Application of Stereo Depth Camera and Alignment Algorithms for Part Monitoring During Machining

by

Samuel Delattre

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

A stereo-depth camera is proposed to be used in conjunction with fiducial markers on a calibration plate and a fine-tuning alignment algorithm for part monitoring in a CNC machine. Together, a selected pyramid-shaped part within the machine could be monitored. The position, orientation, geometry and surfaces of the pyramid part are measured and compared with the pyramid's desired model.

This system can monitor the position and geometry within 1mm of accuracy, orientation within 1 degree of accuracy, and surface fitting within 2mm of accuracy, which closely aligns with the advertised accuracy of the stereo depth camera. While the accuracy is not enough to be confident if the part was machined to an industrial tolerance for most machined parts, this accuracy is sufficient to show that a part roughly matches the position and expected geometry of the model. This information allows the machine to monitor for any significant interference or interruptions during and after machining.

Acknowledgements

This thesis would not be possible without the support of my mentors; Professor Sanjeev Bedi and Professor Stephen Mann. Their guidance, patience, insight and humour have been a huge help and relief throughout the years. I am fortunate to have such thoughtful and incredible supervisors.

Additionally, I am thankful to Jason Benninger and Karl Janzen for their support in manufacturing parts and for always helping me access the CMM. Thank you to Professor Allan Spence for making the calibration plates and providing assistance with Geomagic. Thank you to Eugene Li and Michael Lenover for their ideas and suggestions that pushed my research forward.

Special mention to my friends Coel, Sam, Cassandra, Connor, Laszlo, Emma and John. Their encouragement and well wishes gave me the courage to keep going, and also helped me de-stress with board games.

I would finally like to thank my family for their love, support and guidance during my studies and life.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Computer Numerically Controlled (CNC) machines are used in various industries to remove material from raw stock and manufacture different components. These machines can be used for large batches of similar parts for quick manufacturing or for quick manufacturing of smaller, more customer-specific requirements. Although CNCs are semi-autonomous, much of the CNC work has been moved offshore due to the costs associated. During the life of a machine, most money is spent on operators and software [1]. However, in recent years, machining has been returning to Canada. Unfortunately, there has also been a decline in new students interested in studying to be machine operators, and many students who enter the field are not being taught manufacturing in person and are unaware of practical skills and knowledge that would aid their work [2]. Skilled machine operators are well-versed with knowledge and experience that aids them in many aspects of their work, including efficient toolpath planning and awareness of the sounds and conditions that may lead to chatter. The disparity between the growing machining industry and the decline in machinist graduates, more experienced machinists retiring, and the overall loss of specific skills and knowledge in the machining industry has spurred the need for research into new solutions. The autonomous CNC research group aims to assist in the return of this industry by reducing the dependence on experienced labour. The solution

involves developing systems and embedding specific knowledge and experience into a more autonomous CNC machine.

CNC machines are mostly autonomous. This means that at a base level, if the part and tooling are appropriately mounted, and the machine is zeroed and calibrated for the raw stock and is provided with a toolpath, the machine should be capable of machining the desired part with no additional operator intervention. However, these additional steps before starting the CNC machining often take a lot of the production time for parts machined by these CNC machines.

Additionally, although CNC machines can follow the toolpath commands to achieve the desired machining, the machines need to be monitored for unexpected issues such as tool breakage, chatter, and incorrect movement of the gantry. While some modern CNC machines may include some sensors for determining if the tooling is broken or if the machine is attempting to move the gantry further than the machine is capable, these solutions add additional movements and time during the machining process. And while these potential issues can often be remedied by taking slower or more shallow cuts, this remedy leads to longer manufacturing times.

The overarching issues are the requirement and loss of specific knowledge required by CNC machinists, a general lack of quick and useful autonomous feedback to inform the CNC if issues may have occurred, and the inability of current machines to attempt to fix issues themselves without direct assistance from an operator.

1.2 Objectives

The objective of the research was to determine the viability of using the stereo depth camera for observing a part within a CNC workspace.

The research completed for this thesis suggests a cheaper, scalable solution for the raw stock detection, part monitoring during machining, and part validation after machining is complete.

The proposed solution uses three physical components and various algorithms to monitor the part while it is being machined and validate that the machining was completed to

a certain tolerance.

The components consist of an Intel RealSense stereo depth camera, an Epson LCD Projector H550A, and three or more fiducial markers placed within the CNC machine. The camera can be mounted anywhere. For this research, the camera was mounted above the part facing downwards. The projector must be pointed towards the part, and the markers mounted on the bed of the CNC machine to form two perpendicular lines closely matching with the X and Y axes of the CNC machine. Using this setup, the projector projects a pattern onto the surface of the part, which will improve the quality and precision of the scan created by the stereo depth camera. The stereo depth camera captures the CNC machine and mounted part as a point cloud, which is then transformed based on the axes produced by the fiducial markers to align the part with the CNC coordinate system, which in turn is aligned with the coordinate system of the designed model of the part to be machined. The aligned scan is compared with the desired part to determine the accuracy and precision of the proposed solution, focusing on a few key metrics.

Chapter 2

Literature Review

Many companies have been focusing on Industry 4.0 and looking towards Industry 5.0. Much of this focus has been on making systems more aware and adaptable to changes. CNC machining has also been experiencing this change, including a proposed shift in paradigm proposed by groups such as Poon *et al.* [3] that suggested a system that is more aware using a CAD model of the workpiece existing inside of the CNC controller. The CAD model existing inside of the controller would allow for real-time tool position generation and simulation before sending the commands to be run on the CNC. The generation and simulation are completed without human intervention and would decrease intensive labour and the potential for catastrophic damage.

Part positioning and geometry verification are other areas of exploration to make machines more automated and aware. Dimensional metrology technologies such as Computer Measuring Machines (CMM) are more common in the manufacturing and machining industries. These typically involve the use of a probe to determine the position of the part or features. CMMs can be programmed to automate inspection and improve productivity [4]. Software is used to compare the position and geometry of the probed parts. These machines rely on physically probing the part. Alternatively, 3D scanning technologies have been experimented with for non-contact dimensional metrology [5].

3D scanning technologies and cameras have been primarily used in the robotics industry to make robots more aware of their surrounding. Many groups such as Tadic *et al.* [6] have

applied depth cameras in robotics to detect and extract key geometry such as obstacles. Popular depth cameras include Stereo RBG Depth, LiDAR, and structured light scanners.

Scanning technology has also been used previously in the manufacturing sector for reverse engineering. Budzik [7] used laser scanning and computed tomography for the scanning of an aircraft engine blade. This method also enabled Budzik to scan closed profiles of the aircraft engine blade, including the inner cooling canals of the blade.

An application of 3D scanning technologies is automating part positioning to make machines more aware without human intervention. Pajor *et al.* [8] discuss a vision system for quick matching a workpiece reference point using a non-contact 3D scanning method, based on structural light patterns.

Another method of improving the resulting scans taken using stereo depth scanning technologies was explored by many groups, including Madeira *et al.* [9]. Fiducial markers were used to better refine transformations rather than relying on aligning features that may appear differently between multiple scans or how they are expected to appear. The fiducial markers can be robustly detected which helps to overcome incorrectly matching features. Some of the more popular fiducial markers include ArUco [10] and AprilTag [11].

Many of the sensors used in existing research would output their scans in the form of a point cloud. However point clouds are often noisy and can include multiple points at the same location in space. Additionally, they are large amounts of data that many current systems have trouble efficiently dealing with. For this reason, many other works encode the point clouds into other forms. Maturana *et al.* [12] encode the point clouds into the form of voxels, which make a volumetric occupancy grid to represent the scene. These representations are also capable to be used with Convolutional Neural Networks (CNNs) for object recognition.

The effects of improper lighting of the scene being scanned are also a significant consideration for the quality of any scans taken. McDermott *et al.* [13] experimented with introducing a preprocessing step to mitigate the effect of shadows on the quality of scans they took using LiDAR. Their method was based on spherical gridding and used multiple scans to detect the edges of the shadows and eliminate them.

In this work, a stereo depth camera scans a part within the workspace to determine its

position, rotation, and geometry. ArUco markers are used to initially align the camera's coordinate system with the CMM or CNC machine workspace. An additional algorithm is used to further improve the precision and accuracy of the scans for comparison with either the desired position of the stock material or the final machined part.

Chapter 3

Test Setup Overview

This chapter will explain how the stereo depth camera used in this research is capable of extracting a 3D point cloud of the workspace within its field of view. It will also explain why the type of scanning method was chosen. The additional components, including a calibration plate with fiducial markers, a projector, and a pyramid-shaped part will be shown and explained. Finally, an overview of the test setup on the CMM will also be explained and illustrated. This test setup shows how the CMM was used to verify the locations of all coordinate systems during tests, and the pyramid part features that were observed by the stereo depth camera.

3.1 RealSense Stereo Depth Camera

The primary depth camera used was the Intel RealSense D405 (Figure 3.1), a stereo depth camera designed for close-range applications. The Intel RealSense D455 was also used during the beginning of testing. The D405 was released in April 2022, and the RealSense D455 was used for testing before its release.

Stereo depth cameras leverage two or sometimes more cameras to emulate how humans perceive depth. The Intel RealSense stereo depth cameras have two cameras. The cameras are positioned at a known distance apart from each other and are referred to as the left

Figure 3.1: Intel RealSense D405 Stereo Depth Camera used for experimentation

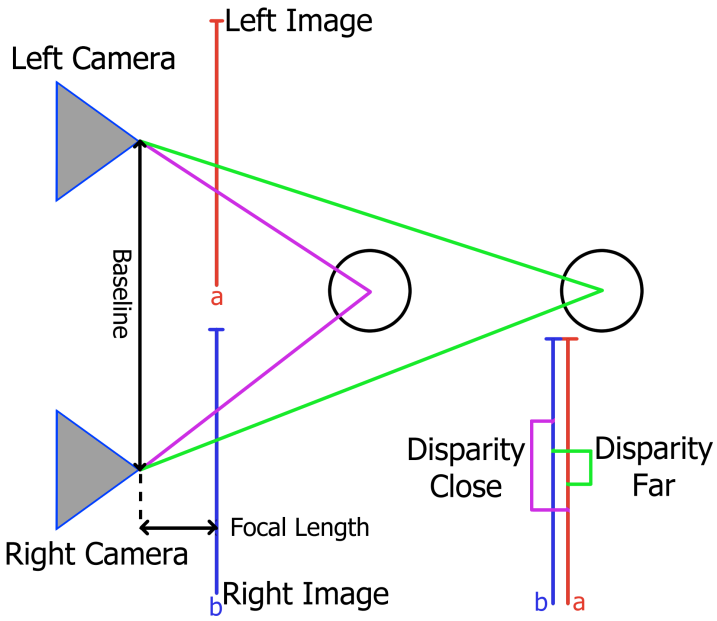


Figure 3.2: Diagram of stereo depth perception. Each camera captures its own image of the scene. The *correspondance problem* must be solved to find the matching points between the two images. By superimposing the left and right images, the *disparity* is the pixel positional difference between the matching points found in each image. Objects that are closer to the cameras have a larger disparity compared to objects that are further away.

and right camera. Since the cameras are positioned apart, they perceive different views. The same point must be determined in each image to determine the distance to objects in the camera's view. The pixel positional difference between each pair of matching points is found. Objects closer to the cameras will appear to be shifted by a larger distance along the horizontal line, also called the epipolar line, compared to things farther away. The shift is known as the *disparity*. The disparity for every pair of matching points in the scene, along with the known baseline distance between the left and right camera and the camera focal lengths, can be found. These distances are shown in Figure 3.2. The objects closer to the camera are found further apart when the images from the left and right cameras are compared. Objects further away from the camera are closer to the same position in the left and right images. These differences in position are what the *disparity* is referring to. This information can be used to calculate the depth or distance from the camera to every point in the scene in front of the camera using triangulation. This depth calculation is done by multiplying the baseline by the focal length divided by the disparity [14].

$$Depth = \frac{Baseline \times FocalLength}{Disparity} \tag{3.1}$$

This process's most essential and challenging aspect is finding the same point within the images produced by the left and right cameras, known as solving the *correspondence problem*. The algorithms used to solve this problem are complex and require a large amount of processing power. The cameras used for this thesis were designed by Intel, who created custom silicon for depth sensing that significantly accelerates the computations to achieve speeds of over 36 million depth points/second using a custom variant of a Semi Global Matching algorithm. The silicon is packed into a chip around 6.3×6.3mm and searches across 128 disparities [14].

The correspondences between the left and right images should be much better than a fraction of a pixel resolution to increase the quality of the stereo depth algorithm. Intel advertises 0.05 sub-pixel accuracy for well-textured passive targets [14], but typically varies and is worse for objects with little to no texture. To assist with these less ideal conditions, external projectors can be used to create a texture on the scene, known as "assisted" or "active" stereo, which acts similar to structured light systems. However, unlike structured

light systems, active stereo does not require a specific pattern or to be calibrated to the laser, and the spots do not need to be much brighter than the background image.

Most models of Intel's stereo depth cameras use an IR projector built into the camera. However, the D405 model does not have a built-in projector and includes filters on both cameras that filter out infrared light. Therefore, a pattern can be projected using the visible light spectrum. While the projected patterns improve the quality of stereo vision for untextured objects, the subpixel performance is usually around >30% worse than well-textured passive targets [14].

Stereo depth cameras have the advantage of accessing the 2D colour image from one of the cameras. Specifically, the RealSense software development kit is openly available, which allows more customization and access to many features that would normally be inaccessible with other cameras. The extraction of additional information, such as the colour of the objects within the scene being scanned, can be achieved by the stereo depth camera. The 2D image can also be used with existing computer vision algorithms, such as edge detection, fiducial markers for pose estimation, or even more modern artificial intelligence algorithms like You Only Look Once (YOLO) [15].

3.2 Components and Setup on CMM

To connect the physical part and the desired model for comparison, the stereo camera must find the physical part in the coordinate system of the CNC machine. However, the stereo camera, the CNC machine, and desired model have different coordinate systems. The connections between the coordinate systems must be determined, which could be done using various methods. Additionally, several self-set goals are desired for the setup. The goals include avoiding the need to mount the camera precisely, calibrating the camera with a touchless approach, and allowing the calibration to be completed without the user requiring specialized skills. During testing, a CMM was used in place of a CNC machine to precisely measure the part position and size for verification and for determining precision.

The chosen method introduces an intermediary coordinate system created by a plate with three precisely located fiducial markers on it, which is used to calculate an initial

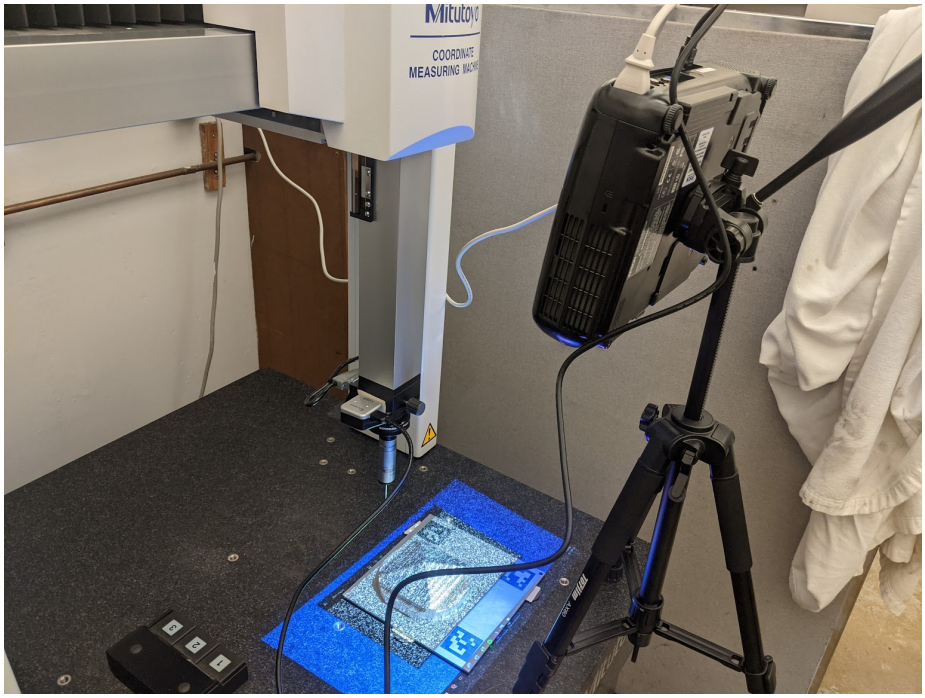


Figure 3.3: Test setup overview for applying the stereo depth camera and algorithms to detect a pyramid-shaped part. The stereo camera is mounted to the head of the CMM for camera location consistency. The calibration plate is mounted in the workspace, and one of the pyramid parts is mounted above. The projector projects a pattern onto the pyramid surface to improve scan quality.

rotation and displacement to be applied to align the camera's coordinate system with the plate. The rotation and displacement are then fine-tuned using a pyramid that is scanned by the camera. The scanned points are fit to the four faces of the pyramid to determine the pyramid's location and rotation to be applied to future scans.

The plate is fixtured inside the CNC, or for testing, the CMM. The stereo depth camera was positioned above the workspace, facing downwards, having a view of the plate and the markers engraved onto it. The camera's orientation was chosen to face down towards the part, assuming that all required features would be only on the top face of the part. The CMM's coordinate system is calibrated to be aligned with the edge of the plate, and the

stereo camera identifies the marker's location in the 2D image created by the stereo camera. The pixel locations of the markers in the 2D image can be converted to their location within 3D space. The camera then uses these markers to represent the coordinate system of the plate and roughly aligns point clouds from scans taken to the plate's coordinate systems. Finally, the pyramid part is placed in the center of the camera's field of view, and an external projector is used to project a pattern on the pyramid's surface to help improve the scan quality. The setup is shown in Figure 3.3.

The choice to calibrate the CMM's coordinate system with the edge of the plate was used to verify the location of the plate's coordinate system and allow the CMM to be used to verify the locations of the markers on the plate that the camera detects in addition to the pyramid part coordinate system and the pyramid's features. The location of the stereo depth camera's coordinate system did not need to be verified because the proposed solution relied on using the markers and alignment algorithms to match the camera's coordinate system with the calibration plate and pyramid part coordinate systems. However, verification was completed by mounting the camera securely to the same arm as the CMM's probing tool. This allowed the precise location of the camera to be measured by the CMM with respect to the plate's coordinate system that the CMM was aligned with.

Multiple point clouds are taken of the pyramid to be used in fine-tuning. This fine-tuning alignment is achieved using an algorithm that determines the four faces of the pyramid from the points and uses the resulting normals and position of the apex to inform the positional and rotational offset between any future point clouds and the real world. The pyramid could then be removed, and the parts to be scanned could be positioned within the camera's field of view. Pyramid parts were again used, and the pyramid fine-tuning algorithm was reused to determine the accuracy and precision of the pyramid location and geometry found by the stereo camera and software.

Chapter 4

Implementation

This chapter will go into detail to explain how the CMM and camera coordinate systems were aligned using a calibration plate with fiducial markers to allow the results from the stereo depth camera and various alignment algorithms with the measured results from the CMM. It will describe the methods and algorithms used to scan and roughly align the resulting point clouds and the fine-tuning alignment algorithm used. The importance of the CMM alignment was to verify the locations of the various coordinate systems of the plate and pyramid, as mentioned in Chapter 3. Finally, the key inspection metrics that were used to compare with the CMM and other existing software will be shown.

4.1 Calibration of Scanning Systems

Tests are performed on the CMM, which was used to verify the coordinate system alignments. The calibration plate was mounted in the CMM, and the CMM probed the edges and top of the plate to calibrate the CMM with the bottom left edge of the plate. During testing, the stereo depth camera was mounted to the CMM probe to move the camera by precise distances. The camera observed the fiducial markers on the calibration plate to calibrate the camera with the bottom left edge of the plate. The pyramid was then mounted in the camera's field of view and scanned. To verify the pyramid's location found by the

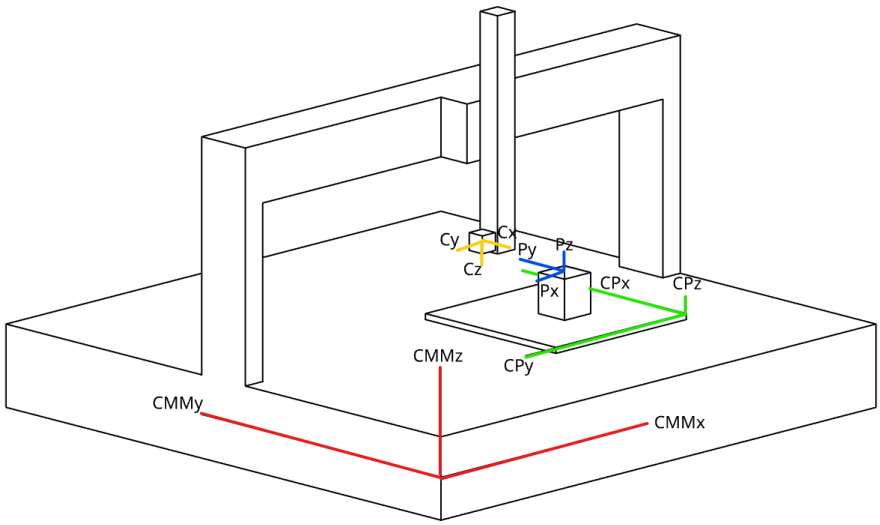


Figure 4.1: Coordinate systems to be aligned with alignment algorithms. The CMM coordinate system (CMM) is shown in red. The calibration plate (CP) coordinate system is shown in green. The pyramid part (P) coordinate system is shown in blue. The camera (C) coordinate system is shown in yellow. Since all coordinate systems are mounted to the CMM, the CMM was used to verify the location of each coordinate system.

camera and the pyramid's coordinate system, the CMM was used to probe the pyramid's edges, faces, and apex. The following sections further detail the calibrations between the CMM, camera, calibration plate, and pyramid.

4.1.1 Motivation for Calibration

The experimental setup has many calibration steps. These steps allow the resulting point cloud to be modified to make simplified and accurate verification methods possible. Additionally, the setup would make the system operate similarly to how CAD models and CNC machines interact and align their coordinate systems. The various coordinate systems are shown in Figure 4.1

The coordinate systems are used to derive transformations that are applied to the points from the camera's coordinate system, C, to align them with the the pyramid part coordinate

system, P. Additionally, transformations from the CMM's coordinate system, CMM, to the pyramid part coordinate system, P were used to verify the pyramid part features. The point cloud created by the stereo depth camera is initially in the camera coordinate system, C. The point cloud then has its orientation and position modified to be relative to the orientation and position of the plate in the calibration plate's coordinate system, CP, and then is further tuned using a pyramid calibration piece in coordinate system P. Similarly, when the CMM measures the object, the measurements are adjusted from the CMM's coordinate system, CMM, to be relative to the orientation and position of the plate's coordinate system, CP. By aligning the CMM with the plate's coordinate system, the CMM was used to verify coordinate systems and pyramid features. The alignment of the camera's coordinate system with the plate and pyramid coordinate systems allows for using either existing point registration algorithms or other methods to compare the desired model and the point cloud created from scanning the machined part. The comparisons can then be verified by the CMM.

While the calibration of the CMM with the plate for verification is relatively simple and can be handled by the CMM, the calibration of the stereo depth camera with the plate could be achieved in various ways. The chosen method relies on a two-step method that first uses fiducial markers precisely engraved upon the plate to be found by the stereo depth camera as shown in Figure 4.2. In the second step, the alignment is fine-tuned using an algorithm for finding and fitting a calibration pyramid.

4.1.2 Design of Calibration Plate

The design of the calibration plate used during the experiments underwent a few design considerations. Two different materials were used for the plate designs, aluminum and TroLase, an acrylic-based laminate with a top and internal layer colour. In both cases, the plate is precisely squared off using a CNC machine, which allows the plate to be aligned with a laser engraving system to add the markers to the plate. For the aluminum plate, typical laser engraving can not be used because the laser would reflect off the plate's surface. However, a metal marking spray can achieve the desired result. The metal marking spray covers the selected areas where a marker should be engraved. The laser then engraves the

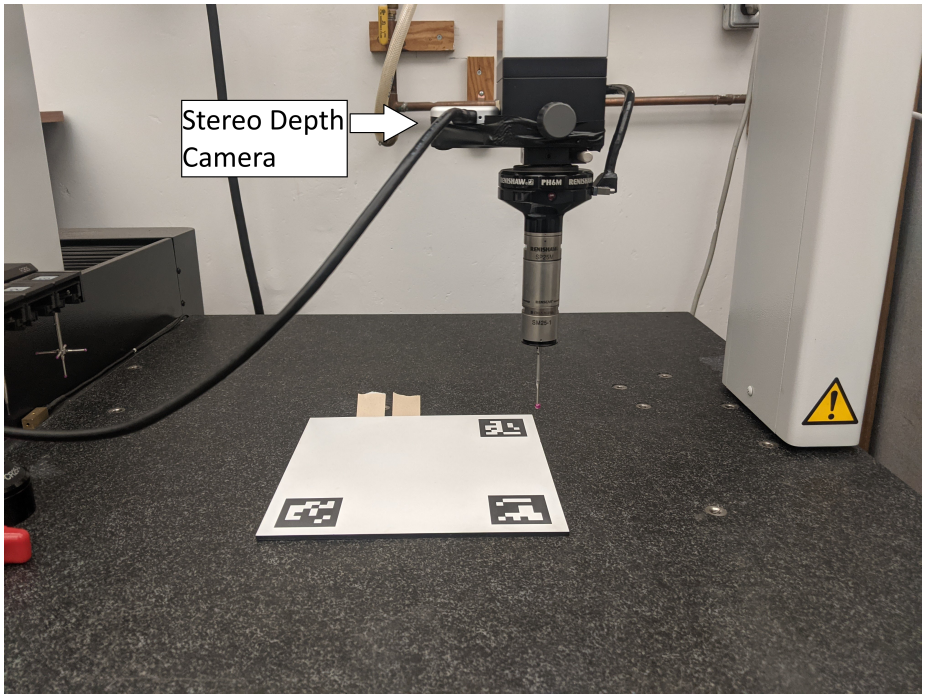


Figure 4.2: Stereo depth camera is mounted to the CMM head. The calibration plate is mounted in the workspace below. The CMM and stereo depth camera can be calibrated to the same coordinate system as the plate.

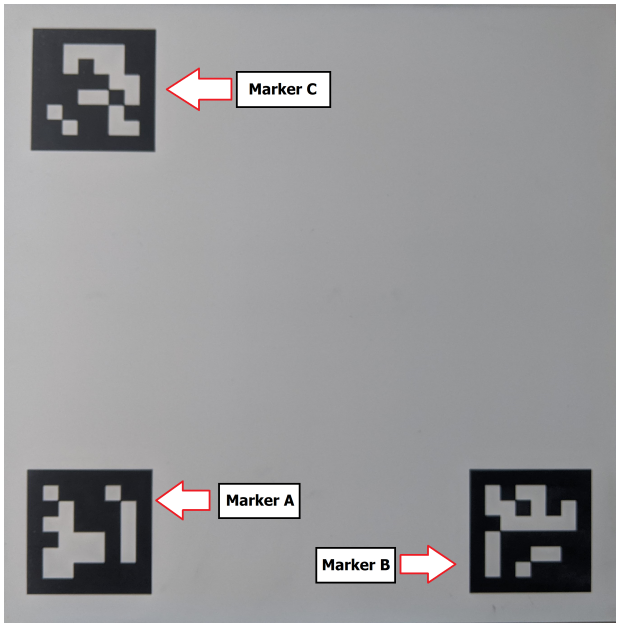


Figure 4.3: Calibration plate with fiducial markers engraved. This plate was made from TroLase White/Black plastic. The markers are labeled A, B, and C.

desired pattern onto the covered areas, which bonds the spray to the aluminum sheet. Once the engraving is complete, the rest of the spray is washed and wiped away, leaving only the areas that were "engraved." This created the appearance of the markers, with the black areas being the metal marking spray bonded to the sheet and the normally white areas being the aluminum plate. For the TroLase White/Black plastic, the black sections of the marker were engraved, uncovering the black layer of plastic under the white layer, which gave a similar appearance to printing the markers on a paper printer, shown in Figure 4.3.

Three fiducial markers were engraved on the sheet, which was positioned to create two perpendicular axes parallel to the edges of the plate. An individual marker could be used to designate up to five points, the four corners and the marker's center. However, three markers were used to define the three points on the plate to define the X and Y axes of the coordinates system of the plate. The points were defined as the center of each marker. Defining each point with its own marker allows the markers to be placed at any distance apart, so long as the X and Y axes created were perpendicular to each other on the face

of the calibration plate. The flexible distance could allow the design of the calibration plate to be adjusted for different sizes of machines while also benefiting from having any error in the alignment of axes being more noticeable due to the larger vectors created. The markers' distance to the plate's edge is also known.

4.1.3 Application of Fiducial Markers

Stereo depth cameras can benefit from using the colour images they rely on to determine depth, for other traditional image processing. One application is finding fiducial markers (QR codes). By designing a plate that can be aligned inside of the machine, markers can be precisely added to the plate to create three points at known offsets and orientations from each other to symbolize the origin of the plate and two other points. The markers are used to define two vectors on the surface of the plate, representing the X and Y axis of the plate. These markers allow the stereo depth camera to capture a 2D image of the markers, and the marker's pixel position within the 2D image can be determined using fiducial pose estimation algorithms.

The marker's pixel position can be used to determine the position of the markers within 3D space in the camera's field of view. The three marker positions can then be used to make two 3D vectors for the X and Y axes of the plate. The cross product of these axes is the Z axis of the plate. Together, these three axes, shown in Figure 4.4, are found in the camera's coordinate system and can be compared to the desired axes found in the plate's coordinate system. This comparison results in a transformation matrix that defines a rough transformation between the stereo camera's coordinate system to the coordinate system of the plate. This transformation can enable the point clouds produced to be accurate within a few centimetres of the real world, as measured by the CMM. This accuracy is many orders of magnitude worse than Intel's minimum object detection claim for the camera of 0.1mm. Therefore, further tuning is required. However, this initial rough tuning is useful to crop the point cloud roughly around the desired area, and the cropped area can be used to further tune the alignment.

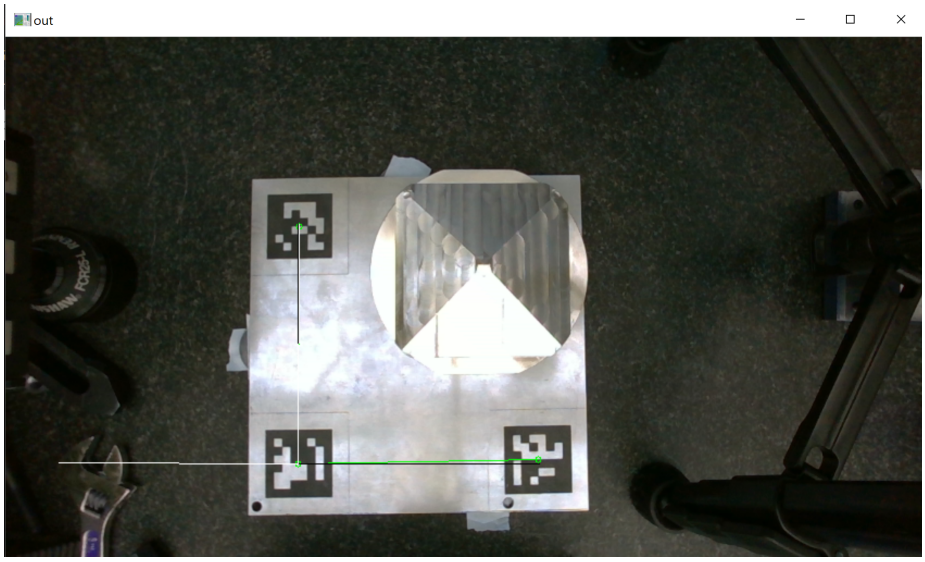


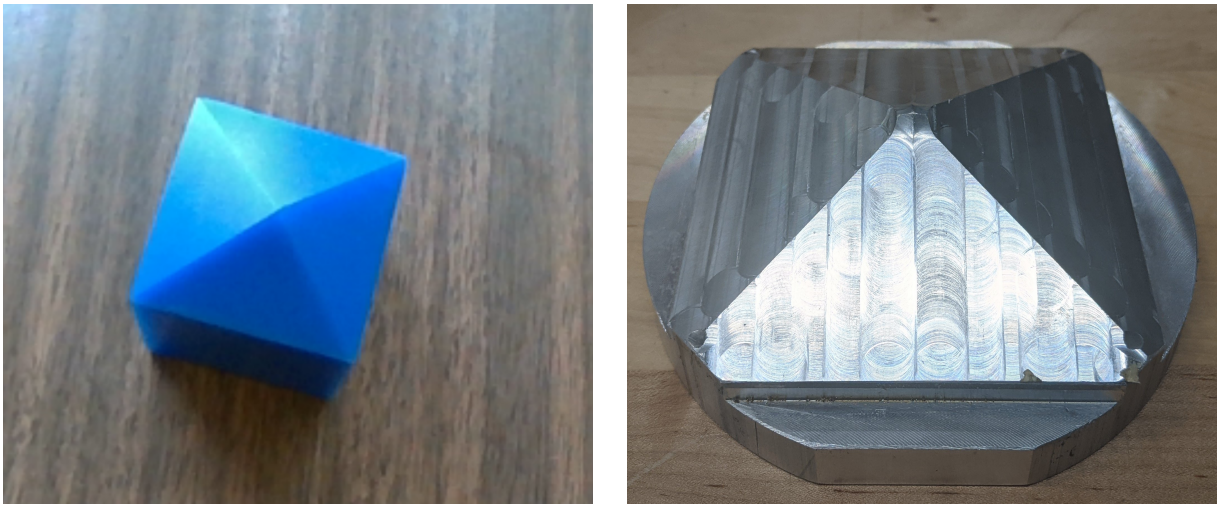
Figure 4.4: Stereo depth camera colour view with ARuCO marker detection for camera calibration

4.1.4 Design of Calibration Pyramid

To get better results, a calibration pyramid is used to improve upon the calibration results found after using the calibration plate. Two pyramids were used during testing. One pyramid was 10cm×10cm with a 2cm apex height, machined from aluminum shown in Figure 4.5a. The second was 5cm×5cm with a 20-degree incline for the faces, machined from a blue Ferris File-A-Wax shown in Figure 4.5b. Multiple point cloud scans are taken by the depth camera and are modified by the positional and rotational offsets found using the calibration plate. These point clouds were used with an algorithm that iteratively determined the equations of the four planes.

4.1.5 Fine-Tuning Calibration Algorithm

This algorithm first split the points into four groups based on their position relative to the desired position of the pyramid faces. Four planes were calculated based on the points using a least square fit. The distance from each point to each of the four planes was found,



(a) (b)

Figure 4.5: (a) Blue Ferris File-A-Wax Pyramid. (b) Aluminum Pyramid

and the points were redistributed among the four groups based on which plane they were nearest. This process was repeated until the size of the groups did not change or after around 50 iterations. Once the groups and points were established and their planes were calculated, the pyramid's apex was found using the pyramid's geometry, where the four planes overlap should be the apex.

However, due to both the noise in the scans and errors depending on the position of the pyramid within the camera's view, the planes did not always overlap where they should. To improve the alignment, the previously calculated apex was compared with the desired apex and the points were offset by the difference. The four groups of points were then rotated around the Z axis passing through the apex to put them all on the same face of the pyramid. All of the points together were used to calculate a new plane. By using all points together to calculate one plane that could be rotated to be used for all faces of the pyramid, the error from the sides further from the center of the camera's view and noise in the point cloud would be reduced, resulting in a better representation of the pyramid. Additionally, the new plane was compared with the actual plane of the pyramid face to determine two rotational offsets: the first rotated the plane around the Z axis and the

second around the Y axis.

The points were then rotated based on the two angles found, and then the plane was rotated around the Z axis passing through the apex to form all four faces. The apex was recalculated based on these faces. The offset of the points to align with the desired apex and the two angles found while aligning the face of the pyramid were used further to calibrate the stereo depth camera's point clouds. This fine-tuning calibration improved the accuracy of the point clouds produced from within centimetres to within millimetres of the real-world part.

This method is only one of many possible methods that could have been used to align the point cloud with the desired model of the part. Some examples include mounting the stereo depth camera at a precisely known location or attaching the camera to the gantry of the CNC or CMM and probing for locating spheres to inform the camera's location. Precisely mounting the camera was not chosen to decrease the effort and precision of the setup required from users. The fiducial marker method was also chosen to decrease the need for skilled knowledge required to calibrate using a probe and allow the camera to be mounted anywhere. Additionally, although not attempted during this research project, the markers might allow more potential for quickly aligning the point clouds from multiple stereo depth cameras placed at different positions around the part, which could be used to better represent the entire part from various angles.

Existing point registration algorithms could have also been used to determine the offset and rotation of the calibration pyramid. These algorithms were considered earlier in the research project, but were initially not chosen due to some issues the method had with symmetric parts, and parts with large flat surfaces. However, towards the end of the project, some experiments were done using Geomagic Design X, which could better align the point clouds with a desired model, and even attempt to clean them up and reduce noise in the data. The results of the experiments and how they were conducted are further explored in Section 5.3.4. Using Geomagic may be a better choice for the future.

4.1.6 Camera Calibration Objective

The fiducial markers on the plate are used to create the coordinate system of the plate, which is used to determine the transformation matrix for the stereo depth camera's coordinate system.

The three markers are located a known distance from the edge of the plate and each other. The first marker denotes the origin point in the bottom left corner of the plate. The second marker is found in the bottom right corner of the plate. Finally, the third marker is located in the top left corner of the plate. The marker's pixel locations are found within the image and then converted to their XYZ location in the camera coordinate system. The X axis of the plate is then defined by creating a vector from the origin marker to the second marker. Similarly, the Y axis is defined by the vector created between the origin marker and the third marker. The cross product of the X and Y axis vectors determines the Z axis. The Z unit vector of the plate in the plate's coordinate system is defined as $[0, 0, 1]$, and an axis-angle method is used to determine an axis that the Z axis in the camera's coordinate system can be rotated by a determined angle to align it with the Z axis in the plate's coordinate system. Once the Z axes are aligned, the X and Y axes are rotated around the Z axis to align the X axis in the camera's coordinate system with the desired X axis in the plate's coordinate system of $[1, 0, 0]$. This rotation is done while also attempting to reduce the error of aligning the Y axis of the camera coordinate system with the Y axis in the plate coordinate system.

The axis-angle combinations are then converted to transformation matrices, which are combined. Additionally, the translations required from the camera origin to the origin marker on the plate, and the translations to the edge of the plate, define the transformation matrix from the camera's coordinate system to the plate coordinate system along the edge of the plate. So long as the camera position is not changed, this transformation can be applied to future point clouds created by the stereo depth camera.

4.2 Scanning and Alignment

The scanning and alignment objective is to determine the geometry and location of the pyramid part within the CNC or CMM workspace. To achieve the objective the algorithm was split into two separate steps, one specifically for scanning, a rough alignment, and cropping the point cloud and another specifically for aligning the point cloud with the desired pyramid model and outputting results. The scanning algorithm is used to scan the pyramid part and roughly align it using the fiducial markers on the calibration plate, similar to the method used by Madeira *et al.* [9]. The alignment algorithm fine-tunes the alignment based on the planes fitted to the scanned points and the apex that is calculated based on the planes. The two algorithms are detailed below.

4.2.1 Scanning Algorithm

The first step of the scanning algorithm must locate the fiducial markers on the calibrated plate. OpenCV's [16] ArUco module, which is based on the ArUco library developed by Garrido *et al.* [10], was used. The markers are located within the 2D image from the stereo depth. Each marker's corners are given by their pixel position within the image. The pixel position of the center of each marker is calculated by

$$O_p = \frac{\sum_{i=1}^{4} C_i}{4} \tag{4.1}$$

where C_i is the pixel position of each corner of the marker given as $[u, v]$.

The pixel position needs to be converted to a position in 3D space. This is done using the RealSense library [17], which deprojects a pixel to a point in 3D space using the camera's intrinsics and the distance to the point represented by the pixel. The distance to the point represented by the pixel is also found using another included function from the RealSense library. The pixel is then deprojected from its pixel position $[u, v]$ to a 3D points $[x, y, z]$.

With all marker positions known in 3D space, two vectors, $\vec{V}_{AB}$ and $\vec{V}_{AC}$ are created from markers A-B and A-C, shown in Figure 4.3 by

$$\vec{V}_{AB} = \begin{bmatrix} B_x - A_x & B_y - A_y & B_z - A_z \end{bmatrix} \tag{4.2}$$

and

$$\vec{V}_{AC} = \begin{bmatrix} C_x - A_x & C_y - A_y & C_z - A_z \end{bmatrix} \tag{4.3}$$

where marker 1 is A, marker 2 is B and marker 3 is C.

Vector $\vec{V}_{AB}$ represents the X axis of the plate, and vector $\vec{V}_{AC}$ represents the Y axis. These vectors are normalized as unit vectors. A vector representing the Z axis is calculated by taking the cross-product of the two unit vectors.

$$\hat{Z}_{calc} = \frac{\vec{V}_{AB} \times \vec{V}_{AC}}{|\vec{V}_{AB} \times \vec{V}_{AC}|} \tag{4.4}$$

These vectors are compared with normalized vectors for the X axis $\hat{X}_A = [1, 0, 0]$, Y axis $\hat{Y}_A = [0, 1, 0]$ and Z axis $\hat{Z}_A = [0, 0, 1]$ to align the camera's coordinate system with the calibration plate's coordinate system. An axis-angle method is used for this alignment. First, the Z axis is aligned by taking the cross-product of the calculated Z axis unit vector and the desired Z axis unit vector to find the axis of rotation to align them $\widehat{RAxis}_z$.

$$\widehat{RAxis}_z = \frac{\hat{Z}_{calc} \times \hat{Z}_A}{|\hat{Z}_{calc} \times \hat{Z}_A|} \tag{4.5}$$

The angle between the axes was calculated by taking the dot-product of the calculated Z axis and the desired Z axis, $RAngle_z$.

$$RAng_z = \hat{Z}_{calc} \cdot \hat{Z}_A \tag{4.6}$$

Using the axis-angle calculated, $\hat{V}_{AB}$ was rotated using the form

$$\begin{aligned} \vec{V}_{AB} = &\cos(RAng_z)\hat{V}_{AB} + \\ &\sin(RAng_z)(\widehat{RAxis}_z \times \hat{V}_{AB}) + \\ &(1 - \cos(RAng_z))(\widehat{RAxis}_z \times \hat{V}_{AB})\widehat{RAxis}_z \end{aligned} \tag{4.7}$$

$$\hat{V}_{AB} = \frac{\vec{V}_{AB}}{|\vec{V}_{AB}|} \tag{4.8}$$

The new $\hat{V}_{AC}$ can be calculated as the cross between the now aligned Z axis and the rotated $\hat{V}_{AB}$.

$$\vec{V}_{AC} = \hat{Z}_A \times \hat{V}_{AB} \tag{4.9}$$

$$\hat{V}_{AC} = \frac{\vec{V}_{AC}}{|\vec{V}_{AC}|} \tag{4.10}$$

The cross product and dot product of the new $\hat{V}_{AC}$ and the desired Y axis is taken to determine the axis-angle to align the Y axis in the same way the Z axis was aligned.

$$\overrightarrow{RAxis}_y = \hat{V}_{AC} \times \hat{Y}_A \tag{4.11}$$

$$\widehat{RAxis}_y = \frac{\overrightarrow{RAxis}_y}{|\overrightarrow{RAxis}_y|} \tag{4.12}$$

$$RAng_y = \hat{V}_{AC} \cdot \hat{Y}_A \tag{4.13}$$

Using the two axis-angle combinations found, two rotation matrices representing the transformations R_z and R_y can be created in the same form [18]

$$R = \begin{bmatrix} \cos\theta + u_x^2(1-\cos\theta) & u_x u_y(1-\cos\theta) - u_z\sin\theta & u_x u_z(1-\cos\theta) + u_y\sin\theta \\ u_x u_y(1-\cos\theta) + u_z\sin\theta & \cos\theta + u_y^2(1-\cos\theta) & u_y u_z(1-\cos\theta) + u_x\sin\theta \\ u_x u_z(1-\cos\theta) + u_y\sin\theta & u_y u_z(1-\cos\theta) + u_x\sin\theta & \cos\theta + u_z^2(1-\cos\theta) \end{bmatrix} \tag{4.14}$$

where θ is either $RAng_z$ or $RAng_y$, u is $\widehat{RAxis}_z$ or $\widehat{RAxis}_y$ and $\hat{u}$ is a unit vector with $u_x^2 + u_y^2 + u_z^2 = 1$.

The two resulting rotation matrices are multiplied to determine the overall rotation matrix from the camera's coordinate system to the calibration plate's coordinate system.

$$R = R_y \times R_z \tag{4.15}$$

This rotation matrix is combined with known distances from the fiducial markers to the edge of the plate $[T_x, T_y, T_z]$ to create the transformation matrix between the camera's coordinate system to the calibration plate edge and is used on all future scans taken with

the camera in the same position.

$$T_m = \begin{bmatrix} R_{xx} & R_{xy} & R_{xz} & T_x \\ R_{yx} & R_{yy} & R_{yz} & T_y \\ R_{zx} & R_{zy} & R_{zz} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.16}$$

Finally, point cloud scans can be taken of the pyramid part. The scanning was done using a slightly modified version of the point cloud sample program provided with the RealSense SDK [17]. The RealSense library includes functions to capture the point cloud in RealSense's `points` data type. The point cloud is in the camera's coordinate system, which is then transformed to align with the calibration plate's coordinate system based on the transformation matrix from Equation 4.16. The point cloud library (PCL) [19] was used to apply the transformations, and later for outputting the point clouds using PCL's point cloud data (`.pcd`) file format. The RealSense `points` data type is converted to PCL's `cloudPointer` data type. Then PCL's `transformPointCloud` function could apply the transformation matrix to the scanned point cloud.

The transformed point cloud has not been cropped and may include points outside the pyramid part. To crop the point cloud around the pyramid, OpenCV [16] was used again to take advantage of their thresholding capabilities, specifically, `inRange` and `findContours`. The `inRange` function creates a thresholding mask based on a range of pixel values in the HSV colorspace. The blue pyramid can be detected by `inRange` and `findContours` can be used to create curves around the detected blue areas. A bounding rectangle is calculated around the curves, the area where the pyramid is located, shown in Figure 4.6. Now, the pixel positions of the corners can be found in 3D space using the same techniques used to find the locations of the fiducial markers. Due to the camera's position above the pyramid, the corners can be assumed to represent the maximum and minimum range of X and Y values of the pyramid. The corner positions also can be used to estimate where the apex of the pyramid is, and the location of the apex in 3D space is calculated, which is the maximum Z value of the pyramid. By assuming the pyramid height, the minimum Z value can be calculated by subtracting the pyramid height from the apex height. These X, Y, and Z ranges are used to crop the overall point clouds roughly around the pyramid part.

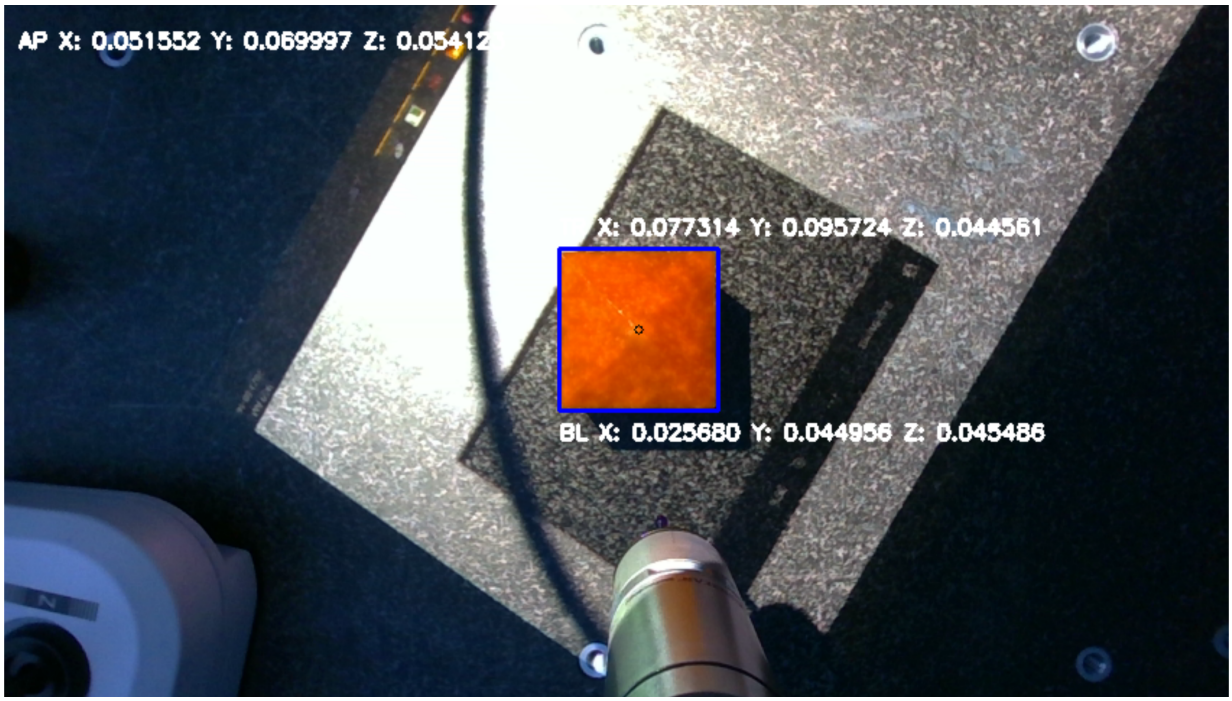


Figure 4.6: The point clouds are cropped by determining the location of the pyramid using OpenCV image processing and thresholding. The pyramid appears orange in the image due to the Red and Blue channels being swapped in the output display.

The final transformed and cropped point cloud of the pyramid part is output as either a (`.pcd`) for further modification by the Point Cloud Library or a (`.csv`) file for the fine-tuning alignment algorithm.

The pseudo-code for the scanning algorithm is presented in Figure 4.7.

4.2.2 Fine-Tuning Alignment Algorithm

The fine-tuning alignment algorithm starts by taking the roughly aligned point cloud points from the scanning algorithm and splits the points into four groups, one for each face of the pyramid. The split is done by using the same bounding box from the scanning algorithm, and assuming that the corners of the pyramid are roughly the same as the bounding box corners. Each corner is in the form $P = [P_x, P_y, P_z]$ and the apex is in the form $A = [A_x, A_y, A_z]$. Each face can then be represented by a function based on the two corners and apex position in the form

$$\begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{bmatrix} (1-v)((1-u)P_x^1 + u(P_x^2)) + v(A_x) \\ (1-v)((1-u)P_y^1 + u(P_y^2)) + v(A_y) \\ (1-v)((1-u)P_z^1 + u(P_z^2)) + v(A_z) \end{bmatrix} \tag{4.17}$$

where P^1 and P^2 are the two corners of the face, A is the apex, and u and v range from 0 to 1.

Equation 4.17 can be rearranged and solved for u and v in the form

$$u = \frac{P_x^1 A_y - P_x^1 S_y - P_y^1 A_x + P_y^1 S_x + A_x S_y - A_y S_x}{P_x^1 A_y - P_x^1 S_y - P_y^1 A_x + P_y^1 S_x - P_x^2 A_y + P_x^2 S_y + P_y^2 A_x - P_y^2 S_x} \tag{4.18}$$

$$v = \frac{P_x^1 P_y^2 - P_x^1 S_y - P_y^1 P_x^2 + P_y^1 S_x + P_x^2 S_y - P_y^2 S_x}{P_x^1 P_y^2 - P_x^1 A_y - P_y^1 P_x^2 + P_y^1 A_x + P_x^2 A_y - P_y^2 A_x} \tag{4.19}$$

where S_x and S_y are the x and y values of each point. If the point is in the pyramid face, u and v will be between 0 and 1, and if the point is outside of the pyramid face, one or both of u and v will be outside the 0 to 1 range. These equations are made for each face, and every point is tested on each face to determine which one it likely belongs to. Points not found to be in the XY area of the faces are discarded.

1.0 Locate the pixel position of the fiducial markers within the 2D image using OpenCV ArUco library: O_{pA}, O_{pB}, O_{pC}.

2.0 Convert the pixel positions of the markers into 3D space using RealSense library.

3.0 Use marker 3D positions to create vectors $\vec{V}_{AB}$ and $\vec{V}_{AC}$ from $\vec{V}_{AB} = [B_x - A_x, B_y - A_y, B_z - A_z]$ and $\vec{V}_{AC} = [C_x - A_x, C_y - A_y, C_z - A_z]$. Normalize $\vec{V}_{AB}$ and $\vec{V}_{AC}$ into unit vectors $\hat{V}_{AB}$ and $\hat{V}_{AC}$.

4.0 Determine $\hat{Z}_{calc}$ from normalizing $\vec{Z}_{calc} = \hat{V}_{AB} \times \hat{V}_{AC}$.

5.0 Determine axis-angles to align found coordinate system defined by $\hat{V}_{AB}$ for X axis, $\hat{V}_{AC}$ for Y axis, and $\hat{Z}_{calc}$ for Z axis with unit vectors of desired axes using method shown by Equations 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, 4.13

6.0 Convert axis-angle to rotation matrices using Equation 4.14.

7.0 Combine rotation matrices using Equation 4.15 and combine with translations to define transformation matrix in Equation 4.16

8.0 Capture point cloud of pyramid part using RealSense's functions

9.0 Convert point cloud to PCL format to use PCL function to transform points based on the transformation matrix from step 8.0.

10.0 Crop point cloud by determining the area where the pyramid exists within the point cloud using OpenCV functions.

11.0 Output transformed and cropped point cloud as either `.pcd` or `.csv` for use in fine-tuning alignment algorithm.

Figure 4.7: Pseudo-code for scanning algorithm for getting transformed and cropped point cloud for use in fine-tuning alignment algorithm.

Once the points are roughly sorted into one of the four faces, a least square fit is used to calculate the equation of a plane that best fits the points. Matlab's `lsqr` function [20] was used to solve the system of linear equations in the form

$$N = Px \tag{4.20}$$

where P is an array of all of the points in the current face being calculated in the form $[p_x, p_y, p_z]$, and N is an array with the same length as the number of points in P of $N = [-1]$. The equation of the plane is $[A, B, C]$, which corresponds to the equation of the plane

$$Ax + By + Cz + D = 0 \tag{4.21}$$

where $D = 1$.

Once the four planes are calculated, the points are resorted based on which plane they are nearest to. Their distance from the plane is calculated by Equation 4.22.

$$Distance = \frac{|Ax + By + Cz + D|}{\sqrt{A^2 + B^2 + C^2}} \tag{4.22}$$

where A, B, C are from the equation of the calculated plane, and $D = 1$ The points are moved to the group corresponding to the plane that they are nearest to. The process of fitting the planes by Equation 4.20 and resorting the points based on with plane they are nearest to by Equation 4.22 is repeated up to 50 times or until the size of each group does not change between iterations. Typically the resorting only occurs around 5-10 times due to the earlier rough sorting by Equations 4.18 and 4.19.

After the equations of the four planes have been calculated, the apex of the pyramid that would fit the four planes is calculated using the equation

$$[ap] = [[AA]^T[AA]]^{-1}[AA]^T[BB] \tag{4.23}$$

where

$$[AA] = \begin{bmatrix} P1_x & P1_y & P1_z \\ P2_x & P2_y & P2_z \\ P3_x & P3_y & P3_z \\ P4_x & P4_y & P4_z \end{bmatrix} \tag{4.24}$$

$$[BB] = \begin{bmatrix} \frac{-1}{\sqrt{P1_x^2+P1_y^2+P1_z^2}} \\ \frac{-1}{\sqrt{P2_x^2+P2_y^2+P2_z^2}} \\ \frac{-1}{\sqrt{P3_x^2+P3_y^2+P3_z^2}} \\ \frac{-1}{\sqrt{P4_x^2+P4_y^2+P4_z^2}} \end{bmatrix}. \tag{4.25}$$

Equation 4.23 is based on solving a system of Equation 4.21 where $D = 1$. The apex lies on all four planes, whose equations are combined into the matrix from Equation 4.24. However, the equations of the planes must be normalized by dividing by $\sqrt{A^2 + B^2 + C^2}$, which results in

$$\hat{A}x + \hat{B}y + \hat{C}z + \frac{1}{\sqrt{A^2 + B^2 + C^2}} = 0. \tag{4.26}$$

The D portion for all equations is rearranged to the other side of Equation 4.21 to become the array shown by Equation 4.25. Matrices from Equations 4.24 and 4.25 together create

$$[AA][ap] = [BB]. \tag{4.27}$$

However, before being able to solve for matrix $[ap]$, since there are four planes, matrix $[AA]$ is a 4×3 matrix which can not be inverted. To make it into a square matrix that can be used, both sides of the equation are multiplied by the transpose of matrix $[AA]$.

$$[AA]^T[AA][ap] = [AA]^T[BB] \tag{4.28}$$

Now both sides of the equation can be multiplied by the inverse of $[AA]^T[AA]$.

$$[[AA]^T[AA]]^{-1}[[AA]^T[AA]][ap] = [[AA]^T[AA]]^{-1}[AA]^T[BB] \tag{4.29}$$

The inverse of a matrix multiplied by the matrix cancels into an identity matrix, resulting in Equation 4.23, which calculates the apex from the four planes.

The location of the apex is used to move all of the points in the point cloud such that the apex will be move to $ap = [0, 0, ap_{zd}]$, where ap_{zd} is the desired height of the apex. By moving the pyramid such that the apex is centred at the origin at the height of the desired pyramid apex, the next steps in the fine-tuning alignment algorithm can take advantage of the pyramid's symmetry.

The equations of the four planes must be recomputed after the points have been moved using the same least square fit in Equation 4.20. The apex must also be recalculated using Equation 4.23 to verify that it is located at or around $ap = [0, 0, ap_{zd}]$.

After all of the previous calculations to determine the planes, these planes should be close to aligning with each other to represent the pyramid model. However, often the planes are not perfectly aligned due to noise or issues during the scanning process. To further improve the accuracy of the planes, the symmetry of the pyramid is used. With the apex at $ap = [0, 0, ap_{zd}]$, and since all faces of the pyramid model are the same size and shape, all of the points and planes are rotated about the Z axis running through the apex so that they are on the right $(+X)$ face of the pyramid. The symmetry allows this rotation to be completed by swapping the X and Y values or from negative to positive values to complete this rotation. To move the top $(+Y)$ face, the points becomes

$$P = \begin{bmatrix} P_y & -P_x & P_z \end{bmatrix} \tag{4.30}$$

To moved the left $(-X)$ face, the points become

$$P = \begin{bmatrix} -P_x & -P_y & Pz \end{bmatrix} \tag{4.31}$$

To move the bottom $(-Y)$ face, the points become

$$P = \begin{bmatrix} -P_y & P_x & P_z \end{bmatrix} \tag{4.32}$$

These three equations can be used to move all of the points from the top, left and bottom faces to the same right face. All points are then combined into one group, and the same least square fit equation 4.20 is used to calculate one plane P_o that, when eventually rotated back onto the other three faces, will represent the four faces of the pyramid.

However, before the faces can be rotated back, the equation of the plane is compared with the desired equation of the right face of the pyramid model to determine any rotations or tilts that are applied to the pyramid. The rotation about the Z axis is calculated as

$$\theta_z = -\arctan \frac{P_{oy}}{P_{ox}}. \tag{4.33}$$

The plane P_o is rotated around the Z axis by θ_z by multiplying it by the rotation matrix from angle θ_z. The rotation matrix is

$$R_{\theta_z} = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.34}$$

The rotated plane is compared to the desired plane $P_d = [-\sin\beta_d, 0, -\cos\beta_d]$, where β_d is the angle of the desired plane with the XY plane, to determine the rotation about the Y axis as

$$\alpha_y = -\arccos P_o \cdot P_d. \tag{4.35}$$

The rotation matrix for the rotation about the Y axis is

$$R_{\theta_z} = \begin{bmatrix} \cos\alpha_y & 0 & -\sin\alpha_y \\ 0 & 1 & 0 \\ \sin\alpha_y & 0 & \cos\alpha_y \end{bmatrix}. \tag{4.36}$$

When both rotation matrices are calculated, the rotation matrices multiply all the of the groups of points before they were rotated onto the same face, and the same is done for the single group of points after all of the faces were rotated onto the same face.

$$P = PR_{\theta_z}R_{\alpha_y} \tag{4.37}$$

The one group of all points, which was initially created by rotating the points from the other three faces onto the right face, is rotated by the rotation matrix from Equation 4.37 to better aligned with the desired plane of the pyramid. The points are then once again used to fit a plane using Equation 4.20.

With this final plane, the plane is copied and rotated back to each side of the pyramid, essentially using a reverse of Equations 4.30, 4.31, and 4.32 to create four planes that align edge to edge with each other and at the apex. The four planes are used to recalculate the apex using Equation 4.23. Additionally, the positions of the corners of the pyramid are calculated. Equation 4.38 shows the calculation for the position of the top-right corner and, due to symmetry, all other corners can be found by flipping the sign of the components as in Equations 4.39, 4.40, and 4.41.

$$C_{tr} = \begin{bmatrix} \frac{(P_y - P_x)}{(P_x^2 + P_y^2)} & -\frac{(P_y + P_x)}{(P_x^2 + P_y^2)} & 0 \end{bmatrix} \tag{4.38}$$

$$C_{tl} = \begin{bmatrix} -C_{tr_x} & C_{tr_y} & 0 \end{bmatrix} \tag{4.39}$$

$$C_{bl} = \begin{bmatrix} -C_{tr_x} & -C_{tr_y} & 0 \end{bmatrix} \tag{4.40}$$

$$C_{br} = \begin{bmatrix} C_{tr_x} & -C_{tr_y} & 0 \end{bmatrix} \tag{4.41}$$

As a final metric, the distance between the points and the desired pyramid planes is calculated using an equation similar to Equation 5.1. However, the equation of the plane defined by $[A, B, C, D]$ is the equation of the planes from the desired pyramid model, which is validated by the CMM.

4.3 Key Inspection Metrics

To determine the viability of the stereo camera and algorithms used, two key inspection metrics were chosen. These were part position and part geometry. The part position encompasses how well the location and any rotations applied to the part can be determined, while part geometry includes how well key geometric features of the part can be found and how well the points fit to the desired surface. The key inspection metrics are shown in Figure 4.8, which is a top view of the calibration plate and pyramid on the CMM.

Since a pyramid was used for testing, part position will be determined by the location of the pyramid's apex and any rotation around the Z axis running normal to the bottom of the pyramid through the apex. Part geometry will be determined by the location of the apex and corners of the pyramid and the point fitting with the surface of the pyramid.

Due to the expected noise found in the scans, five or more scans will be taken, and the results will be averaged to determine a more accurate result. Additionally, these scans can be used to determine the precision of the metrics.

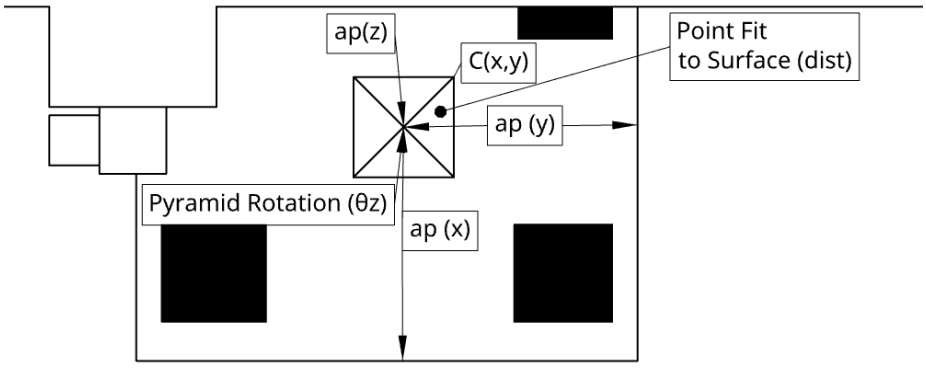


Figure 4.8: Top view sketch of key inspection metrics on the CMM. The part position is determined by $ap(x)$, $ap(y)$, $ap(z)$ and pyramid rotation θ_z. The part geometry is determined by $ap(z)$, $C(x, y)$, and the point fit to surface distance.

Chapter 5

Results and Discussion

This chapter will show and discuss the results of the tests performed by the implemented systems and algorithms. Additionally, the proposed system and algorithms will also be compared with existing software, Geomagic. The previous chapters discussed the method developed for mapping from the camera coordinate system to the calibration plate coordinate system to the part coordinate system. In Section 3.2 and throughout Chapter 4, I tested and verified the mapping from camera's coordination system to both the calibration plate's coordinate system and pyramid part coordinate system using a CMM. Rather than directly validating the other coordinate system mappings, the method is verified by testing the mappings of the point clouds created on the camera and their mapping to the part and part features.

5.1 Calibration Sphere Results

Early experiments with the stereo depth camera and similar alignment algorithms used a cream-coloured calibration sphere that was somewhat reflective due to its surface finish. The sphere had a diameter of 2cm across and was mounted on a peg that was screwed into the table of the CMM. Instead of the least squares fit algorithm for finding the surface described in Section 4.2.2, another similar fit was used to determine a best-fitting sphere.

The location of the center of the sphere, as well as its diameter, were found and used for making conclusions. Although the results from these tests did not use the same test setup and algorithms as the final tests with the pyramid part, the calibration sphere tests did apply the calibration plate. The results of these early tests informed much of the decisions that led to the final algorithms and methodology. The tests demonstrated the effects of using certain filters to attempt to improve the resulting scans, in addition to the importance of the camera position and part position within the camera's field of view for getting better results.

First, the RealSense SDK included a few different types of filters [21] that could be applied to the scans in an attempt to improve the results by reducing noise and holes. The filters experimented with include spatial, decimation, and hole filling. Additionally, to properly apply the spatial filter, two additional filters were required to transform the data from the depth domain to the disparity domain and then back again to the depth domain after the spatial filter had been applied.

When using these filters with the calibration sphere, the results became worse overall. In tests where the location of the camera and sphere did not change, and no other aspects were adjusted, scans with no filters returned results that found the sphere closer to where it was compared with scans taken when filters were applied.

The camera and part position also greatly impacted the results of the scans. The changing the camera's distance from the part showed that as the camera moved further away, the results worsened, often due to more noise. These findings intuitively match what could be expected, as moving the camera further from the scene causes larger areas to be represented by each pixel, resulting in more data being compressed into the same-sized pixel. However, the results also worsen if the camera is too close to the part due to the ideal operating range of the camera and the lens. Additionally, if the camera is brought too close, the camera can not view the entire part, which makes comparisons more challenging. These findings were further expanded upon once the scanned part was switched to the pyramid.

Although these tests with the calibration sphere did not provide acceptable results, their results informed the decisions and methodology that were made moving forward.

The tests with the calibration sphere showed that pursuing the use of filters did not seem as worthwhile, and instead, other methods should be used to improve the point cloud scans. Additionally, it was found that the sphere was challenging for the camera to scan due to the geometry and surface finish. However, since the ideal application of this research would be used during the machining of potentially reflective materials, the issue of material reflections had to be addressed. This led to using the pyramid parts made from aluminum to show that the camera could handle reflective surfaces, and the plastic pyramid to simplify the geometry and surface finish.

5.2 Implementation

The tests and lessons learned from working with the calibration sphere led to the final algorithms and methodology used during the rest of the tests. The algorithms were implemented using C++ to capture the point clouds and Matlab to process the resulting point clouds. Although the methods developed were then explicitly created for pyramid-shaped parts, the algorithms can be generalized for other designs by performing multiple least square fits for the points on the desired faces of the part. Using a pyramid-shaped part allowed the algorithm to take advantage of the symmetries of a pyramid to further enhance the resulting surfaces that were fitted. The stereo depth camera used was the Intel RealSense D405, which has advertised object detection of 0.1mm at 7cm [22] in ideal conditions.

The algorithms were tested on two different pyramid-shaped parts. One was manufactured from a round aluminum stock that cut off the pyramid's corners. The aluminum pyramid had a span of 100×100 mm from the corners and a height of 20mm at the pyramid's apex. The aluminum pyramid was from a previous student's work and had a few undesired scallops on the surface from machining. The second pyramid was manufactured from a cube stock of blue Ferris File-A-Wax. The wax pyramid span was 50×50mm from the corners. The height of the apex was determined by the angle of the faces, which were 20 degrees from the bottom plane, resulting in a height of around 9.1mm. Both calibration plates previously mentioned were used for different tests.

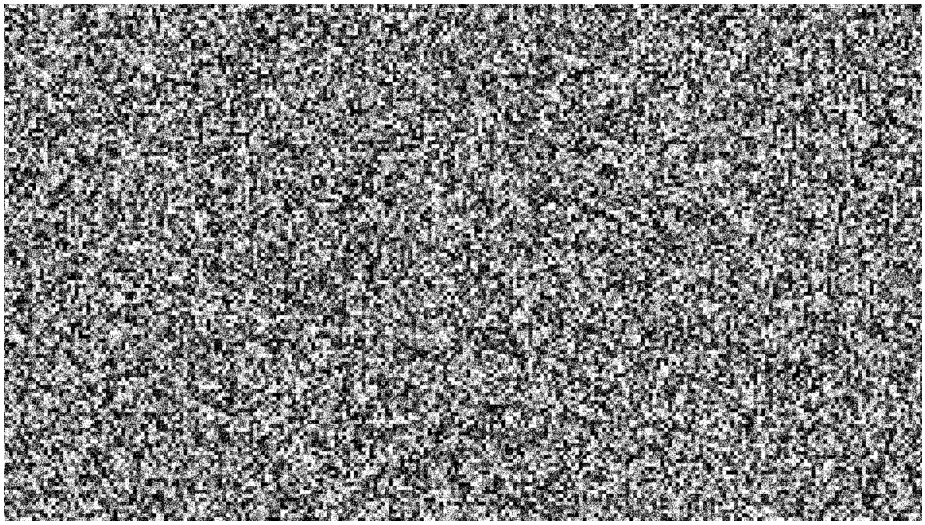


Figure 5.1: Semi-random pattern projected on less-textured objects to improve point cloud quality.

A consumer-grade projector, an Espon LCD Projector H550A, was used to project a semi-random pattern that avoided periodic arrangements of dots while also providing better lighting for the scene. Some tests were completed without the pattern to observe the improvements when the pattern was projected. The pattern was obtained from Intel RealSense documentation [23] and is shown in Figure 5.1.

5.3 Pyramid Results

5.3.1 Lighting and Pattern

The use of the pattern projected on any part to be scanned and proper lighting were found to significantly impact the quality and accuracy of the scans produced. Three separate tests were performed, with the part positioned in the center of the camera's field of view. The camera and part positions were not varied between tests. Each test consisted of five scans that were taken and averaged. In the first test, scans were taken with normal room lighting. The second test used the projector to light the part with a white-coloured light.

Room	x	y	z		Projector	x	y	z		Pattern	X	Y	Z
Avg Offset	-0.2429	-1.28424	0.996592		Avg Offset	-3.61208	2.17475	1.902708		Avg Offset	0.624623	-0.747	-0.01391
Avg Apex	0	0	-6.26057		Avg Apex	0	0	0.954007		Avg Apex	0	0	0.17807
Avg Corner	14.0063	14.0063	0		Avg Corner	2.623158	2.623158	0		Avg Corner	0.491287	0.491287	0

Table 5.1: Deviation (in mm) of Pyramid Offset, Apex Location, and Corner Location in an ideal location within the stereo depth camera's field of view but with different lighting and texturing conditions.

The final test projected the aforementioned pattern onto the surface of the pyramid. The results found are shown in Table 5.1.

With average room lighting, the algorithm was modified to not redistribute the points depending on which face they were nearest in an attempt to fit the points to the planes better. When the redistribution was left in, the resulting planes converged upon included points from planes they were not a part of. Instead, the points were assumed to be correctly distributed in the initial distribution of the points given by the expected corner positions. However, even with the modification, the resulting point cloud was noisy, to the point that the results did not represent the physical model of the pyramid.

When the projector was used to light the scene better, the same modification to the algorithm was necessary to get better results. However, unlike the previous test, the results found were much more representative of the actual dimensions of the pyramid. The data was still somewhat noisy due to the pyramid's relatively untextured surface, making it difficult for the correspondence algorithm to find the correct distance for all of the points.

Finally, when the pattern was projected on the pyramid's surface, the results were much closer to the actual part dimensions, with deviation being 5 to 10 times less than just projecting light. The pattern provided an artificial texture to the pyramid, while also lighting the scene much better in the process. Since the pattern used was made from a pixel-like pattern, the ideal size of the pixels should be near matching the stereo depth camera's pixel resolution. If the pattern's pixels are smaller than the pixel resolution of the camera, they will not improve the results of the correspondence problem, and therefore the results of the point cloud scans. However, if the pattern's pixels are larger than the pixel resolution of the camera, the correspondence problem will have more issues precisely

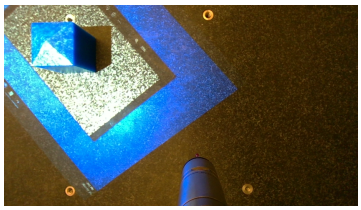
matching the corresponding points within the left and right camera images. This will result in either more noise in the point cloud or incorrect depth values.

The importance of proper lighting is shown by two extremes caused by improper lighting, overexposure and shadows. During some tests with the part positioned along the edges of the camera's field of view, shadows were created on the edges and sides of some of the pyramid's surfaces, while other surfaces were overexposed from the projector light. These can be seen in Table 5.2. The projector was mounted slightly behind the camera, roughly between positions 3-2 and 3-3. Additionally, no other external light sources were used to illuminate the pyramid part more than the room lighting. This positioning may have influenced the results of scans 1-5, 2-5, 3-5, 4-5, and 5-5. These scans have visibly more shadows and a few show faces that appear overexposed. The shadows and overexposure for scans along the right edge of the camera's field of view have some influence on the quality of the point cloud scans produced in these areas which further degrade the geometric and positional results found after the alignment algorithms. Conversely, scans taken with the part closer to the center of the camera's field of view such as position 3-3, benefited from the projector facing directly down upon them with even lighting and minimal shadows.

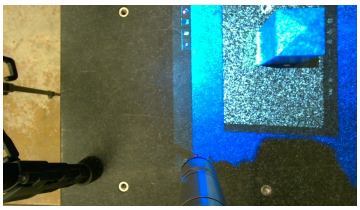
5.3.2 Part Position and Rotation

The part's location within the camera's field of view also significantly affected the results. The camera was mounted on the probe of the CMM, allowing the camera to be moved very accurately. The camera was moved such that the part was scanned in 24 locations across the camera's field of view, from the top left of the field of view to the bottom right. The colour views from the stereo depth camera of a few of the positions are shown in Figure 5.2. Five scans were taken at each location, and the results of the offset, apex height and corner positions were averaged between the five scans to produce the following results in Table 5.2.

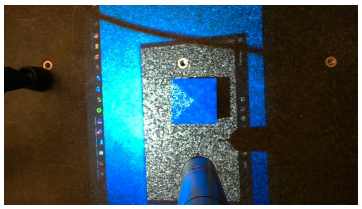
For all tests around the edges of the camera's field of view, the plane fitting algorithm was modified, as mentioned before, to not redistribute the points after the initial distribution of the points given by the expected corner positions. If the redistribution was left in, the resulting planes found for the pyramid faces were incorrect, as shown in Figure 5.3.



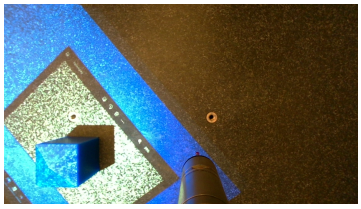
(a) Position 1-1



(b) Position 1-5



(c) Position 3-3



(d) Position 5-1

(e) Position 5-5

Figure 5.2: Colour view from the stereo depth camera. The pattern is projected onto the blue File-A-Wax pyramid.

1-1	x	y	z	1-2	x	y	z	1-3	x	y	z	1-4	x	y	z	1-5	x	y	z
Avg Offset	2.28	1.01	-0.75	Avg Offset	0.09	-1.13	-0.61	Avg Offset	1.96	-0.51	0.46	Avg Offset	1.42	-1.76	0.86	Avg Offset	9.42	1.43	0.14
Avg Apex	0.00	0.00	1.03	Avg Apex	0.00	0.00	1.32	Avg Apex	0.00	0.00	0.80	Avg Apex	0.00	0.00	1.01	Avg Apex	0.00	0.00	-1.75
Avg Corner	2.62	2.62	0.00	Avg Corner	3.52	3.52	0.00	Avg Corner	2.19	2.19	0.00	Avg Corner	2.78	2.78	0.00	Avg Corner	-4.80	4.80	0.00
2-1	x	y	z	2-2	x	y	z	2-3	x	y	z	2-4	x	y	z	2-5	x	y	z
Avg Offset	3.56	-0.43	-0.30	Avg Offset	1.31	1.74	-1.64	Avg Offset	2.30	1.33	-0.22	Avg Offset	0.65	0.19	0.93	Avg Offset	9.91	0.76	-0.25
Avg Apex	0.00	0.00	1.73	Avg Apex	0.00	0.00	0.58	Avg Apex	0.00	0.00	0.23	Avg Apex	0.00	0.00	0.40	Avg Apex	0.00	0.00	-2.32
Avg Corner	4.07	4.07	0.00	Avg Corner	-0.65	0.65	0.00	Avg Corner	0.54	0.54	0.00	Avg Corner	1.11	1.11	0.00	Avg Corner	-6.38	6.38	0.00
3-1	x	y	z	3-2	x	y	z	3-3	x	y	z	3-4	x	y	z	3-5	x	y	z
Avg Offset	1.93	1.45	-1.17	Avg Offset	0.13	1.83	-1.77	Avg Offset	0.62	-0.75	-0.01	Avg Offset	1.80	-0.26	0.62	Avg Offset	11.22	1.50	1.20
Avg Apex	0.00	0.00	0.95	Avg Apex	0.00	0.00	0.60	Avg Apex	0.00	0.00	0.18	Avg Apex	0.00	0.00	0.28	Avg Apex	0.00	0.00	-1.53
Avg Corner	2.62	2.62	0.00	Avg Corner	-0.85	0.85	0.00	Avg Corner	0.49	0.49	0.00	Avg Corner	0.77	0.77	0.00	Avg Corner	-4.19	4.19	0.00
4-1	x	y	z	4-2	x	y	z	4-3	x	y	z	4-4	x	y	z	4-5	x	y	z
Avg Offset	2.12	1.62	-1.49	Avg Offset	1.04	1.10	-1.36	Avg Offset	2.52	-0.54	0.10	Avg Offset	0.92	-0.26	0.57	Avg Offset	8.71	-0.70	0.06
Avg Apex	0.00	0.00	0.36	Avg Apex	0.00	0.00	0.30	Avg Apex	0.00	0.00	0.17	Avg Apex	0.00	0.00	0.24	Avg Apex	0.00	0.00	-2.20
Avg Corner	0.99	0.99	0.00	Avg Corner	0.03	0.03	0.00	Avg Corner	0.47	0.47	0.00	Avg Corner	0.65	0.65	0.00	Avg Corner	-6.03	6.03	0.00
5-1	x	y	z	5-2	x	y	z	5-3	x	y	z	5-4	x	y	z	5-5	x	y	z
Avg Offset	2.71	1.50	-1.28	Avg Offset	0.55	0.98	-1.04	Avg Offset				Avg Offset	1.44	-0.57	0.55	Avg Offset	16.23	-0.61	1.09
Avg Apex	0.00	0.00	0.35	Avg Apex	0.00	0.00	0.48	Avg Apex	Unable to view this area			Avg Apex	0.00	0.00	0.36	Avg Apex	0.00	0.00	-2.56
Avg Corner	0.96	0.96	0.00	Avg Corner	1.08	1.08	0.00	Avg Corner				Avg Corner	1.00	1.00	0.00	Avg Corner	-7.03	7.03	0.00

Table 5.2: Deviation (in mm) of Pyramid Offset, Apex Location, and Corner Location in various areas within the stereo depth camera's field of view. The camera was mounted to the probe of the CMM. By mounting the camera to the probe of the CMM, the camera was moved very accurately relative to the pyramid. This gave the same results as moving the pyramid part accurately to different positions within the camera's view. Locations are labelled as "1-1" for the top left corner, to "5-5" for the bottom right corner. Green values are closer to zero error, red values are further away from zero error. The yellow box represents the area with the best results, making up 70% of the camera's field of view.

Maximum Overall Offset	Minimum Overall Offset	Maximum Apex Offset	Minimum Apex Offset	Maximum Corner Offset	Minimum Corner Offset
16.280	0.974	−2.560	0.173	7.034	0.029
Found at position 5-5	Found at position 3-3	Found at position 5-5	Found at position 4-3	Found at position 5-5	Found at position 4-2

Table 5.3: Maximum and minimum deviation (in mm) of Pyramid Overall Offset, Apex Location, and Corner Location. Overall offset was found by the vector length created by the XYZ error.

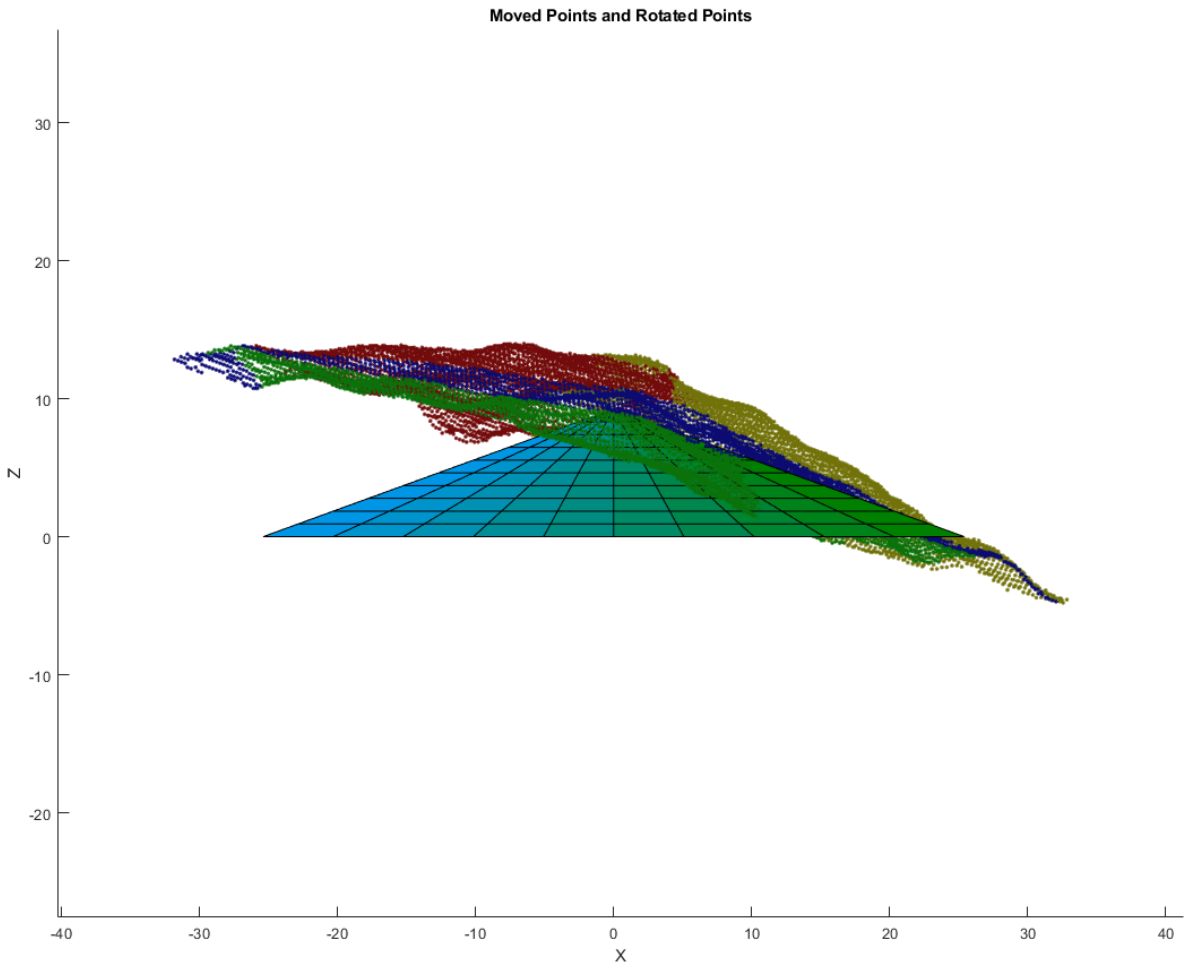
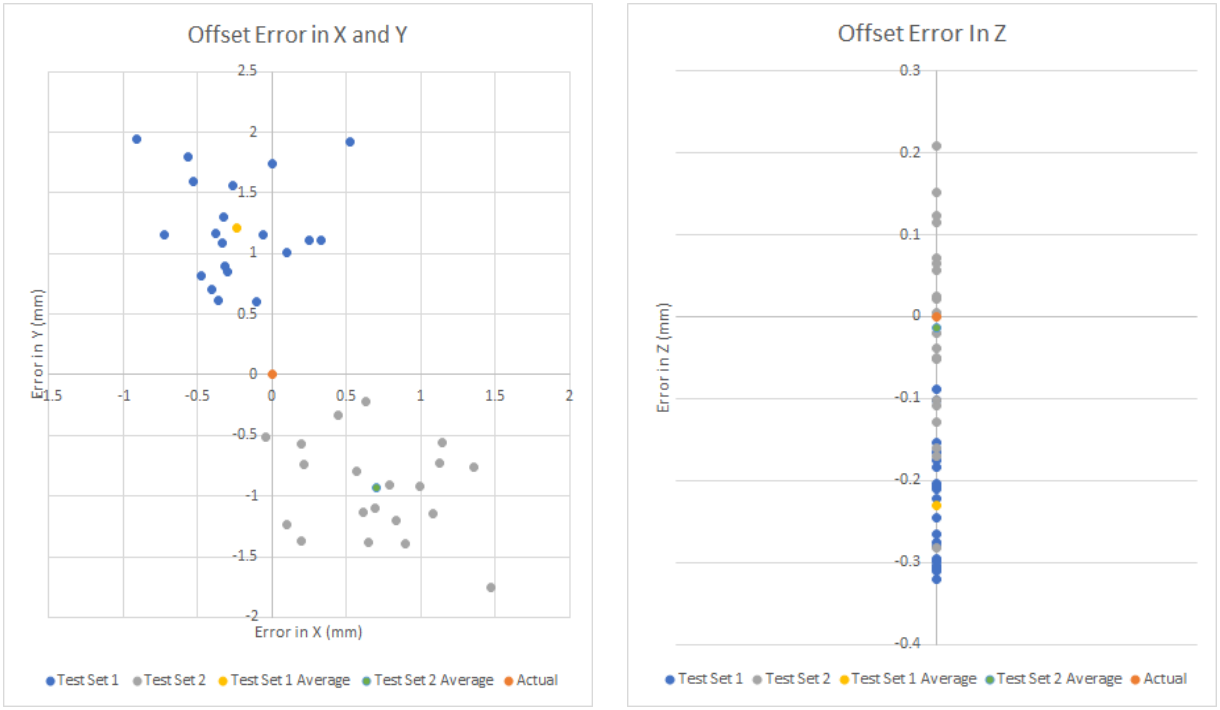


Figure 5.3: Moved and rotated points for position 1-5 after redistributing points and finding planes.

The key metrics observed are the apex height and corner positions to inform the part's geometry and the offset to inform the position of the part. Overall, the most accurate results for both the geometry and offset of the part were found when the part was positioned closer to the center of the camera's field of view, outlined by the yellow box around the center nine positions. The smallest overall positional offset was found in the center position. In contrast, the most accurate apex height was found just below the center, and the most accurate corner position was found just to the bottom left of the center. Conversely, the most significant errors were found around the edges, specifically the right edge and bottom right corner. However, better results tended to be found in the center of the field of view, making up approximately 70% of the camera's field of view. These results are shown in Table 5.3

The position offset precision can vary and is determined by the location of the part within the camera's field of view. Two separate test sets of 20 scans each were taken. Both tests had the part roughly positioned around the center of the image. However, while the ranges of both tests' offset values were similar, the tests produced two different sets of values, shown in Figure 5.4. In Test Set 1, the offsets in X and Y ranged from approximately -1 to 0.5 mm in X and 0.5 to 2 mm in Y, Test Set 2 had its offsets in X and Y range from approximately 0 to 1.5 mm in X and -1.75 to -0.25 mm in Y. The precision was around 1.5 mm in both tests, while the sets of values for both tests differed, shown in Figure 5.4a. In Z, Test Set 1 had an offset between -0.3 and 0 mm, whereas Test Set 2 ranged between -0.3 and 0.2 mm, shown in Figure 5.4b.

The rotation of the pyramid part observed by the stereo depth camera was determined to be precise and accurate. When the part was rotated around the Z axis running through the apex by a known amount, the rotation error ranged between -1.6 to 0.7 degrees, averaging around -0.7 degrees, shown in Figure 5.5. The rotational accuracy greatly depended on how well the pyramid faces were fit. This is due to the equation of the plane representing the faces being used to determine the rotation. The plane fitting was not as effective around the edges of the camera's field of view and was better towards the center. Therefore, the rotation of the pyramid could be more accurately observed by the camera when the pyramid was located around the center of the camera's field of view.



(a) (b)

Figure 5.4: (a) Offset error in X and Y from two sets of 20 scans, (b) Offset error in Z from two sets of 20 scans. These show the average accuracy and overall precision capabilities of the depth camera and algorithm for finding the position of the Pyramid in XYZ

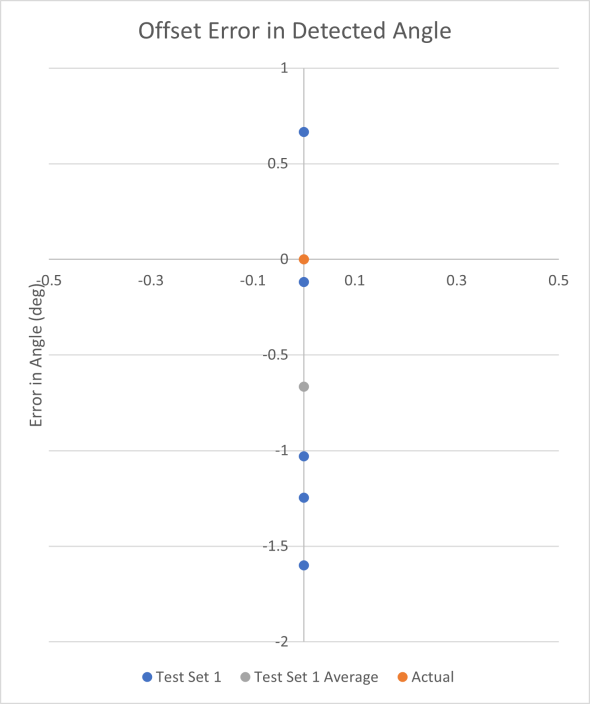


Figure 5.5: Offset error (in degrees) of pyramid rotated about the Z axis. This shows the average accuracy and overall precision capabilities of the depth camera and algorithm for finding the rotation applied to the Pyramid about the Z axis running through normal to the bottom of the pyramid and through the apex

Position 1-5 Max Error	Position 1-5 Min Error	Position 1-5 Avg Error	Position 3-3 Max Error	Position 3-3 Min Error	Position 3-3 Avg Error
2.01	−5.78	−1.28	2.42	−1.57	0.48

Table 5.4: Maximum, minimum, and average distance (in mm) between aligned points and desired pyramid planes for scans taken at locations 1-5 and 3-3.

5.3.3 Plane Fitting

After the points had been translated and rotated to best align with the desired orientation of the pyramid, the distance between the newly aligned points and the desired planes of the pyramid were measured. Table 5.4 shows the maximum, minimum, and average distance between the aligned points and the desired planes. The distance was calculated using the equation of the plane and the points, which is shown in Equation 5.1, where $[A, B, C, D]$ defines the equation of the plane in the form $Ax + By + Cz + D = 0$, and $[x_p, y_p, z_p]$ is the coordinates of the point to be measured.

$$Distance = \frac{Ax_p + By_p + Cz_p + D}{\sqrt{A^2 + B^2 + C^2}} \tag{5.1}$$

The aligned points are plotted with the desired planes. Figure 5.6, Figure 5.7, and Figure 5.8 show the results of the plotted points and desired planes when the part is in the center position 3-3. The points are relatively evenly distributed above and below the desired planes. On average, for position 3-3, the distance between the points and the plane is less than 0.5 mm.

The plane fitting is completed after the points are transformed. Therefore, a similar correlation is found between the location of the part within the camera's field of view and the error in plane fitting. Parts scanned at the edge of the camera's field of view, such as location 1-5, have lower precision and worse accuracy than parts scanned at the center of the camera's field of view, such as location 3-3. Location 1-5 has point distances between −5.78 mm and 2.01 mm compared with location 3-3, with point distances between −1.57 mm and 2.42 mm. Additionally, the average point distance is much more accurate at

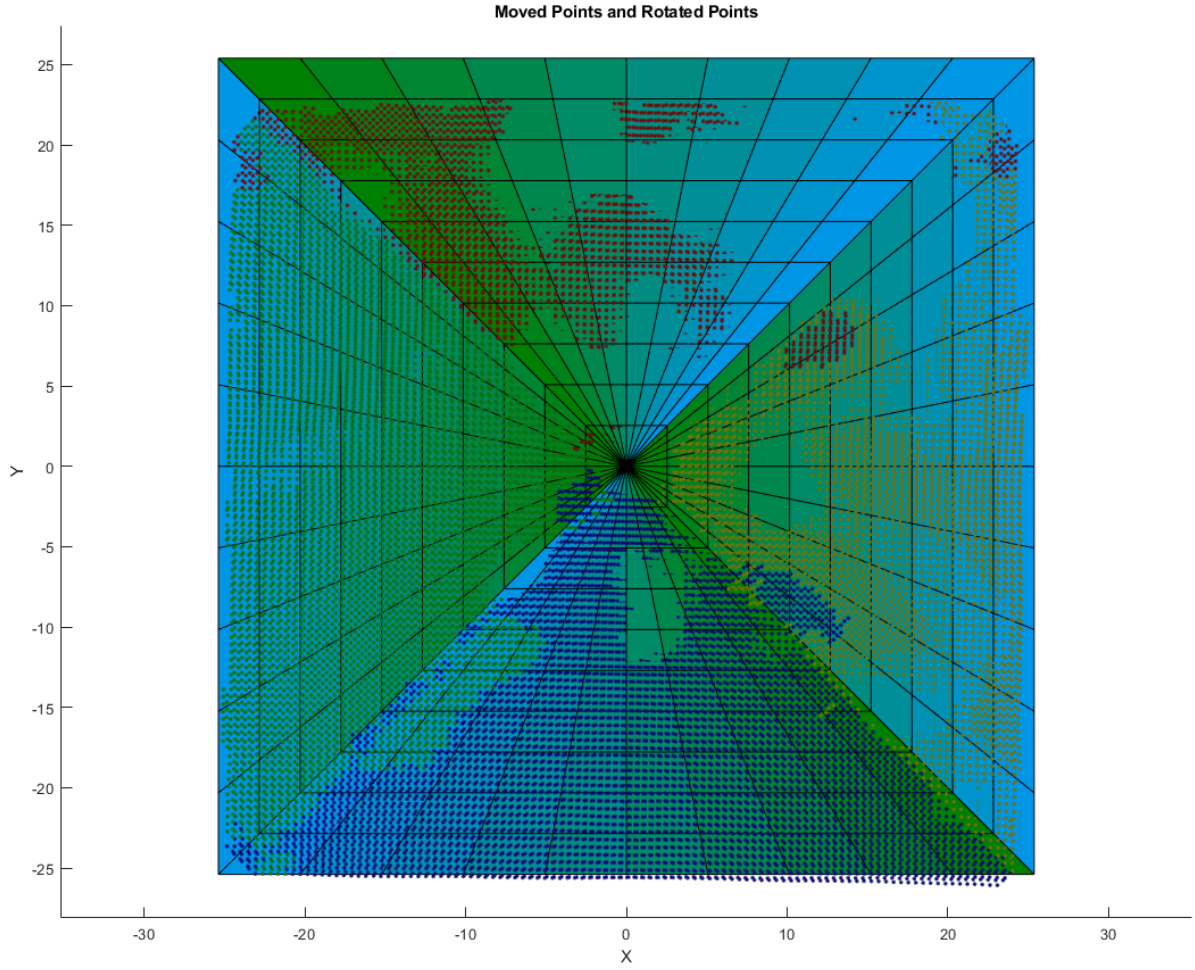


Figure 5.6: Top view of fitted points and desired planes for pyramid at location 3-3

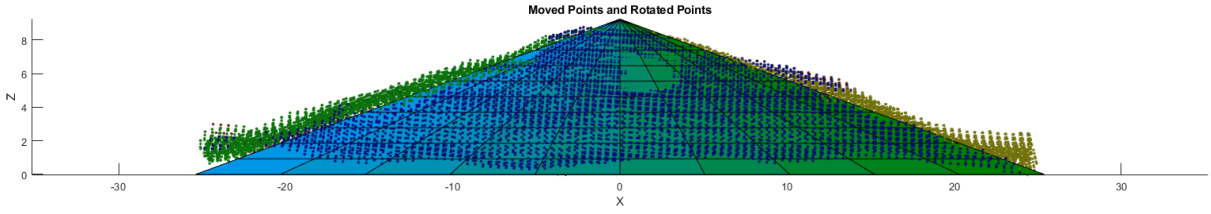


Figure 5.7: Side view of fitted points and desired planes for pyramid at location 3-3

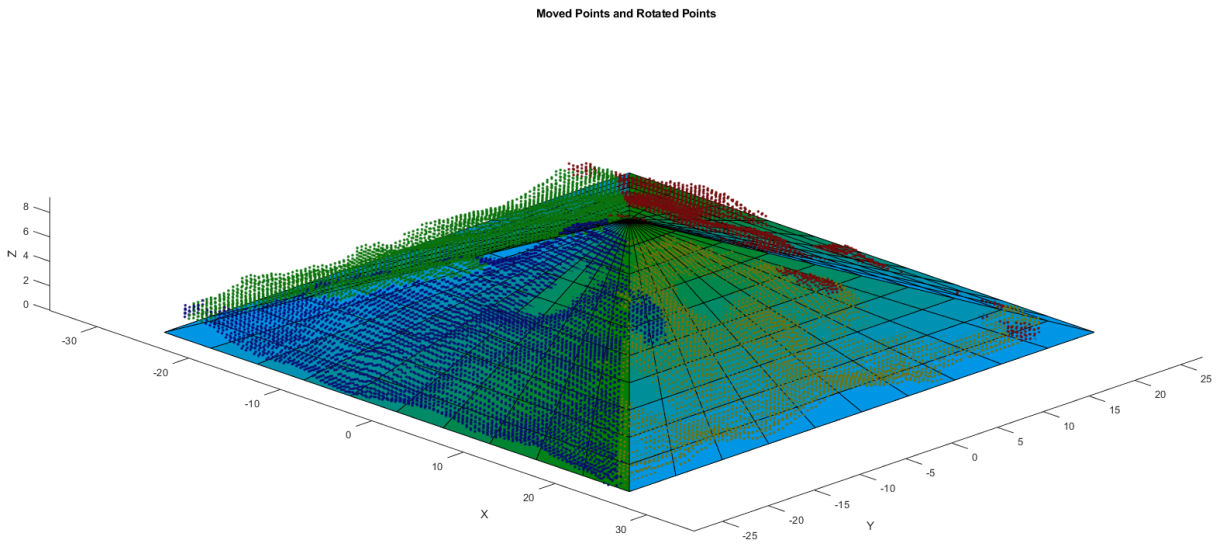


Figure 5.8: Orthographic view of fitted points and desired planes for pyramid at location 3-3

location 3-3, less than 0.5 mm from the desired plane, compared with location 1-5, which is around -1.28 mm from the desired plane.

5.3.4 Algorithm Comparison with Existing Software

The results were compared against another software with a similar alignment method, Geomagic. Geomagic is a professional engineering software for scanning, CAD, inspection and metrology. It is used in this research project for aligning the scanned point clouds with the desired part model and provides information on the transformation required to align the point cloud with the model and the distance between the points and the surface they are meant to represent.

The first step towards aligning the scanned point clouds and the desired model involves the transformation of the points. Two tests were used for the comparison that included the pyramid part at positions 1-5 and 3-3. These were used to additionally observe the impact of the location of the pyramid within the stereo depth camera's field of view. The

	Position 1-5 X Offset Error	Position 1-5 Y Offset Error	Position 1-5 Z Offset Error	Position 3-3 X Offset Error	Position 3-3 Y Offset Error	Position 3-3 Z Offset Error
After alignment	9.42	1.43	0.14	0.62	−0.75	−0.01
After Geomagic	7.829	−4.861	4.957	1.028	0.4806	−1.615

Table 5.5: Comparison of alignment error (in mm) for XYZ between the proposed algorithm and Geomagic for scans taken at locations 1-5 and 3-3.

results from both the proposed algorithm and Geomagic are presented in Table 5.5. Overall the proposed algorithm appears to show improvements over the more generic alignment algorithms that Geomagic uses for determining the position of the pyramid. However, for both the proposed algorithm and Geomagic, the pyramid that was positioned in location 3-3 gives better results. This can further enforce the importance of positioning parts to be scanned in the center of the stereo depth camera's field of view.

Once the scanned point clouds are aligned, the final fitting of the points with the surface can be determined. The same two test locations were used, 1-5 and 3-3. The results of the maximum and minimum distances between the points and the desired pyramid planes are shown in Table 5.6. Overall Geomagic shows better results shown in Figure 5.9, with distances ranging between −1.37mm and 1.37mm at the center position 3-3 compared with the proposed algorithm with distances ranging between −1.57mm and 2.42mm. However, Geomagic did trim out some noisy points to get these results. Without the trimming, test 3-3 would have point distances ranging between −3mm to 3mm, which would be slightly worse results than the proposed algorithm, which also does not trim any points. Additionally, it appears that even when the pyramid was positioned at corner position 1-5, Geomagic found the points to be better fitted compared to the ideal pyramid location in the center of the camera's field of view. However, the better fitting may also be due to the trimming of noisy points.

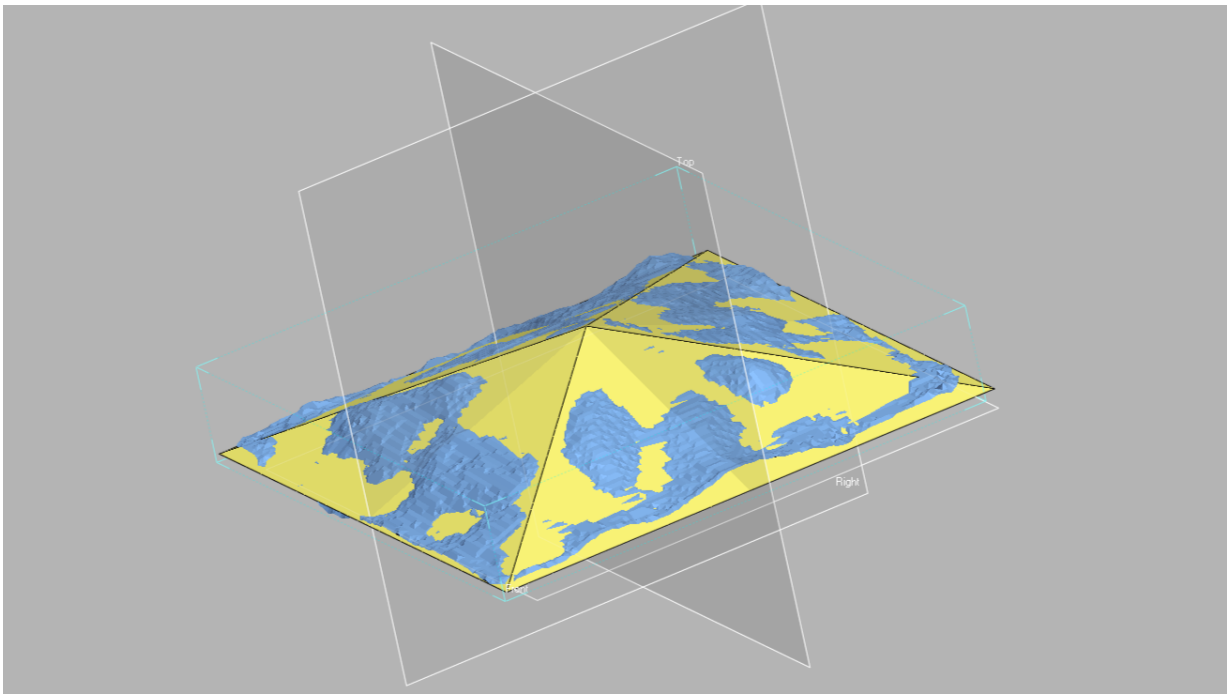


Figure 5.9: Fitted points and desired planes for pyramid at location 3-3 in Geomagic.

	Position 1-5 Max Error	Position 1-5 Min Error	Position 3-3 Max Error	Position 3-3 Min Error
After alignment	2.01	−5.78	2.42	−1.57
After Geomagic	1.18	−1.18	1.36	−1.37

Table 5.6: Comparison of maximum, minimum, and average distance (in mm) between aligned points and desired pyramid planes between the proposed algorithm and Geomagic for scans taken at locations 1-5 and 3-3.

Overall, the proposed algorithm does appear to give similar or better results compared with Geomagic, an existing software that is designed for similar applications. The proposed algorithm is specifically designed to work with pyramid-shaped parts, which would explain the improvements in part offset that the proposed algorithm finds. However, Geomagic benefits from trimming noisy points, leading to overall better results when considering the distances between scanned points and the desired planes.

Chapter 6

Conclusion and Future Opportunities

In this work, a method was developed for testing the viability of using the stereo depth camera for observing a part within a CNC workspace. A pyramid part was chosen to be tested. The Intel RealSense D405 stereo depth camera was used, which alongside a calibration plate with fiducial markers, was first used to determine a transformation to align the camera's coordinate system with the calibration plate and CNC/CMM. The stereo depth camera would then scan the work volume and pyramid part to create point clouds, which were transformed by the calculated transformation to roughly align the point clouds. A proposed algorithm for fine-tuning the alignment was then applied to the roughly aligned point clouds to determine the pyramid part position, orientation, geometry and plane fitting.

The depth camera has advertised object detection of 0.1mm at 7cm [22] in ideal conditions. In the method proposed, the camera and algorithms were able to determine pyramid position and geometry within or less than 1mm accuracy and orientation within 1 degree accuracy in experimental conditions. The points were on average 2mm away from the desired pyramid faces, with good distribution above and below showing a reasonable fitting with the desired planes. These results are comparable with the claimed camera accuracy provided by Intel. Additionally, when compared with existing software, the proposed algorithm had much better results for determining the position of the pyramid part but performed much worse when fitting the points with the desired plane. This showed the

potential for this and similar algorithms for determining the position, orientation and geometry of parts with well-defined faces.

However, the resulting accuracy found would not allow this solution to be confidently used for autonomously determining if a part has been machined properly within a reasonable tolerance for most machined parts. The results could be used to show that a part roughly matching the geometry of the desired model has been manufactured, and could determine if the raw stock was positioned somewhat correctly. In addition, any large errors or obstructions would be easily detected by this system. The largest bottleneck holding back the potential of this project is the point cloud quality scanned by the stereo depth camera. At present, the scans have a large amount of noise. More powerful and expensive scanning methods or future improvements to the technology may improve the results found in this work.

There is much future scope within this domain. Using multiple cameras to scan the part from different angles at the same time and combining the point clouds to get a fuller representation of the scanned part may allow these scanning methods to give improved results or be used in four or five-axis machining. Additionally, in a similar way to how the part was cropped using OpenCV to determine where the part was within the image, undesired geometry could be better cropped from the point cloud, improving the scan quality. This could include part of the CNC machine that block part of the field of view, or chips created during machining or coolant, allowing the scanning to give better results when occurring during the machining operations. A similar method of scanning could also be used to detect the position of mounting fixtures or obstructions to inform automated g-code generation to avoid those areas.

Large improvements could also be made to the quality of the scans. Shadows, overexposure and the effects of lighting in general could be improved using methods such as those suggested by McDermott *et al.* [13]. Additionally, the point cloud scans could be encoded using voxels, which would reduce error and could be used with machine learning for feature detection such as VoxNet [12].

I envision a system where an improved version of my proposed solution could use aligned scans to compare with either the expected raw stock, a simulated model of the part during

its machining, or the final model to determine if the part has been appropriately machined within a certain tolerance. The system will be capable of autonomously monitoring the machining process and when it detects any flaws during the machining, it will be able to stop the machine and inform a technician that an issue has occurred.

References

[1] M. Bengtsson and M. Kurdve, "Machining equipment life cycle costing model with dynamic maintenance cost," *Procedia CIRP*, vol. 48, pp. 102–107, 2016. The 23rd CIRP Conference on Life Cycle Engineering.

[2] M. M. U. P.E., B. A. Stufflestreet, and K. V. Johnson, "Identifying the need for trained machinists in the greater tri-cities area: A survey of employers to evaluate the future of machining," in *2021 ASEE Virtual Annual Conference Content Access*, (Virtual Conference), ASEE Conferences, July 2021. https://peer.asee.org/37271.

[3] G. Poon, P. J. Gray, S. Bedi, and D. E. Miller, "Architecture for direct model-to-part cnc manufacturing," *Journal on Systemics, Cybernetics and Informatics*, vol. 4, pp. 14–18, 2006.

[4] Modern Machine Shop, "5 ways automated on-machine probing improves productivity." https://www.mmsonline.com/articles/5-ways-automated-on-machine-probing-improves-productivity, Jan 2019.

[5] R. Horaud, M. Hansard, G. Evangelidis, and C. Ménier, "An overview of depth cameras and range scanners based on time-of-flight technologies," *Machine Vision and Applications*, vol. 27, pp. 1005–1020, Oct 2016.

[6] V. Tadic, Odry, I. Kecskes, E. Burkus, Z. Király, and P. Odry, "Application of intel realsense cameras for depth image generation in robotics," *WSEAS Transactions on Computers*, vol. 18, pp. 107–112, 09 2019.

[7] G. Budzik, "The selection criteria of scanning method in process of reproducing an aircraft engine blade geometry," *Archiwum Technologii Maszyn i Automatyzacji*, vol. Vol. 27, nr 2, pp. 63–70, 2007.

[8] M. Pajor and M. Grudziński, "Intelligent machine tool – vision based 3d scanning system for positioning of the workpiece," *Solid State Phenomena*, vol. 220-221, pp. 497–503, 01 2015.

[9] T. Madeira, M. Oliveira, and P. Dias, "Enhancement of rgb-d image alignment using fiducial markers," *Sensors*, vol. 20, no. 5, 2020.

[10] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, p. 2280–2292, 06 2014.

[11] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3400–3407, 2011.

[12] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 922–928, 2015.

[13] M. McDermott and J. Rife, "Mitigating shadows in lidar scan matching using spherical voxels," *IEEE robotics and automation letters*, vol. 7, no. 4, pp. 12363–12370, 2022.

[14] A. Grunnet-Jepsen, J. N. Sweetser, and J. Woodfill, "Stereo depth cameras for mobile phones." https://dev.intelrealsense.com/docs/stereo-depth-cameras-for-phones, 2020.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[16] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[17] Intel Corporation, "Intel realsense sdk 2.0." https://github.com/IntelRealSense/librealsense, 2016.

[18] C. J. Taylor and D. J. Kriegman, "Minimization on the lie group so(3) and related manifolds," 1994.

[19] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, (Shanghai, China), May 9-13 2011.

[20] The Math Works Inc., "Matlab version: 9.13.0 (r2022b)." https://www.mathworks.com, 2022.

[21] A. Grunnet-Jepsen and D. Tong, "Post-processing filters." https://dev.intelrealsense.com/docs/post-processing-filters, 2018.

[22] Intel Corporation, "Depth camera d405." https://www.intelrealsense.com/depth-camera-d405/, Jun 2023.

[23] A. Grunnet-Jepsen, J. N. Sweetser, P. Winer, A. Takagi, and J. Woodfill, "Projectors for d400 series depth cameras." https://dev.intelrealsense.com/docs/projectors#5-the-dot-pattern, 2023.