neural Network (CNN) tasked with distinguishing between actual and synthetically generated illuminance delta maps. It classifies these maps through exposure to both real and generated examples. The discriminator's design is crucial in providing a learning signal to the generator, consisting of multiple convolutional layers followed by Leaky Rectified Linear Unit (LeakyReLU) activations. These layers downscale the input image and extract pertinent features for accurate classification, concluding with a sigmoid activation layer that outputs a probability score.

The training of the DCGAN alternates between the discriminator and the generator. The discriminator is trained with a mix of authentic and generated images, aiming for accurate classification. In contrast, the generator is refined to produce images perceived as real by the discriminator. This adversarial training continues until the generator consistently creates convincingly realistic images. Both networks use the Binary Cross-Entropy (BCE) loss function, standard for binary classification tasks.

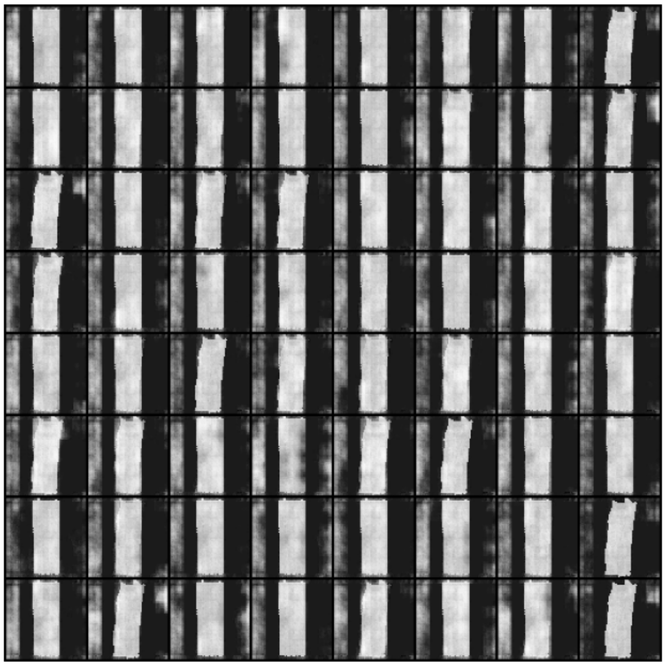


Figure 4.6: DCGAN Result with 5 Epoches with Discriminator Learning Rate = 1e-5, Generator Learning Rate = 1e-2.

In the model's training, significant fluctuations in the losses of both the generator and the discriminator are observed, typical of DCGAN training. This reflects the ongoing adaptation and competition between the networks. The discriminator's performance, measured by D(x), varies across iterations. At times, it demonstrates high accuracy, while at others, its performance decreases. Similarly, the generator loss fluctuates, indicating varying success in deceiving the discriminator.

The value of D(G(z)), reflecting the discriminator's predictions on fake images, varies widely. Ideally, this should average around 0.5, but the observed variation suggests an unstable training process. Instances of very low D(G(z)) values might indicate 'mode collapse', a significant concern in DCGAN training.

Despite these challenges, the quality of the images generated by the DCGAN is commendably high, indicating successful feature learning by the generator, despite the volatile training metrics.

To improve the stability of the training process and enhance results, exploring adjustments in hyperparameters, different architectures, or advanced GAN training techniques like gradient penalty or alternative loss functions is recommended. GAN training requires careful tuning and experimentation for optimal performance.

With the insights gained from the training dynamics and the quality of the generated images, the model is positioned to be implemented in real-world scenarios. This includes using the model to control 384 addressable LED lights, aiming to achieve a desired illuminance distribution. Through careful application and refinement, the model shows significant potential for practical use in diverse lighting environments.

Chapter 5

AUTOMATIC COLOR CORRECTION USING CAMERA AND LED ARRAY

5.1 Problem Statement and Need

The consistency and uniformity of the color scheme in image datasets are critical for employing machine learning algorithms. A prevalent issue identified during the training of defect detection models is the influence of varying color schemes on the accuracy of defect identification. This challenge becomes particularly pronounced when grayscale processing is applied to images for defect analysis.

The conversion of images to grayscale can inadvertently introduce artifacts or variations that the defect detection algorithm misinterprets as defects. This issue is illustrated in Figure 1, where images from a shroud show apparent defects in areas with color inconsistencies. Although the majority of the part appears in a gray color scheme, areas that exhibit a bluish tint are erroneously flagged as defective. This misclassification stems from the grayscale processing failing to account for the inherent color variations in the original image.

Such inconsistencies in color scheme not only lead to false positives but also undermine the overall reliability of the defect detection process. To mitigate this issue, a robust method for color scheme disturbance rejection becomes essential [3]. The objective is to identify outlier images – those with color schemes significantly deviating from the norm – and adjust the imaging setup accordingly.

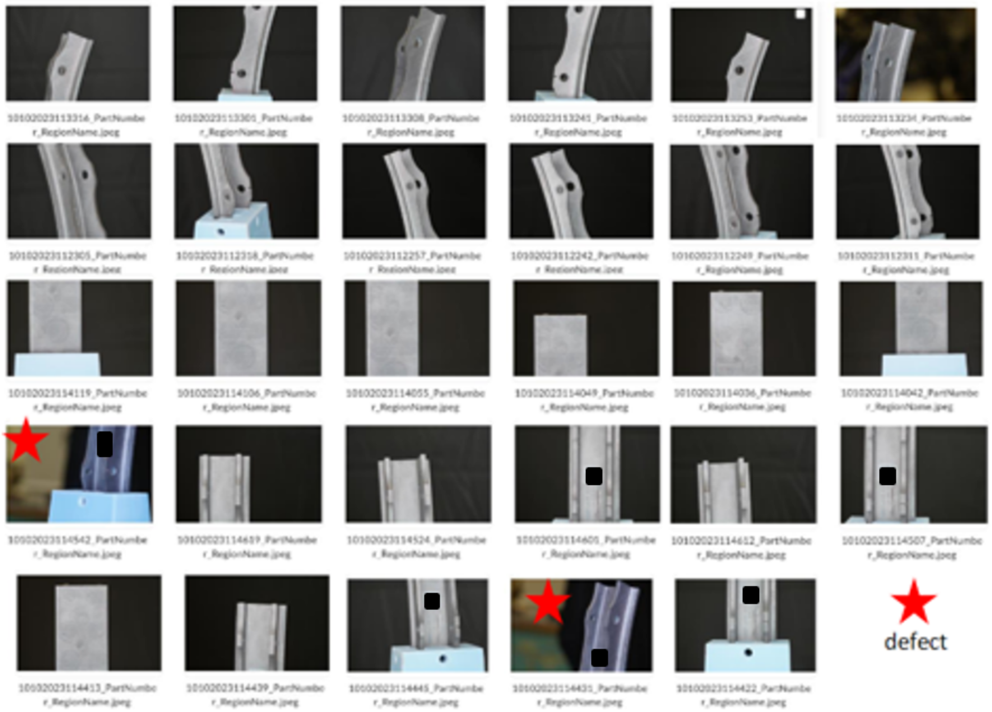


Figure 5.1: Machine Learning Data Sample Set (Stared Image Has Defects)

The proposed solution to address the issue of color variations in defect detection involves a two-pronged approach:

1. *Outlier Detection:*

    - Implement a system to identify outlier images: This system analyzes the color distribution of each image and compares it with a predefined baseline or range of acceptable color schemes. Images deviating from this standard are flagged as outliers.

    - Review and reprocess outlier images: Once identified, outlier images undergo further review or reprocessing to ensure only consistent and uniform images are used for defect detection.

2. *Camera and LED Light Setting Adjustments:*

    - Develop a mechanism for adaptive adjustments: Develop a system to dynamically adjust camera settings and LED light colors to counteract and normalize color scheme variations. This involves adjusting lighting conditions or camera parameters to bring the color scheme of each captured image closer to the desired standard.

    - Enhance consistency and reduce false positives: These adjustments ensure subsequent images adhere to a consistent color scheme, minimizing the likelihood of false positives due to color variations. This improves the overall accuracy and reliability of the defect detection process.

Addressing the challenge of color scheme inconsistency is crucial for enhancing the accuracy and reliability of machine learning-based defect detection systems. By ensuring that all images fed into the algorithm maintain a robust and uniform color scheme, the system can more effectively distinguish between true defects and benign variations in the imaging process. This improvement in accuracy is vital, especially in critical applications such as aerospace component inspections, where the cost of false positives can be significant.

5.2 Initial Approach and Possible Solution for Automatic Color Correction

In machine learning-based defect detection, the consistency and accuracy of color representation in images are critical factors. The initial approach involved using Robust Principal Component Analysis (RPCA) to manage and analyze color variations within image datasets. RPCA is adept at separating sparse anomalies from low-rank data structures, making it a potentially suitable method for identifying images with non-standard color schemes.

The process started with standardizing image dimensions through resizing, a necessary step for streamlining color analysis. RPCA was then applied to segregate the primary color features (low-rank components) from color anomalies (sparse components). The intent was to objectively identify images with unusual color schemes by isolating these components. Metrics were developed to quantify color differences, providing a measurable way to assess the extent of deviation from the expected color range.

Upon identifying images with distinct color schemes, the system was designed to implement corrective measures. Adjusting camera settings to match these outlier images with the majority ensured a uniform color scheme across the dataset. Parallelization in processing was integrated for efficiency and timeliness.

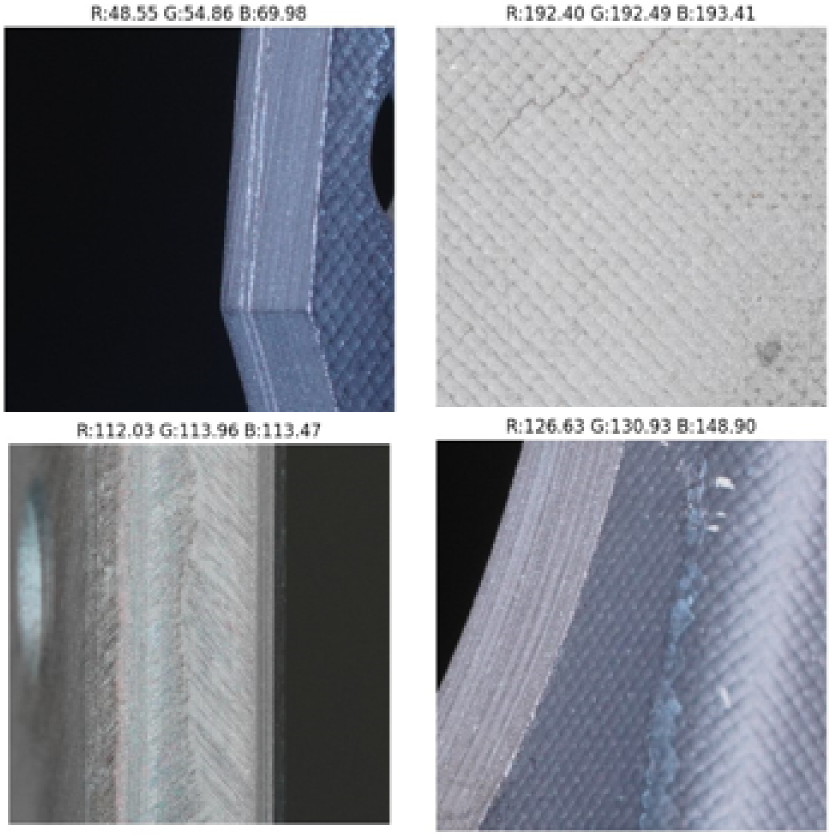
However, in practical application to sample data, a limitation emerged: the sparse ranks in different images showed very similar mean values, reducing RPCA's effectiveness in differentiating normal and outlier color schemes. This led to a pivot in approach. It was determined that while RPCA could identify the overall color scheme using lower rank values, it was not as effective in pinpointing outliers as initially hoped.

The strategy was revised to focus on direct analysis of the color channels in each image. By examining the Red, Green, and Blue (RGB) values, outliers could be identified more precisely. This method involved analyzing the RGB values of each image and comparing them to a standard range. Images with color values significantly diverging from this standard were flagged as outliers.

This revised method allowed for more accurate detection of images with color inconsistencies and more reliable identification of outliers. Necessary adjustments in camera settings and LED lighting were then made to align the color scheme of these images with the rest of the dataset. This adjustment ensured that all images used for defect detection maintained a consistent and uniform color scheme, reducing the likelihood of false positives due to color variations.

5.3 Apply Image Segmentation on Object and Check the RGB Color Value

In refining the color analysis process for robotic inspection systems, a shift was made from analyzing the entire image's RGB color values to a more targeted approach that involves image masking [16]. This change was driven by the need to overcome the limitations posed by background noise in the image data. When analyzing images for specific tasks like defect detection or quality control, it's crucial that the mean color value calculations accurately represent the object of interest, not the entire scene including the background. The initial method, which involved direct RGB color analysis of the entire image, presented a significant challenge. The background, with its diverse colors and patterns, introduced extraneous information—noise—that distorted the mean color values of the object. This noise created a misrepresentation of the actual color characteristics of the object, leading to inaccuracies in any subsequent analysis or decision-making processes.



R:48.55 G:54.86 B:69.98

R:192.40 G:192.49 B:193.41

R:112.03 G:113.96 B:113.47

R:126.63 G:130.93 B:148.90

Figure 5.2: RGB Channel Without Applying Mask

As observed in the analysis, the RGB channels of the images without masks display inconsistent color values. This inconsistency, evident even within the same parts of an image, presents a significant challenge in determining which images are outliers based on deviations in RGB channel colors. To address this issue, a method involving edge detection, specifically Canny edge detection, is employed to assist in achieving color balance.

However, the direct application of Canny edge detection on the images is not straightforward due to their high sharpness. This sharpness can produce a high density of edge information, leading to an overload of details and making it difficult to discern the primary contours of the objects.

To mitigate this challenge, a blurring technique is first applied to the images. Blurring reduces image noise and smoothens high-frequency components, thereby simplifying the image structure. This simplification aids the Canny algorithm in detecting the essential edges by diminishing the impact of minor variations and irrelevant details. Once blurred, the Canny detection process can more effectively and accurately identify the outlines of the objects.

Following the edge detection, the largest contour in each image, presumed to be the object of interest, is identified. This contour is then used to create a mask, isolating the object from the rest of the image. Applying this mask enables the calculation of the average color within the contour, providing a quantitative measure for further analysis.

This combined approach of blurring, followed by Canny edge detection and contour analysis, establishes a more consistent baseline for comparing images. It enhances the ability to identify outliers based on discrepancies in the RGB channels, a crucial step in ensuring the reliability and accuracy of image analysis in aerospace component inspections. By focusing on the average color within the identified contours, a more accurate assessment of the image quality and consistency is achieved, playing a pivotal role in the inspection process.

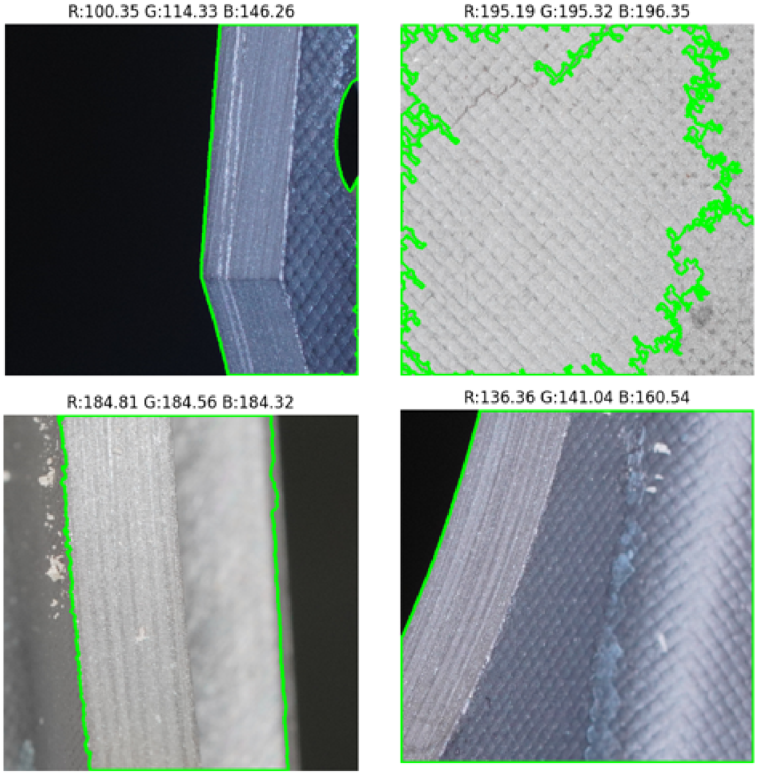


Figure 5.3: RGB Channel With Applying Mask

In the enhanced imaging system, a mask is applied to the view, enabling users to discern the color balance of the part with greater clarity. A significant feature of this system is the inclusion of a histogram in the right upper corner, displaying the distribution of RGB color channels. This visualization is crucial as it aids users in understanding the color characteristics of the object being inspected. Furthermore, the system allows users to adjust the RGB settings of the LED lights through an rqt interface. This capability facilitates the fine-tuning of the lighting environment, enabling users to achieve a color balance that closely matches the original color of the part.

Looking towards future advancements, there are plans to incorporate a color calibration feature into the system. This feature would involve the camera system automatically analyzing the object's color and accordingly adjusting the camera settings and LED light colors to closely match the object's color. Additionally, these color settings would be stored, allowing for the establishment of threshold values. These thresholds would play a critical role in quality control; if an image's color metrics significantly deviate from these established thresholds, the system would trigger a repositioning of the camera (re-pose) and retake the image. This automatic recalibration and re-imaging process is a significant step forward in ensuring consistent image quality and color accuracy.

Such advancements are essential in the context of high-precision aerospace component inspections. They offer a solution to the challenge of maintaining consistent color balance and image quality in varying lighting conditions and across different parts, thereby enhancing the reliability and accuracy of the inspection process. This approach, once fully implemented, will significantly streamline the inspection process, reducing the need for manual adjustments and increasing overall inspection efficiency.

Figure 5.4: Default Color Setting Image vs. Adjust LED Light Color Balance

Chapter 6

DATA PRE-PROCESSING FOR ROBUST DEFECT DETECTION AI MODEL

6.1 Denoise Selected High Frequency Features with Texture Filtering

The role of texture filtering in machine learning, especially in image processing, is crucial. High-magnification images are prone to generating false positives when they contain high-frequency random features. These patterns, while inherent to the image, can be misinterpreted as anomalies, leading to inaccurate outcomes in automated systems.

To mitigate this issue, the study employs smoothing techniques, with a focus on post-processing methods. The primary technique used is the 2D Fast Fourier Transform (FFT), a powerful tool in the realm of digital image processing. The 2D FFT denoising technique aims to remove noise from an image by utilizing the two-dimensional fast Fourier transform algorithm. Its primary goal is to eliminate the high-frequency components in the FFT spectrum that are typically indicative of noise, while retaining the low-frequency components that reflect the essential structure and features of the image. This is achieved by applying a low-pass filter to the FFT spectrum, which effectively attenuates high-frequency components and preserves the low-frequency ones. The filtered FFT spectrum is then converted back to the spatial domain using the inverse FFT algorithm, resulting in a denoised image where the unwanted patterns are significantly reduced or removed [2].

In the specific case of the source image containing a hatch pattern, the initial step involves converting the image to grayscale and then applying the 2D FFT, with the zero-frequency component shifted to the center of the spectrum. This shift is crucial for a more intuitive analysis and filtering process. The next step involves computing the magnitude spectrum of the FFT, which reveals various spatial frequencies present in the image, including horizontal, vertical, and diagonal frequencies. These frequencies correspond to the different textures and details within the image, with high frequencies indicating fine details and low frequencies representing the overall structure.

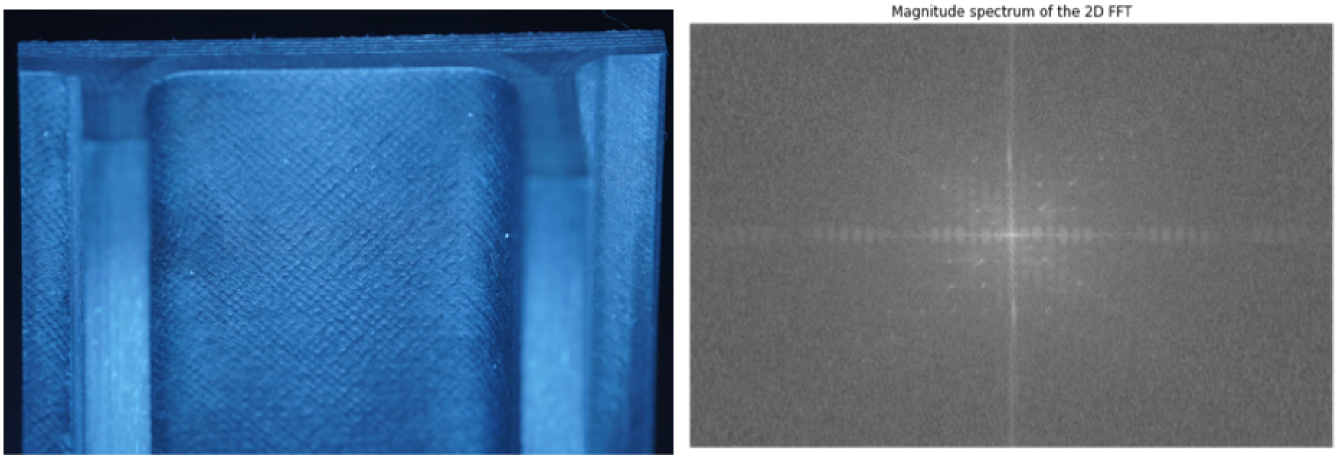


Figure 6.1: Original Image and 2D FFT Magnitude Spectrum of the Image

The critical task then becomes identifying which frequencies are associated with the hatch pattern. To achieve this, a mask is created specifically for the hatch pattern, and 2D FFT is applied to it. This process reveals that the diagonal frequencies are the ones related to the hatch pattern. With this understanding, a mask is applied to the magnitude spectrum, filtering out the frequencies associated with the cross-hatching pattern. The result is a filtered magnitude spectrum and a denoised image where the hatch pattern has been effectively removed, preserving all other details.

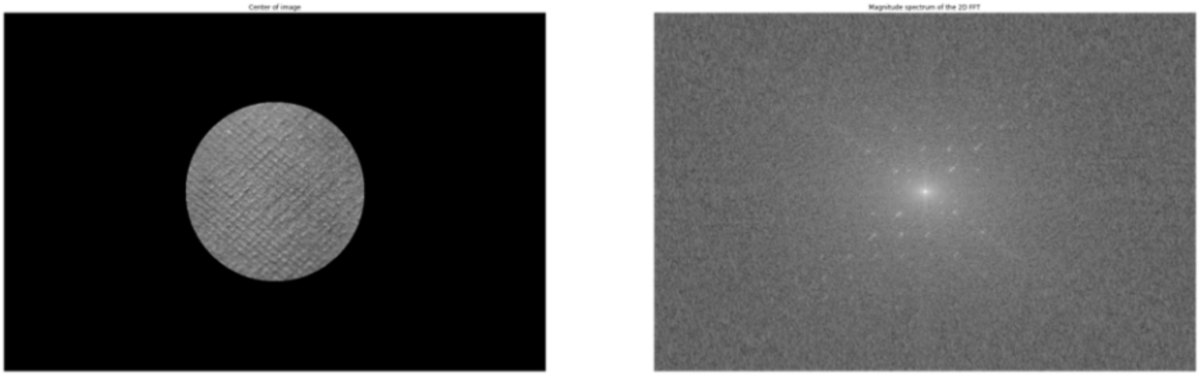


Figure 6.2: Masked on Hatch Pattern and 2D FFT Magnitude Spectrum of the Image

Additionally, the inverse of this process is also demonstrated, where only the frequencies corresponding to the hatch pattern are retained. This results in a filtered image containing exclusively the hatch pattern, devoid of other details. This inverse approach further underscores the effectiveness of the FFT technique in isolating specific texture components within an image.

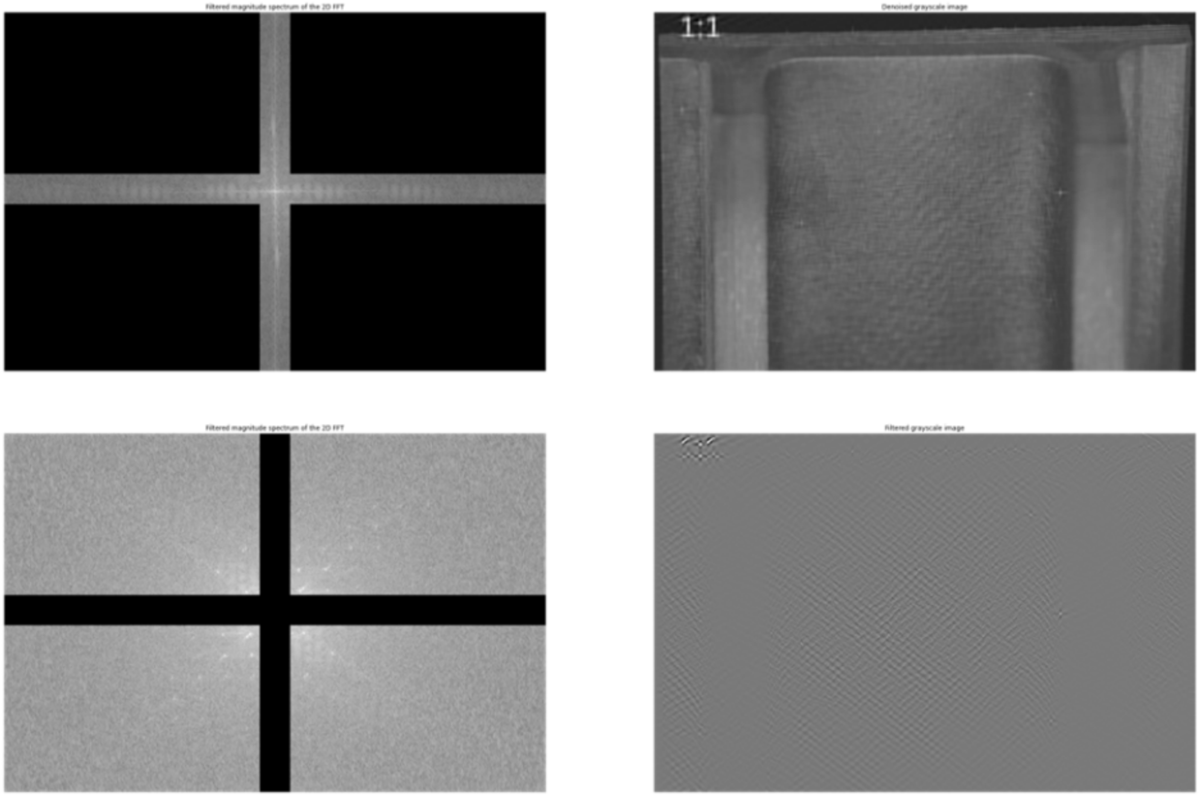


Figure 6.3: 1. Applied 2D FFT Denoise Hatch Pattern on Image, 2. Inverse Denoise Image(Only whows Hatch Pattern)

The utilization of 2D FFT for texture filtering in this study not only may enhances the accuracy of image interpretation by reducing false positives but also showcases the potential for advanced image processing techniques in applications requiring detailed texture analysis.

In the exploration of texture filtering and pattern extraction methods critical to computer vision, one standout technique is the Gabor filter. Renowned for its efficacy in image processing, the Gabor filter is a linear tool designed to extract texture information and detect edges within images. Its construction involves a set of complex sinusoidal waveforms enveloped by a Gaussian function. This unique composition allows the Gabor filter to simultaneously capture both frequency and orientation information from images, making it a versatile and powerful tool in the analysis of textural features [12].

The Gabor filter's ability to analyze spatial frequencies and orientations makes it particularly effective in scenarios where texture and edge detection are paramount. It operates by tuning to specific frequencies and orientations, allowing for detailed and localized analysis of image regions. This attribute is particularly useful in applications such as pattern recognition, image segmentation, and even in biometric security systems like facial recognition, where the detection of specific textural patterns is crucial.

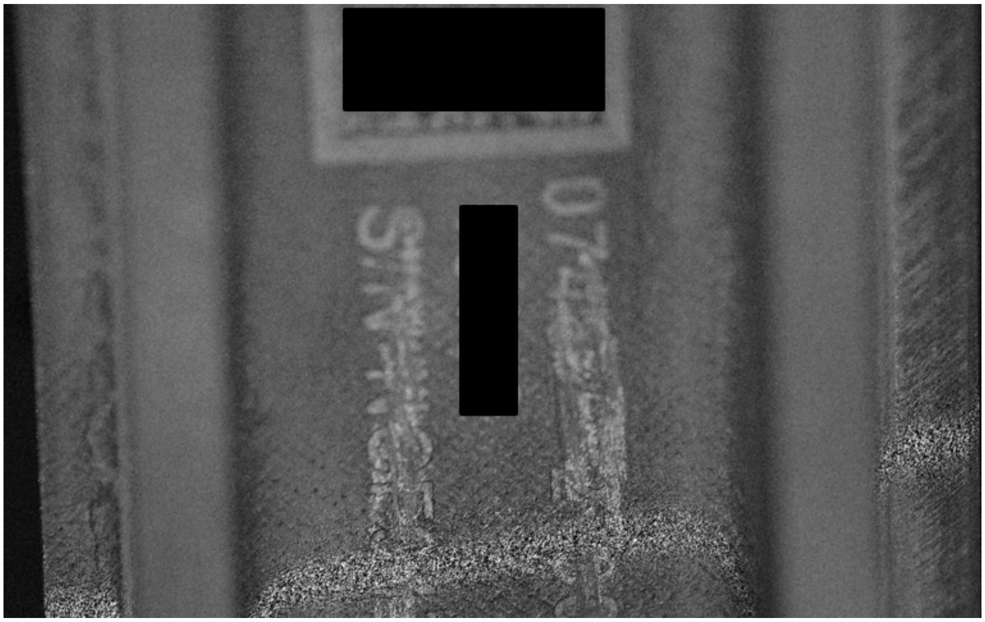

[redacted]
[redacted]

Figure 6.4: Apply Gabor Filter to Denoise the Pattern but Saving the Other Detail(Scratch, Number, etc)

In this study, the results of applying the Gabor filter showcase its proficiency in noise reduction. The filter was employed to target and mitigate the impact of the hatch pattern prevalent in the images. The effectiveness of the Gabor filter is evident in the clear visual quality of the processed image, where most of the hatch pattern has been successfully removed. However, it is observed that some residual noise remains in the lower part of the image. Despite this, considering the specific requirements of the inspection process, which focuses on a small central area of the image, this residual noise does not pose a significant concern.

Moreover, the potential for further optimizing the filter's performance exists. By adjusting the filter's hyperparameters, such as the wavelength, orientation, and bandwidth of the sinusoidal component, or the size of the Gaussian envelope, it may be possible to achieve even more refined results. Fine-tuning these hyperparameters can allow for more precise targeting of specific textures and patterns, potentially reducing noise levels further and enhancing the overall clarity of the image.

The application of the Gabor filter in this context not only highlights its effectiveness in extracting and filtering textural information from high-resolution images but also underscores its adaptability and utility in diverse computer vision tasks. The ability to adjust its parameters to suit specific image characteristics or requirements makes the Gabor filter a versatile and indispensable tool in advanced image processing and analysis.

6.2 Robust Viewpoint Generation using PCA on Surface Normals

Dealing with poor viewpoint segmentation in 3D point cloud data poses a significant challenge, especially in complex geometrical areas such as those with holes and edges. This issue is predominantly seen when utilizing k-means clustering on the normals of 3D point cloud data, where the normals in intricate areas differ markedly from those in more uniform regions. Such discrepancies lead to inaccuracies in the segmentation process, adversely impacting the overall quality of 3D modeling and analysis.

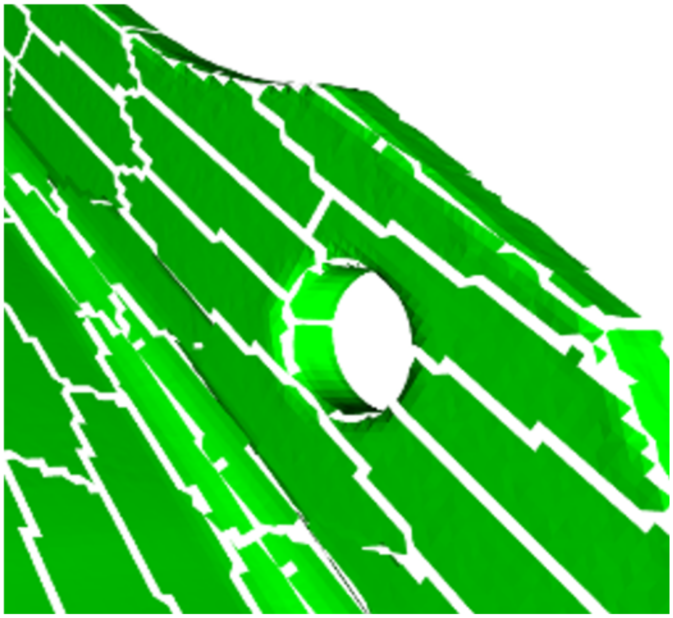


Figure 6.5: Viewpoint Segmentation Including Holes and Edges of the Part

One of the primary consequences of these inaccuracies is the formation of incorrect shapes in the segmented models. This flaw can have practical implications for an inspection system, potentially causing it to miss crucial surfaces of the part under examination. If the segmentation inaccurately represents a part's surface, important areas might be overlooked during the inspection, leading to incomplete or flawed analysis.

Moreover, these segmentation errors can result in the unnecessary inclusion of certain areas, such as minor holes or irrelevant features, in the model. These superfluous elements introduce noise into the data and can misdirect the inspection process towards areas that do not require scrutiny.

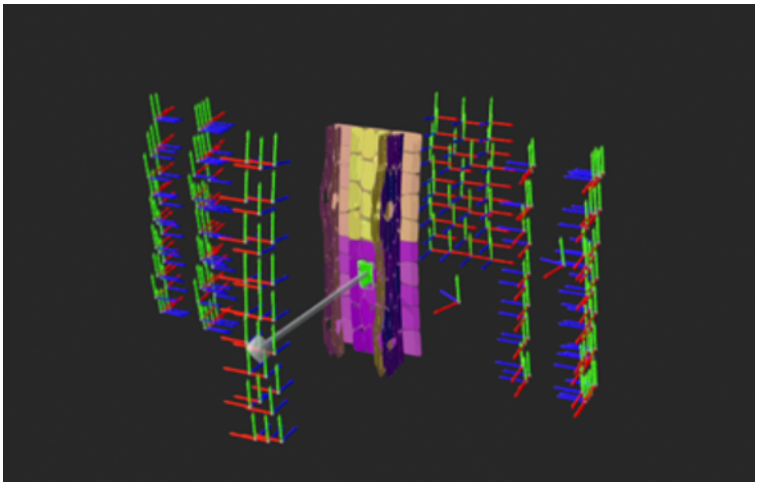


Figure 6.6: Viewpoint Segmentation without Holes and Edges of the Part

Another significant impact of poor viewpoint segmentation is its influence on the optimization of viewpoints needed for a thorough inspection. Inaccurate segmentation can inadvertently increase the number of required viewpoints, thereby extending the overall inspection duration. This inefficiency becomes a critical issue in environments where the speed of inspection is paramount. An ideally optimized number of viewpoints ensures a complete inspection in the shortest possible time. However, with suboptimal segmentation, the system may generate an excessive number of viewpoints, diminishing the efficiency of the entire inspection process.

In this approach, Principal Component Analysis (PCA) is utilized to identify outliers in 3D point cloud data [11], a critical step in addressing the challenges of viewpoint segmentation in areas with complex geometries. The process begins by importing necessary libraries for numerical operations, data plotting, PCA implementation, and 3D point cloud handling. The 3D point cloud data is then loaded from a PLY file, which includes 3D data points along with their associated normals.

The first step in the analysis involves normalizing the normals of the point cloud using Open3D, ensuring uniformity in length across the dataset. This normalization is vital for consistent processing. The normals are then converted into a NumPy array, and PCA is applied. By using PCA, the data is transformed into a new coordinate system that aligns with the direction of maximum variance, effectively reducing the dimensionality while preserving significant variability.

PCA transforms the normals into a new coordinate system aligned with the directions of maximum variance. Following this transformation, the concept of z-scores comes into play. A z-score is a statistical measure that describes the position of a value relative to the mean of a group of values, expressed in terms of standard deviations from this mean. The calculation of a z-score for a value x is done using the formula:

$$z = \frac{(x - \mu)}{\delta} \tag{6.1}$$

Where μ is the mean and δ is the standard deviation of the group. Subsequently, Z-scores are calculated for the PCA-transformed data. These scores standardize the data points based on their deviation from the mean, measured in standard deviations. This calculation is key to outlier detection, as points with high z-scores significantly deviate from the dataset's mean. The distribution of these z-scores across each principal component is visualized using histograms, providing insights into the data's spread and aiding in the identification of outliers.

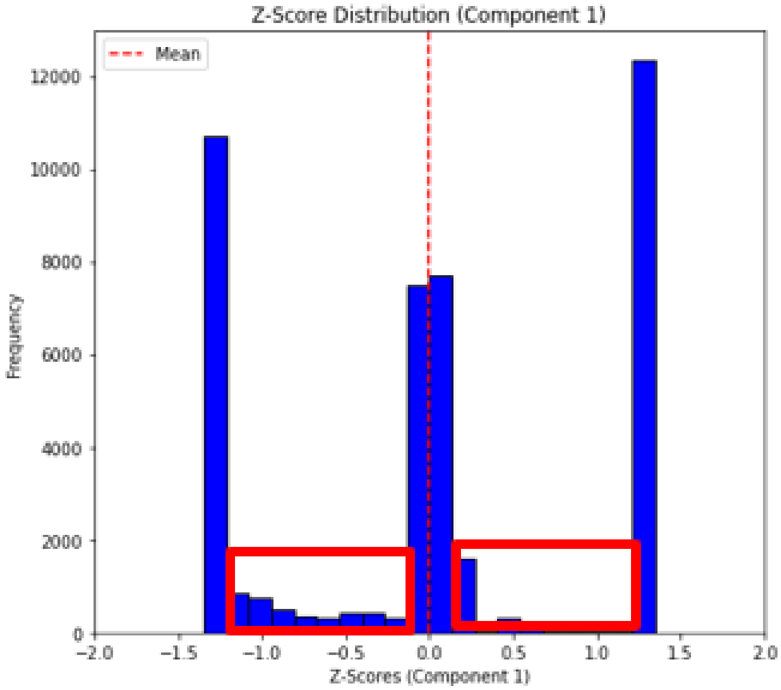


Figure 6.7: Z-Score Distribution with Inliers and Outliers(Red Boxs) Data Points

The next phase involves defining thresholds to distinguish outliers based on these z-scores. Points falling outside these defined thresholds are marked as outliers, indicative of significant deviations possibly due to complex geometrical features like holes and edges. The script then separates these outliers from the inliers for focused analysis.

Finally, both the inliers and outliers are visualized separately using Open3D's visualization tools. This visualization is an essential part of the process as it allows for an assessment of the outlier detection's effectiveness. Through the application of PCA on the point cloud data normals, this method effectively isolates areas with unusual geometries, thereby enhancing the accuracy of 3D modeling. This technique proves particularly beneficial in applications requiring precise geometric representations, such as quality inspection systems in manufacturing or detailed spatial analysis in 3D modeling. The ability to accurately identify and handle outliers in 3D point cloud data is a significant advancement in improving the reliability and precision of 3D point cloud analysis.

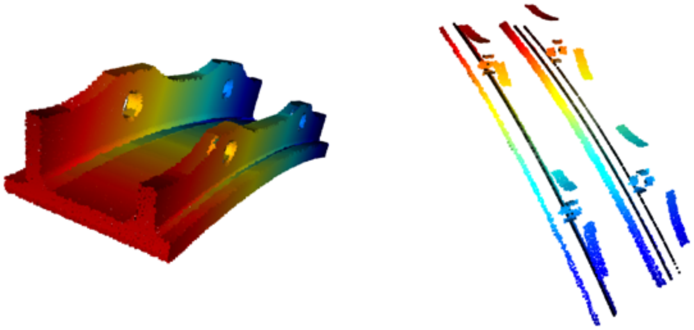


Figure 6.8: Inlier Point Cloud Data Reconstructed in 3D (Left), Outlier Point Cloud Data Reconstructed in 3D (Right)

Chapter 7

POST PROCESS DISTURBANCE REJECTION SYSTEM AND RESULT

The implementation of a post-disturbance check system represents a crucial step towards ensuring the integrity and quality of image data in robotic inspection processes. This system is designed to address the unpredictable errors that may arise during the image acquisition phase, which can adversely affect the data quality.

The core functionality of the post-disturbance check revolves around evaluating the focus value of each image, particularly within a central region measuring 1024×1024 pixels. This specific area is chosen as it generally represents the most critical section of the image, especially in scenarios involving complex surface shapes where focus may vary across the image. By concentrating on this central region, the system effectively assesses the overall image quality with a greater emphasis on the most relevant portion.

If the focus value of an image falls below a predetermined threshold, indicating a potential quality issue, the system automatically triggers a re-acquisition process for that specific pose. This re-queuing mechanism is integral to the system's efficiency, as it ensures that only those poses with subpar image quality are revisited, rather than redoing the entire inspection sequence. This selective approach significantly reduces time and resource expenditure, streamlining the inspection process.

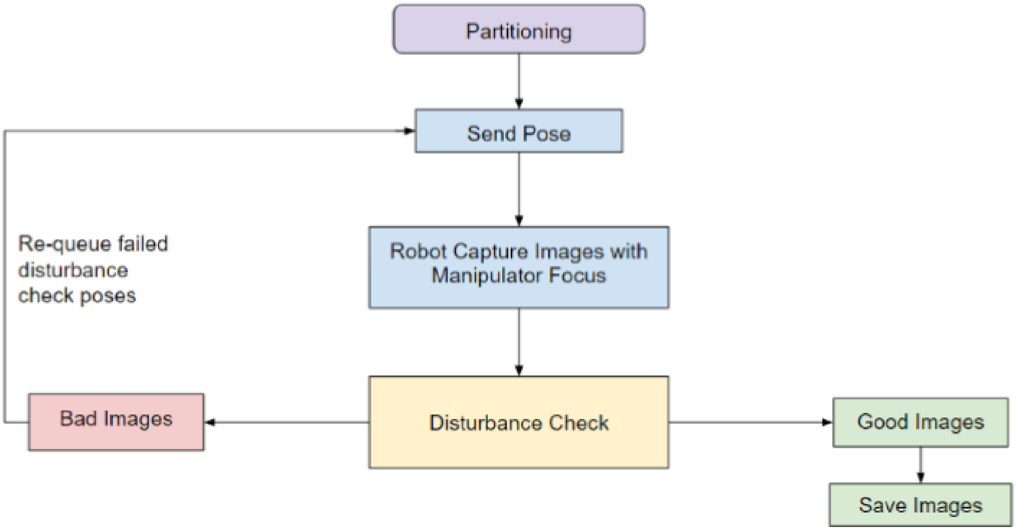


Figure 7.1: Post Process Disturance Rejection System Flow Chart

The significance of the post-disturbance check extends beyond just maintaining image quality. It plays a vital role in preserving the integrity of data fed into machine learning algorithms for defect detection. By ensuring that only high-quality images are used for analysis, the system enhances the accuracy and reliability of the AI models' inspection results. This is particularly crucial in applications like aerospace component inspections, where precision is paramount, and the margin for error is minimal.

Moreover, the system's ability to adaptively respond to unpredictable errors and autonomously rectify them minimizes the need for operator intervention. This automation not only increases efficiency but also reduces the likelihood of human error, further bolstering the reliability of the inspection process.

In this comprehensive study, five trials were conducted to capture images of a shroud using a manipulator, each beginning with different initial offsets ranging from 40 to 52 cm. The shroud was systematically imaged from four distinct sides: front, back, left, and right. The allocation of viewpoints for these sides was carefully planned: 16 for the front, 29 for the

Face	**Avg Total Time (Secs)**	**Avg Focus Value**	**Total # of Failed Disturbance**
Front	122.34	1424.34	1
Back	193.26	476.42	2
Left	144.42	983.32	0
Right	131.53	876.13	1

Table 7.1: 5 Trials Image Capturing the Shroud with Manipulator Focus with random Initial Offset

back, 20 for the right, and 23 for the left, totaling 88 viewpoints. This strategic distribution of viewpoints ensured an all-encompassing inspection of the shroud from every angle. During the image capture process, each pose, on average, was completed in approximately 6.7 seconds. The total duration of inspection for each trial was observed to be between 10 to 12 minutes. This efficient use of time, coupled with the comprehensive viewpoint coverage, highlights the effectiveness of the inspection protocol employed.

A crucial part of the analysis involved calculating focus values for each side of the shroud to quantitatively assess the image quality. The average focus values obtained were as follows: the front side registered an average of 1424.34, the back 476.42, the left 983.32, and the right 876.13. The notably lower focus value for the back side was initially intriguing. Upon closer examination, it was attributed to the smoother texture of the shroud's back compared to its other sides. It is essential to note that this lower focus value did not imply a lack of sharpness or detail in the imagery. Despite the smooth texture, which influenced the focus value calculation, the images of the back side were sharp and detailed, effectively capturing all necessary features. This observation underscores the importance of considering surface texture when interpreting focus values in image quality assessments.

Additionally, the study also evaluated the performance of the disturbance rejection system across the 440 viewpoint poses. Remarkably, only 4 instances of failure were recorded, indicating a failure rate of about 0.91%. This exceptionally low failure rate demonstrates the system's robustness and reliability in maintaining image integrity amidst potential environmental or operational disturbances. Furthermore, in each case where the initial attempt failed, the system successfully corrected the error on the second attempt, reinforcing its efficacy and adaptability.

Chapter 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

This thesis has demonstrated significant advancements in the field of automated inspection systems, specifically tailored for the aerospace industry. The research has successfully addressed the challenges of environmental disturbances in image acquisition, resulting in enhanced robustness and reliability of image data for defect detection. Key developments include an innovative end-of-arm tool and image classification system, designed to mitigate variances caused by unpredictable environmental factors.

The implementation of algorithms, such as the Variational Autoencoder (VAE) and Deep Convolutional Generative Adversarial Network (DCGAN), for training on diverse datasets shows remarkable effectiveness in preparation for real-world scenarios. These models, adept at accurately reconstructing and generating high-quality images, are crucial for defect detection. The research further explores deep learning methods to automate control of lighting and camera settings, thereby ensuring consistent image quality.

Furthermore, this study has contributed to the field by developing algorithms for focus optimization in robotic inspection systems, utilizing macro lenses for high-resolution imaging. This has significantly enhanced the precision and accuracy of inspections, proving especially valuable in detecting minute, safety-critical defects.

In conclusion, this research represents a substantial leap forward in automated inspection systems. It not only achieves a higher level of precision and reliability in defect detection but also provides a scalable and adaptable solution for the aerospace industry. The findings of this study open new avenues for future research and development in this domain, with potential applications extending beyond aerospace to other sectors requiring high-precision inspections.

8.2 Future Work

ROS2 Implementation in Manipulator Focus Algorithm:

- Transition from ROS 1 to ROS 2: This move is significant due to ROS 2's real-time control capabilities, particularly with MoveIt Servo, offering a more efficient and dynamic approach to achieving the highest sharpness in images.

- Control Problem Approach: With ROS 2, the process of finding the highest focus value shifts from being an optimization problem to a control problem. This change implies that the system can continuously adjust the manipulator's position until it reaches the point of highest sharpness, rather than moving in predetermined steps.

- Efficiency in Inspection: This real-time control mechanism promises to significantly reduce the time required for inspections, as the system will no longer need to move in fixed increments but can directly navigate to the point of optimal focus.

Real-World Application of Illuminance Disturbance System with Model Predictive Control:

- Model Predictive Control (MPC) Integration: The use of MPC to control LED lighting in real-world scenarios is a significant step. This system will adjust LED light settings dynamically, ensuring even illumination regardless of the environmental lighting or the position of the object being imaged. Application of VAE-DCGAN Trained Model: Implementing the trained model in practical settings will be a crucial test of its effectiveness. It will involve leveraging the model's predictive capabilities to adjust lighting conditions in various inspection scenarios.

Automatic Color Correction and Post Disturbance Rejection:

- Color Calibration System: Developing a system that allows users to select the color of the parts with a color board, and accordingly adjusts the camera settings and LED light colors, is an innovative approach. This system aims to ensure that the captured image colors closely match the selected color standard.

- Post Disturbance Rejection Enhancement: Adding a feature to re-queue and retake images if there is a deviation from the set color standard or inconsistency with other images will further enhance the robustness of the inspection process. This step will ensure consistent color accuracy across all images.

BIBLIOGRAPHY

[1] A. Agarwal et al. Robotic defect inspection with visual and tactile perception for large-scale components. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10110–10116, Detroit, MI, USA, 2023. IEEE.

[2] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, USA, 1st edition, 2019.

[3] S. Dey. Using pca to detect outliers in images. https://sandipanweb.wordpress.com/2016/06/25/using-pca-to-detect-outliers-in-images/, 2016.

[4] C. G. Drury and J. G. Fox. The imperfect inspector. In C. G. Drury and J. G. Fox, editors, *Human Reliability in Quality Control*, pages 11–16. Taylor Francis, London, UK, 1975.

[5] M. Gerges. Provision of controlled and consistent light distribution over an uneven topography to maximize efficacy of machine-vision based defect identification. Master's thesis, University of Washington, 2021.

[6] H. Golnabi and A. Asadpour. Design and application of industrial machine vision systems. *Robotics and Computer-Integrated Manufacturing*, 23(6):630–637, 2007.

[7] I. Goodfellow. Generative adversarial nets. *Communications of the ACM*, pages 139–144, 2020.

[8] F. C. Groen, I. T. Young, and G. Ligthart. A comparison of different focus functions for use in autofocus algorithms. *Cytometry*, 6(2):81–91, March 1985.

[9] A. Koptelov. Automated visual inspection. https://www.itransition.com/blog/automated-visual-inspection, 2021.

[10] A. Kulkarni. Motion planning and image capturing for robotic inspection of a curved surface subject to imaging constraints. Master's thesis, University of Washington, 2020.

[11] R. Li, W. He, and S. Liu. Three-dimensional point cloud data pre-processing for the multi-source information fusion in aircraft assembly. *Applied Sciences*, 13(8):4719, 2023.

[12] R. Mehrotra, K.R. Namuduri, and N. Ranganathan. Gabor filter-based edge detection. *Pattern Recognition*, 25(12):1479–1494, 1992.

[13] S. Mitra. Improving visual inspection quality with ai. https://metrologicallyspeaking.com/improving-visual-inspection-quality-with-ai/, 2021.

[14] J. L. Pacheco and G. Cristobal. Diatom autofocusing in brightfield microscopy: A comparative study. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*. IEEE, 2000.

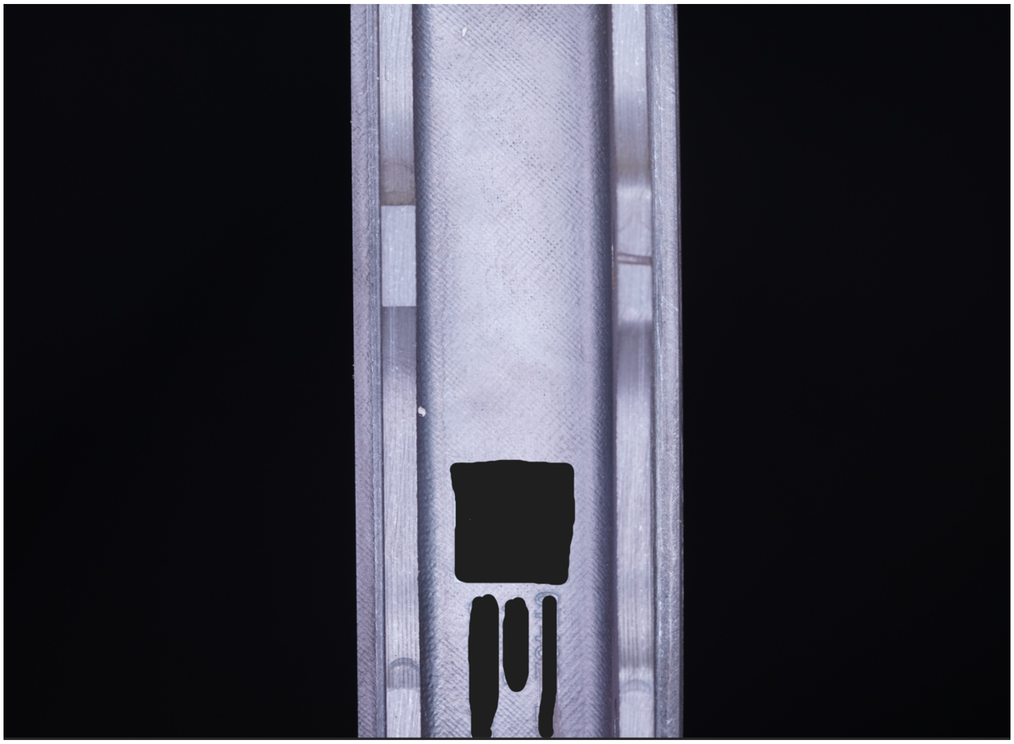
[15] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

[16] A. Rosebrock. Image masking with opencv. https://pyimagesearch.com/2021/01/19/image-masking-with-opencv/, 2021.

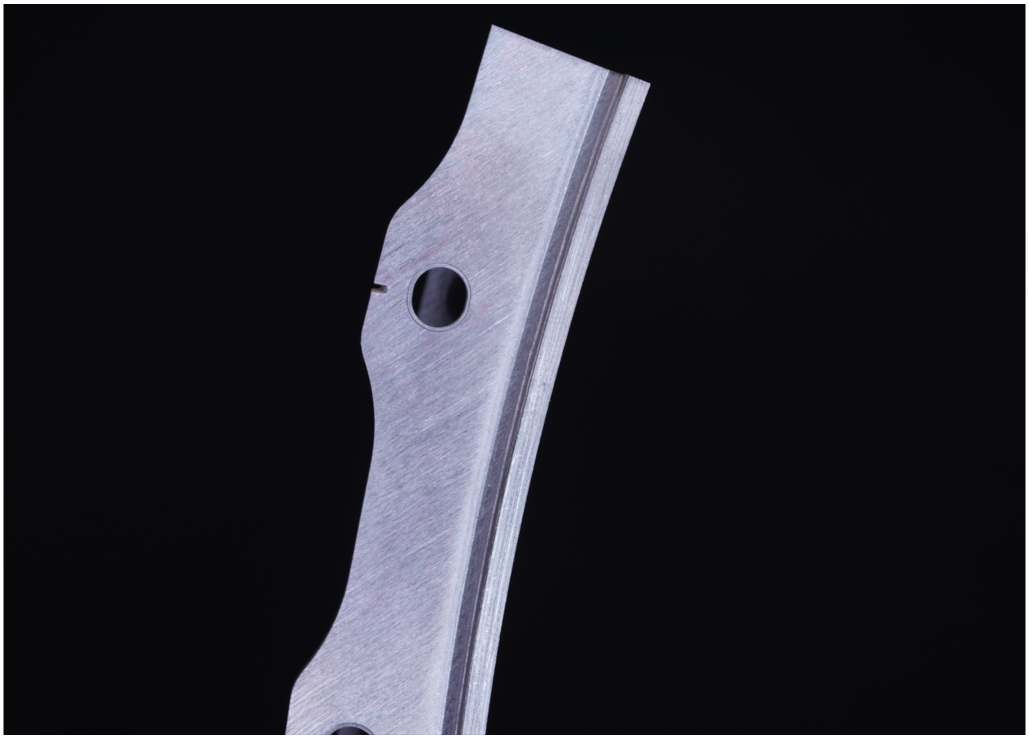
[17] The-Digital-Picture.com. What is a macro lens? https://www.the-digital-picture.com/Canon-Lenses/What-Is-A-Macro-Lens.aspx, n.d.

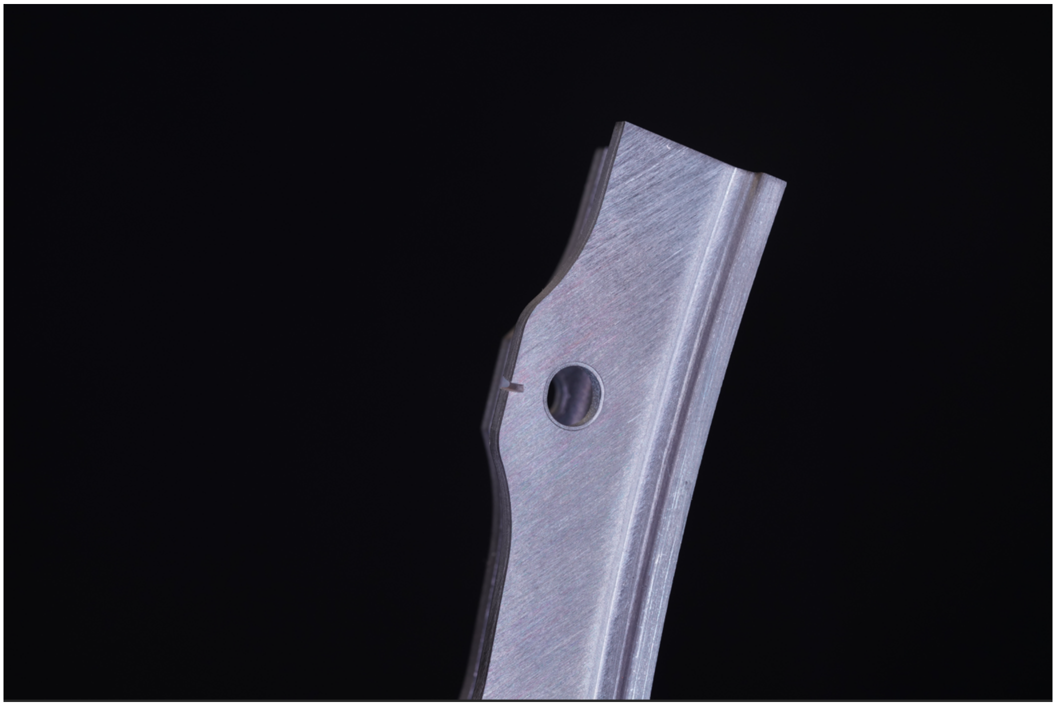
Appendix A

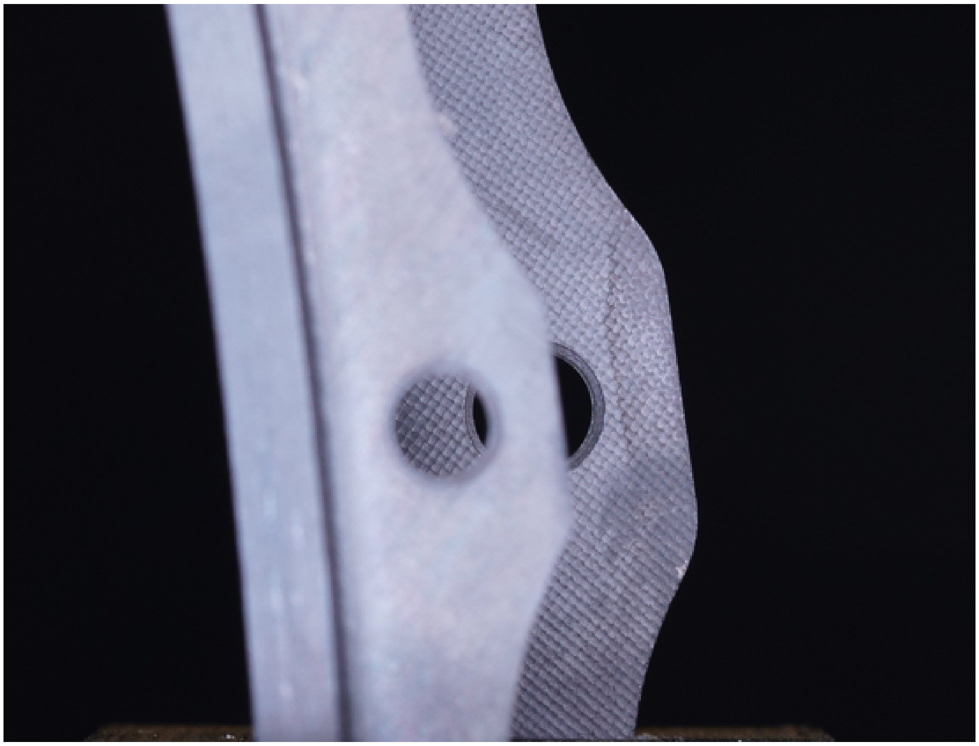
IMAGE DATA

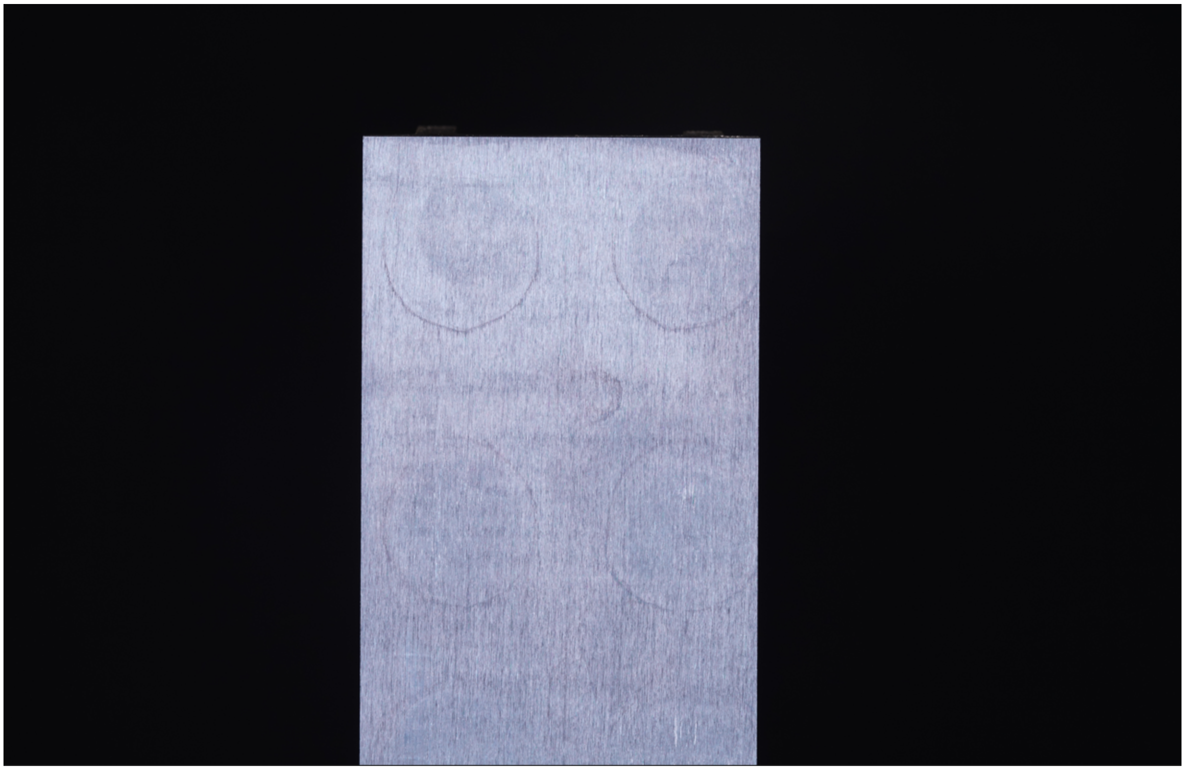


[redacted]



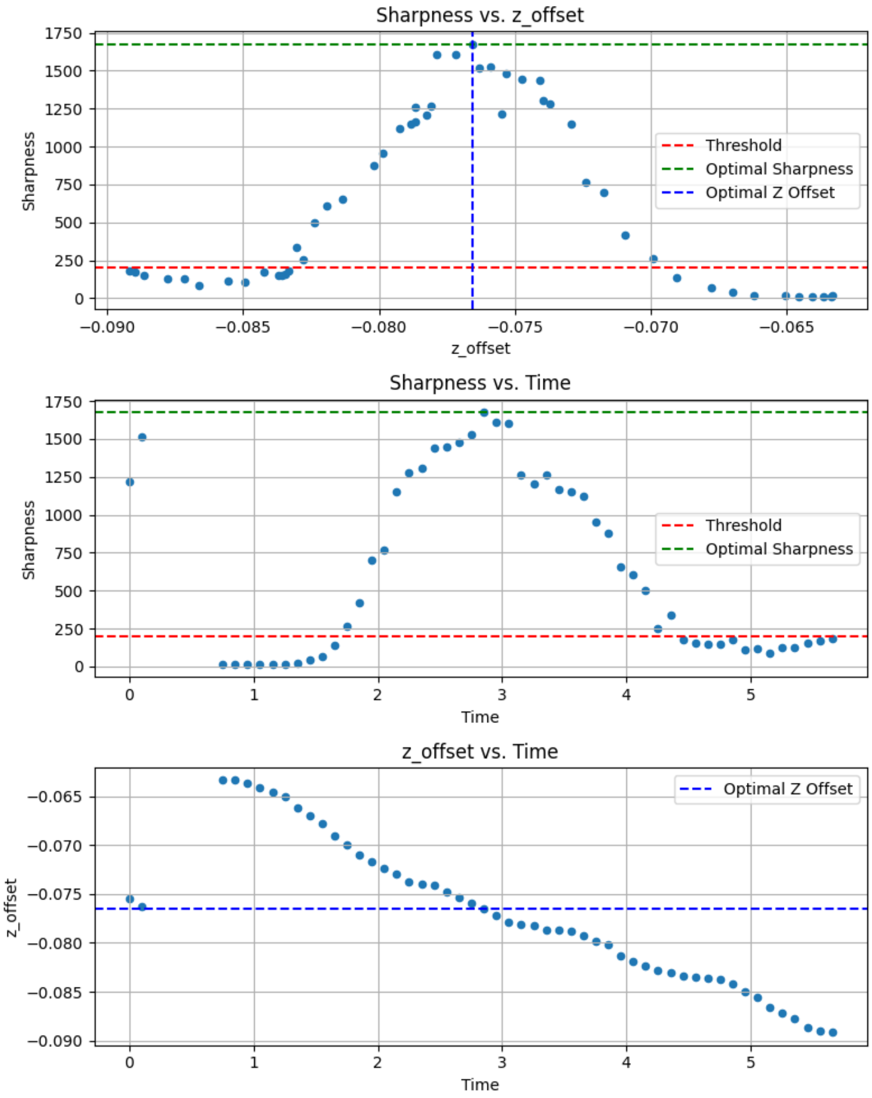


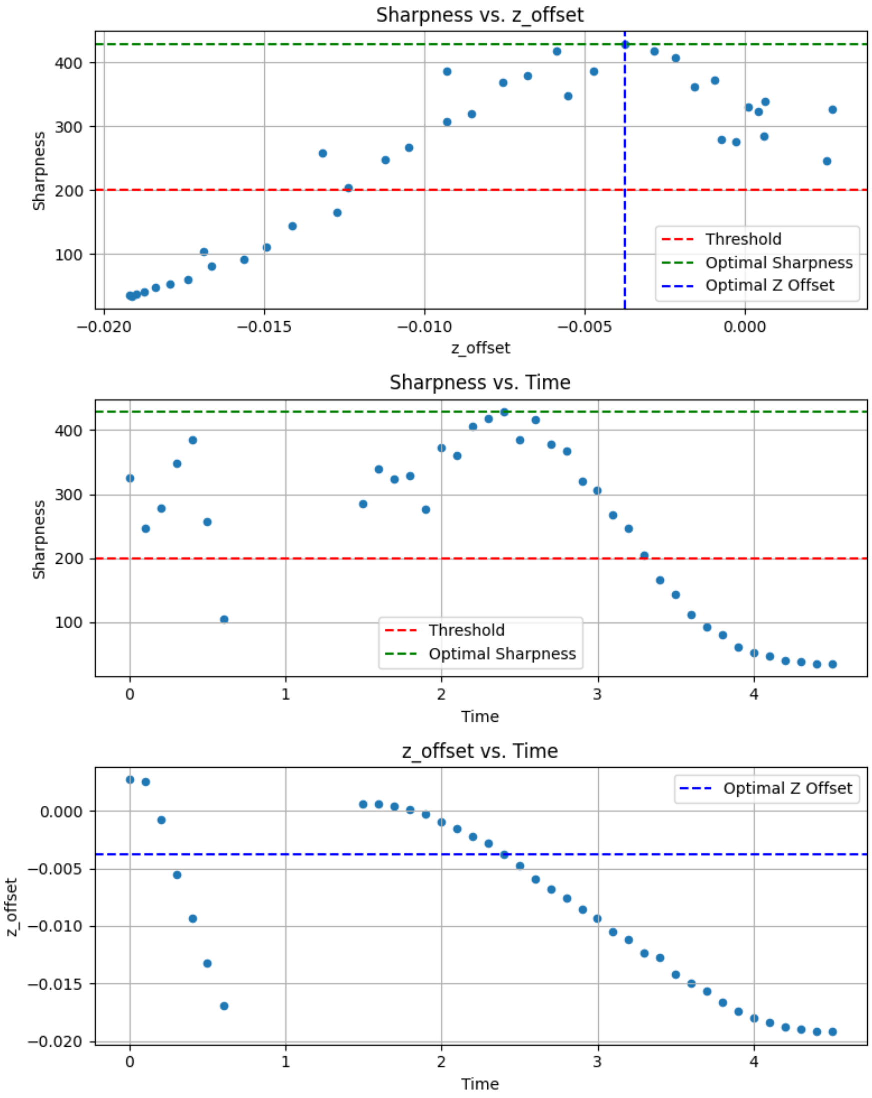


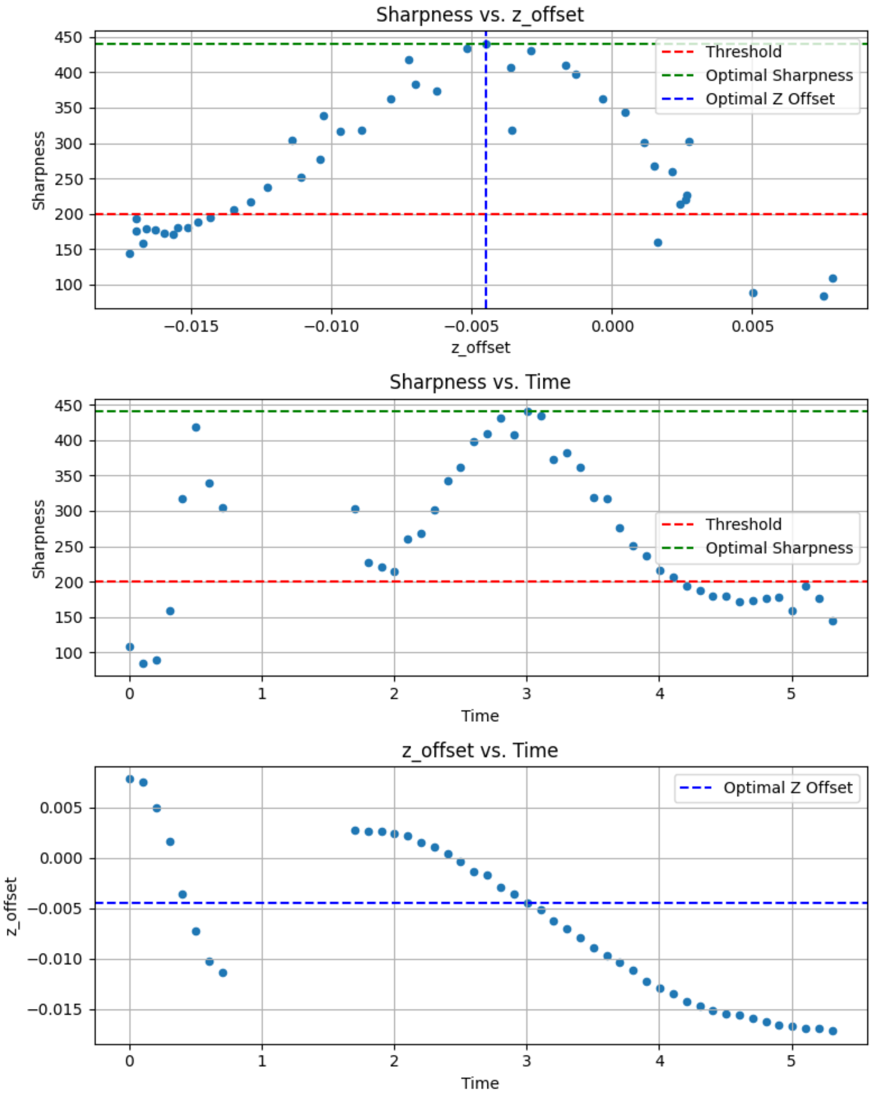


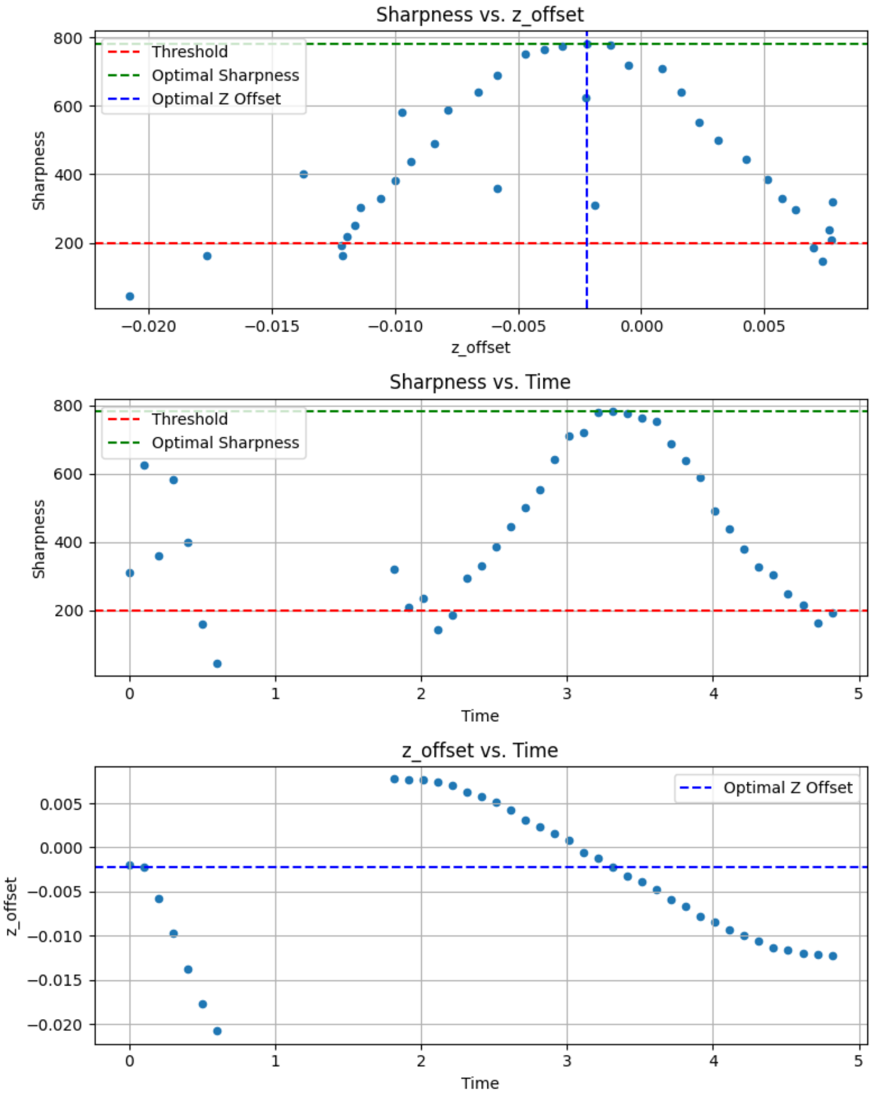
Appendix B

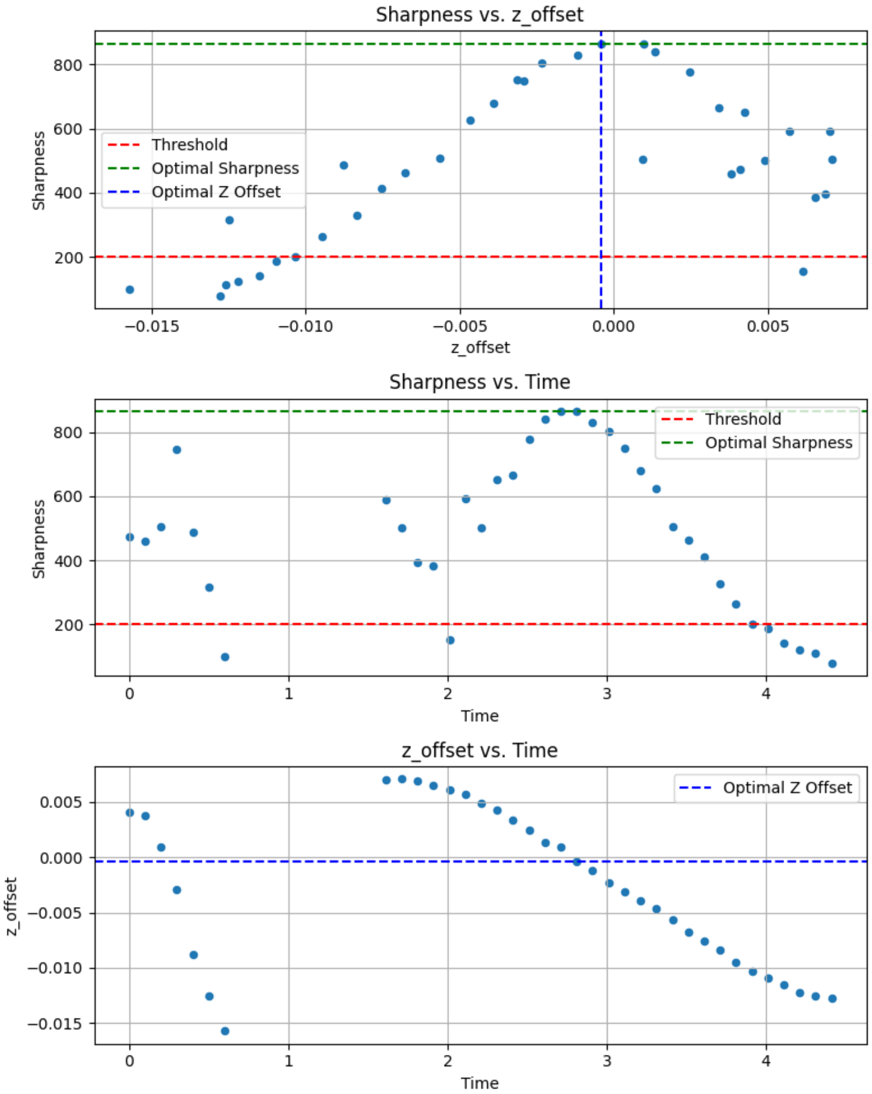
ROSBAG DATA

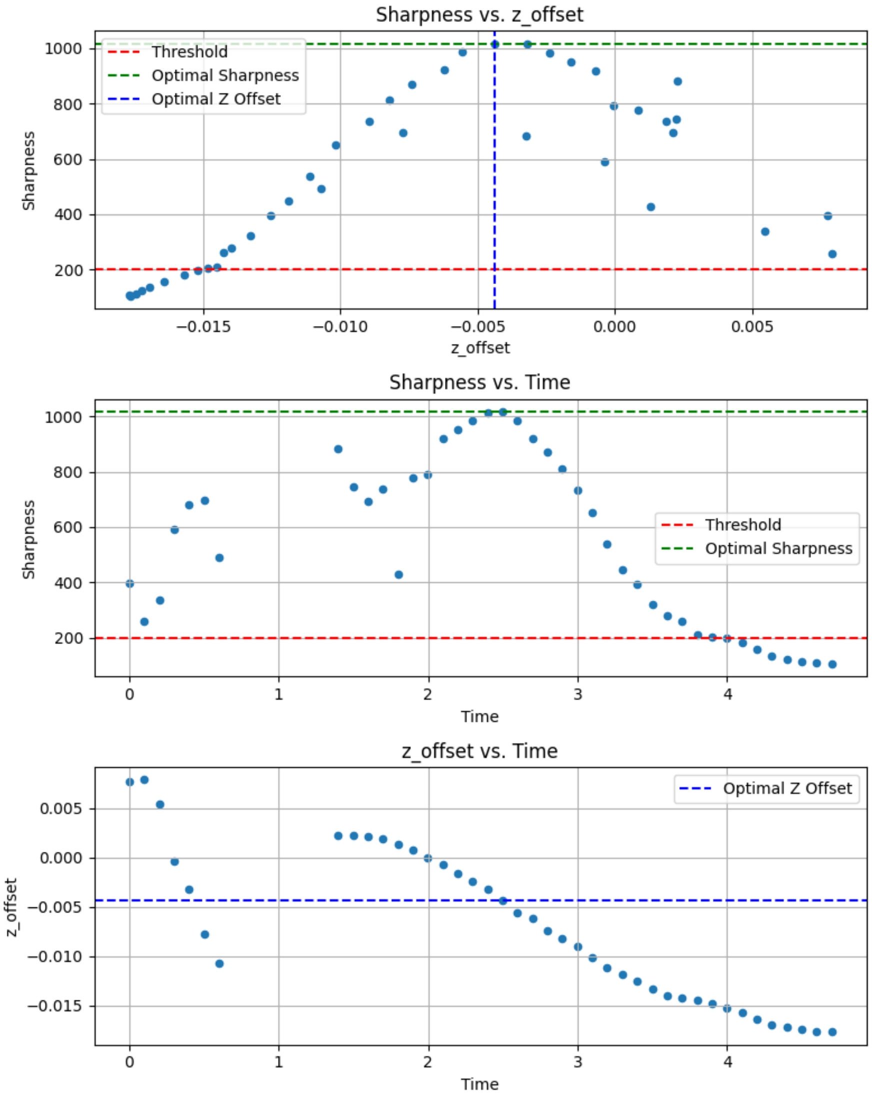


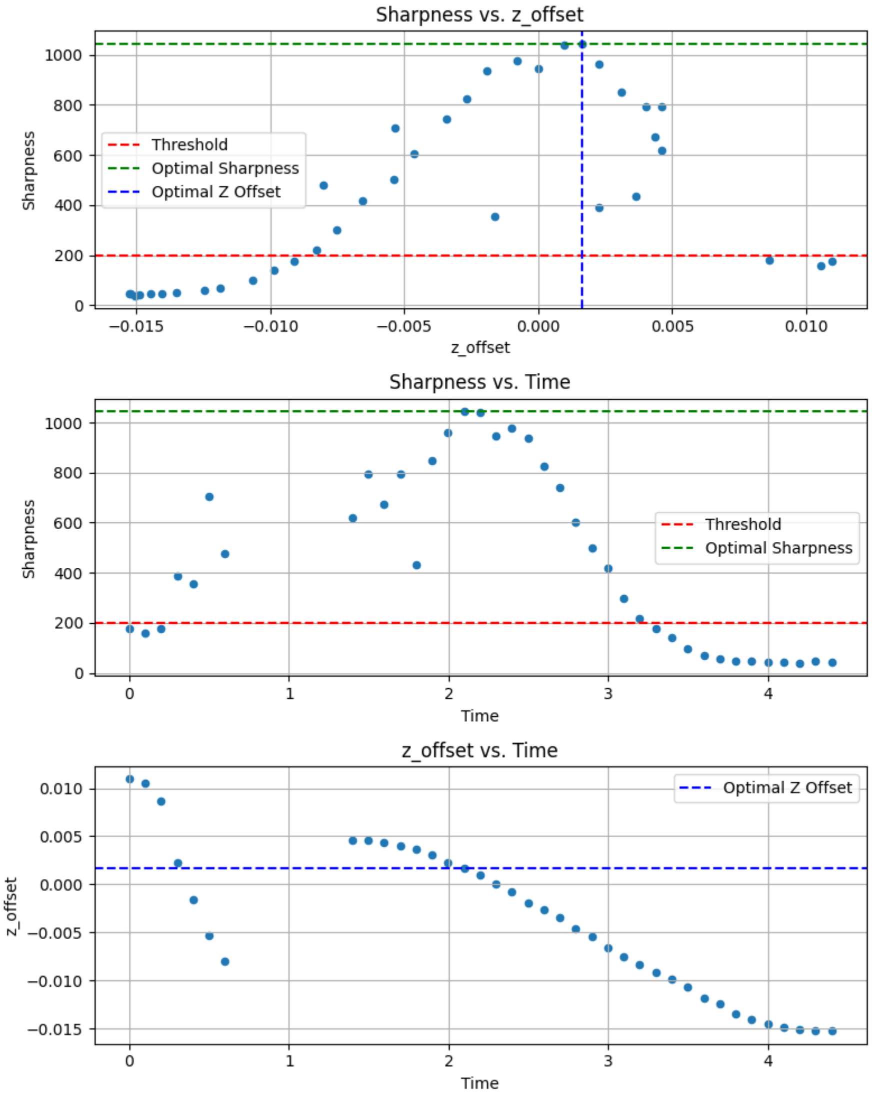

Sharpness vs. z_offset
Sharpness
400
300
200
100
−0.020
−0.015
−0.010
−0.005
0.000
z_offset
Threshold
Optimal Sharpness
Optimal Z Offset
Sharpness vs. Time
Sharpness
400
300
200
100
0
1
2
3
4
Time
Threshold
Optimal Sharpness
z_offset vs. Time
z_offset
0.000
−0.005
−0.010
−0.015
−0.020
0
1
2
3
4
Time
Optimal Z Offset


Sharpness vs. z_offset
Sharpness
z_offset
450
400
350
300
250
200
150
100
−0.015
−0.010
−0.005
0.000
0.005
Threshold
Optimal Sharpness
Optimal Z Offset
Sharpness vs. Time
Sharpness
Time
450
400
350
300
250
200
150
100
0
1
2
3
4
5
Threshold
Optimal Sharpness
z_offset vs. Time
z_offset
Time
0.005
0.000
−0.005
−0.010
−0.015
0
1
2
3
4
5
Optimal Z Offset


Sharpness vs. z_offset
Threshold
Optimal Sharpness
Optimal Z Offset
Sharpness
800
600
400
200
−0.020
−0.015
−0.010
−0.005
0.000
0.005
z_offset
Sharpness vs. Time
Threshold
Optimal Sharpness
Sharpness
800
600
400
200
0
1
2
3
4
5
Time
z_offset vs. Time
Optimal Z Offset
z_offset
0.005
0.000
−0.005
−0.010
−0.015
−0.020
0
1
2
3
4
5
Time


Sharpness vs. z_offset
Threshold
Optimal Sharpness
Optimal Z Offset
Sharpness
800
600
400
200
−0.015
−0.010
−0.005
0.000
0.005
z_offset
Sharpness vs. Time
Threshold
Optimal Sharpness
Sharpness
800
600
400
200
0
1
2
3
4
Time
z_offset vs. Time
Optimal Z Offset
z_offset
0.005
0.000
−0.005
−0.010
−0.015
0
1
2
3
4
Time


Sharpness vs. z_offset
Threshold
Optimal Sharpness
Optimal Z Offset
Sharpness
1000
800
600
400
200
−0.015
−0.010
−0.005
0.000
0.005
z_offset
Sharpness vs. Time
Threshold
Optimal Sharpness
Sharpness
1000
800
600
400
200
0
1
2
3
4
Time
z_offset vs. Time
Optimal Z Offset
z_offset
0.005
0.000
−0.005
−0.010
−0.015
0
1
2
3
4
Time


Sharpness vs. z_offset
Sharpness
z_offset
1000
800
600
400
200
0
−0.015
−0.010
−0.005
0.000
0.005
0.010
Threshold
Optimal Sharpness
Optimal Z Offset
Sharpness vs. Time
Sharpness
Time
1000
800
600
400
200
0
0
1
2
3
4
Threshold
Optimal Sharpness
z_offset vs. Time
z_offset
Time
0.010
0.005
0.000
−0.005
−0.010
−0.015
0
1
2
3
4
Optimal Z Offset